Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D8.1.1**
Title of the deliverable:	**Multi-sensory Scene Exploration**

Contractual Date of Delivery to the CEC:	31 January 2007
Actual Date of Delivery to the CEC:	30 January 2007
Organisation name of lead contractor for this deliverable:	AAU

Author(s): Norbert Krüger, Florentin Wörgötter, Tamim Asfour, Rüdiger Dillmann, Justus Piater, Mark Steedman, Aleš Ude, Danica Kragic, Bernhard Hommel, Joyca Lacroix, Pascal Haazebroek, Alex Bierbaum, Dirk Kraft, Morten Kjaergaard, Sinan Kalkan, Nicolas Pugeault, Christopher Geib, Ronald Petrick, and Renaud Detry
Participant(s): AAU, BCCN, KTH, JSI, UniKarl, CSIC, UEDIN, UL

Work package contributing to the deliverable:	WP1, WP2, WP4.3, WP4.1, WP5.2
Nature:	R
Version:	Draft
Total number of pages:	17
Start date of project:	1st Feb. 2006 **Duration:** 48 month

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)	

Abstract:

This technical reports describes the work the PACO consortium has done in WP4.1 in the first 12 months. We will describe first modules of a vision based robotic system that is able to ground high level entities (formulated in LDEC, see WP4.1 and WP5) in the real world by means of procedural definitions in terms of robot action and percepts. The system, being equipped with a powerful vision system, knowledge about its own body, and a set of basic reflex–like behaviors is able to grasp objects without making use of any model-knowledge, detect objectness of the thing being grasped and to establish a multi-modal representation of it that then leads to a first message to the LDEC level. We also give perspectives how we want to develop the system in the remaining time of the project.

Keyword list: LDEC, Learning of Object Action Representations, Grasping

Table of Contents

1. Introduction

Note: Here, we rather briefly describe different kinds of work that are relevant in the context of the tasks 8.1.1, 8.1.2 and 8.1.3 with the intention to present the 'red thread'. These works are described in more detail in accepted [A, C, F, I, J] or reports [B, D, E, G, H, K] that will be the basis for future submissions.

Demo 1 aims at the learning of representations of objects and associated actions (OACs) that occur in the robot world by means of exploration. These representations are supposed to allow for the generation of plans by means of a high-level logical language (LDEC) [34, 35] (see also DeliverableD5.1). In the beginning (and in contrast to demo 2) we do not assume any interaction with humans but we aim at a purely explorative approach. However, we will point to the need and the potential of integrating the two scenarios in the further development of the project.

At the core of demo 1 is the grounding of high level entities on the LDEC level in the real world by means of procedural definitions in terms of robot action and percepts. In the beginning, the system is equipped with a powerful and rich vision system and a set of reflex-like actions that trigger the establishment of grounded world knowledge in terms of OACs. Furthermore, there is a pre-defined link to a high-level representation, described in terms of an LDEC representation that becomes grounded and extended during the exploration process. In other words, this link describes how the high-level representation is "induced" from the robot vision system during the exploration process: real-world concepts provided by the vision system are abstracted into LDEC rules that model the believed dynamics of the world, permitting high-level plan generation using the LDEC representation.[1]

Our system may learn about the world in terms of different levels of complexity. For example, on a rather low level it might learn to improve a specific grasp applied to a specific object or, on a much higher level, it might learn about the consequences of actions in a symbolic state space. In this context, we want to stress the necessity of different ways of learning on the different levels of processing:

- Reflex driven acquisition of new rules (see, e.g., Section 6)
- Refinement of OACs by statistical estimation of the outcome, or adapting action parameters to increase the success rate of OACs (see, e.g., Section 4.3).
- Learning about affordances of actions and the consequences of actions within the high level domain (Section 5.3).
- Learning supported by different forms of supervision and imitation. Note that in particular these aspects lead to an efficient merging of demo 1 and demo 2 (see Section 2).

After learning, the system will be able to act more successfully and precise. Moreover, the acquired world knowledge will enable it to generate plans and by that, to substitute initial hard-wired behavior by more goal directed actions.

The scenario we chose is a kitchen environment and tasks we want the system to find plans for and perform are of the type of 'clearing a table' or 'emptying a dry rack'. In the consortium there exist at least 4 hardware platforms of different complexity. The most elaborated hardware platform at UniKarl consists of a humanoid

[1] Our current strategy of linking low-level, online codes directly to high-level symbolic representations allows for a first exploration of the interaction between higher and lower levels, and for integrating the work on Demo 1 and Demo 2. However, processing more complex perceptual events and executing more complex, "intentional" actions call for a more complex cognitive architecture. Accordingly, we are currently working on extending our cognitive systems to include a mid-level representational layer. It will include an informationally rich episodic memory collecting relatively abstract, but still subsymbolic perceptual instances, and a symbolic semantic memory, which contains higher level rules extracted from episodic memory [11, 24]. First elements of this mid-level architecture have been identified and explored [5, 12, 13, 17, 18] and will be integrated in the system discribed here in the next phase of the project.

robot with an active vision system, a human like arm, and a five finger hand equipped with tactile sensors. The platform at AAU/SDU consists of an industrial robot arm with a static stereo system. Since the working area of demo 1 is rather narrow and since the robot arm is rather precise in the AAU/SDU scenario, actions can be done largely in the space of generated 3D features without any servoing. The AAU/SDU system has a two finger grasper with tactile sensors. Other platforms exist at KTH and JSI. Most of the experiments described here are made on the AAU/SDU platform but of course the final aim is to transfer these to the platform at UniKarl. More details can be found in Deliverable D1.2.

The existing state of the system is a set of (partly linked) sub-modules that have been described in particular in Deliverable D4.1.1, that become connected to the planning level. In contrast to Deliverable D4.1.1, in Deliverable D8.1.1, we look at these sub-modules in the context of a cognitive agent.

Note that some of the modules are already integrated (e.g., the early cognitive vision system [H], the grasping reflex [A] and the object learning [I]) resulting in a first message for the LDEC level. For some sub-modules specifications for integration have been made (see [D]). Also, since in the consortium there exist multiple platforms (with the most elaborated platform being at UniKarl) some sub-modules are tested only on a subset of platforms with partly different constraints (see Section 4).

The deliverable is structured as follows. In Section 2 we give some pointers to work related to ours. In Section 3, we introduce the idea of OAC-trees to work with processes of explorative behavior. In Section 4, we describe the prior use of the robot/vision system. We elaborate on that because we think that this is a distinguishing feature to other approaches. In Section 5, the initial LDEC representation in the context of the specific scenario is outlined. In Section 6, we give a procedural definition of objects and the transfer of information to the planning level. In Section 7, we described planning on the LDEC level and the backtransfer of information to the robot-vision system. Note that the implementation process for the last point has just started, but that we think it is important to point to the specifications already.

2. Comparison to other Approaches

Here we do not intend to give a broad literature review but to put our approach into a context and describe the distinguishing feature to other related approaches.

The idea of taking advantage of active components for vision is in the spirit of active vision research (see, e.g., [1, 30]). The grounding of vision in cognitive agents has been addressed for example by [7]. Related attempts are described in e.g., [22, 23]. The idea of grounding language in the interaction of agents, which present the link to demo 2, has been formulated in, e.g., [36].

The work of [7] is the most related one to our approach since the overall goal is the same: Finding out about the relations of actions and objects by exploration. We see the main distinguishing feature of this approach (and also [22, 23]) to our approach in the amount of pre-structure we use. For example, we assume a much more sophisticated vision system compared to, e.g., [7], that covers multiple visual modalities in a condensed form as well as visual relations defined upon them. Similar to [7] we assume first 'reflex-like' actions that trigger exploration. However, since in our system the robot knows about its body and the 3D geometry of the world these reflexes can make use of more complex visual events. Furthermore, from the very beginning the robot/vision system is linked to a high-level AI planning system that is able to compute plans and do reasoning in an abstract state space.

The use of LDEC as a high-level representation language follows in the tradition of logical languages inspired by the situation calculus [20]. LDEC has the added advantage, however, that it incorporates into its semantics a STRIPS-style treatment of fluent change [6], making it suitable for modelling planning domains. Our current approach to high-level plan generation takes advantage of this close correspondence

by building on the extended-STRIPS planner PKS [27], which is capable of constructing conditional plans under conditions of incomplete knowledge and sensing.

Our approach also shares some of the ideas concerning "continuous planning" in [4]. For instance, the use of knowledge variables and assertions to manage knowledge requirements for plan-time actions closely resembles the use of known functions in [27] as a form of "run-time variable". An important difference in [4] is that (re-)planning, execution, and sensing are interleaved as an integral part of the planning process, while we propose a different control structure between the robot/vision system and high-level planner.

The work of [25, 44] also addresses the problem of learning STRIPS-style action rules in 3D robot/vision environments. While we are currently interested in learning standard STRIPS actions, [25, 44] offers a way of modelling probabilistic rules that we intend to investigate further.

For addressing complex tasks such as vision and multi-sensorial exploration we need to put into the system a certain amount of prior structure We are also aware that high-level planning systems are often said to be restricted by acting in a rather specified and 'limited world'. However, in this work we want to show that

- categories such as different kinds of objects can be detected and included into the system and that the parameterization of properties of objects can be found out by exploration,
- affordances and consequences of actions can be learned through exploration,
- the LDEC formalism can extended by new actions, rules and affordances using strategies such as linking primitive actions to more complex action chains or to learn specific actions and their consequences by 'guided exploration' through supervision and imitation. In this context, there is a high potential in the complementary aspects of demo 1 and demo 2 by making use of capability of imitating actions (as aimed at in demo 2) with for learning new categorioes in the LDEC framework.

3. OAC-trees guiding the Exploration Process

The goal of every cognitive agent must be to discover (with or without help) the structure and the rules of its world in conjunction with its own embodiment. To this end several complex processes are required and we suggest a certain type of diagram (OAC-tree, Fig. 1) as a helpful tool for structuring such a cognitive process. We are aware that this is at the moment a tool-in-the-making to get a first handle onto the required algorithmic procedures for attaching attributes to Objects, for manipulating OACs, and for discovering new OACs, etc. The main focus of this tree diagram is to define the different required sub-processes when trying to implement a cognitive process into a robot. We note, OAC-trees are not decision trees or planners. They are meant to be a first diagrammatic step towards visualizing and implementing a reasoning and learning process, which could lead to cognitive properties in an agent. A more detailed description is given in [K].

The OAC-tree operates with built-in *perceptual preconditions* as well as *action preconditions* (Reflexes), where we could assume that these preconditions have been earlier acquired by the machine itself. Furthermore we assume that the (ancient) *Law of Cause and Effect* [38] can be used as one of the most reliable driving forces of any cognitive process: If a certain percept triggers a certain reflex (chain of events) and if this leads to a reproducible and perceivable change in the world then this can be stored as a candidate for an OAC. Clearly we also require a learning procedure (here depicted as supervised learning via instructions).

OAC-trees

In the given example, we start with a certain knowledge base namely: *any-object affords filling*. Hence the agent can perform a filling reflex by grasping a (different) container and turning it over, above the *any-object* entity. Clearly we assume that the agent can also perform a turning reflex of its hand/arm (leading to the action of emptying). In the beginning the only existing behavioural repertoire is described by *any-object*

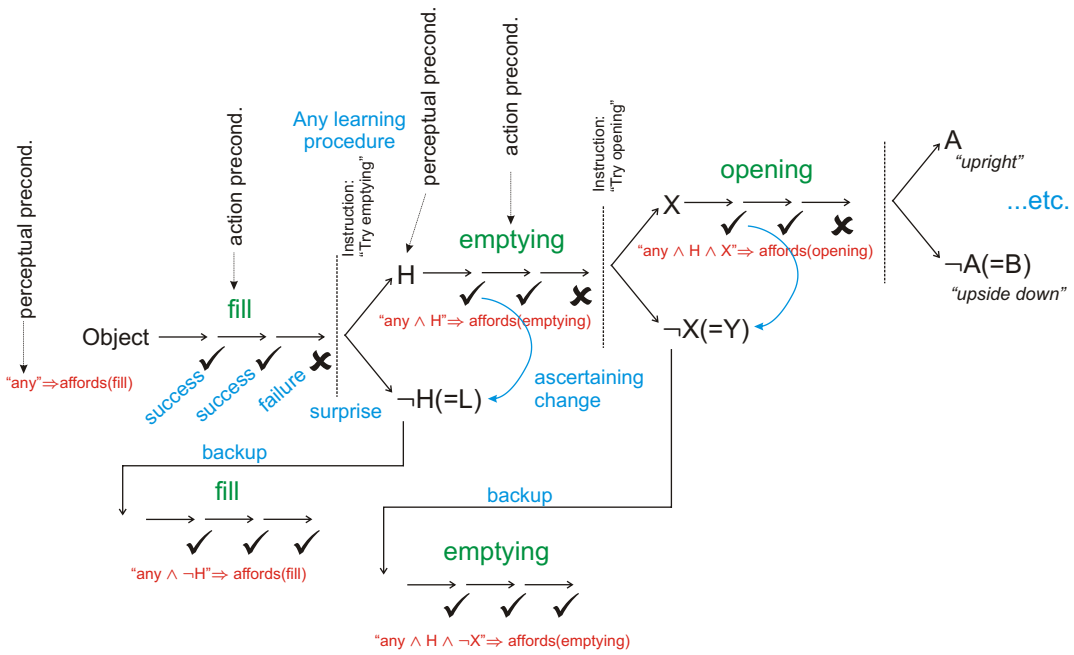


Figure 1: OAC Tree Diagram

affords filling. The OAC-tree (left side) shows that the agent will perform this three times, twice with success and a third time failing leading to surprise and the necessity to resolve the surprise. This can be achieved by any learning procedure. Here we use supervision and the agent is being told to try the emptying reflex. After doing this the agent needs to ascertain change: which of all possibly changing percepts is causally related to the performed emptying action? The agent could now try to back-up and perform the filling action next. If backup is successful the agent could conclude that there is an additional attribute (or correlated set of attributes) to the *any-object*, which makes it fillable and this attribute is the one (or set) that has been changed through emptying. The same branching process continues if the agent encounters another surprise (e.g. a closed object).

Note, this brief description leaves out the actual specification of the required subprocesses (light blue in the tree diagram), for example *success, failure, surprise, learning, ascertaining change, backup, etc.* which is the most important aspect of an OAC-tree. For this see [K].

4. Initial state of robot/vision system

A truly cognitive system neither is a completely hard-wired structure nor a 'blanc table' or 'tabula rasa'. Assaid above, learning faces the *bias-variance dilemma* (see, e.g., [8]), As a consequence of this dilemma, Geman and Bienenstock [8] argue that a certain amount of "bias needs to be *designed* to each particular problem". However, each concrete choice of *a priori* knowledge is a crucial point: A wrong choice may lead to the exclusion of good solutions in the search space. A choice of predetermined structural knowledge that is too restricted may result in an increase of the search space, leading to unrealistic learning time and bad generalization.

Within a biological system, bias can be established by genetic coding. The question of predetermined

components is also most essential for the design of any artificial visual system that is able to learn, since this predetermined knowledge helps the system to focus on essential aspects in the huge amount of data it has to cope with. In the following subsections we make explicit what prior knowledge we assume in our system on the level of the visual representations (Section 4.1), on basic behavioral components (Section 4.2 and Section 4.3) as well as on internal high level representations (Section 5).

4.1 Early Cognitive Vision: Local 3D features and relations

We assume a rich feature processing in terms of an early cognitive vision system [H] being in place (see also Deliverable 4.1.1). This system represents information about multiple visual modalities (such as 2D and 3D position and orientation, phase, colour, local motion) in a condensed way. Furthermore, basic concepts of 3D space as well as knowledge about the transformation of rigid bodies in that space are known to the system. There is psychophysical as well as neurophysiological evidence for these assumptions that are discussed for example in [41, 14, 16].

4.2 Body Knowledge and haptic Sensors

We also assume here an innate representation of the robot's body that is aligned with its visual representation. This is a first approximation that follows the suggestion from developmental psychologists that human perception and action learning might be guided by a genetically prespecified intermodal matching scheme [21]. In the scenario at AAU/SDU with an industrial robot with high precision and static cameras in a very narrow working space (compared to the much more sophisticated humanoid robot Armar equipped with much more degrees of freedom) this is also an assumption we can actually realise (in contrast to, e.g., [22]). We also might loosen this assumption when we work with other platforms.

The potential of the tactile sensor the robot is equipped has been investigated in the context of extracting force and surface normal information as well as object properties such as 'fullness' and 'flexibility' (see [15]). The analysis of the potential of the tactile sensor will be further explored in the next year.

4.3 Grasping reflex

The system is equipped with a set of initial behaviors (called 'reflex OACs') that trigger the initial exploration behavior. First, based on co-planar early features (see [A]) a set of grasping hypotheses becomes computed from which some become executed. Note that the aim at this stage is not to get a very high percentage of successful grasps but at least some grasps that are successful. The success itself can be tested haptically (the distance of the grasper after closing is known to the system). Hence multiple attempts can be executed until a successful grasp is being made.

Once a grasp is performed successfully another reflex initiates a set of movements that allow for the segmentation of the object as well as the extraction of a 3D multi-modal representation (see [I] and Section 6).

4.4 High-level Prior Knowledge

The high-level planning system begins with a very simple ontology, consisting of objects, properties, and affordances. In this ontology, objects are the physical entities of the domain (world) that can be manipulated or observed (e.g., a mug, a car, my scarf, etc.). Properties are perceptual characteristics that can be attributed to an object (e.g., red, cylindrical, hard, fluffy, open, location, ...), and can be either relational or functional

in nature. Some domain properties may also be considered "exogenous", meaning that a system external to the high-level planner tracks information about such properties, but makes this information available to the planner on demand (e.g., properties like over, on, upright, etc.). Finally, affordances are references to motor programs that can be executed to manipulate an object.

These high level base concepts are codified in the interaction protocol between the high-level planner and the low-level robot controller. This protocol provides a method for the low level robot controller to pass information about instances of these concepts to the higher level. This protocol (which is more fully specified in [D]) provides a method for the lower level controller to introduce new instances of objects, properties, and affordances as well as update the high-level models current view of domain properties. While the current protocol does not provide a method for introducing new relations we believe this is needed and anticipate introducing this in a later version of the protocol.

As we will see, it is through the interaction of the low level robot controller and the high-level planner that a high-level model of the world is developed. Thus while the system has no prior knowledge of specific objects, properties, relations and actions, it does initially have these basic concepts and can use them to bootstrap its model of the world.

5. World Knowledge Organization and Planning

5.1 LDEC

In order for us to talk about plans and planning in the abstract it is necessary for us to formalize the objects, properties, and affordances in a formal language. This allows us to specify the kinds of information that we expect the high level system to learn based on the robots interaction with the environment. We will briefly summarize our approach that is covered in [C, D].

We will formalize our robot domain using the Linear Dynamic Event Calculus (LDEC) [34, 35], a logical language that combines aspects of the situation calculus with linear and dynamic logics to model dynamically-changing worlds [20, 10, 9].

Our LDEC representation will define the following actions:

Definition 1 *High-Level Domain Actions*

- *grasp*(x)*: move the gripper to pick up object x,*
- *ungrasp*(x)*: release the object x in the gripper,*
- *moveEmptyGripperTo*(ℓ)*: move an empty gripper to the specified location ℓ,*
- *moveFullGripperTo*(ℓ)*: move a full gripper to the specified location ℓ.*

These actions denote higher level counterparts of some of the motor programs available to the robot controller, but already these actions incorporate elements of the state of the world that are not part of robotic control representations of actions. For instance, *ungrasp* models an action that is quite similar to a motor program that performs this operation. Actions like *moveEmptyGripperTo* and *moveFullGripperTo*, on the other hand, are much more abstract and encode information about the state of the world (i.e., the gripper is empty or full). Note that in this case the actions partition the low-level "move gripper" motor-programs into two separate actions that, as we will see, can more readily be learned from the available ISTFs. This representation will also allow us to bypass the learning of the conditional effects [26] of such actions.

Our LDEC representation will also include a number of high-level properties.

LDEC Action Precondition Axioms

$objInGripper = nil \wedge graspable(x) \Rightarrow affords(grasp(x))$
$objInGripper = x \wedge x \neq nil \Rightarrow affords(ungrasp(x))$
$objInGripper = nil \Rightarrow affords(moveEmptyGripperTo(\ell))$
$objInGripper = x \wedge x \neq nil \Rightarrow affords(moveFullGripperTo(\ell))$

LDEC Effect Axioms

$\{affords(grasp(x))\} \multimap [grasp(x)]\, objInGripper = x \wedge gripperLoc = objLoc(x)$
$\{affords(ungrasp(x))\} \multimap [ungrasp(x)]\, objInGripper = nil \wedge objLoc(x) = locOnTable(objLoc(x))$
$\{affords(moveEmptyGripperTo(\ell))\} \multimap [moveEmptyGripperTo(\ell)]\, gripperLoc = \ell$
$\{affords(moveFullGripperTo(\ell))\} \multimap [moveFullGripperTo(\ell)]\, gripperLoc = \ell \wedge objLoc(objInGripper) = \ell$

Table 1: LDEC Axiomatization of High-Level Domain Actions

Definition 2 *High-Level Domain Properties*

- $graspable(x)$*: a predicate that indicates whether an object x is graspable or not,*
- $gripperLoc = \ell$*: a function that indicates the current location of the gripper is* ℓ*,*
- $objInGripper = x$*: a function that indicates the object in the gripper is x; x is* nil *if the gripper is empty,*
- $objLoc(x) = \ell$*: a function that indicates the location of object x is* ℓ*.*

Finally, we also specify a set of "exogenous" domain properties.

Definition 3 *Exogenous Domain Properties*

- $over(x) = \ell$*: a function that returns a location* ℓ *over the object x,*
- $locOnTable(\ell_1) = \ell_2$*: a function that returns a location* ℓ_2 *relative to the table (e.g., on the table or in a box) for another location* ℓ_1 *above the table.*

Like the properties in Definition 2, the exogenous properties model high-level features of the domain. However, unlike domain properties that are directly tracked by the high-level model, exogenous properties are information provided to the high-level system by some external (possibly lower level) source. (We will say more about exogenous properties in Section 5.3.)

Using these actions and properties we can write LDEC axioms that capture the dynamics of the robot scenario described in Table 5.1. Action precondition axioms describe the properties that must hold of the world to apply a given action (i.e., affordances), while the effect axioms characterize what changes as a result of the action. These axioms also encode the STRIPS assumption: fluents that aren't directly affected by an action are assumed to remain unchanged by that action [6].

We refer the interested reader to [C] for more details about the formalization of the robot domain in terms of LDEC rules.

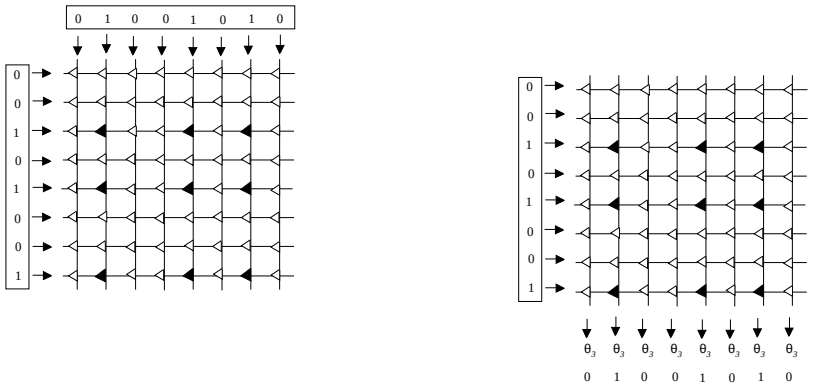


Figure 2: Hetero-associative net: Storage and Retrieval

5.2 Planning

It is easy to see that the LDEC representation we have been sketching supports high-level planning. For instance, with these axioms it is trivial for a planner to construct the following simple plan:

$$[grasp(obj1); moveFullGripperTo(over(box1)); ungrasp(obj1)],$$

to put an object *obj*1 into *box*1, from a state in which the robot's gripper is empty. However, building even this sort of simple plan from first principles is well beyond the capability of the robot controller alone.

Intuitively, the information encoded in a collection of LDEC axioms captures a generalization of the information in a large set of instantiated state transition fragments (ISTFs). The action precondition axioms capture information from the initial state of an ISTF and the action executed, while the effect axioms capture the generalities for the initial state to final state mappings from an ISTFs. We refer the interested reader to [C] for a formalization of the ISTF concept.

5.3 Learning LDEC Representations: Affordances and Consequences of Actions

As we have already said, the LDEC representation of the robot planning problem provides a formal language for specifying the kinds of information that should be learned in order for high level planning to be carried out. We still have yet to propose a mechanism for learning these representations. Our current proposal for learning such action representations involves the use of Willshaw nets or Associative Nets(AN).

ANs were first described in [43, 42] following early work by [37] and [2] extended by [33] and [28]. ANs associate pairs of input and output vectors using a grid of horizontal input lines and vertical output lines with binary switches (triangles) at the intersections (Figure 2). Again we will refer the interested reader to [C] for a much more detailed treatment, however, if we store an input pattern with itself as output (an *auto-associative net*), ANs can be used to complete partial patterns, as needed to recall perceptually non-evident properties of objects, such as the fact that the red cube on the table affords grasping. This is exactly the information that is encoded in action precondition axioms.

Rather than using an auto-associative net we can use a hetero-associative network to learn LDEC style effect axioms. In this case, we again use the initial state, action, and object as the input pattern from each ISTF, however as the output pattern we use the resulting state from the ISTF. This will allow us to learn and retrieve the state-change transitions associated with LDEC operators, with states represented as sparse vectors of relevant facts or propositions.

Thus we envision a scenario wherein as the robot controller explores the world, successful grasps will produce ISTFs. On the basis of multiple reproducible experiences of particular ISTFs we can learn the

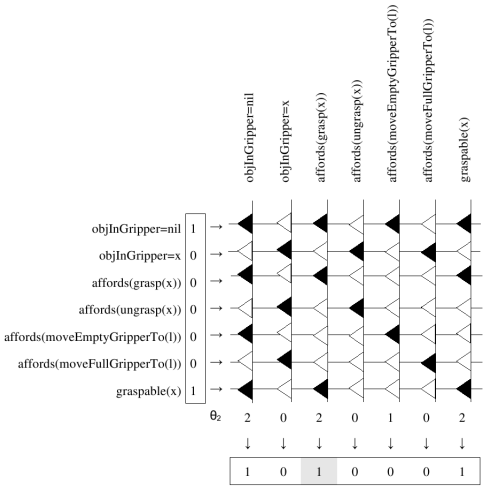


Figure 3: Retrieval of $affords(grasp(x))$ from $objInGripper = nil \wedge graspable(x)$ in the loaded auto-associative net

instantiated versions of the precondition axioms and the effect axioms for the robots actions. The resulting state in each ISTF will vary only in terms of the object-type grasped and the grippers pose. Further, the invariants can be learned as a basis for classifying the world into object classes and action types. As we have discussed, identifiers for actions-types can then be associated with the input conditions for the action via an auto-associative net. Such affordances are added by adding new input and output lines to the net for the new affordance, and using the existing learning algorithm.

This network can be presented with a possibly incomplete set of properties representing the current state of the world, and used to retrieve a complete model of the world state, including non-perceptually available associates including the affordances and object classes (Figure 3).

If the output states and affordances are the same following two different grasp actions for a particular input state, then clearly the effects (as far as the learner and planner are concerned) of the two grasps are the same for that input. If the effects are the same for all inputs then the grasps are equivalent and can be collapsed together.

6. Grounding by Exploration: 'Birth' of objects and first OACs

The high-level LDEC description requires the concept of discrete objects. Hence, our cognitive system needs to be equipped with a mechanism that generates such concepts. A use of prior object model, e.g., in terms of CAD representation — although successfully applied in computer vision research (see e.g., [19]) — is not appropriate in our context since it would represent a too large degree of bias making the system unable to work with unknown objects.

6.1 Procedural Definition of Objects and Grasping Affordances

For our purposes, an object is a distinct, connected, physical entity that can be perceived and acted upon. It is characterized by its perceptual appearance (color, surface hardness, ...), its manipulability (set of *affordances*: can be grasped in a particular way, continues to slide when pushed, ...) and combinations thereof (makes a shattering noise when dropped).

Crucially, defining objects in this way (as opposed to, say, geometric properties) permits (a) the autonomous acquisition of object-related concepts by exploratory learning, and (b) the discovery and goal-directed planning of relations between objects and of sequences of actions.

From this perspective, the fundamental affordance of an object is its graspability, as it permits the construction of an initial object representation in the following terms:

Segmentation: a segmentation of the scene into distinct objects and the object-ness of things as such,

Physical characteristics: a description of the shape of the objects in terms of a multi-modal representation, and

Affordances: an association of successfully performed actions upon the objects.

Such information also provides an essential link between the lower-level robot vision system and the high-level LDEC representation. For instance, objects discovered by segmentation define LDEC level objects; object descriptions are abstracted as sets of high-level relational and functional properties; and affordances suggest the initial object-action relationships that form the basis of LDEC precondition and effect rules.

Segmentation: By grasping an object, the system achieves physical control over it, which can then be visually verified. In the accumulation OAC, entities that move according to the motion of the hand and that do not belong to the gripper increase their associated confidences. At the end, only if the robot has tight physical control over the object, object features become established. By this process, only the visual descriptors belonging to the object are extracted. In case the gripper is empty or the grip is not stable enough such that the object does not follow exactly the motion of the gripper, no visual descriptor becomes established.

Physical characteristics: The accumulation module constructs a 3D visual or multisensory (haptic, ...) representation from the segmented features [I]. These representations are useful for further processes such as object recognition and pose estimation (see Section 6.1).

Affordances: The initial grasp that has been performed has been found to be stable and can now be associated as a *tested* hypothesis to the object as an affordance.

This initial object description in terms of elementary physical characteristics and a primitive grasp affordance mark the 'birth' of an object. A symbolic ID can be created, and it can be made accessible to higher-level planning modules. The object description can then be further refined by probing additional physical characteristics, and by applying exploratory actions and observing their outcomes.

The initial grasps are performed by hard-wired reflexes that can achieve only a certain level of success (cf. Section 4.3 and Aarno et al. [A]). These can be refined and extended into more robust, associative grasp behaviors through further haptic and visual exploration, as outlined in Deliverable D4.1.1.

Note that by the grasping reflex OAC a large set of grasping attempts are produced that are labeled as successful or not successful. This ground truth can be used to further refine the initial grasping reflex and to substitute it by a more elaborated behavior.

In the context of object acquisition, optimal movements for object learning have been investigated in [39] (see also D2.1), in particular addressing the segmentation of the 2D object appearance from the background. Along the lines of this work, we intend to find optimal movements also in the context of 3D object acquisition as done in [I].

In the consortium different object representations are used that cover different aspects of objects. While the visual primitives in [H] cover 2D and 3D edge structures in [40] interest points are used to recognize objects. Furthermore we exploit colour information of objects combined with their segmentation masks to achieve full 6D localization [3]. A major focus with all aspects lies on localisation of objects since the full position and pose is required to execute manipulation tasks with the objects.

ULg is developing probabilistic representations in terms of local appearance and spatial relations [31]. The basic learning procedure is unsupervised; it identifies statistically salient feature co-occurrences and integrates them into higher-level compounds under probabilistic spatial relations. The method has been extended to supervised learning of discriminative features for object recognition [32]. A recent generalization to 6D pose will permit robust, largely view-independent object recognition, pose estimation, and seamless integration of non-visual geometric features. First woprk and experiments are described in [B].

7. Integration of High-level Planning and Robot level Control.

There are a number of system level design issues that must be addressed in order to bring together a high-level planning system and a lower level robotic control system. We have had extensive meetings and discussions about these issues and [D] documents our current thinking on them. However for convenience we summarize a few significant points here.

- **Maintaining a world model for planning** While the robot controller will have direct access to the state of the world, the planning system will not. We assume that the lower level sensing process will push significant changes in the world up to the planning system as they happen. We assume that the planning system will check if its current plan is effected by the changes and replan as necessary.
- **Executing Planned Actions** Once the high-level system has planned actions, it will request execution of the action by the robot controller. All of these requests will be in the form of repeating (with a new object or destination) a previously successful robot action. Effectively the actions the planning system requests will be grounded by the previous experience of the robot controller.
- **Action Monitoring** Since the robot system will not have access to the whole plan as constructed by the planning system, the high-level system will be responsible for continuously monitoring the success of the actions and plan based on the robot controller's reported changes in the state of the world.

8. Links to other Workpackages

Deliverable 8.1.1 is linked to and makes use of work made in a number of workpackages. It is linked to the software and hardware integration issues dealt with in WP1. There are potential links to be exploited in terms of grasp evaluation and optimal actions for object learning in WP2. In Deliverable 8.1.1 a number of sub-modules are used that have been developped in WP4, most notably the grasping reflex and the object learning (see Deliverable D4.1.1) and the integration with the higher level planning system (see Deliverable D4.3.1 and WP5).

9. Publications arising from the Project

The attached publications and reports [A, B, C, D, E, F, G, H, I, J, K] and the publications [29, 3, 40] have been written in the context of the PACO+ project.

Attached Papers

[A] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early Reactive Grasping with Second Order 3D Feature Relations, journal = IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision, year = 2007,.

[B] R. Detry, N. Pugeault, N. Krüger, and J. Piater. Hierarchical integration of local 3D features for probablistic pose estimation. *INTELSIG Technical Report 2007–01–19, Department of Electrical Engineering and Computer Science University of Liege*, 2007.

[C] Ch. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger, and F. Wörgötter. Object action complexes as an interface for planning and robot control. *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, 2006.

[D] Ch. Geib, R. Petrick, K. Mourao, N. Pugeault, M. Steedman, Pacal Haazebroek, N. Krüger, Dirk Kraft, and F. Wörgötter. Paco-plus design documentation for integration of robot control and ai planning. *Technical report, University of Edinburgh*, 2006.

[E] S. Kalkan, N. Pugeault, and N. Krüger. Perceptual operations and relations between 2d or 3d visual entities. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-3, 2007.

[F] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2006.

[G] S. Kalkan, F. Wörgötter, and N. Krüger. Depth prediction at homogeneous image structures. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-2, 2007.

[H] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.

[I] N. Pugeault, Emre Baseski, Dirk Kraft, F. Wörgötter, and N. Krüger. Extraction of multi–modal object representations in a robot vision system. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.

[J] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.

[K] F. Wörgötter. Oac trees for guiding discovery. *PACOPlus Technical Report*, 2007.

References

[1] Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *International journal of computer vision*, 2:333–356, 1987.

[2] John Anderson. A memory storage model utilizing spatial correlation functions. *Kybernetik*, 5:113–119, 1968.

[3] Pedram Azad, Tamim Asfour, and Ruediger Dillmann. Combining Appearance-based and Model-based Methods for Real-time Object Recognition and 6D Localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[4] Michael Brenner and Bernhard Nebel. Continual planning and acting in dynamic multiagent environments. In *Proceedings of the International Symposium on Practical Cognitive Agents and Robots*. University of Western Australia, Perth, 2006.

[5] J. Broekens and P. Haazebroek. Emotion & reinforcement: Affective facial expressions facilitate robot learning. In *Proceedings of the AI for Human Computing workshop at the International Joint Conference on Artificial Intelligence 2007*, 2007.

[6] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[7] P. Fitzpatrick and G. Metta. Grounding Vision Through Experimental Manipulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 361:2165 – 2185, 2003.

[8] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1995.

[9] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[10] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, volume II*, pages 497–604. Reidel, Dordrecht, 1984.

[11] T.E. Hazy, M.J. Frank, and R.C. O'Reilly. Banishing the homunculus: Making working memory work. *Neuroscience*, 139:105–118, 2006.

[12] B. Hommel. On the social psychology of modeling. *Neural Networks*, 19:1455–1475, 2006.

[13] B. Hommel and A. Klippel. Embodying spatial maps. In *Proceedings of AAAI Spring Symposium on Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems*, in press.

[14] P.J. Kellman and M.E. Arterberry, editors. *The Cradle of Knowledge*. MIT-Press, 1998.

[15] M. Kjaergaard, Dirk Kraft, Alex Bierbaum, Tamim Asfour, Rüdiger Dillmann, and Norbert Krüger. Using tactile sensors for multisensorial scene explorations. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-5, 2007.

[16] N. Krüger and F. Wörgötter. Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131:82–147, 2004.

[17] J. P. W. Lacroix, E. O. Postma, B. Hommel, and P. Haazebroek. Nim as a brain for a humanoid robot. In *Proceedings of the Towards Cognitive Humanoid Robots workshop at the IEEE-RAS International Conference on Humanoid Robots 2006*, 2006.

[18] T. Lavender and B. Hommel. Affect and action: Towards an event-coding account. *Emotion and Cognition*, in press.

[19] D.G. Lowe. Three–dimensional object recognition from single two images. *Artificial Intelligence*, 31(3):355–395, 1987.

[20] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[21] A.N. Meltzoff and R.W. Borton. Intermodal matching by human neonates. *Nature*, 292:403–404, 1979.

[22] L. Natale, F. Orabona, G. Metta, and G. Sandini. Exploring the world through grasping: A developmental approach. *CIRA 2005. Proceedings. 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2005*, pages 559– 565, 2005.

[23] L. Natale, S. Rao, and G. Sandini. Learning to act on objects. *Second international workshop, BMCV 2002 in Tübingen*, 2002.

[24] R.C. O'Reilly and M.J. Frank. Making working memory work: A computational model of learning in the frontal cortex and basal ganglia. *Neural Computation*, 18:283–328, 2006.

[25] Hanna M. Pasula, Luke S. Zettlemoyer, and Leslie Pack Kaelbling. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, to appear.

[26] Edwin P. D. Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Reprensentation and Reasoning (KR-89*, pages 324–332, San Mateo, CA, 1989. Morgan Kaufmann Publishers.

[27] Ronald P. A. Petrick and Fahiem Bacchus. A knowledge-based approach to planning with incomplete information and sensing. In Malik Ghallab, Joachim Hertzberg, and Paolo Traverso, editors, *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, pages 212–221, Menlo Park, CA, April 2002. AAAI Press.

[28] Tony Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, San Mateo CA*, pages 30–35, San Francisco, CA, 1991. Morgan Kaufmann.

[29] N. Pugeault, F. Wörgötter, , and N. Krüger. Rigid body motion in an early cognitive vision framework. In *Proceedings of the IEEE Systems, Man and Cybernetics Society Conference on Advances in Cybernetic Systems*, 2006.

[30] R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence Journal*, 78:461–505, 1995.

[31] Fabien Scalzo and Justus H. Piater. Statistical learning of visual feature hierarchies. In *IEEE Workshop on Learning in Computer Vision and Pattern Recognition*, volume 3, pages 44–44, 2005.

[32] Fabien Scalzo and Justus H. Piater. Unsupervised learning of dense hierarchical appearance representations. In *International Conference on Pattern Recognition*, 2006.

[33] Friedrich T. Sommer and Günther Palm. Bidirectional retrieval from associative memory. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[34] Mark Steedman. Temporality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 895–938. North Holland/Elsevier, Amsterdam, 1997.

[35] Mark Steedman. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25:723–753, 2002.

[36] L. Steels. Evolving grounded communication for robots. *Trends in Cognitive Science*, 7(7):308–312, 2003.

[37] K Steinbuch. Die Lernmatrix. *Kybernetik*, 1:36–45, 1961.

[38] E.L. Thorndike. Animal intelligence: An experimental study of the associative processes in animals. *Psychological Review Monograph Supplement*, 2:1–109, 1898.

[39] A. Ude, K. Welke, J. Hale, and G. Cheng. Data acquisition for building object representations: Descerning the manipulated objects from the background. *To be submitted to IEEE Int. Conference on Intelligent robots and system (San Diego)*, 2007.

[40] K. Welke, P. Azad, and R. Dillman. Fast and robust feature-based recognition of multiple objects. *IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, 2006.

[41] T.N. Wiesel and D.H. Hubel. Ordered arrangement of orientation columns in monkeys lacking visual experience. *J. Comp. Neurol.*, 158:307–318, 1974.

[42] David Willshaw. Holography, association and induction. In Geoffrey Hinton and James Anderson, editors, *Parallel Models of Associative Memory*, pages 83–104. Erlbaum, Hillsdale, NJ, 1981.

[43] David Willshaw, Peter Buneman, and Christopher Longuet-Higgins. Non-holographic associative memory. *Nature*, 222:960–962, 1969.

[44] Luke S. Zettlemoyer, Hanna M. Pasula, and Leslie Pack Kaelbling. Learning planning rules in noisy stochastic worlds. In *National Conference on Artificial Intelligence (AAAI),*. AAAI, 2005.

In IEEE International Conference on Robotics and Automation, ICRA'07, *Workshop: "From features to actions - Unifying perspectives in computational and robot vision"*, April, 2007, Rome, Italy (to be presented)

Early Reactive Grasping with Second Order 3D Feature Relations

Daniel Aarno, Johan Sommerfeld, Danica Kragic
Royal Institute of Technology, Sweden
{bishop, johansom, dani}@kth.se

Nicolas Pugeault
University of Edinburgh, UK
npugeaul@inf.ed.ac.uk

Sinan Kalkan, Florentin Wörgötter
University of Göttingen, Germany
{sinan, worgott}@bccn-goettingen.de

Dirk Kraft, Norbert Krüger
Sydansk University and Aalborg University, Denmark
{norbert, kraft}@mip.sdu.dk

***Abstract*— One of the main challenges in the field of robotics is to make robots ubiquitous. To intelligently interact with the world, such robots need to understand the environment and situations around them and react appropriately, they need context-awareness. But how to equip robots with capabilities of gathering and interpreting the necessary information for novel tasks through interaction with the environment and by providing some minimal knowledge in advance? This has been a longterm question and one of the main drives in the field of cognitive system development.**

The main idea behind the work presented in this paper is that the robot should, like a human infant, learn about objects by interacting with them, forming representations of the objects and their categories that are grounded in its embodiment. For this purpose, we study an early learning of object grasping process where the agent, based on a set of innate reflexes and knowledge about its embodiment. We stress out that this is not the work on grasping, it is a system that interacts with the environment based on relations of 3D visual features generated trough a stereo vision system. We show how geometry, appearance and spatial relations between the features can guide early reactive grasping which can later on be used in a more purposive manner when interacting with the environment.

I. Introduction

For a robot that has to perform tasks in a human environment, it is necessary to be able to learn about objects and object categories. It has been recognized recently that grounding in the embodiment of a robot, as-well as continuous learning is required to facilitate learning of objects and object categories [1], [2]. The idea is that robots will not be able to form useful categories or object representations by only being a passive observer of its environment. Rather a robot should, like a human infant, learn about objects by interacting with them, forming representations of the objects and their categories that are grounded in its embodiment.

Central to the approach are three almost axiomatic assumptions, which are strongly correlated. These also represent the building blocks of our approach toward creating a cognitive artificial agent:

- Objects and Actions are inseparably intertwined; Entities ("things") in the world of a robot (or human) will only become semantically useful "objects" through the action that the agent can/will perform on them. This forms so-called Object-Action Complexes (named OACs) which are the building blocks of cognition.
- Cognition is based on recurrent processes involving nested feedback loops operating on, contextualizing and reinterpreting object-action complexes. This is done through actively closing the perception-action cycle.
- A unified measure of success and progress can be obtained through minimization of contingencies which an artificial cognitive system experiences while interacting with the environment or other agents, given the drives of the system.

To demonstrate the feasibility of our approach, we aim at building a robot system that step by step develop increasingly advanced cognitive capabilities. In this paper, we demonstrate our initial efforts towards this goal by designing a scenario for manipulation and grasping of objects.

One of the most basic interactions that can occur between a robot and an object is for the robot to push the object, i.e. to simply make a physical contact. Already at this stage, the robot should be able to form two categories: physical and non-physical objects, where a physical object is categorized by the fact that interaction forces occur. A higher level interaction between the robot and an object would exist if the robot was able to *grasp* the object. In this case, the robot would gain actual physical control over the object and having the possibility to perform controlled actions on it, such as examining it from other angles, weighing it, placing it etc. Information obtained during this interaction can then be used to update the robots representations about objects and the world. Furthermore, the successfully performed grasps can be used as ground truth for future grasp refinement, [2].

In this paper, we are interested in investigating an initial "reflex-like" grasping strategy that will form a basis for

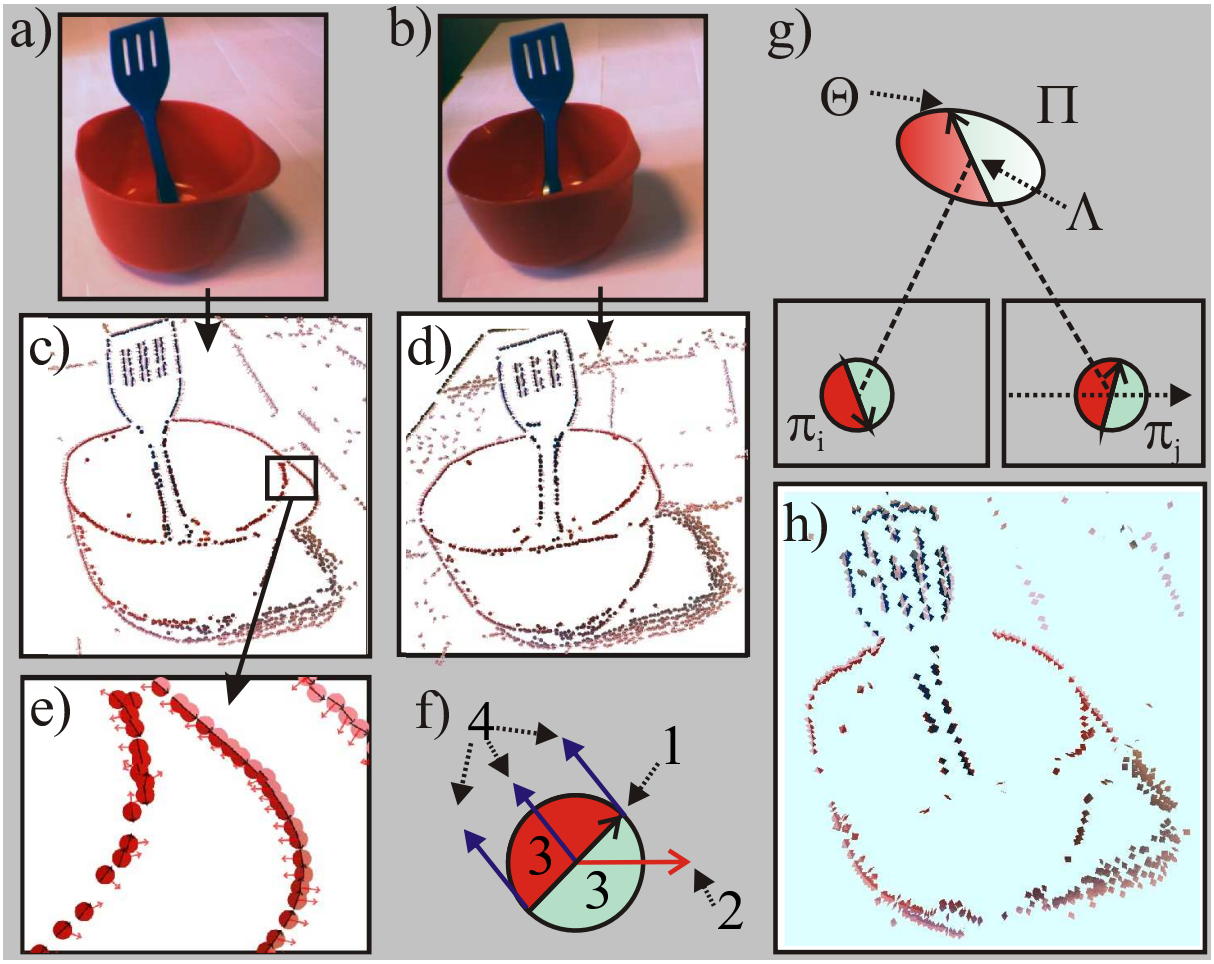


Fig. 1. Illustration of the vision module. a) and b) shows the images captured by the left and right cameras (respectively); c) and d) show the primitives extracted from these two images; in e) a detail of the primitive extraction is shown; f) illustrates the schematic representation of a primitive, where 1. represents the orientation, 2. the phase, 3. the color and 4. the optical flow. g) from a stereo–pair of primitives (π_i, π_j) we reconstruct a 3D primitive Π, with a position in space Λ and an orientation Θ; h) shows the resulting 3D primitives reconstructed for this scenario.

a cognitive robot system that, at the first stage, acquires knowledge of objects and object categories and is able to further refine its grasping behavior by incorporating the gained object knowledge, [3]. The grasping strategy does not require *a-priori* object knowledge, and it can be adopted for a large class of objects. The proposed reflex-like grasping strategy is based on second order relations of multi-modal visual features descriptors, called *spatial primitives*, that represent object's geometric information, e.g. 3D pose (position and orientation) as well as its appearance information, e.g. color and contrast transition etc. [4], see Fig. 1. Co–planar tuples of the spatial primitives allow for the definition of a plane that can be associated to a grasp hypothesis. In addition, these local descriptors are part of semi-global collinear groups [5]. Furthermore, the color information (by defining co–colority in addition to co–planarity of primitive pairs) can be used to further improve the definition of grasp hypotheses. In this paper, we employ the structural richness of the descriptors in terms of their geometry and appearance as well as the structural relations co–linearity, co–planarity and co–colority to derive a set of grasping options from a stereo image.

We note that the purpose of this work is not to develop yet another grasping strategy for a specific setting, but rather to provide low-level grasping reflexes that can be used to generate successful grasps on arbitrary objects. These grasping reflexes are part of a larger framework on cognitive robotics where a robot is equipped only with a set of innate grasps which are used to develop more complex object manipulation abilities through interaction and reinforcement so that 1) more complex feature relations become associated to more precise and successful grasps, and 2) object knowledge becomes acquired and used to further refine the grasping process. We also have to stress out that no scene segmentation is performed, since the system does not even have a concept of an object to start with. In short, the contributions of our work are the generation of a set of grasp suggestions on unknown objects based on visual feedback, grouping of visual primitives for decreasing the size of the grasps and evaluation of grasps using the GraspIt! environment, [6].

In this work, "kitchen-type" objects such as cups, glasses, bowls and various kitchen utensils are considered. However,

our algorithm is not designed for specific object classes but can be applied for any rigid object that can be described by edge–like structures.

This paper is organized as follows. In Section II, we shortly review the related work and in Section III give a general overview of the system. Details about extraction of spatial primitives are presented in Section IV and elementary grasping actions defined in Section V. Results of the experimental evaluation are summarized in Section VI and plans for future research outlined in Section VII.

II. Related Work

The idea to learn or refine grasping strategies is not new. Kamon *et al.* combined heuristic methods with learning algorithms to learn how to select good grasps [7]. Rössler *et al.* used two levels of learners to learn local and global grasp criteria [8], where the local learner learns about the local structure of an object and the global learner learns which of the possible local grasps are best given the object.

There has been a large amount of work presented in the area of robotic grasping during the last two decades [9]. However, much of this work has been dealing with analytical methods where the shape of the objects being grasped is known *a-priori*. This work, referred to as *analytical methods*, has focused primarily on computing grasp stability based on force and form-closure properties or contact-level grasps synthesis based on finding a fixed number of contact locations with no regard to hand geometry, [9],[10]. This problem is important and difficult mainly because of the high number of DOFs involved in grasping arbitrary objects with complex hands. Another important research area is grasp planning without detailed object models where sensor information such as computational vision is used to extract relevant features in order to compute suitable grasps, [11], [12], [13]. In this paper, we denote this approach as *sensor-driven*.

Related to our work, we have to mention systems that deal with automatic grasp synthesis and planning, [14],[15],[16],[17]. This work concentrates on automatic generation of stable grasps given assumptions about the shape of the object and robot hand kinematics. Example of assumptions may be that the full and exact pose of the object is known in combination with its (approximate) shape, [14]. Another common assumption is that the outer contour of the object can be extracted and a planar grasp applied, [16]. Taking into account both the hand kinematics as well as some *a-priori* knowledge about the feasible grasps has been acknowledged as a more flexible and natural approach towards automatic grasp planning [18],[14]. [18] studies methods for adapting a given prototype grasp of one object to another object. The method proposed in [14] presents a system for automatic grasp planning for a Barrett hand [19] by modeling an object as a set of shape primitives, such as spheres, cylinders, cones and boxes in a combination with a set of rules to generate a set of grasp starting positions and pregrasp shapes.

One difference between the analytical and sensor-driven approaches is that the former tend to use complex hands with many DOFs, while the latter use simple ones such as parallel yaw-grippers. One reason for this is that if the reconstruction of the object's shape is not very accurate, using a complex gripping device does not necessarily facilitate grasping performance. For sensor-driven approaches it is also very common to perform only planar grasps where all the contacts between the fingers and the object are confined to a plane. As an example, objects are placed on a table and grasped from above. This simplifies both the vision problem, since only the outer boundary of the object in the image plane has to be estimated, as well as the grasp planning by constraining the search space.

The main differences of our work compared to the above-mentioned work are the following:

- We rely on 3D information based on three dimensional primitives extracted online. This allows us to compute arbitrary grasping directions compared to only planar grasps considered in, e.g. [16].
- The structural richness of the primitives (geometric and appearance based information, collinear grouping) allows for an efficient reduction of grasping hypotheses while keeping relevant ones.
- Our system focuses on generating a certain percentage of successful grasps on arbitrary objects rather than high quality grasps on a constrained set of objects. We will show that with our representations we are able to extract a sufficient number of successful grasping options to be used as initiator of learning schemes aiming at more sophisticated grasping strategies.

III. System Overview

The work presented in this paper serves as a building block for the development of a cognitive robot system. The robot platform considered is comprised of a set of sensors and actuators. The minimum requirements necessary to realize the work presented in this paper is that the sensors are able to deliver a set of visual primitives (section IV) and the configuration of the actuators. The required actuator is a manipulator, comprised of a robotic arm and a gripper device. In this context the term sensor is not necessarily related to a real physical sensing device, but rather an abstract measurement delivered to the system, possibly after performing computations on data sampled from a physical sensor.

The complete system is outlined in Fig. 2. In this paper we are interested in developing grasping reflexes. A grasping reflex is triggered by the vision system. The vision system continuously computes the spatial primitives described in section IV which are feed as sensor input to the set of reflexes and to the cognitives system. If the grasping reflex has not been inhibited by the cognitive system and the sensor stimuli is strong enough, i.e. there are sufficiently many spatial primitives visible, the grasping reflex is performed. This reflex behavior computes a set of possible grasps and tries to perform them. Each grasp evaluated results in a reinforcement signal which can be used by the cognitive system to update its representation of the world. The following

two sections describe the spatial primitives and the rules for generating the grasping actions.

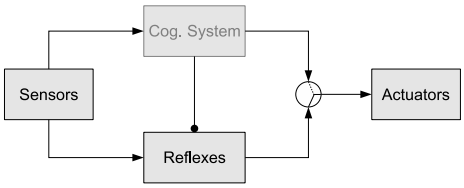


Fig. 2. System overview

IV. Spatial Primitives

The image processing used in this paper is based on multi-modal visual primitives [20], [4], [5]. First, 2D primitives are extracted sparsely at points of interest in the image (in this case contours) and encode the value of different visual operators (hereby referred to as *visual modalities*) such as local orientation, phase, color (on each side of the contour) and optical flow (see Fig. 1.d, 1.e and 1.f). In a second step, the 2D primitives become extended to the spatial primitives used in this work. After finding correspondences between primitives in the left and right image, we reconstruct a spatial primitive, (see Fig. 1.g) that has the following components, (for details see [21], [5]):

$$\Pi = \{\Lambda, \Theta, \Omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)\},$$

where Λ is the 3D position; Θ is the 3D orientation; Ω is the phase (i.e., contrast transition); and, $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color of the spatial primitive, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$).

The sparseness of the primitives allows to formulate three *relations* between primitives that are crucial in our context:

- *Co–planarity:*
  Two spatial primitives Π_i and Π_j are co–planar iff their orientation vectors lie on the same plane, i.e.:

  $$cop(\Pi_i, \Pi_j) = 1 - |\mathbf{proj}_{\Theta_j \times \mathbf{v}_{ij}}(\Theta_i \times \mathbf{v}_{ij})|,$$

  where $\mathbf{v}_{ij}$ is defined as the vector $(\Lambda_i - \Lambda_j)$, and $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$ is defined as:

  $$\mathbf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2}\mathbf{u}. \quad (1)$$

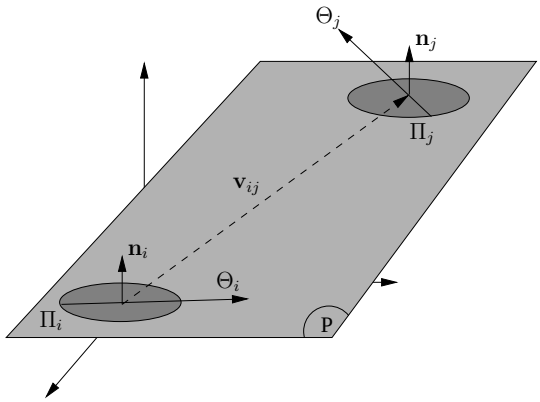
  The co–planarity relation is illustrated in Fig. 3(a).
- *Collinear grouping (i.e., collinearity):*
  Two spatial primitives Π_i and Π_j are collinear (i.e., part of the same group) iff they are part of the same contour. Due to uncertainty in 3D reconstruction process, in this work, the collinearity of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. We define the collinearity of two 2D primitives π_i and π_j as:

  $$col(\pi_i, \pi_j) = 1 - \left| sin\left(\frac{|\alpha_i| + |\alpha_j|}{2}\right)\right|,$$

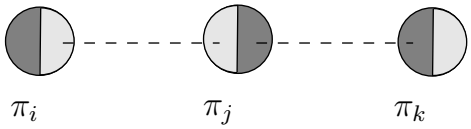
  where α_i and α_j are as shown in Fig. 3(c), see [5] for more details on collinearity.
- *Co–colority:* Two spatial primitives Π_i and Π_j are co–color iff their parts that face each other have the same color. In the same way as collinearity, co–colority of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. We define the co–colority of two 2D primitives π_i and π_j as:

  $$coc(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j),$$

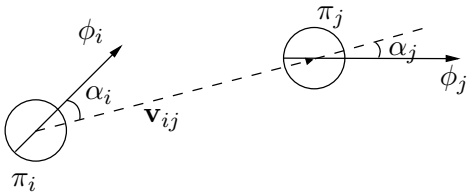
  where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the RGB representation of the colors of the parts of the primitives π_i and π_j that face each other; and, $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is Euclidean distance between RGB values of the colors $\mathbf{c}_i$ and $\mathbf{c}_j$. In Fig. 3(b), a pair of co–color and not co–color primitives are shown.



(a) Co–planarity of two 3D primitives Π_i and Π_j.



(b) Co–colority of three 2D primitives π_i, π_j and π_k. In this case, π_i and π_j are cocolor, so are π_i and π_k; however, π_j and π_k are not cocolor.



(c) Collinearity of two 2D primitives π_i and π_j.

Fig. 3. Illustration of the relations between a pair of primitives.

Co–planarity in combination with the 3D position allows for the definition of a grasping pose; Collinearity and co–colority allows for the reduction of grasping hypotheses. The use of the relations in the grasping context is shown in Fig. 4.

V. Elementary Grasping Actions

Coplanar relationships between visual primitives suggests different graspable planes. Fig. 4 shows a set of spatial primitives on two different contours l_i and l_j with co–planarity, co–colority and collinearity relations.

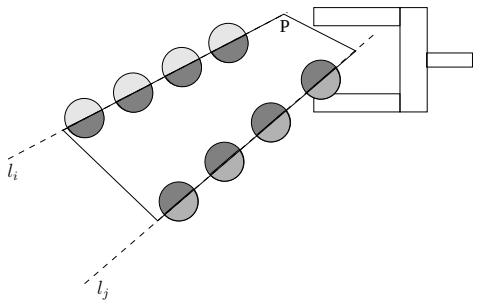


Fig. 4. A set of spatial primitives on two different contours l_i and l_j that have co–planarity, co–colority and collinearity relations; a plane P defined by the co–planarity of the spatial primitives and and an example grasp suggested by the plane.

Five elementary grasping actions (EGA) will be considered as shown in Fig. 5. EGA1 is a "pinch" grasp on a thin edge like structure with approach direction along the surface normal of the plane spanned by the primitives. EGA2 is an "inverted" grasp using the inside of two edges with approach along the surface normal. EGA3 is a "pinch" grasp on a single edge with approach direction perpendicular to the surface normal. EGA4 is similar to EGA2 but its approach direction is perpendicular to the surface normal. Also it tries to go in "below" one of the primitives. EGA5 is wide grasp making contact on two separate edges with approach direction along the surface normal.

The EGAs will be parameterized by their final pose (position and orientation) and the initial gripper configuration. For the simple parallel jaw gripper, an EGA will thus be defined by seven parameters: EGA$(x, y, z, \gamma, \beta, \alpha, \delta)$ where $\mathbf{p} = [x, y, z]$ is the position of the gripper "center" according to Fig. 6; $\gamma,\ \beta,\ \alpha$ are the roll, pitch and yaw angles of the vector $\mathbf{n}$; and δ is the gripper configuration, see Fig. 6. Note that the gripper "center" is placed in the "middle" of the gripper.

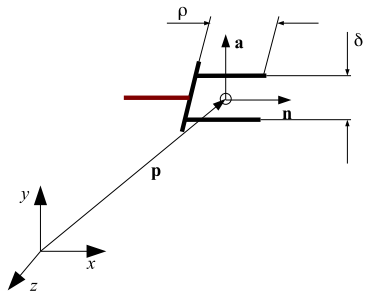


Fig. 6. Parameterization of EGAs.

The main motivation for choosing these grasps is that they represent the simplest possible two fingered grasps humans commonly use. The result of applying the EGAs can be evaluated to provide a reinforcement signal to the system. The number of possible outcomes of each of the EGAs are different and will be explained below.

For all of the EGAs the possibility of an *early failure* exists. That is, the EGA fails before reaching the target configuration. This will result in a reinforcement R_{fe}. Furthermore, it is possible for all EGAs to fail a grasping procedure.

For EGA1, EGA3 and EGA5, a failed grasp can be detected by the fact that the gripper is completely closed. This situation will result in a reinforcement R_{fl}.

For EGA1 and EGA3, the expected grasp is a pinch type grasp, i.e. narrow. Therefore, they can also "fail" if the gripper comes to a halt too early, that is $\delta > \Delta_{min}$. This will result in a reinforcement R_{ft}.

EGA2 fails if the gripper is fully opened, meaning that no contact was made with the object. This gives a reinforcement R_{fh}.

To detect failure of EGA4, a tactile sensor is required on the side of the "fingers". If, after positioning and opening the gripper, there is no contact between the object and the tactile sensor, the EGA has failed. This results in a reinforcement R_{fc}.

If none of the above situations is encountered, a positive reinforcement R_g is given, and the EGA is considered successful.

A. Computing Action Parameters

Let $\Gamma = \{\Pi_1, \Pi_2\}$ be a primitive pair, $\Lambda(\Pi)$ be the position of Π and $\Theta(\Pi)$ be the orientation of Π, also let Γ_i be the i:th pair. From that we can calculate

$$\begin{aligned}
\mathbf{d} &= \Lambda(\Pi_2) - \Lambda(\Pi_1) \\
\mathbf{n}_1 &= \Theta(\Pi_1) \times \mathbf{d} \\
\mathbf{n}_2 &= \Theta(\Pi_2) \times \mathbf{d} \\
sw &= \begin{cases} -1 & \text{if} \quad \mathbf{n}_1 \cdot \mathbf{n}_2 < 0 \\ 1 & \text{else} \end{cases}
\end{aligned}$$

and with those we calculate the plane $\mathbf{p}$

$$\begin{aligned}
\mathbf{P_p} &= \Lambda(\Pi_1) + frac\mathbf{d}2 \\
\mathbf{n_p} &= \frac{\mathbf{n}_1 + sw\mathbf{n}_2}{\parallel \mathbf{n}_1 + sw\mathbf{n}_2 \parallel}
\end{aligned}$$

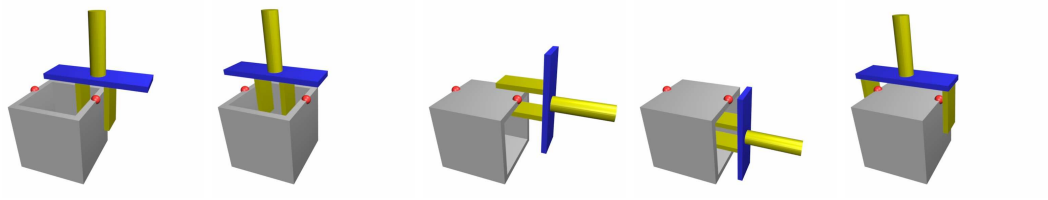
which is used when calculating actions parameters

The parameterization of the EGAs is given with the gripper normal $\mathbf{n}$ and the normal of the surface between the two fingers $\mathbf{a}$ as illustrated in Fig. 6. From this, the yaw, pitch and roll angles can be easily computed.

For EGA1, there will be two possible parameter sets given the primitive pair $\Gamma = \{\Pi_1, \Pi_2\}$. The parameterization is as follows:

$$\begin{aligned}
&\mathbf{p}_{\text{gripper}} = \Lambda(\Pi_i) \\
&\mathbf{n} = \nabla(\mathbf{p}) \\
&\mathbf{a} = \mathbf{perp_n}(\Theta(\Pi_i))/ \parallel \mathbf{perp_n}(\Theta(\Pi_i)) \parallel \quad \text{for } i = 1, 2
\end{aligned}$$

where $\nabla(\mathbf{p})$ is the normal of the plane $\mathbf{p}$ and $\mathbf{perp_u}(\mathbf{a})$ is the projection of $\mathbf{a}$ perpendicular to $\mathbf{u}$. That is $\mathbf{perp_u}(\mathbf{a}) = \mathbf{a} - \mathbf{proj_u}(\mathbf{a})$, where $\mathbf{proj_u}(\mathbf{a})$ is defined according to (1).



(a) EGA1 (b) EGA2 (c) EGA3 (d) EGA4 (e) EGA5

Fig. 5. Elementary grasping actions, EGAs.

For EGA2, there is only one parameter set.

$$\begin{aligned}\mathbf{d} &= \Lambda(\Pi_2) - \Lambda(\Pi_1)\\ \mathbf{p}_{\text{gripper}} &= \Lambda(\Pi_1) + \mathbf{d}/2\\ \mathbf{n} &= \nabla(\mathbf{p})\\ \mathbf{a} &= \mathbf{n} \times \mathbf{d}/ \parallel \mathbf{n} \times \mathbf{d} \parallel\end{aligned}$$

For EGA3, there will be two possible parameter sets for $i = 1, j = 2$ and $i = 2, j = 1$.

$$\begin{aligned}\mathbf{d} &= \Lambda(\Pi_j) - \Lambda(\Pi_i)\\ \mathbf{n} &= \mathbf{d}/ \parallel \mathbf{d} \parallel\\ \mathbf{p}_{\text{gripper}} &= \Lambda(\Pi_i)\\ \mathbf{a} &= \mathbf{n} \times \nabla(\mathbf{p})\end{aligned}$$

For EGA4, there will be two possible parameter sets for $i = 1, j = 2$ and $i = 2, j = 1$. Where ϵ is a step size parameter that will depend on the gripper used.

$$\begin{aligned}\mathbf{d} &= \Lambda(\Pi_j) - \Lambda(\Pi_i)\\ \mathbf{n} &= \mathbf{d}/ \parallel \mathbf{d} \parallel\\ \mathbf{p}_{\text{gripper}} &= \Lambda(\Pi_i) - \nabla(\mathbf{p}) \cdot \epsilon\\ \mathbf{a} &= \mathbf{n} \times \nabla(\mathbf{p})\end{aligned}$$

EGA5 will have the same parameters as EGA2 except that the gripper will be fully opened.

B. Limiting the Number of Actions

For a typical scene, the number of coplanar pairs of primitives is in the order of $10^3 - 10^4$. Given that each coplanar relationship gives rise to 8 different grasps from the five different categories, it is obvious that the number of suggested actions must be further constrained. Another problem is that coplanar structures occur frequently in natural scenes and only a small set of them suggest feasible actions, e.g. objects placed on a table create a lot of 3D line structures coplanar to the table but can not grasped directly by a grasping direction normal to the table. In addition, there exist many coplanar pairs of primitives affording similar grasps.

To overcome some of the above problems, we make use of the structural richness of the primitives. First, their embedding into collinear groups naturally clusters the grasping hypotheses into sets of redundant grasps from which only one needs to be tested. Furthermore, co–colority, gives an additional hypothesis for a potential grasp.

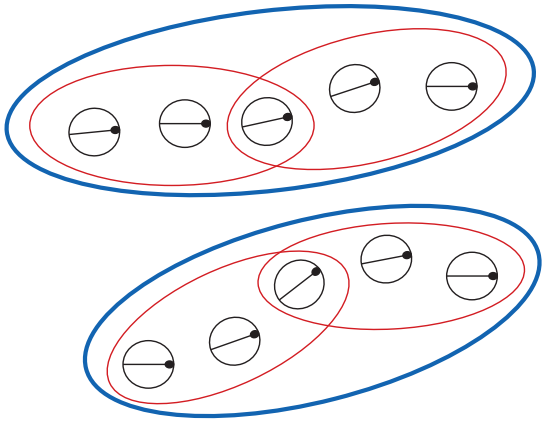


Fig. 7. Small overlapping groups form large groups

1) Using Grouping Information: From the 2D primitives (before stereo reconstruction) collinear neighbors can be found. The collinear neighbors can be mapped to corresponding 3D primitives. These small neighborhoods form the set of *small groups*, $\{g_1, g_2, ..., g_N\}$. The *large groups*, $\{G_1, G_2, ..., G_M\}$, are formed by the grouping of the small groups overlapping each other, Fig. 7 such that if Π_i and Π_j are part of group g_x and Π_j and Π_k is part of group g_y then g_y and g_x is part of the same large group G_z. The result is that the large groups are separated meaning that a primitive that exist in group G_X can not exist in any other group G_Y. Using this grouping information it is possible to add additional constraints on the generation of EGA s.

First, all primitives that are not part of a sufficiently large group G_i are discarded. Secondly, the relations co–planarity and co–colority between small groups of primitives are computed such that primitive $\Pi_i \in g_x$ and $\Pi_j \in g_y$ are only considered to have a co–planarity or co–colority relation if all primitives in g_x are coplanar or cocolor w.r.t all primitives in g_y. Finally, it is possible to constrain the generation of EGAs to only one EGA of each type for each large group.

VI. Experimental Evaluation

Fig. 9, Fig. 10 and Fig. 11 show some of the grasps generated for the scenes evaluated here. Fig. 8 shows visual features generated by the stereo system and a selection of generated actions. Fig. 9 shows a simple plate structure for which the outer contour is generated since the object is

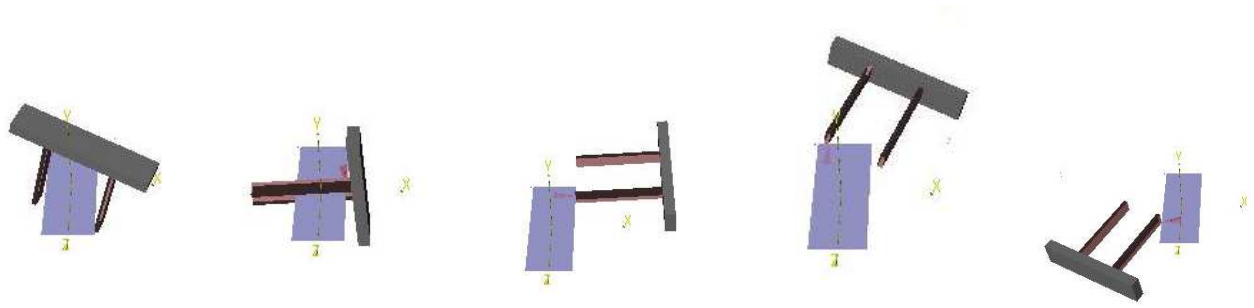


Fig. 9. Examples of tested grasps on a plate (from left): successful grasp using EGA5, and a few early failures using EGA1, EGA3 and EGA5, res5respectively.

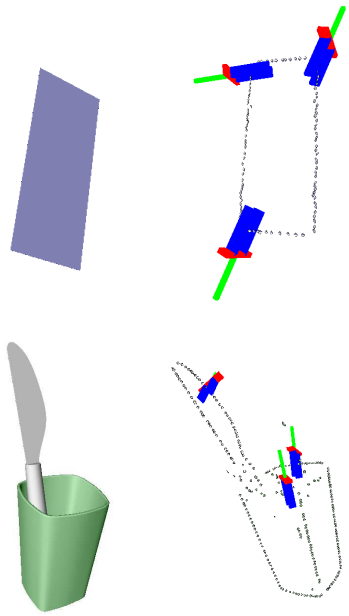


Fig. 8. Two example scenes designed for testing and a selection of the generated actions.

Scene	gr	pl+gr	col+gr	gr+pl+col
Plane	70% (7/10)	83% (5/6)	57% (4/7)	100% (5/5)
Cup	26% (17/66)	38% (14/37)	27% (13/49)	33% (8/24)
Cup/Kn	31% (14/45)	28% (9/32)	31% (11/35)	25% (5/20)
3 objects	8% (33/434)	9% (9/98)	13% (18/139)	15% (8/53)

TABLE I

EXPERIMENTAL EVALUATION OF THE GRASP SUCCESS RATE WHERE THE FOLLOWING NOTATION IS USED: PL (CO-PLANARITY), GR (GROUPING), CL (CO-COLORITY) AND (SUCCESFULL/TESTED) GRASPS.

homogeneous in texture. Fig. 10 shows a scene with a single, but a more complex object than the previous one. Fig. 11 shows two scenes with two (cup and knife) and three objects (box, cup and bottle9.

On each of the scene, after the spatial primitives have been extracted, elementary actions shown in Fig. 5 are tested. There are few reasons for which a certain grasp may fail:

- The system does not have the knowledge of whether the object is hollow or not, so testing EGA2 will results with a collision and thus failure.
- Since no surface is reconstructed, EGA1 will fail for hollow objects which are grasped from “below”.
- If the hand, during the approach, detects a collision on one of the fingers, the grasping process is stopped. In reality, this grasp may happen to be successful anyway if the object is moved so that it is centered between the fingers.

Table I summarizes the results for the generated success rate regarding a number of successful grasps given no knowledge of the object shape. We note that the results are a summary of an extensive experimental evaluation since, given different types and combinations of spatial primitives all generated actions had to be evaluated. It can be seen that for a scene of low complexity (plate) the average number of successful grasps is close to 80%. For more complex scenes this number is dependant on the number and type of objects. It is also important to note not only the percentage but the number of evaluated grasps. Although, in some cases, the success rate is lower when primitives are integrated, there are much fewer hypotheses tested. These results should also be considered together with the results presented in Table II where we show how the integration of grouping, co-colority and co-planarity affects the number of generated hypotheses (affordances). Another thing to point out related to Table I is that most of the unsuccessful grasps happened due to an “early failure” such as that a contact was detected before the grasp was executed. Again, this failure may in some cases result with a successful grasp anyway. Another big source of failure was that there was nothing to lift, i.e. EGA3 could not have been applied.

VII. CONCLUSIONS

Robots should be able to extract more knowledge through their interaction with the environment. The basis for this interaction should not be a detailed model of the environment and lots of *a-priori* knowledge but the robot should be engaged in an exploration process through which it can generate more knowledge and more complex representations. In this paper, we have presented one of the building blocks necessary in such a system.

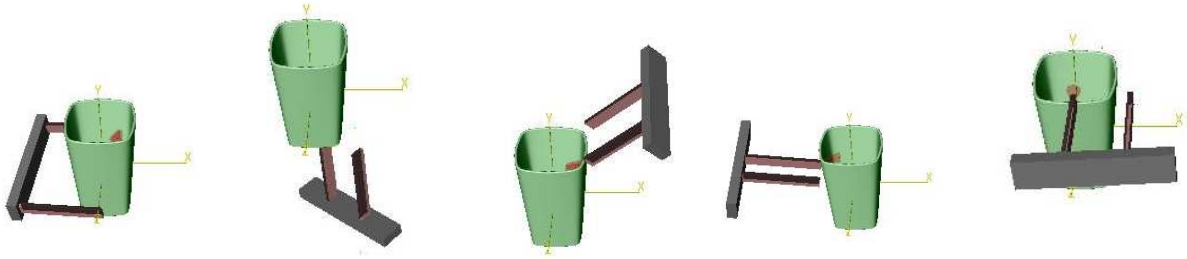


Fig. 10. Examples of tested grasps on a cup (from left): a successful grasp using EGA1, and a few early failures using EGA1, EGA1, EGA2 and EGA3, respectively.

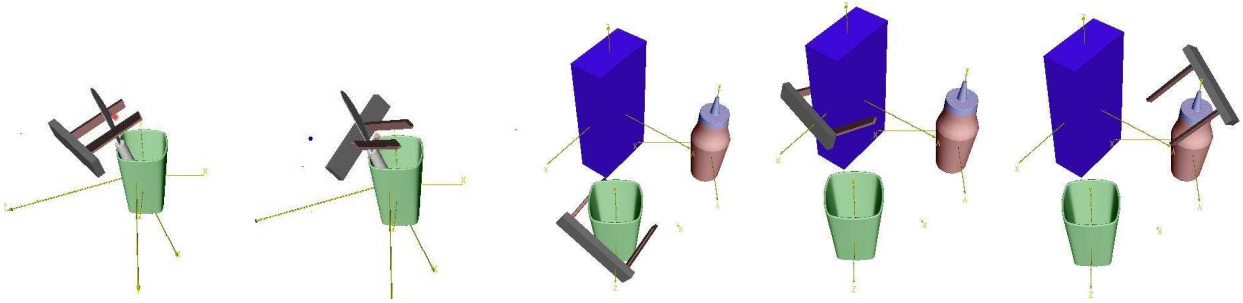


Fig. 11. Examples of successful grasps with two and three objects.

Scene	(no gr)	(no gr)+pl	(no gr)+col	(no gr)+pl+col
Plane	46 224	35 608	38 512	30 224
Cup	172 224	96 112	89 392	56 120
Cup/knife	269 360	140 920	139 136	79 104
3 objects	927 368	303 960	315 336	166 008

Scene	gr	gr+pl	gr+col	gr+pl+col
Plane	80	48	56	40
Cup	528	296	392	192
Cup/knife	360	256	280	160
3 objects	3472	784	1112	424

TABLE II

THE NUMBER OF GENERATED ACTION HYPOTHESES WHERE THE FOLLOWING NOTATION IS USED: NO GR (NO GROUPING), PL (CO-PLANARITY), GR (GROUPING), CL (CO-COLORITY).

In particular, we have designed an early grasping system, based on a set of innate reflexes and knowledge about its embodiment. We relied on 3D information based on primitives extracted online and showed how the structural richness of primitives can be used for an efficient reduction of grasping hypotheses while keeping relevant ones. Rather than dealing with high quality grasps on a constrained set of known objects, we have demonstrated that the system is able of generating a certain percentage of successful grasps on arbitrary objects. This is important for our future research that will develop complex learning schemes aiming at more sophisticated grasping strategies and knowledge representation.

ACKNOWLEDGMENT

This work has been supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657.

REFERENCES

[1] A. Stoytchev, "Behavior-Grounded Representation of Tool Affordances," in *IEEE International Conference on Robotics and Automation*, pp. 3060–3065, 2005.

[2] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning About Objects Through Action - Initial Steps Towards Artificial Cognition," in *IEEE International Conference on Robotics and Automation*, pp. 3140–3145, 2003.

[3] P. Azad, T. Asfour, and R. Dillmann, "Combining appearance-based and model-based methods for real-time object recognition and 6d localization," in *IEEE International Conference on Intelligent Robots and Systems*, 2006.

[4] N. Krüger, M. Lappe, and F. Wörgötter, "Biologically motivated multi-modal processing of visual primitives," *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, vol. 1, no. 5, pp. 417–428, 2004.

[5] N. Pugeault, F. Wörgötter, and N. Krüger, "Multi-modal scene reconstruction using perceptual grouping constraints," in *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision, (in conjunction with IEEE CVPR 2006)*, 2006.

[6] A. T. Miller and P. Allen, "Graspit!: A versatile simulator for grasping analysis," in *ASME International Mechanical Engineering Congress and Exposition*, 2000.

[7] I. Kamon, T. Flash, and S. Edelman, "Learning Visually Guided Grasping: A Test Case in Sensorimotor Learning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 266–276, 1998.

[8] B. Rössler, J. Zhang, and A. Knoll, "Visual Guided Grasping of Aggregates using Self-Valuing Learning," in *IEEE International Conference on Robotics and Automation*, pp. 3912– 3917, 2002.

[9] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *IEEE International Conference on Robotics and Automation*, pp. 348–353, 2000.

[10] D. Ding, Y.-H. Liu, and S. Wang, "Computing 3-d optimal formclosure grasps," in *IEEE International Conference on Robotics and Automation*, pp. 3573 – 3578, 2000.

[11] A. Hauck, J. Rüttinger, M. Sorg, and G. Färber, "Visual Determination of 3D Grasping Points on Unknown Objects with a Binocular Camera System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 272–278, 1999.

[12] M. Rutishauser and M. Stricker, "Searching for Grasping Opportunities on Unmodeled 3D Objects," in *British Machine Vision Conference*, pp. 277 – 286, 1995.

[13] A. Morales, G. Recatalá, P. J. Sanz, and Á. P. del Pobil, "Heuristic Vision-Based Computation of Planar Antipodal Grasps on Unknown Objects," in *IEEE International Conference on Robotics and Automation*, pp. 583– 588, 2001.

[14] A. T. Miller, S. Knoop, and H. I. C. P.K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation*, pp. 1824–1829, 2003.

[15] N. S. Pollard, "Closure and quality equivalence for efficient synthesis of grasps from examples," *International Journal of Robotic Research*, vol. 23, no. 6, pp. 595–613, 2004.

[16] A. Morales, E. Chinellato, A. H. Fagg, and A. del Pobil, "Using experience for assessing grasp reliability," *International Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 671–691, 2004.

[17] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Extending fingertip grasping to whole body grasping," in *International Conference on Robotics and Automation*, pp. 2677 – 2682, 2003.

[18] N. S. Pollard, "Parallel methods for synthesizing whole-hand grasps from generalized prototypes," *PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology*, 1994.

[19] http://www.barrett.com/robot/products/hand/handfram.htm.

[20] N. Krüger and F. Wörgötter, "Multi-modal primitives as functional models of hyper-columns and their use for contextual integration," *International Symposium on Brain, Vision and Artificial Intelligence, Lecture Notes in Computer Science, Springer, LNCS 3704*, pp. 157–166, 2005.

[21] N. Krüger and M. Felsberg, "An explicit and compact coding of geometric and structural information applied to stereo matching," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 849–863, 2004.

Hierarchical Integration of Local 3D Features for Probabilistic Pose Recovery

Renaud Detry
renaud.detry@ulg.ac.be
Montefiore Institute
University of Liège
Belgium

Nicolas Pugeault
npugeaul@inf.ed.ac.uk
School of Informatics
Univ. of Edinburgh
United Kingdom

Norbert Krüger
norbert@mip.sdu.dk
MIP Institute
U. of Southern Denmark
Denmark

Justus Piater
Justus.Piater@ULg.ac.be
Montefiore Institute
University of Liège
Belgium

INTELSIG Technical Report 2007-01-19

Department of Electrical Engineering and Computer Science University of Liège

Hierarchical Integration of Local 3D Features for Probabilistic Pose Recovery

Renaud Detry
renaud.detry@ulg.ac.be
Montefiore Institute
University of Liège
Belgium

Nicolas Pugeault
npugeaul@inf.ed.ac.uk
School of Informatics
Univ. of Edinburgh
United Kingdom

Norbert Krüger
norbert@mip.sdu.dk
MIP Institute
U. of Southern Denmark
Denmark

Justus Piater
Justus.Piater@ULg.ac.be
Montefiore Institute
University of Liège
Belgium

January 19, 2007

This paper presents a two-stage 3D object representation framework. The first stage is an early cognitive vision process that extracts symbolic 3D visual features from stereo views of a scene, combining different visual modalities into condensed local descriptors. The second stage develops a hierarchical model based on probabilistic correspondences and probabilistic relations between 3D features. The bottom of the hierarchy is linked to the first stage visual features; pairs of features that present strong geometric correlation are then iteratively grouped into higher-level meta-features that encode probabilistic relative spatial relationships between their children. The model is instantiated by propagating evidence up and down the hierarchy using the Belief Propagation algorithm, which infers high level features from local evidence and reinforces local evidence from globally consistent knowledge. It is eventually shown how to use our framework to evaluate the pose of a known object in an unknown scene, without point correspondences. An experiment is provided to demonstrate the applicability of the system.

1 Introduction

Objects can be characterized by configurations of parts. This insight is reflected in computer vision by the increasing popularity of representations that combine local appearance with spatial relationships (Burl et al., 1995, 1998; Piater and Grupen, 1999). Such methods are richer and more easily constructed than purely geometric models, more expressive than methods purely based on local appearance such as bag-of-features methods (Leung and Malik, 2001; Dance et al., 2004) and more robust and more easily handled in the presence of clutter and occlusions than methods based on global appearance. Moreover, they not only allow bottom-up inference of object parameters based on features detected in images, but also top-down inference of image-space appearance based on object parameters.

We have recently presented a framework for unsupervised learning of hierarchical representations that combine local appearance and probabilistic spatial relationships (Scalzo and Piater, 2005, 2006). By analyzing a set of training images, our method creates a codebook of features and observes recurring spatial relationships between them. Pairs of features that are often observed in particular mutual configurations are combined into a meta-feature. This procedure is iterated, leading to a hierarchical representation in the form of a graphical model with primitive, local features at the bottom, and increasingly expressive meta-features at higher levels. Depending on the training data, this leads to rich representations useful for tasks such as object detection and recognition from 2D images.

In this paper, we present an extension of this method to 3D features. Here, a low-level feature is an oriented patch in 3–space, annotated by various appearance characteristics (Krüger and Wörgötter, 2005). To learn an object representation, sets of 3D features are constructed using structure-from-motion techniques. A hierarchical object representation is then learned by observing reliable 3D configurations.

To infer the presence of such an object model from a scene represented as a set of 3D features, evidence from local features is integrated through bottom-up inference within the hierarchical model. Intuitively, each observed feature probabilistically votes for all possible object configurations consistent with its pose. During inference, a consensus emerges among the available evidence,

leading to one or more consistent scene interpretations. The system never commits to specific feature correspondences, and is robust to substantial clutter and occlusions.

We illustrate our method on the application of object pose estimation. Object models are learned within a given world reference frame, within which the object is placed in a reference pose. Comparing an instance of the model in an unknown scene with an instance in the learned scene allows us to deduce the object pose parameters in the unknown scene.

2 Early Cognitive Vision

The early cognitive vision stage produces a *compact coding of image information in terms of local multi-modal image descriptors* (Krüger and Wörgötter, 2005). The *multi-modal* qualifier stands for the different visual sub-modalities presented in the descriptor, namely geometric information (image orientation) and structural image information (contrast transition and color).

To begin with, 2D primitives are extracted from 2D imagery. Several local filters are applied to compute the following modalities: orientation, intrinsic dimensionality (degrees of freedom of an image patch), phase (characterization of contrast transition), and color. Primitives are drawn from sparse interest points throughout the image. Whether a point is interesting or not is decided on basis of intrinsic dimensionality.

Following the extraction of 2D primitives comes a 3D reconstruction system for stereo pairs, and for stereo sequences. Stereo matching is applied to stereo pairs to infer 3D feature locations. On top of that, temporal integration procedures allow for the exploitation of stereo sequences. This includes ego-motion computation, stereo using a delayed baseline and accumulation of 3D data over time. The multi-modal information held by a pair of 2D primitives is combined to infer the dual multi-modal information describing the corresponding 3D feature.

3 Hierarchical Features

Our object model consists of a set of generic features organized in a hierarchy. Features that form the bottom level of the hierarchy, referred to as *primitive features*, are linked to early cognitive vision feature observations. The rest of the features are *meta-features* which embody spatial configurations of more elementary features, either meta or primitive. Thus, a meta-feature incarnates the relative configuration of two features from a lower level of the hierarchy.

3.1 Model Presentation

A feature can intuitively be associated to a "part" of an object, i.e. a generic component meant to be instantiated once or several times during a "mental reconstruction" of the object. At the bottom of the hierarchy, *primitive features* correspond to local parts that each have many instances in the object. Climbing up the hierarchy, *meta-features* correspond to increasingly complex parts defined in terms of constellations of lower parts. Eventually, parts become complex enough to satisfactorily represent the whole object. Figure 1 shows

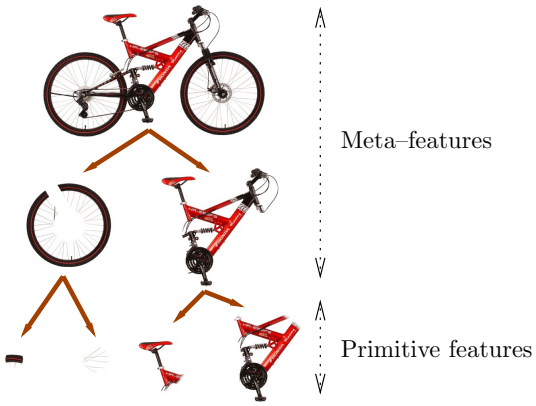


Figure 1: A didactic example of a hierarchy model for a bike.

a didactic example of hierarchy for a bike. The bike is the composition of *frame* and *wheel* features. A wheel is composed of pieces of tire and spokes. The generic piece of tire at the bottom of the hierarchy is a primitive feature; the pieces of tire squared in green in the scene (Figure 2) are instances of that primitive feature.

A meta-feature encodes a spatial relationship between a pair of lower-level *features*. The spatial relationship between a pair of features will often correspond to a *set of relationships* between *feature instances*. In the bike model, let us consider the meta-feature that represents a generic wheel. There are two wheels in the picture; two instances of the wheel feature will be used in a mental reconstruction of the bike. Hence, the meta-feature *bike* will encode two relationships: one between the frame and the front wheel (the front wheel is

on the right side of the frame) and one between the frame and the back wheel (the back wheel is on the left side of the frame).

Figure 2: Instances of the generic piece-of-tire primitive feature in the bike scene.

At the bottom of the hierarchy, primitive features are tagged with an appearance descriptor called *codebook vector*. The set of all codebook vectors forms a *codebook* that makes the link between the object model and feature observations. In this paper, observations come from the early cognitive vision stage. Codebook vectors are thus composed of early cognitive vision structural modalities, namely intrinsic dimensionality, phase, and left–middle–right colors. In summary, information about an object can be stored within the model in the three following forms:

i. The topology of the graph, i.e. the hierarchy that the edge pattern induces between nodes.

ii. The relationships between related nodes.

iii. The codebook vectors annotating bottom level nodes.

When a model is associated to a particular scene (construction or instantiation), features are linked to their instances in that scene. For primitive features, instances will often correspond to observations. For meta-features, instances are abstract entities tagged by a pose. As explained in the next paragraph, these links are not static but probabilistic. This allows the representation of several relationships in one meta-feature, and of several instance poses for the same feature. Also, spatial relationships and feature poses become more flexible: they can be spread or concentrated following the local confidence or variability.

3.2 Model Definition

Formally, the hierarchy is implemented using a Pairwise Markov Random Field (see Figure 3). Features are associated to hidden nodes (in white in Figure 3), and the structure of the hierarchy is reflected by the edge pattern in-between them. Each meta-feature is thus linked to its two child features. Observed variables of the field y_i stand for observational bias.

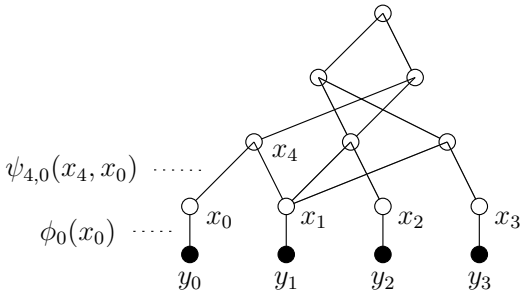


Figure 3: A feature hierarchy Pairwise Markov Random Field. Features correspond to hidden variables, in white. Observations, in black, correspond to early cognitive vision data, linked to bottom-level primitive features.

Parametrization When feature i is linked to its instances in a scene, it is not tagged with a discrete pose, but with a probability density over the pose space

$$SE(3) = \mathbb{R}^3 \times SO(3)$$

held by a random variable x_i.

As noted above, a meta-feature encodes the relationship between its two children. However, the graph records this information in a slightly different way. Instead of recording the relationship between the two child features, the graph records two relationships, between the meta-feature and each of the children. Both ways are strictly equivalent, for any pose associated to the meta-feature whatsoever. The actual pose associated to the meta-feature is non determinant, it depends on the construction procedure that was applied to learn the graph. Recording relationships between meta-features and their children allows to naturally assimilate them to edges of the graph. The relationship between a meta-feature i and one of its children j is parametrized by a *compatibility* potential function $\psi_{ij}(x_i, x_j)$ associated to edge $e_{i,j}$. A compatibility potential function gives, for any given pair of poses for the features it links, the probability of finding that particular configuration for these two features. We only consider rigid body motion relationships. Moreover, relationships are *relative* spatial configurations. Compatibility potentials can thus be represented by a probability density over the meta-feature–to–feature transformation space $SE(3)$.

Finally, the statistical dependency between hidden variable x_i and its observation variable y_i is parametrized by an *observation* potential $\phi_i(x_i)$, also referred to as *evidence* for x_i, which corresponds to the observation spatial distribution for x_i.

Primitive feature instance formally refers to a mode of a primitive feature distribution. A primitive feature instance often corresponds to an observation. However, observations are prior knowledge. Primitive feature instances are posterior; they depend on observations *and* all features of the hierarchy. Owing to inference mechanisms presented in the next paragraph, if a potential observation is occluded, a primitive feature instance may appear at its place. If an observation is wrongly associated to a primitive feature, it may not appear in the primitive feature density.

Inference An interesting aspect of graphical models is that they provide a formalism appropriate for the definition of elaborated *inference algorithms*, i.e. algorithms for efficient computation of statistical quantities. An efficient inference algorithm is essential to the hierarchical model, for it provides the mechanism that will let features communicate and propagate information.

The inference algorithm is currently the Belief Propagation (BP) algorithm (Yedidia et al., 2002; Jordan and Weiss, 2002). Belief Propagation is based on incremental updates of marginal probability estimates, referred to as *beliefs*. The belief at feature i is denoted

$$b(x_i) \approx \mathbf{p}(x_i|y) = \int ... \int \mathbf{p}(x_1, ..., x_N|y) \, dx_1 ... dx_{i-1} dx_{i+1} ... dx_N$$

where y stands for the observational bias. During the execution of the algorithm, messages are exchanged between neighboring features (hidden nodes). A message that feature i sends to feature j is denoted $m_{ij}(x_j)$, and contains feature i's belief about the state of feature j. In other words, $m_{ij}(x_j)$ is a real positive function proportional to feature i's belief about the plausibility of finding feature j in pose x_j. At any time during the execution of the algorithm, the current pose belief (or marginal probability estimate) for feature i is the normalized product of the local evidence and all incoming messages, as

$$b_i(x_i) = \frac{1}{Z} \phi_i(x_i) \prod_{j \in \text{neighbors}(i)} m_{ji}(x_i). \quad (1)$$

To prepare a message for feature j, feature i starts by computing a local "pose belief estimation", as the product of the local evidence and all incoming messages *but* the one that comes from j. This product is then multiplied with the compatibility potential of i and j, and marginalized over x_i. The complete expression for the message is

$$m_{ij}(x_j) = \int \psi_{ij}(x_i, x_j) \phi_i(x_i) \prod_{k \in \text{neighbors}(i) \setminus j} m_{ki}(x_i) dx_i. \quad (2)$$

As we see, the computation of a message doesn't directly involve the complete local belief (1). In general, the explicit belief for each node is computed only once, after all desirable messages have been exchanged.

When BP is finished, collected evidence has been propagated from primitive features to the top of the hierarchy, permitting inference of top feature marginal pose densities. Furthermore, regardless of the propagation scheme (message update order), the iterative aspect of the message passing algorithm ensures that the global belief about the object pose –concentrated at the top nodes– has at some point been propagated back down the hierarchy, reinforcing globally consistent evidence and permitting the inference of occluded features. While there is no theoretical proof of BP convergence for loopy graphs, empirical success has been demonstrated in many situations.

Nonparametric Representation We opted for a non-parametric approach to probability density representation. A density is simply represented by a set of points; the local density of these points in space is proportional to the actual probabilistic density in that region. Compared to usual parametric approaches which involve a limited number of parametrized kernels, problems like fitting or the choice for a number of components can be avoided. Also, no assumption concerning the shape of the density has to be made.

Density points live in the pose space $SE(3)$. The location/translation component is obviously parametrized by a 3–vector. For the orientation/rotation component it was decided to prefer quaternions over rotation matrices, for they provide a well fitted formalism for rotation manipulation like composition or metric definition (see Kuffner, 2004; Karney, 2006).

For inference, we use a variant of BP, Nonparametric Belief Propagation, which essentially de-

velops an algorithm for BP message update (2) in the particular case of continuous, non-Gaussian potentials (Sudderth et al., 2003). The underlying method is an extension of particle filtering; the representational approach is thus nonparametric and fits our model very well.

3.3 Model Instantiation

Model instantiation is responsible for detecting instances of a completed object model in a scene. It will provide pose densities for all features of the model, indicating where the learned object is likely to be seen. Instantiating a model in a scene amounts to inferring posterior marginal densities for all features of the hierarchy. Thus, once priors (observation potentials) have been defined, the instantiation is achieved by generic algorithms. The algorithm currently in use is the Belief Propagation algorithm, described at Section 3.2.

For primitive features, evidence is estimated from feature observations. Observations are classified with the primitive feature codebook; for each primitive feature i, its observation potential $\phi_i(x_i)$ is estimated from observations that are in the class i's codebook vector. For meta-features, evidence is uniform.

3.4 Model Construction

The construction procedure is a rather straightforward transcription of the model itself. It starts by building a codebook of feature observations, by clustering them in the appearance space. The number of classes is a parameter to the system. These classes are then used to initialize the first level of the graph:

1. A primitive feature is created for each class;
2. Each primitive feature is tagged with the codebook vector (cluster center) that characterizes its corresponding class;
3. The spatial probabilistic density of each primitive feature is computed from the spatial distribution of observations of the corresponding class. Since we are using non-parametric representations, the set of observations bound to each primitive feature can be directly used as a density representation.

Clustering is necessary to bring flexibility to the link between visual data and the object model, by relaxing the association of observations to primitive features. Also, it allows to control the number of primitive features, to conserve reasonable computational efficiency.

After primitive features have been computed, the graph is built incrementally, in an iterative manner. The aim of the construction algorithm is to extract feature co-occurrence statistics. Features that tend to occur at non-accidental relative positions are repeatedly grouped into a higher-level meta-feature. At each step, the top level of the graph is searched for strongly correlated pairs of features. The k most strongly correlated pairs are selected to form the k meta-features of the next level. The number of meta-features created at each step is a parameter, which is usually kept equal to the initial number of classes. The search for strong feature combinations is the operation responsible for the *topology* of the graph, and incidentally for the structure of the hierarchy.

The next two operations consist of

1. The synthesis of a spatial probability distribution for each new meta-feature, from a combination of the child densities. The meta-feature is placed in the middle of its children, location- and orientation-wise: the meta-feature distribution will be dense between dense regions of the child distributions.

2. The extraction of spatial relations between each meta-feature and its children, which defines the compatibility potentials. This is achieved by repeatedly taking a pair of samples, one from the parent distribution and one from a child distribution. The spatial relationships between a large number of these pairs form the relationship distribution between the parent and that child.

Where the search for strong combinations was responsible for the topology of the graph, the extraction of spatial relations is responsible for the *parametrization* of the graph through the definition of compatibility potentials associated with edges between adjacent features. This parametrization constitutes the principal outcome of the learning algorithm.

Incremental construction of the graph can, in principle, continue indefinitely, growing an ever-richer representation of the observed scene. The number of levels is a parameter that is set to reach the desired level of abstraction.

4 Object Pose Estimation

Since features at the top of the graph represent the whole object, they will present unimodal and relatively concentrated densities. These densities can be used to estimate the object pose. There is one detail to keep in mind though. The top features all represent the whole object, but as different recursive combinations of features. The poses of the top features are thus expected to be different.

Let us consider a model for a given object, and a pair of scenes where the object appears. In the first scene, the pose of the object is known and denoted π_o, . In the second scene, the pose of the object is unknown. The application our method to estimate the pose of the object in the second scene goes as follows:

1. Instantiate the object model in scene 1. For every top feature i of the instantiated graph, compute an aggregate *feature pose* π_1^i from the unimodal densities.

2. Instantiate the object model in scene 2. For every top feature of that graph, compute an aggregate *feature pose* π_2^i.

   For all top level feature i, the transformations from π_1^i to π_2^i should be very similar; let us denote the mean transformation t. This transformation corresponds to the rigid body motion between the pose of the object in the first scene and its pose in the second scene.

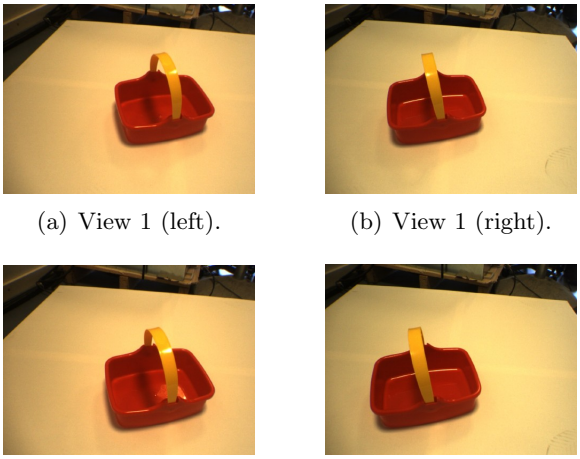
3. From the rigid body motion t between the scenes, it is straightforward to compute the *object pose* in the second scene, by applying t to π_o.

A prominent aspect of this procedure is its ability to recover an object pose without explicit point-to-point correspondences. The estimated pose emerges from a negotiation in all available data.

5 Experiment and Results

This section presents a pose estimation experiment. Input imagery is a set of two stereo pairs, as shown in Figure 4. The motion between the first and second views is a camera translation of 70 units in the direction of the object; the object itself has a size of about 300 units. Early cognitive vision produces a set of feature observations for each view. In order to learn a clean model, background noise is removed from the training view.

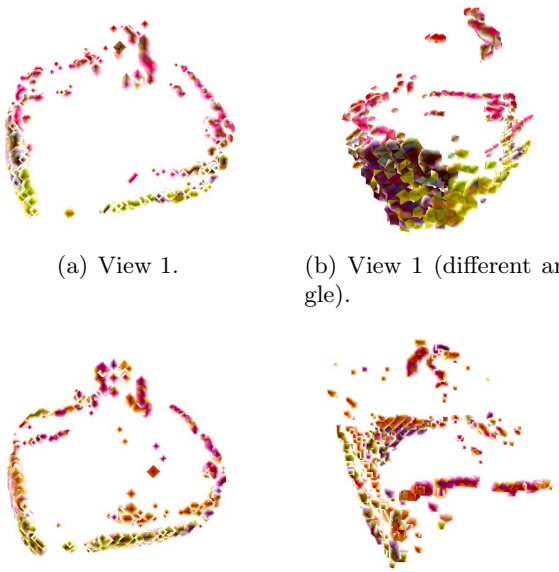
Experiment We will use the first view to learn a model of the basket (training). We will then estimate the pose of the basket model top features in the second view (estimation), and evaluate the result using the ground truth motion (evaluation).



(a) View 1 (left). (b) View 1 (right). (c) View 2 (left). (d) View 2 (right).

Figure 4: Input imagery.

For training, observations from the first view are clustered in 11 classes, which sets up the first level of the basket model. The result is shown at Figure 5(a) where color stands for class membership; Figure 5(b) shows the same set from a different point of view. A hierarchy of 7 levels is constructed, following details given in Section 3.4.



(a) View 1. (b) View 1 (different angle). (c) View 2. (d) View 2 (different angle).

Figure 5: Feature observations (early cognitive vision data).

During the construction of a model, its features are automatically associated to feature instances in the training scene (i.e. a density is available for every feature). Hence, we do not need to instantiate the model in the first scene to get spatial densities for top level features, as is done in Section 4; we can directly compute an aggregate feature pose π_1^i for every top feature i of the model.

For estimation, we instantiate the model in the second view. Since the basket is only present once in the second view, top level features should, after instantiation, present unimodal densities; we can safely compute a mean pose π_2^i for each of them.

For evaluation, one may try to compare the transformations between top feature poses π_1^i and π_2^i, for all i, to a translation of 70u. However, in a RBM transformation, translation and rotation are not independent: an error on orientation will have an impact on translation, as illustrated at Figure 6. For this reason, we proceed differently. We apply the ground truth motion (70u translation) to all 11 training poses π_1^i and denote the resulting poses $\pi_{1\to 2}^i$. Ideally, we would have

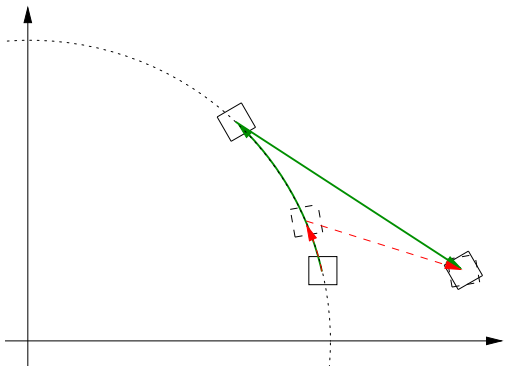


Figure 6: Two rather different transformations can yield a similar transformed location.

$$\pi_{1\to 2}^i = \pi_2^i \qquad \forall i.$$

Hence, to evaluate the pose estimation, we compare $\pi_{1\to 2}^i$ and π_2^i, for all i. Comparison relies on the distances between translations and orientations. For orientations, we use a metric based on quaternion representation. The distance between two orientations is defined by the shortest path on the 3–Sphere between the quaternion representations q and q' (see Kuffner, 2004):

$$\mathbf{d}(q, q') = \arccos(|q \cdot q'|). \tag{3}$$

This metric is roughly reflective of our intuitive notion of a distance between rotations. Results are presented at Table 1, denoting a pose and its translation-rotation components

$$\pi = (\lambda, \theta)$$

where λ is the location and θ is the orientation.

i	$\mathbf{d}(\lambda_{1\to 2}^i, \lambda_2^i)$ (in length units)	$\mathbf{d}(\lambda_{1\to 2}^i, \theta_{1\to 2}^i)$ (in radians)
1	12.0	0.0578
2	12.7	0.0505
3	13.4	0.0540
4	13.5	0.0695
5	11.2	0.0701
6	12.8	0.0743
7	13.2	0.0688
8	11.7	0.0460
9	11.8	0.0763
10	12.0	0.0620
11	12.5	0.0572
Mean	12.4	0.0624

Table 1: Distances between $\pi_{1\to 2}^i$ and π_2^i.

Interpretation Distances between feature locations $\mathbf{d}(\lambda_{1\to 2}^i, \lambda_2^i)$ $(i = 1, .., 11)$ have two interpretations. First, they need to be compared to the 70u object translation responsible for a global change in early cognitive vision observations, appearance-wise (different lighting, different viewpoint) and location-wise (observations are not sampled from the same points of the object, the number of feature observations will differ). The mean location error relatively to the translation length is

$$\frac{\sum_{i=1}^{11} \mathbf{d}(\lambda_{1\to 2}^i, \lambda_2^i)}{11 \cdot 70} \simeq 17.7\%.$$

Second, distances between feature locations need to be compared to the global size of the basket which is about 300u wide:

$$\frac{\sum_{i=1}^{11} \mathbf{d}(\lambda_{1\to 2}^i, \lambda_2^i)}{11 \cdot 300} \simeq 4.1\%.$$

Distances between feature orientations can be interpreted by comparing them with the maximum distance between two orientations. Equation (3) shows that the maximum distance happens when q and q' are orthogonal in $\mathbb{R}^4$, yielding a null dot product and a distance of $\pi/2$. The relative orientation accuracy lies around

$$\frac{\sum_{i=1}^{11} \mathbf{d}(\theta_{1\to 2}^i, \theta_2^i)}{11 \cdot \pi/2} \simeq 4.7\%.$$

6 Conclusion

We presented a probabilistic framework for hierarchical object representation. A hierarchy is implemented by a Pairwise Markov Random Field in which hidden nodes represent generic features, and edges model the abstraction of highly correlated features into a higher-level meta-feature. Once PMRF evidence is extracted from observations, posterior marginal pose densities for all features of the graph are inferred by the Belief Propagation algorithm.

Posterior pose densities can be used to compute a pose for a known object in an unknown scene, which is demonstrated though a rigid body motion estimation experiment. We are thus able to achieve pose recovery without prior CAD model, and without point correspondences. In the context of the PACO+[1] project, the two-stage framework presented in this text can be fed with imagery from the robot visual sensors, and learn to recognise objects, along with their pose. This information can further be used to generate an appropriate action, or decide on deeper exploration.

References

M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *Intl. W. on Automatic Face and Gesture Recognition*, 1995. URL ftp://vision.caltech.edu/pub/tech-reports-vision/IWAFGR95.ps.Z.

Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 628–641, London, UK, 1998. Springer-Verlag. ISBN 3-540-64613-2. URL http://www.vision.caltech.edu/publications/ECCV98-recog.pdf.

Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004. URL http://www.xrce.xerox.com/Publications/Attachments/2004-010/2004_010.pdf.

Michael I. Jordan and Yair Weiss. Graphical models: Probabilistic inference. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks, 2nd edition*. MIT Press, 2002. URL http://www.cs.berkeley.edu/~jordan/papers/jordan-weiss.ps.gz.

Charles F. F. Karney. Quaternions in molecular modeling. *J. Mol. Graph. Mod.*, 25, 2006. doi: 10.1016/j.jmgm.2006.04.002. URL http://charles.karney.info/papers/jmgm06.pdf.

Norbert Krüger and Florentin Wörgötter. Multimodal primitives as functional models of hypercolumns and their use for contextual integration. In Massimo De Gregorio, Vito Di Maio, Maria Frucci, and Carlo Musio, editors, *BVAI*, volume 3704 of *Lecture Notes in Computer Science*, pages 157–166. Springer, 2005. ISBN 3-540-29282-9. URL http://media.aau.dk/~nk/publications/bvaiPrim.pdf.

James Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *Proc. 2004 IEEE Int'l Conf. on Robotics and Automation (ICRA 2004)*. IEEE, May 2004. URL http://www.kuffner.org/james/papers/kuffner_icra2004.pdf.

Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1): 29–44, 2001. URL http://www.cs.berkeley.edu/~malik/papers/LM-3dtexton.pdf.

Justus H. Piater and Roderic A. Grupen. Toward learning visual discrimination strategies. In *CVPR*, pages 1410–1415. IEEE Computer Society, 1999. ISBN 0-7695-0149-4. URL http://www-robotics.cs.umass.edu/Papers/CVPR99.ps.gz.

Fabien Scalzo and Justus H. Piater. Statistical learning of visual feature hierarchies. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 44, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2-3. doi: http://dx.doi.org/10.1109/CVPR.2005.532. URL http://www.montefiore.ulg.ac.be/~scalzo/pubs/cvprWa.pdf.

Fabien Scalzo and Justus H. Piater. Unsupervised learning of dense hierarchical appearance representations. In *ICPR '06: Proceedings*

[1] http://www.paco-plus.org/

of the 18th International Conference on Pattern Recognition (ICPR'06), volume 2, pages 395–398, Washington, DC, USA, August 2006. IEEE Computer Society. ISBN 0-7695-2521-0. doi: http://dx.doi.org/10.1109/ICPR.2006.1144. URL http://www.montefiore.ulg.ac.be/~scalzo/pubs/icpr.pdf.

Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. *cvpr*, 01:605, 2003. ISSN 1063-6919. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2003.1211409. URL http://ssg.mit.edu/nbp/papers/cvpr03.pdf.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2002. URL http://www.merl.com/publications/TR2001-022/.

Object Action Complexes as an Interface for Planning and Robot Control

Christopher Geib, Kira Mourão, Ron Petrick,
Nico Pugeault, and Mark Steedman
School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, Scotland
Email: cgeib@inf.ed.ac.uk

Norbert Krueger
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
DK-5230 Odense M, Denmark

Florentin Wörgötter
Institute for Informatics
University of Göttingen
37083 Göttingen, Germany

***Abstract*— Much prior work in integrating high-level artificial intelligence planning technology with low-level robotic control has foundered on the significant representational differences between these two areas of research. We discuss a proposed solution to this representational discontinuity in the form of object-action complexes (OACs). The pairing of actions and objects in a single interface representation captures the needs of both reasoning levels, and will enable machine learning of high-level action representations from low-level control representations.**

I. Introduction and Background

The different representations that are effective for continuous control of robotic systems and the discrete symbolic AI presents a significant challenge for integrating AI planning research and robotics. These areas of research should be able to inform one another. However, in practice, many collaborations have foundered on the representational differences. In this paper, we propose the use of object-action complexes[1] to address the representational difference between these reasoning components.

The representations used in the robotics community can be generally characterized as vectors of continuous values. These vectors may be used to represent absolute points in three dimensional space, relative points in space, joint angles, force vectors, and even world-level properties that require real-valued models [2]. Such representations allow system builders to succinctly specify robot behavior since most if not all, of the computations for robotic control are effectively captured as continuous transforms of continuous vectors over time. AI representations, on the other hand, have focused on discrete symbolic representations of objects and actions, usually using propositional or first-order logics. Such representations typically focus on modeling the high-level conceptual state changes that result from action execution, rather than the low-level continuous details of action execution.

Neither of the representational systems alone cover the requirements for controlling deliberate action, however, both levels seem to be required to produce human level behavioral control. Our objective is to propose an interface representation that will both allow the effective exchange of information between these two levels and the learning of high level action representations on the basis of the information provided by the robotic control system.

Any such representation must provide clear semantics, and be easily manipulable at both levels. Further it must leverage the respective strengths of the two representation levels. In particular, the robotic control system's access to the actual physical state of the world through its sensors and effectors is essential to learning the actions the planning system must reason about. Each low-level action executed by the robot offers the opportunity to observe a small instantiated fragment of the state transition function that the AI action representations must capture. Therefore, we propose that the robotic control system provide fully instantiated fragments of the planning domains state transition function, that is captured during low-level execution, to the high-level AI system to enable the learning of abstract action representations. We will call such a fragment an *instantiated state transition fragment (ISTF)*, and define it to be a situated pairing of an object and an action that captures a small, but fully instantiated, fragment of the planning domain's state transition function. The process of learning domain invariants from repeated, reproducible instances of very similar ISTFs will result in generalizations over such instances that we will call *object-action complexes (OACs)*. To see how this is done, the rest of this paper will first discuss a detailed view of a robot control system, then we will discuss an AI planning level description of the same domain. We will then more formally define ISTFs and OACS, show how ISTFs can be produced by the robot control system, and how OACs relate to the AI planning level description. We will then discuss the learning of OACs on the basis of ISTFs.

To do all this, we require a particular domain for the robot to interact with. Imagine a relatively standard but simple robot control scenario illustrated in Figure 1. It consists of an arm with a gripper, a table with two light colored cubes and one dark colored cube. The robot has the task of placing the cubes into a box, also located on the table. We will also assume the robot is provided with a camera to view the objects in the domain. However, at the initial stage, the system does not have any knowledge of those objects. The only initial world knowledge available to the system is provided by the vision module, and the hard-coded action reflexes that this visual input can elicit.

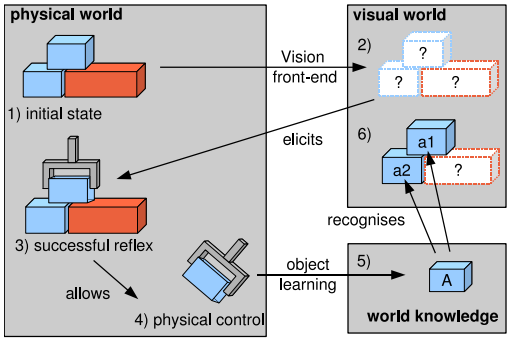


Fig. 1. Illustration of how object classes are discovered from basic uninformed reflex actions.

II. Vision-based Reflex Driven Discovery of Objects and Affordances

We assume a vision front-end based on an *Early Cognitive Vision* framework (see [3]) that provides a scene representation composed of local 3D edge descriptors that outline the visible contours of the scene [4]. Because the system lacks knowledge of the objects that make up the scene, this visual world representation is *unsegmented*: descriptors that belongs to one of the objects in the scene are not explicitly distinct from the ones belonging to another object, or to the background (this is marked by question marks in Figure 1-2). This segmentation problem has been largely addressed in the literature [5], [6], [7]. However, while these segmentation methods are purely vision-based and do not require of the agent to interact with the scene they are unsatisfying for our purpose because they assume certain qualities from the objects in order to segment them: e.g., constant color or texture, moving objects, etc.

Instead we will approach the problem from another angle: we will assume that the agent is endowed with a basic reflex action [8] (Figure 1-3) that is elicited directly by specific visual feature combinations in the unsegmented world representation. The outcome of these reflexes will allow the agent to gather further knowledge about the scene. This information will be used to segment the visual world into objects and identify their affordances.

We will only consider a single kind of reflex here: the agent tries to grasp any planar surface in the scene.[1] The likely locations of such planar surfaces are inferred from the presence of a coplanar pair of edges in the unsegmented visual world. This type of reflex action is described in [8]. Every time the agent executes such a reflex, haptic information allows the system to evaluate the outcome: either the grasp was successful and the gripper is holding something, or it failed and the gripper closed on thin air. A failed attempt drives the agent to reconsider its original assumption (the presence of a graspable plane at this location in the scene), whereas a successful attempt confirms the feasibility of this reflex. Moreover, once a successful grasp has been performed, the agent has gained physical control over some part of the scene (i.e. the object grasped, Figure 1-4). If we assume that we know the full kinematics of the robot's arm (which is true for an industrial robot), it is then possible to segment the grasped object from the rest of the visual world as it is the only part that moves synchronously with the arm of the robot. At this point a new "object" relevant for the higher level planning model is "born".

Having physical control of an object allows the agent to segment it and to visually inspect it under a variety of viewpoints and construct an internal representation of the full 3D shape of the object (see [9]). This shape can then be stored as the description of newly discovered class **A** (Figure 1-5) that affords **grasp-reflex-A** encoding the initial reflex that "discovered" the object.

The object held in the gripper is the first instance **a1** of the class **A**. The agent can use its new knowledge of class **A** to reconsider its interpretation of the scene: using a simple object recognition process (based on the full 3D representation of the class), all other instances (e.g., in our example **a2**) of the class in the scene are identified and segmented from the unknown visual world.

Thus through a reflex-based exploration of the unknown visual world object classes can be discovered by the system until it achieves an informed, fully segmented representation of the world, where all objects are instances of symbolic classes and carry basic affordances.

To distinguish the specific successful instances of the robot's reflexes, we will refer to the specific instance of the reflex that was successful for the object as a particular *motor program*. Note that such motor programs are defined relative to a portion of an object, in our example, the surface that was grasped. We will extend this by assuming *all* motor programs can be defined relative to some object.

The early cognitive vision system [4], the grasping reflex [8] as well as the accumulation mechanism [9] that together provides a segmentation of the local feature descriptors into independent objects currently exist in one integrated system that we will use as a foundation for this architecture.

III. Representing AI Planning Actions

As we have noted, we can also model this robot domain scenario using a formal AI representation. In this case, we will formalize the robot domain using the Linear Dynamic Event Calculus (LDEC) [10], [11], a logical language that combines aspects of the situation calculus with linear and dynamic logics, to model dynamically-changing worlds[12], [13], [14].

Our LDEC representation will define the following actions.

Definition 1: High-Level Domain Actions

- *grasp*(x) – *move the gripper to pick up object* x,
- *ungrasp*(x) – *release the object* x *in the gripper,*
- *moveEmptyGripperTo*(ℓ) – *move an empty gripper to the specified location* ℓ,
- *moveFullGripperTo*(ℓ) – *move a full gripper to the specified location* ℓ.

[1] Note that other kind of reflex actions could be devised to enable other basic actions than grasping.

These actions represent higher level counterparts of some of the motor programs available to the robot controller, but already these actions incorporate elements of the state of the world that are not part of robotic control representations of actions. For instance, *ungrasp* models an action that is quite similar to a motor program that performs this operation. Actions like *moveEmptyGripperTo* and *moveFullGripperTo*, on the other hand, are much more abstract and encode information about the state of the world (i.e. the gripper is empty or full). Note that in this case the actions partition the low-level "move gripper" motor-programs into two separate actions that, as we will see, can more readily be learned from the available ISTFs. This representation will also allow us to bypass the learning of the conditional effects[15] of such actions.

Our LDEC representation will also include a number of high-level properties.

Definition 2: High-Level Domain Properties

- $graspable(x)$ – *a predicate that indicates whether an object x is graspable or not,*
- $gripperLoc = \ell$ – *a function that indicates the current location of the gripper is* ℓ*,*
- $objInGripper = x$ – *a function that indicates the object in the gripper is x; x is nil if the gripper is empty,*
- $objLoc(x) = \ell$ – *a function that indicates the location of object x is* ℓ*.*

Finally, we also specify a set of "exogenous" domain properties.

Definition 3: Exogenous Domain Properties

- $over(x) = \ell$ – *a function that returns a location* ℓ *over the object x,*
- $locOnTable(\ell_1) = \ell_2$ – *a function that returns a location* ℓ_2 *relative to the table (e.g., on the table or in a box) for another location* ℓ_1 *above the table.*

Like the properties in Definition 2, the exogenous properties model high-level features of the domain. However, unlike domain properties that are directly tracked by the high-level AI model; exogenous properties are information provided to the high-level AI system by some external (possibly lower level) source. (We will say more about exogenous properties in Section VI.)

Using these actions and properties we can write LDEC axioms that capture the dynamics of the robot scenario described in Table I). Action precondition axioms describe the properties that must hold of the world to apply a given action (i.e., affordances), while the effect axioms characterize what changes as a result of the action. These axioms also encode the STRIPS assumption: fluents that aren't directly affected by an action are assumed to remain unchanged by that action [16].

We note our LDEC axiomatization is readily able to accommodate the *indexical*, or relative information. For example, an instantiated function like $over(box1)$ represents a form of indexical knowledge, rather than a piece of definite information like the coordinates of the box in three dimensional space. Moreover, our LDEC axiomatization can model spatial relationships expressed with respect to objects. For instance, $moveFullGripperTo(over(box1))$ can represent an action instance that moves the object in the gripper to a location "over *box1*"

TABLE I
LDEC Axiomatization of High-Level Domain Actions

LDEC Action Precondition Axioms

$objInGripper = nil \wedge graspable(x) \Rightarrow affords(grasp(x))$
$objInGripper = x \wedge x \neq nil \Rightarrow affords(ungrasp(x))$
$objInGripper = nil \Rightarrow affords(moveEmptyGripperTo(\ell))$
$objInGripper = x \wedge x \neq nil \Rightarrow affords(moveFullGripperTo(\ell))$

LDEC Effect Axioms

$\{affords(grasp(x))\} \multimap$
$[grasp(x)]\, objInGripper = x \wedge gripperLoc = objLoc(x)$
$\{affords(ungrasp(x))\} \multimap$
$[ungrasp(x)]\, objInGripper = nil \wedge objLoc(x) = locOnTable(objLoc(x))$
$\{affords(moveEmptyGripperTo(\ell))\} \multimap$
$[moveEmptyGripperTo(\ell)]\, gripperLoc = \ell$
$\{affords(moveFullGripperTo(\ell))\} \multimap$
$[moveFullGripperTo(\ell)]\, gripperLoc = \ell \wedge objLoc(objInGripper) = \ell$

Intuitively, the information encoded in a collection of LDEC axioms captures a generalization of the information in a larger set of ISTFs. The action precondition axioms capture information from the initial state of an ISTF and the action executed, while the effect axioms capture the generalities for the initial state to final state mappings from an ISTFs. As such we believe they can be learned from the ISTFs.

It is easy to show that this representation supports high-level planning. For instance, with these axioms it is trivial for an AI planner to construct the following simple plan:

$$[grasp(obj1); moveFullGripperTo(over(box1)); ungrasp(obj1)],$$

to put an object $obj1$ into $box1$, from a state in which the robot's gripper is empty. However, building even this sort of simple plan from first principles is well beyond the capability of the robot controller alone.

So far we have shown that a low level robot controller is capable of producing ISTFs for a domain, we have shown a way an AI level planner could formalize the same domain, and we have shown the necessity of using the AI planner with the robot controller to produce high level behavior. In the remainder of the paper we will outline a process whereby we can learn the AI level representation from the ISTFs produced by the robot controller.

IV. Bridging Robot Control and Planning with ISTFs and OACs

With these two views of the problem in hand, we now, consider how we can bridge the two representational levels. We see that we can obtain a wealth of object-centric information each time the robotic system successfully grasps an object: the object grasped, the type of grasping reflex used, the relative position of the gripper, the fact that the object has been

effectively grasped and is now in the gripper instead of being on the table, etc. This association of before and after states of a particular "grasp" motor program with a specific domain object meets our definition of an ISTF. It completely describes a fragment of the planning domain's transition function.

We more formally define an ISTF as a tuple $\langle s_i, mp_j, Obj_{mp_i}, s_{i+1} \rangle$ comprised of the initial sensed state of the world s_i, a motor program instance mp_j, the whole object containing the component the motor program was defined relative to Obj_{mp_i}, and the state that results from executing the motor program s_{i+1}. Keep in mind that the state representations for this ISTF contain all of the information the robot has about the two states of the world. Some of which may be relevant some of which may be completely irrelevant to the outcome of the action.

It will be the task of the learning module to abstract away this irrelevant information from the ISTFs to produce OACs that contain only the relevant instantiated information needed to effectively predict the applicability of the action and the likely effects of the action. This is only possible if the system is provided with multiple encounters with reproducible ISTFs. Thus as the system repeatedly interacts with the world it is presented with multiple very similar ISTFs which it generalizes into OACs, thereby learning a representation that is not unlike the one we specified in the previous section.

On this basis, we define an OAC as a generalized ISTF tuple: $\langle S_i, MP_j, Obj_k, S_{i+1} \rangle$ comprised of two abstracted states (S_i and S_{i+1}) a set of motor programs MP_j, and an object class Obj_k. The initial state of the world, S_i, is abstracted to contain only those properties that are necessary for any of the set of motor-programs in MP_j when acting on an object of class Obj_k to result in an state that is satisfied but the abstracted state definition S_{i+1}. Thus such an OAC contains all of the information found in our initial LDEC definitions for this domain.

Given the parallels to LDEC representations how are OACs different? The answer to this is, a very subtle point. OACs constrain the kinds of LDEC rules that can be learned. First OACs distribute information in a subtly different manner than LDEC rules. An OAC contains information normally found in two different parts of the LDEC representation. By bringing together information found in precondition rules with the effect rules and the object in question they allow learning to take place that previously couldn't have been accomplished. Second the heavy use of the object and the object centeredness of OACS produce LDEC representations that easily lend themselves to a simple forward looking planning algorithm that is heavily directed by the affordances of the available objects. Third and finally the use of OACs constrains the LDEC representations to a simple form of axioms that are easier to learn. For example, without more complex machinery, OACs induced from ISTFs are not able to create action representations with conditional effects. Learning such conditional effects of actions is a significant problem for other approaches.

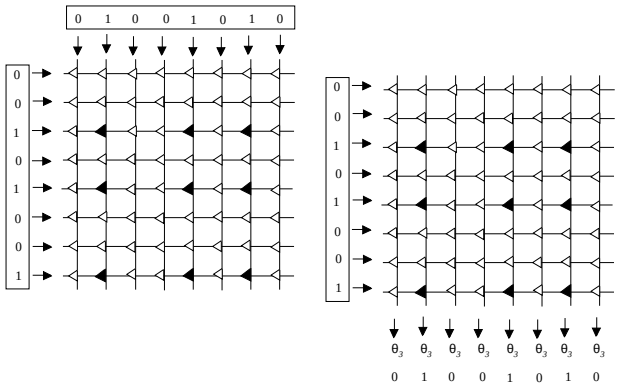


Fig. 2. Hetero-associative net: Storage and Retrieval

V. Learning Action Representations

The ability of a low level robotic control system to identify world-level objects only takes us part of the way to kind of representation we have just described. We must learn from the ISTFs coherent, high-level actions. Our current proposal for learning such action representations involves the use of Willshaw nets or Associative Nets(AN).

ANs were first was described in [17], [18] following early work by [19] and [20] extended by [21] and [22]. They illustrate three basic properties which are characteristic of mechanisms involved in phenomena of human memory and attention: 1) non-localized storage ("Distributivity"), 2) recovery of complete stored patterns from partial or noisy input ("Graceful Degradation"), and 3) effective functioning even in the face of damage ("Holographic Memory").

ANs associate pairs of input and output vectors using a grid of horizontal input lines and vertical output lines with binary switches (triangles) at the intersections (Figure 2). To store an association between the input vector and the output vector, switches are turned on (black triangles) at the intersection of lines which correspond to a 1 in both input and output patterns.

To retrieve the associate of the input, a signal is sent down each input line corresponding to a 1 in the input. When this signal encounters an "on" switch, it increments the signal on the corresponding output line by one. The output lines are then thresholded at a level corresponding to the number of "on" bits in the input. If we store an input pattern with itself as output (an *auto-associative net*), ANs can be used to complete partial patterns, as needed to recall perceptually non-evident properties of objects, such as the fact that the red cube on the table affords grasping. This is exactly the information that is encoded in action precondition axioms. Further it is worthwhile to notice that all of the information needed for this AN is available in each new instance of an ISTF. In this case, the input and output patterns for the AN are the same: the initial state, action, and object for a cluster of reproducable ISTFs observed in the course of interacting with the world. We thereby use repeated presentations of very similar ISTFs (clustered by action and object) to train auto-associative ANs to effectively store and retrieve associations between the LDEC action precondition axioms and the property of *affording* such

LDEC operators.

Now consider the LDEC style effect axioms. Rather than using an auto-associative net we can use a hetero-associative network for this task. In this case, we again use the initial state, action, and object as the input pattern from each ISTF, however as the output pattern we use the resulting state from the ISTF. This will allow us to learn and retrieve the state-change transitions associated with LDEC operators, with states represented as sparse vectors of relevant facts or propositions.

Thus, we hypothesize that such associations can be learned in ANs using repeated presenations of reproducable ISTFs using the Perceptron Learning Algorithm (PLA). We replace the binary AN switches with continuous valued switches and use multiple ISTFs that have the same action, object, and resulting state and the PLA to adjust the weights on the relevant switches. We believe that such an approach can learn consistent state changes or actions, and learn the association between preconditions and associated affordances.

More specifically, in the envisioned scenario, as the robot controller explores the world, successful grasps will produce ISTFs. On the basis of multiple reproducable experiences of particular ISTFs we can learn the instantiated versions of the precondition axioms and the effect axioms for the robots actions. The resulting state in each ISTF will vary only in terms of the object-type grasped and the grippers pose. Further, the invariants can be learned as a basis for classifying the world into object classes and action types. As we have discussed, identifiers for actions-types can then be associated with the input conditions for the action via an auto-associative net. Such affordances are added by adding new input and output lines to the net for the new affordance, and using the existing learning algorithm.

This network can be presented with a possibly incomplete set of properties representing the current state of the world, and used to retrieve a complete model of the world state, including non-perceptually available associates including the affordances and object classes.(Figure 3) For ease of exposition, in this and the following figures we will continue to show weights of 0 and 1. The full pattern including affordances can then be input to the other hetero-associative net, and used to retrieve the effects of carrying out particular actions. (Figure 4).

If the output states and affordances are the same following two different grasp actions for a particular input state, then clearly the effects (as far as the learner and planner are concerned) of the two grasps are the same for that input. If the effects are the same for all inputs then the grasps are equivalent and can be collapsed together. We discuss this next.

A. Learning Multiple Grasp Actions

Recall from our discussion of the high-level action *grasp* that at the lower level there may in fact be many low-level grasps available to the robot at any point. While many of these grasping actions may have effects that are indistinguishable from one another, there will also be grasping actions that result in very different effects. Given this, and our desire to avoid the difficulties of learning actions with conditional

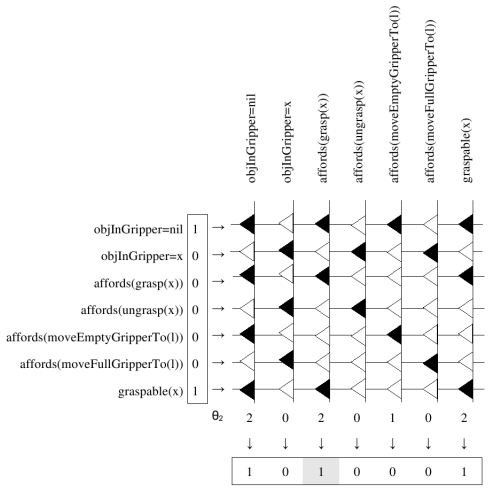


Fig. 3. Retrieval of *affords*(*grasp*(*x*)) from *objInGripper* = *nil* ∧ *graspable*(*x*) in the loaded auto-associative net

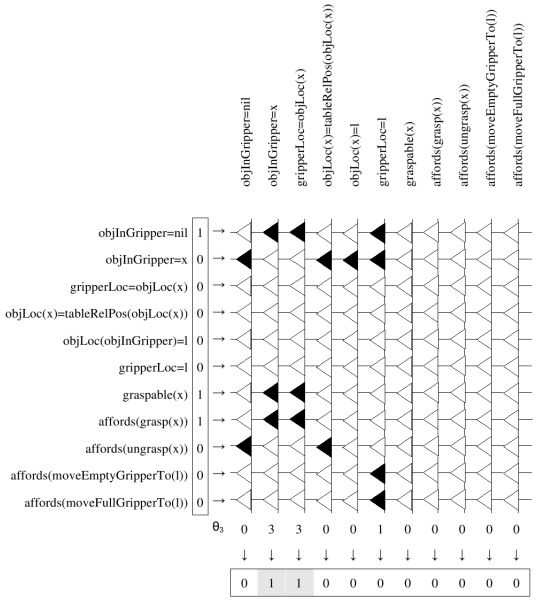


Fig. 4. Retrieval of effect *grasp*(*x*) from the hetero-associative net

effects, it becomes clear that we will need multiple grasp actions at the higher level of abstraction. To distinguish these actions and their effects during planning and learning we will introduce multiple predicates indicating "graspability" by particular motor programs.

Our learning process now operates as follows: when an object is "born" at the lower level of representation (See Section II), the message for the addition of the object (e.g., *obj*23) should include an identifier for the specific action that was executed (e.g., *grasp*28, *grasp*95, etc.) as well as asserting the existence of a new predicate indicating the object has that action as an affordance (e.g., *affords*(*grasp*28(*obj*23))).[2] This predicate is added to the AN and can be used for learning.

[2]Although we only consider grasping actions, we assume other actions, such as pushing, also result in the "birth" of an object-affordance complex.

We make the strong assumption that the invariants of the domain map onto the input units of the associative network, which we assume in animals have evolved to this end and for the robot must be built in, are such as to ensure that when distinct low-level motor programs are indistinguishable at the higher level of abstraction, they will automatically be classified as instances of a single action.

VI. Using Learned Abstract Action Representations

We have described a process that results in learning abstracted action representations that should be close to the LDEC representations we have sketched for this domain. However, by abstracting the actions in this way there remains a number of open concerns we must address.

a) Using Learned Action Knowledge with New Objects: All new objects are initially associated with "new" actions. Our problem is to associate a previously unseen motor-program object pair with an existing high-level action or to mark it as a new action that must be learned at the high level.

b) Using Learned Action Knowledge for Execution: It will be necessary to convert our learned abstract actions to specific motor programs for execution. Keeping the list of the motor program-object pairs abstracted by each high-level action should address this issue. Since all abstracted pairs for a given action should be equivalent, we suggest selecting any one that matches the object bound in the high level plan.

c) Learning Exogenous Domain Properties: Although we have described a process for learning certain domain properties, the question remains as to how we will learn the exogenous properties given in Definition 3. For the present we simply assume the presence of *over* as an exogenous domain property that is computed by a lower level function.

VII. Conclusion

This paper has argued that object-action complexes (OACs) grounded from instantiated actions in robot control-space, can be used as an interface between the very different representation languages of robot control and AI planning. We have shown that OACS can be embodied in an Associative Net, and that they can be learned by a very simple machine-learning algorithm. Almost all of these claims are unproven but we offer them as defining a research program that we shall be pursuing in the coming years in order to combine existing robot platforms and existing planners based on LDEC and other situation/event calculi.

References

[1] B. Hommel, J. Müsseler, G. Aschersleben, and W. Prinz, "The theory of event coding (tec): A framework for perception and action planning." *Behavioral and Brain Sciences*, vol. 24, pp. 849–878, 2001.

[2] R. Murray, Z. Li, and S. Sastry, *A mathematical introduction to Robotic Manipulation*. CRC Press, 1994.

[3] N. Krüger, M. V. Hulle, and F. Wörgötter, "Ecovision: Challenges in early-cognitive vision," *International Journal of Computer Vision*, 2006.

[4] N. Pugeault, F. Wörgötter, , and N. Krüger, "Multi-modal scene reconstruction using perceptual grouping constraints," in *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision, New York City June 22, 2006 (in conjunction with IEEE CVPR 2006)*, 2006.

[5] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *ICCV*, 1998, pp. 1154–1160. [Online]. Available: citeseer.nj.nec.com/shi98motion.html

[6] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal segmentation and based on region merging," *IEEE transactions on Pattern Analysis and Machine Intelligence*, 1998.

[7] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and videos," *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2001.

[8] D. Aarno, J. Sommerfield, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, "Integration of elementary grasping actions and second order 3d feature relations for early reactive grasping," *submitted to 2006 IEEE-RAS International Conference on Humanoid Robots*, submitted.

[9] N. Krüger, M. Ackermann, and G. Sommer, "Accumulation of object representations utilizing interaction of robot action and perception," *Knowledge Based Systems*, vol. 15, pp. 111–118, 2002.

[10] M. Steedman, "Temporality," in *Handbook of Logic and Language*, J. van Benthem and A. ter Meulen, Eds. Amsterdam: North Holland/Elsevier, 1997, pp. 895–938.

[11] ——, "Plans, affordances, and combinatory grammar," *Linguistics and Philosophy*, vol. 25, pp. 723–753, 2002.

[12] J. McCarthy and P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence," *Machine Intelligence*, vol. 4, pp. 463–502, 1969.

[13] D. Harel, "Dynamic logic," in *Handbook of Philosophical Logic, volume II*, D. Gabbay and F. Guenthner, Eds. Dordrecht: Reidel, 1984, pp. 497–604.

[14] J.-Y. Girard, "Linear logic," *Theoretical Computer Science*, vol. 50, pp. 1–102, 1987.

[15] E. P. D. Pednault, "ADL: Exploring the middle ground between STRIPS and the situation calculus," in *Proceedings of the First International Conference on Principles of Knowledge Reprensentation and Reasoning (KR-89*, R. J. Brachman, H. J. Levesque, and R. Reiter, Eds. San Mateo, CA: Morgan Kaufmann Publishers, 1989, pp. 324–332.

[16] R. E. Fikes and N. J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, pp. 189–208, 1971.

[17] D. Willshaw, P. Buneman, and C. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, pp. 960–962, 1969.

[18] D. Willshaw, "Holography, association and induction," in *Parallel Models of Associative Memory*, G. Hinton and J. Anderson, Eds. Hillsdale, NJ: Erlbaum, 1981, pp. 83–104.

[19] K. Steinbuch, "Die lernmatrix," *Kybernetik*, vol. 1, pp. 36–45, 1961.

[20] J. Anderson, "A memory storage model utilizing spatial correlation functions," *Kybernetik*, vol. 5, pp. 113–119, 1968.

[21] F. T. Sommer and G. Palm, "Bidirectional retrieval from associative memory," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10. The MIT Press, 1998.

[22] T. Plate, "Holographic reduced representations: Convolution algebra for compositional distributed representations," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence, San Mateo CA*. San Francisco, CA: Morgan Kaufmann, 1991, pp. 30–35.

Paco-Plus Design Documentation for Integration of Robot Control and AI Planning

UEDIN: Christopher Geib, Ronald Petrick, Kira Mourão, Nicolas Pugeault, Mark Steedman
UL: Pascal Haazebroek
AAU: Norbert Krueger, Dirk Kraft
BCCN: Florentin Wörgötter

15 December 2006

1 Overview

The following document was started as a way to capture the conclusions of the discussions between the authors, for specific proposals about how to interface the high-level "AI Planning layer" and the "Robot control layer". We believe that it most properly should be seen as an evolving design document specifying the conclusions that we came to and the implications for the interface between these two modules.

1.1 Document history

Rev. 1: Results of October 9th–11th meeting in Edinburgh
Rev. 2: Extended and modified as a result of November 23rd–24th meeting in Goettingen.

1.2 Objective

Our overall objectives are to produce a principled interface for the interaction of an AI level planning component with a lower level robotic control component with three requirements:

1. use of Object Action Complexes (OACs) to define and constrain the interactions
2. enable the discovery of objects, actions, and properties by the robot component.
3. enable the learning of AI planning level representations for the objects, actions, and properties on the basis of robotic level observations

1.3 Basic Assumptions

We have been assuming a two layered architecture wherein the robot controller is responsible for operating and controlling "continuous" or real valued sensors and actuators and reporting up to the AI planning level discrete (possibly thresholded) state information. When possible, the AI planner will submit sequences of actions to be executed to the robot controller in order to achieve high-level goals.

This has the basic impact of requiring information to flow both from the robot controller to the AI Planning system and from the planning system back to the robot controller. In Figure 1 we see the flow of information between these two components. The purpose of much of this document is to define the nature of that communication and changes in control that accompany them.

2 Issues for the Interface

During our discussions we identified a number of issues that any interface between the AI planner and Robot controller must address. In this section we identify those issues and sketch any agreed upon solutions.

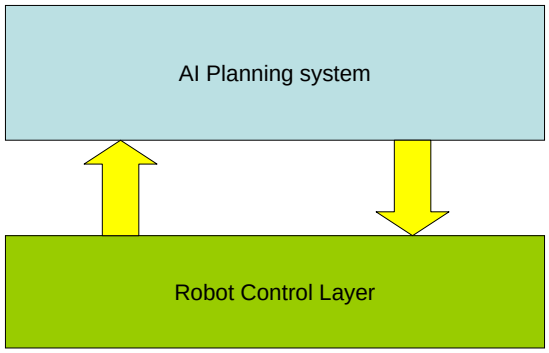


Figure 1: Very high-level system architecture

2.1 System Level control

As with any multi-layer control system where there are multiple controllers that can suggest actions for execution there is an issue about adjudicating between the suggested actions. Effectively the question is: when is each control algorithm in charge of determining the next action for the system as a whole to execute.

As will be clear when we discuss a proposed example of the system's execution, one of the primary objectives of the system is to acquire more information about the state of the world, the objects in it and the actions that can be performed on them. This has the following implications. Most of the actions called for by the lower level robot controller are reflexive and exploratory in nature. Most of the actions called for by the AI planning level are goal directed and require richer models that must be learned from the lower level. As a result the interface is designed such that while the AI planner has sufficient information to suggest plans to achieve goals it is allowed to hand actions to the lower level for execution. When the goals of the AI system have been achieved or the system doesn't have enough information about the state of the world the robot controller's exploratory behaviours are invoked. We will see this in much more detail when we discuss an example of the system working.

2.2 Exogenous Domain Predicates

We define *exogenous domain predicates* as domain information required by the AI level about the state of the world that cannot be determined by AI system reasoning. Such predicates represent critical state information needed for planning, for instance for reasoning about the relative positions of objects. An example of this would be the predicate "over" as in "over(object23, object98)". It has been generally agreed that the lower level robot controller is capable of providing this information. Such predicates may fall into two distinct classes.

1. **Computed:** A simple definition of the predicate "over" would simply be the projection upwards of the bounding box for the object. This is easily derivable from the objects visual features, however may not cover more detailed action needs.

2. **Learned:** Defining "over the box for the purpose of dropping something into the box" may be a predicate that requires a more active learning process. This may include experimentation.

This means that in the AI level representation we will have more "prepositions" including prepositions that are task specific: for example, "over-for-grasp23" and "over-for-drop-into". This is consistent with linguistic evidence about interpreting prepositions within context.

2.3 Monitoring of Lower Level Action Execution

As part of such an architecture it is important for the AI level to be able to monitor lower level action execution and, if the plan is failing to meet its goals, to be able to stop the execution. To meet this requirement we place the following requirement on the AI planner: as the world state is updated at the higher level, the AI level is required to monitor these conditions and issue a "halt action" command if the action is not proceeding according to expectations.

2.4 Agreement not to use Specific Locations

Testing equality of locations is very hard to do at the robot level and not something we should ever expect will be available from the low level. This strengthens the idea that almost all work done at the AI level for planning should be done relative to specific object identifiers rather than positions in an N dimensional space.

2.5 Preparatory Actions

There is a necessity for including actions in the AI level plan for "grasp setup actions". Robotic path planning is very complex and we need to make a nod in the direction of acknowledging that. At this point we believe this will be little (or possibly nothing) more than a stub action. But recognizing the problem is significant. We note that learning such preparatory actions may present a much more significant challenge for the AI level, since it is not as obvious what if any perceptual change in the world would be relevant for the learning of this action. But this may be consistent with the child development evidence.

2.6 Division of Representation

Different representations and the corresponding operations that generate/manage these representations need to be separated into two levels: the lower vision/robot level and the higher AI planning level. We believe that the lower level should manage information concerning visual features, relations, and untested reflexes. Class information and newly born objects would be generated at the lower-level through the operation of the vision system.

The information provided by the lower level would be used to induce the high-level representations. The high level would operate with unique identifiers (possibly generated at the higher level on the request of the lower level) that would be used to index lower-level objects, classes, and motor programs/reflexes. The high level would have no direct access to local visual features and relations, and would avoid reasoning about continuous values. Plans generated at the high level should therefore contain appropriate "links" to the required low-level entities using the unique identifiers.

2.7 Example of Pulling Objects into the Workspace

There are a number of interesting problems that clearly motivate the need for integrating AI level planning and lower level robot planning. For example, consider the case of pulling an object that is currently unreachable, into the working space. This is a textbook example of where AI level planning can be used to make objects more accessible. The robot system is unable to reach the object initially, however it is possible

to invoke the AI planner to build a plan to pull the object into the work space to allow for a better grasping action.

The significant caveat is that we need to be sure to be able to define more exogenous domain predicates that will allow us to do the planning in a discrete space, rather than attempting to do "geometric reasoning" at the AI planning level. For example, the addition of "not in reach" as a perceptual primitive would be needed for this example, and could be provided by the robot level.

3 Demo 1 Example

As a design document we will not include here a detailed discussion of the component technologies that go into this system. Instead we suggest reading [1, 2, 3] for details of how the AI planning system and robot control system work. In this section we will provide a very simple sketch of an imagined interaction between these components to achieve the end of learning about the world, culminating in clearing a table of a number of coffee mugs.

We imagine our AI planning system and robot arm equipped with a vision system confronted with a table that has on it a number of objects:

1. three identical cups,
2. three identical plates, and
3. a box the cups and plates should be put into.

We do not assume the system has knowledge of the cups or plates or their affordances, however, we do assume that the system knows about the box and its affordances, and that the AI system has the goal of putting all objects on the table into the box.

We envision the following scenario:

1. Since the AI planner does not know about the objects and their affordances, it invokes the robot level reflexive exploration behaviour.
2. The robot controller experiments with choosing pairs of coplanar points and attempting to grasp them to see if they define an object. This task repeats until the robot manages to effectively grasp an object. For the first case we assume it is one of the cups.
3. By moving the object and subtracting out its gripper the robot forms a model of the cup, and can then place it back on the table.
4. As a result of this exploration the robot controller reports to the AI planner the "birth of an object". As part of this it reports: a UID for the class of objects, a UID for each instance of the object in the visual field (in this case one for each of the three cups), and a UID for the associated affordance of the object (namely the single successful grasping instance just used).
5. On the basis of this new information the AI planning system now knows that its goal of having a clear table is not satisfied and it attempts to achieve it. Given the new objects and their grasping affordances the planner is able to build a plan for putting the three cups in the box and sends that plan to the robot controller for execution.
6. The robot controller executes the plan, sending reports of state changes back to the AI system (to enable it to monitor the plan for progress). We assume the plan works perfectly, placing all of the cups in the box.

7. The AI system now believes that its goal of having a clear table has been met (since it knows nothing about the plates) and again invokes the robot controller's reflex driven exploration behaviour looping back to the first step.

In discussing this type of demo we have made the commitment that the AI planner would not ask the low-level system to execute any action that it hasn't done before (at least as far as identified classes of objects are concerned). However it is worth noting that the large amount of information we are assuming is already present in the AI planning system. For example, in this document (though not in other places [1]) we have glossed over the question of how the system actually learns the high-level representations. Such learning requires previous knowledge of exogenous domain predicates, classes of actions and much other knowledge that we are assuming we have access to for this first demo.

4 List of Robot Primitives

The following is a proposed list of the primitive robot reflexes that are initially available in the system. Note that the names given in this list may not correspond to the names given to these primitives in the robot system, but instead capture the major classes of reflex behaviours available from the robot system.

- **moveGripper:** This corresponds to the general movement of the robots end effector to a specific location in the domain. It will underlie two of the AI level actions, *moveEmptyGripper* and *moveFull-Gripper*.
- **grasp1, grasp2, . . . , graspX:** This set forms the low-level reflex for grasping based on a pair of features within the visual field. This will be the foundation for the AI system's ability to learn generalized grasping.
- **ungrasp1, ungrasp2:** This set are the reflexes that are used to release an object that has been grasped.

These may of course be extended with other low-level primitives.

5 Definition of Domain Terms

To be more specific in our discussions, we define Object, ObjectType, Pose, Location, and Affordance.

Definition 1 Proposed terms that will be used in following definitions

1. **Object:** a unique object identifier. That is a unique identifier of an object instance in the world model that is shared between the AI planner and the Robot/vision system.
2. **ObjectType:** a unique identifier denoting the type or class of a particular object instance. Each Object has a single ObjectType.
3. **Pose:** an identifier that denotes the current orientation of an object. The set of all available poses is an enumerated type, e.g., $\{1, 2, 3, \ldots, \textit{upright}\}$
4. **Location:** a unique location identifier. That is a unique identifier used to refer to a location in the robot/vision systems world model.
5. **Affordance:** a unique affordance identifier. That is a unique identifier of an affordance that can be re-executed by the Robot/vision system.

6 Functions that map objects to locations

We are already aware of a number of specialized functions that will be required for the AI system to map from Objects to Locations for use by the robot system. These functions will have to be learned by the system and this must take place both at the robot level (training a perceptual primitive to send a message to the AI system when they are true) and at the AI level (learning when they are important for describing the preconditions or effects of actions).

- **into:** This function maps from an object that is concave to a point inside the object. This function is assumed to produce a location such that if an object in the gripper is released from the location, the object can be considered "carefully placed into the object". Note that if the object is not known to be concave the function throws an exception?
- **onto:** This function maps from an object to an arbitrary point that is wholly on the object. This function is assumed to produce a location such that if an object in the gripper is released from the location, the object can be considered "carefully placed on the object".
- **tograsp1, tograsp2, . . . , tograspX:** For each grasp affordance that is learned we introduce a function that will map from an object that will be grasped using that affordance to a location that sets up the grasp.

7 Messages from the Robot: Predicates that Define the World State at the AI Planning Level

In order for the AI Planning system to maintain a world model for planning the robot/vision system must send messages to the AI planning system about any significant perceptual changes to the world. Such messages should be "pushed" to the AI system to allow asynchronous update of the world model even possibly during the execution of an action requested by the AI planning system.

The following table specifies the current complete set of the messages the robot/vision system must be able to send for this purpose. Note that we distinguish the messages that contain meta-information (the introduction of new objects or affordances to the world model) from the messages that update the state of a predicate in the world model.

Predicate Def	**Example Use**	**Descriptive Note**
$in(Object_{container}, Object_{contained})$	$in(box1, obj1)$	Captures locations of objects
$on(Object_{supporting}, Object_{supported})$	$on(table1, obj1)$	Captures locations of objects
$ingripper(Object)$	$ingripper(obj1)$	Captures locations of objects
$pose(Object, Pose)$	$pose(obj1, upright)$	Captures the pose of the object
$gripperempty$	$gripperempty$	Captures the state of the gripper when empty
$gripperat(Location)$	$gripperat(tograsp23(obj1))$	Captures the location of the gripper
Message Def	**Example Use**	**Descriptive Note**
$newobj(Object)$	$newobj(obj1)$	Introduces a new object.
$newaff(ObjectType, Affordance)$	$newaff(objtype1, grasp28)$	Introduces a new affordance for an object.

Figure 2: Messages sent by the Robot to the AI Planning Level

8 Messages from the AI Planning Level: Action Requests from the Planner

The AI planner requests actions for execution. Each such action has specific action effects that should be visible in the world model after a successful execution. The following table provides the specification for the action, an example of its use, and the expected change the robot level will report back to the AI level if the action is successful.

Action Def	Example Use	Successful Execution Result
moveEmptyGripper(*Location*)	*moveEmptyGripper*(*tograsp23*(*obj*1))	*gripperat*(*tograsp23*(*obj*1))
graspI(*Object*)	*grasp23*(*obj*1)	*ingripper*(*obj*1)
moveFullGripper(*Location*)	*moveFullGripper*(*into*(*box*1))	*in*(*box*1, *obj*1)
drop(*Object*)	*drop*(*obj*1)	*gripperempty*
beginexploration	*beginexploration*	*N/A*

Figure 3: Possible AI Planning Level Action Requests

We note that *beginexploration* is special in the sense that it is a meta-level operation that initiates a process at the robot level, without any direct execution results.

9 Tasks for AI Planning Level

The following are the major tasks the AI Planning Level team needs to complete.

1. **Encode Planning Domain as Specified:** This includes encoding the planning problem as specified above with the given domain primitives and actions. This will result in a planner capable of building plans for clearing the table of discovered objects. Note that for our example domain all that will be missing for the system is the specific action/affordance and object instance information that is needed to produce the plan. This will be provided by the robot controller.

2. **Build High-Level Planning System Architecture:** This includes creating an infrastructure that builds plans, submits plans for execution, verifies the resulting state is consistent with its expectations and if so submits the next action in the plan.

3. **Build Truth Maintenance System for High-Level World Model:** Since the robot system is responsible for the assertion of some facts that have multiple effects on the world model at the AI level, we are responsible for building a small system to effectively update the AI world model.

4. **OPTIONAL: Interface to GraspIt Software:** If the interface to the GraspIt software is identical to the interface that will be needed for the Robot then we will make any small changes necessary for the testing of the AI Planning system within this simulation environment. To the degree that the interface would be significantly different and require significant extra code we are not sure that we see the value of the effort in such an integration.

10 Tasks for Robot Level

The following are the major tasks the Robot/Vision team needs to complete. [this needs real work]

1. **Smoothing of percepts to remove irrelevant discontinuities:**

2. **Construction of perceptual functions**

3. **Wrapper for execution of requested tasks**

4. **Others?**

11 Known "Open" Issues that will Require Addressing Later

During our discussions we identified a number of issues that we will need to return to address later. These include:

1. **Dropping and "onto" vs push and "into"**

2. **Pushing and pulling actions**

3. **Objects "disappearing" from the perceptual system**

References

[1] C. W. Geib, K. Mourão, R. Petrick, N. Pugeault, M. Steedman, N. Krueger, and F. Wörgötter. Object action complexes as an interface for planning and robot control. In *Proceedings of the HUMANOIDS-06 Workshop: Toward Cognitive Humanoid Robots*, 2006.

[2] R. P. A. Petrick and F. Bacchus. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, pages 212–221. AAAI Press, 2002.

[3] R. P. A. Petrick and F. Bacchus. Extending the knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS-04)*, pages 2–11. AAAI Press, 2004.

Robotics Group
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Technical Report no. 2007 – 3

Perceptual Operations and Relations between 2D or 3D Visual Entities

Sinan Kalkan, Nicolas Pugeault, Mogens Christiansen, Norbert Krüger

January 22, 2007

Title	Perceptual Operations and Relations between 2D or 3D Visual Entities
Author(s)	Sinan Kalkan, Nicolas Pugeault, Mogens Christiansen, Norbert Krüger
Publication History	

Abstract

In this paper, we present a set of perceptual relations, namely, co-colority, co-planarity, collinearity and symmetry that are defined between multi-modal visual features that we call *primitives*.

1 Introduction

According to Marr's paradigm [29], vision involves extraction of meaningful representations from input images, starting at the pixel level and building up its interpretation more or less in the following order: local filters, extraction of important features, the $2\frac{1}{2}$-D sketch and the 3-D sketch.
There is psychophysical evidence and evidence from the statistical properties of natural images that the human visual system utilizes a set of visual-entity-combining processes, called *perceptual organization* in the literature, for forming bigger, sparser and more complete interpretations of the scene (see, *e.g.*, [18, 19, 35]). Such processes include (i) extraction of the boundary of the objects in the image from the set of unconnected edge pixels or features [3, 8, 10, 21, 27, 31, 39] utilizing Gestalt laws of grouping, and (ii) interpolation and extrapolation of unconnected sparse 3D entities for forming more complete 3D surfaces (see, *e.g.*, [13]) utilizing the relations between the 3D entities. Gestalt principles include collinearity, proximity, common fate and similarity whereas inference of 3D surfaces from a set of 3D entities include relations like coplanarity, collinearity, co-colority etc. These are essentially second order and higher order relations of local features.
In [26], we have introduced a specific form of a local descriptor that we call a 'multi–modal primitive' (see section 2) and which can be seen as a functional abstraction of a hypercolumn (see [24]). We distinguish between 2D primitives describing local image information and 3D primitives covering local 3D scene information in a condensed symbolic way.
These primitives serve as a basis for an early cognitive vision system [23, 26, 33] in which operations and relations on these primitives realizing perceptual grouping principles are used in different contexts (see [26] for applications). We have utilized these relations for different problems including stereo [34], RBM [32], estimation of initial grasping reflexes from stereo [5], estimation of depth at homogeneous image structures [16], and analysis of second-order relations between 3D features [17].
In this paper, we present the set of 2D and 3D relations defined upon the primitives. These relations include collinearity, cocolority, coplanarity and symmetry. Of these relations, collinearity, cocolority and symmetry are defined for 2D as well as 3D primitives whereas by definition, coplanarity is meaningful only for 3D primitives. Table 1 summarizes the relations and on which dimension they are defined.

Relation	2D	3D
co-planarity	$\times$	$\surd$
co-colority	$\surd$	$\surd$
collinearity	$\surd$	$\surd$
symmetry	$\surd$	$\surd$

Table 1: The relations and in which dimension they are defined.

This paper does not focus on any specific application domain but provides a technically detailed definition of these relations that are usually not described in such detail in publications making use of them.
The paper is organized as follows: In section 2, we briefly introduce our visual features, namely primitives. In section 3, we describe our definitions of perceptual relations between the visual primitives. In section 5, we conclude the paper.

2 Primitives

Numerous feature detectors exist in the literature (see [30] for a review). Each feature based approach can be divided into an interest point detector (e.g. [14, 4]) and a descriptor describing a local patch of the image at this location, that can be based on histograms (e.g. [6, 30]), spatial frequency [20], local derivatives [15, 11, 1] steerable filters [12], or invariant moments ([28]). In [30] these different descriptors have been compared, showing a best performance for SIFT-like descriptors.
The primitives we will be using in this work are local, multi–modal edge descriptors that were introduced in [25]. In contrast to the above mentioned features these primitives focus on giving a semantically and geometrically meaningful description of the local image patch. The importance of such a semantic grounding of features for a general purpose vision front–end, and the relevance of edge–like structures for this purposes were discussed in [9].
The primitives are extracted sparsely at locations in the image that are the most likely to contain edges. This likelihood is computed using the intrinsic dimensionality measure proposed in [22]. The sparseness is assured using a classical winner take all operation, insuring that the generative patches of the primitives do not overlap (for details, see [26]). Each of the primitive encodes the image information contained by a local image patch. Multi–modal information is gathered from this image patch, including the position $\boldsymbol{m}$ of the centre of the patch, the orientation θ of the edge, the phase ω of the signal at this point, the colour $\boldsymbol{c}$ sampled over the image patch on both sides of the edge and the local optical flow $\boldsymbol{f}$. Consequently a local image patch is described by the following multi–modal vector:

$$\boldsymbol{\pi} = (\boldsymbol{m}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, \rho)^T, \tag{1}$$

that we will name *2D primitive* in the following.
Note that these primitives are of lower dimensionality than, e.g., SIFT (10 vs. 128) and therefore suffer of a lesser distinctiveness. Nonetheless, as shown in [34] that they are distinctive enough for a reliable stereo matching if the epipolar geometry of the cameras is known. Furthermore, their semantic in terms of geometric and appearance based information allow for a good description of the scene content. It has been previously argued in [9] that edge pixels contain all important information in an image. As a consequence, the ensemble of all primitives extracted from an image describe the shapes present in this image.
Advantageously, the rich information carried by the 2D–primitives can be reconstructed in 3D, providing a more complete scene representation. Having geometrical meaning for the primitive allows to describe the relation between proximate primitives in terms of perceptual grouping.
In a stereo scenario 3D primitives can be computed from the correspondences of 2D primitives (see figure 1 and [34]):

$$\boldsymbol{\Pi} = (\boldsymbol{M}, \Theta, \boldsymbol{\Omega}, \boldsymbol{C})^T, \tag{2}$$

such that we have a projection relation:

$$\mathcal{P} : \boldsymbol{\Pi} \rightarrow \boldsymbol{\pi}\,. \tag{3}$$

3 Relations

In this section, we present collinearity, cocolority, coplanarity and symmetry relations that are defined on our visual features.

3.1 Collinearity in 2D and 3D

As the primitives are local contour descriptors, scene contours are expected to be represented by strings of primitives that are locally close to collinear. In the following, we will explain methods for grouping 2D and 3D primitives into contours.

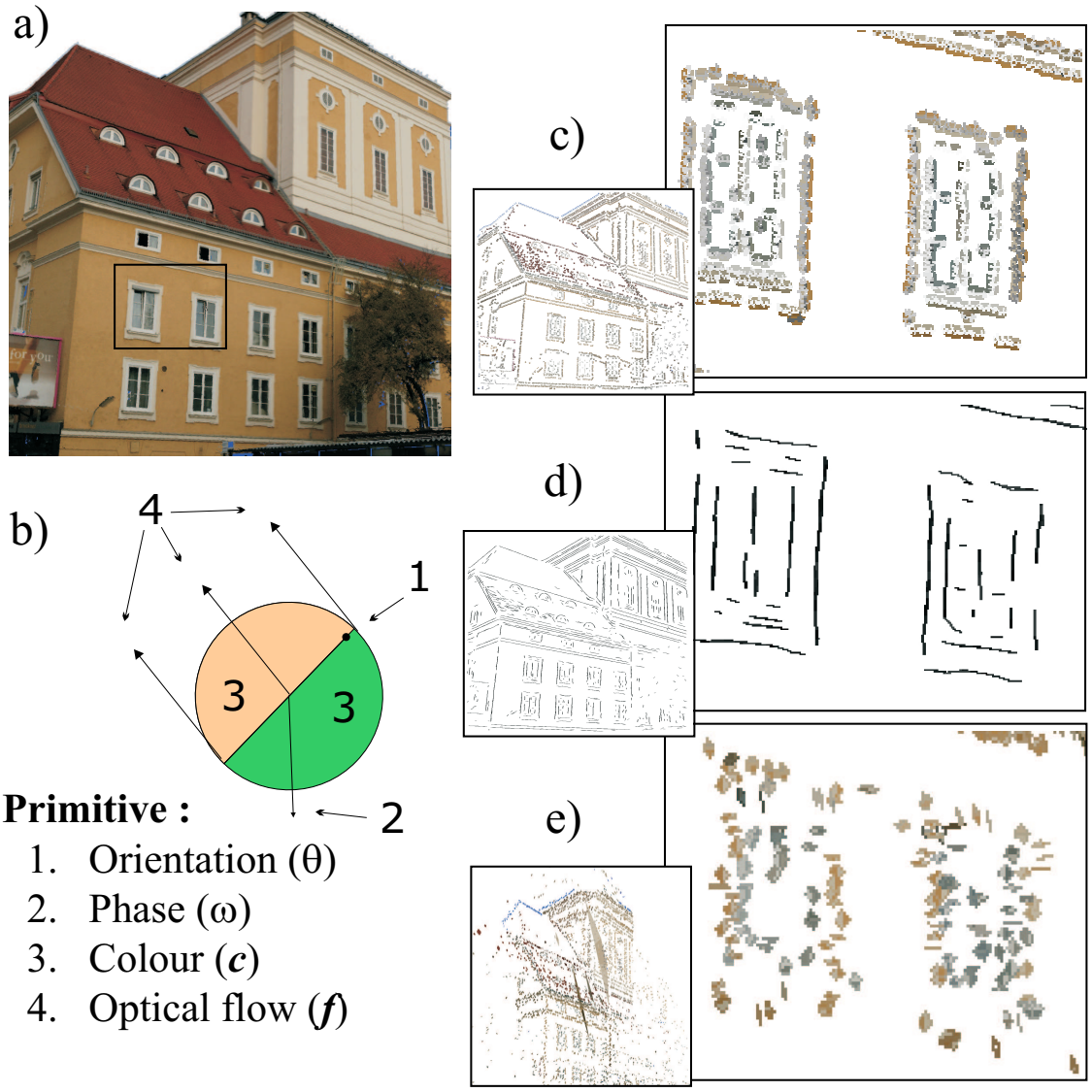


Figure 1: Illustration of the primitive extraction process from a video sequence. The 2D–primitives extracted from the input image (a) (see section 2), and finally the 3D–primitives reconstructed from the stereo–matches as described as described in [34]. **(a)** An example input image. **(b)** A graphic description of the 2D–primitives. **(c)** A magnification of the image representation. **(d)** Perceptual grouping of the primitives as described in [34]. **(e)** The reconstructed 3D entities. Note that the structure reconstructed is quite far from the cameras, leading to a certain imprecision in the reconstruction of the 3D–primitives. A simple scheme addressing this problem is described in [34].

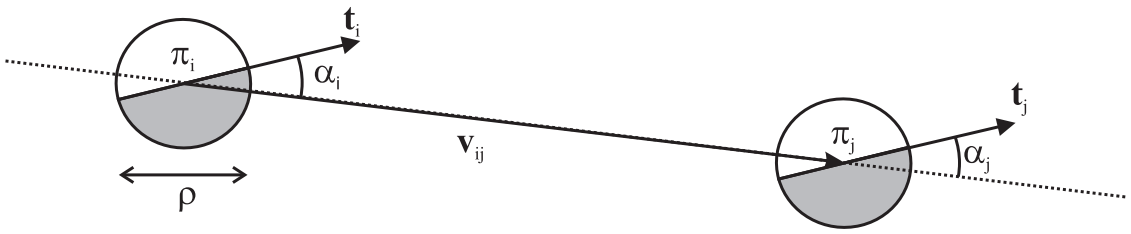


Figure 2: Illustration of the values used for the collinearity computation. If we consider two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, then the vector between the centres of these two primitives is written $\boldsymbol{v}_{ij}$, and the orientations of the two primitives are designated by the vectors t_i and t_j, respectively. The angle formed by $\boldsymbol{v}_{ij}$ and t_i is written α_i, and between $\boldsymbol{v}_{ij}$ and t_j is written α_j. ρ is the radius of the image patch used to generate the primitive.

3.1.1 Collinearity in 2D

In the following, $c(l_{i,j})$ refers to the likelihood for two primitives π_i and π_j to be *linked*: i.e. grouped to describe the same contour.
Position and orientation of primitives are intrinsically related. As primitives represent local edge estimators, their positions are points along the edge, and their orientation can be seen as a tangent at such a point. The estimated likelihood of the contour described by those tangents is based upon the assumption that simpler curves are more likely to describe the scene structures, and highly jagged contours are more likely to be manifestations of erroneous and noisy data.
Therefore, for a pair of primitives π_i and π_j in image $\boldsymbol{\mathcal{I}}$, we can formulate the likelihood for these primitives to describe the same contour as a combination of three basic constraints on their relative position and orientation — see [34].

Proximity ($c_p[l_{i,j}]$): A contour is more likely if it is described by a dense population of primitives. Large holes in the primitive description of the contour is an indication that there are two contours which are collinear yet different. The proximity constraint is defined by the following equation:

$$c_p[l_{i,j}] = 1 - e^{-\max\left(1-\frac{||\mathbf{v_{i,j}}||}{\rho\tau},0\right)}, \tag{4}$$

where ρ stands for the size of the receptive field of the primitives in pixels; $\rho\tau$ is the size of the neighbourhood considered in pixels; and, $\|\mathbf{v_{i,j}}\|$ is the distance in pixels separating the centres of the two primitives.

Collinearity ($c_{co}[l_{i,j}]$): A contour is more likely to be linear, or to form a shallow curve rather than a sharp one. A sharp curve might be an indication of two intersecting or occluding contours.

$$c_{co}[l_{i,j}] = 1 - \left|\sin\left(\frac{|\alpha_i| + |\alpha_j|}{2}\right)\right|, \tag{5}$$

where α_i and α_j are the angles between the line joining the two primitives centres and the orientation of, respectively, π_i and π_j.

Co–circularity ($c_{ci}[l_{i,j}]$): A contour is more likely to have a continuous, or smoothly changing curvature, rather than a varying one. An unstable curvature is an indicator of a noisy, erroneous or under–sampled contour, all of which are unreliable.

$$c_{ci}[l_{i,j}] = 1 - \left|\sin\left(\frac{\alpha_i + \alpha_j}{2}\right)\right|, \tag{6}$$

Geometric Constraint ($\mathbf{G}_{i,j}$): The combination of those three criteria provided above forms the following *geometric* affinity measure:

$$\mathbf{G}_{i,j} = \sqrt[3]{c_e[l_{i,j}] \cdot c_{co}[l_{i,j}] \cdot c_{ci}[l_{i,j}]}, \tag{7}$$

where $\mathbf{G}_{i,j}$ is the geometric affinity between two primitives π_i and π_j. This affinity represents the likelihood that two primitives π_i and π_i are part of an actual contour of the scene.

Multi–modal Constraint ($\mathbf{M}_{i,j}$): The geometric constraint offers a suitable estimation of the likelihood of the curve described by the pair of primitives. Other modalities of the primitives allow inferring more about the qualities of the physical contour they represent. The colour, phase and optical flow of the primitives further define the properties of the contour, and thus consistency constraints can also be enforced over those modalities. Effectively, the less difference there is between the modalities of two primitives, the more likely that they are expressions of the same contour. In [7], it is already proposed that the intensity can be used as a cue for perceptual grouping; our definition goes beyond this proposal by using a combination of the phase, colour and optical flow modalities of the primitives to decide if they describe the same contour:

$$\mathbf{M}_{i,j} = w_\omega c_\omega[l_{i,j}] + w_{\boldsymbol{c}} c_{\boldsymbol{c}}[l_{i,j}] + w_{\boldsymbol{f}} c_{\boldsymbol{f}}[l_{i,j}], \tag{8}$$

where c_ω is the phase criterion, $c_{\boldsymbol{c}}$ the colour criterion and $c_{\boldsymbol{f}}$ the optical flow criterion. Each of the three w_ω, $w_{\boldsymbol{c}}$ and $w_{\boldsymbol{f}}$ is the relative scaling for each modality, with $w_\omega + w_{\boldsymbol{c}} + w_{\boldsymbol{f}} = 1$.

Primitive Affinity ($\boldsymbol{A}_{i,j}$): The overall affinity between all primitives in an image is formalised as a matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{i,j}$ holds the affinity between the primitives π_i and π_j. We define this affinity from equations 7 and 8, such that (1) two primitives complying poorly with the good continuation rule have an affinity close to zero; and (2) two primitives complying with the good continuation rule yet strongly dissimilar will have only an average affinity. The affinity is formalised as follows:

$$c(l_{i,j}) = \boldsymbol{A}_{i,j} = \sqrt{\mathbf{G}\left(\alpha \mathbf{G}_{i,j} + (1-\alpha)\mathbf{M}_{i,j}\right)}, \tag{9}$$

where α is the weighting of geometric and multi–modal (*i.e.* phase, colour and optical flow) information in the affinity. A setting of $\alpha = 1$ implies that only geometric information (proximity, collinearity and co-circularity) is used, while $\alpha = 0$ means that geometric and multi–modal information are evenly mixed.

3.1.2 Collinearity in 3D

Collinearity in 3D is more difficult to define. Due to the inaccuracy in stereo–reconstruction of 3D position and orientation, it is impossible to apply strong alignment constraints such as the ones we applied in the 2D case. Consequently we will define 3D collinearity as follows:

Definition 1 *Two 3D–primitives Π_i and Π_j are said collinear if the 2D–primitives π_i^x and π_j^x they project onto the camera plane x (defined by a projection relation $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$) are all collinear (according to the definition of 2D–primitive collinearity presented above).*

and therefore in the standard case where we have two stereo cameras labelled l and r we have the following relation:

$$c(L_{i,j}) = c(l^l_{i,j}) \cdot c(l^r_{i,j}). \tag{10}$$

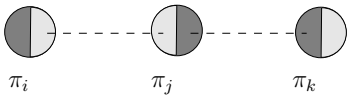


Figure 3: Co–colority of three 2D primitives π_i, π_j and π_k. In this case, π_i and π_j are cocolor, so are π_i and π_k; however, π_j and π_k are not cocolor.

3.2 Cocolority in 2D and 3D

Two spatial primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$ are co–color iff their parts that face each other have the same color. In the same way as collinearity, co–colority of two spatial primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$ is computed using their 2D projections $\mathcal{P}\Pi_i = \pi_i$ and π_j. We define the co–colority of two 2D primitives π_i and π_j as:

$$coc(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j),$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the RGB representation of the colors of the parts of the primitives π_i and π_j that face each other; and, $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is Euclidean distance between RGB values of the colors $\mathbf{c}_i$ and $\mathbf{c}_j$. In Fig. 3, a pair of co–color and not co–color primitives are shown.
Euclidean color distance d_c is a simple one compared to color distance metrics developed by different institutes like International Commission on Illumination (CIE). Such metrics are developed to match our perception of colour and are computationally expensive (see, *e.g.*, [38]). For our purposes, Euclidean distance between RGB values is sufficient and can be replaced by a more complicated distance metric, if desired.
3D co-colority is defined as follows:

Definition 2 *Two 3D–primitives Π_i and Π_j are said cocolor if the 2D–primitives π_i^x and π_j^x they project onto the camera plane x (defined by a projection relation $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$) are co-color (according to the definition of 2D–primitive cocolority presented above).*

3.3 Coplanarity

According to [37],

> a set of points in space is coplanar if the points all lie in a geometric plane. For example, three points are always coplanar; but four points in space are usually not coplanar.

Although the definitions are more or less the same, there are different ways to *check* the coplanarity of a set of points [36, 37]. For a set of n points $\mathbf{x}_1 ... \mathbf{x}_n$ where $\mathbf{x}_i = (x_i, y_i, z_i)$, the following methods can be adopted:

- For $n = 4$, $\mathbf{x}_1 ... \mathbf{x}_n$ are coplanar
  - iff the volume of the tetrahedron defined by them is 0 [36], *i.e.*,

$$\begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0. \tag{11}$$

  - iff the pair of lines determined by the four points are not skew [36]:

$$(\mathbf{x}_3 - \mathbf{x}_1).[(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_4 - \mathbf{x}_3)] = 0. \tag{12}$$

– iff $\mathbf{x}_4$ is on the plane defined by $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$:

$$d(\mathbf{x}_4, P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)) = 0, \tag{13}$$

where $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ is the plane defined by $P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, and $d(\mathbf{x}, \mathbf{p})$ is the distance between point $\mathbf{x}$ and plane $\mathbf{p}$.

- For $n > 4$, $\mathbf{x}_1 ... \mathbf{x}_n$ are coplanar iff point-plane distances of $\mathbf{x}_4 .. \mathbf{x}_n$ to the plane defined by $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ are all zero:

$$\sum_{i=4}^{n} d(\mathbf{x}_i, P(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)) = 0. \tag{14}$$

3.3.1 Coplanarity of bounded planes

A bounded plane $\mathbf{p}^b$ is part of the plane $\mathbf{p}$ with a certain size $\mathbf{s}$ and position $\mathbf{x}$. In other words, $\mathbf{p}^b$ is equivalent to $(\mathbf{n}, \mathbf{x}, \mathbf{s})$ where $\mathbf{n}, \mathbf{x}, \mathbf{s}$ are respectively the normal (*i.e.*, orientation), position (*i.e.*, center) and the size of the bounded plane.
As suggested in [17], two bounded planes $\mathbf{p}_1^b, \mathbf{p}_2^b$ are coplanar if:

$$(\alpha(\mathbf{n}_1, \mathbf{n}_2) < T_\alpha) \ \wedge \ (\frac{d(\mathbf{x}_1, \mathbf{p}_2^b)}{d(\mathbf{x}_1,\ \mathbf{x}_2)} < T_d), \tag{15}$$

where $\alpha(\mathbf{n}_1, \mathbf{n}_2)$ is the angle between the two orientations vectors $\mathbf{n}_1$ and $\mathbf{n}_1$, and T_α and T_d are the thresholds.

3.3.2 Coplanarity of 3D primitives

Two spatial primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$ are co–planar iff their orientation vectors lie on the same plane, i.e.:

$$cop(\mathbf{\Pi}_i, \mathbf{\Pi}_j) = 1 - |\mathbf{proj}_{t_j \times v_{ij}}(t_i \times v_{ij})|, \tag{16}$$

where $\mathbf{v}_{ij}$ is defined as the vector $(\boldsymbol{M}_i - \boldsymbol{M}_j)$; t_i and t_j denote the vectors defined by the 3D orientations Θ_i and Θ_j, respectively; and $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$ is defined as:

$$\mathbf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2} \mathbf{u}. \tag{17}$$

The co–planarity relation is illustrated in Fig. 4.

3.4 Symmetry in 2D and 3D

Two primitives are symmetric if they are located on two contours which are reflections of each other (see figure 5(a)). This reflective symmetry between two primitives can be measured by utilizing the angles between the orientations of the primitives and the line that joins the centers of the primitives.
Let v_{ij} denote the line joining the centers of the primitives, π_i and π_j, and also ϕ_{ij} and ϕ_{ji} be the angles between v_{ij} and the lines defined by the orientations of π_i and π_j, respectively (see figure 5). Then, two 2D primitives π_i and π_j can be considered symmetric, if $\phi_{ij} = \phi_{ji}$ with a symmetry axis a_{ij} defined as follows:

$$a_{ij} = \begin{cases} L(c_{ij}; \theta_i) & \text{if } \theta_i = \theta_j, \\ L(c_{ij}; \alpha_{ij}), & \text{otherwise,} \end{cases} \tag{18}$$

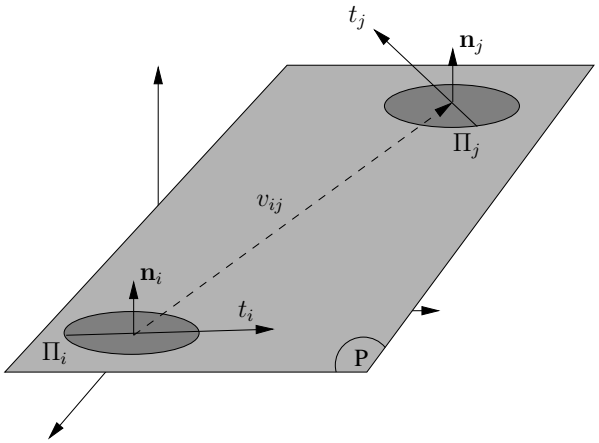


Figure 4: Co–planarity of two 3D primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$. t_i and t_j denote the vectors defined by the 3D orientations Θ_i and Θ_j, respectively.

where $L(x;\theta)$ is a line that goes through a point x with orientation θ; $\mathbf{int}(l_k, l_m)$ is the intersection point of two lines denoted by l_k and l_m; c_{ij} is defined as the mid-point of v_{ij} (*i.e.*, $(\boldsymbol{m}_i + \boldsymbol{m}_j)/2$); and, α_{ij} is the angle of the line that joins the points c_{ij} and $\mathbf{int}(L(\boldsymbol{m}_i;\theta_i), L(\boldsymbol{m}_j;\theta_j))$.
The symmetry axis a_{ij} is undefined if the primitive orientations θ_i and θ_j, and v_{ij} are all parallel, which is the case when both primitives are located on the same linear segment of a contour. This is the case for π_j and π_k in figure 5(b) and 5(c). If the symmetry axis a_{ij} is undefined, a primitive pair should not be regarded as symmetric, but collinear.
Figure 5 illustrates a few symmetric and non-symmetric primitives. In figure 5(b) and 5(c), as the primitives π_j and π_k are on the same contour, a_{ij} is parallel with the primitive orientations θ_j, θ_k and v_{jk}.
Taking collinearity into account, symmetry between two primitives π_i and π_j is defined as follows:

$$sym(\pi_i, \pi_j) = \begin{cases} 0 & \text{if } c_{co}[l_{i,j}] > T_c, \\ 1 - |\sin(\phi_{ij} - \phi_{ji})| & \text{otherwise,} \end{cases} \tag{19}$$

where $c_{co}[l_{i,j}]$ is the collinearity relation and T_c is a threshold, determining if π_i and π_j are collinear.
Like collinearity and co–colority, the symmetry of two 3D primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$ is computed using their 2D projections π_i and π_j:

Definition 3 *Two 3D–primitives $\mathbf{\Pi}_i$ and $\mathbf{\Pi}_j$ are said to be symmetric if the 2D–primitives π_i^x and π_j^x they project onto the camera plane x (defined by a projection relation $\mathcal{P}^x : \Pi_k \rightarrow \pi_k$) are symmetric (according to the definition of 2D–primitive symmetry presented above).*

4 Results

In figure 6, the coplanarity, cocolority and collinearity relations are shown for two different example scenes shown in figure 6(a) and (b). The results are from our 3D display tool called *Wanderer*, and for computational reasons, 3D primitives are shown in squares. The relations are displayed only for a primitive which is selected with the mouse as showing relations between all primitives disables visibility.
From the figure we see that coplanarity is a more common relation than cocolority or collinearity. This suggests that coplanarity alone is not directly usable for analysis or applications in 3D, and it needs to be accompanied with other relations as proposed and utilized in [2, 16].

5 Conclusion

In this paper, we presented cocolority, coplanarity, collinearity and symmetry relations defined on multi-modal visual features, called primitives.
Such relations have been utilized in different perceptual organization problems as well as analysis of how the natural scenes are structured (see, *e.g.*, ([3, 8, 10, 13, 16, 17, 21, 27, 31, 34, 39]), and the importance of such relations, as well as their psychophysical and biological plausibility have been acknowledged in the literature (see, *e.g.*, [18, 19, 35]).

6 Acknowledgments

We would like to thank Florentin Wörgötter and Daniel Aarno for their fruitful contributions. This work is supported by the Drivsco and the PACO+ projects.

References

[1] A. Baumberg. Reliable Feature Matching across Widely Separated Views. In *Proc. Conf. Computer Vision and Patter Recognition*, pages 774–781, 2000.

[2] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Model-independent grasping initializing object-model learning in a cognitive architecture. *IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.

[3] E. Brunswik and J. Kamiya. Ecological cue–validity of 'proximity' and of other Gestalt factors. *American Journal of Psychologie*, LXVI:20–32, 1953.

[4] Cordelia Schmid and Roger Mohr and Christian Baukhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[5] J. S. D. Aarno, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early reactive grasping with second order 3d feature relations. *IEEE Conference on Robotics and Automation (submitted)*, 2007.

[6] David G. Lowe. Distinctive Image Features from Scale–Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[7] J. Elder and R. Goldberg. Inferential reliability of contour grouping cues in natural images. *Perception Supplement*, 27, 1998.

[8] J. Elder and R. Goldberg. Ecological statistics of Gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353, 8 2002.

[9] J. H. Elder. Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122, 1999.

[10] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(25):1–14, 2003.

[11] Frederik Schaffalitzky and Andrew Zisserman. Multi–view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". *Lecture Notes in Computer Science*, 2350:414–431, 2002. in Proceedings of the BMVC02.

[12] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE-PAMI*, 13(9):891–906, 1991.

[13] W. E. L. Grimson. A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427, Sept. 1982.

[14] C. G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[15] J. J. Koenderink and A. J. van Doorn. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987.

[16] S. Kalkan, F. Wörgötter, and N. Krüger. Depth prediction at homogeneous image structures. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-2, 2007.

[17] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.

[18] K. Koffka. *Principles of Gestalt Psychology*. Lund Humphries, London, 1935.

[19] K. Köhler. *Gestalt Psychology: An introduction to new concepts in psychology*. New York: Liveright, 1947.

[20] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.

[21] N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998.

[22] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, pages 261–270, 2003.

[23] N. Krüger, M. V. Hulle, and F. Wörgötter. Ecovision: Challenges in early-cognitive vision. *International Journal of Computer Vision*, accepted.

[24] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.

[25] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behavious, AISB Journal*, 1(5):417–427, 2004.

[26] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.

[27] N. Krüger and F. Wörgötter. Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131:82–147, 2004.

[28] Luc Van Gool and Theo Moons and Dorin Ungureanu. Affine / Photometric Invariants for Planar Intensity Patterns. *Lecture Notes In Computer Science*, 1064:642–651, 1996. in Proceedings of the 4th European Conference on Computer Vision — Volume 1.

[29] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. Freeman, 1977.

[30] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[31] N. Pugeault, N. Krüger, and F. Wörgötter. A non-local stereo similarity based on collinear groups. *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems*, 2004.

[32] N. Pugeault, N. Krüger, and F. Wörgötter. Rigid body motion estimation in an early cognitive vision framework. In *IEEE Advances In Cybernetic Systems*, 2006.

[33] N. Pugeault, F. Wörgötter, , and N. Krüger. Disambiguation .....

[34] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.

[35] S. Sarkar and K. Boyer. *Computing Perceptual Organization in Computer Vision*. World Scientific, 1994.

[36] E. W. Weisstein. Coplanar. from mathworld–a wolfram web resource, 2006. `http://mathworld.wolfram.com/Coplanar.html`.

[37] Wikipedia. Coplanarity — wikipedia, the free encyclopedia, 2006. `http://en.wikipedia.org/w/index.php?title=Coplanarity&oldid=37490165`.

[38] X. Zhang and B. A. Wandell. Color image fidelity metrics evaluated using image distortion maps. *Signal Processing*, 70(3):201–214, 1998.

[39] S. C. Zhu. Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, 1999.

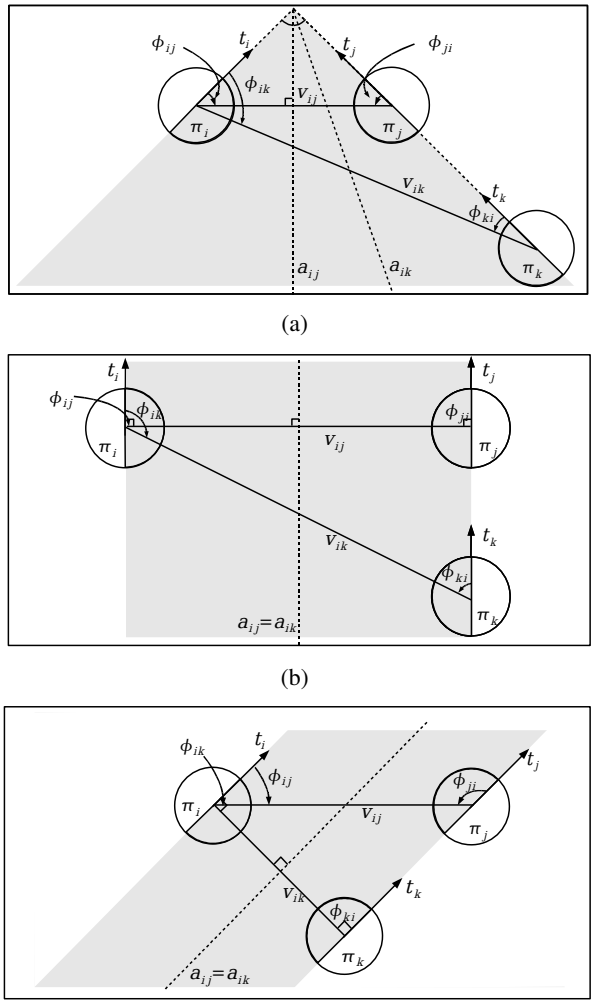


Figure 5: Illustration of the definition of symmetry. t_i, t_j and t_k denote the vectors defined by the orientations θ_i, θ_j and θ_k, respectively. Primitives π_i and π_j are symmetric in (a) and (b), but not in (c). π_i and π_k are symmetric in (c), but not in (a) or (b).

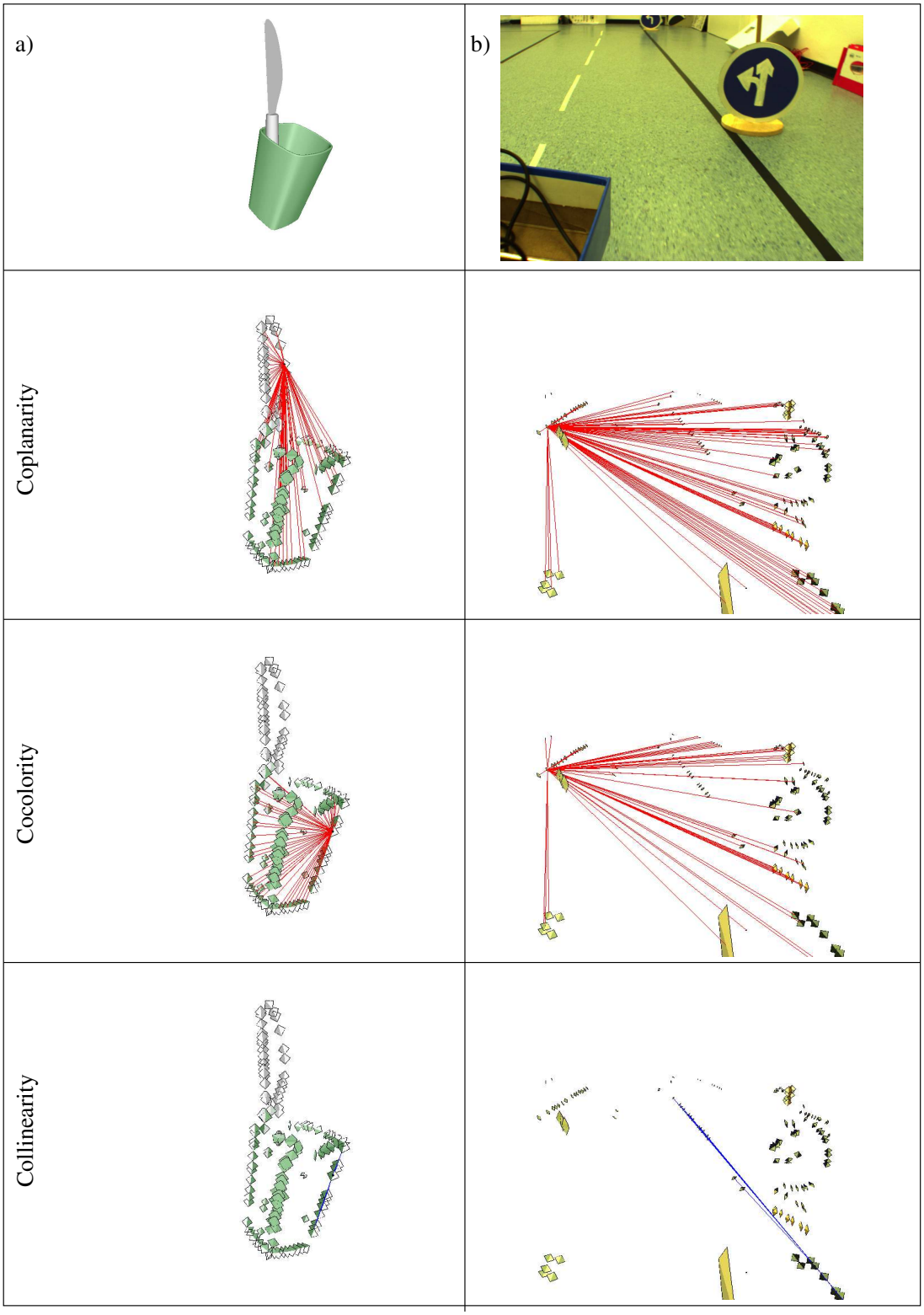


Figure 6: The coplanarity, cocolority and collinearity relations on two different examples shown in (a) and (b). The results are from our 3D display tool called *Wanderer*, and for the sake of speed, 3D primitives are shown in squares. The relations are shown only for a selected primitive as showing relations between all primitives disables visibility.

Published on IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006.

Statistical Analysis of Local 3D Structure in 2D Images

Sinan KALKAN
Bernstein Centre for Computational Neuroscience,
University of Göttingen, Germany
sinan@chaos.gwdg.de

Florentin Wörgötter
Bernstein Centre for Computational Neuroscience,
University of Göttingen, Germany
worgott@chaos.gwdg.de

Norbert Krüger
Cognitive Vision Group,
Aalborg University Copenhagen, Denmark
nk@media.aau.dk

Abstract

*For the analysis of images, a deeper understanding of their intrinsic structure is required. This has been obtained for 2D images by means of statistical analysis [15, 18]. Here, we analyze the relation between local image structures (*i.e.*, homogeneous, edge-like, corner-like or texture-like structures) and the underlying local 3D structure, represented in terms of continuous surfaces and different kinds of 3D discontinuities, using 3D range data with the true color information. We find that homogeneous image patches correspond to continuous surfaces, and discontinuities are mainly formed by edge-like or corner-like structures. The results are discussed with regard to existing and potential computer vision applications and the assumptions made by these applications.*

1. Introduction

With the notion that the human visual system is adapted to the statistics of the environment [2, 13, 15, 18, 22, 21] and its successful applications to grouping, object recognition and stereo [3, 4, 20, 29] the analysis, and the usage of natural image statistics has become an important focus of vision research. Moreover, with the advances in technology, it has been also possible to analyze the underlying 3D world using 3D range scanners [10, 11, 19, 27].

In this paper, we analyze the relation between local image structures (*i.e.*, homogeneous, edge-like, corner-like or texture-like structures) and the underlying local 3D structure using 3D range data with the true color information.

There have been only a few studies that have analyzed the 3D world from range data [10, 11, 19, 27]. In [27], the distribution of roughness, size, distance, 3D orientation, curvature and independent components of surfaces was analyzed. Their major conclusions were: (1) local 3D patches tend to be saddle-like, and (2) natural scene geometry is quite regular and less complex than luminance images. In [11], the distribution of 3D points was analyzed using co-occurrence statistics and 2D and 3D joint distributions of Haar filter reactions. They showed that range images are much simpler to analyze than optical images and that a 3D scene is composed of piecewise smooth regions. In [19], the correlation between light intensities of the image data and the corresponding range data as well as surface convexity were investigated. They could justify the event that brighter objects are closer to the viewer, which is used by shape from shading algorithms in estimating depth. In [9, 10], range image statistics were analyzed for explanation of several visual illusions.

Our analysis differs from these works. For 2D local image patches, existing studies have only considered light intensity. As for 3D local patches, the most complex considered representation have been the curvature of the local 3D patch. In this work, however, we create a higher-order representation of the 2D local image patches and the 3D local patches; we measure 2D local image patches using homogeneous, edge-like, corner-like or texture-like structures, and 3D local patches using continuous surfaces and different kinds of 3D discontinuities. By this, we relate established local image structures to their underlying 3D structures.

By creating 2D and 3D representations of the local structure, we compute the conditional probability $P(\text{3D Structure} \mid \text{2D Structure})$. Using this probability, we quantify some assumptions made by the studies that reconstruct the 3D world from dense range data. For example, we could show that the depth distribution varies significantly for different visual features, and we could quantify already established inter-dependencies such as 'no new is

good news' [6]. This work also supports the understanding of how intrinsic properties 2D–3D relations can be used for the reconstruction of depth, for example, by using statistical priors in the formalisation of depth cues.

The paper is organized as follows: In section 2, we define the types of local image structures and local 3D structures that we extract for our analysis. In section 3, we introduce a continuous classifier for local 2D structures. In section 4, we outline our methods for measuring the 3D structure of a 3D point. We present and discuss our results in section 5. Finally, we conclude the paper in section 6.

2. Local 2D and 3D Structures

We distinguish between the following local 2D structures:

- Homogeneous image patches: Homogeneous patches are signals of uniform intensities.
- Edge–like structures: Edges are low-level structures which constitute the boundaries between homogeneous or texture-like signals (see, *e.g.*, [14, 17] for their importance in vision).
- Corners: Corners are signals where two or more edge-like structures with significantly different orientations intersect (see, *e.g.*, [7, 23, 24] for their importance in vision).
- Texture: Although there is not a widely-agreed definition, textures are often defined as signals which consist of repetitive, random or directional structures (for their analysis, extraction and importance in vision, see *e.g.*, [26]).

Locally, it is hard to distinguish between these structures, and there are structures that carry mixed properties of the 'ideal' cases. The classification of the features outlined above is discrete. However, a discrete classification may cause problems as the inherent properties of "mixed" structures are lost in the discretization process. Instead, in this paper, we make use of a recently developed continuous scheme which is based on the concept of intrinsic dimensionality [5, 16]. In this concept, local image structures are organized continuously in a triangle. This approach is briefly described in section 3. Here, we show that the different classes of local image structures map to different distinguishable areas in the domain of the intrinsic dimensionality triangle (see figure 2) which is the first contribution of this paper.

To our knowledge, there does not exist a systematic and agreed classification of 3D local structures like there is for 2D local image structures (*i.e.*, homogeneous patches, edges, corners and textures). Intuitively, the 3D world consists of continuous surface patches and different kinds of 3D discontinuities. In the imaging process (through the lenses of camera or a retina), 2D local image structures are formed by these 3D structures together with the illumination and reflectivity of the environment.

With this intuition, any 3D scene can be decomposed geometrically into surfaces and 3D discontinuities. In this context, the local 3D structure of a point can be a:

- Surface Continuity: The underlying 3D structure can be described by one surface whose normal does not change or changes smoothly.
- Regular Gap discontinuity: The underlying 3D structure can be described by a small set of surfaces with a significant depth difference. The 2D and 3D views of an example gap discontinuity are shown in figure 1(a).
- Irregular Gap discontinuity: The underlying 3D structure shows high depth variation and can not be described by two or three surfaces. An example of an irregular gap discontinuity is shown in figure 1(b).
- Orientation Discontinuity: The underlying 3D structure can be described by two surfaces with significantly different 3D orientations that meet at the point whose 3D structure is being questioned. In this type of discontinuity, no gap but a change in 3D orientation between the meeting surfaces occurs. An example for this type of discontinuity is shown in figure 1(c).

3. Intrinsic Dimensionality

In image processing, intrinsic dimensionality was introduced by Zetsche and Barth[28] to distinguish between different local image structures. The idea is to assign intrinsically zero dimensionality (i0D), intrinsically one dimensionality (i1D) and intrinsically two dimensionality (i2D) to homogeneous patches, edges and corner-like structures, respectively. The concept of intrinsic dimensionality has been mostly applied in a discrete way which has been extended in [5, 16] to classify the local image structures continuously instead of giving them discrete labels.

In [5, 16], it has been also shown that the topological structure of the intrinsic dimensionality can be understood as a triangle whose corners correspond to the 'ideal' cases of 2D structures (*i.e.*, homogeneous patches, edges and corners). The inner of the triangle spans signals that carry aspects of the three 'ideal' cases, and the distance from the specific corners indicates the similarity (or dissimilarity) to the 'ideal' i0D, i1D and i2D signals. The horizontal and the vertical axes denote the contrast and the orientation variance, respectively. Contrast measures non-homogeneity whereas orientation variance measures the variation of orientation in a local patch describing the local image structure. An 'ideal' homogeneous image patch is expected to have zero contrast and zero orientation variance whereas an 'ideal' edge should have high contrast and zero orientation variance. An 'ideal' corner is supposed to have high contrast and high orientation variance.

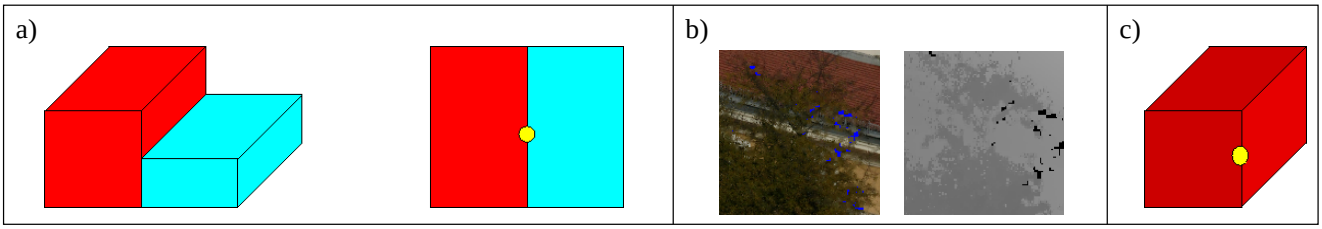


Figure 1. Examples for types of 3D discontinuities. Points of interest are marked with yellow circles. (a) 2D and 3D views of a gap discontinuity, (b) image (on the left) and range data (on the right) of an irregular gap discontinuity and (c) orientation discontinuity.

Figure 2 shows how the triangle of intrinsic dimensionality looks like and how a set of example local image structures map on to it. In figure 2, we see that different visual structures map to different areas in the triangle. A detailed analysis of how 2D structures are distributed over the intrinsic dimensionality triangle and how some visual information depends on this distribution can be found in [12]. Different from [12], in this paper, regarding this distribution, we show that textures also map to a different area of their own. The fact that different local image structures have their own distinguishable areas in the triangle provides us with a continuous classifier that distinguishes between homogeneous, edge-like, texture-like and corner-like structures.

4. Methods

In this section, we define our measures for the three kinds of discontinuities that we described in section 1; namely, gap discontinuity, irregular gap discontinuity and orientation discontinuity. The measures for gap discontinuity, irregular gap discontinuity and orientation discontinuity of a patch P will be respectively denoted by $\mu_{GD}(P)$, $\mu_{IGD}(P)$ and $\mu_{OD}(P)$. The reader who is not interested in the technical details can jump directly to section 5.

In our analysis, we used chromatic range data of outdoor scenes[1] which were obtained from Riegl UK Ltd. (`http://www.riegl.co.uk/`). There were 20 scenes in total, 10 of which are shown in figure 3. The range of an object which does not reflect the laser beam back to the scanner or is out of the range of the scanner cannot be measured. These points are marked with blue in figure 3 and are not processed in our analysis. The resolution range of the data set is [512-2048]x[390-2290] with an average resolution of 1140x1001.

3D discontinuities are detected in studies which involve range data processing, using different methods and using different names like two-dimensional discontinuous edge, jump edge or depth discontinuity for gap discontinuity; and, two-dimensional corner edge, crease edge or surface discontinuity for orientation discontinuity [1, 8, 25].

[1]We would like to note that it is problematic to do range scanning in nature scenes that include trees or other kinds of vegetation because of the unintended motion due to wind. As the image of the scene is taken after the scanning phase, this delay may make the image data fail to correspond to the range data.

4.1. Measure for Gap Discontinuity: μ_{GD}

Gap discontinuities can be measured or detected in a similar way to edges in 2D images; edge detection processes RGB-coded 2D images while for a gap discontinuity, one needs to process XYZ-coded 2D images. In other words, gap discontinuities can be measured or detected by taking a second order derivative of XYZ values [25].

Measurement of a gap discontinuity is expected to operate on both the horizontal and vertical axes of the 2D image; that is, it should be a two dimensional function. The alternative is to discard the topology and do 'edge-detection' in sorted XYZ values, *i.e.*, to operate as a one-dimensional function. Although we are not aware of a systematic comparison of the alternatives, for our analysis and for our data, the topology-discarding gap discontinuity measurement produced better results. Therefore, we have adopted the topology-discarding gap discontinuity measurement in the rest of the paper.

For an image patch P of size $N \times N$, let,

$$\begin{aligned} \mathcal{X} &= \text{ascending\_sort}(\{X_i \mid i \in P\}), \\ \mathcal{Y} &= \text{ascending\_sort}(\{Y_i \mid i \in P\}), \\ \mathcal{Z} &= \text{ascending\_sort}(\{Z_i \mid i \in P\}), \end{aligned} \tag{1}$$

and also, for $i = 1, .., (N \times N - 2)$,

$$\begin{aligned} \mathcal{X}^{\Delta} &= \{ \mid (\mathcal{X}_{i+2} - \mathcal{X}_{i+1}) - (\mathcal{X}_{i+1} - \mathcal{X}_i) \mid \}, \\ \mathcal{Y}^{\Delta} &= \{ \mid (\mathcal{Y}_{i+2} - \mathcal{Y}_{i+1}) - (\mathcal{Y}_{i+1} - \mathcal{Y}_i) \mid \}, \\ \mathcal{Z}^{\Delta} &= \{ \mid (\mathcal{Z}_{i+2} - \mathcal{Z}_{i+1}) - (\mathcal{Z}_{i+1} - \mathcal{Z}_i) \mid \}, \end{aligned} \tag{2}$$

where $\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i$ represents 3D coordinates of pixel i.

The sets $\mathcal{X}^{\Delta}, \mathcal{Y}^{\Delta}$ and $\mathcal{Z}^{\Delta}$ are the measurements of the jumps (*i.e.*, second order differentials) in the sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$, respectively. A gap discontinuity can be defined simply as a measure of these jumps in these sets. In other words:

$$\mu_{GD}(P) = \frac{\phi(\mathcal{X}^{\Delta}) + \phi(\mathcal{Y}^{\Delta}) + \phi(\mathcal{Z}^{\Delta})}{3}, \tag{3}$$

where the function $\phi : \mathcal{S} \rightarrow [0, 1]$ over the set $\mathcal{S}$ measures the homogeneity of its argument set (in terms of its 'peakiness') and is defined as follows:

$$\phi(\mathcal{S}) = \frac{1}{\#(\mathcal{S})} \times \sum_{i \in \mathcal{S}} \frac{s_i}{\max(\mathcal{S})}, \tag{4}$$

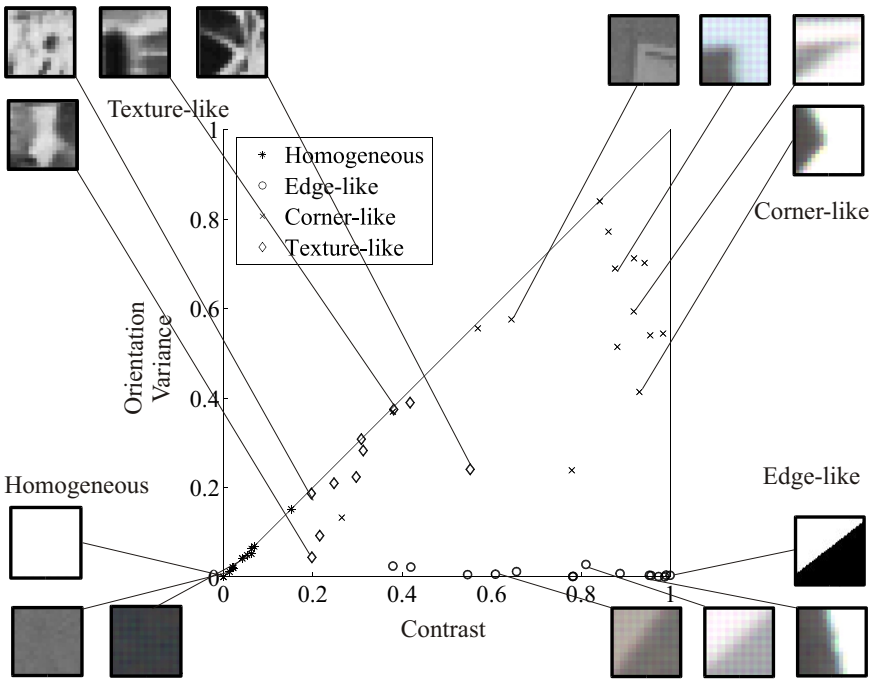


Figure 2. How a set of 54 patches map to the different areas of the intrinsic dimensionality triangle. Some examples from these patches are also shown. The horizontal and vertical axes of the triangle denote the contrast and the orientation variances of the image patches, respectively.

Figure 3. 10 of the 20 3D data sets used in the analysis. The points that don't have range data are marked in blue. The gray image shows the range data of the top-left scene. The resolution range is [512-2048]x[390-2290] with an average resolution of 1140x1001.

where $\#(\mathcal{S})$ is the number of the elements of $\mathcal{S}$, and s_i is the i^{th} element of the set $\mathcal{S}$. Note that as a homogeneous set (*i.e.*, a non-gap discontinuity) $\mathcal{S}$ produces a high $\phi(\mathcal{S})$ value, a gap discontinuity causes a low μ_{GD} value. Figure 5(c) shows the performance of μ_{GD} on one of our scenes shown in figure 3.

4.2. Measure for Orientation Discontinuity: μ_{OD}

The orientation discontinuity of a patch P can be detected or measured by taking the 3D orientation difference of the surfaces which meet at P. As the size of the patch P is small enough, the surfaces can be, in practice, approximated by 2-pixel wide unit planes. The histogram of the 3D orientation differences between every pair of unit planes forms one cluster for continuous surfaces and two clusters for orientation discontinuities.

For an image patch P of size $N \times N$ pixels, the orientation discontinuity measure is defined as:

$$\mu_{\mathbf{OD}}(P) = \psi(H^n(\{\alpha(i,j) \mid i,j \in planes(P), i \neq j\})), \quad (5)$$

where $H^n(S)$ is a function which computes the n-bin histogram of its argument set $\mathcal{S}$; $\psi(\mathcal{S})$ is a function which finds the number of clusters in $\mathcal{S}$; $planes(P)$ is a function which fits 2-pixel-wide unit planes to 1-pixel apart points in P using Singular Value Decomposition[2]; and, $\alpha(i,j)$ is the angle between planes i and j.

For a histogram H of size N_H, the number of clusters is:

$$\psi(S) = \frac{\sum_{i=1}^{N_H+1}(H_i > \frac{\max(H)}{10}) \neq (H_{i-1} > \frac{\max(H)}{10})}{2}, \quad (6)$$

[2]Singular Value Decomposition is a standard technique for fitting planes to a set of points. It finds the perfectly fitting plane if it exists; otherwise, it returns the least-square solution.

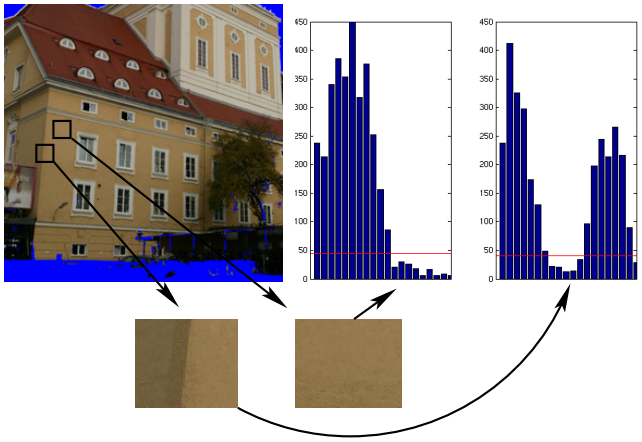


Figure 4. Example histograms and the number of clusters that the function $\psi(S)$ computes. $\psi(S)$ finds one cluster in the left histogram and two clusters in the right histogram. Red line marks the threshold value of the function. X axis denotes the values for 3D orientation differences.

where the operator $\neq$ returns 1 if its operands are not equal and returns 0, otherwise; H_i represents the i^{th} element of the histogram H; H_0 and H_{N_H+1} are defined as zero; and, $\max(H)/10$ is an empirical value which functions as the threshold value for finding the clusters. Figure 4 shows two example clusters for a continuous surface and an orientation discontinuity. Figure 5(d) shows the performance of μ_{OD} on one of our scenes shown in figure 3.

4.3. Measure for Irregular Gap Discontinuity: μ_{IGD}

Irregular gap discontinuity of a patch P can be measured by making use of the observation that an irregular-gap discontinuous patch from nature usually consists of small surface fragments with different 3D orientations. Therefore, the amount of variety in the 3D orientation histogram of a patch P can measure the irregular gap discontinuity of P.

Similar to the measure for orientation discontinuity defined in section 4.2, the histogram of the differences between the 3D orientations of the unit planes (which are of 2 pixels wide) is analyzed. For an image patch P of size $N \times N$ pixels, the irregular gap discontinuity measure is defined as:

$$\mu_{\mathbf{IGD}}(P) = \phi(H^n(\{\alpha(i,j) \mid i,j \in planes(P), i \neq j\})), \qquad (7)$$

where $planes(P)$, $\alpha(i,j)$, $H^n(\mathcal{S})$ and $\phi(\mathcal{S})$ are as defined in section 4.2. Figure 5(e) shows the performance of μ_{IGD} on one of our scenes shown in figure 3.

The relation between the measurements and the types of the 3D discontinuities are outlined in table 1 which entails that an image patch P is:

- gap discontinuous if $\mu_{GD}(P) < T_g$ and $\mu_{IGD}(P) < T_{ig}$,
- irregular-gap discontinuous if $\mu_{GD}(P) < T_g$ and $\mu_{IGD}(P) > T_{ig}$,
- orientation discontinuous if $\mu_{GD}(P) \geq T_g$ and $\mu_{OD} > 1$,

Dis. Type	μ_{GD}	μ_{IGD}	μ_{OD}
Continuity	High value	Don't care	1
Gap Dis.	Low value	Low value	Don't care
Irregular Gap Dis.	Low value	High value	Don't care
Orientation Dis.	High value	Don't care	> 1

Table 1. The relation between the measurements and the types of the 3D discontinuities.

- continuous if $\mu_{GD}(P) \geq T_g$ and $\mu_{OD}(P) \leq 1$.

For our analysis, we have taken N and the threshold values T_g, T_{ig} empirically as 10, 0.4 and 0.6, respectively. The number of bins, n, in H^n is taken as 20.

Figure 5(a) shows the types of 3D discontinuities marked in four different colors for every pixel of the scenes shown in figure 3. We see that our measures can capture the 3D structure of the data sufficiently correct.

5. Results and Discussion

For each pixel of the scene (except for pixels where range data is not available), we computed the 3D discontinuity type and the intrinsic dimensionality. Figure 5(a) and (b) shows the images where the 3D discontinuity and the intrinsic dimensionality of each pixel are marked with different colors.

Having the 3D discontinuity type and the information about the local 2D structure of each point, it is straightforward to compute the probability $P(\text{3D Discontinuity} \mid \text{2D Structure})$, which is shown in figure 6. Note that the four triangles in figures 6(a), 6(b), 6(c) and 6(d) add up to one for all points of the triangle. We see that:

- Figure 6(a) shows that homogeneous image patches correspond to 3D continuities.

  Many surface reconstruction studies make use of a basic assumption that there is a smooth surface between any two points in the 3D world, if there is no contrast difference between these points in the image. This assumption has been first called as 'no news is good news' in [6]. With figure 6(a), we quantify 'no news is good news' and show for which structures and to what extent it holds. In addition to the fact that no news is in fact good news, the figure shows that news, especially texture-like structures and edge-like structures, can also be good news (see below).
- Edges are considered as important sources of information for object recognition and reliable correspondence finding. Approximately 10% of local image structures are of that type (see, *e.g.*, [12]). Figures 6(a), (b) and (d) show that most of the edges correspond to continuous surfaces or gap discontinuities. The edges that correspond to continuous surfaces are mostly low-contrast edges. Little percentage of the edges are formed by orientation discontinuities.

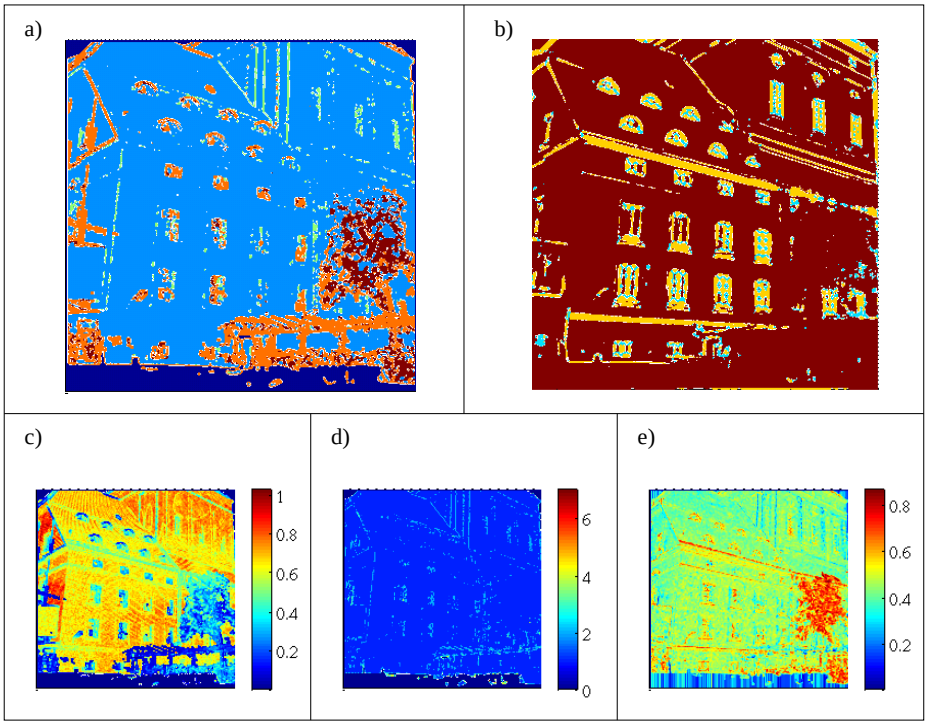


Figure 5. The 3D and 2D information for one of the scenes shown in figure 3. Dark blue marks the points without range data. (a) 3D discontinuity. Blue: continuous surfaces, light blue: orientation discontinuities, orange: gap discontinuities and brown: irregular gap discontinuities. (b) Intrinsic Dimensionality. Homogeneous patches, edge-like and corner-like structures are encoded in colors brown, yellow and light blue, respectively. (c) Gap discontinuity measure μ_{GD}. (d) Orientation discontinuity measure μ_{OD}. (e) Irregular gap discontinuity measure μ_{IGD}.

- Figure 6(b) shows that well-defined corner-like structures result from either gap discontinuities or continuities.
- Textures also map with high likelihood to surface continuities but also to irregular gap discontinuities.

  Finding correspondences becomes more difficult with the lack or repetitiveness of the local structure. The estimates of the correspondences at texture-like structures are naturally less reliable. In this sense, the likelihood that certain textures are caused by continuous surfaces (shown in figure 6(a)) can be used to model stereo matching functions that include interpolation as well as information about possible correspondences based on the local image information.

It is remarkable that local image structures mapping to different sub-regions in the triangle are caused by rather different 3D structures. This clearly indicates that these different image structures should be used in different ways for surface reconstruction.

6. Conclusion

In this paper, using 3D range data with real-world color information, we have analyzed the conditional probability of a 3D structure given the 2D structure. With this probability, we could investigate the relation between 2D structures and the underlying 3D structures as well as analyze the validity of a widely-used assumption/smoothing constraint, namely, 'no news is good news' [6].

Besides, we have presented a continuous classification scheme which can be used to distinguish between homogeneous, edge-like, corner-like and texture-like structures. By taking a higher-order representation than existing range-data analysis studies, we could point to the intrinsic properties of the 3D world and its relation to the image data. This analysis is important because (1) it may be that the human visual system is adapted to the statistics of the environment [2, 13, 15, 18, 21, 22], and (2) it may be used in several computer vision applications like depth estimation in a similar way as in [3, 4, 20, 29].

In our current work, the probability distributions will be used for estimating the 3D structure from 2D structure in a Bayesian framework for surface reconstruction/interpolation studies.

7. Acknowledgments

We would like to thank RIEGL UK Ltd. for providing us with 3D range data. This work is supported by the ECO-

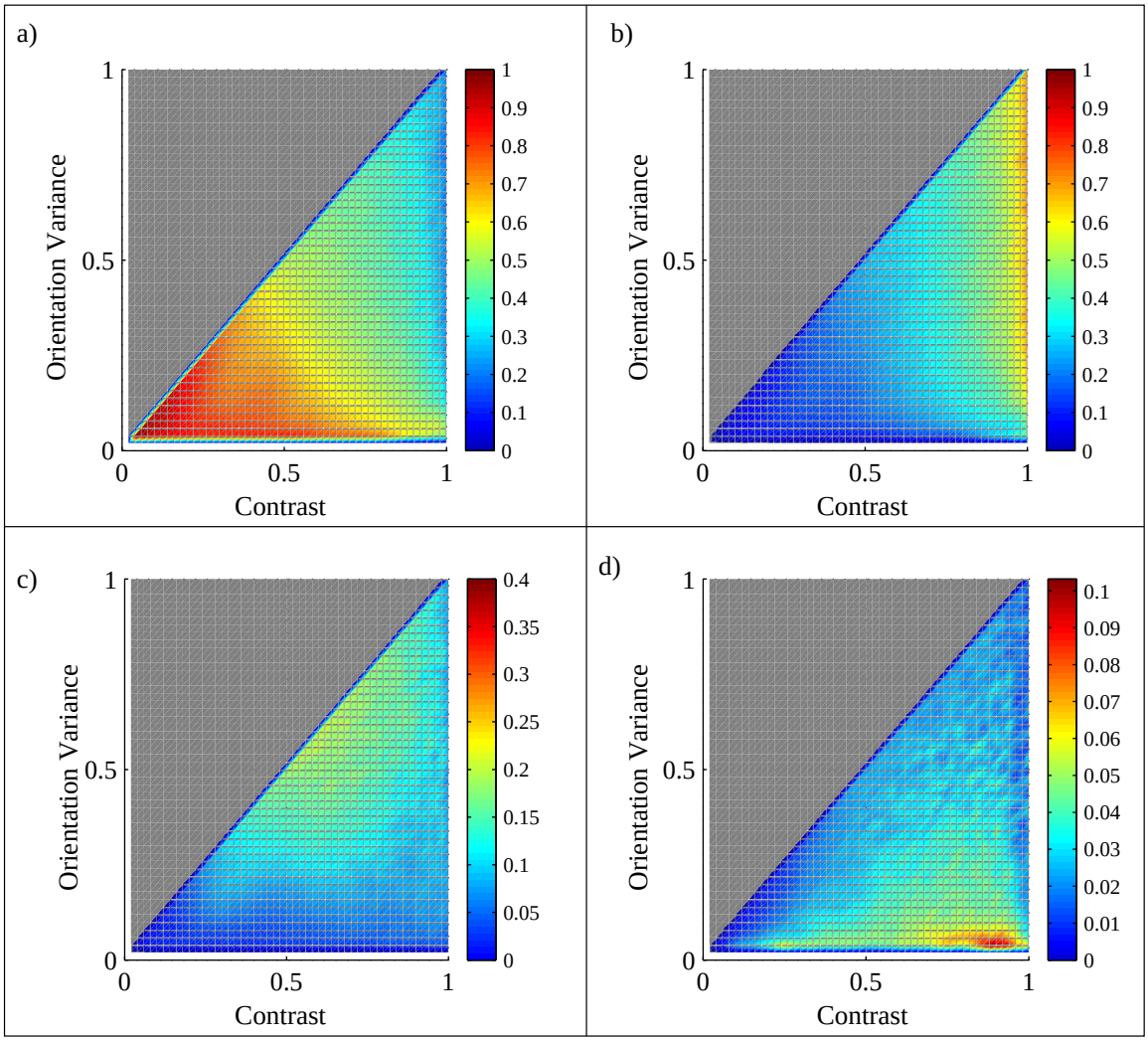


Figure 6. P(3D Discontinuity | 2D Structure): (a) P(Continuity | 2D Structure). (b) P(Gap Discontinuity | 2D Structure). (c) P(Irregular Gap Discontinuity | 2D Structure). (d) P(Orientation Discontinuity | 2D Structure).

VISION project.

References

[1] R. M. Bolle and B. C. Vemuri. On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13, 1991.

[2] E. Brunswik and J. Kamiya. Ecological cue–validity of 'proximity' and of other Gestalt factors. *American Journal of Psychologie*, LXVI:20–32, 1953.

[3] H. Elder and R. Goldberg. Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353, 2002.

[4] J. H. Elder, A. Krupnik, and L. A. Johnston. Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(25):1–14, 2003.

[5] M. Felsberg and N. Krüger. A probablistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*, 2003.

[6] W. E. L. Grimson. Surface consistency constraints in vision. *Computer Vision, Graphics and Image Processing*, 24(1):28–51, Oct. 1983.

[7] A. Guzman. Decomposition of a visual scene into three-dimensional bodies. *AFIPS Fall Joint Conference Proceedings*, 33:291–304, 1968.

[8] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(7):673–689, 1996.

[9] C. Q. Howe and D. Purves. Range image statistics can explain the anomalous perception of length. *PNAS*, 99(20):13184–13188, 2002.

[10] C. Q. Howe and D. Purves. Size contrast and assimilation explained by the statistics of natural scene geometry. *Journal of Cognitive Neuroscience*, 16(1):90–102, 2004.

[11] J. Huang, A. B. Lee, and D. Mumford. Statistics of range images. *CVPR*, 1(1):1324–1331, 2000.

[12] S. Kalkan, D. Calow, F. Wörgötter, M. Lappe, and N. Krüger. Local image structures and optic flow estimation. *Accepted for Network: Computation in Neural Systems*, 2005.

[13] D. C. Knill and W. Richards, editors. *Perception as bayesian inference*. Cambridge: Cambridge University Press, 1996.

[14] J. Koenderink and A. Dorn. The shape of smooth objects and the way contours end. *Perception*, 11:129—173, 1982.

[15] N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998.

[16] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, 2003.

[17] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. Feeman, 1977.

[18] B. Olshausen and D. Field. Natural image statistics and efficient coding. *Network*, 7:333–339, 1996.

[19] B. Potetz and T. S. Lee. Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes. *Journal of the Optical Society of America*, 20(7):1292–1303, 2003.

[20] N. Pugeault, N. Krüger, and F. Wörgötter. A non-local stereo similarity based on collinear groups. *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems*, 2004.

[21] D. Purves and B. Lotto, editors. *Why we see what we do: an empirical theory of vision*. Sunderland, MA: Sinauer Associates, 2002.

[22] R. P. N. Rao, B. A. Olshausen, and M. S. Lewicki, editors. *Probabilistic models of the brain*. MA: MIT Press, 2002.

[23] N. Rubin. The role of junctions in surface completion and contour matching. *Perception*, 30:339–366, 2001.

[24] I. A. Shevelev, V. M. Kamenkovich, and G. A. Sharaev. The role of lines and corners of geometric figures in recognition performance. *Acta Neurobiol Exp*, 63(4):361–368, 2003.

[25] Y. Shirai. *Three-dimensional computer vision*. Springer-Verlag New York, Inc., 1987.

[26] M. Tuceryan and N. K. Jain. Texture analysis. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, pages 207–248, 1998.

[27] Z. Yang and D. Purves. Image/source statistics of surfaces in natural scenes. *Network: Computation in Neural Systems*, 14:371–390, 2003.

[28] C. Zetzsche and E. Barth. Fundamental limits of linear filters in the visual processing of two dimensional signals. *Vision Research*, 30(7):1111–1117, 1990.

[29] S. C. Zhu. Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, 1999.

Robotics Group
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Technical Report no. 2007 – 2

Depth Prediction at Homogeneous Image structures

Sinan Kalkan, Florentin Wörgötter, Norbert Krüger

January 22, 2007

Title	Depth Prediction at Homogeneous Image structures
Author(s)	Sinan Kalkan, Florentin Wörgötter, Norbert Krüger
Publication History	

Abstract

Depth at homogeneous or weakly-textured image areas is difficult to obtain because such image areas suffer the well-known correspondence problem. In this paper, we propose a voting model that predicts the depth at such image areas from the depth of bounding edge-like structures. The depth at edge-like structures is computed using a feature-based stereo algorithm, and is used to vote for the depth of homogeneous image areas. We show the results of our ongoing work on different scenarios.

1 Introduction

Extraction of 3D structure from 2D images is realized utilizing a set of inverse problems that include structure from motion, stereo vision, shape from shading, linear perspective, texture gradients and occlusion [3]. These cues can be classified as pictorial, or monocular (such as shading, utilization of texture gradients or linear perspective) and multi-view (like stereo and structure from motion). Depth cues which make use of multiple views require correspondences between different 2D views of the scene. In contrast, pictorial cues use statistical and geometrical relations in one image to make statements about the underlying 3D structure. Many surfaces have only weak texture or no texture at all, and as a consequence, the *correspondence problem is very hard or not at all resolvable for these surfaces*. Nevertheless, humans are able to reconstruct 3D information for these surfaces, too. Existing psychophysical experiments (see, *e.g.*, [2, 4]) and computational theories (see, *e.g.*, [1, 6, 24]) suggest that in the human visual system, *an interpolation process* is realized that starting with the local analysis of edges, corners and textures, computes depth also in areas where correspondences cannot easily be found.

In this paper, we are interested in prediction of depth at homogeneous image patches (called *mono*s in this paper) from the depth of the edges in the scene using a voting model. We start by creating a representation of the input stereo images in terms of local image patches corresponding to edge-like structures and monos (as introduced in [14] and section 2, and described in detail in [15]). The depth at edge-like patches is extracted using feature-based stereo computation between the two images (using the method introduced in [20]). The depth that is extracted at the bounding edge-like patches of a mono using stereo votes for its depth.

We would like to distinguish *depth prediction* from *surface interpolation* because surface interpolation assumes that there is already a dense depth map of the scene available in order to be able to estimate the 3D orientation at points (see, *e.g.*, [6, 7, 8, 17, 18, 23, 24]) whereas our understanding of depth prediction makes use of only 3D line-orientations at edge-segments which are computed using a feature-based stereo proposed in [20].

(a) (b) (c)

Figure 1: An input stereo pair ((a) and (b)) and how a feature-based stereo algorithm (taken from [20]) looks like (c).

A typical scenario that our model is designed for is shown in figure 1 where an input stereo pair and the stereo data (computed using [20]) are displayed. We see that computed stereo information has strong outliers which prohibit a *surface interpolation* method as it is not possible to differentiate between the outliers and the reliable stereo information. Moreover, the stereo information that should be reliable at the edges of the road turn out not to share a common surface nor the same 3D line (see figure 1(c)). Applying a surface interpolation method on such input data is expected to lead to a wrong road surface prediction. In this paper, we will show that our depth prediction method is able to cope with such strong outliers.

1.1 Related studies

It is fair to count the early works of Grimson [6] as the pioneers of surface interpolation. In [6], Grimson proposed fitting square Laplacian functionals to surface orientations at existing 3D points utilizing a *surface consistency constraint* called 'no news is good news'. The constraint argues that if two image points do not have a contrast difference in-between, then they can be assumed to be on the same 3D surface (see [11] for a quantification of this assumption). This work is extended in [7] with use of shading information. [6, 7] assume that surface information is available, and the input 3D points are dense enough for second order differentiation.

In [1], surface orientation at homogeneous image areas is recovered by *interpreting* line *drawings*. Lines are classified as extremal or discontinuity by making use of the junction labels and global relations like symmetry and parallellism. They assume that (1) extremal points (the boundaries of the objects) in an image correspond to surface orientations which are normal to the image curve and the line of sight, and that (2) discontinuities (lines other than extremal points) lead to surface orientations which are normal to space curve. The underlying assumptions of [1] are that (1) a clean contour of the scene is provided, and that (2) the object is separated from the background. Moreover, the results provided in [11] suggest that it may not be a good idea to assume that edges correspond to only certain types of surface orientations. [19, 22, 25, 26] are similar to [1] as far as our paper is concerned.

In [8], 3D points with surface orientation are interpolated using a perceptual constraint called *co-surfacity* which produces a 3D association field (which is called Diabolo field by the authors) similar to the association field used in 2D perceptual contour grouping studies. If the points do not have 3D orientation, they estimate the 3D orientation first and then apply the surface interpolation step. In [17, 18], it is argued that stereo matching and surface interpolation should not be sequential but rather simultaneous. For this, they employ the following steps: (1) Normalized-cross correlation and edge-based stereo are computed. (2) The disparities are combined and disparities corresponding to inliers, surfaces and surface discontinuities are marked using tensor voting. (3) Surfaces are extracted using marching cubes approach. At this stage, surfaces are over the boundaries. (4) At the last step, over-boundary surfaces are trimmed. They assume sphere as their surface model when interpolating surface orientations.

In [23, 24], stereo is computed at different scales, and instead of collapsing the results of these different scales into a single layer of disparity estimation and then applying surface interpolation, surface interpolation is applied separately for each scale and the results are combined.

Our work is different from the above mentioned worksin that:

- Our approach does not assume that the input stereo points are dense enough to compute their 3D orientation (this is why the authors of this paper prefer to distinguish between depth prediction and surface interpolation). Instead, our method relies on the 3D line-orientations of the edge segments which are extracted using a feature-based stereo algorithm (proposed in [20]).

- We employ a voting method like [17, 18] but is different, allowing long-range interactions in empty image areas, in order to predict *both* the depth and the surface orientation.

The paper is organized as follows: In section 2, we introduce how the images are represented in terms of local image patches. Section 3 describes the 2D and 3D relations between the local image patches that are utilized in the depth prediction process. Section 4 gives the outline of how the depth prediction is performed. In section 5, the results are presented and discussed. Finally, in section 6, the paper is concluded.

2 Visual Features

The visual features we utilize (called primitives in the rest of the paper) are local, multi-modal feature descriptors that were intoduced in [14]. They are semantically and geometrically meaningful descriptions of local patches, motivated by the hyper-columnar structures in V1 ([9]).
An edge-like primitive can be formulated as:

$$\pi^{\mathrm{e}} = (\mathbf{x}, \theta, \omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r), f), \tag{1}$$

where $\mathbf{x}$ is the image position of the primitive; θ is the 2D orientation; ω represents the contrast transition; $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$) of the primitive; and, f is the optical flow extracted using Nagel-Enkelmann optic flow algorithm.
As the underlying structure of an homogeneous image patch is different from that of an edge-like patch, a different representation is needed for homogeneous image structures (called *mono*s in this paper):

$$\pi^{\mathrm{m}} = (\mathbf{x}, \mathbf{c}), \tag{2}$$

where $\mathbf{x}$ is the image position, and $\mathbf{c}$ is the color of the mono.
See [16] for more information about these modalities and their extraction. Figure 2 shows extracted primitives for an example scene.
π^{e} is a 2D feature which can be used to find correspondences in a stereo framework to create 3D primitives (as introduced in [13, 21]) with the following formulation:

$$\Pi^{\mathrm{e}} = (\mathbf{X}, \Theta, \Omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)), \tag{3}$$

where $\mathbf{X}$ is the 3D position; Θ is the 3D orientation; Ω is the phase (i.e., contrast transition); and, $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$) of the 3D primitive.
In this paper, we estimate the 3D representation Π^{m} of monos which stereo fails to compute:

$$\Pi^{\mathrm{m}} = (\mathbf{X}, \mathbf{n}, \mathbf{c}), \tag{4}$$

where $\mathbf{X}$ and $\mathbf{c}$ are as in equation 2, and $\mathbf{n}$ is the orientation (*i.e.*, normal) of the plane that locally represents the mono.

3 Relations between Primitives

Sparse and symbolic nature of primitives allows the following relations to be defined on them. For more information about relations of primitives, see [10].

(a) Input image.

(b) Extracted primitives.

Figure 2: Extracted primitives (b) for the example image in (a).

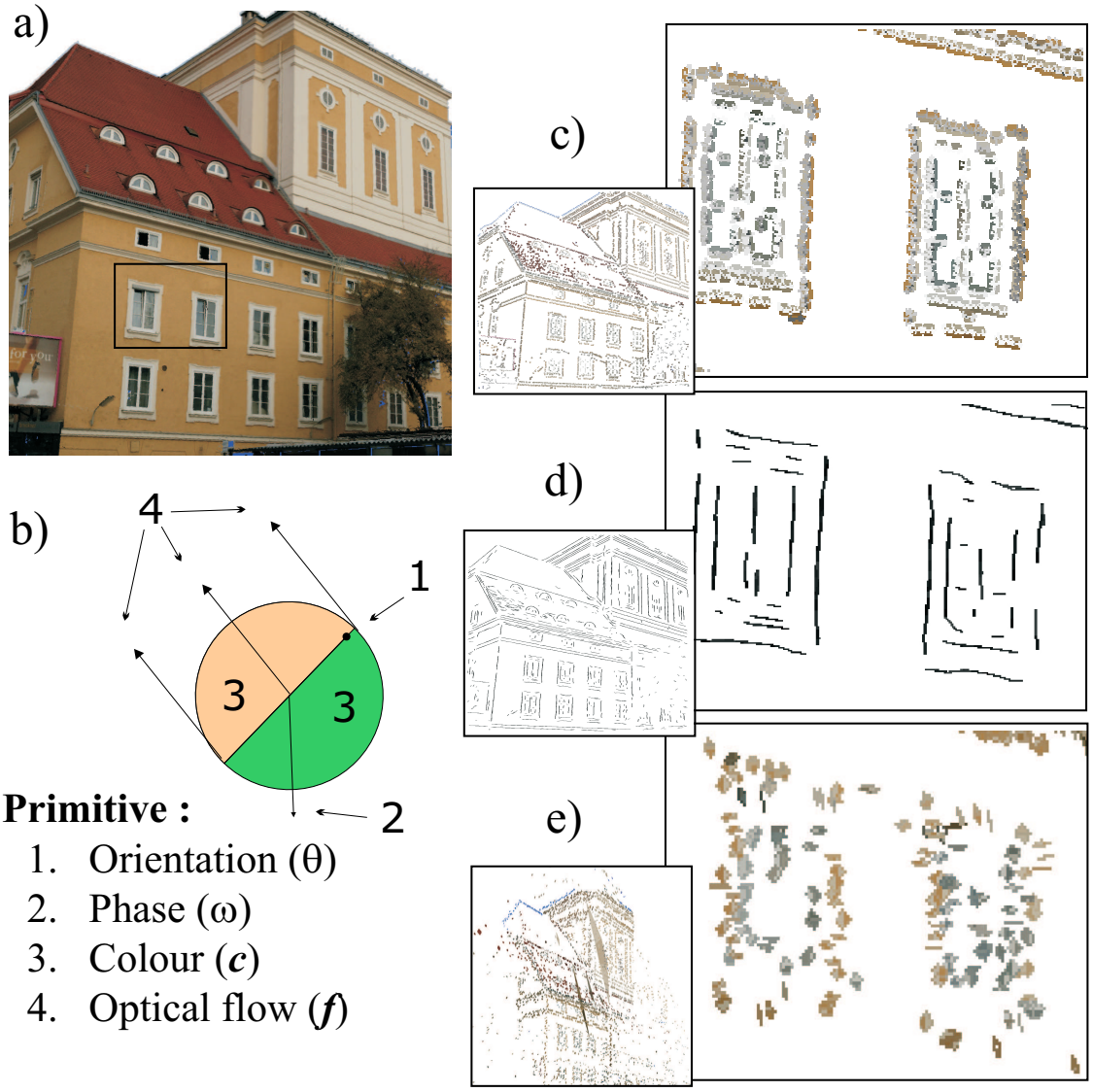


Figure 3: Illustration of the primitive extraction process from a video sequence. The 2D–primitives extracted from the input image (a) (see section 2), and finally the 3D–primitives reconstructed from the stereo–matches as described as described in [21]. **(a)** An example input image. **(b)** A graphic description of the 2D–primitives. **(c)** A magnification of the image representation. **(d)** Perceptual grouping of the primitives as described in [21]. **(e)** The reconstructed 3D entities. Note that the structure reconstructed is quite far from the cameras, leading to a certain imprecision in the reconstruction of the 3D–primitives. A simple scheme addressing this problem is described in [21].

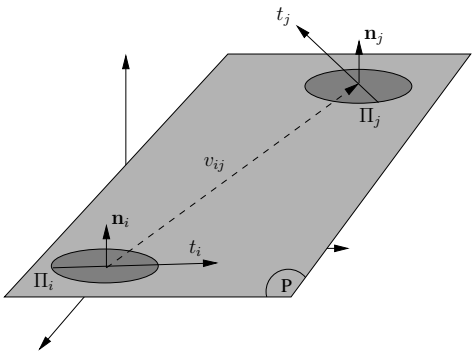


Figure 4: Co–planarity of two 3D primitives Π_i^e and Π_j^e.

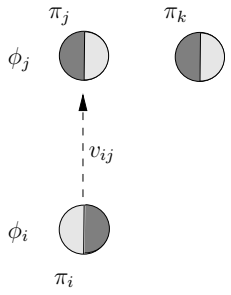


Figure 5: Linear dependence of three $\pi^{\mathrm{e}}{}_i$, $\pi^{\mathrm{e}}{}_j$ and $\pi^{\mathrm{e}}{}_k$. In this example, $\pi^{\mathrm{e}}{}_i$ is linearly dependent with $\pi^{\mathrm{e}}{}_j$ whereas $\pi^{\mathrm{e}}{}_k$ is linearly independent of other primitives.

3.1 Co–planarity

Two 3D edge primitives Π_i^e and Π_j^e are co–planar iff their orientation vectors lie on the same plane, i.e.:

$$cop(\Pi_i^e, \Pi_j^e) = 1 - |\mathbf{proj}_{t_j \times v_{ij}}(t_i \times v_{ij})|, \tag{5}$$

where v_{ij} is defined as the vector $(\mathbf{X}_i - \mathbf{X}_j)$; t_i and t_j denote the vectors defined by the 3D orientations Θ_i and Θ_j, respectively; and, $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$ is defined as:

$$\mathbf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2}\mathbf{u}. \tag{6}$$

The co–planarity relation is illustrated in Fig. 4.

3.2 Linear dependence

Two 3D primitives Π_i^e and Π_j^e are linearly dependent iff the *three* lines which are defined by (1) the 3D orientation of Π_i^e, (2) the 3D orientation of $\Pi^{\mathrm{e}}{}_j$ and (3) v_{ij} are identical. Due to uncertainty in the 3D reconstruction process, in this work, the linear dependence of two spatial primitives Π_i^e and Π_j^e is computed using their 2D projections π_i^e and π_j^e. We define the linear dependence of two 2D primitives π_i^e and π_j^e as:

$$lin(\pi_i^e, \pi_j^e) = |\mathbf{proj}_{v_{ij}} t_i| > Th \ \wedge \ |\mathbf{proj}_{v_{ij}} t_j| > Th, \tag{7}$$

where t_i and t_j are the vectors defined by the orientations θ_i and θ_j, respectively; and, Th is a threshold.

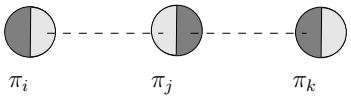


Figure 6: Co–colority of three 2D primitives π_i^e, π_j^e and π_k. In this example, π_i^e and π_j^e are cocolor, so are π_i^e and π_k^e; however, π_j^e and π_k^e are not cocolor.

3.3 Co–colority

Two 3D primitives Π_i^e and Π_j^e are co–color iff their parts that face each other have the same color. In the same way as linear dependence, co–colority of two spatial primitives Π_i^e and Π_j^e is computed using their 2D projections π_i^e and π_j^e. We define the co–colority of two 2D primitives π_i^e and π_j^e as:

$$coc(\pi_i^e, \pi_j^e) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j), \tag{8}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the RGB representation of the colors of the parts of the primitives π_i^e and π_j^e that face each other; and, $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is Euclidean distance between RGB values of the colors $\mathbf{c}_i$ and $\mathbf{c}_j$.
Co-colority between an edge primitive π^{e} and and a mono primitive π^{m}, and between two monos can be defined similarly (not shown here).
In Fig. 6, a pair of co–color and not co–color primitives are shown.

4 Formulation of the model

For the prediction of the depth at monos, we developed a voting model. In a voting model, there are a set of voters that state their *opinion* about a certain event e. A voting model combines these votes in a reasonable way to make a decision about the event e.
In the depth prediction problem, the event e to be voted about is the depth and the 3D orientation of a mono π^{m}, and the voters are the edge primitives $\{\pi_i^e\}$ (for $i = 1, ..., N_E$) that bound the mono. In this paper, we are interested in the predictions of pairs of π_i^es, which are denoted by P_j for $j = 1, ..., N_P$. While forming a pair P_j from two edges π_i^e and π_k^e from the set of the bounding edges of a mono π^{m}, we have the following restrictions:

1. π_i^e and π_k^e should share the same color with the mono π^{m} (*i.e.*, the following relations should hold: $coc(\pi_i^e, \pi_k^e)$ and $coc(\pi_i^e, \pi^{\mathrm{m}})$).

2. The 3D primitives Π_i^e and Π_k^e of π_i^e and π_k^e should be on the same plane (*i.e.*, $cop(\Pi_i^e, \Pi_k^e)$).

3. π_i^e and π_k^e should not be linearly dependent so that they can define only one plane (*i.e.*, $\neg\, lin(\pi_i^e, \pi_k^e)$).

In figure 7, such restrictions are illustrated for an example mono and a set of edge primitives that bound it. The primitives π_j^e and π_m^e are on the same line (*i.e.*, they are linearly dependent), and they define infinitely many planes. As for primitives π_l^e and π_k^e, they cannot define a plane as they are not on the same plane, nor do they share the same color.
The vote v_i by a pair P_j can be parametrized by:

$$v_i = (\mathbf{X}, \mathbf{n}), \tag{9}$$

where $\vec{n}$ is the normal of the mono π^{m}, and z is its depth relative to the plane defined by P_i.

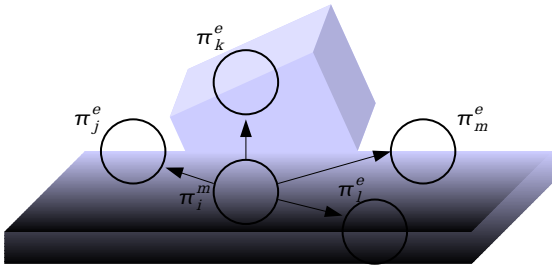


Figure 7: A set of primitives for illustrating why the relations coplanarity, cocolority and linear dependence are required as restrictions for forming pairs from edges.

Each v_i has an associated reliability or probability r_i. They denote how likely the vote is based on the believes of pair P_i. It can be modeled as a function of the distance of the mono π^{m} to the intersection point IP:

$$r_i = f(d(\Pi^{\mathrm{m}}, P_i)). \tag{10}$$

r_i can be weighted by the confidences of the elements of the pair P_i that reflect their quality.

4.1 Bounding edges of a mono

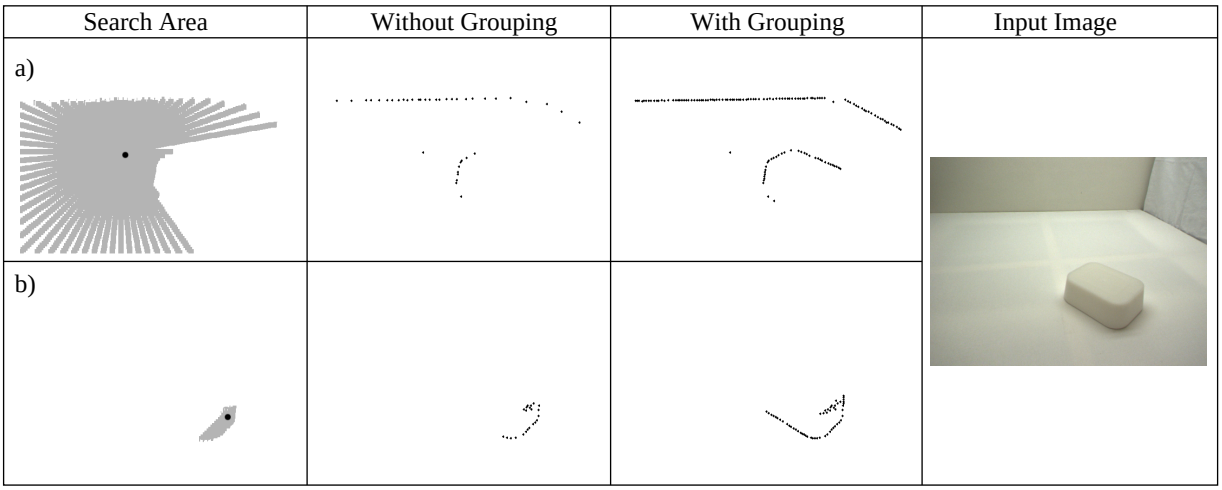


Figure 8: Finding bounding edge primitives with and without grouping information for two different monos which are marked in black in the first column. Using grouping information produces a more complete boundary finding as shown in (a). However, using grouping may include unwanted edge primitives in the boundary as shown in (b).

Finding the bounding edges of a mono π^{m} requires making searches in a set of directions d_i, $i = 1...N_d$ for the edge primitives. In each direction d_i, starting from a minimum distance R_{min}, the search is performed upto a distance of R_{max} in discrete steps s_j, $j = 1...N_s$. If an edge primitive π^{e} is found in direction d_i in the neighborhood Ω of a step s_j, π^{e} is added to the list of bounding edges and the search continues with the next direction.
The above mentioned method for finding the bounding edge primitives will lead to an incomplete and sparse boundary detection (see figure 8) because the search is performed only in a set of discrete directions. This can be improved by making use of the contour grouping information; when an edge primitive π^{e} is found

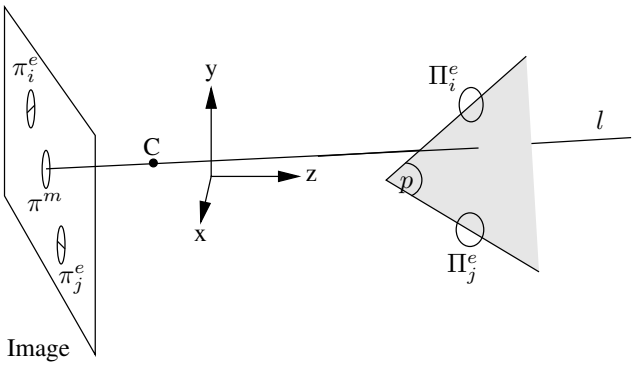


Figure 9: Illustration of how the vote of a pair of edge primitives is computed. The 3D primitives Π_i^e and Π_j^e corresponding to the 2D primitives π_i^e and π_j^e define the plane p. The intersection of p with the ray l that goes through the 2D mono π^{m} and the camera center C then determines the position of the estimated 3D mono Π^{m}. The 3D orientation of Π^{m} is set to be the orientation of the plane p.

in a direction d_i at step s_j, if π^{e} is part of a group G, then all the edge primitives in G can be added to the list of bounding edges (see [21] for information about the grouping method we employ in this paper).
Grouping information can lead to more complete and dense boundary finding as shown in figure 8(a); however, for certain objects, it may lead to worse results due to low contrast edges (see figure 8(b)).

4.2 The vote of a pair of edge primitives on a mono π^{m}

A pair P_i of two edge primitives π_j^e and π_k^e with two corresponding 3D edge primitives Π_j^e and Π_k^e, which are co-planar, co-color and linearly *independent*, defines a plane p with 3D normal $\mathbf{n}$ and position $\mathbf{X}$.
The vote v_l of Π_j^e and Π_k^e is computed by the intersection of the plane p with the ray l that goes through the mono, π^{m}, and the focus of the camera (see figure 9). The ray l is computed using the following formula ([5], pg41):

$$\mathrm{X_a} = P^{-1}(-\tilde{p} + \lambda\tilde{x}), \tag{11}$$

where $\tilde{x}$ is the homogeneous position of π^{m}; P and $\tilde{p}$ are respectively the 3x3 and the 3x1 sub-parts of the 3x4 projection matrix P_m so that $P_m = [P\ \ \tilde{p}]$; and, λ is an arbitrary number. By using two different values for λ, two different points on ray l are extracted which then are used to compute the ray l.
Because the ray l is unique for a mono π^{m}, all the votes processed for the mono π^{m} will be on ray l. This property can be exploited for clustering the votes as discussed in section 4.3

4.3 Combining the votes

The votes can be integrated using different ways to estimate the 3D representation Π^{m} of a 2D mono π^{m}:

- *Weighted averaging:*

$$\Pi^{\mathrm{m}} = C \sum_{i=1}^{N_P} v_i\, r_i, \tag{12}$$

  where C is a normalization constant.

- *Clustering:*
  Weighted averaging is prone to outliers which can be overcome by utilizing the set of clusters in the

votes. Let us denote the clusters by c_i for $i = 1, ...N_c$. Then, one integration scheme would be to take the cluster that has the highest average reliability:

$$\Pi^{\mathrm{m}} = \arg\max_{c_i} \ \frac{1}{\#c_i} \sum_{v_j \in c_i} r_j. \tag{13}$$

where r_i is the reliability (*i.e.*, confidence) associated to the vote v_i.

An alternative can use the most crowded cluster:

$$\Pi^{\mathrm{m}} = \arg\max_{c_i} \ \#c_i. \tag{14}$$

It is also possible to combine the number of votes and the average reliability of a cluster for making a decision.

As mentioned above, weighted averaging is prone to outliers but is fast. Clustering the votes can filter outliers whereas is slow. Moreover, clustering is an ill-posed problem, and most of the time, it is not trivial to determine the number of clusters from the data points that will be clustered.
In this paper, we implemented (1) a histogram-based clustering where the number of bins is fixed, and the best cluster is considered to be the bin with the most number of elements, and (2) a clustering algorithm where the number of clusters is determined automatically by making use of a cluster-regularity measure and maximizing this measure iteratively.
(1) is a simple but fast approach whereas (2) is considerably slower due to the iterative-clustering step. Suprisingly, our investigations showed that (1) and (2) produce almost identical results (the comparative results are not provided in this paper). For this reason, we have adopted (1) as the clustering method for the rest of the paper.

4.4 Combining the predictions using area information

3D surfaces project as areas into 2D images. Although one surface may project as many areas in the 2D image, it can be claimed that the image points in an image area are part of the same 3D surface[SK: This assumption does not always hold. I need to elaborate.].
Figure 10 shows the predictions of a surface. Due to strong outliers in the stereo computation, depth predictions are scattered around the surface that they are supposed to represent. We show that it is possible to segment the 2D image into areas based on intensity similarity and combine the predictions in areas to get a cleaner and more complete surface prediction.
We segment an input image $\boldsymbol{\mathcal{I}}$ into areas A_i, $i = 1, .., N_A$ using co-colority (see section 3) between primitives utilizing a simple region-growing method; the areas are grown until the image boundary or an edge-like primitive is hit. Figure 11 shows the segmentation of one of the images from figure 1.
In this paper, we assume that each A_i has a corresponding surface S_i defined as follows:

$$S_i(x, y, z) = \mathrm{a}x^2 + \mathrm{b}y^2 + \mathrm{c}z^2 + \mathrm{d}xy + \mathrm{e}yz + \mathrm{f}xz + \mathrm{g}x + \mathrm{h}y + \mathrm{i}z = 1. \tag{15}$$

Such a surface model allows a wide range of surfaces to be represented, including spherical, ellipsoid, quadratic, hyperbolic, conic, cylinderic and planar surfaces.
S_i is estimated from the predictions in A_i by solving for the coefficients using a least-squares method. As there are nine coefficients, such a method requires at least nine predictions to be available in area A_i. For the predictions shown in figure 10, the following surface is estimated which is shown in figure 12 using a sparse sampling (only non-zero coefficients are shown):

$$S_0 = 1.5 \times 10^{-5} y^2 + 5 \times 10^{-6} yz - 1.9 \times 10^{-4} x + 8 \times 10^{-3} y + 1.2 \times 10^{3} z = 1. \tag{16}$$

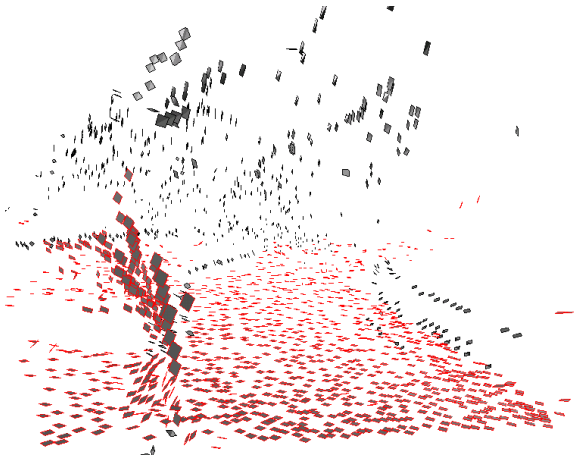


Figure 10: The predictions on the surface of the road for the input images shown in figure 1 (predictions are marked with red boundaries). The predictions are scattered around the plane of the road, and there are wrong predictions due to strong outliers in the computed stereo.

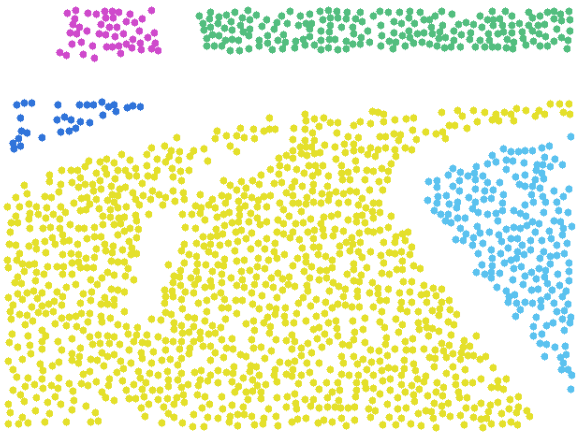


Figure 11: Segmentation of one of the input images given in 1 into areas using region-growing based on primitives.

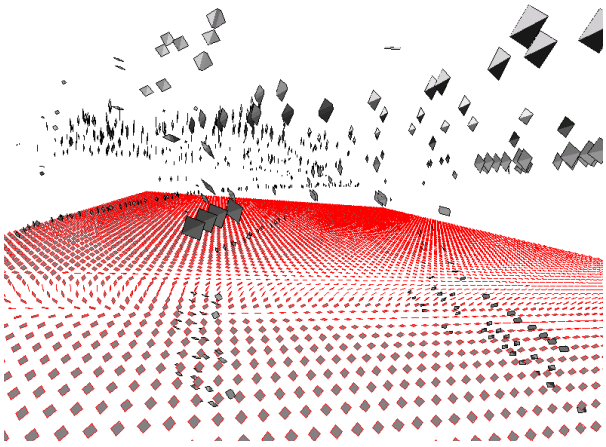


Figure 12: The surface given in equation 16 which is extracted from the predictions shown in figure 10.

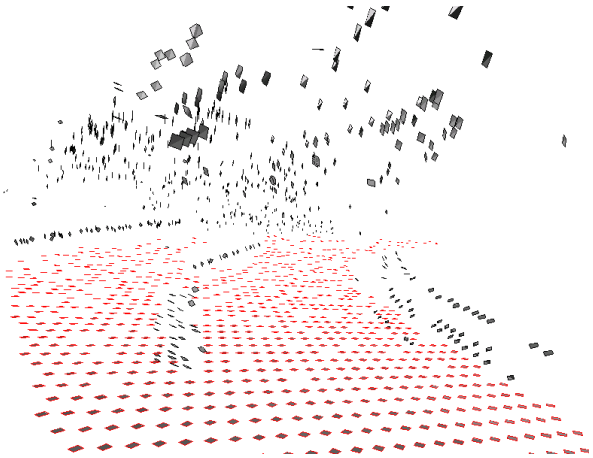


Figure 13: The predictions from 10 that are corrected using the extracted surface S_0 shown in equation 16 and figure 12.

S_0 in equation 16 is mainly a planar surface with small quadratic coefficients caused by outliers.
Having an estimated S_i for an area A_i, it is possible to *correct* the mono predictions using the estimated surface S_i: Let $\mathbf{X}_n$ be the intersection of the surface S_i with the ray that goes through π^{m} and the camera, and $\mathbf{n}_n$ be the surface normal at this point (defined by $\mathbf{n}_n = (\delta S_i/\delta_x, \delta S_i/\delta_y, \delta S_i/\delta_z)$). $\mathbf{X}_n$ and $\mathbf{n}_n$ are respectively the corrected position and the orientation of mono Π^{m}.
Corrected 3D monos for the example scene is shown in figure 13. Comparison with the initial predictions which are shown in figure 10 concludes that (1) outliers are *corrected* with the extracted surface representation, and (2) orientations and positions are qualitatively better.

5 Results

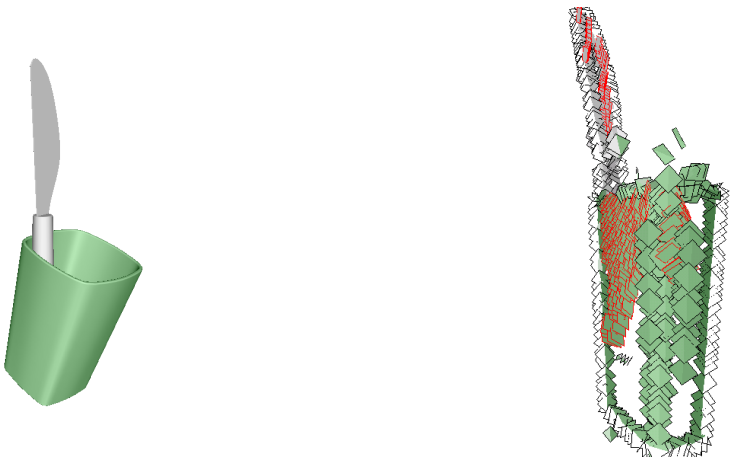
The test cases include kitchen scenarios and road scenarios which are intended for PACO+ and Drivsco projects, respectively. The results of our model is shown for a few examples in figures 14, 15, 16 and 17.
The results show that inspite of limited 3D information from feature-based stereo which may contain strong outliers in some of the scenes (as shown in figure 1), our result is able to predict the surfaces.

6 Conclusion

In this paper, we introduced a voting model that estimates the depth at homogeneous or weakly-textured image patches (called monos) from the depth of the bounding edge-like structures. The depth at edge-like structures is computed using a feature-based stereo algorithm [20], and is used to vote for the depth of a mono, which otherwise is not possible to compute easily due to the correspondence problem.
The method presented in this paper is an ongoing work. In the future, the reliability of each vote will be replaced by the statistics collected from chromatic range data (see [12]). Moreover, comprehensive comparison as well as possible combination with dense stereo methods are going to be investigated.

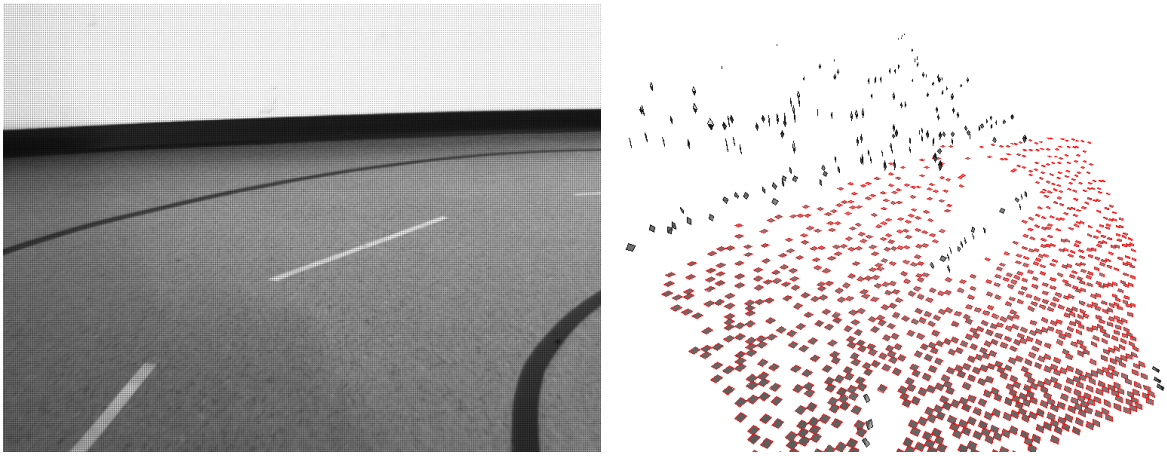
7 Acknowledgments

This work is supported by Drivsco projects.



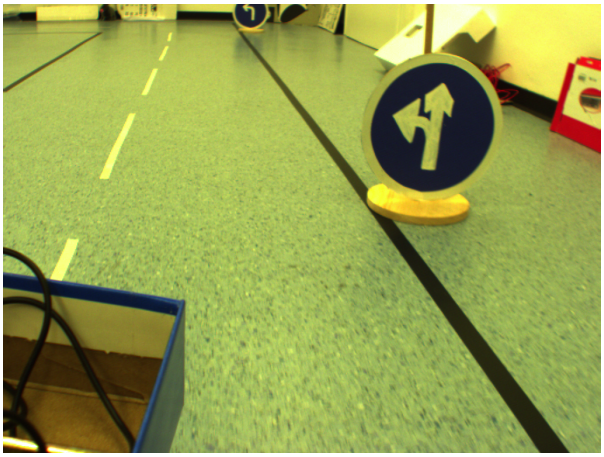
(a) (b)

Figure 14: Experiment results on an artificial *kitchen* scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.



(a) (b)

Figure 15: Experiment results on a road scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.



(a)

(b)

Figure 16: Experiment results on a road scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.

(a)

(b)

Figure 17: Experiment results on a *kitchen* scene. **(a)** Left image of the input stereo pair. **(b)** The predictions of our model.

References

[1] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.

[2] A. B.L., S. M., and F. R.W. The interpolation of object and surface structure. *Cognitive Psychology*, 44:148–190(43), March 2002.

[3] V. Bruce, P. R. Green, and M. A. Georgeson. *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, 4th edition, 2003.

[4] T. S. Collett. Extrapolating and Interpolating Surfaces in Depth. *Royal Society of London Proceedings Series B*, 224:43–56, Mar. 1985.

[5] O. Faugeras. *Three–Dimensional Computer Vision*. MIT Press, 1993.

[6] W. E. L. Grimson. A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427, Sept. 1982.

[7] W. E. L. Grimson. Binocular shading and visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 28(1):19–43, 1984.

[8] G. Guy and G. Medioni. Inference of surfaces from sparse 3-d points. In *ARPA94*, pages II:1487–1494, 1994.

[9] D. Hubel and T. Wiesel. Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750, 1969.

[10] S. Kalkan, N. Pugeault, M. Christiansen, and N. Krüger. Relations between primitives. Technical report, University of Southern Denmark, 2006. (to be submitted).

[11] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2006.

[12] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.

[13] N. Krüger and M. Felsberg. An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863, 2004.

[14] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.

[15] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 2007-4, 2007.

[16] N. Krüger, N. Pugeault, and F. Wörgötter. Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. *To be submitted.*, 2007.

[17] M. S. Lee and G. Medioni. Inferring segmented surface description from stereo data. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 346, Washington, DC, USA, 1998. IEEE Computer Society.

[18] M.-S. Lee, G. Medioni, and P. Mordohai. Inference of segmented overlapping surfaces from binocular stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):824–837, 2002.

[19] V. S. Nalwa. Line-drawing interpretation: Bilateral symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(10):1117–1120, 1989.

[20] N. Pugeault and N. Krüger. Multi–modal matching applied to stereo. *Proceedings of the BMVC 2003*, pages 271–280, 2003.

[21] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.

[22] K. A. Stevens. The visual interpretations of surface contours. *Artificial Intelligence*, 17:47–73, 1981.

[23] D. Terzopoulos. Multi-level reconstruction of visual surfaces: Variational principles and finite element representations. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1982.

[24] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):417–438, 1988.

[25] F. Ulupinar and R. Nevatia. Constraints for interpretation of line drawings under perspective projection. *CVGIP: Image Underst.*, 53(1):88–96, 1991.

[26] F. Ulupinar and R. Nevatia. Perception of 3-d surfaces from 2-d contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):3–18, 1993.

Robotics Group
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Technical Report no. 2007 – 4

Multi-modal Primitives: Local, Condensed, and semantically rich visual Descriptors and the Formalisation of contextual Information

Norbert Krüger, Nicolas Pugeault and Florentin Wörgötter

January 23, 2007

Title	Multi-modal Primitives: Local, Condensed, and semantically rich visual Descriptors and the Formalisation of contextual Information
Author(s)	Norbert Krüger, Nicolas Pugeault and Florentin Wörgötter
Publication History	

1 Introduction

There exists a large amount of evidence that the human visual system in its first cortical stages processes a number of aspects of visual data (see, e.g., [19, 39]). These aspects, in the following called visual modalities, cover, e.g., local orientation [19, 20], colour [20], junction structures [46], stereo [3] and optic flow [20]. At the first stage of visual processing (called 'early vision' in [29]), these modalities are computed locally for a certain retinal position. At a later stage (called 'early cognitive vision' in [29]), results of local processing become integrated with the spatial and temporal context. Computer vision has dealt to a large extent with these single modalities and in many computer vision systems, one or more of the above–mentioned aspects are processed in the first stages (see, e.g., [35, 45, 33]).
An important problem, the human visual system as well as any artificial visual system has to cope with, is the high degree of ambiguity and noise in these low level modalities that is unresolvable by local processes only. Reliable actions require a more stable representation of visual features. As a consequence, a disambiguation process that makes use of contextual information is needed. In [32] we have described two main regularities in visual data (that are also well recognised in the computer vision community) that underlie such an disambiguation process: (i) Coherent motion of rigid bodies and (ii) statistical interdependencies underlying most grouping processes. These two regularities allow to make predictions between locally extracted visual events and thereby to verify the spatio–temporal coherence of hypotheses.
The establishment of such a disambiguation process presupposes communication of temporal and spatial information. An efficient condensation of the locally extracted information implies:

Property 1. *The condensed information vector should allow for rich predictions between related (e.g. the change of position and appearance of a local patch under a rigid body motion) visual events;*

and

Property 2. *The condensed information vector need to reduce the dimensionality of the local signal to allow the process to work with limited bandwidth.*

In [25] it is argued that the need for properties 1 and 2 naturally result in symbolic representations. In this work, we present a novel kind of scene representation based on local symbolic descriptors that we call visual primitives (see figure 1).[1] In these primitives different visual modalities become combined in one local feature descriptor (section 2 and 3) that allows for the representation of visual scenes in a condensed way (satisfying property 2).
Furthermore, the primitives allow for rich predictions (property 2) since we can formulate efficiently statistical dependencies as operating in most perceptual grouping mechanisms as well as the change of image structure under a coherent motion (see section 5). Hence, locally computed primitives work as first guesses in a disambiguation process that is described in [41].
Our scene representation based on multi–modal primitives addresses a number of issues in an original way:

Multi–modality: Primitives cover the main visual modalities established in computer– and human vision and, hence, carry a rich semantic interpretation that facilitates the disambiguation process.

Condensation: Although primitives reduce the dimensionality of the image data, the significant aspects of image information are kept. For example, by using the primitives, we were able to achieve a stereo matching performance similar to correlation based methods that use the full image information (see [28]).

[1] A possible biological equivalent of the primitives are so called hyper–columns in the visual cortex (for a discussion, see [30]).

Dynamic Positioning and Completeness: The primitives semantically describe the image information in terms that are meaningful for image and scene understanding. This is achieved by dynamic search for primitives position resulting in localised symbolic descriptors that preserve a complete representation of structures. Namely, we have a description of contours, corners and surfaces and their mutual relations. We will show that in the case of contours this semantic extends naturally to 3D space (see 4.1).

Different Experts for different Structures: The interpretation of the local signal by the primitives is not static but depends on the intrinsic signal structure leading to a system of different experts for different signal structures such as edges, lines, homogeneous patches and corners (as also established in the human system).

Primitives Initialise Disambiguation: The primitives are not understood as a final statement about the local structure of a scene but a confidence associated to each primitive as well as its parameters as well become modified in disambiguation processes formalising contextual information. This paper is the first technical description of our visual primitives that have been applied already in various contexts (see, e.g., [31, 28, 22]). The primitives make use of a rather complex body of signal processing methods associated to the different visual modalities. Some of these aspects have been published earlier (such as e.g., the monogenic signal [14], a continuous concept of intrinsic dimension [27]) and are described briefly in this paper to make the presentation self-contained.
The system processes information over multiple stages (for an overview see figure 1) described in the following sections. In section 2, we will describe the processing of the individual modalities by linear and non–linear filtering processes. In section 3, we describe the condensation process generating primitives. In section 4, stereo–pairs of primitives are used to reconstruct information about the scene structure into *3D–primitives*. In section 5, we briefly describe the application of our primitives in an early cognitive architecture integrating perceptual grouping and motion as well as in the context of vision based robotics. A more detailed description the application of the primitive representation resulting in reliable and precise scene representations is given in [41].

2 Analysis of the local Signal Structure

In section 2.1 we will first describe how we distinguish different kinds of local image structures. The processing of the modalities orientation, phase and optic flow is then described in section 2.2 and 2.3. The results of the process described in this section are illustrated in a compact way in figure 1b).

2.1 Intrinsic Dimension

Different kinds of image structures coexist in natural images: homogeneous image patches, edges, corners, textures. Furthermore, certain concepts are only meaningful for specific classes of image structures. For example, the concept of orientation is well defined for edges or lines but not for junctions, homogeneous image patches or for most textures.
As another example, the concept of position is different for a junction as compared to an edge or an homogeneous image patch — see figure 2. a) in homogeneous areas of the image no particular location can be defined, and therefore an equidistant sampling is appropriate. b) For a line or edge structure the position can be defined using energy maxima. However, because of the aperture problem, this energy maxima will span a one–dimensional manifold, and therefore the feature can be localised only up to this manifold. This result in a fundamental ambiguity in the localisation of edge/line local features. c) At the contrary, the locus of a junction can be unambiguously defined by the point of line intersection (see figure 2c).
Similar considerations are required for other modalities such as colour, optic flow and stereo (see below).

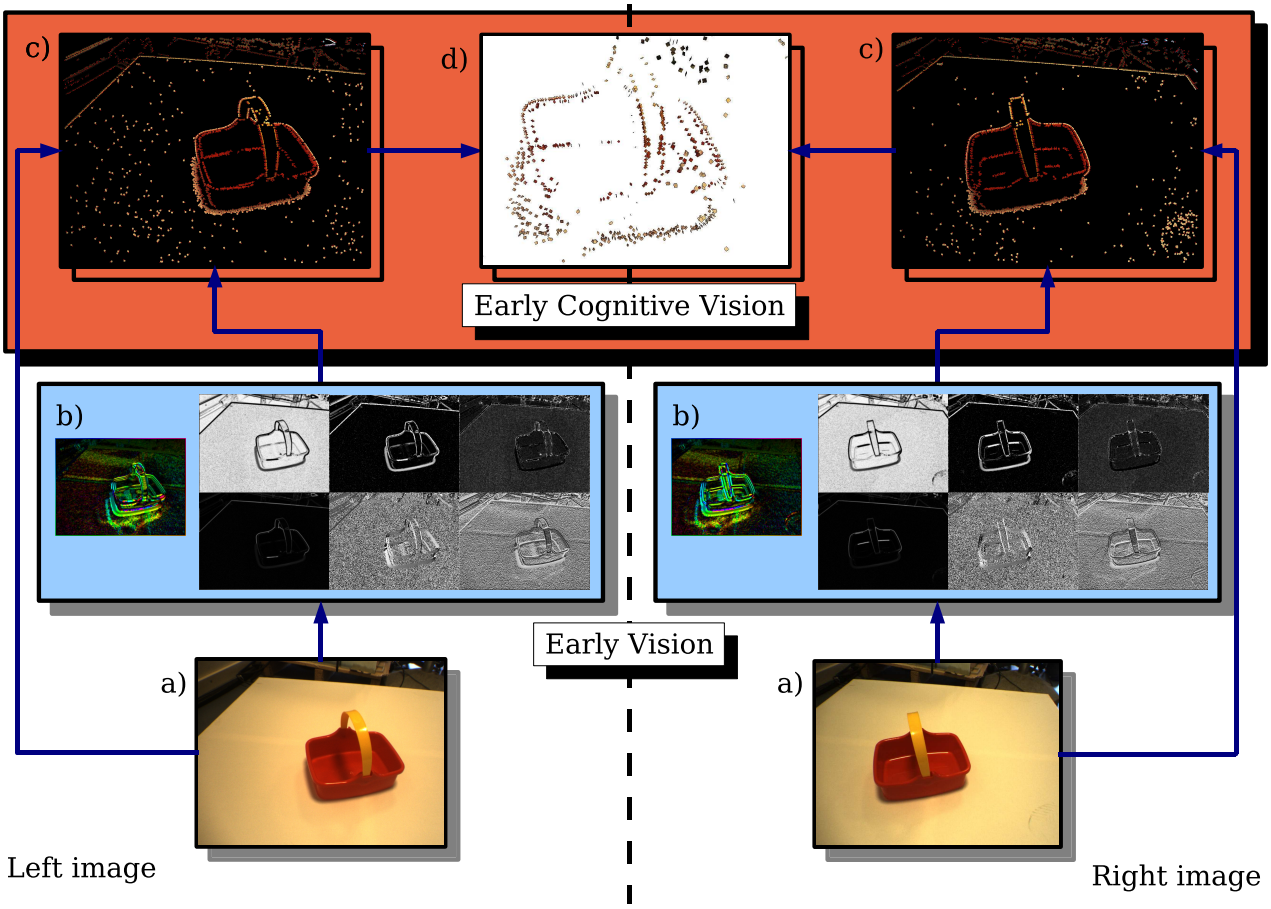


Figure 1: Overview of the primitive extraction scheme. **a)** a stereo–pair of images obtained from a pre–calibrated stereo rig. Therefrom, Early Vision processes are computed as shown in **b)**: the left image shows the optical flow extracted using the — see section 2.3. The hue of the pixels indicate that the orientation of the optic flow at this pixel is towards the margin of similar hue, and the intensity illustrate the magnitude of the flow vector); the bottom row of images shows the magnitude, orientation and phase of the signal — see section 2.2— from left to right respectively; The upper row shows the i0D, i1D and i2D confidences — see section 2.1 — from left to right respectively. In all those graphs the intensity encodes the strength of the filter response (white for high, black for low). In **c)** the information from the Early Vision module is combined in a sparse, condensed way into the Early Cognitive Vision module — see section 3. The image shows the primitives extracted from the images shown in a) **d)** these primitives are then matched across the two stereo-views and the correspondences thereof allows to reconstruct 3D–primitives, that extend naturally the primitive information to 3D space — see section 4.

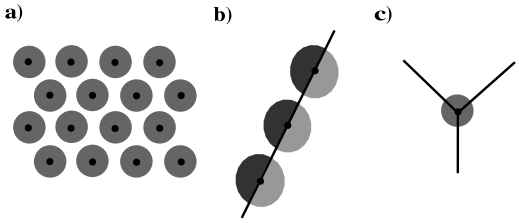


Figure 2: the different localisation problems faced by the different classes of image structure: a) homogeneous area; b) edge or line; and c) junction (see text).

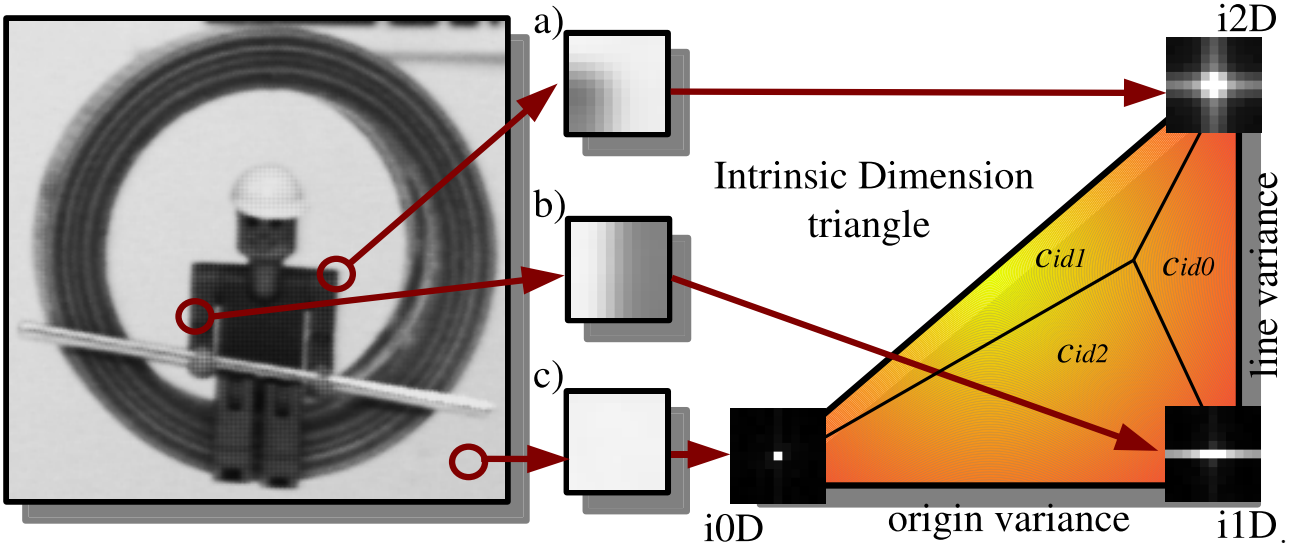


Figure 3: Illustration of the triangular topology of the intrinsic dimension — see [11]

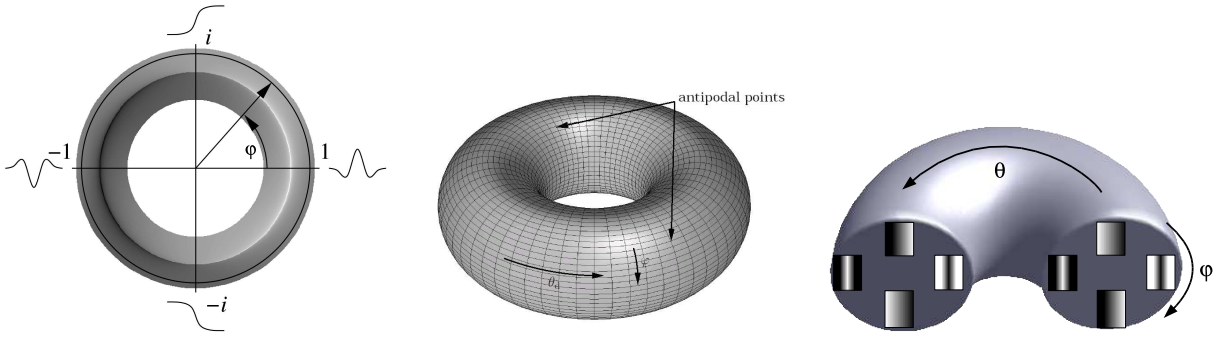
Hence, before applying concepts such as orientation or position, we need to classify image patches according to their junction–ness, edge–ness or homogeneous–ness. The intrinsic dimension (see, e.g., [51, 10]) has proved to be a suitable classifier in this context [11]. Ideal homogeneous image patches have an intrinsic dimension of zero (i0D), ideal edges are intrinsically 1–dimensional (i1D) while junctions and most textures have an intrinsic dimension of two (i2D). Going beyond common discrete classification [51, 21], we utilise a *continuous* concept [11, 12, 27] that allows for a formulation of reasonable confidences for the different image structure classes.

We classify image patches according to the dimension of the subspace that is occupied by the local spectral energy. When looking at the spectral representation of a local image patch (see figure 3), we see that the spectral energy of an intrinsically zero-dimensional signal is concentrated in the origin (figure 3a), whereas the energy of an intrinsically one-dimensional signal spans a line (figure 3b) and the energy of an intrinsically two-dimensional signal varies in more than one dimension (figure 3c).

It has been shown [27, 12] that the topological structure of the intrinsic dimensionality must be understood as a triangle that is spanned by two measures: origin variance and line variance. The origin variance describes the deviation of the energy from a concentration at the origin whereas the line variance describes the deviation from a line structure (see figure 3). We define the intrinsic dimension triangle such that each vertex corresponds to one ideal case of intrinsic dimension (homogeneous, linear or corner), and that its surface represents image patches that contains mixed aspects from these three ideal classes. It was shown in [11, 27, 12], that such a triangular interpretation allows for a *continuous formulation* of intrinsic dimensionality, parametrised by 3 confidences that are assigned to each of the mutually exclusive intrinsic dimension classes. For any image patch, the origin and line variances yield a point in this intrinsic dimension triangle (see figure 3d) and the confidence for this patch to belong to each of the three classes is computed using barycentric coordinates (see, e.g., [5]); namely, the confidence in a local patch to be of one of the classes (i0D, i1D or i2D) is the area of the sub–triangle defined by the origin and line variance of the patch, and by the ideal cases for the two other classes of intrinsic dimension — see figure 3.

Thus we compute for each pixel position $\boldsymbol{x}$ the three confidences $c_{id0}(\boldsymbol{x})$,$c_{id1}(\boldsymbol{x})$,$c_{id2}(\boldsymbol{x})$ that take values in $[0, 1]$ and add up to one — illustrated for different scales in the three bottom rows of figure 5. For details of the computation we refer to [11, 27, 12], and to [22, 23] for some applications of this concept.

The current version of our system focuses on intrinsically one dimensional signals and uses the triangular representation defined above to discard non–edge/non–line structures. There is some ongoing work on the integration of homogeneous (iD0) and corner structures (iD2) into this framework — see, [23, 50].



(a) phase (b) direction/phase space (c) orientation/phase space

Figure 4: a) The phase describes different intensity transitions, e.g., $\varphi = \pi$ encodes a dark line on bright background, $\varphi = -\pi/2$ encodes a bright/dark edge, $\varphi = 0$ encodes a bright line on a dark background and $\varphi = \pi/2$ encodes a dark/bright edge. The phase embed these distinct cases into a 2π-periodic continuum shown in (a). [Acknowledgement: Michael Felsberg] b) The torus topology of the orientation–phase space. The phase φ value is mapped on the cross section of the torus' tube whereas the orientation θ is maps to the revolution angle the torus. c) When direction is constrained to orientation (i.e. to the interval $[0, \pi)$) we get a half torus that is connected as indicated by the connecting strings.

2.2 Orientation and Phase

The extraction of a primitive starts with a rotation invariant quadrature filter that performs a *split of identity* of the signal [14]: it decomposes an intrinsically one-dimensional signal (as defined in the previous section) into local amplitude (see figure 5 top row), orientation (see figure 5 second row), and phase (symmetry, see figure 5 third row) information.[2]

The local amplitude is an indicator of the likelihood for the presence of an image structure. The orientation encodes the geometric information of the local signal while the phase can be used to differentiate between different image structures ignoring orientation differences. The phase for possible grey level structures forms a continuum between $[-\pi, \pi)$ and encodes the grey level transition of the local image patch across the edge (as defined by the orientation) in a compact way (as one parameter only), e.g., a pixel positioned on a bright line on a dark background has a phase of 0 whereas a pixel positioned on a bright/dark edge has a phase of $-\pi/2$ — see figure 4a and, e.g., [16, 26, 14]).

Note that phase is 2π-periodic and continuous such that a phase of $-\pi$ designate the same contrast transition as a phase of π.

Orientation θ (taking values in the the interval $[0, \pi)$) and phase φ are topologically organised on a half torus (see figure 4c), and if we extend the concept of orientation to that of a direction (therefore taking values in $[-\pi, \pi)$, see also [21]) then the topology of the direction/phase space becomes a complete torus (see figure 4b). On a local level the direction is not decidable[3] therefore we will use the half torus topology. This topology is crucial for the definition of suitable metrics for phase and orientation. For example, a black/white step edge ($\varphi = \pi/2$) with orientation θ should have small metrical distance to a white/black step edge ($\varphi = -\pi/2$) of orientation $\pi - \theta$ but large distance to a black/white step edge of orientation $\pi - \theta$. However, a white line on a black background with an orientation θ ($\varphi = 0$) should be have only a small distance to a white line on a black background with an orientation $\pi - \theta$ but a large one to any black line on a white background. Therefore the extremities of the half-torus are linked in a continuous manner as is shown in figure 4c. For a discussion of the orientation/phase metric we refer to [28, 40].

Figure 5 shows the filter responses in terms of the local amplitude $m(\boldsymbol{x})$, orientation $\theta(\boldsymbol{x})$ and phase

[2]Note that amplitude, orientation and phase can be analogously computed by Gabor wavelets or steerable filters and that our representation does not depend on the filter introduced in [14]. For a discussion of different approaches to define harmonic filters as well as their advantages and problems we refer to [43].

[3]Even taking the context into account there exists always two global solutions [16].

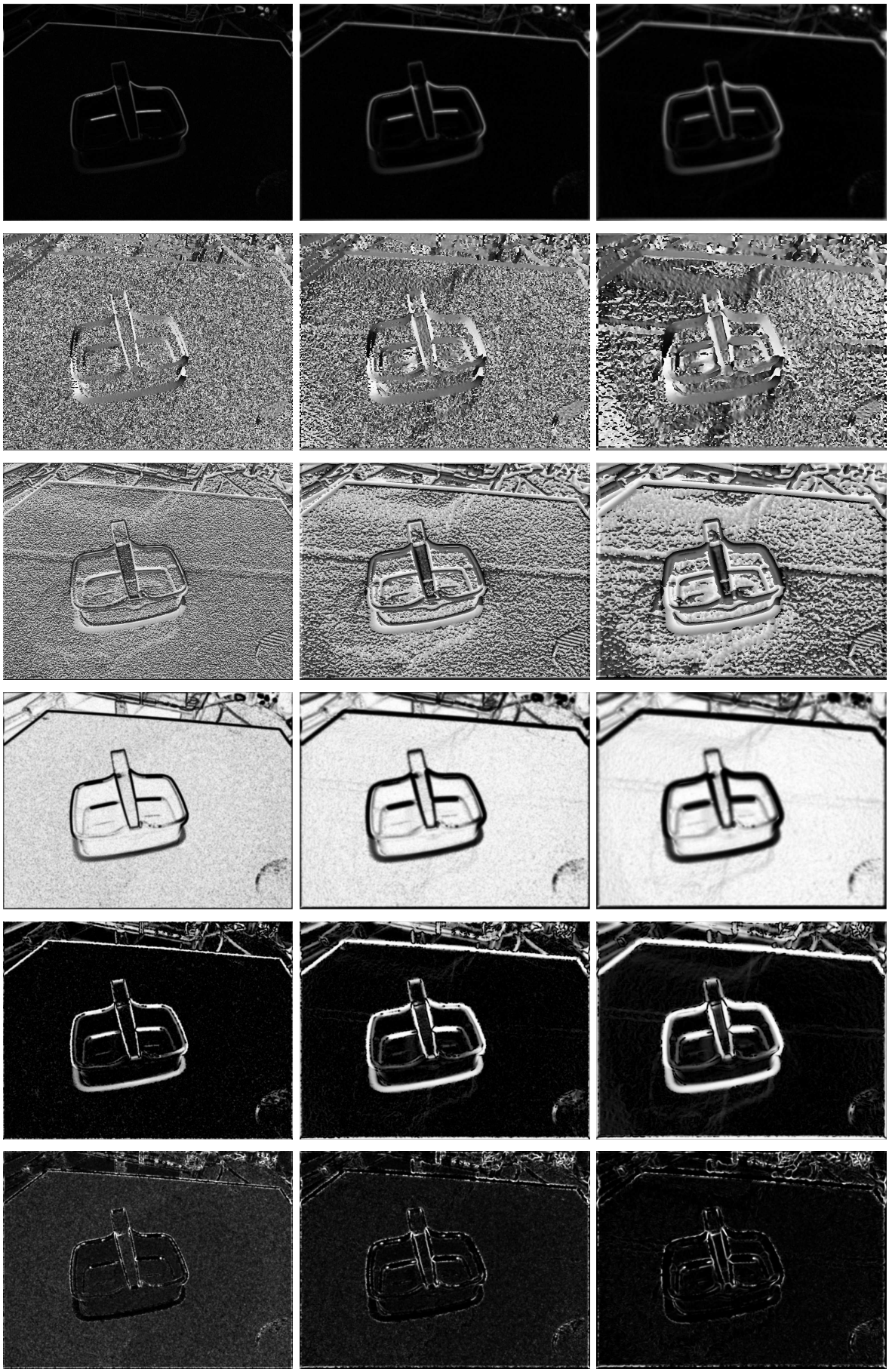


Figure 5: Illustration of the low–level processing for primitive extraction. Each column shows the filter response for a different peak frequency: respectively 0.110 (left), 0.055 (middle) and 0.027 (right). Each row show a response maps for, respectively from top to bottom, local amplitude,orientation, phase, intrinsically Zero–Dimensional (i0D), One–Dimensional (i1D) and Two–Dimensional (i2D) confidences. In all of those graphs white stands for high response and black for low ones.

$\varphi(\boldsymbol{x})$, alongside the resulting primitives, for three scales. The mathematical definition of the kernels and the split of identity is decried in appendix A.
The application of such a spherical quadrature filter for the processing of our Primitives has two main advantages:[4]

1) It allows us to utilise general advantages of the analytic signal (the aforementioned split of identity, see [16]). Hence, phase is an immediate output of the spherical quadrature filter processing and can directly be used as an attribute that describes the structural information of an oriented image structure (see figure 4A).

2) Compared to the use of a Gabor wavelet transform (see, e.g., [6]) we do not need to sample across different orientations but orientation is a direct output of the computation. Hence, we only need to apply 3 filter operations compared to, e.g., 16 for Gabor wavelets (see, e.g., [33]).

We compute filter responses for three different scales (the three scales used in the present work are described in appendix A).[5]

2.3 Optic Flow and Colour

Besides orientation, phase and the intrinsic dimensionality confidences, colout and the local optic flow vector is also associated to the primitive description vector.
In [22], we compared the performance of different optic flow algorithms depending on the intrinsic dimensionality, i.e., the effect of the aperture problem and the quality on low contrast structures. It appeared that different optic flow algorithms might be optimal in different contexts. In our system we primarily use the Nagel–Enkelmann algorithm [38] since it gives stable estimates of the normal flow at i1D structures. We denote the optic flow computed at a position $\boldsymbol{x}$ by $\boldsymbol{f}(\boldsymbol{x})$.
Colour is not processed by (non–)linear filtering operations but sampled (i) on each side of a step edge, or (ii) on each side of a line and on the line itself, depending if the phase describes a step edge or line structure.

3 Condensation Scheme

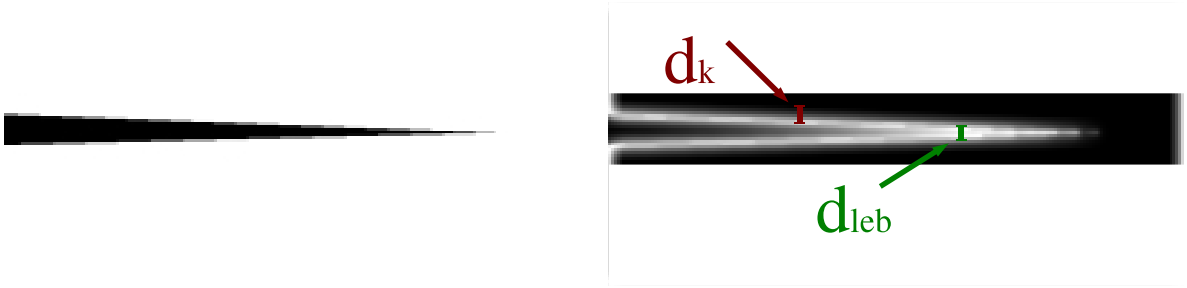
Based on the pixel–wise processing described in section 2, we now want to extract a condensed interpretation of a local image patch by selecting a sparse set of points to which visual modalities become associated. An important aspect of the condensation scheme is that all main parameters can be derived from one property of the basic filter operations called *line–edge bifurcation distance*. This value expresses the minimal distance between two edges for them to be represented by two distinct primitives. Below this distance, one single line primitive will be extracted. In 6(a) shows a narrow triangle for which two edges get closer until the vertex. Vertical sections of the local local amplitude (b) close to the vertex features only one maximum, whereas it splits into two distinct maxima further on, where the triangle is broader.

Definition. *The line–edge bifurcation distance d_{leb} for a given scale is the minimal distance between two edges for them to produces two distinct maxima.*

Using the above definition we propose a condensation procedure in three steps:

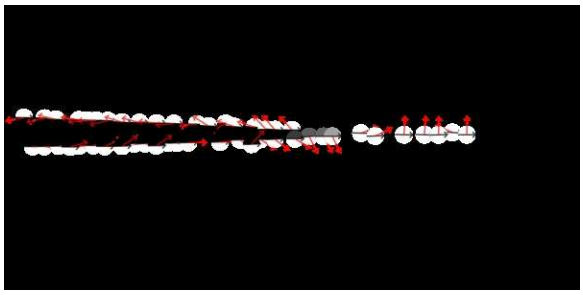
[4]Note that there are also some problems involved with filters realising the monogenic signal we are suing. These are discussed in [43]. First, it turned out that for the monogenic signal it is more difficult to construct filter which allow for stable orientation and phase estimates at high frequencies (compared to, e.g., Gabor wavelets) Second, in the monogenic filter approach there is only one orientation estimate and one phase (in connection to the one orientation) estimate. However, for intrinsically two dimensional signals such as corners and most textures more parameters are needed to represent the local structure (e.g., most textures are characterised by multiple orientations at different frequencies). Third, estimates for, e.g., optic flow can profit from averaging processes over estimates over different orientations. However, in the context of intrinsically one dimensional structures the monogenic signal allows for a good representation.

[5]Note that for step edges, we can expect high amplitudes over different frequency levels, while line structures might become represented at a high frequency level as two step-edges and on a lower frequency level as a line (see section 3).

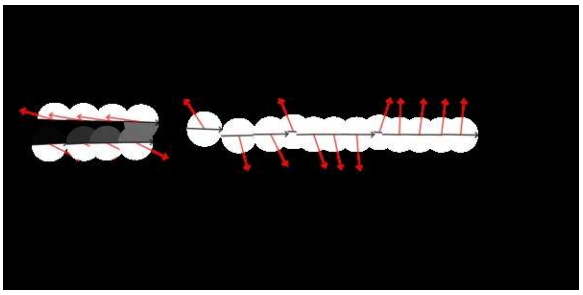


(a) original image

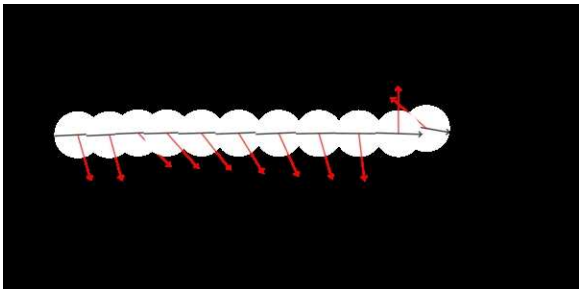
(b) local amplitude



(c) peak frequency 0.110



(d) peak frequency 0.055



(e) peak frequency 0.027

Figure 6: Definition of the elimination parameters d_{leb} and d_k. See text for an explanation.

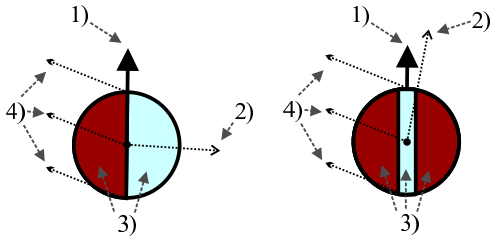
Sampling: The positions of features are computed with sub–pixel accuracy, according to the local intrinsic structure (section 3.1).

Elimination: Positions that are too close to each other (and therefore would lead to redundant descriptors) become deleted (section 3.2).

Local Interpretation: Semantic attributes become associated to the computed positions. (section 3.3).
Figure 6 (c), (d) and (e) shows the primitives extracted after condensation for the three scales used in the present paper — for peak frequencies of 0.11, 0.055 and 0.027, respectively.

3.1 Sampling

In section 2.1 it was discussed that the concept of position is different for different type of image structures as defined by the three classes of intrinsic dimensionality.
The coding of intrinsic dimension by three values $c_{i0D}, c_{i1D}, c_{i2D}$ allows us to select the most likely structure for this patch, and thence to define an appropriate (according to its intrinsic dimension interpretation) position candidate. However, if we do not want to make a decision about the type of local image structure at such an early stage we can also code the three different candidates according to their intrinsic dimension class (see figure 8b). These two approaches are implemented by two different modes of the condensation algorithm with different advantages and disadvantages (see below).



(a) An edge primitive (b) A line primitive

Figure 7: Illustration of the symbolic representation of a primitive for a i1D interpretation, for a) a bright-to-dark step–edge (phase $\varphi \neq 0$) and b) a bright line on dark background (phase $\varphi \neq \frac{\pi}{2}$. 1) represents the orientation of the primitive, 2) the phase, 3) the colour and 4) the optic flow.

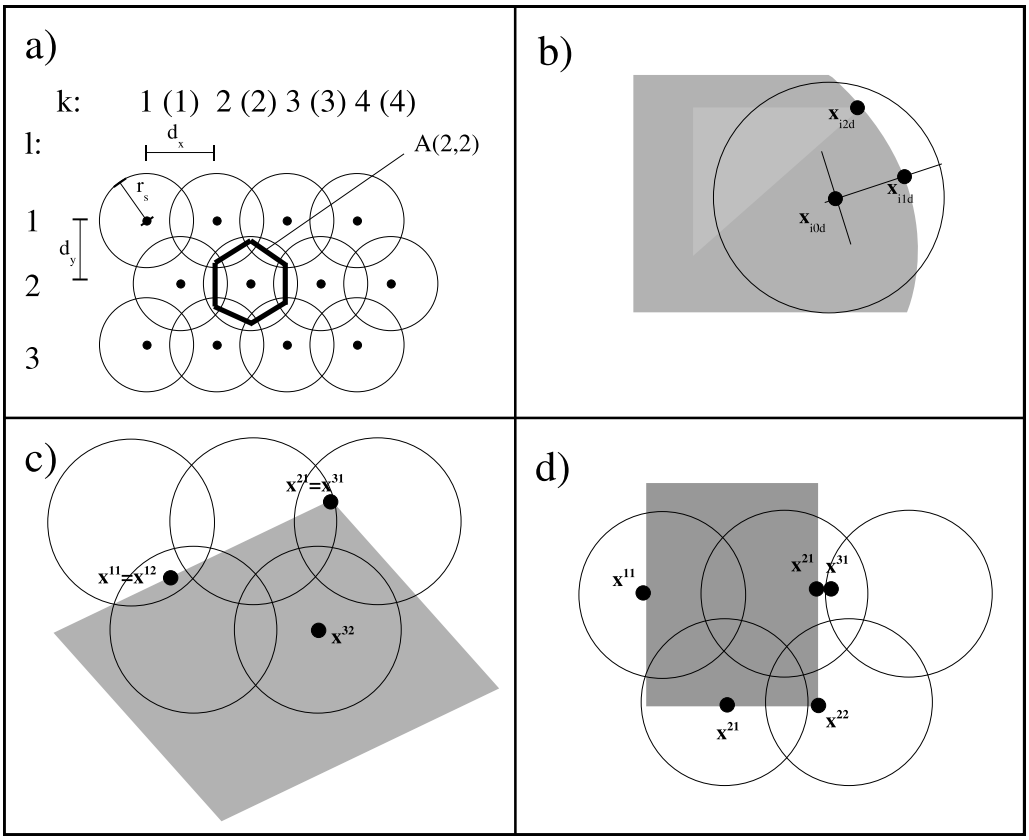


Figure 8: a) Hexagonal Sampling. b) Three possible hypotheses for positions according to the three different intrinsic dimensions. c) Because of the overlap in the hexagonal sampling the same position can be found in areas with different index. For these redundant structures one sample needs to be deleted. d) Since the local amplitude can still be high for pixels with a certain distance from high contrast structure it might be that a position is found that is actually not on the edge structure. These points represent redundant structures since they are already represented more accurately (in terms of position) by other primitives. These hypotheses need also to be deleted.

Figure 9: Top row: positions associated to the primitives assuming different intrinsic dimensionality (from left to right, i0D, i1D and i2D). Middle row: Primitives in each of those cases (from left to right, i0D, i1D and i2D). Bottom row, left: positions using the interpretation given by the intrinsic dimension with the highest confidence; middle: primitives extracted at those locations; and right: primitives at non-i0D locations.

Peak frequency	f_p	0.1103	0.0551	0.0275
Wavelength	f_p	9.06	18.12	36.25
Number of tabs	n_t	11	23	33
Line/edge bifurcation	d_{leb}	3	6	7.5
Hex. grid spacing in x	$d_x = 0.85 d_{leb}$	2.55	5.1	6.37
Hex. grid spacing in y	$d_y = \sqrt{3}/2 d_x$	2.21	4.42	5.52
2^{nd} elimination param.	$d_k = 2.2 d_{leb}$	6.6	13.2	16.5
Condensation rate	d_{co}	85%	94%	97%

Table 1: Frequency–dependent parameters

To get candidates for our primitives, we first perform an hexagonal sampling (see figure 8a) of the image into overlapping areas $A^{(k,l)}$ with radius r_s, with k, l coding the hexagonal grid points. Hexagonal sampling has a number of advantages discussed for example in [47, 37].[6] In the context of this paper, the most important difference to a rectangular sampling is that in case of hexagonal tiles the distance between the midpoints of neighbour tiles is uniform whereas in a rectangular grid diagonal neighbours are are $\sqrt{2}$ times further than horizontal or vertical neighbours. Since we want to extract symbolic descriptors for each tile, the hexagonal sampling allows for a more evenly distributed symbolic description and also reflects more closely the isotropic structure of the original image filters.

The sampling distance depends on the *line/edge bifurcation distance* and thereby on the peak frequency for the scale being used (note that it is also related to the spatial size, and the minimal number of tabs n_t, needed to represent the filter, for a detailed discussion see [43]). The parameters d_x and $d_y = \frac{\sqrt{3}}{2} d_x$ determine the spatial distance in x and y between the centre $A_c^{(k,l)}$ of the tile $A^{(k,l)}$ and the centres of the neighbour tiles.[7] For a description of the mathematics of the hexagonal sampling we refer to, e.g., [37].

The sampling distance d_x is related to the line/edge bifurcation distance d_{leb} that depends of the peak frequency f_p and the band-width B of the filter applied.

In appendix A we describe the derivation of the kernels of the monogenic signal which bandpass characteristics are controlled by the two parameters s_1 and s_2. The peak frequency is computed by

$$f_p = \frac{1}{2\pi(s_2 - s_1)} ln(\frac{s_2}{s_1}) \tag{1}$$

Since in our case we have $s_2 = 2s_1$ this becomes

$$f_p = \frac{1}{2\pi s_1} ln(2) \tag{2}$$

with s_1 set to 1, 2, and 4 covering the frequency domain in a reasonable way (see figure 19).

It turned out that a reasonable estimate for d_{leb} is

$$d_{leb} = \frac{1}{3 f_p} \tag{3}$$

hence we set

$$d_x = round(d_{leb}) + 1 \tag{4}$$

being the smallest possible sampling distance within which structures based on the amplitude information can be resolved. All frequency depended parameters are shown in table 1:

We search on disk around each $A_c^{(k,l)}$ for candidate positions of primitives. The radius r_s of this disk is chosen such that each point of the image is covered by at least one of the disks. In a hexagonal

[6] For example Mersereau [36] showed that hexagonal sampling is optimal for certain band limited signals.

[7] Note that the odd rows have an onset of $d_x/2$

grid, the maximum distance to the border of a tile is $\frac{2}{\sqrt{3}}d_x$ hence we set

$$r_s = round(\frac{2}{\sqrt{3}}d_x) + 1 \tag{5}$$

We then look for optimal structure dependent positions inside each tile, distinguishing between the three intrinsic dimension classes:

i0D **Homogeneous image patches:** At homogeneous image patches the position can not be defined by properties of the local signal since it is constant. Therefore, the position $\boldsymbol{x}_{id0}^{(k,l)}$ of a Primitive representing an image patch $A^{(k,l)}$ is defined by the equidistant sampling (see figure 2a):

$$\boldsymbol{x}_{id0}^{(k,l)} = A_c^{(k,l)}$$

i1D **Lines and edges:** For a line or edge, the position $\boldsymbol{x}_{id1}^{(k,l)}$ can be defined through energy maxima that are organised as a one–dimensional manifold. Therefore, an equidistant sampling along these energy maxima is appropriate (see figure 2b). For this, we look for the energy maximum along a line orthogonal to the orientation at $A_c^{(k,l)}$ which is within the area $A^{(k,l)}$.

$$\boldsymbol{x}_{id1}^{(k,l)} = \max_{\mathbf{x} \in g^{(k,l)}} m(\boldsymbol{x})$$

where $g^{(k,l)}$ is a local line going through $A_c^{(k,l)}$ with orientation perpendicular to $\theta(A_c^{(k,l)})$.

i2D **Junction–like structures:** For a junction the position $\boldsymbol{x}_{id2}^{(k,l)}$ can be defined unambiguously as the maximum of the i2D confidence in a local region (see figure 2c and [13]):

$$\boldsymbol{x}_{id2}^{(k,l)} = \max_{A^{(k,l)}} \{c_{id1}(\boldsymbol{x})\}.$$

Our system runs in two modes. In the first mode, hereafter named *complete mode*, all three hypotheses are conserved (see figure 8b), however the position corresponding to the maximum of three confidences $c_{i0D}, c_{i1D}, c_{i2D}$ is called the *external position* $\boldsymbol{x}^{(k,l)}$ and it is used in the following process of reduction of redundant descriptors to compete with candidates computed in other tiles of the hexagonal grid. In the second mode, named *contour mode*, we only look at intrinsically one-dimensional signals, i.e., we do the positioning according to figure 2b. The first mode allows for a complete representation of the signal by also taking into account i0D and i2D structures. However, the symbolic representation as well as the 3D reconstruction of i0D and i2D signals differ and are ongoing research topics (see, e.g. [23, 50]). In the second mode, the symbolic representation of the primitives, their 3D reconstruction (see section 4) as well as important structural relations between primitives such as co-colority, symmetry and co-planarity are defined (see section 5.1).
All positions are computed with sub-pixel accuracy using the formula :

$$\begin{array}{rcl} \tilde{x}_0 & = & \frac{1}{s_g} \sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0+i, y_0+j)(x_0+i) \\ \tilde{y}_0 & = & \frac{1}{s_g} \sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0+i, y_0+j)(y_0+i) \end{array} \tag{6}$$

with $m(x, y)$ being the local amplitude at pixel position (x, y) and

$$s_g = \frac{1}{\sum_{i=-w_s}^{w_s} \sum_{j=-w_s}^{w_s} m(x_0+i, y_0+j)} \tag{7}$$

where w_s is set to $w_s = d_{leb}$ In section 3.3 the modalities phase and orientation of the extracted features are computed at the sub-pixel accuracy position by bi-linear interpolation.
Figure 9 shows the positions found for different intrinsic dimensions and also the external positions.

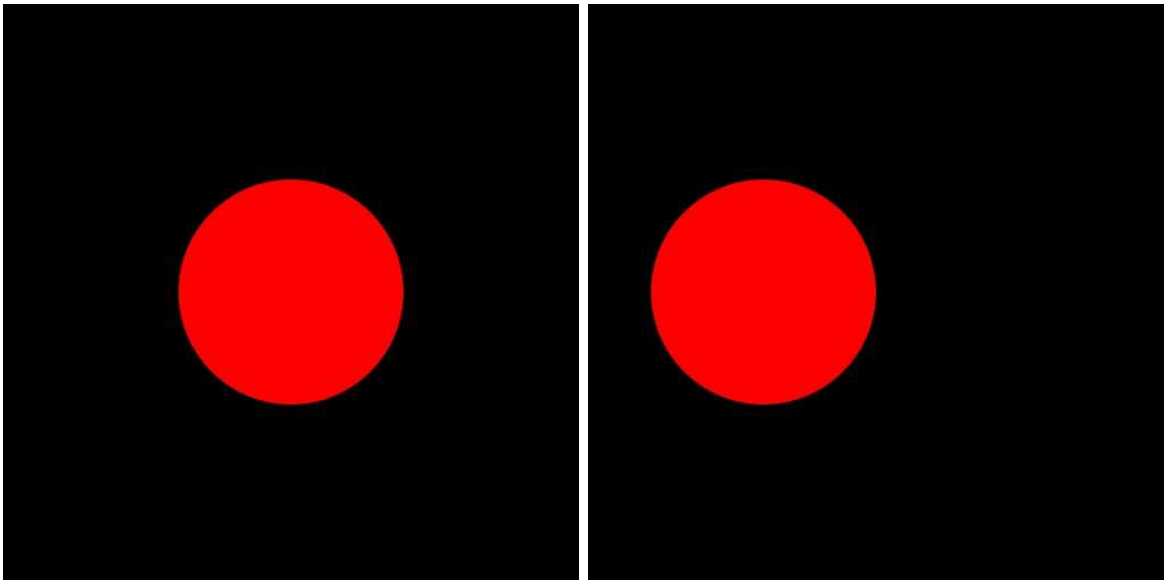
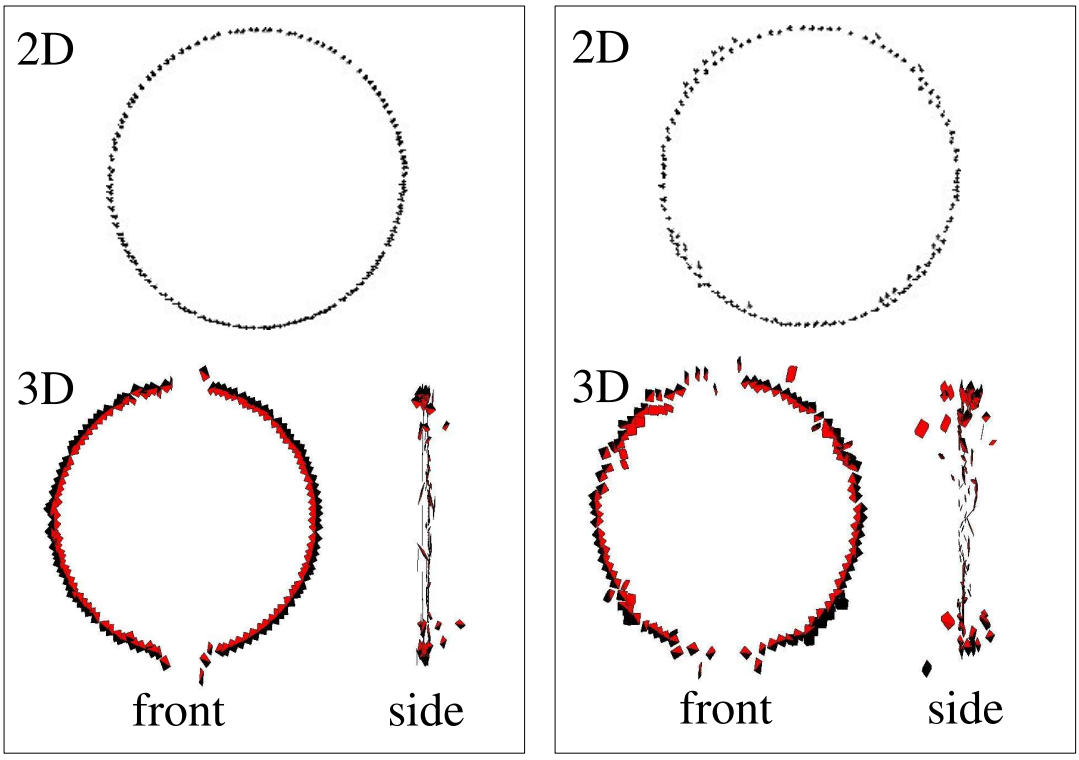


Figure 10: Artificial sequence used to evaluate the accuracy of primitive extraction (see figure 11).

Figure 14 shows the primitives extracted from a simple indoor scene (a). The primitives are extracted with a an origin variance > 0.3 and a line variance < 0.3 are shown for the three scales considered in this work: namely for peak frequencies of 0.110 (b), 0.055 (c), and 0.027 (d). Different scales highlight different structures in the scene.
In figure 10 an artificial sequence featuring a red circle on black background is shown. We evaluated the accuracy of the primitive extraction on this scene, and the results are recorded in figure 11. The top images compare the primitives extracted with (a) and without (b) the sub–pixel localisation of the primitives. Note that the sub–pixel localisation requires a symbolic interpretation of the primitive and that therefore we only considered i1D primitives. Effectively we only considered primitives with an origin variance larger than 0.3 and a line variance lower than 0.3. The upper graphs in (a) and (b) show the 2D primitives extracted and whereas the bottom ones show the 3D–primitives reconstructed using stereopsis.

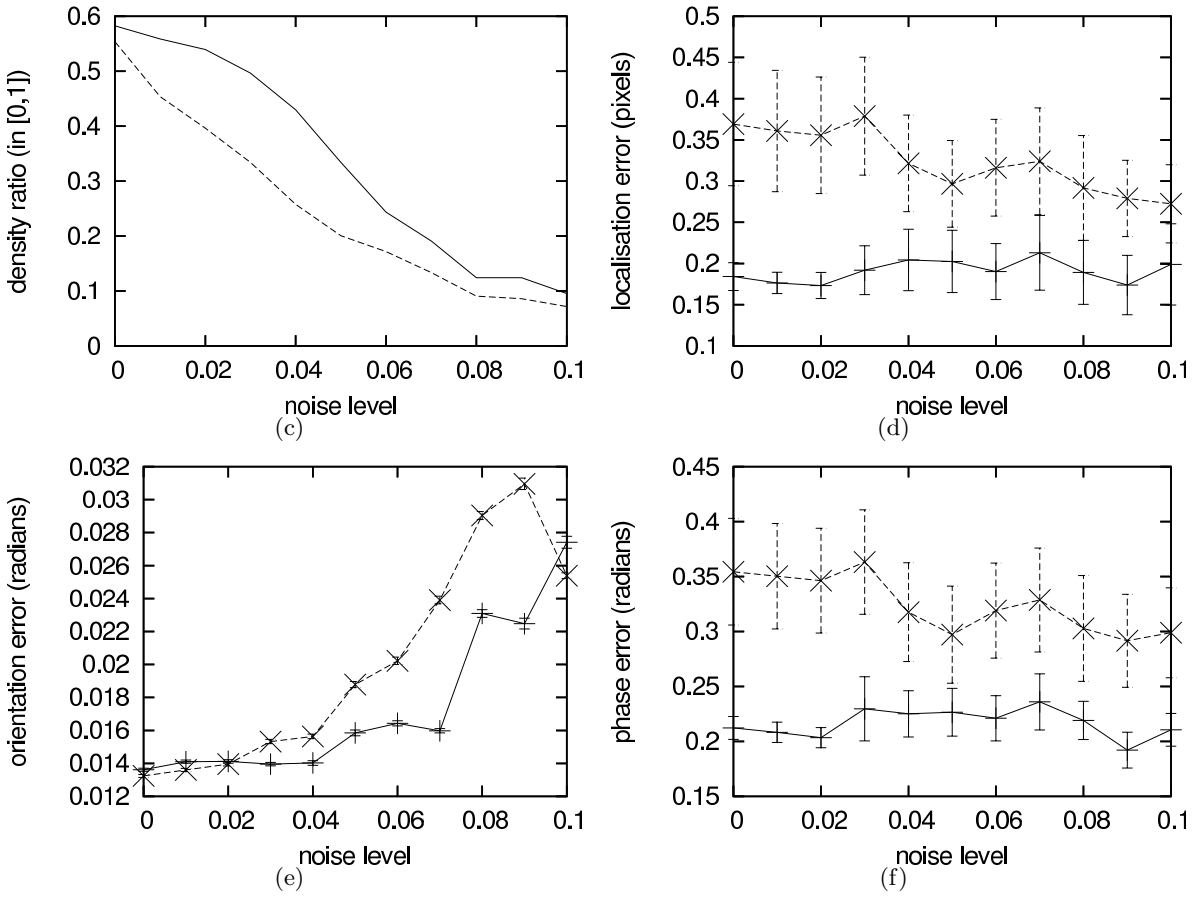
3.2 Elimination of redundant descriptors

Since the areas $A^{(k,l)}$ are overlapping, the process described above can lead to identical positions found in neighbouring areas (see figure 8c, $\boldsymbol{x}^{(2,1)} = \boldsymbol{x}^{(3,1)}$, $\boldsymbol{x}^{(1,1)} = \boldsymbol{x}^{(1,2)}$). And since the applied filters are extended in space it can also lead to positions with close spacing describing essentially the same structure (see figure 8d, $\boldsymbol{x}^{(2,1)}$ and $\boldsymbol{x}^{(3,1)}$.
In the second step described now, these redundant positions become eliminated. In this elimination process we face the following difficulty: On the one side, we do not want to eliminate 'independent structures' that are close to each other. For example, in the triangle in figure 6 two edges converge. At some point, these edges become interpreted as a line and the position should be on this line and the phase should become 0 or $\pm\pi$. Until then, the triangle should be represented by two edges with phase $\pm\pi$. Hence, the elimination process should not eliminate these 'independent' edges although they can be rather close to each other. The limit of separability is the line/edge bifurcation distance d_{leb} defined above. On the other side, since our kernels have an extension (expressed in the number of tabs n_t used to approximate the spatial filter) that is larger than d_{leb} there will still be a significant amplitude at pixel distances larger than d_{leb} (see figure 6).
As a consequence, eliminating candidates with distance smaller than d_{leb} would preserve all 'independent' edge structures but would also preserve a lot of redundant structures. However, eliminating candidates with distance smaller than n_t would eliminate all redundant but also the 'independent' structures.



(a) with sub–pixel localisation

(b) without sub–pixel localisation



(c) (d) (e) (f)

Figure 11: a) and b): 2D– and 3D–primitives extracted in the scenario illustrated in figure 10, respectively with and without sub–pixel localisation. c), d), e) and f) report the density and accuracy in localisation, orientation and phase of the primitives, wherein the solid line show the accuracy with sub–pixel localisation and the dashed line without. The error bars in d), e) and f) show the variance.

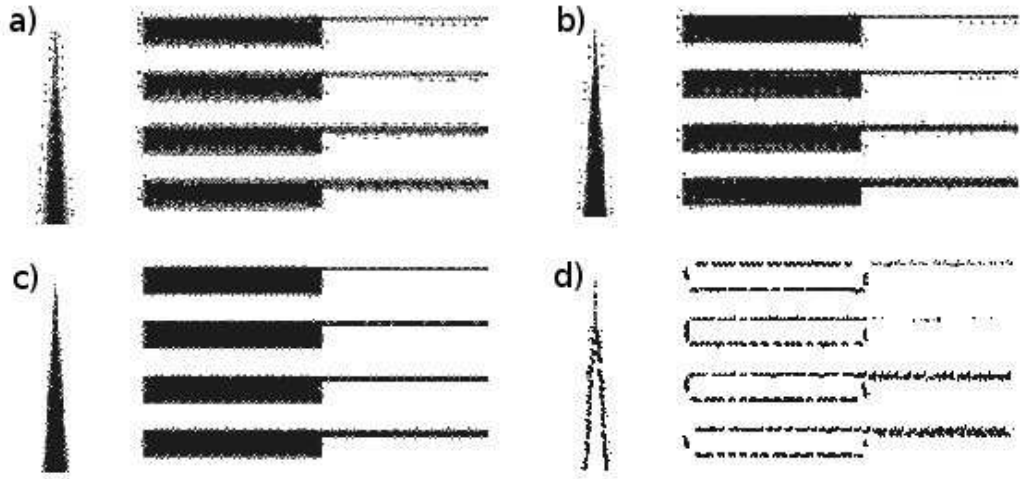


Figure 12: Three stages of the elimination process and the final primitive representation.

We tackle this problem by a two stage elimination process described in section 3.2.1 and 3.2.2.

3.2.1 Elimination based on the line/edge bifurcation distance

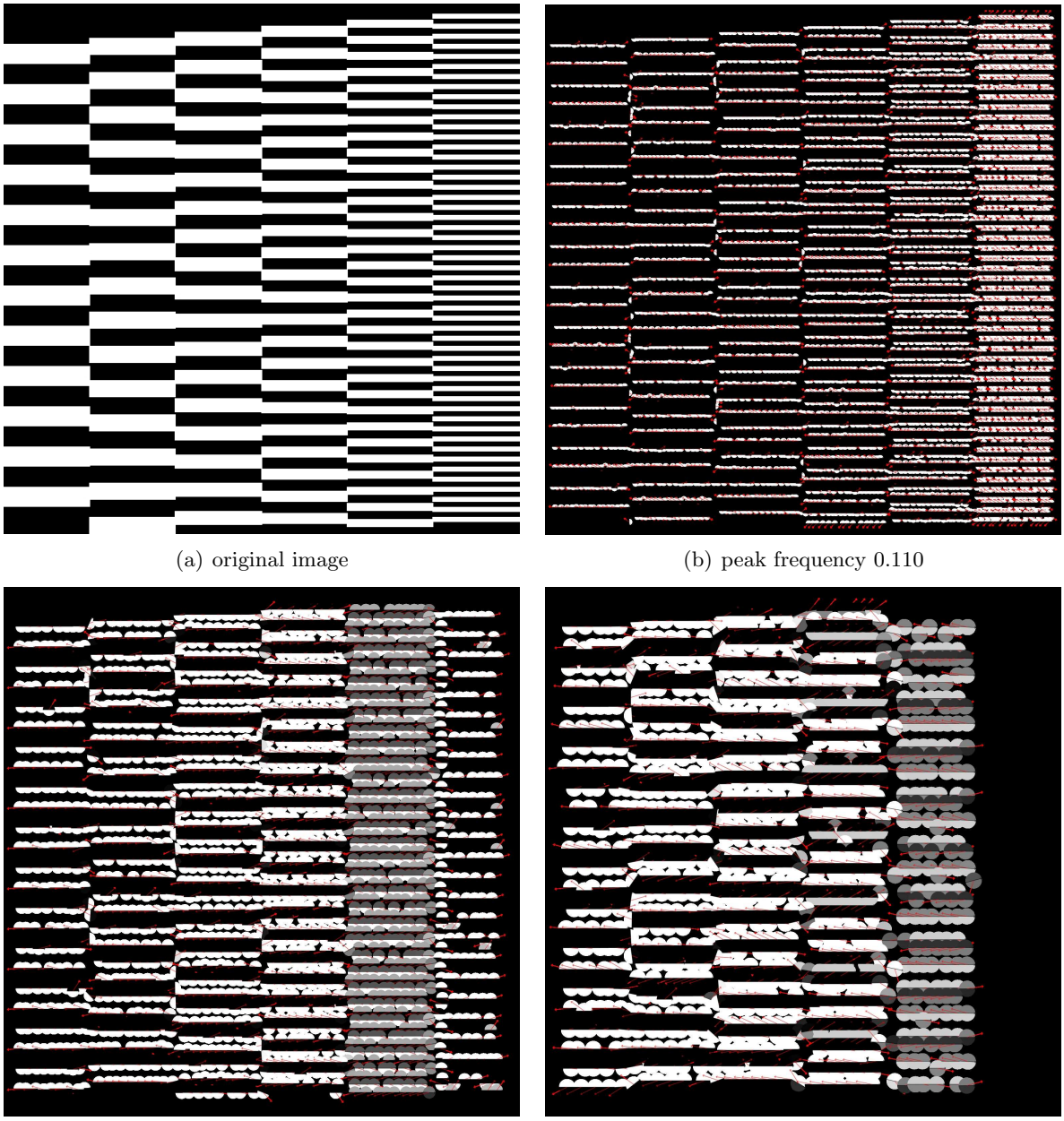
First, all candidates $\boldsymbol{x}^{(k,l)}$ become ordered according to the associated amplitude $m(\boldsymbol{x}^{(k,l)})$. Starting with candidates with the highest local amplitude we delete all other candidates $\boldsymbol{x}^{(k',l')}$ with a distance $d(\boldsymbol{x}^{(k,l)}), \boldsymbol{x}^{(k',l')}) = ||\boldsymbol{x}^{(k,l)}) - \boldsymbol{x}^{(k',l')}||$ smaller than d_{leb}.[8] Since we order the candidates according to the local amplitude, the candidate corresponding to a 'stronger' structure suppress candidates with weaker structure. Thereby all non–distinct edges (according to the line edge bifurcation distance) become deleted but redundant edges are still being preserved. In figure 12 upper–left, we see that many spurious candidates remain after the first elimination process that are caused by edges with distance smaller then d_k.

3.2.2 Elimination based on the kernel size

In the second step, again starting with the candidates with highest local amplitude, all remaining candidates become tested according to a distance d_k. d_k expresses the distance to which a structure can essentially effect pixels in the vicinity and is set to $d_k =$.
For a pair of intrinsically two dimensional structures it is sufficient to have distance smaller than d_{leb} since they naturally represent maxima in the amplitude representation [13]. If an intrinsically one-dimensional structure is involved there will be a slant in the local amplitude surface at the 'dependent' structure having its maximum at the edge/line structure decreasing with distance from the edge (see figure 6). This slant can be checked for: For each pair of candidates found with distance smaller d_k a test is made whether it represents an 'independent' structure. The criterion for independence we are using is whether the structure is a maximum on the line orthogonal to the local orientation. For each remaining candidate the amplitude is compared to the amplitude at pixels at a distance $d_{co} =?$ at both sides of the edge indicated by the local orientation.[9] If that is the case the candidate with lower local amplitude is discarded.
Thereby the remaining spurious candidates become eliminated. Figure 13 shows the primitives extracted for an artificial test image, for different scales. The figure in (a) shows vertically alternating black/white step–edges, getting narrower to the right of the figure. The primitives extracted at the three scales, for peak frequencies of 0.110, 0.055 and 0.027, are shown in (b), (c) and (d), respectively. The different effect of the double elimination process at different scales can be seen in this figure. For

[8] Note that for the quality of the process it is important that all positions are computed with sub-pixel accuracy already at this stage.

[9] Note that the criterion 'local maxima' that is applicable for i2D structures can not be applied since edge like structures form a ridge in the local amplitude surface (see figure 6).



(a) original image

(b) peak frequency 0.110

(c) peak frequency 0.055

(d) peak frequency 0.027

Figure 13: Dense sampling.

example if all of the narrower step edges to the right of the figure are distinctly extracted in (b), only one of the two is extracted in (c), whereas in (d) the same edges become intrinsically two–dimensional and are not extracted anymore.

3.3 Association of Visual Attributes and Confidences

Based on the found positions $\boldsymbol{x}^i$ we can associate visual attributes. The attributes orientation θ, phase φ, and optic flow $\boldsymbol{f}$ are computed pixel-wise using filter processes of spatial extend d_k[10]. Therefore, we associate the orientation, phase, as well as the optic flow according to the found positions $\boldsymbol{x}^i$.
Since, positions are computed with sub–pixel accuracy we can also interpolate the orientation, phase and optic flow value by bi–linear interpolation []. Let $\tilde{x}_0$ and $\tilde{y}_0$ be the positions computed with sub-pixel accuracy (see section 3.1). Let δ_x and δ_y be the distance to the discrete lower pixels x_l and y_l (and $x_h = x_0 + 1$ and $y_h = y_0 + 1$, then the bi-linear interpolation computation leads to the formula:

$$\begin{aligned}\tilde{\theta}(\tilde{x}) \quad = \quad & \hat{\theta}(x_l, y_l)(1-\delta_x)(1-\delta_y) + \hat{\theta}(x_l, y_h)(1-\delta_x) * \delta_y \\ & \hat{\theta}(x_h, y_l)\delta_x(1-\delta_y) + \hat{\theta}(x_h, y_h)\delta_x\delta_y\end{aligned}$$

Note that for the interpolation of orientation and phase the specific topology of the orientation phase space needs too be taken into account. Hence $\hat{\theta}$ is transformed such that the distance between all pairs of the set $\hat{\theta}(x_l, y_l), \hat{\theta}(x_l, y_h), \hat{\theta}(x_h, y_l), \hat{\theta}(x_h, y_h)$ is smaller than $\frac{\pi}{2}$ and $\hat{\theta}(\tilde{x})$ is in $[0, \pi)$. Phase is computed analogously.[11]
For the test picture shown in figure 10 we get a localisation error in the area of 0.1 pixel (i.e., improvement of a factor 10). Bi-linear interpolation of orientation and phase based on the the sub-pixel accuracy positioning leads also to improvements of a factor 2 and 6 respectively (on the highest frequency level). The effect on reconstruction is also demonstrated in figure 11.
Also colour information is available for each pixel position. However, especially for i0D and i1D signals the representation of colour is highly redundant. For a step-edge like structure it is natural to distinguish between the colour on the left and right side of the edge $(\boldsymbol{c}_l, \boldsymbol{c}_r)$ while for a line structure also the colour of a middle strip $\boldsymbol{c}_m$ should be coded (see figure 6c–e and 7).
As discussed in section 2.2 by the phase we can distinguish these two cases. For an homogeneous image patch (i0D), colour pixels can even be subsumed into one colour attribute.
Finally, we have a parametric description of a local area that we call a primitive. For a step edge we get

$$\pi_i = (\boldsymbol{x}_i, \theta(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_i), (\boldsymbol{c}_l(\boldsymbol{x}_i), \boldsymbol{c}_r(\boldsymbol{x}_i)), \boldsymbol{f}(\boldsymbol{x}_i))$$

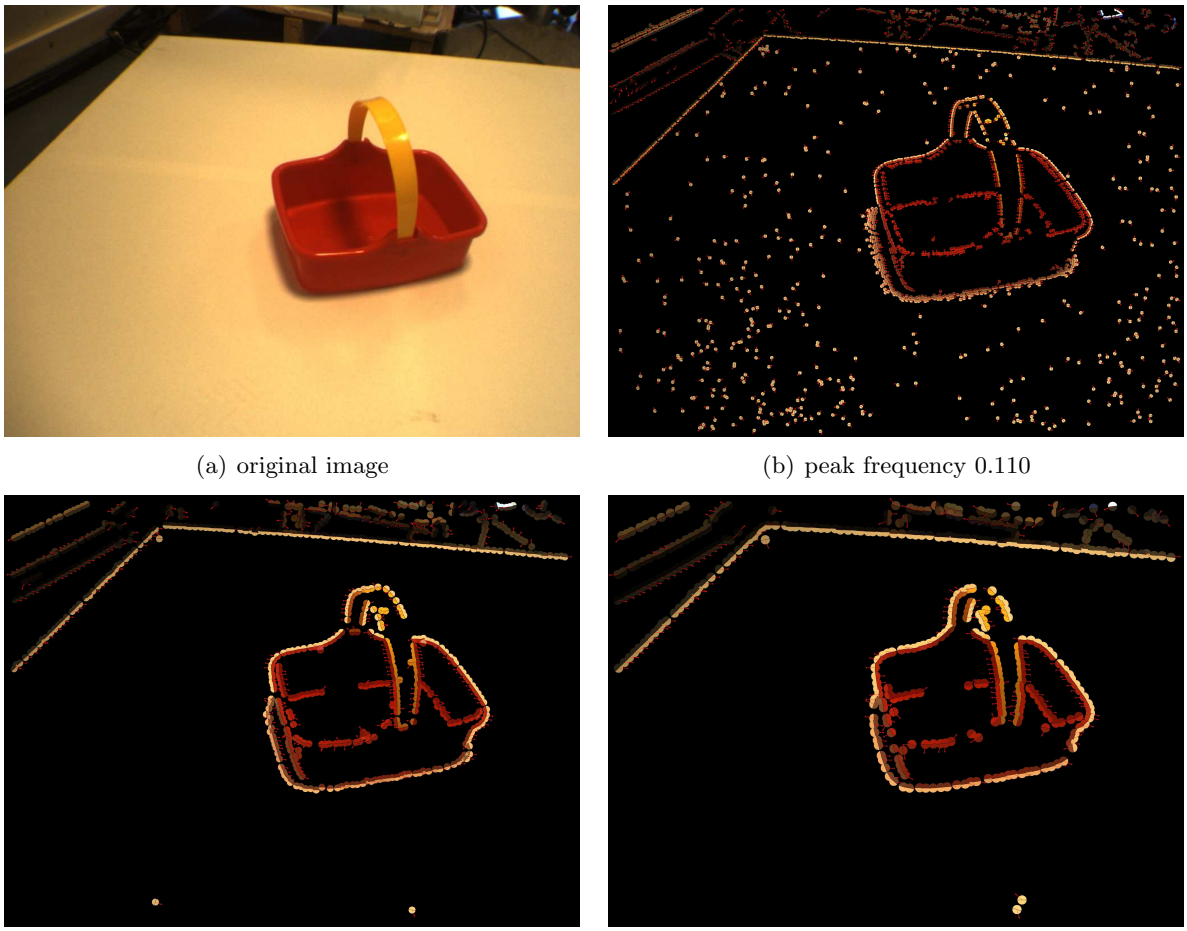
while for a line we get

$$\pi_i = (\boldsymbol{x}_i, \theta(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_i), (\boldsymbol{c}_l(\boldsymbol{x}_i), \boldsymbol{c}_m(\boldsymbol{x}_i), \boldsymbol{c}_r(\boldsymbol{x}_i)), \boldsymbol{f}(\boldsymbol{x}_i))\,.$$

The parameters of the primitives have a clear semantic and are a condensed representation of the local image patch. Condensation can be computed by the ratio of the number of bit needed to store a local image patch a primitive stands for. For the highest frequency, such a primitive represents a local image patch of a radius of appr. 3 pixels (i.e., $\pi \cdot 3^2 \cdot 3 \approx 85$ values). The primitive has a dimension of 10 for an edge like structure and 13 for a line–like structure(not counting the optic flow which indicates temporal information). That means that a primitive for the highest frequency level only requires maximal the $\frac{13}{85} = 0.15$ amount of bytes compared to the original image information leading to a condensation rate $d_{co} \approx 85\%$. Analogously, we get a condensation rate of $\approx 94\%$ and $\approx 97.\%$ for the other two frequency levels. Note that after computing the 3D primitives (see section 4) the condensation rate increases again significantly.

[10] Phase and orientation are output of the spherical quadrature filters while the area the optic flow estimation is based on can be determined in different flow algorithms in different ways.

[11]
$$\varphi(\tilde{x}) = \text{to be made}$$



(a) original image

(b) peak frequency 0.110

(c) peak frequency 0.055

(d) peak frequency 0.027

Figure 14: 2D–primitives extracted for different peak frequencies

Table 1 shows all parameters included in the primitive extraction. Note that these parameters are either naturally derived from the line edge bifurcation distance (d_{leb}) or are non-critical (w_s) or are based on decisions involving a trade off between computational complexity and precision (d_k).

4 Computation of 3D-Primitives

So far we have described multi-modal *image* descriptors that code *2D* information. However, these descriptors describe visual events occurring at a certain 3D position in space. This depth information is of essential use for higher level processes because of two reasons. First, human and robots act in a 3D world where depth information gives valuable indication where actions such as moving or grasping are possible. Second, since many structural dependencies of visual events (e.g., rigid body motion) are working on 3D structures the association of 3D information is essential for the formalisation of the disambiguation processes (see [41]).
In the following, we describe an extension of the image primitives to spatial primitives. In these spatial primitives, the semantic information coded in the image primitives is transferred into the 3D domain. Therefore we need to come to good interpretations of image information as 3D events.
Assuming the correspondences between primitives in two images are known (for how this is done, see [41]) we are able to extract spatial primitives as described in section 4.1 (see also figure 16).

4.1 Constructing Spatial multi–modal Primitives

Given a pair of corresponding points between the left and right image, a meaningful 3D interpretation of this stereo–pair is a 3D point. Contours, however, hold a 2D orientation, and therefore 3D–primitives need to encode the reconstructed 3D orientation $\boldsymbol{\Theta}$ beside the 3D position $\boldsymbol{X}$; this orientation is computed as the intersection of two planes in space, each defined by the optical centre of one camera and the line in the image plane described by the image primitive's position and orientation — see figure 15. The intersection of these two planes in space is a 3D line that provides us with the orientation of the 3D primitive. In [48] it was shown that using line correspondences for the reconstruction of 3D orientation was generally more accurate than points correspondences.
Phase and colour are reconstructed in space as the mean value between the two corresponding image primitives.

$$\Phi = \frac{\varphi^L + \varphi^R}{2} \tag{8}$$

$$\boldsymbol{C} = \frac{\boldsymbol{c}^L + \boldsymbol{c}^R}{2} \tag{9}$$

Moreover these two modalities encode surface information (respectively contrast and colour transition across an edge) thus we need to define a 3D surface patch onto which they apply. Unfortunately it is not possible to reconstruct the exact surface from local information: for a pure i1D signal the surface on one side does not allows to find the additional correspondence that would be required for the reconstruction of a 3D surface. Moreover, in case of a depth discontinuity the colour information might come from a 3D position that is completely independent from the 3D orientation information (i.e. the background).
We propose to define as *a priori* 3D surface the plane that is most stable under small viewpoint variation (see figure 15). This surface is computed using the 3D orientation of the primitive and an additional vector Γ that is defined as follows:

$$\Gamma = \boldsymbol{\Theta} \times V_{pov} \tag{10}$$

such that the surface is normal to V_{pov}, and V_{pov} is defined as follows

$$V_{pov} = \frac{1}{2}\left(\overrightarrow{C_L\boldsymbol{X}} + \overrightarrow{C_R\boldsymbol{X}}\right) \tag{11}$$

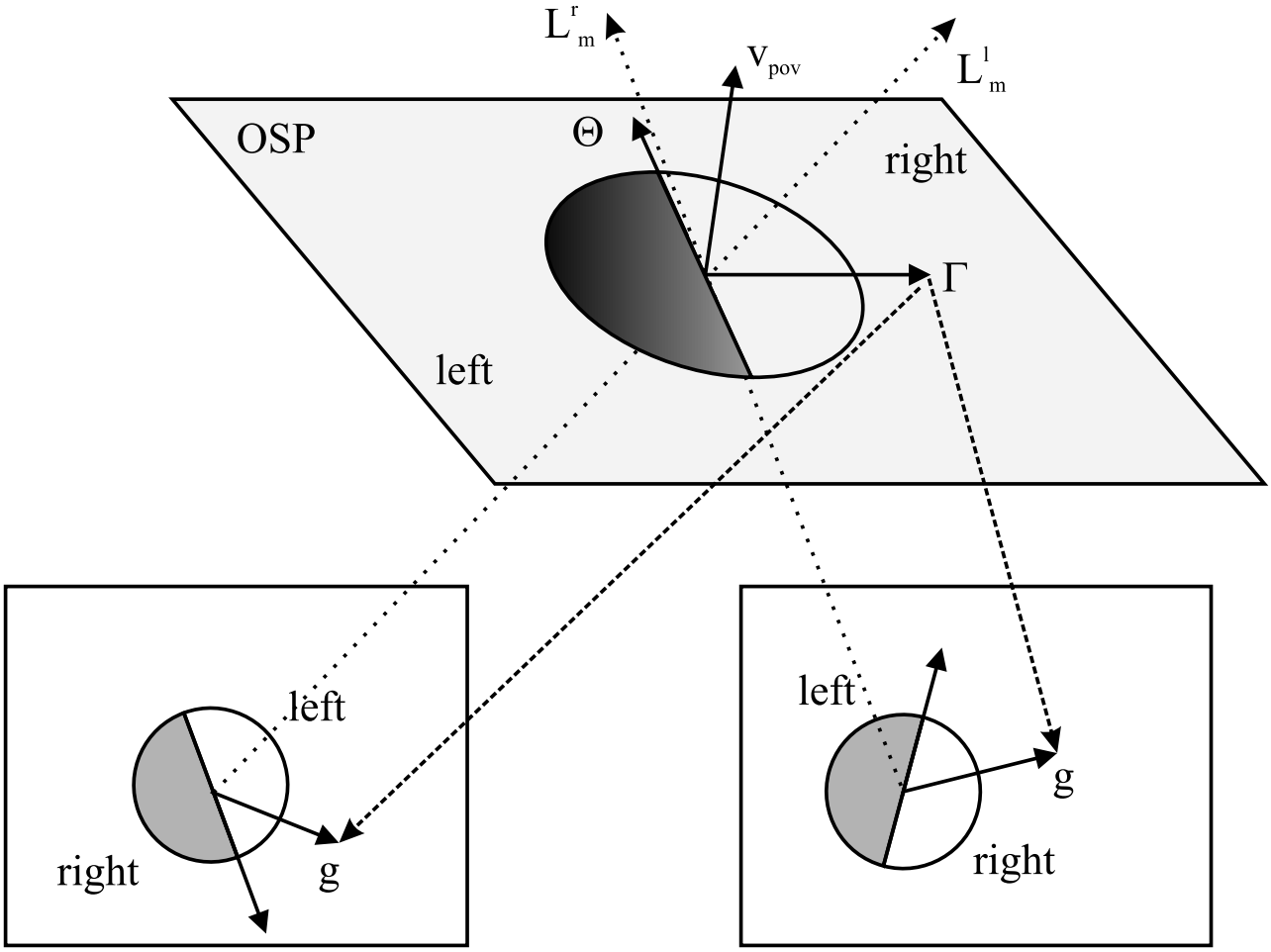


Figure 15: Illustration of the reconstruction of the 3D orientation.

where $\overrightarrow{C_L\boldsymbol{X}}$ and $\overrightarrow{C_R\boldsymbol{X}}$ are the two optical rays joining the location of the primitive $\boldsymbol{X}$ with the optical centre of the left (C_L) and right (C_R) cameras. The vector Γ also identifies each side of the 3D line, which is critical for modalities like colour and phase that describe the modality transition across the contour.
We end up with a set of spatial primitive $\Pi^{(i,j)}$ each having the parametric description

$$\Pi^{(i,j)} = (\boldsymbol{X}, \boldsymbol{\Theta}, \Phi, (\boldsymbol{C}_l, \boldsymbol{C}_m, \boldsymbol{C}_r)) \tag{12}$$

The j-index represents the alternative 3D entities generated from different correspondences in the right image to the i-th primitive in the left image. Since a final decision can usually not be made with high reliability solely based on local information, multiple hypotheses are kept at this stage. In the following section we will describe different approaches to overcome this ambiguity.
In figure 11 (a) and (b), bottom, the 3D primitives reconstructed with (a) and without (b) sub–pixel localisation are shown from front and side view. The side view offer a better vision of the quality of the depth estimation from stereopsis.[12] It is visible in these images that the sub–pixel localisation of the primitives described in section 3.1 allows for a notably better 3D–reconstruction.
In figure 16 the 3D–primitives reconstructed in an indoor scene are shown. Figures (a) and (b) show the stereo pair of images used, (c) (resp (d)) shows the 2D–primitives extracted with (resp. without) sub–pixel accuracy, and the subsequently reconstructed 3D–primitives are shown in (e) (resp. (f)).

5 Applications

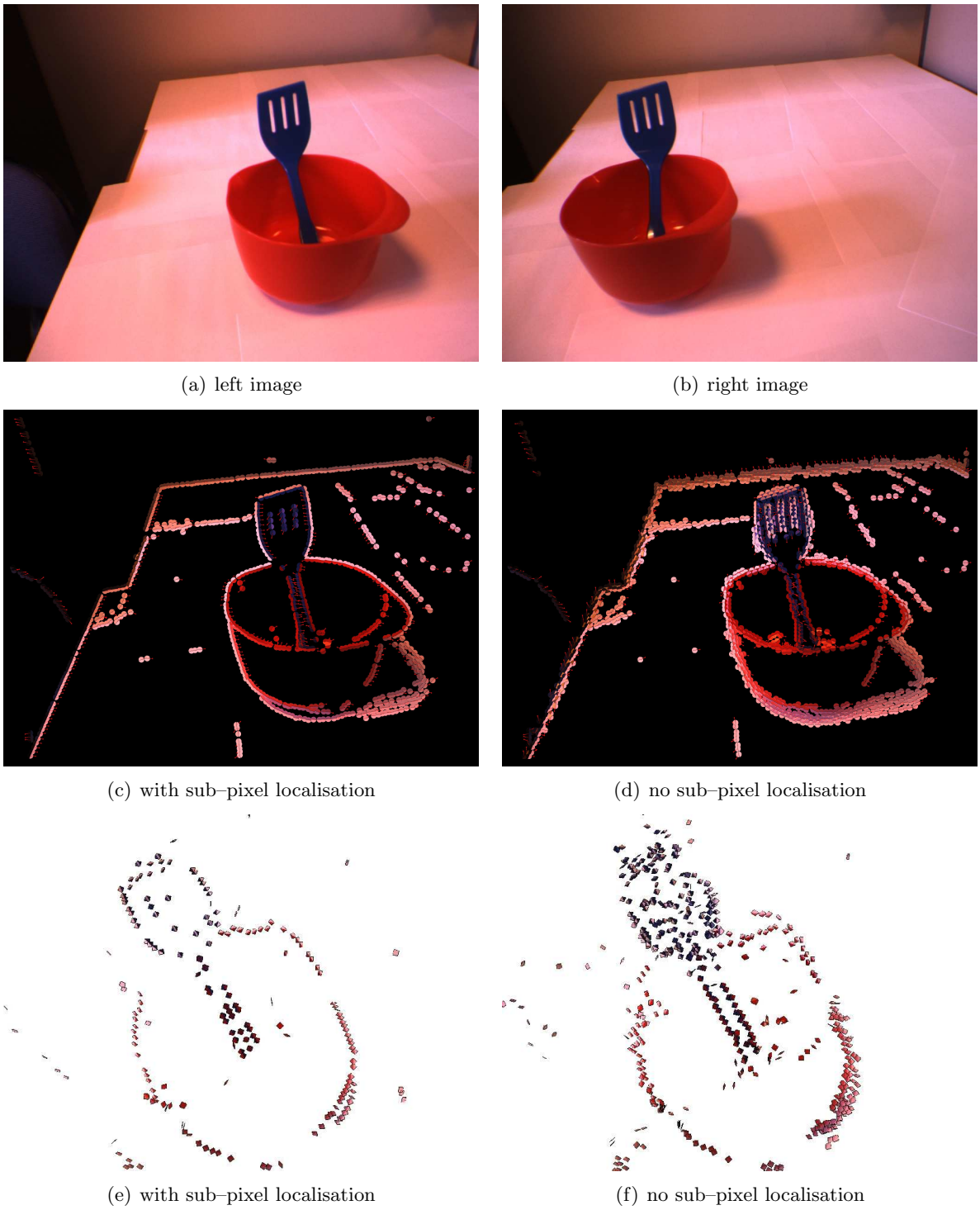
The primitive representation introduced in this paper has been applied in various contexts (briefly described in subsection 5.2 to 5.5) and has been part of three different European projects [8, 1, 18] in the area of Cognitive Vision and visual based robotics. The primitives described so far are condensed localised descriptors with clear semantics, and by this, symbolic descriptors of a local image patch. Since they are processed locally they are necessarily as ambiguous as the locally computed modalities that are represented by them. However, the data format the primitives provide allows for the definition of a set *semantic relations* upon them (see figure 17a). Since the primitives are a symbolic description of the local image patch, the relations and operation defined on the primitives provide the context in which information is processed.
The relations are used at a stage of processing after the condensation step (called early cognitive vision in [29]). More specifically, by the relations

- predictions between visual events become formulated (such as the change of a local image patch under motion or the likelihood of being part of the same collinear group) and by that the locally ambiguous information becomes, disambiguated (see section 5.2),
- the sets of primitives can become connected to higher visual entities such as 3D surfaces (section 5.3) and objects (section 5.4),
- low–order combinations of primitives become associated to robot actions such as grasping (section 5.5).

5.1 Relations and Operations defined on Primitives

Here we briefly describe the definition of four second order relations on primitives: Collinearity, rigid body motion, co-planarity and Co-colority (see also figure 17a).
Collinearity: In [41] a measure for the likelihood of two 2D primitives being part of the same collinear group $Coll(\pi_i, \pi_j)$ is defined (see figure 17a,i). This allows for the definition of a stereo constraint (see, e.g., [4, 44] that makes use of local image information as well as the semi–global context (see [41]). The collinearity constraint can naturally be extended to 3D primitives ($Coll(\Pi_i, \Pi_j)$).

[12] Note that the accuracy of the depth estimates decreases for horizontal structure. This is due to the ambiguity in reconstructing lines parallel to the epipolar line.



(a) left image (b) right image

(c) with sub–pixel localisation (d) no sub–pixel localisation

(e) with sub–pixel localisation (f) no sub–pixel localisation

Figure 16: Reconstruction of 3D–primitives in a real scenario. The two stereo images are shown in (a) and (b) (c) (resp. (d)): 2D–primitives extracted with (resp. without) sub–pixel localisation; and (e) (resp. (f)): spatial primitives reconstructed with (resp. without) sub–pixel localisation.

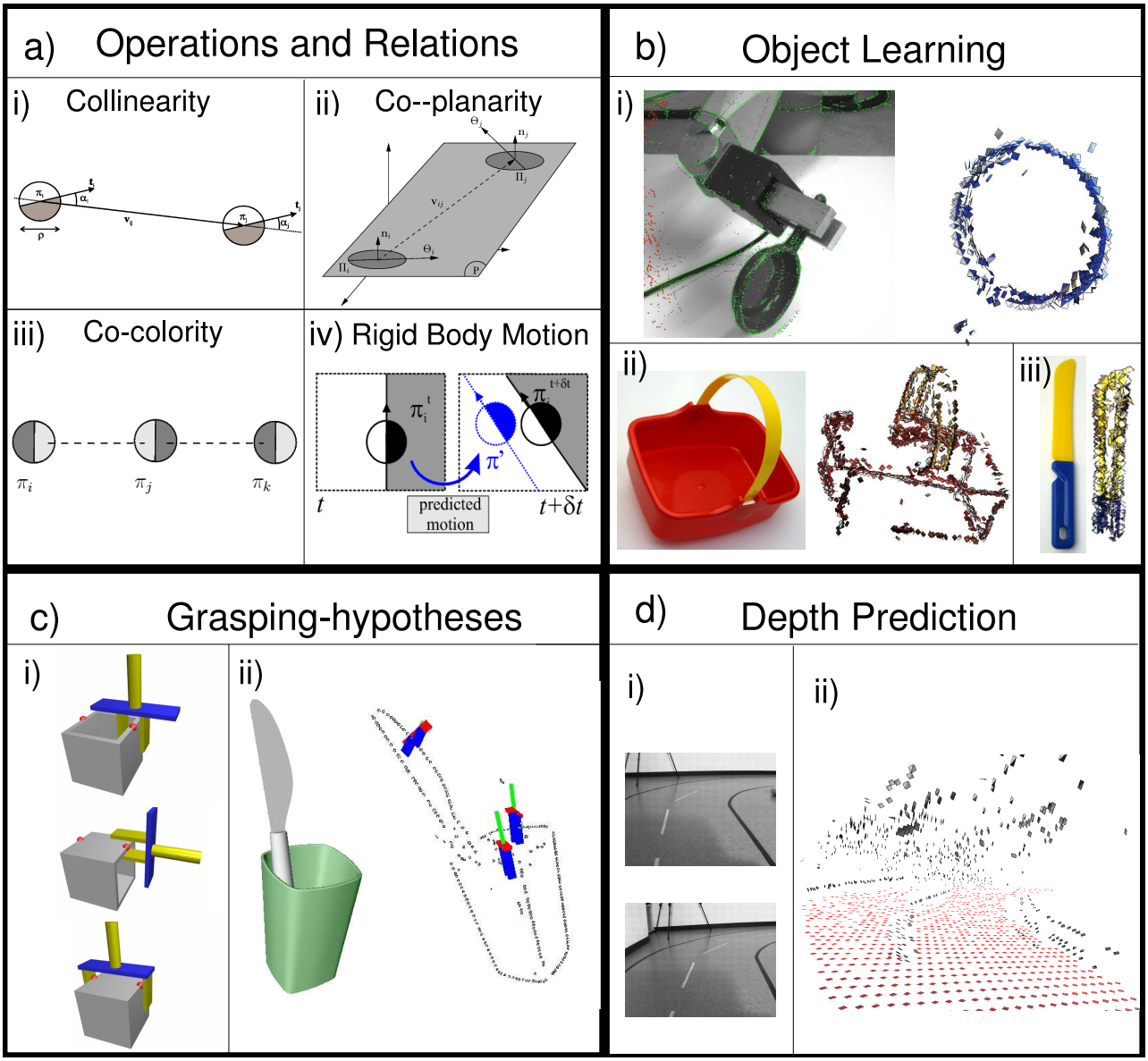


Figure 17: a) Relations defined on the multi–modal primitives. b) Grasping options generated by second order relations of primitives. c) Extraction of object representations. d) Depth predictions based on co–planarity relations.

Rigid body motion: The change of the parameters position and orientation under a rigid body motion ($RBM(\Pi)$) can be computed analytically (see, e.g., [9]) while the parameters phase and colour can be approximated to be constant under a motion (see figure 17a,iv).
Co–planarity: The relations co–planarity $Cop(\Pi_i, \Pi_j)$ between two 3D primitives (see figure 17a,ii) indicates the likelihood of the primitives to be part of the same surface (see section 5.3) and by this can be related to a grasping option (see section 5.5).
Co-colority: The relation co–colority (see figure 17a,iii) expresses the similarity of the colours at the side of two edges that are pointing towards each other.

5.2 Disambiguation using Motion and perceptual Grouping

In [42] it has been shown that such a representation allowed for computing the ego–motion of the camera rig with an accuracy sufficient for tracking individual primitives over time. It was discussed in [49] that the knowledge of this motion allows to predict the transformation between representations of the same scene at different instants, thereby correct the scene representation over time. The 3D–

hypotheses that are confirmed over time by the estimated motion gain a stronger confidence whereas hypotheses that are contradicted can be discarded as outliers.

5.3 Depth prediction at homogeneous image Areas

The primitives introduced in here represent i1D structures. It is known that it becomes increasingly difficult to find correspondences between local patches the more they are lacking structure (i.e., tending toward the i0D corner of the iD triangle (see figure 3). On the other hand, it is known that the lack of structure also indicates the lack of a depth discontinuity [17, 23]. However it was statistically shown in [24] that coplanarity allows to predict depth at homogeneous image surfaces (see figure 17d).

5.4 Object Learning and Recognition

The primitives are rich and condensed descriptors of scene information. Hence they are suitable for memorising objects in an efficient way, In particular the relations $RBM(\Pi)$ can be used to (1) get a disambiguated and hence reliable representation of objects (see section 5.2) and (2) to segment an object from the background (see figure 17b. This second property is in particular relevant in the context of the European project PACO+ [1] in which the early cognitive vision system introduced here will be linked with an AI planning system that requires objects as discrete entities (see [15]). Hence, a cognitive robot vision system should be able to find out about the 'objectness' of a set of visual features as well as the shape of the object by itself. This is achieved by combining the object learning introduced described here with the grasping approach described in section 5.5. Once representations of objects are extracted that way they can be used for pose estimation and object recognition [7].

5.5 Generating Grasping Hypotheses

Also, in the European project [1] our primitive representation is used to define grasping options in a scene (see figure 17c) and [2]). Essentially, co-planar primitives (supported by the relations co–linearity and co-colority) define planes that are good candidates for an initial grasping hypothesis. In figure 17c,i) the definition of grasping hypotheses from co-planar primitives is shown. Figure 17c,i) shows generated grasps at scenario created by the grasping simulation software GraspIt used for the evaluation of our approach (for details, see [2]). Once evaluated as successful by haptic information, gives the physical control over objects required for the object learning sketched in section 5.4.

6 Summary and Discussion

At the current state of development our system treats different scales independently. Since we are dealing with edge like structures which tend to show stable properties over different scales that is appropriate. However, it would be advantageous to find the appropriate scale to reduce memory and computational requirements. A treatment of our approach in a scale–space approach where the scale itself expressed by a feature (see, e.g., [34]) is currently being considered.
Furthermore, we intend introduce symbolic descriptors for different image structures. For homogeneous image patches this has been already discussed in section 5.3. In [50] we have discussed an extenofn opf our approach to junction–like structures. We note that this requires not only a junction detection and interpretation algorithm but also the definition of appropriate relations between different junctions as well as between edges and junctions. We are also doing first steps towards the representation of texture which in particular requires a representation of different scales.

Acknowledgement: The work on the multi–modal primitives started in 1998 in Kiel, Germany. It became an important part of the European project ECOVISION (2001–2003) [8] and is now applied in the context of vision based robotics as well as driver assistant systems in the two European projects PACOplus (2006–2010) [1] and Drivsco (2006–2009) [18]. Many master and PhD students have been involved in this project and we would like to thank Markus Ackermann, Emre Baseski, Kord Ehmcke,

Michael Felsberg, Christian Gebken, Oliver Granert, Danial Grest, Marco Hahn, Thomas Jäger, Sinan Kalkan, Dirk Kraft, Florian Pilz, Martin Pörksen, Torge Rabsch, Bodo Rosenhahn, Morten Skov, Shi Yan, Daniel Wendorff and Jan Woetzel. We would like to thank in particular and for their contributions to this work.

References

[1] Pacoplus: Perception, action and cognition through learning of object-action complexes. *Integrated Project*, 2006-2010.

[2] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early Reactive Grasping with Second Order 3D Feature Relations, journal = IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision, year = 2007,.

[3] H. Barlow, C. Blakemore, and J.D. Pettigrew. The neural mechanisms of binocular depth discrimination. *Journal of Physiology (London)*, 193:327–342, 1967.

[4] R.C.K. Chung and R. Nevatia. Use of monucular groupings and occlusion analysis in a hierarchical stereo system. *Computer Vision and Image Understanding*, 62(3):245–268, 1995.

[5] H.S.M. Coxeter. *Introduction to Geometry (2nd ed.)*. Wiley & Sons, 1969.

[6] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by 2d visual cortical filters. *Journal of the Optical Society of America*, 2(7):1160–1169, 1985.

[7] R. Detry, N. Pugeault, N. Krüger, and J. Piater. Hierarchical integration of local 3D features for probablistic pose estimation. *INTELSIG Technical Report 2007–01–19, Department of Electrical Engineering and Computer Science University of Liege*, 2007.

[8] ECOVISION. Artificial visual systems based on early-cognitive cortical processing (EU–Project). *http://www.pspc.dibe.unige.it/ecovision/project.html*, 2003.

[9] O.D. Faugeras. *Three–Dimensional Computer Vision*. MIT Press, 1993.

[10] M. Felsberg. *Low-Level Image Processing with the Structure Multivector*. PhD thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002.

[11] M. Felsberg, S. Kalkan, and N. Krüger. Continuous characterization of image structures of different dimensionality. in preparation.

[12] M. Felsberg and N. Krüger. A probablistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*, 2003.

[13] M. Felsberg and G. Sommer. A new extension of linear signal processing for estimating local properties and detecting features. In G. Sommer, N. Krüger, and C. Perwass, editors, *22. DAGM Symposium Mustererkennung, Kiel*, pages 195–202. Springer-Verlag, Heidelberg, 2000.

[14] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144, December 2001.

[15] Ch. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger, and F. Wörgötter. Object action complexes as an interface for planning and robot control. *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, 2006.

[16] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1995.

[17] W.E.L. Grimson. Surface consistency constraints in vision. *CVGIP*, 24(1):28–51, 1983.

[18] http://www.pspc.dibe.unige.it/ drivsco/, editor. *DRIVSCO: Learning to Emulate Perception-Action Cycles in a Driving School Scenario (FP6-IST-FET, contract 016276-2)*. 2006-2009.

[19] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Phyiology*, 160:106–154, 1962.

[20] D.H. Hubel and T.N. Wiesel. Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750, 1969.

[21] B. Jähne. *Digital Image Processing – Concepts, Algorithms, and Scientific Applications*. Springer, 1997.

[22] S. Kalkan, D. Calow, F. Wörgötter, M. Lappe, and N. Krüger. Local image structures and optic flow estimation. *Network: Computation in Neural Systems*, 16(4):341–356, 2005.

[23] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1121, 2006.

[24] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2007.

[25] P. König and N. Krüger. Perspectives: Symbols as self-emergent entities in an optimization process of feature extraction and predictions. *Biological Cybernetics*, 94(4):325–334, 2006.

[26] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.

[27] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, pages 261–270, 2003.

[28] N. Krüger and M. Felsberg. An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863, 2004.

[29] N. Krüger, M. Van Hulle, and F. Wörgötter. Ecovision: Challenges in early-cognitive vision. *International Journal of Computer Vision*, submitted.

[30] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.

[31] N. Krüger and F. Wörgötter. Multi modal estimation of collinearity and parallelism in natural image sequences. *Network: Computation in Neural Systems*, 13:553–576, 2002.

[32] N. Krüger and F. Wörgötter. Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131:82–147, 2004.

[33] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamik link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[34] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[35] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. Freeman, 1977.

[36] R.M. Mersereau. The processing of hexagonally sampled two–dimensional signals. *Proc. IEEE*, 67(6):930–949, 1979.

[37] L. Middleton and J. Sivaswamy. *Hexagonal Image Processing : A Practical Approach*. Springer Verlag, 2005.

[38] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[39] M.W. Oram and D.I. Perrett. Modeling visual recognition from neurobiological constraints. *Neural Networks*, 7:945–972, 1994.

[40] N. Pugeault. *Working Title: Early Cognitive Vision*. 2006.

[41] N. Pugeault, F. Wörgötter, , and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision, New York City June 22, 2006 (in conjunction with IEEE CVPR 2006)*, 2006.

[42] N. Pugeault, F. Wörgötter, , and N. Krüger. Rigid body motion in an early cognitive vision framework. In *Proceedings of the IEEE Systems, Man and Cybernetics Society Conference on Advances in Cybernetic Systems*, 2006.

[43] S.P. Sabatini, Karl, Javier, and Nico. Working title: Analysis of harmonic filter design. to be submitted.

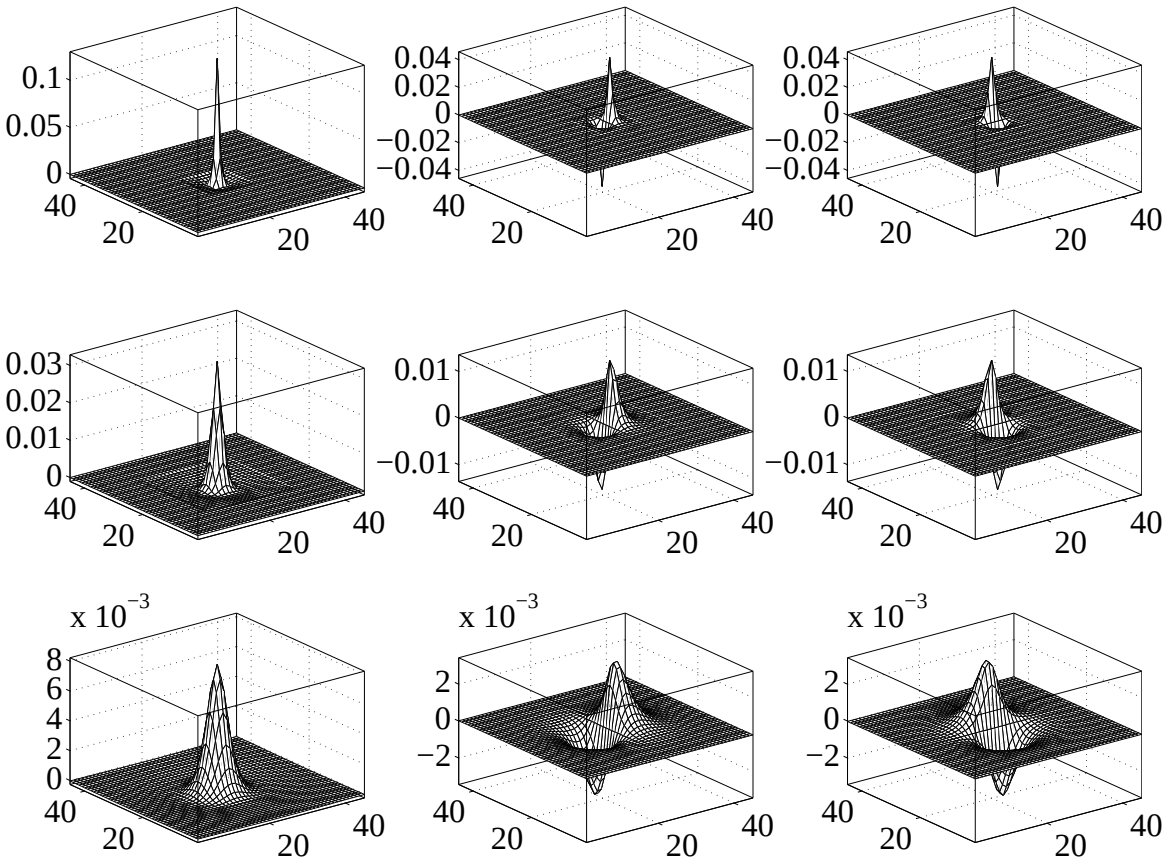


Figure 18: Impulse responses of the DOP filter and its Riesz transforms. From left to right: DOP filter, first Riesz transform, second Riesz transform. From top to bottom: scales (1,2), (2,4), (4,8).

[44] S. Sarkar and K.L. Boyer. *Computing Perceptual Organization in Computer Vision*. World Scientific, 1994.

[45] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. *Advances in Neural Information Processing Systems*, 8:865–871, 1996.

[46] I.A. Shevelev, N.A. Lazareva, A.S. Tikhomirov, and G.A. Sharev. Sensitivity to cross–like figures in the cat striate neurons. *Neuroscience*, 61:965–973, 1995.

[47] R.C. Staunton and N. Storey. A comparison between square and hexagonal sampling methods for pipeline image processing. *Proc. SPIE*, 1194:142–151,, 1989.

[48] Lawrence B. Wolff. Accurate measurements of orientation from stereo using line correspondence. 1989.

[49] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. Van Hulle, S. Tan, and A. Johnston. Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing. *Natural Computing*, 3(3):293–321, 2004.

[50] Shi Yan, Sinan Kalkan, Nicolas Pugeault, and Norbert Krüger. Corner stuff. to be submitted.

[51] C. Zetzsche and E. Barth. Fundamental limits of linear filters in the visual processing of two dimensional signals. *Vision Research*, 30, 1990.

A Split of Identity

Quadrature filters based on the monogenic signal [14] are rotation invariant, i.e., they commute with the rotation operator. Hence, for an appropriate choice of polar coordinates, two coordinates do

not change under rotations (amplitude and phase), whereas the third coordinate directly reflects the rotation angle. This kind of quadrature filter, which is called *spherical quadrature filter* [10], is formed by triplet of filters: a radial bandpass filter and its two Riesz transforms [21]. As in [10] we construct the bandpass filter from *difference of Poisson* (DOP) filters, in order to get analytic formulations of all filter components in the spatial domain and in the frequency domain. The DOP filter is an even filter (w.r.t. point reflections in the origin) and its impulse response (convolution kernel) and frequency response (Fourier transform of the kernel) are respectively given by:

$$h_{\mathrm{e}}(\boldsymbol{x}) = \frac{s_1}{2\pi(|\boldsymbol{x}|^2+s_1^2)^{\frac{3}{2}}} - \frac{s_2}{2\pi(|\boldsymbol{x}|^2+s_2^2)^{\frac{3}{2}}} \tag{13}$$

$$H_{\mathrm{e}}(\boldsymbol{u}) = \exp(-2\pi|\boldsymbol{u}|s_1) - \exp(-2\pi|\boldsymbol{u}|s_2) \ . \tag{14}$$

For convenience, we combine the two Riesz transforms of the DOP filter in a complex, odd filter, yielding the impulse response and the frequency response:

$$h_{\mathrm{o}}(\boldsymbol{x}) = \frac{\boldsymbol{x}_1+i\boldsymbol{x}_2}{2\pi(|\boldsymbol{x}|^2+s_1^2)^{\frac{3}{2}}} - \frac{\boldsymbol{x}_1+i\boldsymbol{x}_2}{2\pi(|\boldsymbol{x}|^2+s_2^2)^{\frac{3}{2}}} \tag{15}$$

$$H_{\mathrm{o}}(\boldsymbol{u}) = \frac{u_2-iu_1}{|\boldsymbol{u}|}(\exp(-2\pi|\boldsymbol{u}|s_1) - \exp(-2\pi|\boldsymbol{u}|s_2)) \ , \tag{16}$$

respectively. The impulse responses of the filters for $(s_1, s_2) = (1,2), (2,4), (4,8)$ are shown in figure 18. The split of identity (i.e., the separation of the signal into local amplitude, orientation and phase) is obtained by switching to appropriate polar coordinates. In particular, we transform the filter responses according to

$$m(\boldsymbol{x}) = \sqrt{I_{\mathrm{e}}(\boldsymbol{x})^2 + |I_{\mathrm{o}}(\boldsymbol{x})|^2} \tag{17}$$

$$\theta(\boldsymbol{x}) = \arg I_{\mathrm{o}}(\boldsymbol{x}) \pmod{\pi} \tag{18}$$

$$\varphi(\boldsymbol{x}) = sign(\Im\{I_{\mathrm{o}}(\boldsymbol{x})\}) \arg(I_{\mathrm{e}}(\boldsymbol{x}) + i|I_{\mathrm{o}}(\boldsymbol{x})|) \ , \tag{19}$$

which gives the desired amplitude, orientation, and phase information.
Figure 19 shows a radial cut through the DOP bandpass filters for a certain range of scales and their superposition, demonstrating a homogeneous covering of the frequency domain. For infinitely many bandpass filters, the superposition is one everywhere, except at the origin. In our system, we apply filters on three frequency levels (see figure 18). The applied bandpasses are indicated by the darker colour in figure 19.

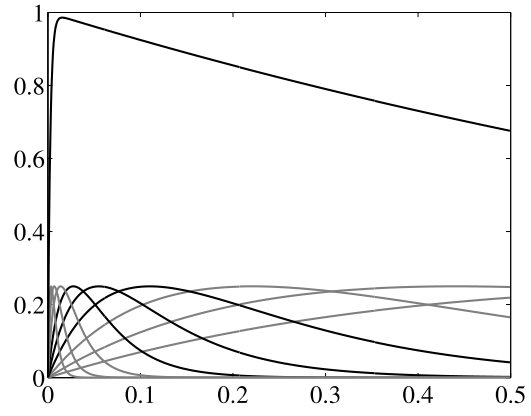


Figure 19: DOP bandpass filters and their superposition approaching the identity (x–axis representing the frequency). The superposition and the filters applied in this paper are indicated by the darker lines.

Extraction of multi–modal Object representations in a Robot Vision System

International Workshop on Robot Vision, in conjunction with VISAPP'07

Nicolas Pugeault[1], Emre Baseski[2], Dirk Kraft[2], Florentin Wörgötter[3], and Norbert Krüger[2]

[1] University of Edinburgh,
United Kingdom.
npugeaul@inf.ed.ac.uk
[2] Syddansk Universitet,
Denmark.
emre,kraft,norbert@mip.sdu.dk
[3] Göttingen University,
Germany.
worgott@jupiter.chaos.gwdg.de

Abstract. We introduce one module in a cognitive system that learns the shape of objects by active exploration. More specifically, we propose a feature tracking scheme that makes use of the knowledge of a robotic arm motion to: 1) segment the object currently grasped by the robotic arm from the rest of the visible scene, and 2) learn a representation of the 3D shape without any prior knowledge of the object. The 3D representation is generated by stereo–reconstruction of local multi–modal edge features. The segmentation between features belonging to the object those describing the rest of the scene is achieved using Bayesian inference. We then show the shape model extracted by this system from various objects.

1 Introduction

A cognitive robot system should to be able to extract representations about its environment by exploration to enrich its internal representations and by this its cognitive abilities (see, e.g., [4]). The knowledge about the existence of objects and their shapes is of particular importance in this context. Having a model of an object that includes 3D information allows for the recognition and finding of poses of objects (see, e.g., [9]) as well as grasp planning (e.g. [1], [10]). However, extracting such representations of objects has shown to be very difficult. Hence many systems are based on CAD models or other manually achieved information. In this paper, we introduce a module that extracts multi–modal representations of objects by making use of the interaction of a grasping system with an early cognitive vision system (see Fig. 1 and [7]). After gaining physical control over an object (for example by making use of the object-knowledge independent grasping strategy in [2]) it is possible to formulate predictions about the change of rich feature description under the object motion induced by the robot.

If the motions of the objects within the scene are known, then the relation between features in two subsequent frames becomes deterministic (excluding the usual problems of occlusion, sampling, etc.). This means that a structure (e.g. in our case a contour) that is present in one frame is guaranteed to be in the previous and next frames (provided it does not become occluded or goes out of the field of view of the camera), subject a transformation that is fully determined by the motion: generally a change of position and orientation. If we assume that the motions are reasonably small compared to the frame–rate, then a contour will not appear or disappear unpredictably, but will have a life–span in the representation, between the moment it entered the field of view and the moment it leaves it (partial or complete occlusion may occur during some of the time–steps).
These prediction are relevant in different contexts

- **Establishment of objectness:** The objectness of a set of features is characterised by the fact that they all move according to the robot motion. This property is discussed in the context of a grounded AI planning system in [5].
- **Segmentation:** The system segments the object by its predicted motion from the other parts of the scene.
- **Disambiguation:** Ambiguous features can be characterised (and eliminated) by not moving according to the predictions.
- **Learning of object model:** A full 3D model of the object can be extracted by merging different views created by the motion of the end effector.

In this work, we represent objects as sets of multi–modal visual descriptors called 'primitives' covering visual information in terms of geometric 3D information (position and orientation) as well as appearance information (colour and phase). This representation is briefly described in section 2. The predictions based on rigid motion are described in section 3. The predictions are then used to track primitives over frames and to accumulate likelihoods for the existence of features (section 4). This is formulated in a Bayesian framework in section 4.3. In section 5, we finally show results of object acquisition for different objects and scenes.

2 Introducing visual primitives

The primitives we will be using in this work are local, multi–modal edge descriptors that were introduced in [8] (see figure 1). In contrast to the above mentioned features these primitives focus on giving a semantically and geometrically meaningful description of the local image patch. The importance of such a semantic grounding of features for a general purpose vision front–end, and the relevance of edge–like structures for this purposes were discussed in [3].
The primitives are extracted sparsely at locations in the image that are the most likely to contain edges. The sparseness is assured using a classical winner take all operation, insuring that the generative patches of the primitives do not overlap. Each of the primitive encodes the image information contained by a local image patch. Multi–modal information is gathered from this image patch, including the position $\boldsymbol{x}$ of the centre of the patch, the orientation θ of the edge, the phase ω of the signal at this point, the colour $\boldsymbol{c}$ sampled over the image patch on both sides of the edge, the local optical flow $\boldsymbol{f}$ and the size of the patch ρ. Consequently a local image patch is described by the following multi–modal vector:

$$\boldsymbol{\pi} = (\boldsymbol{x}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, \rho)^T, \tag{1}$$

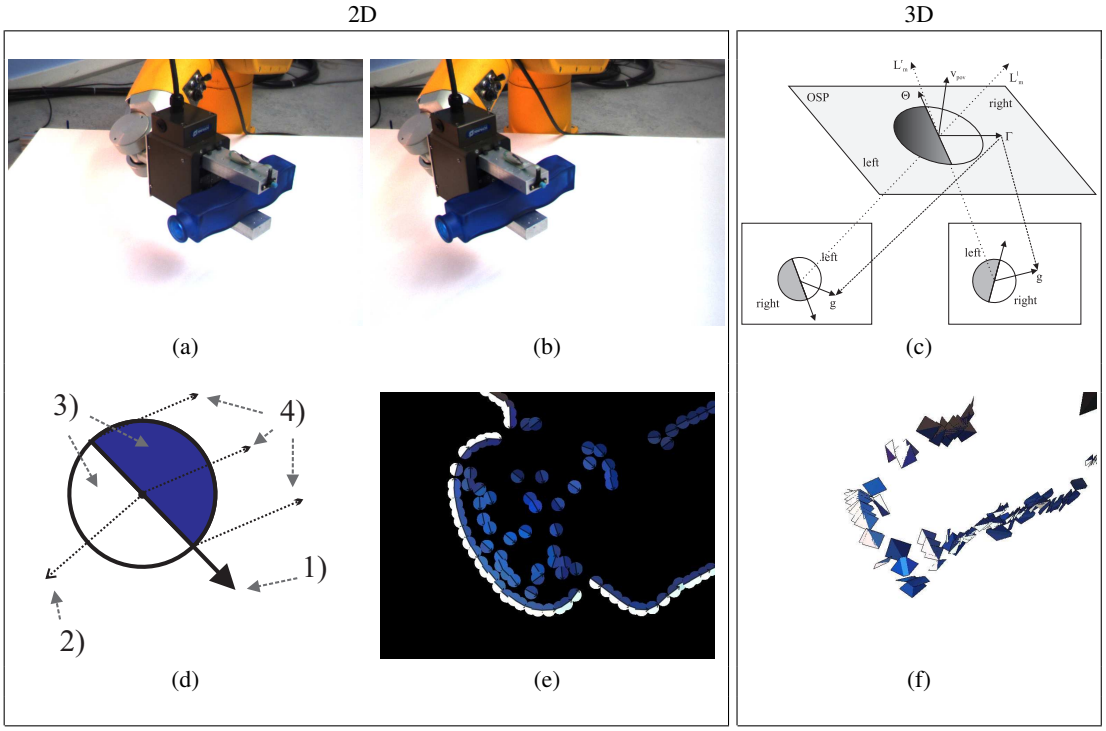


Fig. 1. Overview of the system. (a)-(b) images of the scene as viewed by the left and right camera at the first frame. (d) symbolic representation of a primitive: wherein 1) shows the orientation, 2) the phase, 3) the colour and 4) the optic flow of the primitive. (e) 2D–primitives of a detail of the object. (c) reconstruction of a 3D–primitive from a stereo–pair of 2D–primitives. (f) 3D–primitives reconstructed from the scene.

that we will name *2D primitive* in the following. The primitive extraction process is illustrated in Fig. 1.

Note that these primitives are of lower dimensionality than, e.g., SIFT (10 vs. 128) and therefore suffer of a lesser distinctiveness. Nonetheless, as shown in [11], they are distinctive enough for a reliable stereo matching if the epipolar geometry of the cameras is known. Furthermore, their semantic in terms of geometric and appearance based information allow for a good description of the scene content.

In a stereo scenario *3D primitives* can be computed from correspondences of 2D primitives (see Fig.1)

$$\boldsymbol{\Pi} = (\boldsymbol{X}, \boldsymbol{\Theta}, \Omega, \boldsymbol{C})^T, \tag{2}$$

where $\boldsymbol{X}$ is the position in space, $\boldsymbol{\Theta}$ is the 3D orientation, Ω is the phase of the contour and $\boldsymbol{C}$ is the colour on both sides of the contour. We have a projection relation

$$\mathcal{P} : \boldsymbol{\Pi} \rightarrow \boldsymbol{\pi} \tag{3}$$

linking 3D–primitives and 2D–primitives.

We call scene representation $\boldsymbol{\mathcal{S}}$ the set of all 3D–primitives reconstructed from a stereo–pair of images.

3 Making predictions from the Robot Motion

If we consider a 3D–primitive $\boldsymbol{\Pi}_i^t \in \boldsymbol{\mathcal{S}}_t$ part of the scene representation at an instant t, and assuming that we know the motion of the objects between two instants t and $t + \Delta t$, we can predict the position of the primitive in the new coordinate system of the camera at $t + \Delta t$.
Concretely, we predict the scene representation $\boldsymbol{\mathcal{S}}_{t+\Delta t}$ by moving the anterior scene representation ($\boldsymbol{\mathcal{S}}_t$) according to the estimated motion between instants t and $t + \Delta t$. The mapping $\mathcal{M}_{t \to t+\Delta t}$ associating the any entity in space in the coordinate system of the stereo set–up at time t to the same entity in the new coordinate at time $t + \Delta t$ is explicitly defined for 3D–primitives:

$$\hat{\boldsymbol{\Pi}}_i^{t+\Delta t} = \mathcal{M}_{t \to t+\Delta t}(\boldsymbol{\Pi}_i^t) \tag{4}$$

Assuming a scene representation $\boldsymbol{\mathcal{S}}_t$ is correct, and that the motion between two instants t and $t + \Delta t$ is known, then the moved representation $\hat{\boldsymbol{\mathcal{S}}}_{t+\Delta t}$ according to the motion $\mathcal{M}_{t \to t+\Delta t}$ is a *predictor* for the scene representation $\boldsymbol{\mathcal{S}}_{t+\Delta t}$ that can be extracted by stereopsis at time $t + \Delta t$.

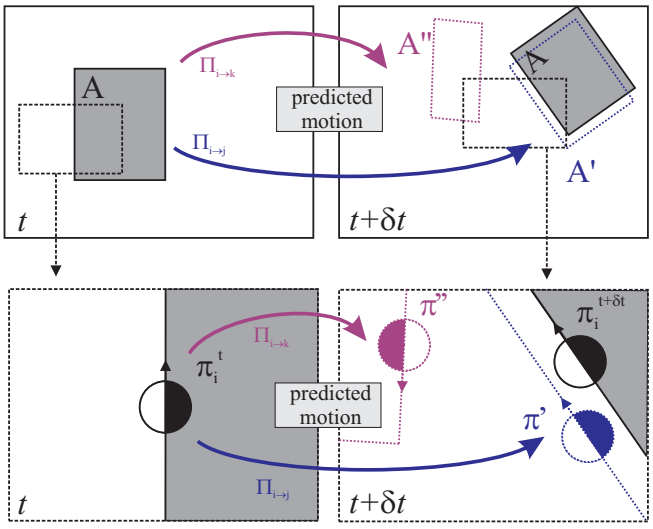


Fig. 2. Example of the accumulation of a primitive (see text).

Note that the predicted representation stems from the primitives extracted from the cameras at time t whereas the real scene representation is issued from primitives extracted at time $t + \Delta t$.
By extension, this relation also applies to the image representations reprojected onto each of the stereo image planes $\boldsymbol{\mathcal{I}}^F$, $F \in \{\text{left,right}\}$, defined by a projection $\mathcal{P}^F$:

$$\hat{\boldsymbol{\pi}}_i^{F,t+\Delta t} = \mathcal{P}^F\left(\mathcal{M}_{t \to t+\Delta t}(\boldsymbol{\Pi}_i^t)\right) \tag{5}$$

This prediction/verification process is illustrated in Fig. 2. The left column shows the image at time t whereas the right column shows the image at time $t + \Delta t$. The top row shows the complete image of the object and the bottom row shows

details of the object specified by the black rectangle. If we consider the object **A** with (solid rectangle in the top–left and top–right images) that between time t and $t + \Delta t$ according to a motion $M_{t \rightarrow t+\Delta t}$ Two hypotheses on the 3D shape of the object lead to two distinct predictions at time $t + \Delta t$: **A'** (correct and close to the actual pose of the object, blue rectangle in the top–right image) and **A"** (erroneous, red rectangle). In the bottom row, we study the case of a specific 2D–primitive $\boldsymbol{\pi}_i^t$ lying on the contour of **A** at the instant t (bottom–left image). If one consider that, at time t, there was two ambiguous stereo correspondences $\boldsymbol{\pi}_j^t$ and $\boldsymbol{\pi}_k^t$ then we have two mutually exclusive 3D reconstructions $\boldsymbol{\Pi}_{i \rightarrow j}^t$ and $\boldsymbol{\Pi}_{i \rightarrow k}^t$, each predicting a different pose at time $t + \Delta t$: 1) the correct hypothesis $\boldsymbol{\Pi}_{i \rightarrow j}^t$ predicts a 2D–primitive $\boldsymbol{\pi}'$ that matches with $\boldsymbol{\pi}_i^{t+\Delta t}$ (blue in the bottom–right image), one of the a 2D–primitive newly extracted at $t + \Delta t$ from the contour of **A**, comforting the original hypothesis; 2) when moving the incorrect hypothesis $\boldsymbol{\Pi}_{i \rightarrow k}^t$ we predict a 2D–primitive $\boldsymbol{\pi}''$ (red in the bottom–right image), that do not match any primitive extracted from the image, thereby revealing the erroneousness of the hypothesis.

Differences in viewpoint and pixel sampling lead to large variation in the primitives extracted and the resulting stereopsis. In other words, this means that the same contours of the scene will be described in the image representation, but by slightly shifted primitives, sampled at different points, along these contours. Therefore we need to devise a tracking algorithm able to recognise similar structures between heterogeneous representations.[4]

If a precise robot like the Staubli RX60 is used to move the objects the motion of the robot can be used to predict the primitive positions. Hereby it needs to be mentioned that the primitive position and orientation are usually represented in the camera coordinate system (placed in the left camera) while the robot movements are relative to the robot coordinate system (for the RX60 this is located at its first joint). To compute the mapping between the two coordinate systems we use a calibration procedure in which the robot end effector is moved to the eight positions of a virtual cube. At each location the position of the end effector in both coordinate systems are noted. The transformation between the two systems can then be computed by solving the overdetermined linear equation system represented by the eight positions. We use the RBM estimation algorithm described in [12] to do this.

4 Tracking 3D-primitives over time

In this section we will address the problem of integrating two heterogeneous scene representations, one extracted and one predicted that both describe the same scene at the same instant from the same point of view. The problem is three–fold: 1) comparing the two representations, 2) including the extracted primitives that were not predicted, and 3) re–evaluating the confidence in each of the primitives according to their predictability.

[4] We note here that the transformation described in this section does not describe the change of edges for a specific class of occlusions that occurs when round surfaces become rotated. In these cases the reconstructed edges do not move according to an RBM.

4.1 2D comparison

We propose to compare the two representations in the 2D image plane domain. This can be done by reprojecting all the 3D–primitives in the predicted representation $\hat{\boldsymbol{\mathcal{S}}}_{t+\Delta t}$ onto both image planes, creating two predicted image representations

$$\hat{\boldsymbol{\mathcal{I}}}^{F}_{t+\Delta t} = \mathcal{P}^{F}\left(\hat{\boldsymbol{\mathcal{S}}}_{t+\Delta t}\right), F \in \{\text{left,right}\} \tag{6}$$

Then both predicted image representations $\hat{\boldsymbol{\mathcal{I}}}^{F}_{t+\Delta t}$ can be compared with the extracted primitives $\boldsymbol{\mathcal{I}}^{F}_{t+\Delta t}$. For each predicted primitive $\hat{\boldsymbol{\pi}}_i$, a small neighbourhood (the size of the primitive itself) is searched for an extracted primitive $\boldsymbol{\pi}_j$ whose position and orientation are very similar (with a distance less than a threshold t_θ). Effectively a given prediction $\hat{\boldsymbol{\Pi}}_i$ is labelled as matched $\mu(\hat{\boldsymbol{\Pi}}_i)$ iff. for each image plane F defined by the projection $\mathcal{P}^F$ and having an associated image representation $\boldsymbol{\mathcal{I}}^F_t$, we have the projection $\boldsymbol{\pi}^F_i = \mathcal{P}^x(\boldsymbol{\Pi}_i)$ satisfy the following relation:

$$\exists \boldsymbol{\pi}_j \in \boldsymbol{\mathcal{I}}^F_t, \begin{cases} d_{\text{2D}}(\hat{\boldsymbol{\pi}}^F_i, \boldsymbol{\pi}_j) < r_{2D}, \\ d_{\boldsymbol{\Theta}}(\hat{\boldsymbol{\pi}}^F_i, \boldsymbol{\pi}_j) < t_{\boldsymbol{\Theta}} \end{cases} \tag{7}$$

with r_{2D} being the radius of correspondence search in pixels, $t_{\boldsymbol{\Theta}}$ being the maximal orientation error allowed for matching, d_{2D} stands for the two–dimensional Euclidian distance, and $d_{\boldsymbol{\Theta}}$ is the orientation distance. This is also illustrated in Fig. 2.

This 2D–matching approach has the following advantages: First, as we are comparing the primitives in the image plane, we are not affected by the inaccuracies and failures due to the 3D–reconstruction (see also [6]). Second, using the extracted 2D–primitives directly allows for 2D–primitives that could not be reconstructed at this time–step due to errors in stereo matching, etc.

4.2 Integration of different scene representations

Given two scene representations, one extracted $\boldsymbol{\mathcal{S}}_t$ and one predicted $\hat{\boldsymbol{\mathcal{A}}}_t$ we want to merge them into an accumulated representation $\boldsymbol{\mathcal{A}}_t$.

The application of the tracking procedure presented in section 4.1 provides a separation of the 3D–primitives in $\boldsymbol{\mathcal{S}}_t$ into three groups: confirmed, unconfirmed and not predicted.

The integration process consist into adding to the accumulated representation $\boldsymbol{\mathcal{A}}_{t-1}$, all 3D–primitives issued from the scene representation $\boldsymbol{\mathcal{S}}_t$ that are not matched by any 3D–primitive in $\boldsymbol{\mathcal{A}}_{t-1}$ (*i. e.* the non–predicted ones).

$$\boldsymbol{\mathcal{A}}_t = \boldsymbol{\mathcal{A}}_{t-1} \cup \boldsymbol{\mathcal{S}}_t \tag{8}$$

This allows to be sure that the accumulated representation always strictly include the newly extracted representation ($\boldsymbol{\mathcal{S}}_t \subseteq \boldsymbol{\mathcal{A}}_t$), and enables to include new information in the representation.

4.3 Confidence re–evaluation from tracking

The second mechanism allows to re–evaluate the confidence in the 3D–hypotheses depending on their resilience. This is justified by the continuity assumption, which

states that 1) any given object or contour of the scene should not appear and disappear in and out of the field of view (FoV) but move gracefully in and out according to the estimated ego–motion, and 2) that the position and orientation of such a contour at any point in time is fully defined by the knowledge of its position at a previous point in time and of the motion of this object between these two instants.

As we exclude from this work the case of independent moving object, and as the ego–motion is known, all conditions are satisfied and we can trace the position of a contour extracted at any instant t at any later stage $t + \Delta t$, as well as predict the instant when it will disappear from the FoV.

We will write the fact that a primitive $\boldsymbol{\Pi}_i$ that predicts a primitive $\hat{\boldsymbol{\Pi}}_i^t$ at time t is matched (as described above) as $\mu_t(\hat{\boldsymbol{\Pi}}_i)$ We define the tracking history of a primitive $\boldsymbol{\Pi}_i$ from its apparition at time 0 until time t as:

$$\boldsymbol{\mu}(\boldsymbol{\Pi}_i) = \left(\mu_t(\hat{\boldsymbol{\Pi}}_i), \mu_{t-1}(\hat{\boldsymbol{\Pi}}_i), \cdots, \mu_0(\hat{\boldsymbol{\Pi}}_i)\right)^T \tag{9}$$

thus, applying Bayes formula:

$$p\left(\boldsymbol{\Pi}_i|\boldsymbol{\mu}(\hat{\boldsymbol{\Pi}}_i)\right) = \frac{p\left(\boldsymbol{\mu}(\hat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}\right) p\left(\boldsymbol{\Pi}\right)}{p\left(\boldsymbol{\mu}(\hat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}\right) p\left(\boldsymbol{\Pi}\right) + p\left(\bar{\boldsymbol{\mu}}(\hat{\boldsymbol{\Pi}}_i)|\bar{\boldsymbol{\Pi}}\right) p\left(\bar{\boldsymbol{\Pi}}\right)} \tag{10}$$

where $\boldsymbol{\Pi}$ and $\bar{\boldsymbol{\Pi}}$ are correct and erroneous primitives, respectively.

Furthermore, if we assume independence between the matches we have, and assuming that $\boldsymbol{\Pi}$ exists since n iterations and has been matched successfully m times, we have:

$$\begin{aligned} p\left(\boldsymbol{\mu}(\hat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}\right) &= \textstyle\prod_t p\left(\mu_t(\hat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}\right) \\ &= p\left(\mu_t(\hat{\boldsymbol{\Pi}}_i) = 1|\boldsymbol{\Pi}\right)^m p\left(\mu_t(\hat{\boldsymbol{\Pi}}_i) = 0|\boldsymbol{\Pi}\right)^{n-m} \end{aligned} \tag{11}$$

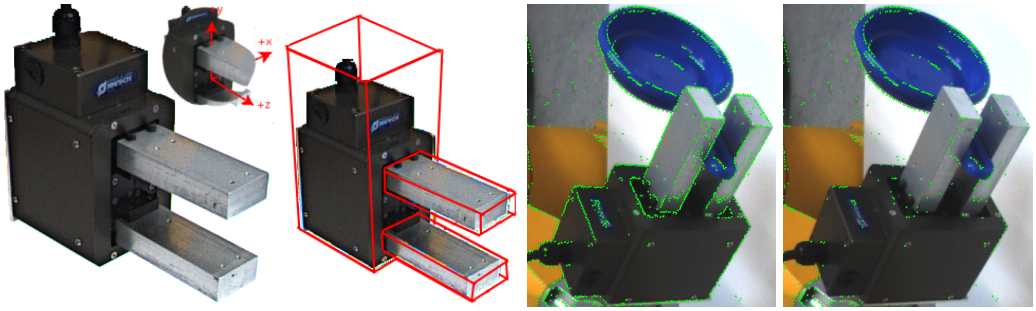
In this case the probabilities for μ_t are equiprobable for all t, and therefore we define the quantities $\alpha = p\left(\boldsymbol{\Pi}\right)$, $\beta = p\left(\mu_t(\hat{\boldsymbol{\Pi}}) = 1|\boldsymbol{\Pi}\right)$ and $\gamma = p\left(\mu_t(\hat{\boldsymbol{\Pi}}) = 1|\bar{\boldsymbol{\Pi}}\right)$ then we can rewrite (10) as follows:

$$p\left(\boldsymbol{\Pi}_i|\bar{\boldsymbol{\mu}}(\hat{\boldsymbol{\Pi}}_i)\right) = \frac{\beta^m(1-\beta)^{n-m}\alpha}{\beta^m(1-\beta)^{n-m}\alpha + \gamma^m(1-\gamma)^{n-m}(1-\alpha)} \tag{12}$$

We measured these prior and conditional probabilities using a video sequence with known motion and depth ground truth obtained via range scanner. We found values of $\alpha = 0.46$, $\beta = 0.83$ and $\gamma = 0.41$. This means that, in these examples, the prior likelihood for a stereo hypothesis to be correct is 46%, the likelihood for a correct hypothesis to be confirmed is 83% whereas for an erroneous hypothesis it is of 41%. These probabilities show that Bayesian inference can be used to identify correct correspondences from erroneous ones. To stabilise the process, we will only consider the n first frames after the appearance of a new 3D–primitive. After n frames, the confidence is fixed for good. If the confidence is deemed too low at this stage, the primitive is forgotten. During our experiments $n = 5$ proved to be a suitable value.

4.4 Eliminating the grasper

The end-effector of the robot follows the same motion as the object. Therefore, this end-effector becomes extracted as well. Since we know the geometry of this end-effector (Figure 3 (a)), we can however easily subtract it by eliminating the 3D primitives that are inside the bounding boxes that bounds the body of the gripper and its fingers (Figure 3 (b)). For this operation, three bounding boxes are calculated in grasper coordinate system (GCS) by using the dimensions of grasper. Since the 3D primitives are in robot coordinate system (RCS), the transformation from RCS to GCS is applied to each primitive and if the resultant coordinate is inside any of the bounding boxes, the primitive is eliminated. In Figure 3 (c) 2D projection of 3D primitives extracted from a stereo pair is presented. After gripper elimination, 2D projection of remaining primitives are shown in Figure 3 (d).



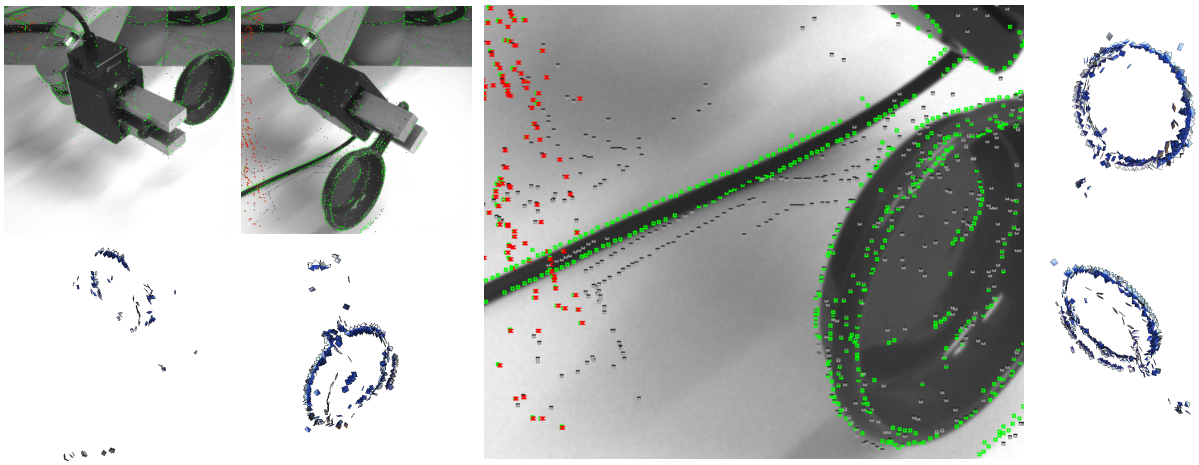
(a) (b) (c) (d)

Fig. 3. Gripper elimination (a) grasper and grasper coordinate system (b) bounding boxes of grasper body and its fingers (c) primitives before grasper elimination (d) primitives after grasper elimination

5 Results and Conclusion

We applied the accumulation scheme to a variety of scenes where the robot arm manipulated several objects. The motion was a rotation of 5 degrees per frame. The accumulation process on one such object is illustrated in Fig. 4. The top row show the predictions at each frame. The bottom row, shows the 3D–primitives that were accumulated (frames 1, 12, 22, and 32). The object representation becomes fuller over time, whereas the primitives reconstructed from other parts of the scene are discarded. Figure 5 shows the accumulated representation for various objects. The hole in the model corresponds to the part of the object occluded by the gripper. Accumulating the representation over several distinct grasps of the objects would yield a complete representation.

Conclusion: In this work we presented a novel scheme for extracting object model from manipulation. The knowledge of the robot's arm motion gives us two precious information: 1) it enables us to segment the object from the rest of the scene; and 2) it allows to track object features in a robust manner. In combination with the visually induced grasping reflex presented in [2], this allows for an



(a) (b) (c) (d)

Fig. 4. Birth of an object (a)-(b) top:2D projection of the accumulated 3D representation and newly introduced primitives, bottom:accumulated 3D representation. (c) newly introduced and accumulated primitives in detailed. Note that, the primitives that are not updated are red and the ones that have low confidence are grey (d) final accumulated 3D representation from two different poses.

Fig. 5. Objects and their related accumulated representation.

exploratory behaviour where the robot attempts to grasp parts of its environment, examine all successfully grasped shapes and learns their 3D model and by this becomes an important submodule of the cognitive system discussed in [5].

Acknowledgement: This paper has been supported by the EU-Project PACOplus (2006-2010).

References

1. C. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3D objects. In *IEEE International Conference on Robotics and Automation*, pages 1890–1896, Detroit, Michigan, May 1999.
2. J. Sommerfeld D. Aarno, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early reactive grasping with second order 3d feature relations. *IEEE Conference on Robotics and Automation (submitted)*, 2007. submitted.
3. James H. Elder. Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122, 1999.
4. P. Fitzpatrick and G. Metta. Grounding Vision Through Experimental Manipulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 361:2165 – 2185, 2003.
5. Ch. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger, and F. Wörgötter. Object action complexes as an interface for planning and robot control. *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, 2006.
6. R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.
7. N. Krüger, M. Van Hulle, and F. Wörgötter. Ecovision: Challenges in early-cognitive vision. *International Journal of Computer Vision*, accepted.
8. N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behavious, AISB Journal*, 1(5):417–427, 2004.
9. D.G. Lowe. Three–dimensional object recognition from single two images. *Artificial Intelligence*, 31(3):355–395, 1987.
10. A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1824–1829, 2003.
11. N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal scene reconstruction using perceptual grouping constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*, 2006.
12. B. Rosenhahn, O. Granert, and G. Sommer. Monocular pose estimation of kinematic chains. In L. Dorst, C. Doran, and J. Lasenby, editors, *Applied Geometric Algebras for Computer Science and Engineering*, pages 373–383. Birkhäuser Verlag, 2001.

Multi–modal Scene Reconstruction using Perceptual Grouping Constraints

5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision

Nicolas Pugeault
University of Edinburgh
npugeaul@inf.ed.ac.uk

Florentin Wörgötter
University Göttingen
worgott@chaos.gwdg.de

Norbert Krüger
Aalborg University Copenhagen
nk@imi.aau.dk

Abstract

In this work we propose a scheme integrating perceptual grouping into stereopsis to reduce the ambiguity of those early processes. We propose a simple perceptual grouping algorithm that – in addition to the geometric information – makes use of a novel multi–modal affinity measure between local primitives. We then use this group information to 1) disambiguate the stereopsis by enforcing that stereo matches preserve groups; and 2) correct the reconstruction error due to the image pixel sampling using a linear interpolation over the groups. We show quantitative and qualitative demonstrations of those processes on a variety of sequences.

1. Introduction

We propose in this paper an approach using feedback between two mid–level processes, namely perceptual grouping and stereopsis to reduce the ambiguity omnipresent at this level of processing. We base our framework on a novel image representation based on multi–modal local image descriptors called *primitives*, introduced by [21] and applied to stereo by [20]. In this work, we will focus on primitives describing line structures, and we propose a perceptual grouping mechanism which makes use of this rich multi–modal information.

Perceptual grouping can be divided in two tasks: 1) defining an affinity measure between primitives and use it to build a graph of the connectedness between the primitives, and 2) extracting groups, which are the connected components of this graph. We will only define the affinity measure between primitives, and not extract the groups themselves explicitly, as we only need the local grouping information for a primitive to apply the correction mechanisms we propose in this paper. Similar affinity measures have been proposed by [27, 26], which formalised a good continuation constraint, or [9] which included the intensity on each side of the curve into a Bayesian formulation of grouping. Yet in this paper we propose a multi–modal similarity measure, composed of phase, colour and optical flow measurement, and combine it with a classical good continuation criterion forming a novel multi–modal definition of the affinity between primitives. Note that an explicit description of the groups could be extracted easily using a variety of techniques including: normalised [34] or average cuts [32], affinity normalisation [27], dynamic programming [33], *etc*.

The interest of using perceptual organisation in the spatial and temporal domains has been outlined by [31]. Here, we will study how this perceptual grouping information can be used to disambiguate stereopsis and 3D reconstruction using primitives. If we assume that a contour of the image is likely to be a projection of a contour of the 3D scene, then we can expect each 3D contour of the scene to project as a 2D contour on each camera plane (except in the case of occlusion). Conversely, this also implies that any contour in one image has a corresponding contour in the second image (or it is occluded). Thus we will propose an *external* stereo confidence which estimates how well primitives that are part of the same group agree with a putative stereo–match. This allows to discard a large number of potential stereo–correspondences hence reducing the ambiguity of the stereo matching and of the scene reconstruction processes.

We will test this scheme with four different calibrated stereo sequences, illustrated in figure 1. For sequences (a) (b) and (c) we have depth values obtained from a range scanner. Ten different frames from those three sequences were used for quantification in this paper. Sequence (d) was recorded outdoors in a moving car. for which we will show qualitative results.

The novel contributions of this paper are

- a 2D grouping that uses geometric and appearance based information,
- using the 2D grouping for improving stereo matching from a very local level (in contrast to, e.g., [30], where more elaborate features, like ribbons, were considered),

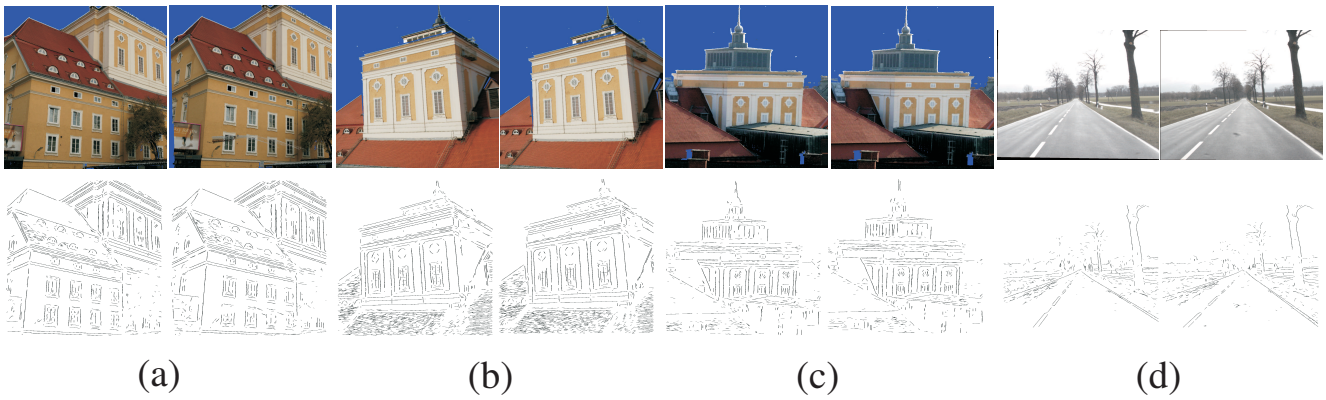


Figure 1. The four sequences on which we tested our approach.

- applying an interpolation method that leads to more reliable estimates of 3D position and 3D-orientation.

The grouping is part of an early cognitive vision framework including ego-motion estimation and temporal accumulation (for an outline see [37]).

The paper is structured as follows: Section 2 will present the image primitives on which we are basing our processing. In section 3, we define the affinity between two primitives. In section 4 we present a stereo–matching process based on primitives similar to [20]. Then in section 5 we propose a simple scheme to 1) increase the reliability of matching and 2) smooth the reconstruction of a stereo sequence using information gained from the perceptual grouping defined earlier.

2. 2D–primitives

Numerous feature detectors exist in the literature (see [22] for a review). Each feature based approach can be divided into an interest point detector (e.g. [3, 4]) and a descriptor describing a local patch of the image at this location, that can be based on histograms (e.g. [6, 22]), spatial frequency [28], local derivatives [15, 13, 1] steerable filters [36], or invariant moments ([23]). In [22] these different descriptors have been compared, showing a best performance for SIFT-like descriptors.

The primitives we will be using in this work are local, multi–modal edge descriptors that were introduced in [21]. In contrast to the above mentioned features these primitives focus on giving a semantically and geometrically meaningful description of the local image patch. The importance of such a semantic grounding of features for a general purpose vision front–end, and the relevance of edge–like structures for this purposes were discussed in [7].

The edge map and the local phase are computed using the monogenic signal (see [11]), although some other kind of filtering could alternatively be used (e.g., steerable filters [36]). The primitives are extracted sparsely at locations in the image that are the most likely to contain edges. This likelihood is computed using the intrinsic dimensionality measure proposed in [19]. The sparseness is assured using a classical winner take all operation, insuring that the generative patches of the primitives do not overlap. Each of the primitive encodes the image information contained by a local image patch of a same size ρ as the kernel used by the filtering operation. Multi–modal information is gathered from this image patch, including the position $\boldsymbol{m}$ of the centre of the patch, the orientation θ of the edge, the phase ω of the signal at this point, the colour $\boldsymbol{c}$ sampled over the image patch on both sides of the edge and the local optical flow $\boldsymbol{f}$, computed using the classical Nagel algorithm (see [25]). Consequently a local image patch is described by the following multi–modal vector:

$$\boldsymbol{\pi} = (\boldsymbol{m}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, \rho)^T \tag{1}$$

that we will name *primitive* in the following. The set of primitives describing the stereo images is called *image representation* and written $\boldsymbol{\mathcal{I}}^l$ and $\boldsymbol{\mathcal{I}}^r$ for the images from the left and right camera. The image representation extracted from one image is illustrated in figure 2.

Note that these primitives are of lower dimensionality than, e.g., SIFT (10 vs. 128) and therefore suffer of a lesser distinctiveness. Nonetheless, we will show in section 4 that they are distinctive enough for a reliable stereo matching if the epipolar geometry of the cameras is known. Advantageously, the rich information carried by the 2D–primitives can be reconstructed in 3D, providing a more complete scene representation. Having geometrical meaning for the primitive allows to describe the relation between proximate primitives in terms of perceptual grouping.

3. Perceptual Grouping of 2D–Primitives

Decades ago, the Gestalt psychologists proposed a series of axioms describing the way the human visual system binds together features in an image (see [16, 35, 17]). This process is generally called *perceptual grouping* the Gestalt psychologists proposed that it was driven proper-

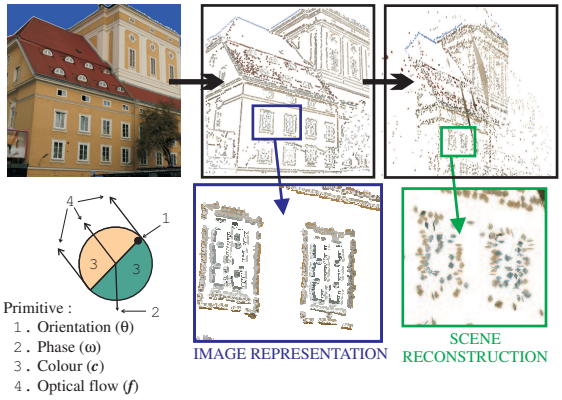


Figure 2. Illustration of the primitive extraction process from a video sequence. The figure shows one image from the sequence (a) from figure 1, on the right, then the 2D–primitives extracted from this image (see section 2), and finally the 3D–primitives reconstructed from the stereo–matches as described in section 4. The bottom row shows a description of the graphic representation of the 2D–primitives, as well as a magnification of the image representation and the reconstructed entities. Note that the structure reconstructed is quite far from the cameras, leading to a certain imprecision in the reconstruction of the 3D–primitives. We will propose a simple scheme addressing this problem in section 5.3

ties like proximity, good continuation, similarity, symmetry, amongst others. More recently, psychophysical experiments measured the impact of different cues for perceptual grouping (see, e.g., [12]). Furthermore, Brunswik and Kamiya [2] proposed that those processes should be related to statistics of natural images, which has been recently confirmed by several studies [18, 8, 14].

We previously defined the primitives as local edge descriptors, and that a group of primitives describe a contour of the image. The Gestalt rule of *proximity* implies that primitives that are closer to one another are most likely to lie on the same contour. According to the Gestalt rule of *good continuation*, we will consider that contours in the image are smooth, and therefore that two proximate primitives in a group will be nearly either collinear or co–circular. In this formulation, a strong inflexion in a contour will lead this contour to be described as *two* groups joining at the inflection point. Furthermore the position and orientation of primitives that are part of a group are the local tangents to the contour described by this group. Finally, the rule of *similarity* states that primitives that are similar (in terms of the colour, phase and optical flow modalities) are most likely to be grouped together. Also, we would expect such properties as colour on both side of a contour to change smoothly along this contour.

The two first cues are joined into a *Geometric constraint* that we describe in section 3.1 and the multi–modal similarity cue is detailed in section 3.2. These two measures are combined into an overall affinity measure that we describe in section 3.3.

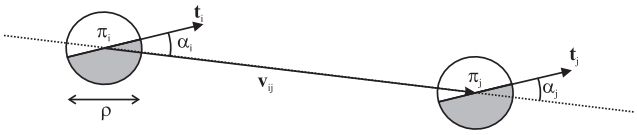


Figure 3. Illustration of the values used for the collinearity computation. If we consider two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, then the vector between the centres of these two primitives is written $\boldsymbol{v}_{ij}$, and the orientations of the two primitives are designated by the vectors t_i and t_j, respectively. The angle formed by $\boldsymbol{v}_{ij}$ and t_i is written α_i, and between $\boldsymbol{v}_{ij}$ and t_j is written α_j. ρ is the radius of the image patch used to generate the primitive.

3.1. Geometric constraint

If we consider two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ in $\mathcal{I}$, then the likelihood that they both describe the same contour can be formulated as a combination of three basic constraints on their relative position and orientation — see figure 3.

Proximity (c_p []):

$$c_p\left[g_{i,j}\right] = 1 - e^{-\max\left(1-\frac{||\mathbf{v_{i,j}}||}{\rho\tau},0\right)} \tag{2}$$

Here, ρ stands for the radius of the the primitives in pixels. $\rho\tau$ is the size of the neighbourhood considered in pixels. $\|\mathbf{v_{i,j}}\|$ is the distance in pixels separating the centres of the two primitives.

Collinearity (c_{co} []):

$$c_{co}\left[g_{i,j}\right] = 1 - \left|\sin\left(\frac{|\alpha_i| + |\alpha_j|}{2}\right)\right| \tag{3}$$

Here α_i and α_j are the angles between the line joining the two primitives centres and the orientation of, respectively, $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$.

Co–circularity (c_{ci} []):

$$c_{ci}\left[g_{i,j}\right] = 1 - \left|\sin\left(\frac{\alpha_i + \alpha_j}{2}\right)\right| \tag{4}$$

The combination of those three criteria forms the *geometric* affinity measure:

$$\mathbf{G}_{i,j} = \sqrt[3]{c_e\left[g_{i,j}\right] \cdot c_{co}\left[g_{i,j}\right] \cdot c_{ci}\left[g_{i,j}\right]} \tag{5}$$

where $\mathbf{G}_{i,j}$ is the geometric affinity between two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. This affinity represent the likelihood for a curve having for tangents those two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_i$ to be an actual contour of the scene.

3.2. Multi–modal Constraint

Effectively, the more similar are the modalities between two primitives, the more likely are those two primitives to lie on the same contour. Note that [8] already proposed to use the intensity as a cue for perceptual grouping, yet here

we use a combination of the phase, colour and optical flow modalities of the primitives to decide if they describe the same contour:

$$\mathbf{M}_{i,j} = 1 - w_\omega d_\omega\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right) - w_{\boldsymbol{c}} d_{\boldsymbol{c}}\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right) - w_{\boldsymbol{f}} d_{\boldsymbol{f}}\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right) \tag{6}$$

where d_ω is the phase distance, $c_{\boldsymbol{c}}$ the colour distance and $c_{\boldsymbol{f}}$ the optical flow distance between the two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. These metrics are similar to the ones used in [29, 20]. w_ω, $w_{\boldsymbol{c}}$ and $w_{\boldsymbol{f}}$ are the relative weight of the modalities, such that $w_\omega + w_{\boldsymbol{c}} + w_{\boldsymbol{f}} = 1$.

3.3. Primitive Affinity

The overall affinity between all primitives in an image is formalised as a matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{i,j}$ holds the affinity between the primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. We define this affinity from equations (5) and (6), such that 1) two primitives complying poorly with the good continuation rule have an affinity close to zero; and 2) two primitives complying with the good continuation rule yet strongly dissimilar will have only an average affinity. The affinity is formalised as follows:

$$c\left[g_{i,j}\right] = \boldsymbol{A}_{i,j} = \sqrt{\mathbf{G}\left(\alpha \mathbf{G}_{i,j} + (1-\alpha)\mathbf{M}_{i,j}\right)} \tag{7}$$

where α is the weighting of geometric and multi–modal (*i.e.* phase, colour and optical flow) information in the affinity. A setting of $\alpha = 1$ implies that only geometric information (proximity, collinearity and co-circularity) is used, while $\alpha = 0$ indicates that geometric and multi–modal information are evenly mixed. The groups generated for the left and right frames for each sequence are drawn in figure 1, bottom row. Dark lines describe strings of grouped primitives. One can see in those images that the major contours of the images are adequately described.

4. Stereopsis using 2D–primitives

Classical stereopsis allows reconstructing a 3D point from two corresponding stereo points. A review of stereo–algorithms was presented in [24], dense two frames stereo algorithms were also compared in [5]. In these papers the different algorithms were compared on mainly artificial images, with a disparity d that ranges in $0 \leq d \leq 16$. In this work we make use of a sparse, feature based representation, applied on high resolution video sequences of natural scenes, where the ground truth was obtained using a range scanner. The allowed disparity range for these scenes is $0 \leq d \leq 200$, leading to a comparable level of ambiguity (*i.e.* between 10 and 20 candidates depending on the primitive being matched).

The stereopsis used for this paper is a simple local winner–take–all scheme: all primitives in the right image that lie on the epipolar line are *potential correspondences* and their individual likelihood is set as their multi–modal similarity with the original primitive in the left image. Then the most similar primitive is taken as the most likely correspondence. The multi–modal distance between two primitives is defined as a linear combination of the modal distances between the two primitives:

$$d_m(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \sum_m w_m d_m(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) \tag{8}$$

where w_m is the relative weighting of the modality m, with $\sum_m w_m = 1$ (we use distance functions for the modalities that are similar to the ones proposed in [29, 20]).

In figure 6(a) the ROC curves showing the performance of the stereo–matching when using as likelihood estimation the similarities in each of the modalities held by a primitive, alongside with the performance of the multi–modal distance proposed in equation (8). We can see that: 1) all modalities offer a discrimination better than chance between correct and erroneous correspondences; and 2) the multi–modal distance offers a better discrimination than the individual modalities. In this figure we can see that the colour modality is a particularly strong discriminant for stereopsis. This is explained by the fact that the hue and saturation are sampled on each side of the edge, leading to a 4–dimensional modality, where phase and orientation are only 1–dimensional and optical flow is 2–dimensional (albeit the aperture problem reduces it to one effective dimension: the normal flow). On the other hand the poor performance of the optic flow modality could be explained by the relative simplicity of the motion in this scene: a pure forward translation of the camera, with no moving object. Therefore, we would expect the performance of individual modalities to vary depending on the scenario, and the robustness of the multi–modal constraint could be further enhanced by a contextual weighting. Nevertheless, in a variety of scenarios the use of a static weighting proved robust enough to obtain reliable stereopsis.

Moreover, by making use of the rich semantic information carried by the primitives, the stereopsis yield a set of geometrically meaningful entities rather than an mere disparity map We call the reconstructed entities 3D–primitives $\boldsymbol{\Pi}$:

$$\boldsymbol{\Pi} = (\boldsymbol{M}, \boldsymbol{\Theta}, \Omega, \boldsymbol{C})^T \tag{9}$$

where $\boldsymbol{M}$ is the location in space, $\boldsymbol{\Theta}$ is the 3D orientation of the edge, Ω is the phase across this edge, and $\boldsymbol{C}$ holds the colour information for this edge — see attached material. In figure 7(a) we show the 3D–primitives that were reconstructed after a stereo–matching based on the multi–modal confidence from equation (8).

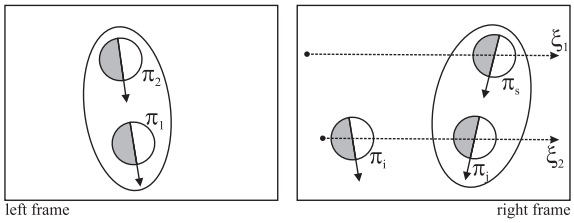


Figure 4. The BSCE criterion: Let $\boldsymbol{\pi}_1$ be a primitive in the left frame forming a group with a second primitive $\boldsymbol{\pi}_2$. $\boldsymbol{\pi}_2$ has a stereo correspondence $\boldsymbol{\pi}_s$ in the right image. Both $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ in the right image lie on the epipolar line ξ_1 of $\boldsymbol{\pi}_1$; hence these two primitives are both putative correspondences of $\boldsymbol{\pi}_1$. Furthermore, the primitive $\boldsymbol{\pi}_i$ is clearly the most similar to $\boldsymbol{\pi}_1$ (due to a closer orientation), hence this stereo–correspondence $s_{1\to i}$ yield a higher multi–modal confidence than would, e.g. $s_{1\to j}$. Yet, when considering the BSCE criterion we realise that only the putative correspondence $\boldsymbol{\pi}_j$ forms a group $g_{j,s}$ with $\boldsymbol{\pi}_s$, conserving the group relation $g_{1,2}$ between $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$.

5. Perceptual Grouping Constraints to Improve Stereopsis

In addition to their richness, primitives are very redundant along contours, and this redundancy allows us to use perceptual grouping to derive the following two constraints for the matching process:

Isolated primitives are likely to be unreliable: As primitives are extracted redundantly along the contours, conversely an isolated primitive is likely to be an artifact. Hence isolated primitives can be neglected.

Stereo consistency over groups: If a set of primitives forms a contour in the first image, the *correct correspondences* of these primitives in the second image also form a contour.

5.1. Basic Stereo Consistency Event (BSCE)

As explained in section 3, 2D–primitives represent local estimators of image contours. A constellation of those 2D–primitives describe the contour as a whole. Those contours are consistent over stereo, with the notable exception of partially occluded contours — see figure 1, bottom row. Hence, if two primitives describe a contour in one image then their correspondences in the second image should also describe the same contour, and those two 2D contours are the projection of the same 3D contour onto the two different optical planes. In section 3, we defined the likelihood for two primitives to describe the same contour as the affinity between these two primitives, hence we can rewrite the previous statement as:

Given two primitives $\boldsymbol{\pi}_i^l$ and $\boldsymbol{\pi}_j^l$ in $\boldsymbol{\mathcal{I}}^l$ and their respective correspondences $\boldsymbol{\pi}_n^r$ and $\boldsymbol{\pi}_p^r$ in a second image $\boldsymbol{\mathcal{I}}^r$; if $\boldsymbol{\pi}_i^l$ and $\boldsymbol{\pi}_j^l$ belongs to the same group in $\boldsymbol{\mathcal{I}}^l$ then $\boldsymbol{\pi}_n^r$ and $\boldsymbol{\pi}_p^r$ should also be part of a group in $\boldsymbol{\mathcal{I}}^r$. — see figure 4.

We call the conservation of the link between a pair of primitives in the stereo–correspondences of those primitives the *Basic Stereo Consistency Event* (BSCE).

This condition can then be used to test the validity of a stereo–hypothesis. Consider a primitive $\boldsymbol{\pi}_i^l$, and a stereo hypothesis:

$$s_{i\to n} : \boldsymbol{\pi}_i^l \to \boldsymbol{\pi}_n^r \tag{10}$$

and consider a neighbour $\boldsymbol{\pi}_j^l \in N(\boldsymbol{\pi}_i^l)$ of $\boldsymbol{\pi}_i^l$ such that the two primitives share an affinity $c\,[g_{i,j}]$. For this second primitive a stereo–correspondence $\boldsymbol{\pi}_p^n$ with a confidence of $c\,[s_{j\to p}]$ exists. We can then estimate how well the stereo–hypothesis $s_{i\to n}$ preserves the BSCE:

$$E(g_{i,j}, s_{i\to n}) = \begin{cases} \sqrt{c\,[s_{j\to p}] \cdot c\,[g_{i,j}]} & if\ c\,[g_{n,p}] > \varepsilon \\ -\sqrt{c\,[s_{j\to p}] \cdot c\,[g_{i,j}]} & else \end{cases} \tag{11}$$

In other words, considering a stereo–pair of primitives: the BSCE of a primitive in the first image with one of its neighbour is high if they share a strong affinity and if this second primitive creates a stereo–hypothesis such that the correspondences in the second image of both primitives *also* share a strong affinity. It is low if the stereo–correspondence of this primitive and the stereo–correspondences of other primitives part of the same group, do not form a group in the other image. This naturally extends the concept of group as defined in section 3 into the stereo domain.

5.2. Neighbourhood Consistency Confidence

Building on the formula (11), we can define how *the whole neighbourhood* of a primitive is consistent with a given stereo hypothesis.

The previous formula tells us how a 2D–primitive stereo correspondence is consistent with our knowledge of the set of stereo hypotheses for a second 2D–primitive, in its neighbourhood. Now, if we consider a primitive $\boldsymbol{\pi}_i^l$ and an associated stereo–correspondence $s_{i\to n}$, we can integrate this BSCE confidence over the neighbourhood of the primitive $\boldsymbol{\mathcal{N}}_i^l$ — as defined in section 3.3.

$$c_{ext}[s_{i\to n}] = \frac{1}{\#\boldsymbol{\mathcal{N}}_i^l} \sum_{\boldsymbol{\pi}_k^l \in \boldsymbol{\mathcal{N}}_i^l} E(\boldsymbol{\pi}_1^l, \boldsymbol{\pi}_k^l, s_{i\to n}) \tag{12}$$

Where $\#\boldsymbol{\mathcal{N}}_i^l$ is the size of the neighbourhood — *i.e.* the number of neighbours of $\boldsymbol{\pi}_1^l$ considered. We call this new confidence the *external confidence* in $s_{i\to n}$, as opposed to the internal confidence given by the multi–modal similarity between the 2D–primitives— equation (8). In figure 5, one can see that the correct correspondences have mostly positive external confidences, while incorrect ones have mainly negative values. Therefore, applying a threshold on the external confidence will remove stereo hypotheses that are inconsistent with their neighbourhood, and thus reduce the ambiguity of the stereo–matching. Note that selecting a

threshold higher than zero implies the removal of all the isolated primitives (as an isolated primitive has an external confidence of zero by definition).

Figure 6(b) shows ROC curves of the performance for varying thresholds on the multi–modal similarity. Each of the curve drawn shows the performance for different thresholds (respectively threshold values of $-0.6, -0.3, 0, +0.3$, and without threshold) applied to the external confidence prior to the ROC analysis. We can see from those results that applying a bias on the decision based on the external confidence is improving significantly the accuracy of the decision process. Depending on the type of selection process desired — very selective and reliable, or more lax, but yielding a denser set of correspondences — another threshold can be chosen. The best overall improvement seems to be reached for a threshold of -0.3 over the external confidence. Nonetheless, when we consider a case where very high reliability is required, a threshold of 0 (meaning discarding all primitives which are part of no group) might be preferred. Note that when a threshold is applied to the external confidence prior to the ROC analysis, the resulting curve do not reach the $(1, 1)$ point of the graph. This is normal as the threshold already remove some stereo–hypotheses even before the multi–modal confidence is considered.

The 3D–primitives reconstructed after such a scheme are shown in figure 7(b).

5.3. Interpolation in Space

One issue when reconstructing 3D structures from stereopsis is that the accuracy of the reconstructed entities is decreasing with the distance to the cameras, due to the pixel sampling of the images — see [10]. Figure 7(b) shows the reconstruction of the tree (along with the road markings) in sequence (d) — see figure 1. There we can see that, although all primitives describe the contour of the tree from the same point of view, their exact position and orientation in space vary, and they certainly do not form a contour in space.

Yet, we do know that the 2D–primitives they are reconstructed from a group in both stereo images (*c.f.* section 5 and figure 1 bottom row), and as such that they form a smooth continuous contour. Hence we can assume that they are the projection on the image planes of a smooth and continuous contour of the scene (except in some extreme cases and under rare viewpoints), and as such that the reconstructed 3D–primitives should also describe such a curve.

A common way of reducing such noise in the sampling of a smooth function is to use linear smoothing, hence we propose to apply it to the 3D–primitives. For each iteration n of this smoothing, the position $\boldsymbol{M}$ and orientation $\boldsymbol{\Theta}$ of the primitive $\boldsymbol{\Pi}_i^{(n)}$ are changed to the average between their previous values $\boldsymbol{\Pi}_i^{(n-1)}$ and values interpolated from the primitives reconstructed out of the two closest neighbours of the 2D–primitive in the images $I(\boldsymbol{\Pi}_j^{(n-1)}, \boldsymbol{\Pi}_k^{(n-1)})$.

$$\boldsymbol{M}_i^{(n)} = \frac{1}{2}\left(\boldsymbol{M}_i^{(n-1)} + I(\boldsymbol{M}_j^{(n-1)}, \boldsymbol{M}_k^{(n-1)})\right) \quad (13)$$

$$\boldsymbol{\Theta}_i^{(n)} = \frac{1}{2}\left(\boldsymbol{\Theta}_i^{(n-1)} + I(\boldsymbol{\Theta}_j^{(n-1)}, \boldsymbol{\Theta}_k^{(n-1)})\right) \quad (14)$$

Figure 7 illustrate the reconstructed 3D–primitives from the sequence (d) (*c.f.* figure 1). Note that it is necessary to choose a point of view sufficiently different from the one of the camera to highlight the reconstruction errors, while being sufficiently similar for the shapes of the scene to be recognisable. We chose a point of view located high on the right side of the scene, looking downwards at the road.

When comparing figures 7(a) and 7(b) we can see that a large number of outliers are discarded from the reconstructed 3D–primitives, leading to a cleaner description of the scene. Figure 7(c) shows the same part of the scene (d) after 3 iterations of the linear smoothing. The 3D–primitives forming the contour of the tree and the road markings are now smoothly aligned.

6. Conclusion

In this paper we defined an affinity relation between image primitives making use of the rich multi–modal information available. Therefore the resulting affinity measure encompass more than just the good continuation cue but also continuity in phase, colour and optical flow. We have illustrated that, on varied sequence, the resulting groups follow adequately the contours of the image. In a second part we proposed a simple measure of the conservation of those groups, and hence of the neighbourhood structure of a primitive, across stereo. Using this conservation we could formalise a contextual estimation of the likelihood of a stereo correspondence. We show that using this new external confidence measure in conjunction with a similarity measure we can improve significantly the performance of the stereo–matching process. Furthermore, we show that interpolation can be used over a group to correct the smoothness of the reconstructed representation.

Acknowledgement: We thank the company Riegl for the images with known ground truth used for sequence (a), (b) and (c). This work described in this paper was part of the European project ECOVISION.

References

[1] A. Baumberg. Reliable Feature Matching across Widely Separated Views. In *Proc. Conf. Computer Vision and Patter Recognition*, pages 774–781, 2000. 2

[2] E. Brunswick and J. Kamiya. Ecological cue validity of 'proximity' and other gestalt factors. *Journal of Psychology*, 66:20–32, 1953. 3

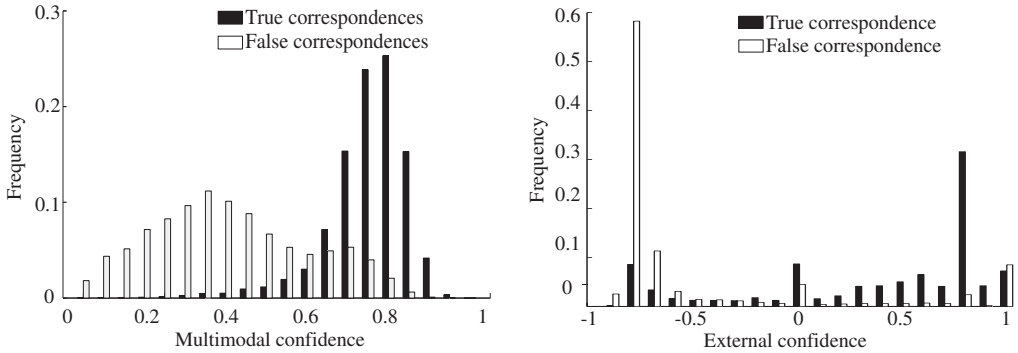
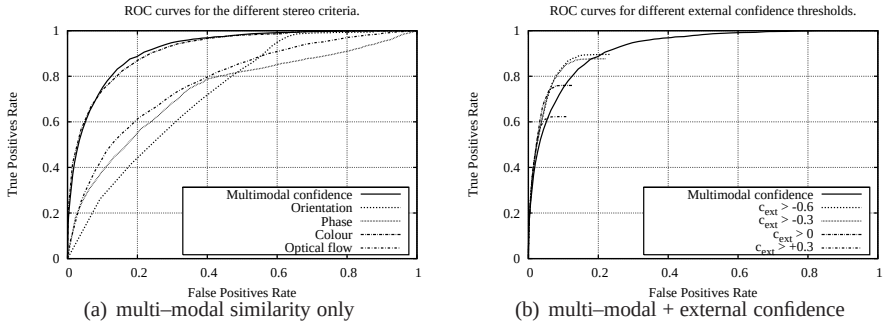


Figure 5. Distribution of multi–modal similarity and external confidence for correct (black bars) and false (white bars) correspondences. These data have been collected over 10 frames of the sequences (a), (b) and (c) — see figure 1.



(a) multi–modal similarity only (b) multi–modal + external confidence

Figure 6. ROC curves for the performance of the multi–modal confidence to discriminate correct from erroneous correspondences. (a) Comparisons of the different modalities for stereo–matching (see for a discussion of the role of colour in the text). (b) Each curve stands for a the application of a different threshold over the external confidence, prior to the ROC analysis. Those curves represent the statistics over 10 frames of the two sequences with ground truth — see figure 1.

[3] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Proceedings of Alvey Conference*, pages 189–192, 1987. 2

[4] Cordelia Schmid and Roger Mohr and Christian Baukhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 2

[5] Daniel Scharstein and Richard Szeliski. A Taxonomy and Evaluation of Dense Two–Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, 2002. 4

[6] David G. Lowe. Distinctive Image Features from Scale–Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. 2

[7] J. H. Elder. Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122, 1999. 2

[8] J. H. Elder and R. M. Goldberg. Inferential reliability of contour grouping cues in natural images. *Perception*, 27(11), 1998. 3

[9] J. H. Elder and R. M. Goldberg. Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2:324–353, 2002. 1

[10] O. Faugeras. *Three–Dimensional Computer Vision: A Geometric ViewPoint*. MIT Press, 1993. 6

[11] M. Felsberg and G. Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 41(12), 2001. 2

[12] D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: evidence for a local "association field". *Vision Research*, 33(2):173–193, 1993. 3

[13] Frederik Schaffalitzky and Andrew Zisserman. Multi–view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". *Lecture Notes in Computer Science*, 2350:414–431, 2002. in Proceedings of the BMVC02. 2

[14] W. Geisler, J. Perry, B. Super, and D. Gallogly. Edge co–occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001. 3

[15] J. J. Koenderink and A. J. van Doorn. Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55:367–375, 1987. 2

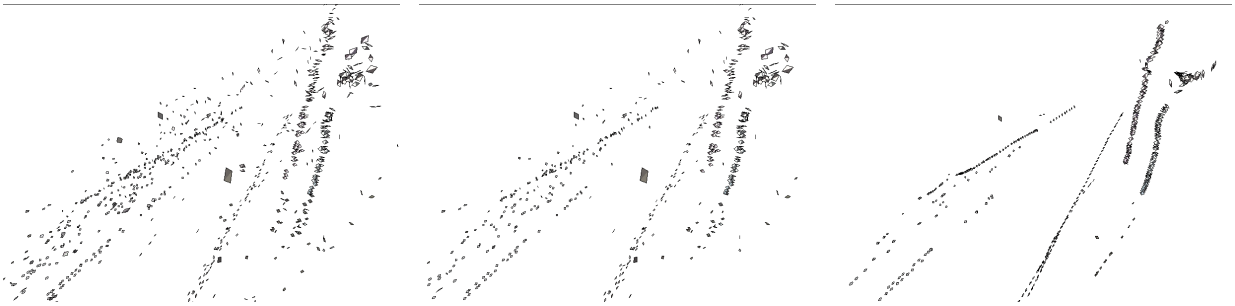
[16] K. Koffka. *Principles of Gestalt Psychology*. Lund Humphries, London, 1935. 2

[17] K. Köhler. *Gestalt Psychology: An introduction to new concepts in psychology*. New York: Liveright, 1947. 2

[18] N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998. 3

[19] N. Krüger and M. Felsberg. A continuous formulation of intrinsic dimension. In *Proceedings of the British Machine Vision Conference*, 2003. 2

[20] N. Krüger and M. Felsberg. An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8), 2004. 1, 2, 4



(a) Stereo reconstruction (b) After external confidence threshold (c) After linear smoothing

Figure 7. Reconstruction of 3D–primitives from stereo–matches obtained from sequence (d) (*c.f.* figure 1). (a) shows the reconstruction resulting from a stereo–matching done using only the multi–modal stereo approach (with a threshold of 0.4 on the multi–modal confidence). (b) shows reconstruction obtained when an additional threshold of 0 is applied to the external confidence. (c) shows the corrected entities, after 3 iterations of the linear smoothing process.

[21] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behavious, AISB Journal*, 1(5):417–427, 2004. 1, 2

[22] Krystian Mikolajczyk and Cordelia Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct. 2005. 2

[23] Luc Van Gool and Theo Moons and Dorin Ungureanu. Affine / Photometric Invariants for Planar Intensity Patterns. *Lecture Notes In Computer Science*, 1064:642–651, 1996. in Proceedings of the 4th European Conference on Computer Vision — Volume 1. 2

[24] Myron Z. Brown and Darius Burschka and Gregory D. Hager. Advances in Computational Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, Aug. 2003. 4

[25] H.-H. Nagel. On the estimation of optic flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987. 2

[26] P. Parent and S. W. Zucker. Trace interface, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):823–839, 1989. 1

[27] P. Perona and W. Freeman. A factorization approach to grouping. In *Proceedings of the ECCV*, volume 1406, 1998. 1

[28] Peter Kovesi. Image Features from Phase Congruency. *Videre: Journal of Computer Vision Research*, 1(3), 1999. 2

[29] N. Pugeault and N. Krüger. Multi–modal matching applied to stereo. In *Proceedings of the British Machine Vision Conference 2003*, 2003. 4

[30] Ronald Chung and Ramakant Nevatia. Use of Monocular Groupings and Occlusion Analysis in a Hierarchical Stereo System. *Computer Vision and Image Understanding*, 62(3):245–268, Nov. 1995. 1

[31] S. Sarkar and K. L. Boyer. *Computing Perceptual Organization in Computer Vision*. World Scientific Publishing Co. Pte. Ltd., 1994. 1

[32] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):504–525, 2000. 1

[33] A. Sha'ashua and S. Ullman. Grouping contours by iterated pairing network. In *Neural Information Processing Systems (NIPS)*, volume 3, 1990. 1

[34] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 1

[35] M. Wertheimer, editor. *Laws of Organsation in Perceptual Forms*. Harcourt & Brace & Javanowitch, London, 1935. 2

[36] William T. Freeman and Edward H. Adelson. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 13(9):891–906, Sept. 1991. 2

[37] F. Wörgötter, N. Krüger, N. Pugeault, D. Calow, M. Lappe, K. Pauwels, M. V. Hulle, S. Tan, and A. Johnston. Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing. *Natural Computing*, 3(3):293–321, 2004. 2

PACO_PLUS TECHNICAL REPORT AND APPENDIX TO D8.1.1

OAC-trees for guiding Discovery

F. Wörgötter, BCCN, Universität Göttingen, Germany

This appendix lays down a more complete description of the OAC-tree concept. It is therefore partly redundant to the writing of the main text. This Appendix reflects work in progress and does not represent publishable material in its current status.

1. Introduction

The goal of every cognitive agent must be to discover (with or without help) the structure and the rules of its world in conjunction with its own embodiment. To this end several complex processes are required and we suggest a certain type of diagram ("OAC-tree", Fig. 1) as a helpful tool for structuring such a cognitive process. We are aware that this is at the moment a tool-in-the-making to get a first handle onto the required algorithmic procedures for attaching attributes to Objects, for manipulating OACs, and for discovering new OACs, etc.

1.1 Perceptual Preconditions:

As discussed above we will not start with a tabula rasa setup, but instead we have built in several perceptual properties. The robot can operate with: (Low Level) Color, Depth, Edges, Flow and haptically, Touch. It can to a more limited degree also operate with: (Higher level) Surfaces, Surface-Relations, Shapes, Rigid Body Motion, Weight, and Softness.

1.2 Action Preconditions:

Furthermore there are built in reflexes (like grasping reflexes, lifting reflexes, turn-hand reflexes, etc). These reflexes are pre-programmed and stereotypical.

1.3 Chain of events:

Simply we assume that a certain percept will in an un-reflected way trigger a certain reflex leading to an automatism and a chain of events.

Perception and action preconditions together with the chain of events constitute the momentarily existing the *knowledge base* of the agent, where knowledge includes the notion of behavioural repertoire.

1.4 Most basic assumption:

Following others we believe that the (ancient) law of cause and effect (e.g. Thorndike, 1911) can be used as one of the most reliable driving forces of any cognitive process: If a certain percept triggers a certain reflex (chain of events) AND IF this leads to a reproducible and perceivable change 'in the world' then this can be stored as a candidate for an OAC, where OACs are – in our hand – the most fundamental (atomic) cognitive entities. The finally stored OAC needs to be the archetype (the "abstracted average") of all single instantiations of the individual "perception→reflex→changed-perception" entities. How this can possibly be achieved will also be specified below via the OAC-tree structural diagram.

1.5 Induction Law and Compositionality of OACs:

An important notion at this point is that OACs can be used to build new OACs in an inductive (bottom up) way. Hence the general goal of discovering OACs is open ended. For example at a low level an unspecific visual percept, where the naïve agent assesses at a pixel level a change from one frame to the next may trigger a grasping reflex with very low success rate. But, if successful, it will lead to a haptic sensation and most often also to a changed visual percept due to having moved the grasped thing (here we assume that the robot has already "subtracted" the seeing of its own body/arm). This has been called "birth of an object" and creates an OAC at a very low level. Through such very basic conjectures we could let the agent self-discover many low-level OACs like what happens to the born object when dropping it, or discovering the difference between rollable (along the cylindrical side) and non-rollable (across the cylindrical side) for a cylinder (an experiment that had been performed by Giorgio Metha from the RobotCub consortium). However, it is obvious that previously stored OACs can be combined in an exploratory way to create a new (slightly higher level) OAC as soon as such an exploratory combination leads (again) to a reproducible outcome. OACs are compositional! Hence we can make use of compositionality in a much different way and just predefine much more complex perceptual entities as well as reflexes assuming that an agent could have reached this level of sophistication though composing higher and again higher level OACs from lower ones. For the "birth of an object" in demo 1 we do this by immediately assuming that the agent can assess co-planarity and perform a much more guided grasping reflex, leading to a much higher success rate. Essentially this agent can now discover the law of rigid body motion and alongside the existence of true physical entities (things, as compared to shadows, etc.; birth of an object). Through compositionality we are in the same allowed to assume that, for example, an agent can perform even more difficult pre-programmed reflexes like "filling" and that the machine can also assess other complex perceptual features like the change of disparity or the closed-ness of a thing (a surface on top could be the indicator here). Such an example will now be used to explain how a "cognitive process" could be started and formalized in an algorithmic way to be useful for a machine.

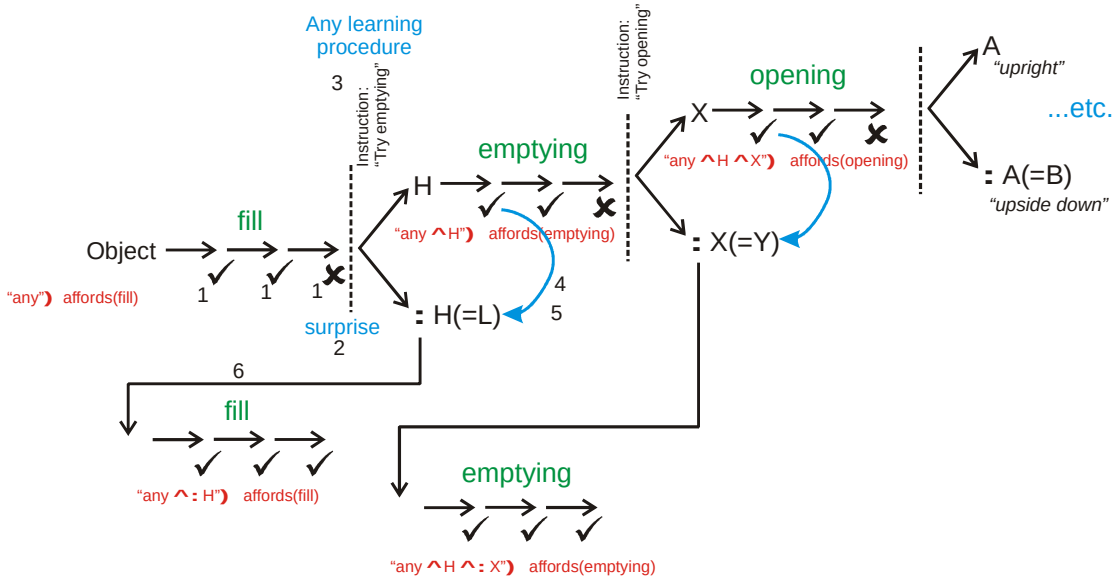


Fig. 1) OAC – Tree Diagram

2. OAC-trees

In the following description numbers in braces (e.g. 1…6, given in Fig.1) will denote that at this point a sub-process is required, which will be specified later. First we will give a general description. We start with a certain knowledge base namely: *any-object affords filling*[1]. Hence the agent can perform a filling reflex by grasping a (different) container and turning it over, above the *any-object* entity. Clearly we assume that the agent can also perform a turning reflex of its hand/arm (leading to the action of emptying) and in a similar way it can do a swiping motion slightly below a seen surface, which will be good enough to remove a loose lid from a container ("opening") and more along these lines. In the beginning the only existing behavioural repertoire is described by *any-object affords filling.* The OAC-tree (left side) shows that the agent will perform this three times, twice with success (1) and a third time failing (1) leading to surprise (2) and the necessity to resolve the surprise (3). This can be achieved by any learning procedure. Here we use supervision and the agent is being told to "try the emptying reflex". After doing this the agent needs to ascertain change (4): which of all possibly changing percepts is causally related to the performed emptying action? This requires also a process of "abstraction" (5), for example to figure out that sometimes changes occur in a correlated way (e.g. grasping will usually lead to a touch sensation *and* a changed visual percept). Clearly an agent is here faced with a credit assignment problem and possibly also with a frame problem. Hence ascertaining change is a non-trivial process (to be specified below) and constitutes possibly the most sophisticated cognitive accomplishment of such an agent. The agent could now try to back-up (6) and perform the filling action next. If backup is successful the agent could conclude that there is an additional attribute (or correlated set of attributes) to the *any-object*, which makes it fillable and this attribute is the one (or set) that has (have) been changed through emptying. The same branching process continues if the agent encounters another surprise (e.g. a closed, full or empty, object). The process of backup (6) will continue to attach attributes to the object class "fillable objects" (commonly called "containers"), which need to be empty, open, upright, obey certain shape characteristics (hollowness) and must not have a hole at the bottom, where the tree in this Figure does not show all these branches.

We note, OAC-trees are not decision trees or planners. They are meant to be a first diagrammatic step towards visualizing and implementing a fairly complete reasoning and learning process. Furthermore we note that the depicted tree treats the case where through the process of backup attributes can be attached to objects. Different trees are possible, which allow discovering new object classes on their own (see section 4).

[1] Note, the "object-ness" required to start this tree could have been derived from the "birth of an object" process of demo 1.

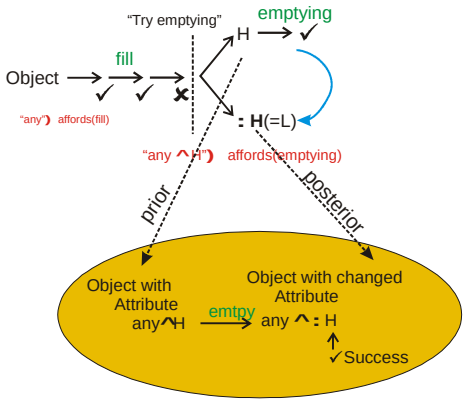


Fig. 2) Ascertaining Change (Process 4) creates a new OAC-candidate.

3. Towards an algorithmic specification of processes 1-6

Note, all processes described below are to some degree stochastic, hence require repetition to ascertain validity.

Process 1, "Success&Failure": This process will also lead to the building of an inner model. To this end we define the following recursive double-process here:
Aa) Success is measured as the deviation of the actual observed outcome of a chain of events from the expected outcome, where the expected outcome is given by an inner model of the given OAC in the agent. A small enough deviation (assessed by thresholding) from the inner model triggers is counted as Success a too large deviation as Failure.
Ab) The inner model can initially be empty. The inner model will be updated with every success by using the observed outcome and averaging it with the currently existing inner model[2].

Process 2 "Surprise": Surprise is elicited on Failure (passing the threshold, see above).

Process 3 "Resolving Surprise": Surprise can be resolved by any learning method. We have either Trial&Error Learning or Reinforcement Learning or Supervised Learning (Instructions). The influence from outside increases along these lines, but so does the speed of learning. One or more of these methods needs to be implemented. In humans trial and error is usually the immediate choice after surprise has happened and will be replaced only after frustration with asking for help.

Process 4 "Ascertaining Change": The agent needs a difference-sensitive mechanism that compares a stored prior percept before the new action with the observed outcome (posterior) after the new action. This process is very similar to the difference-mechanism used to trigger surprise in Processes 1 and 2 only here it is operating on percepts and not on the inner model. The observed change(s) should be used to start filling the next stage using the posterior (percept *after* action) as the next expected

[2] We note: Since the first inner model can be empty, in this case first success will be TRUE whatsoever and the second inner model will be identical to the first observed outcome. If this (for an external observer) is wrong, it will not matter as long as the used OAC will change the world in a roughly reproducible way because repeating the OAC will drive averaging away from the at first encountered contingency. Note, in this case the agent will experience surprise when doing this OAC the second time.

outcome of the next stage of the inner model. The stored prior should be used to build the prior of this OAC. This way a new OAC becomes visible (see Fig. 2), but at this stage the agent cannot be sure about the new OAC (Process 6, below, can ascertain this though). In section 4 we elaborate on the pitfalls of Process 4. This is a difficult process and PACO-PLUS cannot claim that we have fully understood all underlying implications here. This will be part of the work of the next period.

Process 5 "Perceptual Abstraction": The agent requires a process that allows subsuming multiple perceptual changes into one relevant (more abstract) OAC attribute. This can be achieved by assessing correlations between perceptually changing entities: If A always changes together with H then they can be subsumed into one larger attribute-type for the new OAC.

Process 6 "Backup": If the agent can perform the prior planned action, then we can ascertain that the suspected new OAC, found with process 4, is indeed likely to exist. We can attach a new attribute to the object.

At the moment PACO-PLUS is working on the algorithmic specification of these and some other processes hoping to arrive this way at a procedurally specifiable cognitive architecture.

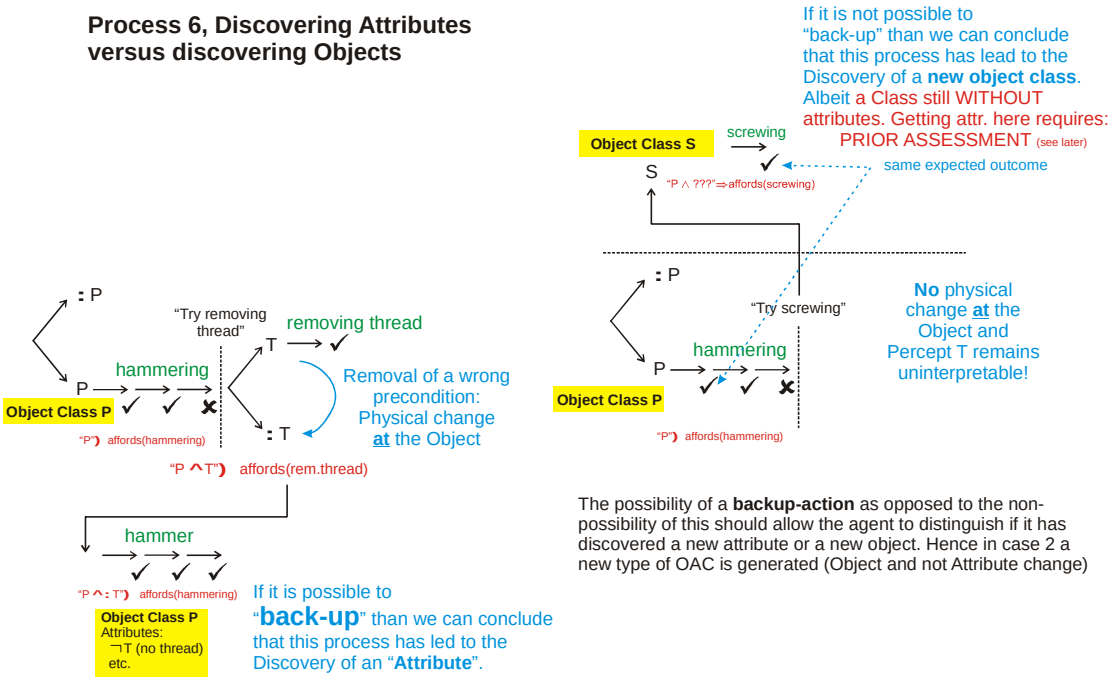
4. More specific cases

The following discussion will briefly introduce several more specific cases to show that the OAC-tree concept is fairly general and can be applied across a wider domain. Clearly we may encounter restrictions at some point, but so far this tree could be augmented to also be able to discuss more complex cases and additional processes required to resolve these. This section is mainly meant for the interested reader to be able to better assess how OAC-trees can be used to visualize different processes and their difficulties. We keep descriptions very brief here and refer to the trees themselves for explanations.

1) Fig 3 shows the distinction between attaching an attribute to an object (pointy objects without thread [or after removing the thread with a file] can be hammered) as compared to discovering a new object class (objects with thread [which we choose not to remove] need to be screwed).
2) Fig. 4 shows a work-in-progress diagram that tries to elaborate on Process 4 (Ascertaining change), discussing some of the difficulties on assessing changes at perceptual priors or posteriors, respectively. Here we also begin to discuss that actions can change (damage) the tool, which is a special case but can be resolved using the OAC-tree to depict yet another process (called "Transferring Inference").

References:

Thorndike EL (1911) Animal intelligence. New York: Macmillan.

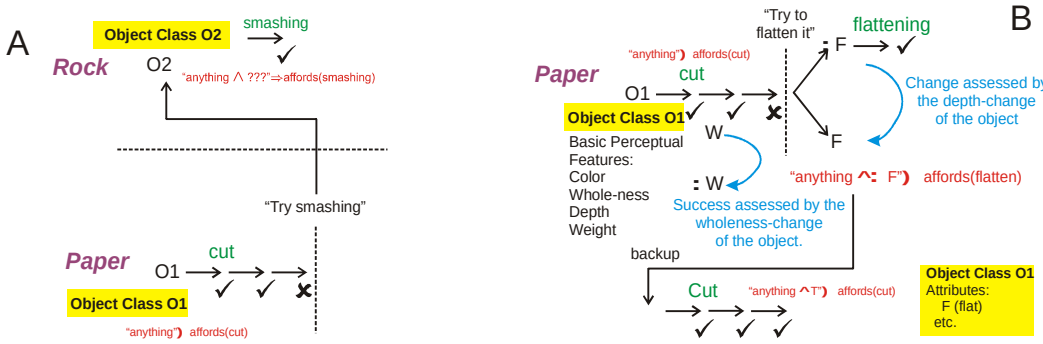


IMPORTANT NOTE: This process uses the posterior of the second action as the necessary prior of the first action defining the object's attribute this way. **Called the: Assessment of Posteriors**

Fig. 3) OAC-trees allow for the distinction between discovering attributes as compared to discovering new objects or object classes. If "backup" is possible then this is a strong indication for an attribute.

Elaborating on Priors and Posteriors (again a note here: We can also 'tell' the robot which attributes are relevant, which we do often with kids!! The described processes allow to self-discover this, though, but might take time or go wrong!!)

The key question in case "smashing" (A, top) is: Is there a process which makes it possible to attach an attribute to "affords(smashing)", hence replacing the "???" with an attribute. This ought to be possible making use of the bottom part of this case (A, bottom) but requires that the agent needs to have performed successful cutting beforehand to get to this.
The idea would be to **Assess Perceptual Priors,** hence to compare Objects O1 and O2 !!!BEFORE!!! performing the respective actions on them. (or to remember how they had been before having changed them - a thing much harder for a human/agent).
This should at least allow to attach some hypotheses to O2 (like "Rocks are 'usually' non-smooth, heavy and non-flat as compared to paper which is smooth, light and flat").
Clearly the example B shows that flat versus non-flat is problematic, if the paper is crumpled. Also clearly this problem shines up for ALL such hypotheses: **Assessment of Priors** is a weak stochastic method. There is a "credit assignment" problem together with a pronounced frame problem (which of all zillions of possible Priors could have been the relevant one???????). Still if one uses two action-cases only (pairwise comparison) and looks only at very low levels of the Assumption Hierarchy (Page1) some mild conclusions ought to be possible even here (Heavy and non-smooth are the candidates, here).
For case B the key claim is that: **Backup allows for Assessment of Posteriors and that this is a strong method** with less credit assignment problem because of the law of cause and effect. The frame problem might shine up and can only be reduced by repetition and assessment of repeatedly changing perceptual posteriors (as opposed to spuriously changing ones).



On Transferring Inference:
If we have through repetition arrived at a "well-consolidated inner model" of an OAC, then we must conclude that "something else" has changed if the expected outcome (predicted by the inner mode) of the action does not happen. In a limited scenario this requires concluding that "the other object" (the scissors) must have changed. Either we can now at the scissors try to assess priors (which is hard) or someone could tell us to "sharpen the scissors and then perform a backup. This way the attribute "sharp" could be attached to object class O3 as a precondition for cutting.

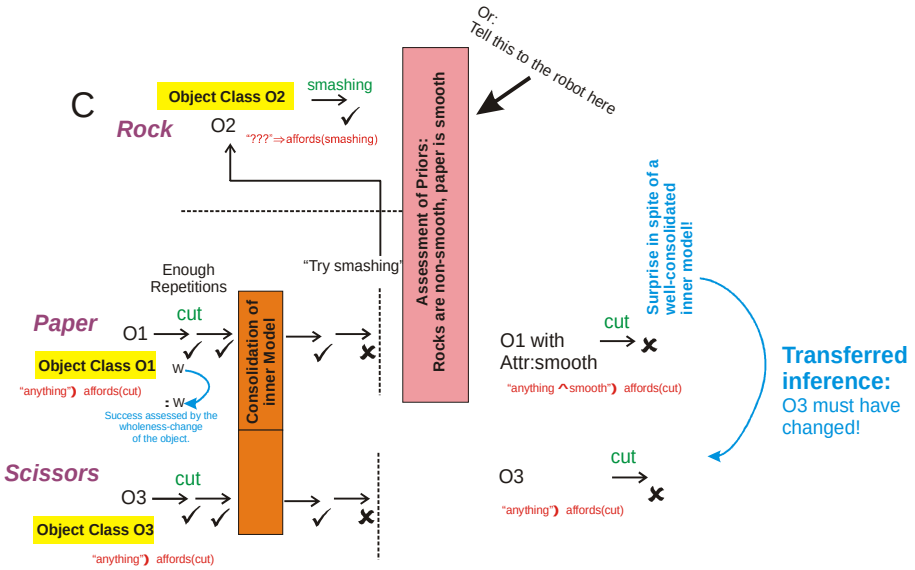


Fig. 4) Different, more complex, reasoning processes depicted by OAC-trees.