Computer Vision and Image Understanding 113 (2009) 682–692

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Integrated vision system for the semantic interpretation of activities where a person handles objects ☆

Markus Vincze [a,*], Michael Zillich [a], Wolfgang Ponweiser [a], Vaclav Hlavac [b], Jiri Matas [b], Stepan Obdrzalek [b], Hilary Buxton [c], Jonathan Howell [c], Kingsley Sage [c], Antonis Argyros [d], Christoph Eberst [e], Gerald Umgeher [e]

[a] Automation and Control Institute, Vienna University of Technology, Vienna, Austria
[b] Center for Machine Perception, Czech Technical University, Prague, Czech Republic
[c] School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK
[d] Computer Vision and Robotics Laboratory, Foundation for Research and Technology Hellas (FORTH), Heraklion, Greece
[e] PROFACTOR Produktionsforschungs GmbH, Steyr, Austria

ARTICLE INFO

Article history:
Received 6 April 2006
Accepted 30 October 2008
Available online 17 November 2008

Keywords:
Activity interpretation
Cognitive vision
System integration
Semantic interpretation
Task orientation

ABSTRACT

Interpretation of human activity is primarily known from surveillance and video analysis tasks and concerned with the persons alone. In this paper we present an integrated system that gives a natural language interpretation of activities where a person handles objects. The system integrates low-level image components such as hand and object tracking, detection and recognition, with high-level processes such as spatio-temporal object relationship generation, posture and gesture recognition, and activity reasoning. A task-oriented approach focuses processing to achieve near real-time and to react depending on the situation context.

1. Introduction

Future operator manuals could be based on virtual or augmented displays to facilitate experience based training on how to operate a machine or carry out an industrial process. Examples include guiding a user to operate a video or CD-player or to clear a paper jam in a copy machine. Using augmented displays, such a training system could overlay the next actions onto the equipment itself, assist the user with contextual hints and check if the overarching task is being performed in a correct manner and order. Such a system would need to interpret the activities of the user while verifying these activities against a stored reference activity plan. Furthermore, the recording of the reference plan itself could be generated from observing an expert user executing the correct sequence of activities. While the vision is still some years ahead, this paper presents initial work towards these end goals. The objective was to develop an integrated vision system to interpret the activity of a person who handles objects. The result is a task-driven architecture and system for Activity Interpretation, which we call the ActIPret system.

One way to obtain the necessary input is to use several sensing modes, e.g., gloves, magnetic and optical trackers or 3D cameras [26,13]. However, the broader adoption of these technologies, potentially all the way to home use level, will depend on the proliferation of cheaper hardware, e.g., cameras.

Our task-driven architecture approach to activity interpretation contrasts to approaches drawn from areas such as surveillance, smart rooms, or human interaction. In these approaches a generally fixed notion of human motion (e.g., gestures and postures) is the main target of fixed view visual processing. In our case, operating a machine also requires visual capabilities for object detection, tracking, recognition and possibly even classification. But activity synthesis also requires methods to reason about the contextual relationships of humans who handle objects and means to derive the interpretation of the observed activity. But a key barrier to developing practical vision systems is that they contain a large number of vision and reasoning functions. Recent approaches to create practical vision systems that combine vision with reasoning, task-control and learning have been referred to as cognitive vision [19]. For example, an approach to traffic analysis combines low-level vision processes such as tracking with semantic interpretation [20] or exploiting input from low-level vision components to learn rules of a card game [8].

☆ This work was mainly supported by the EU-Project ActIPret under Grant IST-2001-32184 and partially supported by projects S9101-N04 and S9106-N04 of the Austrian Science Foundation.

* Corresponding author.
E-mail address: vincze@acin.tuwien.ac.at (M. Vincze).

1077-3142/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2008.10.008

The ActIPret architecture and system presents an attempt to integrate components from all levels of a cognitive vision system. It is special in the sense of combining both object as well as gesture detection, tracking and recognition approaches, and it continues to include components for intermediate 3D representations of spatio-temporal relationships, to finally arrive at the level of activity interpretation that leads to a symbolic description.

The ActIPret system is unique in the respect that it sets out to demonstrate that top-down contextual knowledge can be exploited to focus processing of all these bottom-up vision processes to achieve processing in near real-time. In this manner the intention was to arrive at purposive *and* reactive processing and interpretation. This task-driven approach made near real-time operation in a known but not prepared environment possible. It was demonstrated in a scenario of operating a CD player with one or two hands. Different persons executed the task in a pre-learned, natural environment. An evaluation shows good performance on sequences recorded from a set-up with two active stereo heads under dynamic and task-driven control.

The paper proceeds as follows. The next section reviews related work. Section 2 introduces the cognitive vision system approach and Section 3 shortly describes the vision and pre-reasoning components used to fulfil the task. Individually they have been reported to enable the reader to have a comprehensive and self-contained picture of the cognitive vision system. The use of contextual coordination to focus processing depending on the situation and to extract the activity plan is presented in Section 4. Section 5 presents experiments and Section 6 evaluation using one-hand and two-hand activities.

1.1. Related work

Applications such as surveillance, smart rooms and human-machine interfaces require work to recognize activities of humans. For a recent review see [6]. An example of work is [5], which investigates probabilistic approaches to gesture and behaviour recognition. This is one of the modules in the task-driven architecture proposed here. Another source of references is the series of PETS workshops (performance evaluation of tracking and surveillance), which investigates activities such as walking and meeting of persons, pointing gestures or speaker identification. The solutions are systems built for the specific tasks processing the input off-line in a bottom-up fashion. Interpretation can be achieved from observing the motion of humans or group of humans, which makes it possible to focus processing on detecting and tracking humans and to finally analyse the motion patterns. Objects are not treated as separate items and they are only indirectly involved, e.g., detecting changes in walking motion with and without suitcase or luggage [3]. For analyzing traffic at a street crossing [20] suggests a structure for the processing steps to extract the spatiotemporal geometric descriptions in the scene domain. Compared to the work presented here processing steps are fixed and procedures not called depending on the context or task relationship.

The relation of humans to objects in their environment is studied in approaches to programming by demonstration (PbD), where the task is to interpret user commands to teach a robot. This interpretation task involves several more capabilities: gestures must be related to objects in the scene. Hence objects need to be detected, tracked and recognized. When including speech commands, a semantic interpretation is also needed. As a consequence, a PbD system is composed of several more functional blocks as the surveillance systems presented above. In present work this is solved by using substantial equipment to reliably detect hand activities (data glove, magnetic trackers, force sensors). Vision is only used to recognize marked or colored objects on uniform background [26,13]. To relax the need for extensive equipment [29] use depth images generated from multiple cameras to track hands and tools to analyse the task sequence.

From the point of view of a task-oriented vision system the most related approach are the process federations presented in [10]. Process federations are constituted of executive components and a hierarchy of controlling supervisors on top of them. The selection of vision modules is driven by cost type functions, with the goal being to achieve a visual task at lowest computational cost. In our system, the focus is on real time selection from a range of components, some visual, some not, with a view to selecting task relevant components as well as having a view on the cost. The task relevance extends from a real time Bayesian statistical analysis of the probability of a specific action emerging based on all the prior evidence available at any time step. So the task relevance can emerge, rather than being pre-packaged into a predefined goal set.

A key feature of the ActIPret system is the real time purposive control of the resources to extract a symbolic description of activities. The focus is on active observation and interpretation of activities, on parsing the sequences into constituent behaviour elements, and on extracting the essential activities and their functional dependencies. The activities are interpreted and stored using natural language expressions in an activity plan.

2. System description

The system developed for activity interpretation is structured in three main levels: attentive level, pre-reasoning level, and synthesis level (see Fig. 1). The attentive or lowest level comprises the main vision components such as hand and object detection, tracking and recognition. It operates on stereo image data and the pose information of each camera is available. We separate between pre-attentive and attentive processes to denote processes that are fast and operate continuously versus slower processes which might operate only on demand, e.g., hand tracking operating at frame rate is a pre-attentive process. An attentive process is, e.g., object recognition, which is computationally expensive. Hence it is only invoked when a higher level process requires it given a specific context such as at key events discovered in gesture recognition. For more details please refer to Section 3.

The pre-reasoning level has the task to extract meaningful information from the low level vision processes. Hand posture recognition (HPR) uses direct image information and is, hence, view dependent. The two other components in this level work on the 3D data delivered by the tracking and recognition components: the gesture recogniser (GR) uses the 3D hand pose (position and orientation), the object relationship generator (ORG) the pose data of hand and objects. This pose data is independent of the specific camera views and given in a common world coordinate system.

All vision components and the HPR operate on image data. The view contract manager in Fig. 1 has the task to coordinate the cameras and their viewing directions to supply the best views as demanded by these components. The implementation of the full system used two stereo vision systems (=four cameras) on active heads as explained in more detail in Section 3.8. Components are connected to either one of the stereo systems for simplification.

The top level component is the activity reasoning engine (ARE), which combines the pre-reasoning information and extracts the symbolic description of the activity.

To realise context-driven control of all these components, a component-based framework has been developed [25]. It uses the mechanism of *services*. Each component makes public the services it can perform in the service list. Depending on the actual task and context components can check in the service list, which component could deliver the requested service and the communication

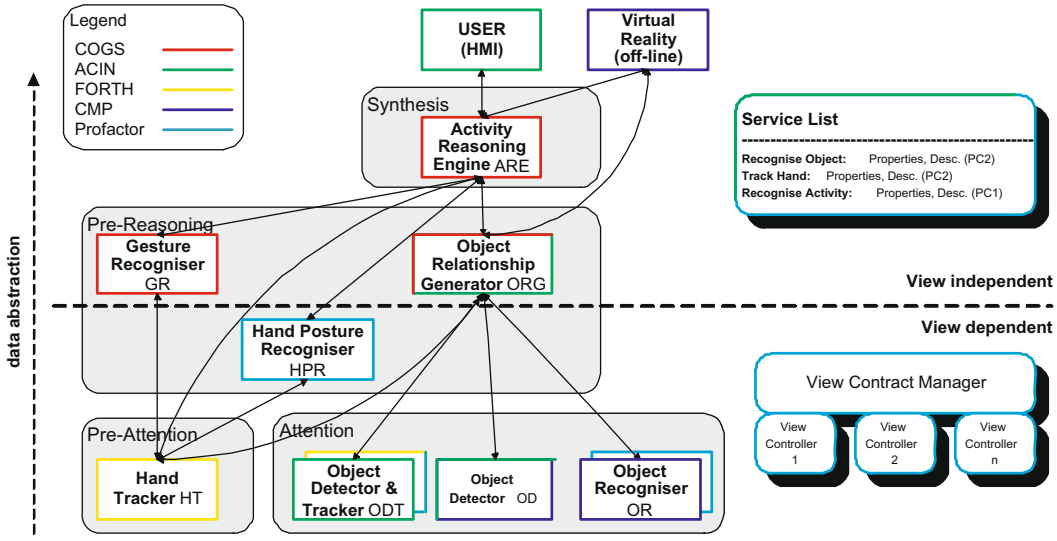

Fig. 1. Diagram of the components of the activity interpretation system.

is established. Section 4 will describe in detail how this contextual control is realised starting from the ARE.

3. Vision and pre-reasoning components

This section gives a short overview of the vision and pre-reasoning methods developed for the activity interpretation tasks such that the reader will obtain a good picture of the complete system and is able to better understand the contextual control mechanisms. Note that the components are not the original contributions of this paper. References to more detailed descriptions are given.

3.1. Hand tracker

The goal of the hand tracker (HT) Component is to deliver 3D position and pose information regarding all skin-colored hand hypotheses in the field of view of a calibrated stereoscopic observer. In the heart of the developed component lies a method for tracking multiple skin-colored objects in a monocular image sequence [1]. This method encompasses a collection of techniques that allow the modeling, detection and temporal association of skin-colored objects across image sequences. A non-parametric model of skin color is employed. Skin-colored objects are detected with a Bayesian classifier that is bootstrapped with a small set of training data and refined through an off-line iterative training procedure. By using on-line adaptation of skin-color probabilities the classifier is able to cope with considerable illumination changes. Tracking over time is achieved by a novel technique that can handle multiple objects simultaneously. Tracked objects may move in complex trajectories, occlude each other in the field of view of a possibly moving camera and vary in number over time.

Besides the basic detection and tracking mechanisms, a contour extraction module computes robustly the external contour of skin-colored blobs. Additionally, a multi-scale contour processing module detects hand fingertips as the local maxima of the curvature function computed over the extracted contours.

Two instances of this 2D skin-color detection, tracking and finger detection components run independently on the video streams provided by the two cameras of the stereoscopic observer. Hand hypotheses are then matched across the synchronized stereo pairs through dynamic programming. The cost function employed in this dynamic programming-based stereo matching of hand hypotheses is based on the distance of hand centroids from their epipolar lines. Epipolar constraints are derived analytically based on the knowledge of the camera and stereo configuration parameters. Once the hand hypotheses have been matched across the images of the stereo pair, the iterative closest point (ICP) algorithm [4] is employed to match contour points at the pixel level. Since fingertips are simply special contour points, the method delivers the 3D locations of fingers, as well.

A prototype implementation of the proposed tracker operates on live video at a rate of 22 Hz on a Pentium IV processor running under MS Windows, without resorting to assembly optimizations or special hardware instructions such as MMX or SSE. Being non-parametric, the proposed approach is independent of the shape of skin color distribution. Also, it adapts the employed skin-color model based on the recent history of tracked skin-colored objects [1].

3.2. Hand posture recogniser

Hand Postures are a very powerful mean of identifying the intention of a human operator, building gesture-based interfaces and recognizing sign language and provides important information for the interpretation of activities of expert operators. Therefore a significant amount of research has addressed building systems to recognize the posture of hands from visual input [11,21,31].

In the interpretation task at hand this kind of functionality is used as a pre-reasoning step for the interpretation of activities and provides relevant input for the activity reasoning component. The posture recognition functionality is using two kinds of input. The first is the contour of the hand where the posture recogniser benefits from the capabilities of the hand tracker and the second is the intensity image of the scene itself. The contour is used to mask the hand region from the input frame and the resulting hand region is mapped to an image of fixed orientation and size which is called attention image. This component was implemented following [11].

In the training phase the component learns the mapping of the attention image from many views of the hand to the correct posture with a PCA and RBF based technique. In the execution phase this mapping enables the system to classify the attention image into the learned hand postures and the component is able to calculate the probability of correct classification. Up to now postures like a pointing, a grasping and a normal hand posture are recognised.

3.3. Gesture recogniser

The gesture recogniser (GR) component is designed to classify hand trajectory data into a small number of functional gestures. These gestures were chosen for their generic nature (suitable for a wide range of hand manipulation tasks) and the extent to which they could provide useful information for other parts of the ActIPret system (in particular the Activity Reasoning Engine). The GR component provides two services. The first, HAND_INFO is key in the early pre-attentive selection of task relevant hand candidate objects (discussed in the section on contextual control). The second, RECOGNISE_GESTURE, specifically builds a history of hand trajectory data for a reference hand object (using data supplied by the HT) and returns probability estimates of each of the functional gestures to the calling component (exclusively the activity reasoning engine at this time).

Gestures are defined relative to the torso. The torso is defined as the point from which the gesture starts. The first point of significant motion above a threshold defines the origin of the torso centric co-ordinate systems.

At present, there are three functional gestures; hand moving away from a nominal torso position (“reaching out”), hand moving towards a nominal torso position (“reaching in”) and a hand moving a constant radial distance about a nominal torso position (“lateral motion”). These functional gestures are appropriate for characterising the hand motion element of a range of hand manipulation tasks such as “picking up” and “pressing a button”.

The GR component contains an embedded real-time gesture recogniser built using a time delay radial basis function (TDRBF) neural network classifier [14]. The TDRBF network is a two-layer, hybrid learning network [17,18] which combines a supervised layer from the hidden to the output units with an unsupervised layer from the input to the hidden units. The network model is characterised by individual radial Gaussian function for each hidden unit, which simulate the effect of overlapping and locally tuned receptive fields. Supported by well-developed mathematical theory, the model provides rapid computation and robust generalisation, powerful enough for real-time, real-life tasks. The non-linear decision boundaries of RBF networks make better general function approximations than the hyperplanes generated by the multi-layer perceptron (MLP) with sigmoid units and they provide a guaranteed, globally optimal solution via simple linear optimisation. The radial basis function (RBF) neural networks is trained off line with data from recorded sequences.

3.4. Object recogniser

The object recognition component (OR) reports presence and location of object of interest in the scene. The MSER-LAF (locally affine frames on maximally stable extremal regions) recognition method [22,23,16] proceeds as follows: 1. Detect distinguished regions (DRs). MSERs are used here, but any process producing image regions stable under affine transformations can be exploited, 2. Build local coordinate systems (LAFs), applying various affine covariant constructions, 3. Define a measurement region (MR), an image region of shape and size relative to the local coordinate systems, 4. Geometric normalisation: transform MRs of individual LAFs into a canonical form. 5. Photometrically normalise RGB values in MRs, 6. Represent the normalised MRs by low frequency DCT coefficients, 7. Local correspondences are established by correlation of the DCT coefficients, and 8. Verify the object detections hypothesised by local correspondence. The MSER-LAF method was successfully applied to various object recognition tasks and achieves state-of-the-art recognition results [22,23].

Two improvements towards robustification of the recognition approach were proposed in relation to the ActIPret project. First, the distinguished region detection was generalised. Besides intensity MSER, extremal regions with other ordering of RGB values were used, yielding lower false negative rate. The second improvement was in decision making: a background model reflecting spatial dependencies lowered the false positive rate. All details about the robustification can be found in [24].

Besides standard performance measures of a recognition system—the recognition rate and the false positive rates—the process of activity interpretation requires the object recogniser to operate in “near-real” time. Precision of localisation is also important, since gestures and activity interpretation depends on co-locations of objects. The requirements are contradictory. To achieve maximum recognition rate and highly precise localisation, complex recognition strategies have to be employed. The flexibility of the Actipret framework prevents us to rely on fast, object-specific, hand tailored approaches. On the other hand, the near-real time performance requirement limits the generality and complexity of the recognition method.

To optimise the speed of the recognition, we have proposed to use an adaptive recognition strategy. First, in a learning phase that is executed off-line, the recognition system is run in a mode that maximises the recognition rate. After analysing the processes that lead to correct recognition, only the smallest subset of recognition sub-processes guaranteeing an acceptable recognition rate was selected for further operation. Through adaptation, posed as constrained optimization, the fastest set up of the recognition system achieving desired performance was found.

3.5. Object detector

Detection refers to the process of finding and locating an object class. The task is to initiate tracking and recognition with a fast process. This is in particular true for object recognition, which operates too slow for real-time response. The object detector could be tailored to the task of the CD-player scenario, where the main target objects are CDs. Hence, the object detector (OD) finds elliptical objects in the image. The algorithm works on edge segmented images and makes explicit use of edge connectivity information [34]. It is based on a hypothesise and verify method. Ellipse hypotheses are formed of groups of arc segments. Efficient grouping of arc segments based on tangent intersections is used to significantly reduce the exponentially large number of possible groupings of arcs. The search for consistent groupings thus becomes a linear problem. Standard techniques are then used to fit ellipses into groups of arcs.

The main task of the OD in the system is to provide precise pose information to increase the robustness of the initialisation of the object tracker. While object recognition is the slowest process (seconds), object detection operates faster but still slower than frame time (about 200 ms) and helps to refine the pose accuracy. This information is then exploited in the object tracker, which operates at frame rate (25–50 Hz). Additionally, running the object detector can serve as redundant information server to the object recogniser and, hence, it improves overall system robustness.

3.6. Object detector and tracker

The object detector and tracker (ODT) has the task to detect and then track an object. It uses the OD for initialisation and subsequently a model-based tracking procedure. Object tracking exploits several model and image cues in the EPIC (edge-projected integration of cues) scheme to obtain high robustness over an image sequence [32]. It extends edge-based tracking of geometrical features (lines, ellipses, arcs) towards integrating intensity and colour cues depending on object and background information. Depending on the perceived scene complexity each feature can

evaluate the present detection quality and use this to fine tune the cue integration process [2]. While the initialisation using the OD takes about one second, tracking using EPIC operates at frame rate. The tracking method is available for exploitation as the tool vision for robotics (V4R) [33].

Using the object model information and the appearance information of the object from the last frame object tracking is the fastest and most precise way to follow the motion of an object. When used in a larger system context, the crucial step of object tracking is its initialisation. Hence this component is initialised with the pose information generated by the OR and further refined by the OD (see above).

3.7. Object relationship generator

The object relationship generator (ORG) has the task of abstracting the metric information of the vision components into symbolic information as used in the ARE. It has been introduced between ARE and the vision components to relieve final activity interpretation of this additional task. It will be shown in the next Section (4) that it can efficiently and autonomously control vision processing. Specifically, it executes a spatio-temporal evaluation and delivers significant events in the scene and a nearness measure to activity reasoning. Events extracted are, for example, lost or appeared objects. These events are extracted in relation to the main actor. For the scenario presented here the main actor is the hand but could be any other motion generator in the scene.

The relationships that are needed for the activity interpretation task are mutual proximity of two entities (hand and objects in the scene at one instance) and the relation of an object near to the trajectory of the main actor (object to object trajectory relationship). Besides lost and appeared objects, objects moving consistently with the main actor are also reported.

These two relationships can be subsumed by the measure "nearness". Nearness is defined as the center-to-center Euclidian distance between entities in 3D space. The nearness relation is fulfilled if the Euclidean distance between two objects is smaller than a certain value. This value depends on the type of the objects as well as the velocity and orientation of the reference object motion. Especially temporal changes of the set of related objects are of interest, because this defines which objects became or are still of interest or are no longer of relevance. Accordingly, hypotheses depending on object relations can be created, further investigated or cancelled. In a response to a query, entities are ordered according to nearness. All relationships found are reported to ARE (see Section 4). This has proven to be robust because of avoiding cut-off parameters or other thresholds.

The ORG is implemented as an active database. It is active in the sense of coordinating vision processing, that is, starting vision processes depending on contextual information and keeping track of relevant relationships. One part of this coordination task is the scheduling of the components regarding their different temporal and precision behaviour. Another part is the restriction of processing to arranged parts of the work space, so called space of interest (SOI), near the reference object. Section 4 will give examples of how ORG is used for contextual control of the vision components.

3.8. View controller, view contract manager

In a system like ActIPret, which has restricted resources, it is essential to control the processing of visual behaviour/services. The most restricted resource in ActIPret is the view,[1] therefore handling the resource "view" is of main interest. A fitting view is vital for the vision components to allow robust performance [7,12,15]. One option how the resource view can be handled appropriately is to use active vision systems.

Therefore the ActIPret demonstrator consists of two heterogeneous robot systems.[2] Robot system 1 is a four degree of freedom stereo camera head and system 2 a six degree of freedom robot arm. On each robot system a stereo camera.[3] pair is mounted and the robots are controlled via a PC based industrial controller.

The idea behind the control mechanism [30] is based on a decentral approach in which each involved service has limited specific responsibility, that matches the local knowledge inherent to its task. A vision or interpretation component can decide what the space of interest is and which service the component needs to fulfil its task. But there is not enough knowledge to decide which robot can provide the necessary view point. In contrast the view control component knows motion capabilities, dangerous zones, the status of the robot system and the quality to fulfil view requests of individual components and the view contract manager knows the overview of vision services running on individual robots, including the quality of the provided view. The quality is calculated based on the current and requested robot position and potentially contradicting view requests of other vision components. The combination of the local knowledge of these components in the ActIPret system enables vision services to request a space of interest (like: close to the hand or in front of the hand) and an appropriate camera system will provide the best possible view point for the vision component. This concept provides the ActIPret system with a task-driven control of the view behaviour of the vision components.

4. Contextual control

As the ActIPret framework is a task driven system, no visual processing at all takes place until a high-level task is established by the activity reasoning engine (ARE). In the use case illustrated here, the highest level task is to establish an activity plan. An activity plan is a concise account of the scenario specifying the relevant objects and how they are acted upon [27].

The highest level of reasoning within the ARE is called the control policy. This represents the highest and most abstract level of attentional control. The control policy is concerned with identifying relevant initial objects in the scene and with determining what type of behaviours it might be appropriate to look for. The top level control loop for the ARE can be summarised by the following pseudo-code:

```
Start a service HAND_INFO to look for moving hand
objects
   anywhere in the scene
WHILE (HAND_INFO is running)
   HandCandidateSet = hand objects identified by
   HAND_INFO
   Maintain set of Visual Indexes for
   HandCandidateSet
   FOR ALL members y in HandCandidatesSet
      Request a service for GESTURE_RECOGNITION for
      object (y)
      Request a service for HAND_POSTURE_RECOGNITION
      for object (y)
      Request a service to determine objects for
      tuple (y, NEAR)
```

[1] A view is defined as the orientation of a camera and its optical ray through the image centre towards the centre of the 3D space of interest requested by a component.

[2] Amtec robotic system (http://www.amtec-robotics.com).

[3] Sony FireWire camera DFW-VL500.

```
      Determine Visual Index internal variables for
      hand (y)
    END_FOR
  END_WHILE
```

The service HAND_INFO is provided by the gesture recogniser (GR)/hand tracker (HT) pairing. The GR requests a service from the HT to provide data on all hand candidate objects based on a skin colour detection algorithm [1]. The HT provides this data to the GR which then uses a simple frame by frame measure of Euclidean distance between centroids to determine if any of these hand candidates are moving. Object labels for those that are (the ARE has no ability to process absolute positional data) is then returned to the ARE. The ARE then selects those candidates that are "interesting" and makes a request to the ORG to observe whether the selected hand objects are in relationships with other scene objects. The ORG uses information about the trajectories and posture of the hand objects to determine what 3-D space models need to be investigated (task based selection of 3-D 'spaces of interest'). The service to generate hand posture data is generated by a hand posture recogniser (HPR) and the services that generate data on spatial relationships are provided by the ORG.

For each object warranted worthy of investigation, the ARE creates a visual index (VI). A VI is simply a collection of data items and variables that relate to a specific hand object. These variables are used to encode internal beliefs about the state of a hand object that are relevant for task based control. Some of these beliefs are determined directly using information supplied by lower-level components (e.g. whether the hand is currently moving), some are inferred using internal knowledge acting on data returned from lower-level components (e.g. whether the hand is near an object whose categorisation suggests that it may be picked up) and some is inferred according to previous actions (e.g., whether the hand is full as a result of previously picking something up and the identity of the objects picked up).

The ARE uses the VI state data and a decision tree (shown in Fig. 2) to generate action hypotheses (also stored in the VI as they are hand specific) for each selected hand object, which, when confirmed, are collected and recorded in the activity plan. Whilst the hand object remains in scope, these hypotheses are generated and updated regularly. When the hand object goes out of scope (i.e. disappears from view), the VI is deleted. The hypotheses are derived from instances of dynamic bayesian belief networks (BBN) which define what spatial and temporal evidence is relevant and to be combined over time in a way that characterises the actions [28].

A key feature of the ARE/ORG pairing is the ability to select task relevant objects (attentional selection). The computational load on the ARE, and the number of service requests it makes to lower-level vision components, is a function of how well the ARE and ORG can select task relevant objects. If the selection criteria are too wide, the benefit of attentional selection on the computational efficiency of the whole system is lost. Conversely, if the selection criteria are too narrow, there is a risk that the real significant objects are not identified and an incorrect activity plan results, or no activity plan at all. The ORG generates task based 3D spaces of interest for reference objects nominated by the ARE. Currently, the reference object is a hand candidate selected by the ARE/HT pairing.

Information about the outputs of specific nodes in BBNs for a specific reference object can also be used to generate ORG service requests. The key concept here is that certain data is only required to specifically confirm that certain actions have taken place. For example, we visually infer that an object has been picked up by combining data about the gesture of the reference hand object and the disappearance of essentially non-hand objects at appropriate times in the overall gesture sequence. We only need to look for such lost objects when the reference hand has started to pull away form the 3D location where the objects were previously believed to be. So the activation levels of certain BBN nodes are used to determine whether it is appropriate for the ARE to generate additional service requests to the ORG. For example, in the example BBN for

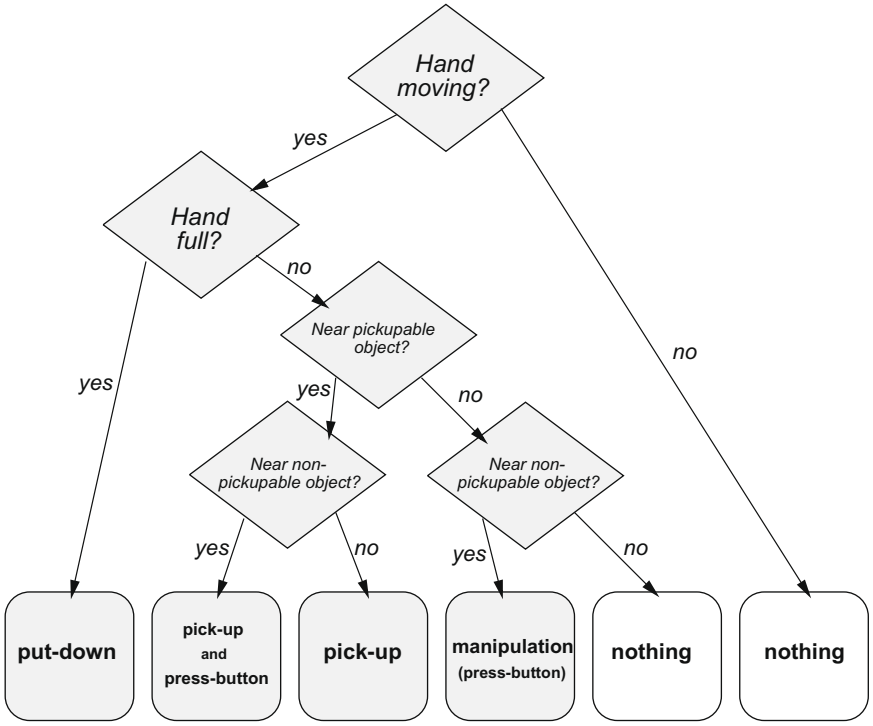

Fig. 2. Predictive decision tree, based on control policy rules, for the selection of BBNs (Bayesian belief networks).

the action "pickup" shown in Fig. 3 the ORG is only requested to identify "lost" candidate objects that the reference hand may have picked up when the activation level of the node $p(X, ReachIn)$ has exceeded a pre-determined threshold value.

For the CD-player scenario below, there are three actions encoded: pickup, putdown and press. A number of states are deduced, notably the state of holding (also see Section 7).

5. Experiments

The system set-up as given in Section 2 has been implemented in a component-based system that is distributed over six PCs (on average 1.8 GHz CPUs). This enables operation in frame rate for the hand and object trackers and reasonable detection and recognition times of up to one second. Activity interpretation can follow leisurely hand motions. If hands are moved fast, detection and recognition processes are too slow to report events in time. This set-up has been tested with several persons. In the following we show operation of the system at image level and subsequently evaluate one-hand and two-hand sequences (Section).

Contextual control at reasoning level has been described in detail above. This section will show the results of contextual control at image level. The main goal of contextual control is to limit processing time at components. Near real time can only be reached, if highly focused processing can be achieved. The main source of reducing computing time is to limit the region of interest (ROI) in the image. The region of interest is obtained from projecting the 3D SOI generated by the ORG into each image. Fig. 4 presents two images from a sequence of one of the stereo images during a hand activity. The boxes indicate the Region of Interest (ROI) in the image that is generated from the SOI produced by the ORG. The hand contour is indicated as well as the hand centroids of the last tracking steps. Finally, results of ellipse detection are shown. The ellipse shown in the right images indicates the search for the CD after the put down action.

Since contextual control via SOI most influences computing time, their operation is explained in more detail. A SOI is generated for each request to a vision component motivated by the hand motion. The SOI generated for calculating the ROI in each image is highly task dependent. The following contextual information controls the size and placement of the SOI. The motion of the hand changes the predicted pose and scales the size of the SOI. Faster motion increases SOI size. The hand posture and the expected object (from task knowledge) both influence the SOI size, for example a grasping hand posture produces a larger SOI than a pointing hand posture. Finally, the specific vision task influences the shape and pose of the SOI. E.g., detection of objects the hand might interact with produces a large SOI depending on the nearness parameter set in component ARE. When attempting to verify the existence of an object, e.g., after picking up an object it is checked which object is gone, SOI size is twice the object size. If the vision task is to initialise the object tracker using the result from Object detection or recognition, SOI size is only 20 percent larger than the object size.

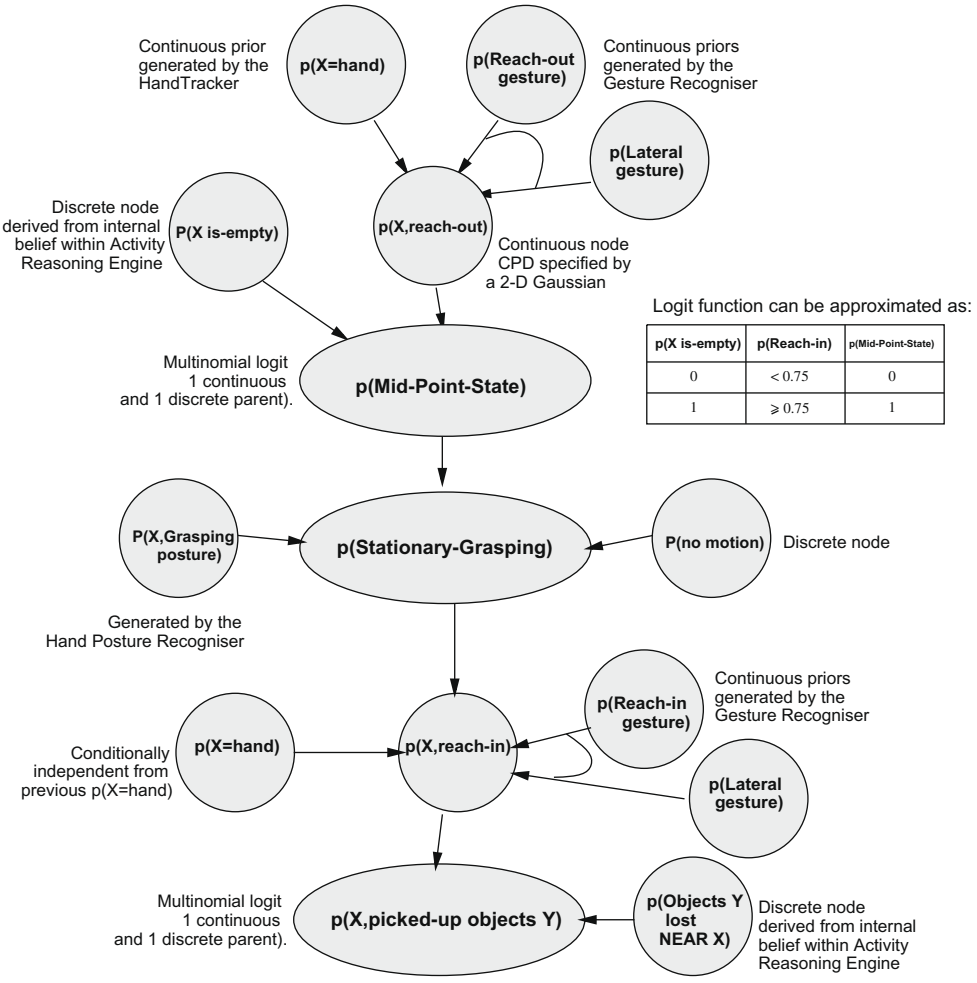

Fig. 3. Template BBN for the pickup action.

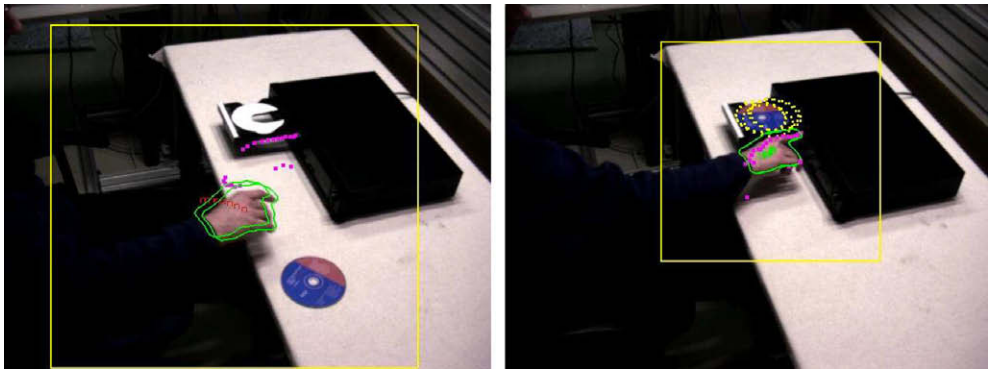

Fig. 4. Two samples of an activity sequence of 223 images at the moment when the first concept is generated, shortly after the open button is pressed, and at the end of the sequence. The boxes indicate the region of interest (ROI) in the image that is generated from the SOI. The hand contour, as well as the last hand centroids and results of ellipse detection are shown. Ellipse detection in the last image operates as confirmation if the CD was placed in the drawer after it has been invisible during the transfer in the hand.

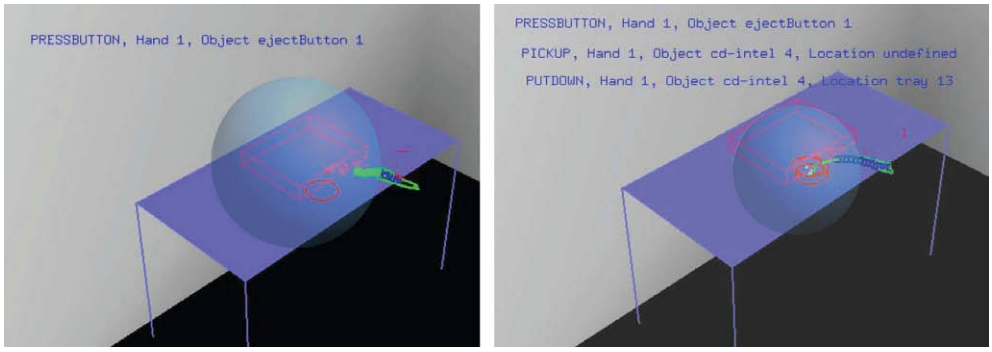

Fig. 5. The same two instances as in Fig. 4 showing the 3D display with annotated data of hand tracks, locations of CDs and the concepts interpreted up to this point. The spheres represent the SOI, larger in the left image to search for an object near the hand, smaller in the right image to confirm the CD in the drawer.

Fig. 5 gives the resulting SOI in correspondence to the two images of Fig. 4. It uses the 3D interface of the framework to display the table configuration with player and objects and the 3D hand trajectory.

The final outcome for the activity sequences shall produce the natural language output: Hand 1 pressed the Eject button, picked up Object cd-intel at an unknown location, put down Object cd-intel in the tray, and pressed the Eject Button. The screen shot in Fig. 5-right is taken after the first three have been reported correctly.

Contextual control enables the near frame-time activity interpretation in this scenario. If processes would run on the full image size, neither tracking could be as precisely as here (e.g., with 3D contour and finger points) nor detection or recognition would deliver results adequately for on on-line activity interpretation. Contextual control focuses processing to typical ROIs of about 150–200 pixels square, which gains a factor of 8–15 over using full images for processing.

Even more drastic is the situation in a two-hand set-up. Using the same scenario, two hands have been used to execute the pressing and grasping motions to put a CD in the CD-player. In this scenario, two main actors are present. For each main actor the same processes of reasoning and of contextual control in ARE and ORG are executed. While ARE finally fuses the high level results to obtain the activity sequence, ORG generates a SOI for each hand. When coordinating the vision components, the two SOIs are merged to reduce processing effort. In some cases, e.g., the hands are far apart, two ROIs remain, e.g., in Fig. 5. Fig. 6 shows an example from the two-hand scenario again with the 3D display. The left and right hand are visible as well as the spherical SOIs. Transformed to the image this means that, e.g., detection operates in a region as indicated in the image on the right.

6. Performance evaluation

The main feature of this work is to achieve real time control for a complete vision system integrating twelve components to interpret activities of a person handling objects. One-hand and two-hand activities have been evaluated with respect to the achieved performance in respect to achieving real-time and extracting the correct symbolic description. This has been evaluated with the use of several sequences. Specifically we report two one-hand and two two-hand sequences, where the speed of hand motion throughout the sequence varies slightly. If requested the sequences can be made available.

Fig. 7 summarises the evaluation.[4] Performance is indicated in relation to achieving the optimal result of reporting the four correct activities as indicated in the previous Section. The slow down factor indicates the slow down versus frame rate of 25 Hz. While the hand tracker operates at frame rate, in particular attentive processes such as object detection and object recognition are slow and cause a delay in reporting results. Hence interpretation at the ARE is made more complicated and ARE needs to keep track of open hypotheses until results become available. An example is the put down of the CD in the CD-player, which is confirmed by looking at the location where the object has been put down. If processing is slow or hand mo-

[4] See http://robsens.acin.tuwien.ac.at/actipret/Sites/videos.htm for the sequences.

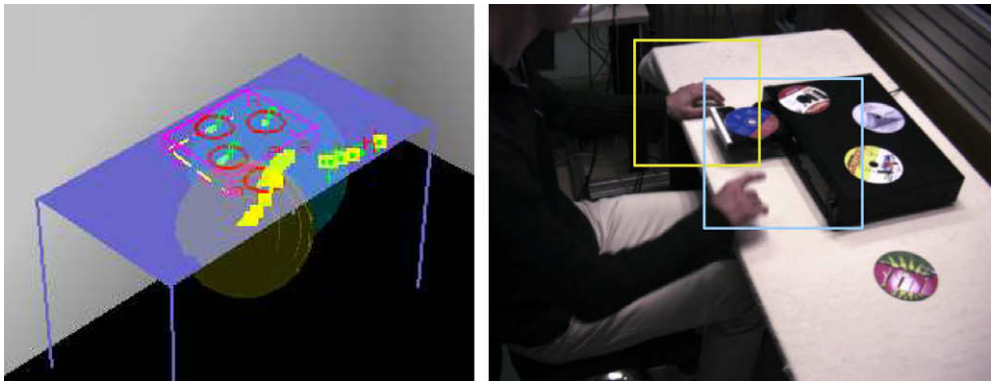

Fig. 6. The spherical (circular) SOIs in a two-hand scenario (left) and the projected ROIs in one of the stereo images (right). The yellow left regions (SOI resp. ROI) are a contextual confirmation step to verify that the CD has been placed in the drawer, while the blue regions indicate the search for potential objects of interest in front of the present hand motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Sequence #	Slow down factor	Performance [%]
1Hand-a	4	100
1Hand-a	2	95
1Hand-b	4	97
1Hand-b	2	95
2Hand-a	8	50
2Hand-a	4	66
2Hand-b	8	71
2Hand-b	4	74

Fig. 7. Results of evaluation on one-hand and two-hand sequences. Example images are given in Figs. 4 and 6, respectively. The performance values are averages over 10 runs. The two-hand scene contains more CDs than the one-hand scene.

tion fast, the drawer of the CD-player is closed before ARE requests a contextual control command over ORG to OR. With a slow down of factor of 4 the best results are achieved for all sequences. Two-hand sequences do not work for a slow down of two due to computational overload. Allowing further slow downs does also not increase performance, since ARE needs to cope with even more hypothesis and hence requests even more time to resolve the interpretation task.

A further evaluation regards specific components or subsystems. Components such as hand tracker or gesture recogniser worked with 100% performance at frame-rate for the one-hand sequences and, given enough time, on the two-hand sequences. Components such as object recogniser or the org subsystem (together with object detection, recognition and tracking) do not show perfect operation even when run alone. For example, OR fails in 2% of recognitions. This means that for each CD and each image of the sequence, in 98% of these cases the CD is correctly localised and recognised. ORG fails in 3% of cases, which means that of course wrong recognition does no allow to initiate tracking and additionally tracking can fail, e.g., the location is not accurate enough.

The advantage of adding a hand posture recogniser (HPR) to the system has been specifically evaluated. Tests have been conducted with the 1Hand-a Sequence over a mean of ten runs as above. Without the HPR performance drops for both slow down factors by 7%. Without HPR the difference between grasping and pointing must be inferred from the gesture and the interaction with the object (CD vs. button). With the HPR an additional cue is available that aids in pruning the hypothesis and, hence, increases system robustness.

To summarise, with high performance PCs this system should operate at frame rate without any slow down factor today. Nevertheless, the example images of the sequences indicate that the set-up is natural though limited to a few CDs. Adding cues from multiple sources, e.g., HPR, increases system performance. Moving to an open setting would require to add even more cues and to further improve all components individually.

7. Results and discussion

The objective has been to demonstrate that a task-oriented approach enables to focus processing such that extensive visual processing (two tracking and four detection and recognition processes, which individually require most processing power of the six computers) is feasible near real-time (i.e., frame rate). Contextual control has been exploited starting from activity reasoning in ARE over spatio-temporal reasoning in the ORG down to coordinating the visual processes.

This interplay of reasoning and visual processes enabled to generate on-line a semantic description of hand–object activities. The system achieves the integration of top-down task/context-driven processes with bottom-up visual processes. A problem encountered in symbolic interpretation is to ground symbols [9]. While it is classically considered a bottom-up problem, the task-oriented approach shows that it is also a top-down problem. A good example is the concept of pick-up-able, which is required to focus processing to relevant items near the hand. It is difficult to translate pick-up-able into specific visual features or processes. Within this work it has been solved by restricting detection and recognition to the specific and known objects for recognition (e.g., categories such as CDs, buttons as part of the CD-player). In a more general set-up it would be also interesting to cope with new objects.

The activities in this scenario could be composed of three actions: grasping, putting down and pressing. They are generic primitives and it was easy to handle a different set of objects and another task, e.g., Lego bricks and a stacking task (see Fig. 8). In this case the hand tracker has been additionally trained to detect the colour of and outline of the bricks.

The activities have been learned (e.g., GR and ARE) and thus achieve a signal to symbol conversion. As detailed in Section 4, the action models are given by the structure of the dynamic Bayesian networks. The instantiated sets of actions making up activity plans are learned from running the system on scenario examples. Of particular value for the on-line performance turned out to be "common sense" assumptions. Examples are the focus on hands as main actor, starting the activity with an empty hand, each activity is executed with one hand at one time, a hand with an object

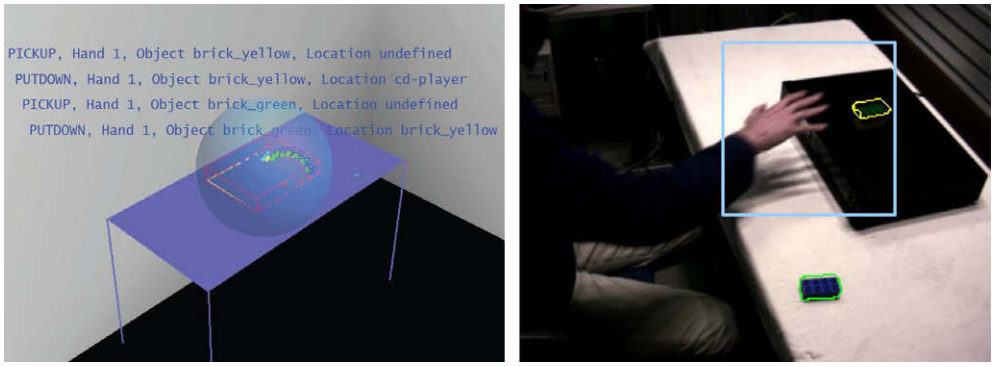

Fig. 8. Activity interpretation for stacking Lego bricks using the pickup and putdown activities. The images are from the situation (the 3D representation with SOI and concepts is right and the image with ROI left) where the hand has just put down the second block. The block is detected with a location on top of the first block.

remains in view, and there is enough visible movement between activities. These assumptions help ARE to select services at ORG (and subsequently the vision processes) and thus aid in pruning the activity hypotheses to achieve real-time processing.

Another principle that turned out to be useful in such a system but also for each of the components, is the principle to clearly design what decision can be made at each level. This results in the proposed architecture and implements the principle that the next higher level will decide what cannot be more locally decided at lower levels. Examples are the nearness reasoning in ORG, which does not consider all possibilities but a reasonable selection and leaves the final decision to the ARE, since the nearest object might not be the correct instant in each case. This principle also naturally exploited the task-driven operation, since higher level processes will also request information to enable this decision making. Another example is the more local vision processing of hierarchical grouping in the ellipse detector [34].

8. Conclusion

This paper presents a system for activity interpretation integrating all levels of a cognitive vision system starting with image-based detection, tracking and recognition approaches, over using intermediate 3D representations of spatio-temporal relationships, to finally arrive at the level of activity interpretation that leads to a symbolic description. The objective was to demonstrate that such a system can be achieved and purposefully coordinated. We then showed that contextual information enables the efficient control of visual processing. In this manner purposive and reactive processing could be combined to arrive at the level of symbolic activity interpretation. The scenario chosen served the interpretation of activities to operate a CD player with one or two hands. Different persons executed the task in a pre-learned (type of objects, gestures and activities) but natural environment. The system was demonstrated life, e.g., at ECCV 2004. Presently the system requires six PCs to operate near frame-rate, however with the steady increase of computing power moving to more complex scenarios or more portable hardware can be realised shortly.

The component-based approach and framework realisation proved to be a valuable tool for integration and to study the interaction of the vision and reasoning components. It provided the necessary tools for debugging such a complex system [25]. The approach to encapsulate capabilities in components is a mean to combine diverse methodologies to vision in one system. It also aids in reusing components for similar tasks. And the data abstraction obtained produces generic concepts (SOI, pose, symbolic) for potential integration with other sensing modalities. As an example, the same set-up has been used to interpret activities in a stacking task. Fig. 8 gives an image and the concepts generated.

The evaluation of the work has been done by using recorded sequences extensively. They helped to improve and strictly evaluate performance. However, visual processes tend to be tuned to (or even adapt and learn) the specific characteristics in the images. Hence, to achieve good performance in an open set-up and different users, development should include real world scenarios regularly. A final lesson learned was that although individual processes (tracking, recognition) become more and more robust by themselves, system robustness could be increased by combining multiple cues reaching from common-sense assumptions to the integration of detection (OD) and recognition (OR) or the additional use of hand postures besides hand tracking.

References

[1] A. Argyros, M. Lourakis, Real-time tracking of multiple skin coloured objects, European Conference on Computer Vision (2004) 368–379. Prague.
[2] M. Ayromlou, M. Zillich, M. Ponweiser, M. Vincze, Measuring Scene Complexity to Adapt Feature Selection of Model-based Object Tracking ICVS'03 Int. Conf. on Computer Vision Systems, Springer-Verlag, 2003. pp. 448–459.
[3] C. BenAbdelkader, L. Davis, Detection of people carrying objects: a motion-based recognition approach, Fifth IEEE International Conference on Automatic Face and Gesture Recognition (2002) 363–368.
[4] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, IEEE Transaction on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.
[5] A.D. Wilson, A.F. Bobick, Parametric hidden Markov models for gesture recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence 21 (9) (1999) 884–900.
[6] H. Buxton, Learning and understanding dynamic scene activity: a review, Image and Vision Computing 21 (2003) 125–136.
[7] C. Capurro, F. Panerai, G. Sandini, Dynamic vergence, Int. Conf. on Intelligent Robotie Systems 96 (1996) 1241–1249.
[8] Cohn, A.G., Hogg, D.C., et al., Cognitive Vision: Integrating Symbolic Qualitative Representations with Computer Vision; pp. 221-246 in [19].
[9] S. Coradeschi, A. Saffiotti, An introduction to the anchoring problem, Robotics and Autonomous Systems 43 (85-96) (2003).
[10] J.L. Crowley, D. Hall, R. Emonet, Autonomic computer vision systems, in: J. Blanc-Talon (Ed.), IE EE Advanced Concepts for Intelligent Vision Systems ICIVS 2007, 2007.
[11] Y. Cui, J. Weng, Appearance-based hand sign recognition from intensity image sequences, Computer Vision and Image Understanding: CVIU 78 (2) (2000) 157–176.
[12] J.A. Fayman, O. Sudarsky, E. Rivlin, Zoom tracking, Proc. IEEE Int. Conf. On Robotics and Automation (ICRA 98) (1998) 2783–2789.
[13] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Pltz, G.A. Fink, G. Sagerer, Multi-modal anchoring for human–robot interaction, Robotics and Autonomous Systems 43 (2003) 133–147.
[14] Howell, A.J., Sage, K.H., Buxton, H., Developing task-specific RBF hand gesture recognition cognitive science research papers, CSRP 562, University of Sussex, UK, April 2003.
[15] T. Lindeberg, K. Brunnstrm, J.O. Eklundh, Active detection and classification of junctions by foveation with a head–eye system guided by the scale-space

primal sketch, European Conference on Computer Vision 92 (1992) 701–709. May.
[16] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image and Vision Computing 22 (10) (2004) 761–767.
[17] J. Moody, C. Darken, Learning with localised receptive fields, Proceedings of 1988 Connectionist Models Summer School (1988) 133–143.
[18] J. Moody, C. Darken, Fast learning in networks of locally tuned processing units, Neural Computation 1 (1989) 281–294.
[19] Nagel, H.-H., Christensen, H.I. (Eds.), 2006. Cognitive Vision Systems-Sampling the Spectrum of Approaches, Proceedings of the Dagstuhl Seminar 03441, October 27–31, 2003, Springer-Verlag, LNCS, vol. 3948, ISBN:978-3-540-33971-7.
[20] Nagel, H.-H., Cognitive vision systems: from ideas to specifications, pp. 57–72 in [19].
[21] C. Noelker, H. Ritter, Visual recognition of hand postures, IEEE Trans. Neural Networks 13 (4) (2002) 983–994.
[22] Obdržálek, Štěpán, Matas, Jiří, Object recognition using local affine frames on distinguished regions. The British Machine Vision Conference (BMVC02) (2002).
[23] Obdržálek, Štěpán, Matas, Jiří, Image retrieval using local compact DCT-based representation, DAGM 2003: Proceedings of the 25th DAGM Symposium (2003) 490–497.
[24] Obdržálek, Štěpán, 2007. Object recognition using local affine frames. PhD thesis, Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University, 2007.
[25] W. Ponweiser, M. Vincze, M. Zillich, A software framework to integrate vision and reasoning components for cognitive vision systems, Robotics and Autonomous Systems 52 (1) (2005) 101–114.
[26] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, R. Dillmann, Using gesture and speech control for commanding a robot assistant, IEEE International Workshop on Robot and Human Interactive Communication (2002) 454–459.
[27] K.H. Sage, A.J. Howell, H. Buxton, Recognition of action, activity and behaviour in the ActIPret project, KI Special Issue (2005).
[28] K.H. Sage, A.J. Howell, H. Buxton, Temporally structured Bayesian belief networks for the recognition of action, activity and behaviour in the ActIPret project, Computer Vision and Image Understanding (2005).
[29] Y. Sato, K. Bernardin, H. Kimura, K. Ikeuchi, Task analysis based on observing hands and objects by vision, IEEE/RSJ International Conference on Intelligent Robots and System (2002) 1208–1213.
[30] Takagi, M., Eberst, C., Umgeher, G., Control of redundant attentive and investigative behaviors in an active cognitive vision system. Workshop on Computer Vision System Control Architectures (VSCA 2003), Graz Austria, 2003.
[31] J. Triesch, C. Malsburg, A system for person-independent hand posture recognition against complex backgrounds, IEEE Trans. Pattern Anal. Mach. Intell. 23 (12) (2001) 1449–1453.
[32] M. Vincze, M. Ayromlou, W. Ponweiser, M. Zillich, Edge-projected integration of image and model cues for robust model-based object tracking, International Journal of Robotics Research 20 (7) (2001) 533–552.
[33] M. Vincze, M. Schlemmer, P. Gemeiner, A. Ayromlou, Vision for robotics—a tool for model-based object tracking, IEEE Robotics & Automation Magazine—Special Issue: Software Packages for Vision-Based Control of Motion 12 (4) (2005) 53–64.
[34] M. Zillich, J. Matas, Ellipse detection using efficient grouping of arc segments, 27th Workshop of the Austrian Association of Pattern Recognition OEAGM/AAPR (2003) 143–148.