A DISTRIBUTED SYSTEM FOR INTEGRATED SPEECH AND IMAGE UNDERSTANDING

Gernot A. Fink, Nils Jungclaus, Franz Kummert,
Helge Ritter, Gerhard Sagerer
University of Bielefeld, SFB 360
Postfach 100131, 33501 Bielefeld, Germany
email: {gernot,nils}@techfak.uni-bielefeld.de

Abstract

Most published work in the field of pattern analysis, in our opinion, underestimates the importance of system architecture or system development. When building large distributed systems both become a prerequisite in order to achieve the overall goal — a complex system solving a demanding task. In our paper we present the approach we have taken and the first solutions we found for an architecture integrating speech and image understanding. We will focus on those aspects that facilitate the process of developing a large distributed system in cooperation between many researchers.

1 Introduction

When reviewing literature in the field of pattern analysis system development or system architecture are often considered to be less relevant than mathematical or algorithmic solutions of individual problems. This may be true as long as the overall system complexity is limited and only a few components can be screwed together using whatever solutions are most readily at hand. If, however, large systems have to be built system architecture and system development may become the most important problems to be solved as they are the prerequisites for the overall goal of a complex pattern analysis system to be achieved.

The goal of the project we are jointly working on with researchers in the field of pattern recognition, artificial intelligence, and linguistics is to study advanced human-machine communication. The machine should be able to process acoustic and visual input and react meaningfully by producing speech output or by manipulating objects in the environment of the communicating partners. This device is called a “situated artificial communicator” where “situated” means, that its capabilities are not meant to be generally applicable but are limited to some given task domain.

The domain was chosen to be the cooperative construction of a toy-airplane with parts from a wooden construction-kit for children. In the first phases of the project the artificial situated communicator shall act as a kind of servant carrying out simple tasks specified by the human instructor. It is not yet intended to plan the construction process. It is, however, intended to recognize acoustically or visually referenced complex objects from the task domain, provide information about its current understanding of the environment, and perform basic manipulations.

Fig. 1 shows a natural communicator and our robot posing for a fictuous scene illustrating what system capabilities we try to achieve in our long-term research project.

In the first phase our goal was to lay the foundations for such an ambitious system, namely:

- bring together a relevant subset of the total capabilities. We decided to integrate speech recognition, speech understanding, object recognition, scene interpretation, and language generation in the first prototype.
- define an architecture for a large pattern analysis system combining the approaches mentioned above. This concept will be described in section 2.
- develop a communication system that makes it easy to implement the logical system architecture on top of it. At least basic mechanisms for inter-module communication have to be provided. Our approach which extends beyond the flexibility of traditional systems in order to make distributed application development less error prone will be outlined in section 3.

* This research was supported by the German Research Foundation (DFG) within SFB 360. Only the authors are responsible for the contents of this publication.
2 SYSTEM ARCHITECTURE

The architecture we defined for our first prototype system is shown in Fig. 2. Its development was influenced by the following aspects:

- Though, we are working in a long-term research project it is not reasonable to believe that all aspects of image processing, speech understanding, or artificial intelligence can be solved and implemented completely from the beginning. Rather, existing solutions for similar partial problems form the baseline of our development efforts. Therefore, architectural restrictions arise from the capabilities of the existing modules and tools used.

- Architecture, in our opinion and from our experiences, is something that can not be defined before the scientific work on individual modules has begun as the quality of input data and intermediate results, the parameters of algorithms, and the methods of interaction can not be planned totally yet. Therefore, in an initial project phase basic development of modules was performed. The capabilities of modules found to be stable then influenced but also restricted the architecture of the system.

The overall system architecture shown in Fig. 2 mainly consists of three logical building blocks:

- In the lower left object recognition is performed by a set of interacting modules based on stereo images of the scene and resulting in 3D-reconstructed hypotheses of the objects found.

- Speech recognition is performed by the modules in the lower right taking into account predictions generated by the understanding component.

- Object and speech recognition are coordinated by a scene and speech interpretation component [1] that is currently implemented as a single module. The knowledge based processing of object and constituent hypotheses uses the semantic network system ERNEST [2]. The understanding component accesses a language generation system that produces synthesised speech output of the system’s utterances. The robotic component is planned to be attached in the oncoming phase of our project.

The internal structure of the speech recognition component was adopted from an integrated system for information inquiry dialogues built in our labs previously [3]. Recognition is basically performed using Hidden-Markov-Models. In contrast to traditional
Figure 2: Overall system architecture

approaches predictions from the speech understanding component initiate and guide the recognition process. These predictions consist of structured linguistic language models for which complex constituent hypotheses are generated incrementally. Details of this procedure can be found in [4, 3]. The basics of the distributed recognition system used were published in [5].

Object recognition is performed in two lines of processing. Starting from intensity images based on edge detection and contour approximation principles...
of perceptive grouping are applied [6] to produce contour group hypotheses (Fig. 3). The second line of processing works region oriented. The baseline form color images in HSV-encoding. They are used for a holistic object recognition applying neural techniques [7] (Fig. 4) and a color based region segmentation. For this purpose a pixel-wise color classification with a polynomial classifier of degree four is calculated and subsequently smoothed. The results are rather reliable as we are dealing with intensely colored objects from a wooden construction kit for children. For the regions obtained (Fig. 5) some simple shape parameters are computed. Together with the raw object hypotheses from the holistic object recognizer these region hypotheses form the input for a knowledge based object recognition module based on the semantic network language ERNEST [2]. This module verifies raw object hypotheses on the basis of color regions and their shape parameters producing more reliable object hypotheses (Fig. 6). E.g. the cube in the upper right of our sample scene was misclassified by the holistic object recognizer. This error could be corrected by the knowledge based module as more complex structural restrictions are applied. The resulting object hypotheses — which are still region oriented — are then combined with corresponding contour hypotheses. All tasks working on 2D data structures only (region segmentation, holistic and knowledge based object recognition, perceptual grouping) are performed in parallel for corresponding stereo images. The 2D object hypotheses from left and right stereo channel are then passed to the 3D-reconstruction module [8]. There based on knowledge of the objects' geometry their 3D-position is calculated approximately (Fig. 7).
The goal of scene interpretation currently is to identify an object in the scene that was referred to by a spoken utterance. First from the results of image recognition a conceptual scene description is built. Within the same knowledge representation framework — the semantic network language ERNEST — also the utterance related to the scene is analysed. From both representations some characteristic qualitative features of objects — either present in the scene or named by the user — are computed e.g. object type, color, and relative size. A Bayesian network is then used to judge the compatibility of “spoken” features with each object hypothesis computed during image analysis. If a single compatible object is found it is selected as the one most likely referred to by the user. Otherwise a simple dialogue is carried out trying to resolve ambiguities [9, 10].

3 Fundamentals

To build up an integrated system running on many machines in a heterogeneous environment with a complexity like the one shown in Fig. 2 it is necessary to have a solid and easy to handle system to support the exchange of data between different modules. Considering well known tools for distributed computing like PVM [11], LAM [12] or DCE[13] we found none of them meeting all our requirements for this task. Also, the ICE system [14] which is used in the VERBMOBIL project and has possibly the most similar goal is limited in its capabilities.

A new approach for a simple but powerful system to integrate large distributed pattern analysis systems is our Distributed Applications Communication System (DACS) [15]. Similar to PVM we use a communication demon on each participating machine that runs in user mode but, in contrast to PVM, allows multiple users to access the system simultaneously and does not provide a virtual machine dedicated to a single user. The demon acts as a router for all internal traffic and establishes connections to demons on remote machines. Fig. 8 shows a possible configuration of the system.

Communication is based on simple asynchronous message passing with some extensions to handle dynamic reconfigurations of the system during runtime. On top of that DACS provides some more advanced communication semantics like remote procedure calls (synchronous and asynchronous) and demand streams that will be discussed in section 4.1. The user interface is kept as simple as possible to allow existing modules to be integrated easily.

All messages transmitted are encoded in a Network Data Representation which includes type and structure information. Therefore it is possible to inspect messages in every point of the system and to develop generic tools that are able to handle any kind of data. A similar representation does not exist in any tool for distributed computing that is used widely.

Every message is routed through the local demon process which allows to easily attach maintenance
tools for debugging purposes. To avoid a potential bottleneck DACS internally uses POSIX threads to handle connections independently in parallel. The system configuration is stored in a database in a central name service to keep the network traffic during dynamic reconfigurations low.

4 Development

4.1 A Special Communication Primitive

DACS provides a special communication primitive for continuous data streams since they typically occur when processing sensor data like live video or speech signals. Since complex pattern analysis systems mostly are not able to process the sensor data in real time usually only parts of the stream are used. To handle this we introduced so called demand streams. A module may provide a data stream using a unique name in the system. Other modules can attach to this stream and receive all the data or well defined parts of it using the basic message passing interface. Demand streams may be synchronized on different hierarchical levels.

For example, the output of the holistic object recognition contains a number of object hypotheses for each processed image. For each object the three best alternative hypotheses are included. In consequence, the basic data type for the demand stream is object hypotheses where the three alternatives are grouped in a first hierarchical level and all hypotheses belonging to one image are grouped in a second one. Fig. 9 shows the grouping and the required insertion of synchronization marks for this example.

In Fig. 9 consumer 1 orders a set of three alternative object hypotheses while the hypothesis A2 is put into the demand stream. The ordering mechanism delays the request until the next synchronisation mark of an appropriate level (2 in this case) occurs and sends the desired information to the consumer. An analog case is shown for consumer 2 that orders object hypotheses for a whole frame.

The advantage of this communication semantic is that all management like registering clients or distributing the data is done by the communication system and the providing module only has to update the stream. Any number of clients can request data from any machine dynamically.

4.2 Generic development tools

The consequent use of included type and structure information in all messages in DACS allows the simple creation of powerful generic tools. The following example shows how to divide large systems into parts to be able to develop single modules without the need of a whole running system.

Fig. 10 shows a generic recorder for demand streams. During the runtime of the whole system it is possible to “tap” demand streams and store all provided data in files without modifying any module. Later, the recorded data may be used to provide the streams again resulting in a virtual running complete system. Again, no modification of the modules is required. This method allows a complex system to be developed by many researchers working on different modules in parallel.

DACS offers some advanced features for controlling distributed applications. The multi-connective interface of the demons is used to attach the DACS Debugging Tool (DDT). Using an easy to handle graphical interface, it allows:
A Distributed System for Integrated Speech and Image Understanding

Figure 10: Using generic tools for stand alone development

- the inspection of every message before it is actually delivered. The demon may be stopped and the messages be sent step by step.

- to watch the configuration of the system using a combined information retrieval from the central name service and the demons.

- to get status information of the connections between applications and the local demon.

4.3 Examples of Practical System Development

The usefulness of the concepts introduced above can most strikingly be shown with examples from real system development. In a very early design phase one has to decide which communication semantics different modules are supposed to use for the exchange of data. On the one hand side one could try to adopt the philosophy of client-server computing. This, however, has the severe disadvantage that a single point of activation is passed through all participating modules. If no very sophisticated interface is implemented on top of the ordinary remote-procedure calls the topmost module in the hierarchy would have to initiate every step of processing down to image grabbing and then patiently wait for the results to be computed. As a second well-known primitive simple message passing could be used. However, this too has several disadvantages. First, e.g. the image grabber would have to know which modules want to process the provided images in order to send the messages to the correct addresses. Secondly, simple message passing does not provide a means for elementary event synchronisation or blocking of data into hierarchical groups.

We found, however, the features listed above that traditional approaches are not able to provide without additional development effort very important when processing sensor data in a complex system. As our architecture currently makes very little use of top-down predictions the bottom up processing e.g. within the object recognition section can be realized completely using the concept of demand streams introduced earlier. This approach offers the following advantages:

- Modules producing data do not have to take care of delivering them to the consumers. This is especially useful when the total number of data consuming modules can not be known in advance.

- Consuming modules can be attached and detached dynamically during run-time. This makes reconfiguration as well as the attaching of inspection and visualisation tools easy. It is also very important for the startup phase of a complex system where not all services can be supplied instantaneously.

- Hierarchical grouping of data items transmitted provides a first simple method of synchronisation between different data streams. Furthermore, logical groups of data items can be formed.

As an example consider the object hypotheses produced by the holistic object recognition. They are
grouped in two hierarchical levels when output on the corresponding demand stream. The smallest groups contain competing object hypotheses for the same color region in the image. On the second level object hypotheses for one image are separated from those of the following image from the sequence recorded.

The use of demand streams is also the basis for our first approach to providing a virtual system context for the development of individual modules. As there is no difference between “real” and simulated demand streams modules do not have to implement different interfaces used during development and within a complete running system. E.g. as long as the combination module was still being worked upon the 3D-reconstruction module could already be tested in a realistic environment by providing results from a sample set of corresponding contour and object hypotheses that were combined manually.

For technical reasons it is often not possible to provide online sensor data or sensor based hypotheses adequately for the different needs of all researchers involved in the development process. With demand streams it is fairly easy to provide realistic data sets of matching color images, color regions, and e.g. object hypotheses. These can be retrieved from a live session with a partial system producing the appropriate results. Afterwards, modules processing these data can be tested off-line but with realistic sensor input easily.

The most useful aspect of our system development environment arises from both the fact that data are transmitted with encoded structure and type information and the special capabilities of demand streams. When beginning the development of a complex pattern recognition system there might not yet exist methods to inspect or even visualize the various types of intermediate results produced as e.g. contour groups or object hypotheses. Within the DACS environment it is easy to attach a generic tool to the various communication links provided — especially also to demand streams. The data provided on this stream, e.g. object hypotheses, can be output in a generic textual external representation and thus already be inspected and checked for plausibility in a very early design phase. This general method can similarly be applied to gain insight about data actually transmitted between applications by our graphical debugging tool DDT. As soon as specialized visualisation tools are available they can be attached to the appropriate streams to dynamically display any data communicated via this mechanism. This can be done while the whole system or only parts of it are running as additional consumers of data communicated do not affect the system’s behaviour.

For our application we developed a generic visualisation tool for all kinds of image based intermediate results of processing, e.g. region or object hypotheses. This tool was used to produce the different images shown in Fig. 3–7.

5 System Performance

For systems performing pattern recognition tasks there exist established and widely used methods for evaluation. However, evaluation of complex pattern analysis systems is still an open problem. In the field of speech understanding some measures were developed (e.g. [16, 3]) trying to condense the performance of a complex system into numerical values. The evaluation of our first prototype is even more complicated as 11 different vision and 2 different speech modules produce intermediate results which contribute to success or failure of the overall result computed by the integrated interpretation component. Therefore, a complete numerical analysis of the prototype’s performance is not yet possible. It will, however, be an important goal for our future work to develop methods for automated testing and evaluation of complex pattern analysis systems.

In [9] a first attempt was made to evaluate the prototype’s performance. This end-to-end evaluation was based on spontaneous utterances produced by 10 naive speakers. The subjects had to refer to randomly selected objects in 11 different scenes. Using a lexicon of 258 words and 12 discourse particles and no language model a word accuracy of 57% could be achieved on this test set. The 2D object recognition was run on the HSV-encoded image data for the 11 different scenes recorded under controlled lighting conditions. The scenes contained a total of 159 known objects from our scenario of 6 different basic types and varying colors and sizes. Scene complexity ranges from small scenes with as few as 5 to 10 objects to large ones containing 25 to 35 different objects. Two of the latter contain also some out-of-domain objects e.g. a human hand. Mostly, objects do not overlap as occlusions can not be handled correctly by our recognition system yet. Scenes were recorded using different focal lengths of the cameras. A size estimation based on parameters of characteristic regions detected in the scene enables the 2D object recognition system to adapt to different object sizes on a single image from a sequence. For the total of 159 detectable known objects the object error rate i.e. the percentage of all classification errors was 25%. If no scenes containing out-of-domain objects
are considered error rate is only 14%. For 57% of the
270 utterances the object referred to by the user was
among the ones selected automatically by the sys-
tem. For 9.6% false identifications were made and
for the remaining 34% no object identification could
be computed due to failures of speech understanding.

Though the overall quality of results can surely be
improved significantly the prototype system allows
quick and easy testing of modules on live data. As
we do not aim at a real time implementation we lim-
ited the frequency of image grabbing to one stereo
HSV image pair and corresponding holistic object
hypotheses every 2 seconds. Region segmentation
takes approximately another 2 seconds including the
transfer of image data over the local area network.
Roughly 1–2 seconds later the first knowledge based
object hypotheses are available. The complete incre-
mental computation takes approximately 7–12 sec-
onds depending on scene complexity. Contour hy-
potheses generation currently is rather slow due to
great numerical complexity taking 50–60 seconds
per image. The incremental speech interpretation
takes about 20–30 times real time [3]. The subse-
quent object identification finishes within less than 5
seconds including answer generation or visualisation
of possible objects referenced.

Those approximate runtime figures assume low com-
peting traffic on the local area network, 8 or more
high end workstations with a computational power
of 100–150 SPECint92 and a special purpose image
processing system used for image grabbing and cal-
culation of initial hypotheses. With this hardware
configuration the system was successfully presented
to the reviewers of the German Research Foundation
(DFG) during the evaluation of SFB 360 in March
1996.

6 Conclusion

In this paper we presented a distributed system that
integrates speech and image understanding. We pro-
posed an architecture for a large pattern analysis sys-
tem running in a heterogeneous computing environ-
ment. The integration of different modules is based
on DACS — our approach for a simple but power-
ful tool for system integration. It offers many use-
ful features for developing and maintaining large
distributed systems. The advantages of the concepts
introduced over traditional approaches were demon-
strated with examples from the development of our
first prototype system.

Acknowledgements

We would like to thank all our colleagues from
SFB 360 who provided modules of the prototype sys-
tem for their useful discussions and active support.

References

[1] U. Naeve, G. Socher, G. A. Fink, F. Kummert,
and G. Sagerer, “Generation of language mod-
els using the results of image analysis,” in Proc.
European Conf. on Speech Communication and

[2] F. Kummert, H. Niemann, R. Prechtel, and
G. Sagerer, “Control and Explanation in a Sig-
nal Understanding Environment,” Signal Pro-
cessing, special issue on ‘Intelligent Systems
for Signal and Image Understanding’, vol. 32,

und Sprachverstehen, vol. 103 of Dissertatio-
nen zur Künstlichen Intelligenz. Sankt Au-

“A Close High-Level Interaction Scheme for
Recognition and Interpretation of Speech,”
in International Conference on Spoken Lan-
guage Processing, vol. 4, (Yokohama, Japan),

“A New Approach to Distributed Word Recogni-
tion,” in Proceedings in Artificial Intelligence
(H. Niemann, R. de Mori, and G. Harrrieder,
ed.), vol. 1, (Sankt Augustin), pp. 253–256, in-

Groupings in a Contour-Based Approach,” in
Second Asian Conference on Computer Vision,

object recognition architecture using optimized
gabor filters,” in Proc. 13th Int. Conf. on Pat-
tern Recognition, vol. IV, (Vienna), pp. 70–74,

[8] G. Sagerer, F. Kummert, and G. Socher,
“Semantic models and object recognition in com-
puter vision,” in International Archives of Pho-
togrammetry and Remote Sensing (K. Kraus
and P. Waldhäuser, eds.), vol. XXXI, Part B3,
Commission 3, (Vienna), pp. 710–723, Com-
mmittee of the XVIII International Congress for

[9] G. Socher, G. A. Fink, F. Kummert, and
G. Sagerer, “A Hybrid Approach to Identifying
A Distributed System for Integrated Speech and Image Understanding


