The Ulm Sparrows 2006

Gerd Mayer, Ulrich Kaufmann, Martin Clauss, Carmen Hartmann, Michael
Monsch, Michael Munz, Thomas Ruland, Bernhard Seibold, Christoph Sitter,
Fabian Wolf, and Günther Palm

Neuroinformatics
University of Ulm
D-89069 Ulm, Germany

1 Introduction

The Ulm Sparrows RoboCup team has successfully competed now for eight years in the RoboCup tournaments. It is mainly a student-driven team and serves both as a research project as well as a teaching tool at the University of Ulm. We try to develop solutions not only applicable in the RoboCup scenario but in other robotic projects too.

Three main contributions are presented within this paper: our middleware for mobile robots Miro including its framework based building blocks for common robotic tasks, our visual object detection system and our current and ongoing work in the field of communication and interaction in teams of heterogeneous robots.

2 Miro– Middleware for Robots

Miro [1] is a complete middleware for autonomous mobile robots developed at the University of Ulm. It includes up to date technologies and follows modern approaches like object oriented programming, a modular architecture, and XML parameterization. Following a list of major topics:

- Network transparency by using CORBA: Software modules that implement CORBA interfaces can be easily plugged together, even if they are located on different computers. Even more, different modules can be written in different programming languages.
- Event based publisher/subscriber model: To enable modules to receive asynchronous events from other parts of the architecture, an event-based publish/subscribe model is used. Locally on a system, the CORBA Notification Service is responsible for event distribution. Once events have to be transmitted to other robots, events are sent via IP-multicast. This approach saves bandwidth and decouples the individual robots.
- Generic logging facilities: Miro contains a framework for generic logging to save sensor data and internal world model states during program execution [2]. After the run it is possible to transparently replay the data e.g. for debugging purpose or for neural network training.

- Configuration and parameter framework: Configuration parameters can be specified using XML files which are then transformed to native source code. The actual configuration again is specified in XML files and are used to configure e.g. the behavior configuration and the image processing tree (see below).
- Sensor and actuator services: There are different sensor and actuator services on different abstraction levels. So it is very easy to generate a new sensor or actuator service by specialize the generic service. Using these services and the CORBA layer it is easy to distribute the data for example over the network.

Miro is not only used by our RoboCup Team but also in other projects, for example the MIRRORBOT (European Union IST-2001-35282t) project [3]. The RoboCup teams Mostly Harmless (TU Graz, Austria) and Agilo RoboCuppers (TU Munich, Germany) also rely on Miro even though the hardware is very different. Thus, Miro has proven to be a very flexible framework for heterogeneous robot platforms in various scenarios.

2.1 Hierarchical Behavior Organization

Miro additionally contains a couple of frameworks for common robot tasks. One of these frameworks is the so called BAP framework [4] , which initials stands for its basic building blocks, the “Behaviors”, “Actionpatterns” and “Policies”. This BAP framework allows to build complex and sophisticated reactive control programs by use of hierarchical structuring methods. Major design issues considered during development are outlined next:

- Reactivity by supporting concurrent behavior execution: Behaviors can be invoked time triggered and event triggered, but they can also run in parallel their own thread.
- Flexibility by allowing different arbitration mechanisms: Every single action pattern, i.e. a set of behaviors, can use its own arbitration method allowing for specific, problem oriented decision schema.
- Taskability by supporting behavior sequencing: Allows to execute only the minimal set of behaviors at any one time and switch to another set of behaviors to produce the next desired system functionality.
- Functional abstraction by providing behavior parameterization: Fully integrated into Miro. This allows to use the already mentioned configuration and parameterization framework.
- Modularity by hierarchical policy specification: One or more action patterns are contained in a policy, but also so-called sub-policies may be added to a policy. Thus, BAP supports modeling of policy hierarchies. This allows to model even very complex reactive control programs without loosing the overview.

An additional highlight is the possibility to dynamically reconfigure the control program during runtime and therefore includes support for learning and adaptivity during development. Extensive tool support allows to model the control structure using a easy to use GUI based program. Again, the use of this framework is not limited to RoboCup, but can be used in different scenarios like e.g. for a classical indoor localization and mapping task [5].

2.2 Video Image Processing

Another framework included into Miro is the video image processing framework [6] [7]. It is designed especially to ease the development of robot vision applications and to mediate between the partially contradictory requirements of advanced vision processing in a real-time constraint environment. In particular, a framework for robot vision must address a simplified specification of image processing architectures, control and synchronization issues of image processing steps, and the integration of the image processing architecture into the overall robot control architecture.

- Organization of control and data flow: That is, the vision application programmer only needs to implement the individual image operations and direct the data flow for the target application. The VIP framework then executes the implemented code as the execution logic implies.
- Parallel and asynchronous evaluation: Each processing tree its executed within its own thread and is processed in parallel, while the data flow can stay connected. The framework then ensures appropriate synchronization between the image streams. Additional processing trees can be added to decouple time-consuming image operations (e.g. complex object detection) from fast image evaluations, needed at full frame-rate (e.g. obstacle detection). Each processing tree can be executed with its own thread priority and scheduler choice.
- Timeliness and resource management: In order to maximize performance, VIP keeps track of which filters are actually queried by client modules. Based on this connection management, a dynamic graph pruning is performed to process only the minimal required filter tree.
- Communication of results: The results can be polled by consumers or send to a notification channel. Additionally, it is possible to show the results and provisional results of the trees for testing.
- Development Support: Extensive development support is provided in the form of parameter management, GUI-based configuration as well as generic inspection of images and meta-information.

3 Image Processing

Image processing on our soccer robots is beyond a pure color segmentation approach. This step is only used as very fast and simple attention control. All complex objects are then validated using artificial neural networks. Advanced image processing is up to now only rarely done in the middle-size league in particular and RoboCup in general.

3.1 Neural Object Classification

The approach presented here is a multilevel architecture. In the first step, potential object positions are searched within the recorded images to direct the robots attention to possibly interesting places. Subsequent image processing is restricted to these regions of interest (ROI). This is used to save computational resources and to omit obvious noise in the image, that might distract some specialized feature detectors. In order not to miss important information in the image, the ROI detection must not be overly restrictive, so false positives will be upon the ROIs.

In the next step feature vectors are calculated for each of the detected ROIs. They describe different attributes of the object as general as possible to be sensitive to the different views of the object or different occurrences of the same type, yet specific enough to allow to reliably reject false positives. Used features are for example simple geometrical properties, like the width and height of the object, or histogram representations of various different features, like e.g. color distribution or local orientation information.

Finally, all these features (i.e. the vector describing the features) are passed to the object classification. Here, artificial neural networks vote for each feature vector whether it belongs to the expected object type or not (this applies, if there is one dedicated attention control module for each object type like used in the RoboCup scenario. Alternatively, a general purpose attention control module may just emit unlabeled regions of interest and a subsequent, more complex hierarchical neural network classifier then estimates the found object type [3] as well as a confidence value for this classification result). For a more elaborate description of the procedure see [8] or [9].

3.2 Temporal Integration

Traditionally, object detection and temporal position integration are realized in two different, separated modules. But when interweaving the temporal integration directly into a multi-stage classification process, one receives a lot of advantages, which seems to be contradictory at the first sight. On the one hand, the classification results are stabilized by the temporal integration, on the other hand, the effort needed for the object retrieval can be reduced significantly [10]. Even more, by integrating both modules, we can consider a lot more types of information like e.g. different certainty values for different stages of the classification or statistical properties of the environment [11].

These improvements in object classification and tracking are of great advantage especially in highly dynamic environments like e.g. the RoboCup, as they allow for a reliable, precise and fast visual object classification system.

4 Team Behavior and Cooperation

Due to scientific as well as pragmatic reasons, there is a growing interest in the robotics field to join the efforts of different labs to form mixed teams of autonomous mobile robots. The recent rule change in the RoboCup middle-size league allows for more robots per team, however, the limited financial resources and the additional maintenance effort for further robots exceeds the capabilites of many research labs. Mixed teams are also motivated from a scientific perspective. They introduce the research challenge of cooperation within teams of extremly heterogeneous autonomous mobile robots.

For cooperation between robots, the sharing of information about the environment is initially sufficient for successful cooperation. If all robots share both the same belief about their environment, as well as the same set of goals, similar conclusions should be drawn.

To share beliefs, the teams must agree upon structures that encapsulates this information. The interface for sharing this world model informations is realized in a project called SharedBelief[1] (described in detail in [12]). The three main elements are a time-stamp, the probabilistic dynamic pose of the robot itself, and a list of observed objects. We already demonstrated successful cooperation using this infrastructure in Lisbon with the RoboCup teams from the Universities of Graz and Munich.

5 Conclusion

As we want to continue research in this direction, we're currently working in a joint project with Munich to learn models of different teammates to enhance the robot's own comprehension of other robot's intentions. We further try to deepen the research in this field with the University of Kassel now.

Further research effort is done in the field of a more complex, yet still very fast attention control which is currently based on a very simple color segmentation algorith using fixed color space boundaries.

Additional frameworks for common robot tasks are likely to be integrated into Miro as they evolve during our constant work in the field of autonomous mobile robots in general and the robot soccer domain in particular.

Acknowledgments

The work described in this paper was supported by DFG SPP 1125 within the project *Adaptivity and Learning in Teams of Cooperating Mobile Robots.*

References

1. Utz, H., Sablatnög, S., Enderle, S., Kraetzschmar, G.K.: Miro – middleware for mobile robot applications. IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures **18** (2002) 493–497

---

[1] Miro and the shared belief state specification are both freely available at `http://smart.informatik.uni-ulm.de/MIRO/` resp. `http://sharedbelief.sourceforge.net`.

2. Utz, H., Mayer, G., Kraetzschmar, G.K.: Middleware logging facilities for experimentation and evaluation in robotics. 27th German Conference on Artificial Intelligence (KI2004), Ulm, Germany (2004) Workshop on Methods and Technology for Empirical Evaluation of Multiagent Systems and Multirobot Teams.
3. Fay, R., Kaufmann, U., Knoblauch, A., Markert, H., Palm, G.: Combining visual attention, object recognition and associative information processing in a neurobotic system. In Wermter, S., Palm, G., Elshaw, M., eds.: Biomimetic Neural Learning for Intelligent Robots. Intelligent Systems, Cognitive Robotics, and Neuroscience. Volume 3575 of Lecture Notes in Computer Science LNAI. Springer, Berlin, Heidelberg (2005) 118–143
4. Utz, H., Kraetzschmar, G.K., Mayer, G., Palm, G.: Hierarchical behavior organization. In: Proceedings of the 2005 International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, IEEE/RSJ (2005)
5. Kraetzschmar, G.K., Pagès Gassull, G., Uhl, K.: Probabilistic quadtrees for variable-resolution mapping of large environments. In Ribeiro, M.I., Santos Victor, J., eds.: Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles. (2004)
6. Utz, H., Kaufmann, U., Mayer, G., Kraetzschmar, G.K.: VIP – a framework-based approach to robot vision. Journal on Advanced Robotics, Special Issue on Software Development and Integration in Robotics (2005) (to appear).
7. Utz, H., Kaufmann, U., Mayer, G.: Advanced video image processing on autonomous mobile robots. 19th International Joint Conference on Artificial Intelligence (IJCAI) (2005) Workshop on Agents in Real-time and Dynamic Environments.
8. Kaufmann, U., Mayer, G., Kraetzschmar, G., Palm, G.: Visual robot detection in robocup using neural networks. In: Proceedings of RoboCup-2004 Symposium. Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, Springer-Verlag (2004)
9. Mayer, G., Kaufmann, U., Kraetzschmar, G., Palm, G.: Neural Robot Detection in RoboCup. In: Biomimetic Neural Learning for Intelligent Robots. Springer, Heidelberg (2005)
10. Mayer, G., Melchert, J., Utz, H., Kraetzschmar, G., Palm, G.: Neural object classification and tracking. In: 4th Chapter Conference on Applied Cybernetics, IEEE Systems, Man and Cybernetics Society (2005)
11. Mayer, G., Utz, H., Palm, G.: Information integration in a multi-stage object classifier. In: 19. Fachgespräch Autonome Mobile Systeme AMS 2005. (2005)
12. Utz, H., Stulp, F., Mhlenfeld, A.: Sharing belief in teams of heterogeneous robots. In Nardi, D., Riedmiller, M., Sammut, C., J.Santos-Victor, eds.: RoboCup 2004: Robot Soccer World Cup VIII. Volume 3276 of Lecture Notes in Artificial Intelligence., Berlin, Heidelberg, Germany, Springer-Verlag (2005)