Image Processing for Next-Generation Robots

Gabor Sziebig[1], Bjørn Solvang[1] and Peter Korondi[2]
[1]Narvik University College,
[2]Budapest University of Technology and Economics,
[1]Norway
[2]Hungary

1. Introduction

During the 21st century rapid progress in computer communication and technologies influences robot systems: becoming larger and more complicated than ever previously. To manage the rapidly growing sensor data the single robot systems transformed to networked systems, facing new challenges. The demand of low-cost mass production of robots (industrial or service type), the rising number of elderly people, who needs support of their everyday life, called forth of middleware technology in robot technologies (Brugali & Reggiani, 2005). The middleware technologies provide the heterogenic environment that hides low level functions (hardware specific implementations) and provides the required flexibility for robot system developers. In the beginning of the middleware developments it was doubtful, that the technology causes retardation in speed, but after benchmarks of Christopher D. Gill and William D. Smart (Gill & Smart, 2002) it showed that it has more advantages then disadvantages. They strongly believe that Common Object Request Broker Architecture (CORBA)–based middleware offers great advantages in robotics and embedded sensor applications. The urgent need of middleware forced the robot programmers create their own middleware's that meets best their need. Unfortunately, most of the pioneering initiatives are developed independently of the others, driven by specific applications and objectives (Kotoku & Mizukawa, 2006).
In conventional robot system development, the different robot parts (sensors, processing elements and actuators) are combined together in a compact, self contained system. Both type of robot system (industrial or service type) faces challenges in the sense of flexibility, reusability and new part integration. In case of industrial robots, the end-user has very limited intervention to the control system itself (Brogårdh, 2007) and in case of the service robots, the human – robot co-existence demands an extreme flexibility to care take all of human ever changing needs. Robot systems usually consist of sensors that provide information about the environment, computational units that process information about the environment and actuators that are controlled according to the decisions made by the computational units. In recent robot systems one of the most important sensors are image sensors that provide visual information about the environment. Effective robot system design requires that image sensors and image processing functionalities can as easily be integrated with robot systems as any other component. The vision related components of a robot system should thus be integrated using the same middleware as the robot system

Source: Computer Vision, Book edited by: Xiong Zhihui,
ISBN 978-953-7619-21-3, pp. 538, November 2008, I-Tech, Vienna, Austria

itself. Thus, development of new robot systems (industrial and service type) should be addressed with great concern of the user demand for flexible solutions.

In Japan the basics of Next-Generation Robots are being developed with the purpose of improving the efficiency of robot design. The development was started in 2004 at the International Robot Fair, Fukuoka, Japan. The following three expectations were defined for the Next-Generation Robots (IRF, 2004):

1. Be partners that coexist with human beings.
2. Assist human beings both physically and psychologically.
3. Contribute to the realization of a safe and peaceful society.

To achieve these expectations new technologies are being promoted and spread in wider areas. To satisfy every user's individual needs, robot systems must be constructed more flexibly. Creation of new functions is made easy by using RT (Robot Technology) - Middleware (Ando et al., 2005 b), which is a modularized software supporting robot program development for Next-Generation Robots (Ikezoe et al., 2006). RT-Middleware was chosen as the platform also in the proposed work, because it is the only middleware solution that is under standardization (Object Management Group, 2007). This solution has proved to be industry ready and used by many industrial partners (Toshiba (different system components), Honda (ASIMO humanoid robot), AIST (OpenHRP humanoid robot), etc.) and also many research institutes (Ando et al., 2005 a).

In this chapter the Distributed Image Analyzer and one example application of the framework will be introduced. Distributed Image Analyzer is a distributed image processing framework for RT-Middleware. Originally it was designed to be grid computing like distributed image processing framework (with own communication interfaces), which was developed to increase processing speed by applying distributed computational resources. The modules created for the grid type version can be utilized without any change in the RT-Middleware based. As the system is modular, the high computational costs of image processing can be shared with different components; on board the robot or the environment can be utilized to execute the computation.

The rest of this chapter is organized as follows: in Section 2 the idea and basic structure of the RT-Middleware is presented. In Section 3 the Distributed Image Analyzer framework is presented. In Section 4 example application of vision system in industrial environment is introduced.

2. RT-Middleware

This section is intended to give an overview about the RT-Middleware. The proposed distributed image processing framework is built upon this middleware and will be introduced in the next section.

In 2002 the Japanese Ministry of Economy, Trade and Industry (METI) in collaboration with the Japan Robot Association (JARA) and National Institute of Advanced Industrial Science and Technology (AIST) started a 3 year-national project "Consolidation of Software Infrastructure for Robot Development". With the purpose of implementing robot systems to meet diversified users' needs, this project has pursued R&D of technologies to make up robots and their functional parts in modular structure at the software level and to allow system designers or integrators building versatile robots or systems with relative ease by simply combining selected modular parts (Ando et al., 2005 b).

To realize the proposed robot architecture, a robot technology middleware was developed, named "OpenRTM-aist", where OpenRTM stands for Open Robot Technology Middleware.

The concept of RT-middleware can be seen in Fig. 1. Not only thousands of hours of robot programming could be saved, but even more the interoperability between simulation and real applications in robots is solved.

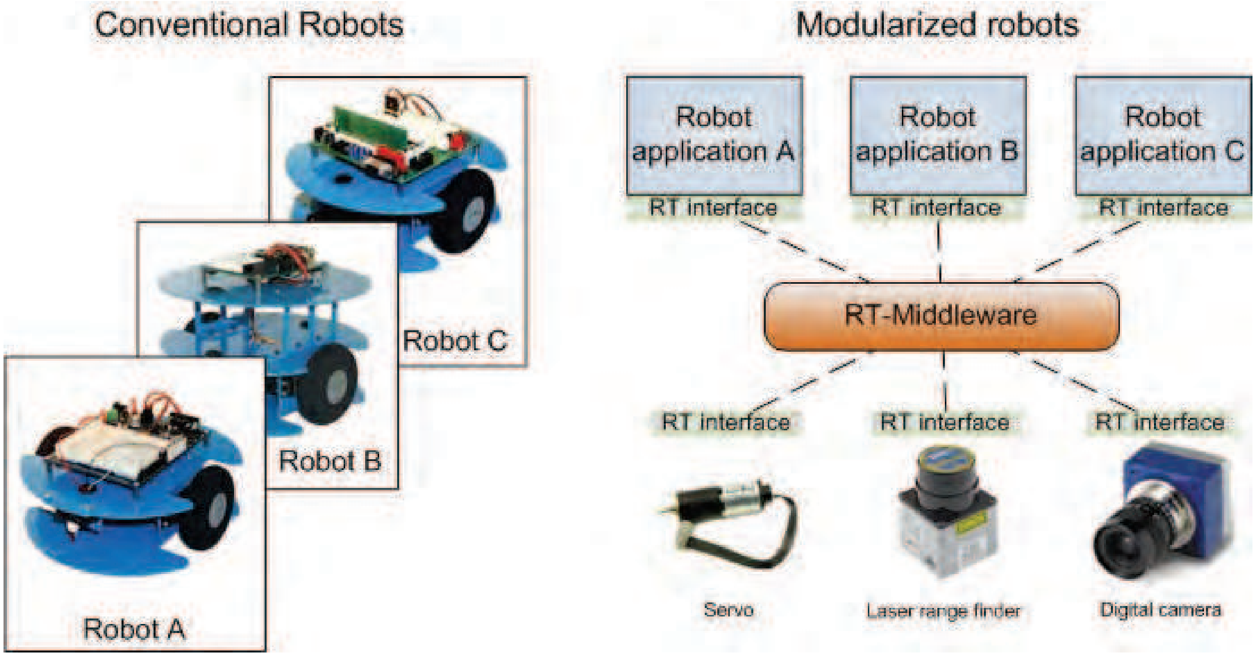


Fig. 1. Difference between the conventional and modularized robot concepts

Two prototype systems have been made to ascertain the effectiveness of the developed RT-middleware (Ando et al., 2005 a).

1. A robot arm control system based on real time control.
2. A life supporting robot system (also known as iSpace (Lee & Hashimoto, 2002)), one of the promising applications.

2.1 Architecture

The framework's basic functional unit is the RT-Component. Modularization is achieved by utilizing the RT-Components. The necessary functions, structure and a realization method based on distributed objects are defined within this component. Fig. 2. shows the architecture block diagram of the RT-Component.

For reasons of platform independency, the RT-Components are modelled using CORBA as a distributed object middleware. An RT-Component consists of the following objects and interfaces (items not listed here are mainly for administrative functions and description can be found at (Ando et al., 2005 a)):

- Component object (Describes the component itself (e.g. name))
- Activity (Core logic, this must be implemented separately in every component)
- InPort as input port object (Data is received here)
- OutPort as output port object (Data is sent from here)
- Command interfaces (Management of Component object)

In general the distributed object model can be described as some interfaces that contain operations with parameters and a return value. Every single component has the same structure (contains the same interfaces) and the only difference is inside the core logic (Activity) and the number of the InPort and OutPort. This allows system transparency, creating a “black-box” of all components. Every RT-Component has an activity which is responsible for data processing with the purpose of device control, such as controlling a motor drive, speech recognition, video processing, etc. The life-cycle of an RT-Component can be observer in Fig. 3.

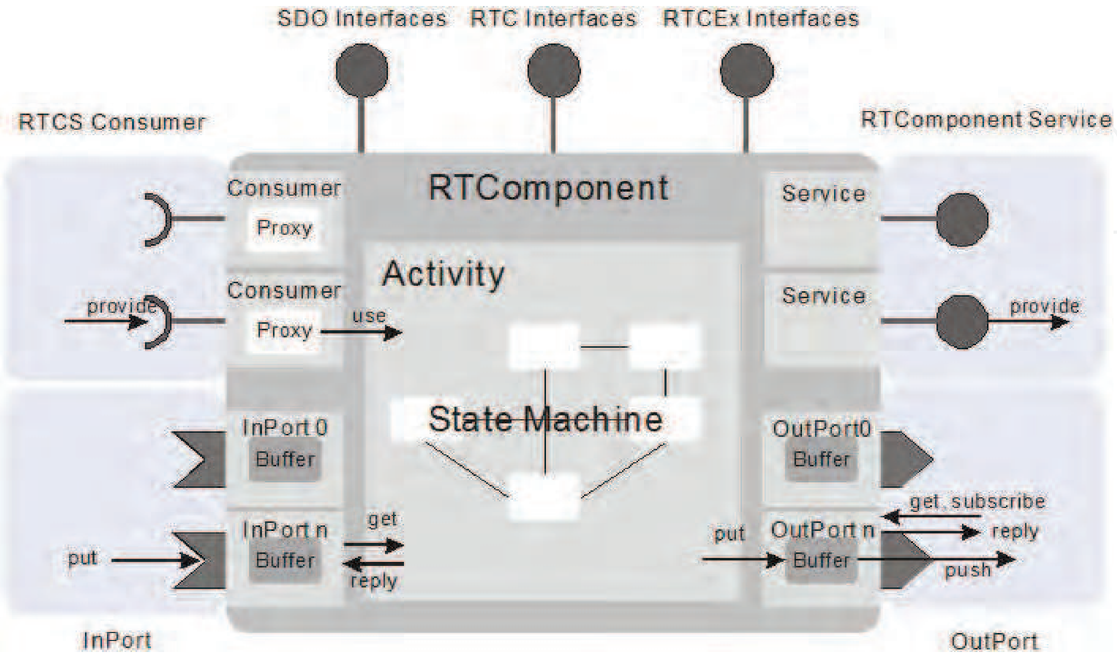


Fig. 2. RT-Component architecture (OpenRTM-aist, 2007)

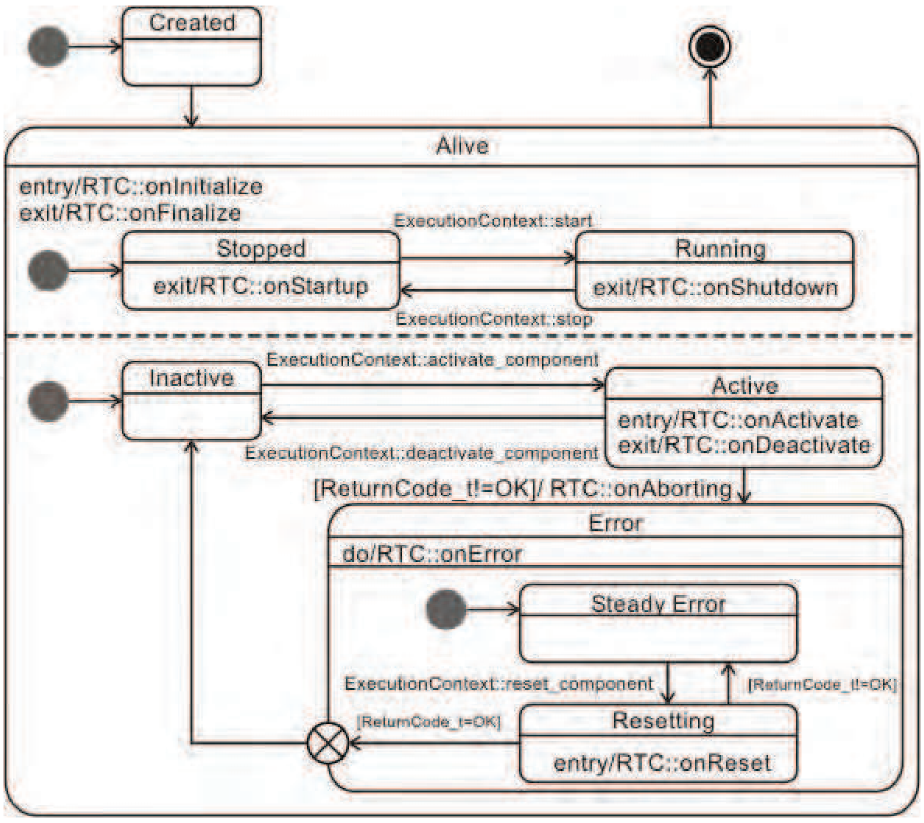


Fig. 3. RT-Component life-cycle (OpenRTM-aist, 2007)

Once the RT-Components are filled up with content and ready to be used, they have to be integrated to form a robot system. The assembly of a system is helped by a Graphical User Interface (GUI) tool that manages a connection of InPort/OutPort between RT-Components like a control block diagram and performs activation/deactivation of an RT-Component. This GUI can be seen in Fig. 4.

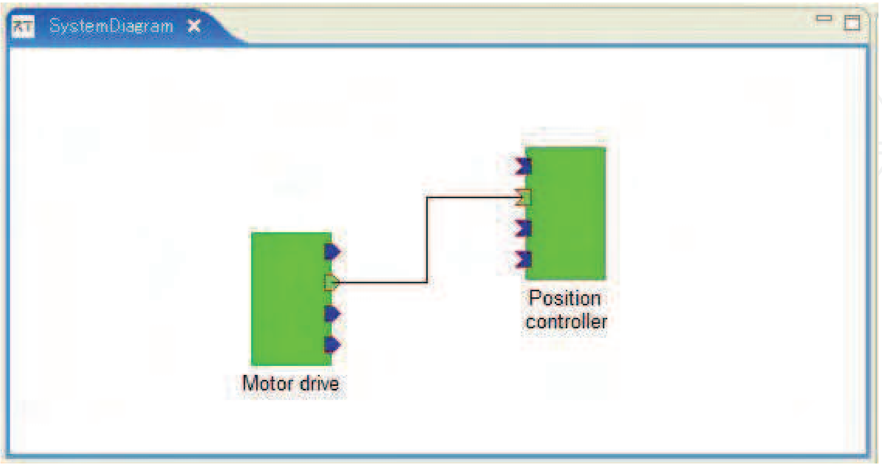


Fig. 4. User interface for system construction

The scope of this chapter is narrow to involve further details about RT-Middleware and its evaluation. More can be found in references (Ando et al., 2005 a), (Ando et al., 2005 b).

3. Distributed image analyzer

This section presents the Distributed Image Analyzer, which is a framework for distributed information processing, with a special respect on image processing. It was designed to overcome the heavy computational load of image processing and vision related operations. Image processing tasks are packed in modules and are distributed on several computers, forming a grid computing system. Also it provides a high level of modularity so that modules can easily be connected with each other. Another consideration is that the display of visual data can be done on any of the participating computers. This allows simultaneous supervision of many steps of an image processing algorithm on many screens connected to the computers participating in the framework. This feature is usually unavailable with most of the grid computing systems. The framework can accommodate several modules that are processing nodes of a dataflow-like module graph.

New modules can easily be constructed and added to the framework; only the information processing function has to be implemented. The modules represent the operators for information processing in the framework. They are designed to cooperate as a distributed network of modules, allowing a higher level of complexity. A module provides standard interfaces for communication through a container. Every module can be treated the same way, as a “black-box”. The development of a new module is very efficient. A new module is derived from the same class (*CoreModule*). This holds the communication interfaces and the Application Programming Interface (API), only the data processing part of the module has to be implemented.

The modularized system architecture is not only achieved by standardized communication, in Distributed Image Analyzer framework the Dynamically-Loadable Library (DLL) technology is also used, which is a basic technology in Microsoft Windows environment. Every module is isolated in one DLL file and can easily be distributed over the network. There is no need for setup or installation, the DLL containing a certain module is copied to a specified directory and can be immediately used. This becomes possible by introducing a parent class (*CoreModule*) for the modules, where the basic interfaces are specified as virtual functions. When a new module is created, it is derived from *CoreModule*, and the virtual functions are implemented in order to perform the necessary data processing functions. The framework handles the modules through the interfaces defined in *CoreModule*, and has no specific information about the inside of any module. A DLL containing a module is loaded into memory and treated as a *CoreModule* type. The communication between modules is based on events: there is an event for data receiving (called *IncomingDataEvent*) and data sending (called *OutgoingDataEvent*). The management of each module and connection establishment between modules are done by the framework. This architecture can be seen in Fig. 5.

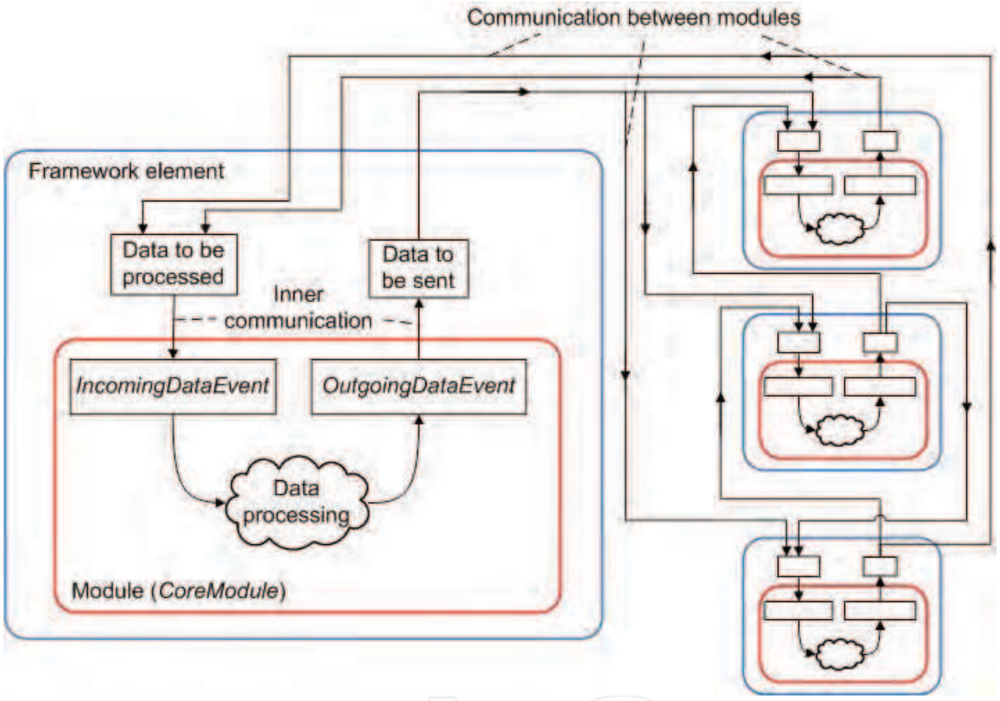


Fig. 5. Architecture of Distributed Image Analyzer

As a reference the following modules are implemented:

- Camera Module, to simulate an eye of a robot
- Edge Detector Module, to detect edges
- Motion Detector Module, to demonstrate an application of image processing
- Colour Mode Converter Module, conversion between colour spaces
- 2D-3D Converter Module, 3D stereo calculations
- Display Module, to display the result of image processing

The similarity between the architecture of the Distributed Image Analyzer framework and the RT-Middleware, which can be noticed from the previous paragraph, makes the module integration to RT-Middleware easy. Only an interface conversion is needed between the modules used in Distributed Image Analyzer framework and RT-Middleware. This allows new image processing algorithms to be tested without involving a robot and after successful test, seamless integration to real environment is possible. This is demonstrated in Fig. 6.

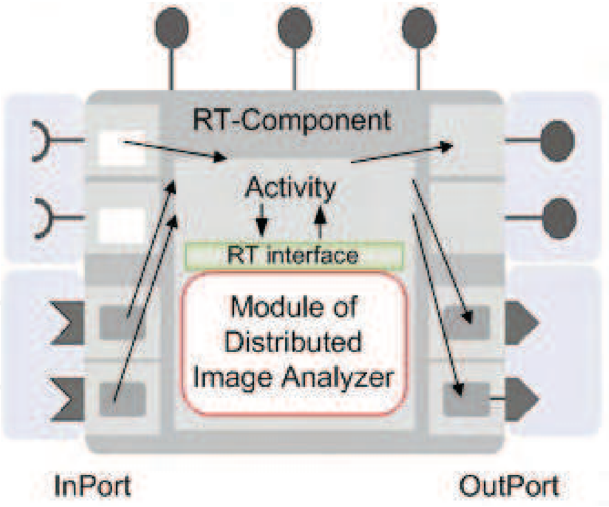


Fig. 6. Integration of an image processing module to RT-Component

After the introduction of the basics of a vision system for robots, in the next section one of its applications is shown.

4. Vision system for old Numerical Control (NC) machines

One example of vision systems in industrial environment is a supervisory system of old NC machines; these machines are usually manufactured before 1980s and lacks capability of Input/Output communication. In that time, these were produced to be as user friendly as possible and only had output through low quality monitors. Even though some might be able to communicate with Personal Computers (PC), it was only one way communication and can only be used from program uploading/downloading. In order to monitor the state of manufacturing process an operator should always observe the screen of NC machine and decisions made by the operator are mainly based on the information shown on the screen.

Based on this idea the operator can be replaced with a vision system, which can detect changes in the screen of machines and can also provide input for other elements in flexible manufacturing systems, also an industrial robot that is used for feeding the NC machine with raw material can be utilized to "operate" the machine by pushing buttons on the operator console. Full remote control of old NC machines can be achieved by this solution. Fig. 7. shows a flexible manufacturing system with an industrial robot and an NC machines.

In this case the vision system is constructed of a digital camera and a PC, which uses image processing tools for Optical Character Recognition (OCR). From the image acquired by the camera, numerous data can be extracted (position, distance to go, current line of the running program, operator messages, etc.). These data describes the current state of the NC machine.

For OCR artificial neural networks are trained and used for character recognition. This makes the system robust and self learning. The novelty of the system is not the technology behind (OCR is used for over 20 years now), but the application of it in such an industrial environment. OCR is mainly used in industry for tracking objects (serial number recognition, container localization (Elovic, 2003)), but in this case utilized as a supervisory system.

In the following section after a small introduction of neural networks, the structure and realization of the vision system will be introduced.

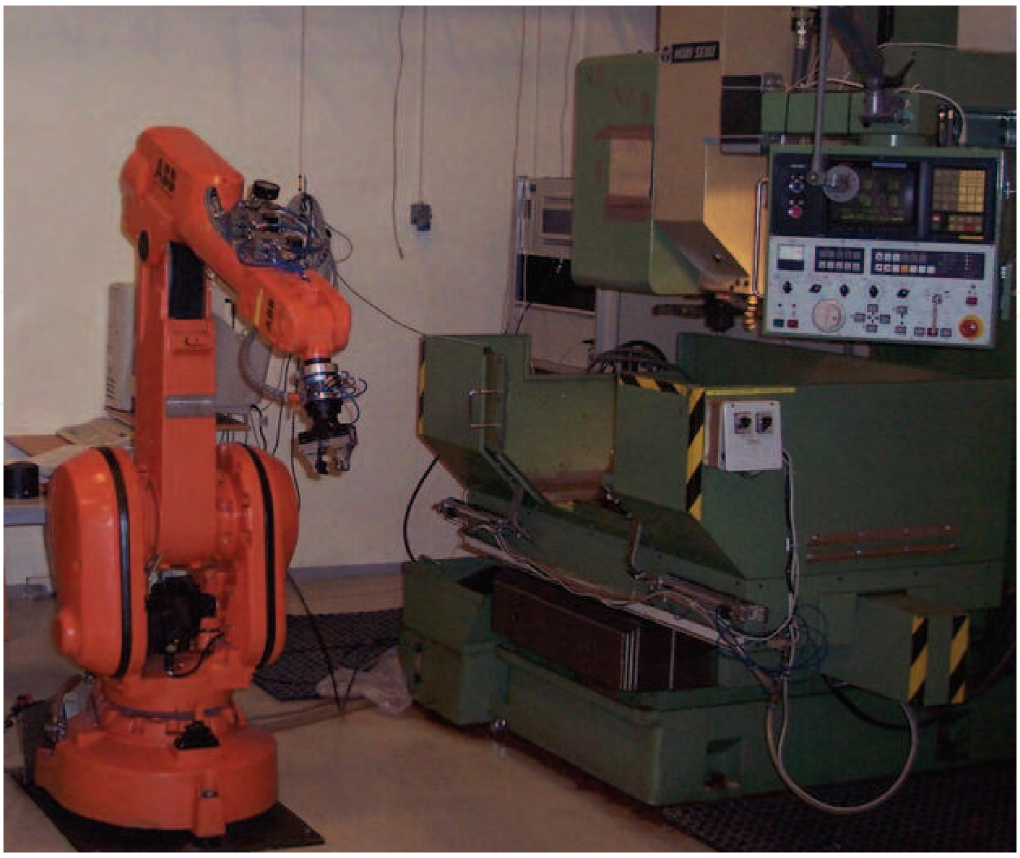


Fig. 7. Example setup of flexible manufacturing system, where the operator is replaced by a camera system and an industrial robot

4.1 Artificial neural network

Modeling functions and systems with neural networks (Gurney, 2003) is a developing science in computer technologies. This special field derives basis from biology, especially from brain of humans. It is well known, that human brain is constructed from neurons and interconnections among neurons, which are called synapses. One neuron can be connected to thousands of other neurons, which results a complex system and provides high level of parallelism. So thus, makes the human brain capable of recognition, perception of objects, human beings, animals, speech, etc. Also this is the explanation for the damage recovery capabilities of the brain.

On the other hand the computers used in everyday life are working in serial mode, which means that only one instruction is processed at a time. Measuring the reaction time of a human brain would result millisecond response, and would result nanoseconds in case of an average computer. Despite of this speed difference, until now, computer technology has not reached the level of the understanding of human brain. This inspired researchers to use the speed of modern computers and the parallel model of the human brain to create artificial intelligence. The combination of the speed and parallel computation model resulted mega leap in solving complex problems, which were previously believed to be impossible to solve.

Power of the neural networks is the fact, that they are nonlinear, which make them capable of solving nonlinear problems, while observing the limits in dimensions. It is easy to apply to any kind of problem and there is no need to know the exact data representation, unlike traditional statistical nonlinear methods. A neural network learns by examples, the more provided the better network is created, which is the same as how children learn.

4.2 Multi-layer perceptron neural network model

Nonlinear functions, such as Optical Character Recognition (OCR) (Bhagat, 2005), cannot be learned by single-layer neural networks. Hence use of multi-layer neural network is a must.

A multi-layer neural network is constructed from one input layer, one or more hidden layers and one output layer. In Fig. 8 this structure is introduced. Each layer has a predefined amount of artificial neurons (perceptons). The Multi-Layer Perceptron Neural Network (Haykin, 1998) is a feed-forward network, where the layers are in distinct topology. Every artificial neuron is connected to each of artificial neuron in preceding layer.

The artificial neuron receives number of inputs. Input can be either original data (in case of input layer) or from the output of other neurons in the network. Each connection has strength, called weight, which corresponds to synaptic efficiency of biological neuron. Every artificial neuron also has a threshold value, which is subtracted from the sum of incoming weights. This forms the activation of neuron. Activation signal is passed through the activation (transfer) function, called sigmoid function. The result of the activation function is the output of the artificial neuron.

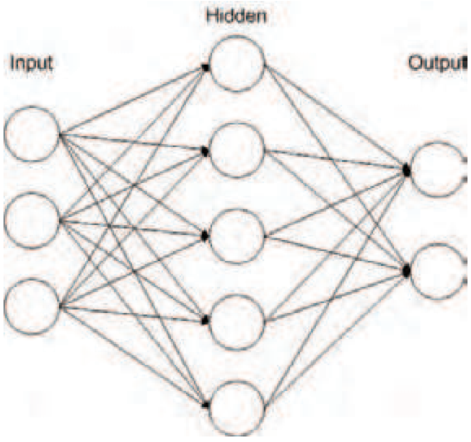


Fig. 8. Example of Multi-Layer Perceptron Neural Network

4.3 Structure

The vision system structure can be observed in Fig. 9. The image acquired by a digital camera is transferred to a PC, where the OCR is executed. The result of the OCR is the information shown on the operator screen. Based on this, different tasks can be carried out: instruct industrial robot to push a button, change I/O levels, stop NC machine, program upload, etc.

The system is constructed from modular parts. Every component can be replaced to machine (where is the needed information on the screen) and camera (connection type) specific module. In the setup phase also the neural network (layer and neuron number, weights and learning rate) can be customized.

4.4 Realization

The Multi-Layer Perceptron Neural Network is trained with back propagation learning algorithm (Rojas & Feldman, 1996) and for the activation function bipolar sigmoid function is used (1):

$$\begin{aligned} f(x) &= \frac{2}{1+e^{-\alpha * x}} - 1 \\ f(x)' &= \frac{2 * \alpha * e^{-\alpha * x}}{(1+e^{-\alpha * x})^2} = \alpha * \frac{1-f(x)^2}{2} \end{aligned} \tag{1}$$

where α is the parameter that decides the gradient of the function and x is the sum of the outputs of the previous layer.

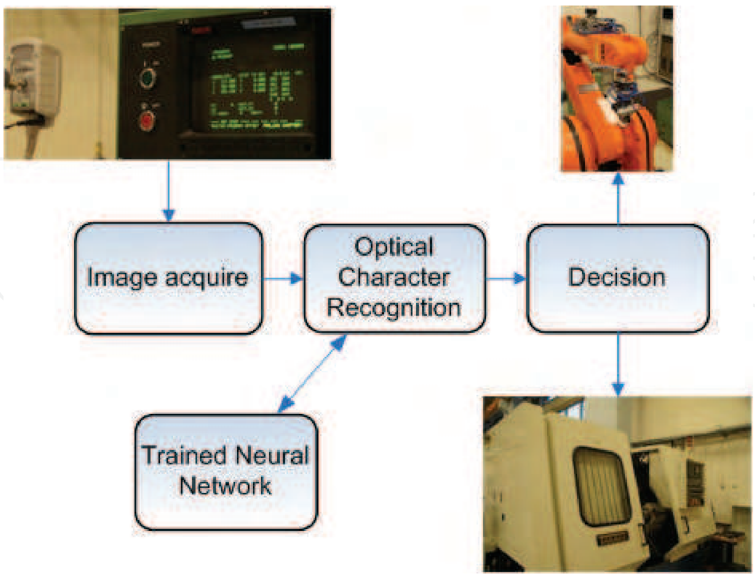
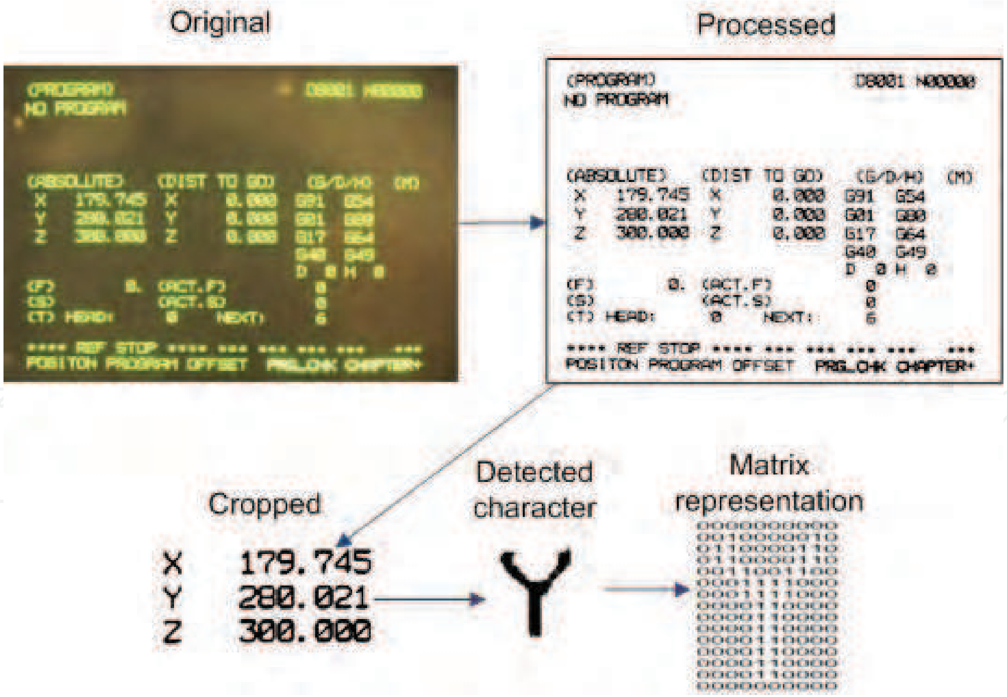


Fig. 9. Vision system structure

The training uses two inputs: the well formatted input image, captured by the digital camera, and the desired output. Both inputs are represented as binary values (0 or 1) and the process is shown in Fig. 10.



```
(PROGRAM)                              O8001 N00000
NO PROGRAM


(ABSOLUTE)    (DIST TO GO)     (G/D/H)    (M)
 X   179.745   X       0.000   G91   G54
 Y   280.021   Y       0.000   G01   G80
 Z   300.000   Z       0.000   G17   G64
                               G40   G49
                               D  0 H  0
(F)          0.  (ACT.F)          0
(S)              (ACT.S)          0
(T) HEAD:        0    NEXT:       6

**** REF STOP **** *** *** *** ***   ***
POSITON PROGRAM OFFSET  PRG_CHK CHAPTER+
```

```
X   179.745
Y   280.021
Z   300.000
```

```
00000000000
00100000010
01100000110
01100000110
00110001100
00011111000
00011111000
00001100000
00001100000
00001100000
00001100000
00001100000
00001100000
00001100000
00000000000
```

Fig. 10. Processing of input image

In order to get a well formatted image the following steps are executed:

1. Screen colour extraction (usually the old NC machines use one colour), the goal is to achieve higher contrast
2. Threshold filtering

With these steps an image will only contain black and white pixels. The black pixels are representing the binary value 1 and the white pixels represent 0. When only a specific region (Region of Interest) is needed, the image is cropped after this stage to specific dimension.

1. Characters are detected
2. Characters are converted to matrix representation (binary values of pixels)

These matrixes form the first input of the training.

The second input is the desired output: the 16bit Unicode representation of the detectable character (e.g. X = 0000000001011000).

The training is using character sets, which are machine specific (images and binary value of the characters).

4.5 Results

The neural networks weakest point is the training. Experiments were carried out to define the layer number, neuron numbers in each layer, best learning rate and sigmoid function alpha value. The goal with the experiments was to achieve 100% character detection in a 30 minute video stream of the operator screen. The character is stored as a 10*15 matrix, which results 150 neurons in the input layer and the output layer is composed of 16, because of the Unicode characters. Table 1. concludes the experimental results.

Parameter name	Parameter value
Layer number	3
Neurons in 1. layer	150
Neurons in 2. layer	300
Neurons in 3. layer	16
Maximum learning iteration	300
Maximum average error	0.0002
Initializing weight	30
Learning rate	120
Sigmoid function alpha value	0.014

Table 1. Results of experimental trials for neural network

5. Conclusion

In this chapter an image processing framework (Distributed Image Analyzer) and its integration into RT-Middleware as RT-Components is introduced. The integration becomes possible by introducing a simple conversion between interfaces. By this interface, image processing modules can easily be loaded to RT-Middleware and provides an image processing and vision toolbox for building complex robot systems.

If a simple image processing system is designed and implemented using the Distributed Image Analyzer toolbox in RT-Middleware, its advantages become apparent. Until now, vision components in robots were highly integrated to robot systems, without the possibility of reuse and easy adjustment.

As a special kind of vision system, vision based observation of old NC machines is introduced. This supervisory system replaces an operator with a camera and an industrial robot.

In the future the proposed image processing framework will be implemented and tested on a humanoid robot.

6. References

Ando, N.; Suehiro, T.; Kitagaki, K. & Kotoku, T. (2005 a). RT-middleware: distributed component middleware for RT (robot technology), *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3933- 3938, ISBN 0-7803-8912-3, Aug. 2005.

Ando, N.; Suehiro, T.; Kitagaki, K.; Kotoku, T. & Yoon, W.-K. (2005 b). RT-Component Object Model in RT-Middleware: Distributed Component Middleware for RT (Robot Technology), *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2005)*, pp. 457-462, ISBN 0-7803-9355-4, Jun. 2005.

Bhagat, P. (2005). *Pattern Recognition in Industry*, Elsevier Science, ISBN 0-0804-4538-1, USA

Brogårdh T. (2007). Present and future robot control development—An industrial perspective. Annual Reviews in Control, Vol. 31, No. 1, 2007, pp. 69-79, ISSN 1367-5788

Brugali, D. & Reggiani, M. (2005). A Roadmap to crafting modular and interoperable robotic software systems, *Proceedings of IEEE ICRA 2005 Workshop on Software Development and Integration in Robotics*, pp. 1-4, Apr. 2005, IEEE Robotics and Automation Society, Barcelona

Elovic, P. (2003). Implementation of Gate and Crane OCR Systems for Container Terminal Automation and Security, *Proceedings of Terminal Operations Conference ASIA (TOC ASIA 2003)*, pp. 1-7, Feb. 2003.

Gill, C.D. & Smart, W.D. (2002). Middleware for Robots?, *Proceedings of AAAI Spring Symposium on Intelligent Distributed and Embedded Systems*, pp. 1-5, 2002.

Gurney, K. (2003). *An Introduction to Neural Networks*, CRC Press, ISBN 1-8572-8503-4, United Kingdom

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, ISBN 0-1390-8385-5, USA

Ikezoe, A.; Nakamoto, H. & Nagase, M. (2006). Development of RT-Middleware for Image Recognition Module, *Proceedings of SICE-ICASE International Joint Conference, pp. 2036-2041*, ISBN 89-950038-5-5, Oct. 2006.

International Robot Fair (IRF) Organizing Office (2004). *World Robot Declaration*, press release, 25. Feb. 2004.

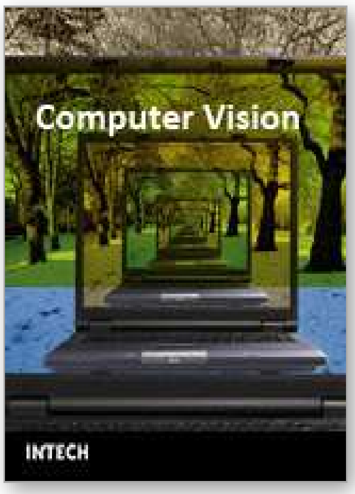
Kotoku, T. & Mizukawa, M. (2006). Robot Middleware and its Standardization in OMG - Report on OMG Technical Meetings in St. Louis and Boston, *Proceedings of SICE-ICASE International Joint Conference*, pp. 2028-2031, ISBN 89-950038-5-5, Oct. 2006.

Lee, J. H. & Hashimoto, H. (2002). Intelligent Space—concept and contents. *Advanced Robotics*, Vol. 16, No. 3, 2002, pp 265-280, ISSN 0169-1864

Object Management Group (OMG) Robotics DTF (2007). *Robotic Technology Component Specification [Online]*, http://www.omg.org/docs/ptc/07-08-18.pdf, 1.0 beta 2, Aug. 2007.

OpenRTM-aist project team (2007). *OpenRTM-aist [Online]*, http://www.openrtm.org/, 2007.

Rojas, R. & Feldman, J. (1996). *Neural Networks: A Systematic Introduction*, Springer, ISBN 3-5406-0505-3, Germany



Computer Vision
Edited by Xiong Zhihui

ISBN 978-953-7619-21-3
Hard cover, 538 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

This book presents research trends on computer vision, especially on application of robotics, and on advanced approachs for computer vision (such as omnidirectional vision). Among them, research on RFID technology integrating stereo vision to localize an indoor mobile robot is included in this book. Besides, this book includes many research on omnidirectional vision, and the combination of omnidirectional vision with robotics. This book features representative work on the computer vision, and it puts more focus on robotics vision and omnidirectioal vision. The intended audience is anyone who wishes to become familiar with the latest research work on computer vision, especially its applications on robots. The contents of this book allow the reader to know more technical aspects and applications of computer vision. Researchers and instructors will benefit from this book.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821