Cognitive Infocommunication in Automated Manufacturing Systems

Gábor Sörös

Supervisor:
Péter Baranyi, D.Sc.

Submitted in total fulfilment of the requirements for the degree
Bachelor of Science
in
Electrical and Computer Engineering

December 12, 2008

Department of Telecommunications and Media Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
to my grandfathers
Hallgatói nyilatkozat

Ahulírott Sörös Gábor, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója kijelentem, hogy a szakdolgozatot meg nem engedett segítség nélkül saját magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azoncés értelmen, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltém.

Tudomásul veszem, hogy az elkészült szakdolgozatban található eredményeket a Budapesti Műszaki és Gazdaságtudományi Egyetem, a feladatot kiíró egyetemi intézmény saját céljaira felhasználhatja.

Budapest, 2008. december 12.

__________________________
Sörös Gábor
Contents

1 Introduction 3
   1.1 Motivation ........................................... 3
   1.2 Goals ............................................... 4
   1.3 Structure of the report .............................. 4

2 Preliminaries 5
   2.1 Cognitive science .................................... 5
   2.2 Cognitive infocommunication .......................... 6
   2.3 Automated manufacturing ............................. 10
   2.4 Stereoscopic visualization ............................ 11
      2.4.1 Theoretical principles ............................. 11
      2.4.2 Visualization Center ............................... 13
   2.5 ShapeWrap motion capture system ..................... 13
   2.6 Ogre 3D ............................................. 14
   2.7 Qt framework ........................................ 14
   2.8 Firewalls and NATs ................................... 15

3 Achievements 16
   3.1 Cognitive robot supervision ........................ 16
      3.1.1 General description ............................... 17
      3.1.2 The manipulator .................................... 18
      3.1.3 The motion-tracking device ....................... 18
      3.1.4 The visualization subsystem ...................... 19
      3.1.5 The central unit .................................. 21
      3.1.6 Establishing connections ......................... 21
   3.2 Maestro .............................................. 24
      3.2.1 Design ............................................ 24
      3.2.2 Implementation ................................... 27
   3.3 Overview ............................................. 29

4 Test and evaluation 30
   4.1 Component tests ....................................... 30
   4.2 System test .......................................... 33

5 Summary 35
   5.1 Conclusion ........................................... 35
   5.2 Future potentials ..................................... 36
List of Figures

1  Cognitive science: connection between research areas (figure
inspired by [3], p.12) .............................. 5
2  Main fields of Information and Communication Technology 6
3  Communication channels [4] ............................. 7
4  Homunculus models ................................. 8
5  Non-conventional channels of communication ............. 9
7  Motoman SSF2000 ..................................... 11
8  Parallax effect ........................................ 12
9  Different stereoscopic glasses ............................ 13
10 ShapeWrap components [19] ............................. 14
11 The concept of cognitive robot supervision ................. 16
12 System schematic ...................................... 17
13 Projection planes ..................................... 20
14 UDP hole punching, figure inspired by [17] ................ 23
15 Modules of Maestro .................................... 24
16 The difference between TCP and UDP ..................... 25
17 Jitter: delay uncertainty ................................ 26
18 Rendering with OGRE in a Qt application ................. 28
19 System schematic with detailed components ............ 29
20 Test locations: Budapest, Lille, Narvik ................. 30
21 Maestro in the 3D room ............................... 31
22 Structure of a datagram sent by ShapeRecorder .......... 33
23 Setup screen on Windows .............................. 34
24 The system components in operation .................... 34
Kivonat

Jelen szakdolgozat célja a kognitív infokommunikáció kutatási eredményeinek alapuló új ember-gép kapcsolatok alkalmazási lehetőségeinek bemutatása robotok tanítására és felügyeletére. A robotra úgy tekinthetünk, mint egy képzetlen, de kitartó munkára, aki precíz műveletek elvégzésére képes, különleges intelligenciával rendelkezik, de bizonyos érzékei felettenek, és ezért speciális felügyeletet igényel. Ha megtanuljuk, hogyan kommunikáljunk ezzel az "új munkással", egy különleges képességű új "kollégát" kaphatunk. A hosszú távú cél az, hogy a főnök úgy adhassa ki a napi teendőket egy robotnak, ahogy azt embereivel teszi, például CAD dokumentációkkal, mozdulatokkal és szóbeli magyarázattal.


A dolgozat egy rövid bevezetést ad a kognitív infokommunikáció robotizált gyártórendszerekben betöltött szerepére és lehetőségeiről, megvizsgálja egy elosztott kognitív robotfelügyeleti rendszer tervezési kérdéseit, bemutatja a kísérleti elfedezést és értékeli a megvalósított rendszerrel elérhető eredményeket.

Különszavak: brain-in-the-loop gyártástechnológia, motion-capture, kognitív robotfelügyelet, sztereoszkopikus megjelenítés
Abstract

The main goal of this B.Sc. thesis project is to apply new cognitive info-communication channels in human-machine interaction to develop a new paradigm of robot teaching and supervision. The robot is considered as an unskilled worker who is strong and capable for precise manufacturing. It has a special kind of intelligence but it is handicapped in some senses that are the reasons why it needs special supervisory treatment. We have to command it clearly in a special way and we have to supervise its work. If we can learn how to communicate to this "new worker" we can get a new capable "colleague". The long term goal of the research is that the boss would be able to give the daily task to a robot in a similar way as he/she gives the jobs to the human workers, for example using CAD documentations, gestures and some verbal explanation.

This thesis presents my industrial robot supervisory application inspired by research results of cognitive infocommunication. The operator can steer the remote manipulator by certain gestures using a motion capture suit as input device. Every gesture has its own meaning, which corresponds to a specific movement of the robot. The manipulator interprets and executes the instructions invoking its on-board artificial intelligence, while feedback through a 3D visualization unit closes the supervisory loop. The system has been designed to be independent of the geographical distance between the user and the manipulated environment, allowing to establish control loops spanning through countries and continents. Successful results have been achieved between Norway, France and Hungary.

The document gives a short introduction on the role of cognitive info-communication in robotized manufacturing, discusses the design questions of a distributed cognitive supervision system, describes the experimental setup and presents the experimental results achieved by the realized system.

Keywords: brain-in-the-loop manufacturing, motion-capture, stereoscopic visualization, cognitive robot supervision
1 Introduction

1.1 Motivation

The development of information and communication technology (ICT) tends toward the direction that a number of computer systems (and robots) will be embedded into our environment. They appear in our everyday life and especially in manufacturing. These environments will impose need for new types of human-computer interaction (HCI) with interfaces that are natural and easy to use. In particular, the demand to interact with computerized environment without the need for any special external equipment is increasing, although the recent experimental interfaces are not efficient enough. The main problem is that it takes approximately 400 times longer to program a robot than to execute the actual task. Off-line programming is difficult, and the robot cannot produce anything during on-line programming. So, there is a continuous demand for new and effective robot teaching methods.

Today, the keyboard, the mouse and the remote control are used as the main interfaces for transferring information and commands to computerized equipment. In some applications involving three-dimensional information transfer, such as product visualization, computer games and control of robots, other interfaces based on trackballs and joysticks are used. In our daily life, however, we humans use our vision, hearings and gestures as main channels of communication. It would sound desirable to use devices able to communicate with humans in the similar way. This can be achieved by understanding visual and auditory input and producing similar output as well. These processes are key research areas of an emerging discipline, the cognitive science.

Interfaces based on speech have already started to evolve in a number of commercial and technical applications. For example, several systems are available, in which speech commands can be used for dialing numbers or making ticket reservations. Concerning visual input, the processing power of computers has reached a point where real-time processing of visual information is possible with common workstations, so in the near future the use of such systems is expectable. There are hundreds of research projects on face and gesture recognition with more or less success.

Using datagloves and motion-capture suits is an other enticing option. Though the user has to wear these, they provide six degrees of freedom, they can produce position data with same accuracy as vision-based systems [1], and, furthermore, they are also operable in camera shadow. Having these advantages they are suitable for secure and precise robotic applications, such as surgery or brain-in-the-loop manufacturing (see Section 3.1), in which human intelligence plays essential role in the control.
1.2 Goals

The goal of this thesis project is to design and implement an industrial robot supervision system inspired by research results of cognitive infocommunication. The system has to support the user to operate the robots with high-level human instructions and visualize the manipulated environment being independent from the geographical distance between the user and the manipulator. In case of long distances point-by-point trajectory control is not applicable because of communication unreliability, so a remote manipulator equipped with some on-board intelligence is required. Though the manipulator is able to perform actions autonomously, the human brain remains the key part of the loop by making decisions and giving appropriate instructions to steer the remote intelligence. In this respect the specific objectives are:

- Constructing a geographically distributed brain-in-the-loop robot supervision system
- Preferring arc welding and painting purposes in a static and well-known environment
- Evading point-by-point trajectory control supposing some on-board intelligence on the remote side
- Human-machine interaction with usage of high level human instructions (e.g. with certain gestures)
- Designing feedback with 3D visualization
- Testing in a real long-distance application

This architecture is applicable to bring spaces with different scalings together, allowing to work also in situations, in which the operator is not able to be present physically due to geographical distances, dimensional limitations or dangerous environments. This way we can combine the strength, the speed and the accuracy of machines with the perfect vision, recognition and decision-making competencies of the human brain.

1.3 Structure of the report

The structure of this document is as follows: Section 2 deals with the background of the R&D work and introduces the applied components. Section 3 presents the implemented system in details, while Section 4 is devoted to component and system evaluations. Section 5 is a summary with respect to the goals, and shows potential applications of the proposed system within modern manufacturing.
2 Preliminaries

2.1 Cognitive science

What is cognitive science? Let’s look at the French translation of this phrase: *les sciences cognitives*. This plural form shows that cognitive science is not a stand-alone discipline but a connection between several fields. There are psychology, philosophy, biology, neuroscience, mathematics, theoretical computer science and others meeting together as seen in Figure 1. Cognitive science is dealing with mind and intelligence in the first place, and in a broader perspective with all aspects of human and non-human (be it animal or artificial) life influenced by "what we know" and "how we know". Its aim is to explore and understand mental processes such as thinking, learning, perceiving, imagining, acting and planning. Despite its young age (the idea of this new study was born in the mid-1950s, and the process of its institutionalization started at the end of the 1970s), it could be regarded as a paradigm-forming field among the broad area of all human-related sciences, with a great impact on many spheres of life... [2]

It is neither a "super discipline" nor a "universal model", which could be used for every scientific problem, but it is connecting several independently grown studies, showing the common fields in them. Engineering related people tend to reserve reflections on technical fields: cognitive computer science and cybernetics. For further reading about cognitive science see [3].

![Diagram of Cognitive Science Connections](image)

Figure 1: Cognitive science: connection between research areas (figure inspired by [3], p.12)
2.2 Cognitive infocommunication

Many aspects of the way in which human beings do complex tasks are not well understood, and the models of underlying cognitive and neurological mechanisms are still being researched. Communication theory and cybernetics are research areas born together with cognitive science having a strong inspiration on each other. Each one of them tries to describe the world and living organisms with similar models, on the one hand as a communication process transferring and processing pieces of information, on the other hand as a sophisticated control loop of the body. According to these theories, the world consists of signals and systems.

![Figure 2: Main fields of Information and Communication Technology](image)

Information and communication technology can be divided into three main sections as shown in Figure 2. The media field stands for creating and storing information content like videos, music and databases. Communication deals with transferring information from one place to another with high efficiency, reliability and security. The third part, informatics, represents all kinds of modern tools of information processing. Overlapping areas are getting more and more into focus while the main ones are converging towards each other. For instance, infocommunication is the study of information transfer between humans and processing devices, or between the elements of the computerized environment.

The triangle can be considered from the viewpoint of cognitive science. Cognitive informatics tries to give models of information processing mechanisms of the brain involved in perception and cognition. Meanwhile, cognitive communication theory deals with possible transfer channels and expression methods in the communication situations. Cognitive media could be regarded as media content creation to exercise influence on mind, e.g creating political advertisements. According to these, the definition of cognitive infocommunication is the following:
Cognitive infocommunication is the process of transferring information between cognitive processes of mind and computerized equipment using cognitive informatic models.

This area deals with the questions of transferring complex high-level information, using a set of communication channels and perceptual modalities. The assumptions of cognitive science suggest that the key parts of the whole communication process are perceptions and interpretations that the communicant brings to a given interactive situation, e.g. how a person speaks or acts, and how these circumstances affect the process. The question is, which parts does the process consist of?

![Communication channels](image)

Figure 3: Communication channels [4]

**Elements of communication** Human communication uses verbal, para-verbal and non-verbal channels. Verbal communication is the speak itself, para-verbal is the way of speaking (pitch, tone, etc.) and every other accompanying element, such as gestures and mimic, are called non-verbal elements. According to statistics, 55% of the information transfer is succeeded within the non-verbal channel beside 38% para-verbal and 7% verbal usage, as also seen in Figure 3. This means the largest amount of information is transferred via light and perceived through the eyes.

Researchers say that hearing is the latest sense of organisms that appeared during the evolution. Speaking is said to be evolved only hundred thousand years ago [5]. One may then suppose that the brain is specialized in the processing of visual information, so the visual channel should be preferred in a communication interface between humans and computers. Also, it is desirable to develop machines able to use non- and para-verbal channels during interaction with operators, or if future technologies will make it possible, with other intelligent devices.
**Perceptiveness of organs**  Concept of a homunculus (little man, /ˈlɑːtɪn/) is often used to illustrate the relative space human body parts occupy on the somatosensory cortex and the motor cortex in the brain, as shown in Figure 4(a). Lips, hands, feet and sex organs have more sensory neurons than other parts of the body, so the homunculus has correspondingly distorted areas for processing input from lips, hands, feet, and genitals. Figure 4(b) is also commonly called 'the little man inside the brain'.

![Homunculus models](image)

(a) Sensory homunculus and motor homunculus, original picture taken from [6]
(b) Model of the human body, picture from [7]

Figure 4: Homunculus models

An important thing has to be mentioned here: the world humans perceive through their senses is different from what other species do. Every organ is tuned to the stimulus domain that is the most important for survival. Different species have different needs and diverse circumstances, therefore the organs are sensitive to other stimuli. This must be kept in mind when intelligent spaces expand also on animal-related applications.

The following table shows a comparison of absolute thresholds of response of different human senses, according to an experiment from 1962 [8]. The outliers of the vision is obvious.

<table>
<thead>
<tr>
<th>Sense</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vision</td>
<td>Candle light from 50 km in a dark and clean night</td>
</tr>
<tr>
<td>Hearing</td>
<td>Tick-tack of a wristwatch from 6 m in silence</td>
</tr>
<tr>
<td>Taste</td>
<td>One teaspoon sugar in 9 l water</td>
</tr>
<tr>
<td>Smell</td>
<td>One drop perfume in an air volume of 6 rooms</td>
</tr>
<tr>
<td>Touch</td>
<td>A wing of a fly on the face fallen down from 1 cm</td>
</tr>
</tbody>
</table>
Non-conventional channels  An exciting research area in cognitive info-
communication is opening non-conventional channels (see Figure 5) between
participants: using our ears to see, using our fingers to hear . . .

A research team in MTA-SZTAKI\(^1\) has specialized in the topic of mixing
senses, but this area is not closely connected with the proposed system, so
these details will not be discussed here. The third component of the cognitive
ICT triangle, cognitive media could also be conceived as content management
for these non-conventional channels, e.g. creating scented movies.

![Diagram of non-conventional channels of communication]

Figure 5: Non-conventional channels of communication

High-level information transfer  Another important research topic of
cognitive infocommunication is finding ways for transferring high-level instruc-
tions to computers and other information processing devices, such as
robot controllers. In case of manufacturing systems, the aim is to simplify
the teaching of robots needed before starting the production because point-
by-point description of the fabrication process is too complicated. Since the
easiest way to describe robot movements is by the motion of human arms
and body, this project concentrates on gesture based infocommunication.

HCI studies  Human-computer interaction or HCI is the study of inter-
faces between people and computers, and as such, it is in strong connection
with cognitive infocommunication. Its long-term goal is to design systems
that minimize the barrier between the human’s approach and the computer’s
understanding of the task. A comprehensive book about HCI is listed in
the bibliography under [9]. Another article turning attention to software-
ergonomics can be found under [10]. The author’s advisements discussed
in the paper have been kept in mind during development of the software
components.

\(^1\)Computer and Automation Research Institute of the Hungarian Academy of Sciences,
Budapest, Hungary
2.3 Automated manufacturing

Automated systems for home, educational or industrial use are well-suited for dangerous or massively repetitive tasks. Thousands of industrial and laboratory robots are in operation these days. The material handling systems are able to move objects of several hundred kilograms with an accuracy of millimeters or even better. These robots are no other just computers equipped with integrated sensors and manipulators. Latter ones are in most cases welding devices (MIG-welding guns, spot-welders, etc.), spray guns, grinding and deburring devices (such as pneumatic disk or belt grinders, burrs, etc.) or grippers (devices that can grasp an object, usually electromechanical or pneumatic). Using machines for filling, sorting, painting, welding or other applications is very attractive due to their high precision, safeness and productivity. However, in case of often changing demands the point-by-point programming of controllers is inconvenient, so a more flexible teaching method for applications with small batch size is necessary.

As Figure 6 shows, the modern automation systems tend towards the intelligent environment concept, where various robots and other devices cooperate with each other autonomously. With these distributed intelligent production systems problem solving competencies will increase and manufacturing will become easier than ever.

![Diagram of automation systems and robots]

Figure 6: Future development of automation systems and robots [11]
**Industrial robots**  Industrial robots are the key parts of automated manufacturing systems because they form the factory workcells with their accessories and peripheral machines. An exemplary industrial robot, the Motoman\(^2\) SSF2000 arc welding robot is shown in Figure 7. The SSF2000 has got six axes of freedom and can operate with a payload of 6 kilograms with 0.08 mm precision. It can be mounted on the floor, wall or on the ceiling, according to the actual task, but it is mainly used for welding. The robot is equipped with some on-board intelligence thanks to the NX100 controller that is responsible for inverse kinematic computations and servo control. Due to NX100, this robot can perform movements between given endpoints autonomously.

![Motoman SSF2000](image)

**Figure 7: Motoman SSF2000**

### 2.4 Stereoscopic visualization

In most systems visual output is the main way to give information back to the human user. The classic way is to use two dimensional graphical user interface with labels, buttons, and other standard controls. In this setup the user can issue precise commands to the system, but the visual feedback is limited. This standard approach works efficiently on numerous systems. However, in some situations, such as working with multidimensional data or when the operator needs to manipulate in space, the control via the usual 2D graphical user interface becomes a complex task. With the rapid evolution of computer hardware, software visualization started exploring and successfully applying new 3D representations for various tasks. For an overview of the current state of 3D visualization, see [13].

#### 2.4.1 Theoretical principles

Human vision is a complex and advanced system. One of its purposes is to deliver spatial information to assist the control of our actions. In nature the extent of depth perception is variable among different species. Humans have their eyes positioned on the front of their heads, thereby reducing field of view

\(^2\)Motoman is the robotic division of Yaskawa Electric Corporation
in favor of precise depth perception. The objects which are visible with both eyes have different images, which enables the brain to calculate distances. The difference of the same object in the same environment viewed along two different lines of sight is called parallax. The human nervous system is responsible for mapping the parallax into spatial information.

Figure 8: Parallax effect

The aim of stereoscopic systems is to create a real 3D sense of space in a limited environment. This task can be decomposed into two parts. First, render the two pictures separately with the proper settings. This is supported by decent hardware, which uses four buffers instead of the usual two and renders into the dedicated buffers continuously. Second, the images have to reach the eyes separately using some form of physical separation: the images must only be seen by the intended eye.

There are several different solutions, which all utilize one of the physical properties of light to achieve this separation.

**Anaglyph stereo** uses the spectral property of light to separate the images: red and blue color filters are used in front of the left and right eyes so that red-colored and blue-colored images are separated.

**Passive stereo** uses a pair of glasses containing two oppositely polarized filters. The light, which is usually polarized by using a polarizing filter in front of the projector lens, can only pass through its corresponding filter.

**Active stereo** uses time domain separation. The left and the right images are alternating repeatedly at a high frequency, and an active stereo shutter glass synchronized to the projection is responsible to separate the images by changing the glasses’ opacity as required.

**Binocular Head Mounted Displays** can display different image to each eye. In this case the separation is solved by the placement of the displays.
Using stereoscopic visualization allows high levels of immersion. The potential benefits includes better spatial understanding and decrease in information clutter. See [14] for a detailed discussion.

2.4.2 Visualization Center

The Visualization Center (hereafter VC) at the Faculty of Science, Eötvös University, Budapest, Hungary was founded in 2006. One of the major projects undertaken was a construction of a large stereoscopic projection system with the primary aim of providing a platform for scientific visualization. For a detailed information on the Visualization Center, see [12][16]

2.5 ShapeWrap motion capture system

Measurand’s ShapeWrap III is a portable motion capture system for full-body motion capture purposes. It also includes two ShapeHands that capture hand, finger and shoulder movements (see Figure 10). Measurand’s system is widely used in the entertainment industry.

ShapeWrap comes with a software called ShapeRecorder which can be used to view, synchronize or store motions, and also can simulate presence of the suit from pre-recorded files acting as a server and forwarding the actual coordinates periodically.
2.6 Ogre 3D

OGRE3D (Object-Oriented Graphics Rendering Engine) is a scene-oriented 3D engine written in C++ and made very flexible to be used in as many application fields as possible. Its class library abstracts both OpenGL’s and Direct3D’s libraries, which are the most commonly used Application Programming Interfaces (API) for rendering 2D or 3D scenes. OGRE is open-source and has an extensive developer community. For details see [21].

2.7 Qt framework

Qt [cute] is a very powerful cross-platform object-oriented application framework, which enables programmers to develop GUI (Graphic User Interface) or non-GUI applications in C++ or Java. Its class library is very intuitive and makes the creation of complex programs much easier than using native C++ or standard template library. Qt applications are portable between several platforms including Windows, Linux and Mac OS X. For more information see [22].
2.8 Firewalls and NATs

Firewalls are network devices or softwares that handle access control between two networks. Firewalls enable only pre-configured traffic channels and traffic types while blocking any other data flow inhering in prohibited services or intrusions. However, communication between computers behind firewalls can be established by tricky techniques mentioned later in this document.

Network Address Translators (NATs) are routing devices in computer networking that map a given address space into another. This is done by modifying the address information of the datagram packet headers, according to configurable translation tables. The routing device rewrites the outgoing Internet Protocol (IP) packets on exit so that they appear to originate from the router. In the reverse communications path, responses are mapped back to the originating IP address using the same rules. The mechanism has been introduced because the original IP address realms were close to depletion, and with address translation a group of computers can use the same global address. It is widely used in home and office networking also for security and anonymity purposes. In numerous cases, Internet Service Providers (ISPs) also concentrate their network traffic by using NATs in the global address domain in order to reduce the number of global IP addresses used.

NAT causes well-known difficulties for peer-to-peer communications, since the peers involved may not be reachable at any globally valid IP address. The problem is crucial in VoIP applications, in which the data field of the IP packet contains address information as well, but the NAT device translates only the packet header causing inconsistency. Several NAT traversal techniques are known such as relaying, connection reversal, Universal Plug and Play (UPnP), Simple Traversal of UDP Through NAT devices (STUN). The present project uses a technique based on UDP hole punching [17]. The realization will be discussed later in Section 3.1.6. The drawback of these methods is that they need third party server resources or support on every routing device on the way between peers.
3 Achievements

As mentioned in Section 2.3, modern manufacturing systems tend towards the intelligent environment concept, in which various robots cooperate with each other. Several sensors, manipulators and processing units are installed in the space, providing distributed computations and performing actions together. A fundamental assumption of the present project is that an intelligent environment already exists, and manipulators are ready to execute commands.

3.1 Cognitive robot supervision

A classical control loop consists of a controller, a controlled process and a feedback from measured variables. Improved control loops also contain an Artificial Intelligence (AI) component, which automatically modifies the controller’s options according to the desired operation. However, for certain processes with high complexity the AI unit is not efficient enough. Production systems are designed to displace the expensive human resources, however, above a certain grade of intelligence the prices increase exponentially. There are several applications in which the optimal solution is embedding human intelligence into the control loop to supervise the process (see Figure 11). The human operator’s assignment is to give appropriate instructions to the (remote) manufacturing system, which independently performs the ordered tasks using its own artificial intelligence.

![Figure 11: The concept of cognitive robot supervision](image)

In fact the meaning of telemanipulation is quite close to this concept, except that there is no AI unit in the traditional telemanipulation chain. Now the question is how to design an efficient and easy-to-use communication interface between the human brain and the AI unit. To answer this question, research results of cognitive infocommunication can be adopted. The next section presents a gesture-based cognitive robot supervision system that achieves the goals precomposed in Section 1.2.
3.1.1 General description

The traditional interfaces between operator and the controller’s AI unit are based on joysticks or trackballs. However, it is desirable to replace these and create something that fits the natural communication situations better. The proposed system is based on a motion-capture suit (see Section 3.1.3), which tracks movements of the operator’s hand allowing that he/she can command the system with certain gestures. Coordinates of the body are transferred to the central processing unit that evaluates the gestures and also manages a virtual world model. The user gets feedback from his operations in the world model through a 3D visualization subsystem (see Section 3.1.4). The central unit transmits instructions to the AI of the robot (see Section 3.1.2), which runs the path-planning and inverse kinematic computations and moves the robot to the desired position. The robot is supposed to interact within a world of well-known static objects.

![System schematic](image)

Figure 12: System schematic

The main system components as shown in Figure 12:

- A manipulator with on-board intelligence
- A motion-capture input device
- A stereoscopic visualization unit as feedback
- A central coordinator unit with world model

3D world model is applied in the feedback; it enables separating the user from the manipulator space, in order to establish control loops spanning through countries and continents. Using point-by-point operation is futile in these cases because of delay, jitter and uncertainty problems. Hence an intelligent manipulator environment is supposed, which understands high-level instructions and performs actions autonomously. The command transfer between user and the central unit is enhanced by non-verbal channels by
virtue of the 3D projection and motion-capture. The thesis project focuses on tracking the operator's hands beside the full virtual body animation, and controlling the robot with hand movements sending hand position as path endpoints.

3.1.2 The manipulator

A manipulator is a device used under human control to manipulate objects without direct contact. The geometry of the manipulator space does not have to be identical with the geometry of the user space, and similarly the manipulator's structure can differ from the user's body. For example, humans have two hands with five fingers on each, but in the manipulator space there can be numerous double-fingered robot arms. The AI unit has to transform the movements between the geometries and translate the solution into a command sequence.

An additional assumption of the present project is that the robot space is well-known and static. So one could create its precise model and display it in the virtual space. Such case is a closed room with hazardous material containers on well-defined positions of the room. Then, user's manipulations in the virtual space can be easily mapped into the manipulator space. For security purposes, an extra web-camera is placed in the room.

In this project a Motoman SSF2000 arc welding robot (described in Section 2.3) plays the role of the manipulator unit. It is a six degrees of freedom (6DoF) industrial robot with its end-effector removed and replaced by a pen. The robot is equipped with some on-board intelligence embedded in the NX100 controller. It can find the optimal path between two given endpoints and set the pen in any desired position. This robot is designed for arc welding purposes, so its motion is smooth and fast enough to satisfy our needs. However, the speed of the path calculations should be improved.

There is a PC in the manipulator space as well, connected to the controller and to the internet. A small communication application called MotoProg has been installed on the PC that can send and receive messages from NX100. The application has been extended by a component to integrate it into the cognitive control system. The central unit can communicate with the modified MotoProg software through the internet, so these components can be separated far from each other.

3.1.3 The motion-tracking device

The ShapeWrap motion-capture device of Measurand Inc. (described in Section 2.5) has been chosen as the input device of the system. One suit with two ShapeHand gloves have been available at a partner institute Narvik University College, Norway. The suit contains several fiber optic based 3D bend and twist sensors called ShapeTapes. The sensors are connected to
a data concentrator unit on the body (see Figure 10), which transmits the coordinates to a PC via WiFi. The PC runs ShapeRecoder software of Measurand and is connected to the Internet with an open UDP port. The central software unit can connect to ShapeRecoder’s service and receive coordinates in datagrams periodically. A typical period is 14ms.

ShapeRecoder can use and convert body coordinates between several formats including global position and orientation (GPO, world coordinate system), biovision hierarchical format (BVH, human coordinate system), another one with quaternions, etc. The system uses the GPO format that contains data on the positions of all major joints including the wrist, elbow, shoulder, hip, knee and back, and orientation of the bones between these joints. All positional data is measured in mm and all orientation data is measured in roll, pitch, yaw orientation angles in degrees. The X-axis extends outwards in the direction that the person is facing during the homing calibration, the Y-axis extends upwards, and the Z-axis extends to the person’s right during homing. The order of rotation angles is yaw first about the +Y axis, followed by pitch about the rotated +Z axis, and then roll about the doubly rotated +X axis.

Although ShapeWrap can capture full-body motion, it is optional to use every part of the suit. For example gloves are not needed in every situation. In this case, just a long bending sensor is mounted on the middle finger.

3.1.4 The visualization subsystem

Ogre 3D (as described in Section 2.6) is class library that abstracts all the details of using the underlying system libraries: Direct3D and OpenGL. Direct3D uses a technique called Multithread to utilize devices with multiple buffers. OpenGL applies a different principle called Quad Buffering for achieving stereoscopic effects. Ogre3D aims to hide the inherent differences between rendering backends, therefore, until DirectX has the same support for stereo as OpenGL does, stereoscopic support will not be introduced. Also this is not a main priority for the community at the moment, however there exists a number of different patches to introduce some kind of stereo support.

OpenGL stereo support for GLX rendering subsystem has been insinuated in order to employ Quad Buffering on the Quatro graphic card at the Visualization Center, Eötvös University. In the render system, the selection of the visual mode has been modified to deal with the stereo requirements, which allows to use the stereo rendering if it is supported both by the hardware and the overlying driver. Also a new scene manager has been constructed to do the necessary additional work in the rendering loop.

Rendering a standard 3D scene requires the scene itself, as a set of graphical renderable objects, and a camera position with the corresponding projection plane. By duplicating the camera and positioning them in a way which imitates the human binocular vision, we can capture human-friendly
three-dimensional images. The eyes are approximately separated horizontally by 6 cm, however studies show that any interocular distance of greater than 3 cm used in the stereo projection model is adequate to provide a user with maximal performance in the depth perception task[15]. This means that there is no real need to tailor the focal distance used in rendering due to individual differences.

The other part of stereo setup is the alter the projection planes. The naive approach is the following: displace the two cameras and set their focal point to the target. This is a fairly easy solution, but it introduces eye strain because of the vertical parallax caused by the symmetric aperture (see Figure 13(a)). The solution is to use asymmetric frustums (Figure 13(b)).

![Symmetric frustum](image1.png) ![Asymmetric frustum](image2.png)

(a) Symmetric frustum (b) Asymmetric frustum

Figure 13: Projection planes

In OpenGL, this can be accomplished by setting up different asymmetric frustum for each camera. The scene manager hides the two internal cameras from the user, only the interocular distance is a visible parameter. The off axis cameras are calculated from the original camera’s frustum and up vector and the interocular distance. This way a quasi-transparent stereo scene manager is constructed using the decorator pattern. The actual difference is that the decorator first calculates the matching camera then calls the render function of the decorated scene manager for each eye respectively; and then, lastly, it writes the results back to the corresponding buffers.

In the Visualization Center (see Section 2.4.2), there is a projection room with active stereo projectors. Active stereo projection provides an adequate image quality, and particularly the one screen setup, is a cost-effective yet suitable solution for the purposes of this system. The statements above and the opportunity to access the facility were considered for selecting the

20
stereoscopic environment used in the project. The projection canvas splits
the space into two parts, isolating the audience from the projectors. In
order to create a highly immersive impression, the projected 3D image has
to be the natural extension of the split space. This means that the projected
floor has to be the continuation of the original floor and also the size of
the rendered objects has to match the size of their real counterparts. To
fulfill this requirement, the location of the observer has to be taken into
consideration. In the current implementation the position of the observer
can only be adjusted by hand. Fortunately, there is an ongoing development
of precise ultrasonic localization in the stereo room, which makes it possible
to automatize the personalization of the rendered image to the user.

3.1.5 The central unit

The central unit coordinates the whole system during operation. It is a
modular software framework for robot control developed in frames of this
thesis project. The software is called Maestro and its details are discussed
in Section 3.2.

3.1.6 Establishing connections

The system components are located far from each other, which makes the
development more difficult: issues in information transfer and synchroniza-
tion have to be solved. Visualization Center, Narvik University College and
the robot's location are behind different firewalls and network address trans-
lators (NATs), so the system's components doesn't have globally routable IP
addresses, and even they can be in different white address realms. It means
that direct connections can not be established without modifying the routers
of these institutes to forward datagrams, which is initially precluded because
of obvious security reasons. However, there are possible workarounds to this
problem listed in Section 2.8.

The issue has been omitted by creating a virtual private network with a
software called Hamachi. Virtual IP addresses can be used as if they were
global, since Hamachi takes care of networking in the background. It is based
on the UDP hole punching technique that was first mentioned in [18], and
is also used by Skype and many multi-player computer games. The concept
is the following:

Figure 14(a) Let's suggest that Alice wants to communicate with Bob,
but both are in private networks behind network address translators. There
must be a rendezvous-server located in the global address realm with a de-
dicated UDP port. Peers connect to the dedicated server first, as its address
is globally routable, and routers do not block any outgoing request from the
private LANs. The peers send their private IP address and UDP port number in the initial connect message. The server registers the peers’ private and public address information from the connect message (the public from the header and the private from the data content). With Skype’s analogy, this is the log-in session.

**Figure 14(b)** If user Alice wants to connect to user Bob, then she will send a message to the server "connection request to Bob". In reply to this, the server sends Alice both of Bob’s addresses. Parallel to that the server sends Alice’s addresses to Bob in a message "connection request from Alice". These packets will not be blocked either, because routers identify them as responses to the earlier outgoing requests. At this point, each computer knows the private and public address information of the other peer.

**Figure 14(c)** Now, Alice starts to flood UDP requests to both of Bob’s endpoints, and Bob does the same to Alice. Let’s suggest, that the request of Alice arrives first to Bob. As this packet is originated from the external network, Bob’s router will drop it. When Bob’s first request arrives to Alice’s router, it will be interpreted as a response to the outgoing traffic, so the router will forward the packet to Alice. At this point, Bob has "punched a hole" on Alice’s NAT device. After Bob’s first outgoing message other requests from Alice can also reach Bob’s computer, because they seem to be responses to the traffic generated by Bob. So Alice has also "punched a hole" on Bob’s NAT device.

**Figure 14(d)** During the operation the working address (private or external) of the other peer is stored locally. Both addresses are needed in the process, because initially it is unknown, whether or not the peers are behind NATs.

"The approach described above has some problems. For example, a client could try to contact a private address, which is already being used locally, causing unexpected results. If the node that was contacted unintentionally has an incompatible service or no service bound to the UDP port, then the connect messages will be dropped. In the unlikely event, a registered application chooses to interpret the message, the outcome is unpredictable." [18] To minimize the risk of such a failure, the components use ephemeral ports to send and receive in order to avoid collision with other applications. Other computers in the private network are supposed to run firewall software to block occurrence of requests.
Figure 14: UDP hole punching, figure inspired by [17]
3.2 Maestro

As Maestro is the key component of the system, it is dealt within this distinguished section.

3.2.1 Design

In fact Maestro has been designed to be a modular framework for virtual reality applications using Measureand’s ShapeWrap as input device and the robot control part is just one pick of the possible applications. The realized modules are shown in Figure 15. As there are several independent exercises such as rendering, gesture recognition and robot guidance, Maestro has been designed as a multithreading application. Every module has to be implemented as a separate thread with different CPU priorities. The main part is the rendering module which is also responsible for user interaction before the suit is connected. A virtual scene is assembled, in which the operator can perform his actions. The operator’s body is displayed and animated within the virtual space as a skeleton. Objects of the manipulator space could be displayed here but at the moment no obstacles are supposed within the operation area. A thread called ShapeConnect manages the data transfer from the motion-capture suit, while the RobotControl thread communicates with the MotoProg application installed at the remote manipulator space.

![Diagram of Maestro modules](image)

**Figure 15: Modules of Maestro**

**Networking considerations** The proposed distributed system relies on Internet connection between the components. However, the traffic types can be different between the components. Let’s take a look at the possibilities to provide Quality of Service.

TCP (Transmission Control Protocol) is the most commonly used transmission protocol of the Internet. The reason for this is delivery guarantee.
TCP offers error correction and flow control, that determines when data needs to be re-sent, and stops the flow of data until previous packets are successfully transferred. If a collision occurs, the client re-requests the packet from the server until the whole packet is complete and is identical to the original. TCP is well suited to data transfer that is quality and not speed oriented. The data transfer between Maestro and the robot is relatively rare but sensitive of delivery. That's why this protocol has to be used in the RobotControl module.

UDP (User Datagram Protocol) is also a transmission protocol over IP, commonly used for streaming audio and video or other speed oriented services. The reason UDP can be faster than TCP is that it does not deal with flow control or error correction (it contains a checksum to identify corrupted packets). The packets sent over UDP protocol may become lost, duplicated or reversed, but if no error occurs, transmission is very fast. UDP is used for data streaming between the suit and Maestro.

![Figure 16: The difference between TCP and UDP](image)

The ShapeConnect thread The ShapeConnect thread takes care of the whole communication between Maestro and the computer on which Shape-Recorder is running and the ShapeWrap suit has been installed. Shape-Recorder can forward the data received from the suit to an other computer in UDP packets. If a client wants to receive these data, it sends a request in a special format of UDP datagram, and after this coordinates will be flowing to the client at every frame (in about every 14 ms).

Since UDP is a connectionless, unreliable protocol, Maestro has to take care of misdelivered datagrams. RTP (Real-Time Transport Protocol) is a protocol developed for real-time data streaming (audio, video), where speed is the most important aspect. However, RTP is unavailable here because ShapeRecorder is not compatible with it (and its source-code is not open), so ShapeConnect needs to extend UDP by additional features to correct the mistakes, as if RTP were used (in fact RTP is built on UDP anyway). The thread extracts useful information from datagrams and fills out a DataSet object, which consist of all position and orientation coordinates of the suit, plus a timestamp. If a timestamp is older than the latest one, the tardy packet will be dropped. The synchronization of the user's motion with the animation in the virtual space is achieved by using these timestamps. Poin-
tens of DataSet objects are filled into a FIFO storage called DataBuffer. This compensates the jitter (see Figure 17) too keep the delay constant. The coordinates of the FIFO are used in the main part of Maestro to update position of every bone during 3D rendering and to recognize certain gestures. In Maestro, the user has to select which tapes are in use, because the program needs to know how much data and in which form is contained in a received datagram.

![Figure 17: Jitter: delay uncertainty](image)

**The RobotControl thread** This part deals with the communication between Maestro and the computer the AI unit is connected to. The MotoProg application abstracts a layer above the communication port of the computer, which enables the user to send commands to the robot very easily. RobotControl can establish a TCP connection with the other computer, and send it a signal every time that the person wearing ShapeWrap is making a particular movement. This signal is a TCP packet containing the position and orientation of the hands (left or right, depending on the option selected by the user of Maestro). Once the remote PC computer receives this packet, MotoProg transforms the message and forwards a command to the robot, ordering it to move the endpoint to the corresponding position, with the corresponding orientation. Thereafter it requests the robot’s coordinates and sends them back to Maestro to refresh the world model.

**Synchronizing threads** ShapeConnect fills a buffer every time it receives a datagram, but OGRE takes a data set only at rendering a frame, and the latter process can be slower depending on the scene to render and the GPU. This means that the buffer could overrun. The problem is avoided by setting the maximum length of the queue and use it as a FIFO (First-In-First-Out) storage. Secondly, if the rendering thread needs a data set, it takes the most recent one according to timestamps and clears other obsoleted ones.

As there is a Producer and a Consumer thread, preventing simultaneous reading and writing is required. This can be easily done by Qt’s built-in mutexes (objects that realize mutual exclusion). Every time a thread wants to use the data buffer for reading or writing it locks the mutex attached to the queue. After having finished the task, the mutex gets unlocked and the other thread gains right to use it.
Motion compression  Without further processing of the user's motion
data the virtual environment would be limited to the size of the user's real
environment, which is not desirable in most cases. As the present project
is concentrating on a manufacturing environment as a "virtual desk", the
3D room is big enough to satisfy the needs. Otherwise, if modeling of a
whole factory were needed, motion compression techniques could be used.
This allows the user to explore a relatively large virtual environment while
he is actually moving in a room of limited size. Motion compression aims
at giving users a realistic impression of the virtual environment by keeping
distances and turning angles in the two environments locally equal, only the
curvature of the path is changed. When walking, humans continuously check
if they are on the direct way to their desired target, and adjust their direction
accordingly. This behavior is exploited for user guidance. When walking,
the look direction is rotated slightly, the user tries to compensate it, and so
he/she follows the transformed path in the user environment, while moving
on the desired target path in the world model.

Navigation  Tracking a user's head makes the virtual world user-centered
by providing information used to create graphics (or audio, or other displays)
that are correct from the user's current perspective. In Maestro, the user can
switch between a first-person point of view or a world camera view on the
fly by appropriate function keys. If a large space was modeled and the user
wishes to navigate to an area not covered by the tracked area he/she could
navigate in the space by using the world camera. Dragging the viewpoint
around the world by hands in first-person-view can be realized as well.

Tracking the hand (or other body parts) allows a user to interact directly
with the environment. When the user is ready to work with models in the
environment, they can select modes and commands from a virtual toolbox
in the environment simply by reaching out their hand and grabbing a tool.
The same direct manipulation metaphor is used to select and manipulate
objects and give commands to the autonomous manipulators. We can see in
these two examples that using 6DOF devices for input can greatly enhance
interaction in 3D contexts.

Despite the user is standing in the middle of the 3D visualization room,
he perceives his actions in the virtual environment and can fully immerse
into the target environment.

3.2.2 Implementation

Maestro has been developed in Qt and C++ with Microsoft Visual Studio
keeping platform independency as a key aspect in mind. It has been also
tested on different Linux distributions including Debian with success. It is
operable on every platform where the Qt and OGRE libraries are available.
Embedding Ogre into Qt  Using Ogre for rendering has many advantages. It is platform independent and also abstracts the graphic libraries. Programming in Ogre is very intuitive and its developer community shares thousands of sample applications. There are several templates to create simple scenes and demos. An open-source modeling software Blender 2.46 has been applied to create objects for the 3D world model, e.g. the skeleton\textsuperscript{3} that represents the user in the virtual space.

![Figure 18: Rendering with Ogre in a Qt application](image)

Many reasons have led to create a GUI application with Qt instead of a pure Ogre3D application. First, allowing the user to select various different options without having to compile the source code again and again. The user can now set up everything in a window anytime he/she wants to use a different configuration. For instance, user might want to use Maestro program only for its rendering part without the robot control part. Secondly, the program has to be platform-independent because of obvious reasons. Furthermore, it has to support multithreading: receiving data from ShapeRecorder, sending commands to the robot, rendering the movements of ShapeWrap on the screen, each one of these actions has to be run independently from the others in a different thread. Finally, as mentioned before programming with Qt is much easier than using native C++ and makes the source code easier to read for future developments.

\textsuperscript{3}The skeleton model was made by Pierre Grenier
3.3 Overview

After the descriptions of every component Figure 19 repeats the schematic of the proposed system indicating the applied devices.

![System schematic with detailed components](image)

Figure 19: System schematic with detailed components

**Final components:**

- Motoman SSF2000 with NX100 controller as manipulator
- ShapeWrap III as input device
- Visualization Center as feedback
- Maestro software with Ogre rendering as central unit

The proposed system can accomplish the precomposed goals as it will be shown in the following Section.
4 Test and evaluation

Organizing tests has been a quite difficult task during the development, because the components are located at different places, and at each location must be a person. The motion-capture suit is located in Narvik, Norway while the 3D room and the manipulator are in Budapest. The central unit can theoretically be anywhere. Several successful tests have been executed with my French colleague Pierre Grenier. To explore system bugs and provide authentic measurements, testing the functionality of individual components is necessary before combining all together. As such, the first part of this section deals with testing each component and the second part evaluates the whole system in operation.

![Map of Europe with points marked for test locations](image)

Figure 20: Test locations: Budapest, Lille, Narvik

4.1 Component tests

The suit: As the suit is located in Norway, Gábor Sziebig from Narvik University College had to support the live-tests occasionally. He needed to check the ShapeWrap components every time prior to usage. The sensors require a certain amount of time to warm up, and they need to be calibrated if the body parameters of the user have changed from previous saved session. The suit itself is connected via wireless LAN to a PC on which Measurand’s ShapeRecorder is running and forwarding data packets to the Maestro unit through the internet. For other simple tests pre-recorded data files have been replayed by a locally installed ShapeRecorder instance. Accuracy of ShapeWrap is in sub-millimeter range.
**The robot:** The manipulator unit is located at GI Flex Szentesgyártó Kft., Budapest in a real industrial environment. The NX100 robot controller has a very intuitive graphical user interface with a PDA-like remote controller. After switching on it runs several self-tests automatically and in a few seconds it is ready to receive commands. The robot is mounted on a table and can reach a space of $\pm 50 cm$ on the sides and operate in a height of $50 - 130 cm$ measured from the ground. These limits are set because of present obstacles, the geometry of the robot could allow more freedom. The robot’s environment consists of a set of static known objects. There has been communication failures at the manipulator environment several times because of poor wireless network quality at the factory. The AI of the controller causes about 1.5 – 2s delay while calculating of the optimal path. On the contrary, accuracy of the robot is excellent for painting and welding applications indited in Section 1.2.

![Figure 21: Maestro in the 3D room](image)

**Visualization Center:** The visualization component was introduced me by my colleague Péter Zanaty. After tuning some parameters the stereoscopic feedback was adequate (see Figure 21). During the tests of the stereoscopic system the usage of the projectors had to be minimized because of the enormous amortization cost and the power consumption. The postulated 3D immersion has been achieved.

**Network:** Every part of the system has to be in the same network domain, so a virtual private network (VPN) has been created using Hamachi. The Hamachi clients must be running on every PC of the system and be connected to a central Hamachi server. The architecture is similar to the one that Skype is using. Clients need the server’s support to find other clients
but the data transfer is conveyed directly (see Section 3.1.6). The delay caused by Hamachi is not considerable. Average round-trip times (RTT) between the system components during the tests are listed here:

<table>
<thead>
<tr>
<th></th>
<th>distance</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budapest-Lille</td>
<td>~ 1400km</td>
<td>39ms</td>
</tr>
<tr>
<td>Lille-Narvik</td>
<td>~ 2100km</td>
<td>150ms</td>
</tr>
<tr>
<td>Narvik-Budapest</td>
<td>~ 2300km</td>
<td>80ms</td>
</tr>
</tbody>
</table>

These measurements must not be taken as stationary results. Round-trip time is affected by the actual network traffic and the router’s forwarding decisions are also dynamic. But these results give an idea on how much delay has to be expected when using Maestro while being geographically far from other components. It is to mention, that the Lille-Narvik connection has a much higher delay because of the higher number of hops during the way. This can be analyzed by trace route applications. In the final test only the Narvik-Budapest connection was established.

**Testing Maestro:** First and foremost it must be tested whether the communication between ShapeRecorder and the central unit is well established, which means that data sent by the suit can be correctly received and used in Maestro. The test was successful: at each frame Maestro received a UDP packet composed of numbers representing the position (X, Y, Z) and orientation (roll, pitch, yaw) of ShapeTapes and closed by a timestamp (see Figure 22). The timestamps are measured in ms and begin from a random number at a particular use. Typical timestamp difference between two datagrams is about 14 ms. The amount of coordinates may vary depending on used sensors in the suit. Therefore, the desired ShapeTapes must be checked in the setup window of Maestro.

The next step was checking the connection between Maestro and the NX100 robot controller. Pre-recorded motion files were played back on notebook A and Maestro were running on notebook B. There was a third PC with MotoProg which received the commands and forwarded to the controller. All PC’s were joined in the VPN. During the test everything went fine, the robot moved to the actual position of the hand at pressing a certain key. The 1 – 2s delay caused by the AI unit.

The third step was testing whether special gestures can be recognized by the central unit. The selected movement for the test was touching the thumb and the small finger of the left /right hand (the option can be selected in the setup window) with each other. After recognizing this event, Maestro forwarded the actual hand coordinates to the robot controller, which moved to the correspondent position. In order to avoid collision with other obstacles, the sent coordinates were limited to a narrow virtual box. These limits were also signed in the virtual space.
4.2 System test

The user can configure the behaviour of the system with the setup window by adjusting the parameters (see Figure 23). The user can select between re-playing a prerecorded movement or using movement data directly from ShapeWrap. Under the Tapes tab the user can select which tapes are attached. The network tab is devoted for setting the required IP addresses and the port numbers. The control of the robot and the stereo mode is also adjustable here.

Different configuration options have been tested to work under different platforms (windows, linux). The constant delay between Narvik and Budapest can not be shortened without improving the network which is expected in the future. However, the highest delay component is not caused by the information transfer but the information processing in the robot’s AI. This could be decreased by finding a more expensive manipulator, which does not correspond to the original goal of cost-efficiency. Due to this lag in the controller the symbol of the manipulator in the virtual space is in an invalid state for a second. In case of normal operation the feedback would be the changing color of the symbol according to the distance of the desired and the real position of the end-effector.

The assembled system works well even with the AI delay, however, demonstrating it in operation is too difficult, if possible, within this report being a printed material. A screen-shot of the video clip cut from the testing process in industrial environment is shown on Figure 24.
Figure 23: Setup screen on Windows

Figure 24: The system components in operation
5 Summary

5.1 Conclusion

The present B.Sc. thesis project has dealt with the integration of the human brain into the control loop of semi-automated manufacturing processes that require human supervision. The report has described the cognitive infocommunication discipline and the way its accomplishments can be taken into consideration by designing modern control systems.

A system has been described to integrate the human operator and the modules of computerized system employing cognitive infocommunication principles with success:

- A supervisory loop has been established over the control loop eliminating point-by-point trajectories utilizing on-board artificial intelligence
- Geographically dispersed parts of the system have been shown to cooperate successfully despite the underlying unreliable network connections thanks to the developed coordinator unit
- Intuitive gesture based human-machine interaction has been applied in order to increase productivity and reduce the learning curve
- The limited embedded intelligence of the industrial robot used has been proven to comply with the requirements of arc-welding
- Immersive stereo environment has been applied to aid the operator

Based on above conclusions and the successful test results, the initial objectives of the thesis project have been achieved.

Possible applications The list of possible applications includes:

- Applications of manipulators in mass production, painting, welding and dangerous material handling
- Complex human-aided operations in unconventional environments for humans: micromanipulation, operation in hazardous environments and under extreme circumstances (e.g. deep sea, outer space)
- Virtual laboratories, cyberlabs\(^4\)

\(^4\)http://extra.ivf.se/cyberlab/BudapestU.asp
5.2 Future potentials

This thesis project has been carried out as the first step of a long-term research activity within frames of the ITM Norwegian-Hungarian Joint Laboratory\(^5\). I am willing to continue the development of the proposed system in cooperation with other laboratory members. In the future I plan to design a thread in Maestro that deals with gesture recognition to increase input possibilities with more sophisticated movements. For example by clustering poses or by using a tracking framework to detect a gesture from stored or learned sequence of poses. The future development will also focus on computer supported cooperative work (CSCW) with two or more suits, enabling remote access to a process for a group of people. With enhanced security and high-tech robots, further applications in nuclear plants, in the deep ocean, in space or in micro/nanoscale can be envisioned.

\(^5\)Joint Laboratory for Emerging Information Technology in Manufacturing (ITM)

Members:

- Productive Programming Methods AS (PPM)
- Norwegian University of Science and Technology (NTNU)
- Narvik University College (NUC)
- Budapest University of Technology and Economics (BME)
- Computer and Automation Research Institute of Hungarian Academy of Sciences (MTA-SZTAKI)
Acknowledgement

Hereby I would like to express my gratitude to my supervisor, professor Péter Baranyi for his suggestions and valuable feedback. I also wish to thank Barna Reskó for his professional and moral support. Special thanks to Gábor Sziebig from Narvik University College for his help with the motion-capture suit, GI-Flex Szerszámgyártó Kft. for allowing me use their robot, Péter Zanaty for familiarizing me with the 3D room, Pierre Grenier for teaching me how to use Ogre and the Qt framework, and his kind support during the tests.

Furthermore I wish to thank my parents and my family, who have always supported me during my studies and provided all the conditions of starting a successful career. I am grateful to my girlfriend for her patience and love.
References

[1] Wilson, Emmanuel - Accuracy Analysis of Electromagnetic Tracking within Medical Environments, p.10 *CAIMR TR-2006-2 Technical Report, Georgetown University, January 2006*

(http://plato.stanford.edu/entries/cognitive-science/)


[4] Image courtesy of Péter Baranyi


[16] Visualization Center, Eötvös University (http://vc.elte.hu)


[21] OGRE3D 1.6.0 RC1 (http://www.ogre3d.org)

[22] Qt 4.4.2 (http://www.trolltech.com)