Ubiquitous Sensory Intelligence in Idustrial Robot Programming

Barna Reskó[1,3], Andor Gaudia[1,2], Péter Baranyi[1,3]
and Trygve Thomessen[1,4]

[1]Norwegian-Hungarian Joint Laboratory for
Emerging Information Technology in Manufacturing

[2]Budapest University of Technology and Economics, Budapest, Hungary

[3]Computer and Automation Research Institute
Hungarian Academy of Sciences
Budapest
Hungary

[4]Productive Programming Methods AS
Trondheim
Norway

Abstract: Intelligent Space based on ubiquitous computing is a space which has distributed sensory intelligence and is equipped with actuators. The various devices of ubiquitous sensory intelligence cooperate with each other autonomously, and the whole space has high intelligence, where we can easily interact with computers and robots, and get useful services from them. This paper gives an overview of a system that is based on the Intelligent Space concept. The proposed system is one of the key components of the so called Intuitive Robot Programming (IRP), used for intelligent fast and effective programming of industrial robots. In IRP the desired motion of the robot is performed by a skilled worker. The human motion is captured and its path is edited and simulated. Finally the result can be programmed into the robot.

1 Introduction

Intelligent space is a limited space (room or building, street or area, or even a whole country), which has ubiquitous computing type computing and sensory intelligence. The sensors might be various types of equipment, such as cameras, microphones, haptic devices, weight sensors, or any other devices that collect information on the state of the space. A conceptual figure of the Intelligent Space is shown in Fig. 1. [1]

The various devices of ubiquitous sensory intelligence cooperate with each other autonomously and the whole space has a ubiquitous computing background. This is true even if there is a supervision system involved, which is acting as an autonomous agent itself. Each agent in the space has sensory intelligence. (Or has intelligent inputs coming from other agents.) An intelligent agent has to operate

even if the outside environment changes, so it needs to switch its roles autonomously and knowing its role it can still help and support humans within the space. Intelligent Space with ubiquitous computing recomposes the whole space from each agent's sensory information and returns intuitive and intelligible reactions to human beings. In this way the Intelligent Space is the space where human beings and intelligent ubiquitous computing agents can interact mutually.[2] [3]

Nowdays Intelligent Space based systems can be found mainly in research laboratories, but they are still rare in everyday life. In this paper a possible industrial application of an Intelligent Space will be proposed.

Intuitive Robot Programming (IRP) is a new concept for fast and efficient industrial robot programming. The main idea in IRP is to teach the robot to perform a motion that is the same or similar to the motion of a human. This way a skilled worker can show the robot the desired motion. This motion can later be edited, processed and modified, and finally it can be uploaded into the robot, which will then be able to perform it millions of times.

The IRP system consists of the followning components:

- motion capture component
- motion processing component
- motion performing component

The motion capture component tracks and records the motion of a skilled operator performing the desired task. The motion processing component allows to display the motion to be programmed into a robot. It also offers possibilities to process and modify the motion path. Finally the motion processing component transforms the path data into a standard robot program which can be directly uploaded into the motion performing component, which is ususally an industrial robot.[4] [5]

This paper proposes an Intelligent Space based system as the motion capture and motion processing component in an IRP concept.
The proposed system recovers 3D path information using 2D information of multiple video cameras. The motion capture component is structured according to a client-server model. Each client handles one video camera and is responsible for image acquisition and image processing. Each client sends 2D data to the server, which is responsible for 3D path reconstruction. A 3D path consists of a series of control points that hold position and orientation information. The proposed system can be organized into a distributed camera system, where the clients are intelligent agents.

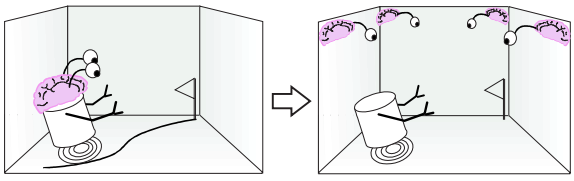

Figure 1: Conventional concept vs. Intelligent Space concept

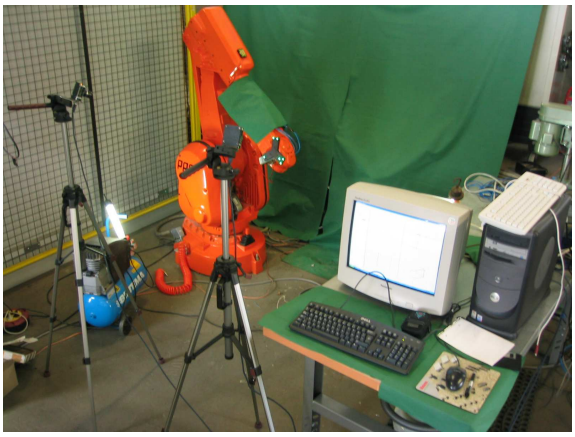

Figure 2: Testing the Intelligent Space based system in an IRP environment.

They communicate with the server, which receives intelligent data from the clients and produces more informative 3D data from it. This distributed setup is a small Intelligent Space. The Intelligent Space concept had been proposed by the Hashimoto Laboratory of the Institute of Industrial Sciences at the University of Tokyo.[6] The main goal of the authors was to design and implement a system that can easily be extended to become the part of the Intelligent Space at the Hashimoto Laboratory, while it perfectly satisfies the needs of an industrial application based on the IRP concept.

A two camera version of the proposed system has been implemented at the Computer and Automation Research Institute of the Hungarian Academy of Sciences. The system was integrated in an IRP environment and has been presented and tested in a real industrial environment in Norway (Fig. 2).

The organization of this paper is as follows: in section 2 a description about the background of the research is given. In section 3 the developped two-camera system is discussed along with the motion processing component. In section 4 the results of the tests is presented and the system is evaluated. And finally in section 5 we conclude the paper and point on some further issues.

2 Background

2.1 The Intelligent Space and Ubiquitious Computing

Intelligent Space is a space (room, corridor or street), which has distributed sensory intelligence (various sensors, such as cameras and microphones with intelligence, haptic devices to manipulate in the space) and it is equipped with actuators [2]. Actuators are mainly used to provide information and physical support to the inhabitants. This is done by speakers, screens, pointing devices, switches or robots and slave devices inside the space. The various devices of sensory intelligence cooperate with each other autonomously, and the whole space has high intelligence [7]. Each intelli-

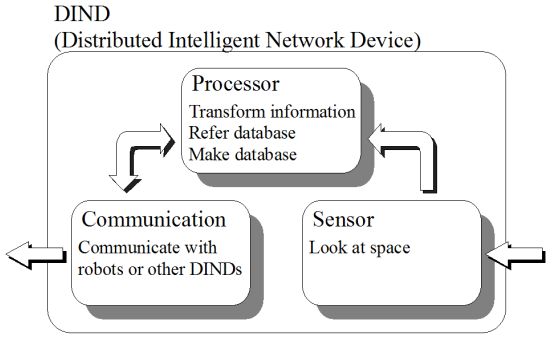

Figure 3: Fundamental structure of a DIND.

gent agent in the Intelligent Space has sensory intelligence [6]. An intelligent agent has to operate even if the outside environment changes, so it needs to switch its roles autonomously. The agent knows its role and can support human beings in the Intelligent Space. Intelligent Space recomposes the whole space from each agent's sensory information, and returns intuitive and intelligible reactions to human beings. In this way, Intelligent Space is the space where human beings and agents can act mutually.

Basic elements of Intelligent Space:

- Distributed Intelligent Network Device
- Virtual Room
- Ubiquitous Human Machine Interface

The key element of Intelligent Space is called the Distributed Intelligent Network Device (DIND), which consists of three basic components (Fig. 3). These are the sensor, the processor (computer, neural network or even a brain) and a communication device. [8] Thus, a DIND is a unit based on three functions. The dynamic environment, which contains people, vehicles, robots, etc., is monitored by the sensors of the DIND, the information is processed into a form easily captured by the clients in the processor and the DIND communicates with other DINDs through a network or a supervision system, which is itself an autonomous agent.

2.2 Stereo vision, stereo matching

The proposed client-server system recovers 3D path information using 2D information of video cameras. Despite the wealth of information contained in a photograph, the depth of a scene point along the corresponding projection ray is not directly accessible in a single image. With at least two pictures, on the other hand, depth can be measured through triangulation. This is of course one of the reasons why most animals have at least two eyes and/or move their head when looking for friend or foe, as well as the motivation for equipping autonomous robots with stereo systems (Fig. 4).

Having several views of the same scene make it possible to construct its three-dimensional structure. Stereo vision involves two processes: the binocular fusion of features observed by the two eyes, also called stereo matching, and the reconstruction of their three-dimensional preimage.
The latter is relatively simple: the preimage of matching points can (in principle) be found at the intersection of the rays passing through these points and the associated pupil (camera) centers. Thus, when a single image feature is observed at any given time, stereo vision is easy. However, each picture consists of hundreds of thousands of pixels, with tens of thousands of image features such as edge elements, and some method must be devised to establish the correct correspondences and avoid erroneous depth measurements [9].

3 Description of the proposed system

As a starting point a brief description will be given about the developed system. The goal is to sketch the main aspects and components of the system in order to give a complete overview. The goal of the system is to track the motion path of a skilled worker and then after processing and modification upload the path to the robot, which will then perform the desired task.

The developed system is composed of a set of hardware and a software components. The hardware components were either purchased or self-designed and manufactured. The hardware consists of the following items:

- One or more PC connected in a network
- Two or more video cameras
- A marker object
- A calibration target object

The PC is used to acquire images from the cameras. The cameras observe the marker object that is moving in the three-dimensional space. The camera images are processed by a software on the PC and are used to reconstruct the three-dimensional position and orientation of the marker. In order to be able to reconstruct the position of points in space, the cameras have to be calibrated. The calibration

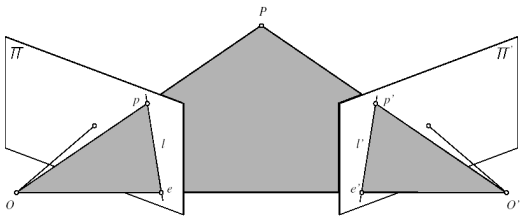

Figure 4: The geometry of stereo vision.

target object has been designed to help the calibration of the cameras. The software is composed of a motion capture package and motion processing package. The motion capture package has two components that cooperate with the hardware. One of the components is a client application attached to one camera, and is responsible for the management of the camera attached to it. The other component is a server application associated with many client application and is responsible for the three-dimensional reconstruction of a point according to data received from the client applications.

The basic tasks of the client application can be summarized in the following points:

- Camera calibration
- Image acquisition
- Image processing
- 2D marker extraction and identification - 2D data management
- Graphical User Interface

The server application is responsible for the following tasks:

- Data acquisition from the clients
- 3D data reconstruction

3.1 A distributed multi-camera motion capture system

After taking all constraints, ideas and goals into consideration, a distributed multi-camera motion capture system was designed and implemented. Each client application is connected to a camera, which is observing the space and looking for the markers fixed on the hands or the tools of the skilled worker. The client application does the image processing task, the result of which is a data set of three 2D points. These points are measured and sent to the server application periodically from all the clients.

The server application collects the 2D data as well as the calibration matrices from the clients. The calibration matrices contain information about the spatial parameters of the cameras in a fixed coordinate system. The server application then calculates the 3D point from the 2D data. The 3D point should correspond to the intersection point of the rays starting from the 2D projection on each CCD chip and passing through the center of projection of each camera. This intersection point however exists only in ideal cases, but in practice the rays will never intersect each other. The solution can be defined as the 3D point whose mean squared distance from the rays is minimal.

A pure algebraic solution can also be given to reconstruct a 3D scene point. In a calibrated stereo rig given two matching image points p and p' and two perspective projection matrices $\mathbf{M}$ and $\mathbf{M}'$. The constraints $z\mathbf{p} = \mathbf{MP}$ and $\mathbf{z}'\mathbf{p}' = \mathbf{M}'\mathbf{P}$ can be rewritten as

$$\left\{ \begin{array}{l} \mathbf{p} \times \mathbf{MP} = 0 \\ \mathbf{p}' \times \mathbf{M}'\mathbf{P} = 0 \end{array} \right. \Leftrightarrow \left(\begin{array}{c} [\mathbf{p}_\times]\mathbf{M} \\ [\mathbf{p}'_\times]\mathbf{M}' \end{array} \right) \mathbf{P} = 0. \quad (1)$$

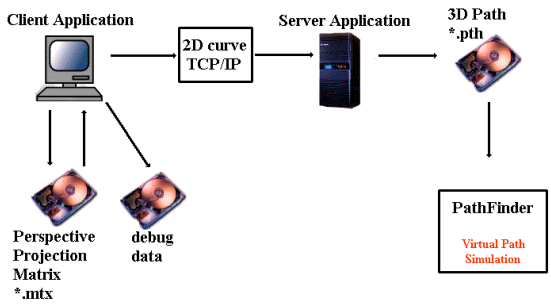

Figure 5: Distributed system

Equation $\mathbf{p} \times \mathbf{MP} = 0$ is the equation of the line that is passing through points p and P. If the number of lines $n \geq 2$ in the system, there is an over-constrained system of $n \times 3$ linear equations in the homogeneous coordinates of P, that is easily solved using the SVD based linear least-squares techniques [10]. Unlike the geometrical approach, this reconstruction method does not have an obvious geometric interpretation, but it generalizes readily to the case of three or more cameras, each new imaging device simply adding two additional constraints.

In the application the cameras follow the motion of a marker object, which has three fixed points on its surface. These points define a plain and a position. The orientation of the plain of the marker object is described by a quaternion, and its position is described by a 3D vector. In all, there is a 7-element-long vector for each position of the marker in time.

The 3D information can be used to generate a robot program, which would drive the robot to repeat the same motion performed by the marker object.

The use of more than one computer is also possible with the developed system. This allows to build up an Intelligent Space where the clients are considered as DINDs. Their sensors are the cameras, the processing unit is the image processing unit, while they also have a communication unit that uses TCP/IP to communicate with the server. For this reason the ability to communicate via the TCP/IP protocol has been added. As a result, the server can run on a remote computer, and the clients can also be split to run on many computers. Fig. 5 shows the concept of the distributed system using TCP/IP of the Internet.

The TCP/IP connection also allows to remote control the clients from the server application. This feature is quite useful. If the clients are far away from each other, it would be impossible for the user to go to each remote computer in order to start and stop the capture process, or to calibrate the cameras. Instead, all these functions can be centralized and controlled using the server application.

3.2 Robot Simulations - PathFinder

Pathfinder is the part of the motion processing component of the IRP based system. This software receives the 3D path data from the motion capture component and it helps users in programming robots. (Fig. 6).

The following operations can be done using the PathFinder:

- Importing different pathways created by the Optical Motion Tracking system
- Editing pathways
- Joining pathways
- Displaying an industrial robot
- Placing obstacles around the robot
- Simulating robot movements
- Detecting collisions and invalid arm positions
- Generating a robot program based on S3 robot language

The simulator software is quite flexible. It supports a VRML file format so it can be used together with many well known 3D software. Different industrial robot models can be downloaded; the software user only has to provide some robot specific datas like: Denavit-Hartenberg matrices.

A realistic 3D robot room can be built up by placing some objects to the virtual space (Fig. 7).

In case some error occurs during simulation, the user is warned. For example: the robot's arm crashes into the robot cell's wall. Using simulations these situations can be discovered and by modifying the robot's trajectory the collision in the real situation can be avoided.

4 Testing and Evaluation

The final testing of the system took place in a laboratory in Norway using several industrial robots. The goal was to find out the limits, the precision of the motion capture system. The PathFinder application was very important part of the system testing procedure. It could also be considered as the part of the system, and the tests also aimed to reveal how PathFinder performed. The experimental test setup had been put together in the laboratory. The setup was composed of the following components

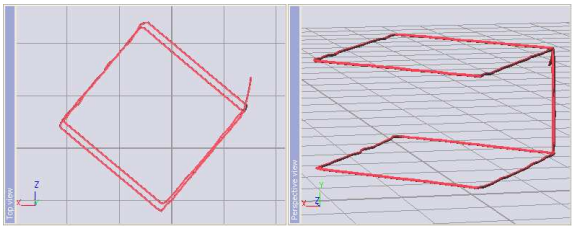

Figure 6: Imported pathway

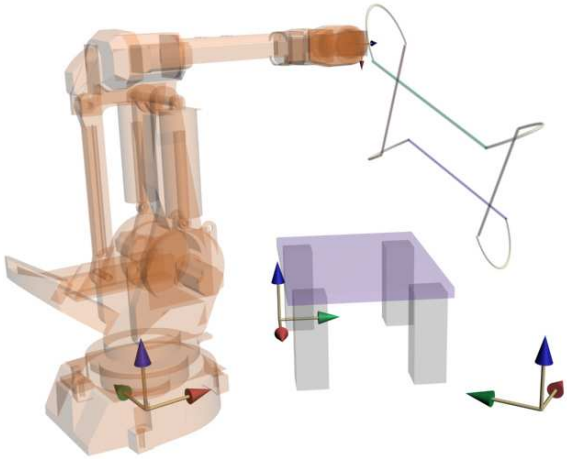

Figure 7: 3D environment

- A Personal Computer running Microsoft Windows XP
  - Server application
  - Two client applications
  - PathFinder application
- Two PL-A633 cameras connected to the PC
- Calibration object
- Marker object
- ABB IRB2000 6-axes industrial robot
- Controlled light conditions and dark green background

The testing procedure has to reveal the accuracy of the system. The accuracy may depend on the speed of the marker object. Since the camera reads the image from the CCD line by line, the times when the image was taken on the top and on the bottom of the image do not correspond. For this reason the time stamp of each point had to be compensated by the time that has elapsed between the lecture time of the first line and the actual line. Since no official information was available about the elapsed time between scanning two lines on the camera, this time had to be assessed.

Originally the system was developed to track human motion and transfer that motion into a robot. Let's turn this idea inside-out! If the robot is programmed to move on a well-known path, this path could be measured by the motion tracking system. The marker object simply has to be mounted on the end-effector of the robot, and everything is working.

To test the motion tracking system, the ABB robot was programmed to move the endeffector on the wireframe of a box, whose size was $400 \times 400 \times 200$ millimeters. During the motion the robot maintained a constant orientation of the end-effector.

After calibrating the cameras, the motion of the robot was launched, and the system could follow the marker without any problem. The speed of the end-effector was set to 100 mm/sec in the robot program. The speed could be multiplied by a speed scale factor between 50% and 400% using the control console of the robot. This scale factor allowed to test the system at different speeds. Initially the speed was set to 50 mm/sec, then to 100, 200 and 400 mm/sec. The results can be seen in Fig. 8.

Observing the first image, a perfect cube is seen. The system is working perfectly at such a speed. However, if the speed is increased, the accuracy of the system decreases, especially in the corners of the cube. The cause is that the robot is moving at a constant speed, and in the corners its acceleration is very high, in order to change the velocity by 90 degrees in the fraction of a second. The overall precision of the system was measured using the zoom option of the PathFinder. The path of the cube at 50 mm/sec was used. In order to better view the magnitude of the error, a grid of 1mm was laid in the background. The result is shown in Fig. 9. It is visible that there is a small deviation from a perfect line in the output path. However, this deviation is in the order of one millimeter. Since the ABB IRB2000 robot has also a precision of approximately one millimeter, it is impossible to find out from this image if the error is due to the robot or the motion tracking system. The path of a cube was perfect for measuring the precision of the system.

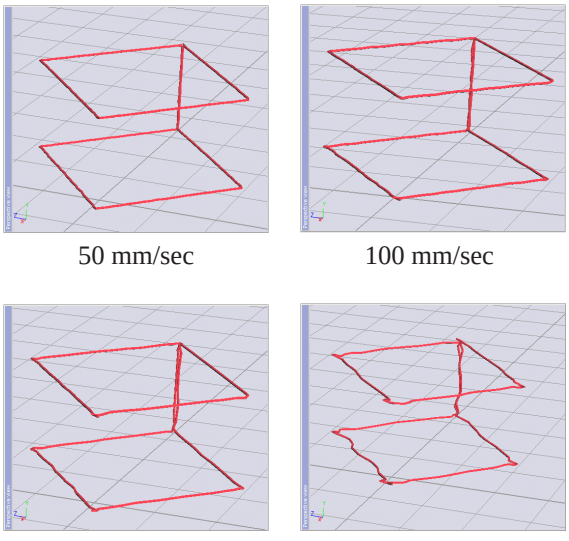

Figure 8: The output path of the system at different velocities of the end-effector.

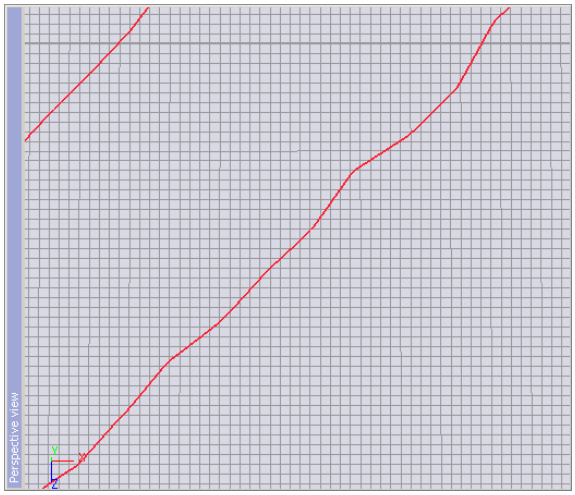

Figure 9: The output path of the system at a speed of 50 mm/sec. The grid behind the path has a resolution of 1mm.

5 Conclusion

The precision of the system depends on the speed and curvature of motion, the time between two measurements (i.e. the frame rate), the number of pixels on the camera, the field of view of the camera and also its distance from the tracked object. For example with the PL-A633 camera with 1280×1024 pixels and 7 frames per second the results show that it's possible to track an object within a workspace of 1.2 meters from a distance of roughly 2.5 meters. Meanwhile the accuracy remains 1mm. If the object is moving at 100mm/sec, it's possible to track a curve with radius down to 100mm within 1mm error. However, if the speed gets to 400mm/sec, the maximum possible radius of the curve is 1.6 meters in order to have an error within 1mm. These results are within the requirements to application of the system in the industry. The resulting system was presented and evaluated in the laboratory in Norway. The goal of the evaluation was to see how the system performs in a real industrial environment as the part of an IRP based system.

The results of the evaluation were presented in the previous section. The system can also be considered as a small Intelligent Space, since it is capable of operating as a distributed motion capture application communicating through a TCP/IP based network. The present application also proves the ability of an Intelligent Space based system to operate as the part of a state of the art industrial system.

In the future further intelligence can be added to the system. For example the Intelligent Space could observe the robot and supervise its operation. Using a calibrated robot, the camera calibration can be made by moving the robot to well specified spatial positions. In this case the calibration process becomes much

more flexible.

Acknowledgement

The authors wish to thank the JSPS Fellowship program, National Science Research Fund (OTKA T034654) and Control Research Group of Hungarian Academy of Science for their financial support.

References

[1] Hideki Hashimoto Peter KORONDI, Peter Szemes. Ubiquitous sensory intelligence. *International Conference in Memoriam John von Neumann*, pages 73–86, 2003.

[2] J.-H. Lee and H. Hashimoto. *Intelligent Space - Its concept and contents*, volume 16. Advanced Robotics Journal, 2002.

[3] C. Szabó P. Korondi P. Szemes R.-E. Precup, S. Preitl. A low cost solution for the navigation problem of wheeled mobile robots. *Transactions on AUTOMATICCONTROL and COMPUTER SCIENCE*, 49(63):77–82, 2004.

[4] Hideki Hashimoto Peter T. Szemes, Peter Korondi. Human behavior based mobile agent control in intelligent space. *Transactions on AUTOMATIC CONTROL and COMPUTER SCIENCE*, 49(63):5–11, 2004.

[5] Kazuyuki Morioka Hideki Hashimoto, Peter T. Szemes. Intelligent space -robots and spaces-. *Proceedings of International Symposium on Electronics for Future Generations*, pages 383–388, 03 2004.

[6] Hideki Hashimoto Kazuyuki Morioka, Joo-Ho Lee. *Human Centered Robotics in Intelligent Space*. 2002.

[7] J.-H. Lee T.Akiyama and H. Hashimoto. *Evaluation of CCD Camera Arrangement for Positioning System in Intelligent Space*. International Symposium on Artificial Life and Robotics, 2001.

[8] Peter T. Szemes Hideki Hashimoto. Ubiquitous haptic interface in intelligent space. *SICE Annual Conference*, pages 3277–3282, 08 2003. Fukui, Japan.

[9] Jean Ponce David A. Forsyth. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[10] Eric W. Weisstein. Singular value decomposition. *MathWorld–A Wolfram Web Resource*, 1999.