Robot Programming Based on Ubiquitous Sensory Intelligence

Andor Gaudia
Norwegian-Hungarian Joint Laboratory for
Emerging Information Technology in Manufacturing
Budapest University of Technology and Economics

Budafoki út 8, H-1111 Budapest
Hungary
andor.gaudia@mailbox.hu

Barna Reskó
Norwegian-Hungarian Joint Laboratory for
Emerging Information Technology in Manufacturing
Computer and Automation Research Institute
of the Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest
Hungary
rbarna@datatrans.hu

Trygve Thomessen
Norwegian-Hungarian Joint Laboratory for
Emerging Information Technology in Manufacturing
Productive Programming Methods As
Leirfossveien 27, N-7038 Trondheim
Norway
trygve.thomessen@ppm.no

Péter Korondi
Norwegian-Hungarian Joint Laboratory for
Emerging Information Technology in Manufacturing
Budapest University of Technology and Economics
Kende u. 13-17, H-1111 Budapest
Hungary
korondi@elektro.get.bme.hu

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 document proposes a system which is using Intelligent Space to help the process of the Intuitive Robot Programming. In IRP the desired motion of the robot is performed by a skilled worker. The motion is recorded by some motion capture equipment. Finally the captured motion path can be edited and simulated by a software, and the result can be programmed into a robot.

I. INTRODUCTION

A. Intelligent Space

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. [6]

The various devices of ubiquitous sensory intelligence cooperate with each other autonomously and the whole space has 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. I-space having ubiquitous computing recomposes the whole space from each agents sensory information and returns intuitive and intelligible reactions to human beings. In this way, i-space is the space where human beings and intelligent ubiquitous computing agents can interact mutually.[5] [8]

B. Intuitive Robot Programming

Intuitive Robot Programming (IRP) is a new concept. This concept significantly reduces the programming time of industrial robots. The IRP system consists of:

- coordinate tracking device
- software package
- industrial robot

The coordinate tracking device captures the motions of a skilled operator that accomplishes the desired task, the software package which processes the data into a standard robot program to be uploaded directly to the robot controller. The software package also contains 3D visualization, editing functions and application packages. IRP will contain interfaces to

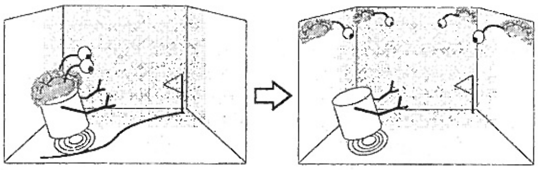


Fig. 1
CONVENTIONAL CONCEPT → INTELLIGENT SPACE CONCEPT

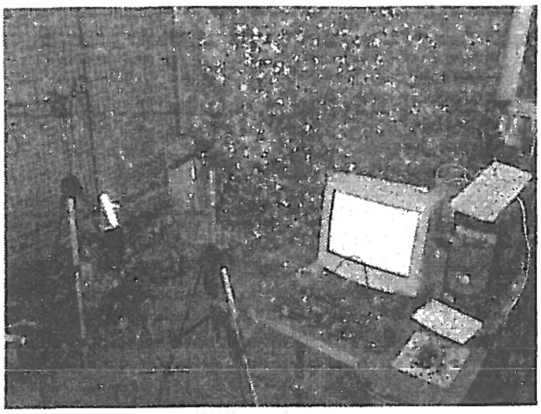


Fig. 2
TESTING THE MOTION CAPTURE SYSTEM.

the most common robots on the market and is connected to the robot controller via standard interfaces and thus, without any need for adaptations.[7] [2]

C. *IRP and Intelligent Space*

This document proposes a multiple camera based system to perform the motion capture task in the IRP (intuitive Robot Programming) concept.

The proposed system recovers 3D path information using 2D information of video cameras. The system is structured according to a client-server model. Each client handles one video camera and is responsible for image acquisition, 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 system, where the clients are intelligent agents. They communicate with the server, which receives intelligent data from the clients and produces more intelligent 3D data from it. This concept is called the Intelligent Space concept, proposed by the Hashimoto Laboratory of the Institute of Industrial Sciences at the University of Tokyo. 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.

D. *Test results*

A two camera version of the designed system has been implemented at the Computer and Automation Research Institute of the Hungarian Academy of Sciences. The system was presented and evaluated in a real industrial environment in Norway (Fig. 2). The evaluation provided valuable information about the limitations and the precision of the system. The precision is within one millimeter at an object distance of about 2 meters. The system perfectly fits into both the IRP and the I-Space.concept. All the objectives of the present work have been satisfied.

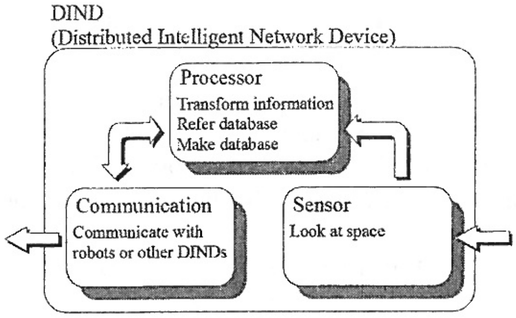


Fig. 3
FUNDAMENTAL STRUCTURE OF DIND.

II. DETAILED DESCRIPTION

A. *The Intelligent Space*

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 [5]. 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 [9]. Each intelligent agent in the Intelligent Space has sensory intelligence [4]. 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 concept called the Distributed Intelligent Network Device (DIND) consists of three basic elements (Fig. 3). These are the sensor, the processor (computer, neural network or even a brain) and the communication device. [3] Thus, a DIND is a unit based on three functions that the dynamic environment, which contains people, vehicles and robots, etc., is monitored by the sensor, 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.

B. 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).

Before building such a program, we must understand how several views of the same scene constrain 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 [1].

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 developed system is composed of a set of hardware and a software components. The hardware components were either purchased or self-designed and implemented. 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 calibrates. The calibration target object has been designed to help the calibration of the cameras. The software is composed of two components that were conceived to 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.

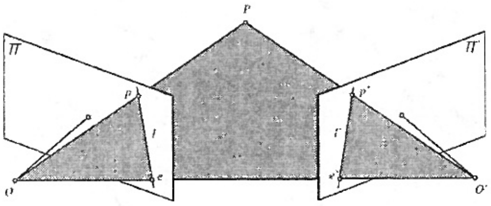


Fig. 4

THE GEOMETRY OF STEREO VISION.

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

C. A distributed multi-camera vision system

After taking all constraints, ideas and goals into consideration, a distributed multi-camera vision system was designed and implemented. The system is composed of a set of hardware and software components. The hardware components are:

- One or more PC connected to the Internet
- One PCI FireWire IEEE 1394 interface card for each camera
- Two or more PL A633 video cameras
- A marker object
- A calibration target object

The cameras are connected through the IEEE 1394 interface to the PC. The bandwidth of the FireWire interface is 400 Mbit/sec, 200 Mbit/sec in each direction (uplink and downlink). The PL-A633 camera has a 1.3 MegaPixel resolution at 8 bit color depth. This implies that the FireWire can carry

$$\left\lfloor \frac{200}{1.3 \cdot 8} \right\rfloor = 19\text{fps}.$$

This value is the theoretical frame rate, in practice if the image is acquired in the form of a video stream from the camera, 15 frames per second can be reached.

III. PRACTICAL RESULTS

The solution is to use two FireWire cards both inserted in the PCI slot of the PC, since the PCI slot has a bandwidth of 1056 Mbit/sec. The one camera - one FireWire card configuration works perfectly, without losing the high frame rate. The client application that handles one PL-633 camera, and the server application that collects 2D information from clients, and

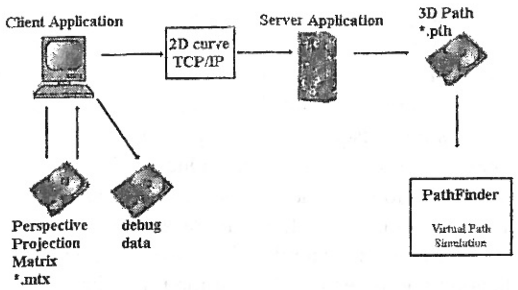


Fig. 5

DISTRIBUTED SYSTEM

reconstruct 3D information. 2D curves can be transferred into the server application by

- Files on the Hard Drive
- TCP/IP protocol

The 3D information can be utilized to generate a robot program, which would drive the robot to repeat the same motion performed by the marker object. The path can also be used to generate a virtual path using the PathFinder application. It is to note that PathFinder can also generate robot programs using the path data received from the developed server application. The data flow within the system was first based on the file system of the operating system. The client application could save the 2D curve data into a file whose extension is usually .cur. The server application can import many 2D curve files. The 3D reconstruction is done according to all the .cur files imported in the server. The output of the server is a file that contains a 3D path. The client application also allows to save and open the data stored in the perspective projection matrix. The files are saved into a file with an extension of .mtx. Imagine when a camera is fixed on a wall, and its extrinsic and intrinsic parameters remain the same. As a result, the perspective projection matrix will also remain the same, thus the cameras do not need to be calibrated. Their previous perspective projection matrix can simply be opened from a file that was previously created.

The file based communication works well if both the client and the server run on the same computer. But if they have to be split, file transfer from one PC to another may take a long time, and it makes the system quite rigid, uncomfortable for the user. The use of more than one computer would not be a rare case, since the system had been designed to be distributed. For the reasons above, in the final steps of the development, the communication between the clients and the server was extended. 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 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.

A. Robot Simulations - PathFinder

This software helps users in programming robots, since it is a universal interface designed to load different 3D paths generated by the Optical Motion Tracking system (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.

IV. TESTING AND EVALUATION

The final testing of the system took place in a laboratory in Norway where several industrial robots helped our work. The goal was to find out the limits, the precision of the motion tracking system, and finally try to crash it. 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.

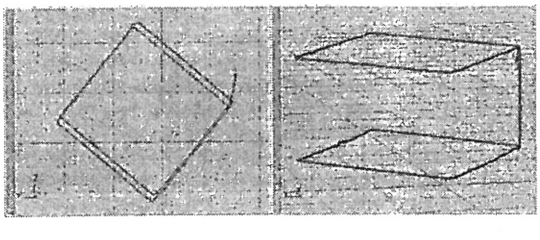


Fig. 6

IMPORTED PATHWAY

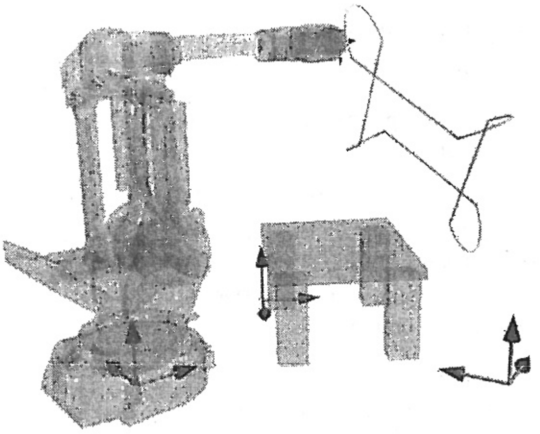


Fig. 7
3D ENVIRONMENT

The experimental test setup was put together in the laboratory. The setup was composed of the following components

- 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 to accuracy of the system. The accuracy may depend on the speed of the marker object. The time compensation between the lines of the image acquired from the camera can also affect the precision of the system. Since no official information was available about the elapsed time between scanning two lines on the camera, this time had to be assessed.

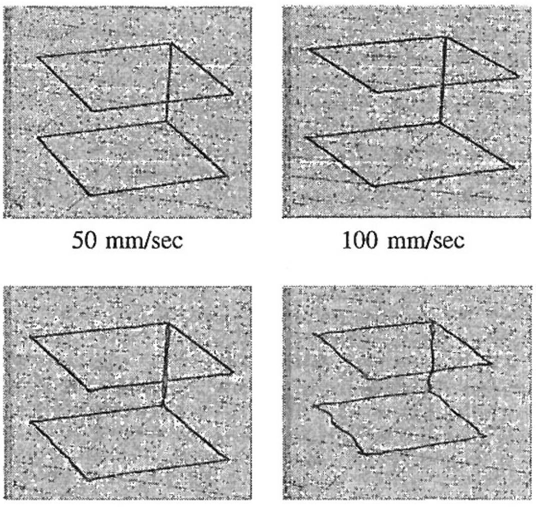
Originally the system was developed to track human motion and transfer that motion into a robot. Lets turn this idea inside-out! If the robot is programmed to move on a well-known path, this path could be measured by the motion tracking system. The marker object simply has to be mounted on the end-effector 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 400×400×200 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.

V. 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 its 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, its possible to track a curve with radius down to 100mm within



50 mm/sec
100 mm/sec
200 mm/sec
400 mm/sec

Fig. 8
THE OUTPUT PATH OF THE SYSTEM AT DIFFERENT VELOCITIES OF THE END-EFFECTOR.

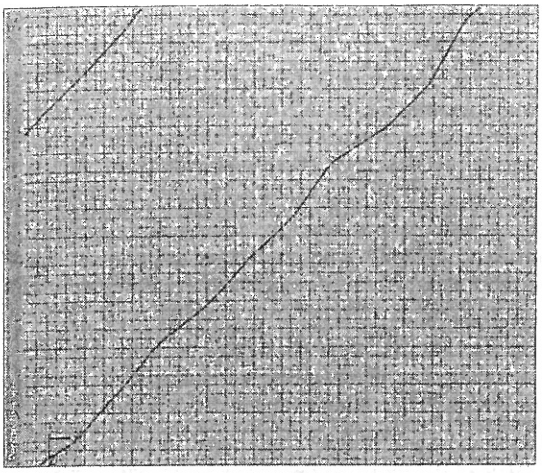


Fig. 9

THE OUTPUT PATH OF THE SYSTEM AT A SPEED OF 50 MM/SEC. THE GRID BEHIND THE PATH HAS A RESOLUTION OF 1MM.

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.

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

[2] 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.

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

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

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

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

[7] 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.

[8] 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.

[9] 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.