Application of the computer vision technology to control of robot manipulators

V. V. Riabchenko, A. V. Nikitin, V. N. Khomenko, A. A. Melnyk

The paper displays main results of the Ukrainian-French research project which studies interaction of a robot with its environment. Following applications of computer vision technology for various issues of robotics are shown: robot follows movements of a person, detects speed of the moving object, studies its environment, learns particular manipulations, gets trained to focus attention on the objects of interest and synchronizes its motion with an interacting agent. The article proposes approaches to development of the software interface for the robot control based on the computer vision technology.

Introduction

Computer vision is a theory and technology of creating machines that can detects and classify objects and their movement receiving information from the series of images. The domain of computer vision can be characterized as developing, diverse and dynamic. Important issue of the artificial intelligence is an automatic planning or decision-making in systems which can performs mechanical actions, such as moving a robot through certain environment. This type of processing usually needs the input data given by the computer vision systems, which provides high-level information about robot's environment, related sensors and devices or video devices [1]. The computer vision expands the number of solved problems in robotics. For example, programming by demonstration [2] for construction of visual-motor maps during robot learning [3]; robot and human interaction based on synchronization of their activities [4].

Objectives of research

The main objectives of the research are as follows:
- study of the various computer vision technologies applied for robotics;
- practical implementation of algorithms for robot's behavior, based on the incoming visual information;
- solutions to certain problems of artificial intelligence which use digital image data sets (optical flow, projection of the scene upon image and stereo image, etc...)

This work is performed in the framework of Ukrainian-French research project DonNTU — University of Cergy-Pontoise and the collaborative Ukrainian-French educational master program "MASTER SIC-R".

Subject of research

Robots which are used in our research are: 1) Katana5M, 5 axes (figure 1, a) provided by the laboratory "Equipes Traitement de l'Information et Systemes" (ETIS, ENSEA, University of Cergy-Pontoise, CNRS, F-95000, Cergy-Pontoise) placed in the laboratory "Control of interactive robotic electro-mechanical systems", Electrotechnical Faculty of DonNTU; 2) Katana6M, 6 axes (figure 1, b) located at the laboratory ETIS of UCP.

Katana robot is controlled by the controller that provides an interface — pre-installed commands set (Firmware). Control system allows the robot to follow commands and updates the software about robot's present status. Consequently, the master program sends the command

---

**Figure 1.** a) Katana5M, 5 axes; b) Katana6M, 6 axes.

stream according to the desired behavior of the robot. Developed architecture (software) for controlling the robot can be divided into two parts: master software which runs on the computer, and robot Firmware, which is executed by the robot's controller. Packets between computer and Katana (command, response) are transmitted via serial port.

In our work we use the OS Linux (kernel build 2.6.32-37) for many reasons, for instance: high stability (so-called uptime), security and efficiency, broad opportunities for configuring Linux to the task performed by the computer (or other control device).

Controlling the robot only by using the controller is a matter of some difficulties, such as: need to form a package for each single connection, motion, data acquisition, and time-taking analysis of the responses. The good point is that at the same time, the object of control can be fully represented at different levels of abstraction. Therefore, we developed libraries provide application programming interface (API) for the robot Katana (classes, functions, entities, etc.), which can be used for a wide range of scientific and practical problems. This API is based on KNI (Katana Native Interface), which is C++ library for controlling robots series Katana, distributed by their manufacturer — Neuronics AG. Development is conducted with using GNU gcc and g++ compilers and IDE ECLIPSE, so that the libraries can always be compiled and ready for using on any LINUX-compatible device. Architecture interface is separated into five levels (figure 2):

1. On the communications layer (KNI Communication Layer) implemented functions (sending and receiving control information) transmit between the computer and the robot. It consists of following:
   (a) device layer (KNI Device Layer) where the entities of data transmission are described (structures, functions);
   (b) protocol layer (KNI Protocol Layer), on which these entities are implemented in accordance with current technical features (operating system, device types, ports, etc.). Since we exploit Linux as an operating system and a serial port for data transfer we used the appropriate tools.
2. On the base layer the basic entities are described, such as: a robot Katana, engine, capture, sensor, etc. and the basic functions are implemented (turn the i-th joint in position XX enc readout sensors, move to position "XX, XX, XX, XX, XX", etc...) Classes and functions at this level use the entities of the communication level and do not depend on the specific of implementations.
3. Entities of abstract layer provide capability of direct and inverse kinematics, coordinate systems, and intelligent but easy-to-use functions to control the robot Katana.

Robot controlled by this API that is functionally ready to perform almost any tasks. Visual data processing programs, which are naturally located at the highest level (application layer), will use this functionality of the robot to implement the appropriate robot's behavior and reaction to stimuli (moving or varying objects) [5]. Visual data (images, optical flow) obtained from the cameras are processed by OpenCV. It is a C++ library of algorithms for computer vision, image
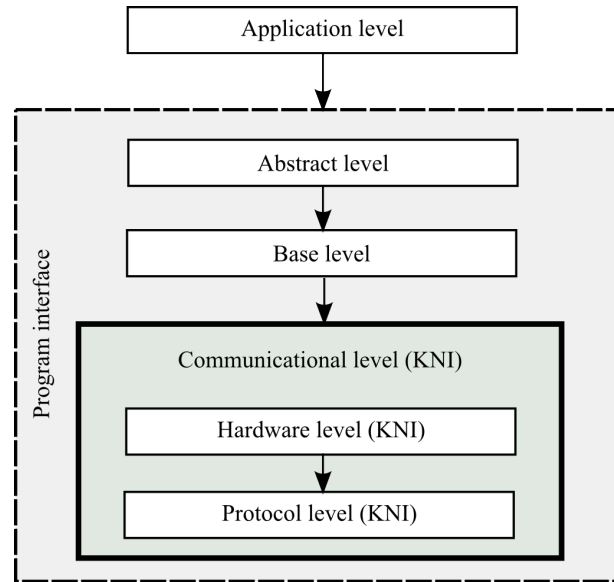
Figure 2. Architecture of the program interface.

processing, and numerical methods. These algorithms have been developed for use in real-time systems.

Extracting data of the surrounding space on the optical flow.

Optical flow is the distribution of apparent velocities of the movement of brightness patterns in an image[7] [5].

There are several types of motion that can occur in the optical flow and can be processed accordingly. The optical flow applications vary from the detection of a moving object to determination of the multiple objects relative motion.

There are four general cases of a motion [6]: first,the camera is fixed, the object is moving, a constant background; second, the camera is fixed, multiple moving objects, a constant background; third, moving camera and relatively constant background; fourth, moving camera and multiple moving objects.

Algorithms for calculating of the robot position in space will use the data obtained from the processing of the third and fourth types of motion, and thus will detect movements of other objects in the stream in the first two cases.

Most algorithms choose a certain number of control points in the optical flow. Changing the position of these points in time indicates what motion is happening. The movement of these points are represented in motion. This is a two-dimensional array of two-dimensional vectors expressing the motion of points in three-dimensional scenes (figure 3).

In the first case is a distancing the camera from control points, as if the robot is moving back. Meanwhile the second case shows an approaching the camera to objects. Third case shows that the camera turns to the left because control points move to the right relatively to the direction of motion.

Human-guided robot movement through visual stimuli.

Preliminary experiments to explore the possibility of human-guided robot movement through visual stimuli are investigated. We used a base setup for our researches as displayed in (figure 4). The human is in front of the robot. The hand is interposed into the field of view of the robot. Then human behaves the movement. Movement in the perceivable area of robot (we restrict our motion in up-down directions only) is estimated by an optical flow algorithm. Velocity vectors (figure 5) are then converted into positive and negative robot's speed reference (figure 6).

188
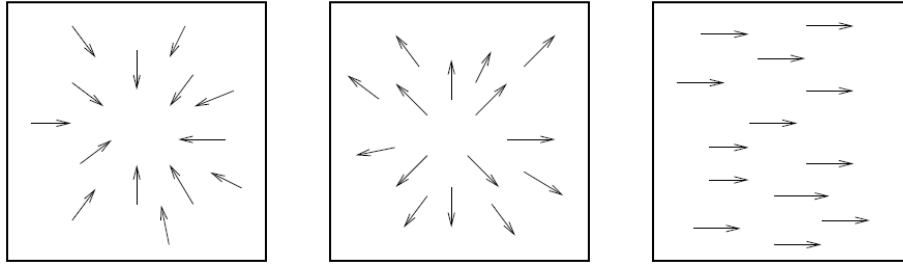
Figure 3. Variations of the field of movement.
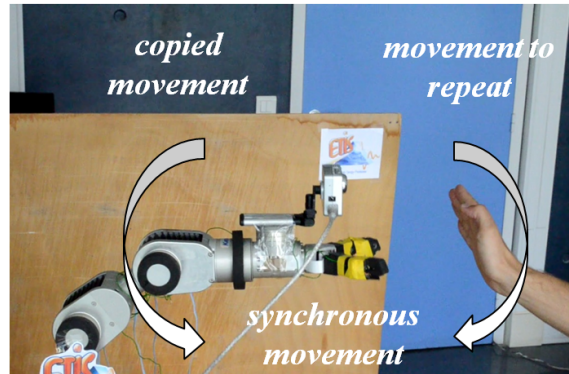
Figure 4. Experimental setup. Left: Robot arm Katana equipped by i-Fire camera. Right: human's arm.

During the experiments, the human produced different characters of movements: slow and fast movements with small amplitude end fast with large amplitude. All types of movements were converted into positive and negative robot speed references. For fast movements results are less successful due to speed limits of the robot arm Katana.

Conclusion

This paper presented preliminary experimental results in the human-guided robot's movement through its exteroceptive sensing abilities. The originality of this work lies in the usage on the link fixed camera which moves with it. This aim is achieved using the technology of computer vision.

References

[1] A. Nikitin, A. Melnyk, V. Khomenko. Video processing algorithms for robot manipulator visual serving // VI International practical conference Donbass-2020. Donetsk, DonNTU, 2012.

[2] A. de Rengerve, S. Boucenna, P. Andry, Ph. Gaussier, Emergent imitative behavior on a robotic arm based on visuo-motor associative memories, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference. — P. 1754 - 1759.

[3] A. de Rengerve, J. Hirel, P. Andry, M. Quoy, Ph. Gaussier, Continuous On-Line Learning and Planning in a Pick-and-Place Task Demonstrated Through Body Manipulation, IEEE International Conference on Development and Learning (ICDL) and Epigenetic Robotics (Epirob), Frankfurt am Main : Germany 2011. — P. 1-7.

[4] S. K. Hasnain, P. Gaussier and G. Mostafaoui. A synchrony based approach for human robot interaction. Paper accepted in Postgraduate Conference on Robotics and Development of Cognition (RobotDoC-PhD) organized as a satellite event of the 22nd International Conference on Artificial Neural Networks ICANN 2012. Lausanne, Switzerland, 10-12 September, 2012.
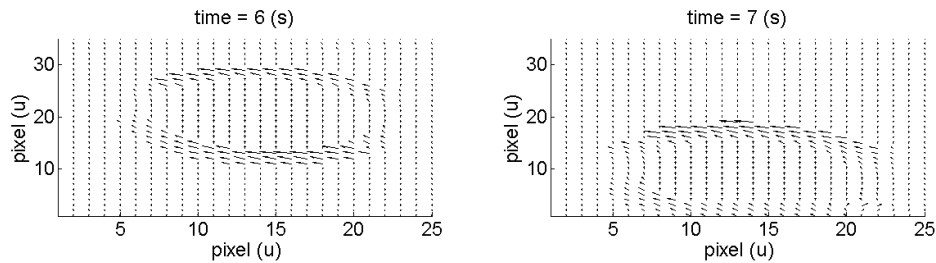
Figure 5. Flow pattern computed for simple translation of the brightness pattern.
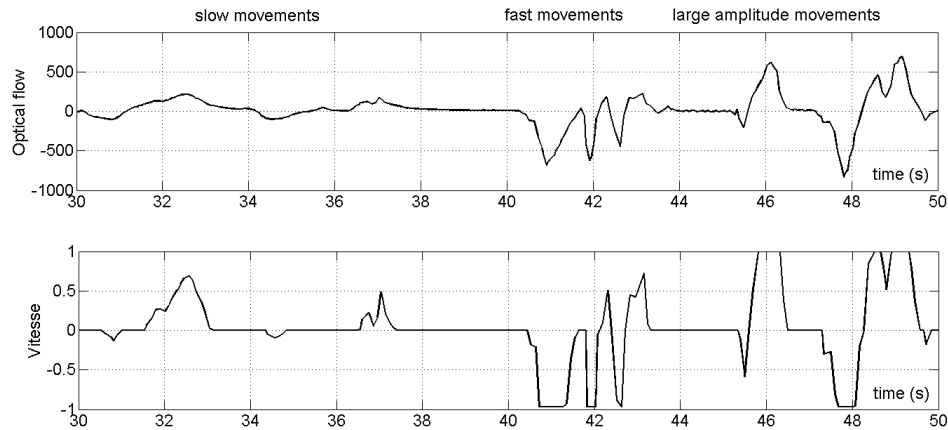
Figure 6. Value of the optical flow and angular speed or the robot link.

[5] Fundamentals of Robotics. Linking Perception to Action. H. Bunke, P. S. P. Wang. - New Jersey: World Scientific, 2004. — 718 p.

[6] L. Shapiro, G. Stockman. Computer Vision //The University of Washington, Seattle, - Washington: Robotic Science, 2000. — 609 p.

[7] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artificial Intelligence, vol. 17, no. 1-3, pp. 185-203.

Authors

Vladyslav Volodymyrovych Riabchenko — the 1st year master, ETIS laboratory of the University of Cergy-Pontoise, Cergy-Pontoise, France, Faculty of Computer Science and Technology, Donetsk National Technical University, Donetsk, Ukraine; E-mail: *riabchenko.vlad@gmail.com*

Artur Viacheslavovych Nikitin — the 2nd year master, ETIS laboratory of the University of Cergy-Pontoise, Cergy-Pontoise, France, Electrotechnical Faculty of the Donetsk National Technical University, Donetsk, Ukraine; E-mail: *nikitin.artour@gmail.com*

Viacheslav Mykolaiovych Khomenko — the 4th year PhD student, LISV laboratory of the Versailles Saint Quentin-en-Yvelines University, Versailles, France, Electrotechnical Faculty of the Donetsk National Technical University, Donetsk, Ukraine; E-mail: *khomenko@lisv.uvsq.fr*

Artem Anatoliiovych Melnyk — the 4th year PhD student, ETIS laboratory of the University of Cergy-Pontoise, Cergy-Pontoise, France, Electrotechnical Faculty of the Donetsk National Technical University, Donetsk, Ukraine; E-mail: *melnyk.artem@ensea.fr*