Autonomous Robots and Active Vision Systems: Issues on Architectures and Integration

Helder Araújo, Jorge Dias, Jorge Batista, Paulo Peixoto

Institute of Systems and Robotics/ Dep. of Elect. Eng.
University of Coimbra
3000 Coimbra, Portugal
email: helder@mercurio.uc.pt

Abstract. The use of active vision systems on mobile robots changes significantly the way vision can be used and integrated. An active vision system has the capability of performing motions that are independent of the platform's motion. In this paper we discuss several issues related with the integration of active vision systems on mobile robots. We claim that higher levels of autonomy and integration can be obtained by designing the system architecture based on the concept of *purposive behavior*.

1 Introduction

Autonomous or semi-autonomous operation of robots is required for remote and/or hazardous environments. For this kind of operation vision is obviously a very important sensor. Until a few years ago, whenever vision was used with mobile robots the main goal was to recover the 3D structure of the environment. According to this paradigm vision is a recovery problem being its goal the creation of an accurate 3D description of the scene (shape, location and other properties) which then would be given to other cognitive modules (such as planning or reasoning, see [4]). However no major breakthroughs have resulted from this approach. Systems based on this approach typically use one or two cameras rigidly attached to the mobile platform. The cameras move with the platform. This approach (general recovery) addresses the question of what range of mechanisms could exist in intelligent systems possessing visual capabilities. It does not address the question of how actual biological vision systems are designed as well as the question of what sort of vision systems would be desirable for particular classes of animals or robots. The "reconstructivist" approach addresses a problem which might not be directly related to the way biological or successful machine vision systems are designed. Biological vision systems are designed in many different ways. They have different needs, sizes and characteristics and, in general, they do different things.

Instead of trying to find general solutions for the vision modules we can consider the problem of vision in terms of an agent that sees and acts in its environment ([1], [2]). An agent can be defined as a set of intentions (or purposes) which translate into a set of behaviors [3]. The visual system can then be considered as a set of processes working in a cooperative manner to achieve various behaviors ([7], [8]). This is a new paradigm known as active/purposive vision. Within this framework we consider that the system is active because it has control over the image acquisition process and acquires images that are relevant for what it intends to do. The control over the image acquisition process enables the introduction of constraints that facilitate the extraction of information about the scene [2]. Therefore our goal when using the active vision system is not the construction of a general purpose description. The system only needs to recover partial information about the scene. The information to be extracted and its representation have to be determined from the tasks the system has to carry out (its purpose). Vision is considered as part of a complex system that interacts with the environment [5]. Since only part of the information contained in the images needs to be extracted, the visual system will operate based on a restricted set of behaviors (sets of perceptions and actions).

By considering vision within this framework, it can be used in real time to control a mobile platform. Also since complex representations of the world are not necessary, the visual system can interface and interact directly with all the low level subsystems enabling a high level of autonomy in terms information processing.

2 Autonomy

Autonomy is extremely important for the operation of a mobile robot. Here we will consider autonomy only in terms of the behaviors of the agents (mobile robots). We consider that higher levels of autonomy are more easily achievable when the architecture of the system is designed based on *purposive behaviors* as opposed to the more traditional approaches based on cognitive modules. With this kind of approaches behaviors are planned at a cognitive level, using, for example, search procedures. Architectures based on these design principles are hierarchical with modules such as perception, planning and navigation. As a result it is more difficult to obtain higher levels of autonomy because that type of modules imply the use of more complex representations and models with a resulting overhead on the required processing power. The use of complex representations implies the use of some kind of reasoning. Also the robustness of operation may be affected. On the contrary *purposive behaviors* imply the use of the minimal representations required to solve the task. By using *purposive behaviors* perception can be directly converted into actions. Purposive behaviors can be achieved by a number of different mechanisms. These mechanisms can be, for example [3]:

- Goal-achieving system: a system that recognizes the goal once it is arrived at.
- Goal-directed behavior: a system where the difference between the "desired" state and the actual state provides the error signal that actuates the behavior-control mechanism.

In designing one purposive behavior of a system the set of state variables can be divided into three different state spaces [3]: the environmental space, the behavior space and the task space. The environmental state space defines the topology and the laws of movement within which the agent can move. The task space includes the state variables that define the goals of the agent. The behavior space includes the variables to which the agent responds in its behavior and the variables that figure in the planning of its behavior. The architecture of the system directly reflects the relationship between behaviors and visual modules.

The advantage of the principles above described is that they can be applied both to the design of the visual system and of all the robot system architecture. The vision system can be considered as a collection of processes or modules which perform particular subtasks [8]. These modules are connected dynamically and can be connected to other sensory modules (e.g., odometry, sonars).

To describe agents we can use a state transition system. The formalism of Discrete Event Dynamic Systems [6] is one possibility to describe such a state transition system.

In order to show the advantages of application of these principles of architecture design we chose two examples: a complex active vision system and a mobile robot performing a very specific task.

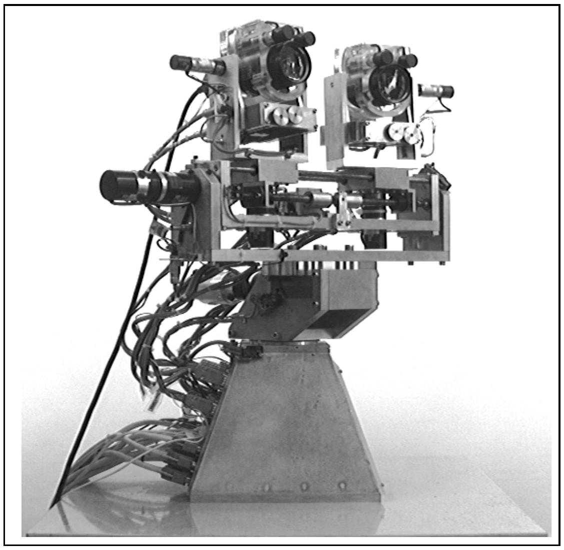

Fig. 1. ISR MDOF Active Vision Head

3 Example 1: A Complex Active Vision System

In order to experiment with visual behaviors and to study active vision issues (inspired by biological implementations and in particular by the human visual system) we decided to build a multi-degrees of freedom (MDOF) robot head [9].

We call it the ISR MDOF active vision head. Currently this is probably the head with the highest number of degrees of freedom. In addition to the common degrees of freedom for camera heads (pan, tilt and independent vergence for each of the eyes), this head includes the swing movement of the head neck, baseline control, cyclotorsion of the lenses and the ability of adjusting the optical center of the lenses. The mechanical structure of this head is quite similar to the KTH head structure (developed at the Royal Institute of Technology in Sweden [10]), and just like the KTH head, biological reasons played the main role in the design strategy. One important aspect in the design stage of these robotic systems is their performances. The analysis of some characteristics of the human active visual system can be useful for determining performance requirements for velocity and acceleration of a mechanical device that is aimed at simulating the human visual system behavior.

The ISR MDOF active vision robot head (see Fig. 1) has the following mechanical degrees of freedom:

Eyes-mechanical Each eye has three degrees of freedom (a total of six):
  - elevation (tilt)
  - azimuth (pan)
  - cyclotorsion (being developed)
  - an aditional degree of freedom is included to keep the optical center at the crosspoint of the azimuth and elevation axes of the lens.

Neck-mechanical The neck has three degrees of freedom:
  - tilt
  - pan
  - swing or lateral tilt movement

Baseline The ability of mechanicaly change the distance between the two eyes.

In addition to the common degrees of freedom for camera heads (pan, tilt and independent vergence for each of the eyes), this head includes the swing movement of the head neck, independent tilt movement for both eyes, and the ability of adjusting the optical center of the lenses. The latter is to ensure pure rotation when verging the cameras and compensate for the translation movement of the optical center when changing the focal length of the lens. Cyclotorsion of the eyes is at the moment being developed.

The ISR MDOF active vision robot head is connected to a pair of PC's being one dedicated to the control of the mechanical degrees of freedom and the other dedicated to the control of the optical degrees of freedom and image aquisition (see Fig. 2). This later PC is a Pentium (90Mhz) running as a *Master* being the other PC the *Slave* unit. These control units are connected between each other through an Ethernet link. A special protocol for commands exchange as been developed using the UDP/IP protocol. A modular multi-axis motion controller was used to control all degrees of freedom of the head. This modular system consists of a motherboard where up to six daughterboards or modules can be connected. On-board *Multitasking* executes up to 10 independent programs or background tasks simultaneously without interrupting motion control.

The command exchange protocol developed to connect the Master and Slave units uses three levels of protocol. At the lowest level the *command exchange manager* is responsible for sending and receiving packets of data through the UDP/IP protocol. The *command interpreter* establishes the connection between the lowest and the highest level of the protocol, analyzing the packets of data and deciding to which machine should the command be sent. This level is only active at the Master unit. At the higest level is the *command execute* that is responsible to send the commands to the board controller.

4 Example 2: A Mobile Robot Pursuing a Moving Object

To demonstrate the principles above described we decided to develop a complex system integrating an active vision system on a mobile robot [11]. This system was built to perform a well defined task: pursuing moving objects. Two main problems had to be dealt with: integration and cooperation between systems. This integration [13] had two distinct aspects: the interaction and cooperation between different control systems and the use of a common feedback information provided by the vision system. The solution developed was based on the visual *gaze holding* [12] process to establish a pursuit mechanism for *targets* moving in front of the mobile robot. The system is controlled to maintain constant the distance and the orientation of the robot and the vision system. The solution for this problem deals with the interaction of different control systems using visual feedback. It also addresses the real-time tracking of objects by using a vision system. This problem has been addressed in different fields such as surveillance, automated guidance systems and robotics in general [14].

4.1 Pursuit of Moving Objects

To perform the pursuit of a moving *target* we use two basic control schemes: a visual *fixation* control and the trajectory control of the robot. The visual *fixation* control guarantees that the *target* is continuously tracked by the vision system, and gives information about its position to the robot control. The robot control uses that information as a feedback to maintain the distance and orientation to the *target*.

The visual *fixation* control must be one visual process that runs in the active vision system and has capabilities to define a *target*, to concentrate the vision system on the *target* and follow it. A process with these characteristics has similarities with the visual gaze-shifting mechanism in humans. The gaze-shifting mechanism generates movements in the vision system to put a new object of interest in the center of the image and hold it there. The movement used to put the object in the center is called *saccade*, it is fast and it is performed by the two eyes simultaneously. If the *target* of interest is moving relative to the world, the vision system must perform movements to hold the *target* in the image center. These movements are composed by two types of movements called *smooth pursuit* and *vergence*. These movements are the consequence of the control performed by the process that we designate as *fixation*.

The *fixation* centers and holds the orientation of the vision system on a point in the environment. The principle is described graphically by the Fig. 3 where the mobile robot with an active vision system is concentrated on a person. *Fixation* gives a useful mechanism to maintain the relative orientation and translation between the referential in the vehicle and the *target* that is followed. This results from the advantages of the *fixation* process, where the selected *target* is always in the image center (foveal region in the mammals). This avoids the segmentation of all the image to select the *target* and allows the use of relative coordinate systems which simplifies the spatial description of the *target* itself (relationship between the observer reference system and the object reference system).

The pursuit process can be described graphically by the state diagram in Fig. 4. The *pursuit* process must be initiated before starting. During the initiation a *target* is chosen, the gaze must be shifted by a *saccade* movement and the vergence must be stabilized. In our system the *target* is chosen based in visual motion stimuli. The selection corresponds to a region in the images that generates a large visual motion in the two images.

If a *target* is detected, a *saccade* movement is initialized to put the *target* in the image center, and the system changes from the state *Rest* to *Vergence Stabilization*. During the *saccade* movement no visual information is processed. In the *Vergence Stabilization* state the system adjusts its *fixation* in the *target*. This is equivalent to establish the correct correspondence between the centers of the two images, defining a *fixation* point in the *target*. Since the vergence is stabilized, the system is maintained in the *Pursuit* state.

4.2 System Control and Architecture

The main hardware components of the system are the mobile robot and the active vision system. These two basic units are interconnected by a computer designated *Master Processing Unit*. This unit controls the movements of the active vision system, communicates with the robot's onboard computer and is interconnected with two other computers designated *Slave Processing Units*. These units are responsible for processing the images provided by the active vision system. The connections between different processing units are represented in the diagram shown in Fig. 5 and a photograph of the system is presented in Fig. 6.

The *Right* and the *Left Slave Processing Units* are computers with i486DX2 CPUs running at 66MHz. Each contains a DT-IRIS (50Hz) frame grabber connected to each one of the cameras. The *Slave Processing Units* process the images and communicate their results to the *Master Processing Unit* (anothe r computer with a i486DX2 CPU running at 66MHz). These communications use a 10MBits connection provided by ethernet boards (one board on each computer). The active vision system has two CDD monochromatic video cameras with motorized lenses (allowing the control of the iris, focus and zoom) and five stepper motors that confer an equal number of degrees of freedom to the system (vergence of each camera, baseline shifting, head tilt and neck pan). The *Master Processing*

Unit is responsible for the control of the degrees of freedom of the active vision system (using stepper motor controllers) and for the communication with the mobile platform (using a serial link). The actual control of the mobile platform is done by a multi-processor system based on a 68020 CPU, installed on the platform. The management and the interface with the system is done by a computer, connected to the *Master Processing Unit* using the serial link and a wireless modem.

A complete implementation of the methods necessary to obtain the desired behavior of the system requires fast processing capabilities for vision processing and control. The implementation used in this work was based in control loops working in parallel and based in the visual *pursuit* process. Since the inertia of the neck and the mobile robot are greater than the vergence mechanism inertia, this type of control simulates a control system with different levels - see Fig. 7. The inmost level comprises cameras and the vergence motors of the active vision system, responsible for tracking the *target* in real-time. This sub-system controls the cameras' position to maintain the visual system fixed in the *target.* At the intermediate level there is the neck sub-system, that provides the control of the orientation of the vision system and compensates for the cameras' vergence movements. At the outmost level is the mobile robot sub-system that provides the compensation for the orientation of the active vision system and also controls the orientation and distance to the *target.*

Conceptually, the error between the actual distance and orientation of the *target* and the system is propagated from the outmost level to the inmost level. This concept is graphically described in the Fig. 7. In each level the error is compensated for by the sub-system associated to each level. This error must be such that the maximum characteristic values of each sub-system are not exceeded. In the cases where the error exceeds these maximum values, the difference of error that can not be compensated for in that level is passed to the next in-most level. This scheme establishes a mechanism to propagate the error through the different control systems, giving more priority to the mobile robot, followed by the neck and eyes at the end. This gives the effect of compensating for the
it target's movements, simulating its *pursuit.*

The pursuit control loop consists basically of three stages: image acquisition, error estimation (the orientation and distance) and error correction. These steps are realized by the *Slave* and *Master Processing Units* at cycles synchronized by a general clock in the system. This clock has a $200msec$ cycle and the system's parameters are adjusted for that cycle. During this cycle the system performs different computations, depending on the state of the system. The different states of the system are illustrated by Fig. 4.

The images generated by each camera are adquired and analyzed by the *Slave Processing Units.* These units analyse the images and give the position of the *target* in each image. That position gives the necessary information to compute the system state. This state is passed to the (α-β-γ) tracker. The information provided by the *Slave Units* is delayed by one cycle of $200msec$. To avoid the lateral effects of this delay we use the prediction capabilities of the (α-β-γ) filter to estimate a value for the system state.

The *Master Processing Unit* repeatedly performs the control algorithm by using the error between the predicted system state and the desired system state. This error is passed to the different sub-systems. This error is passed to the different PID discrete time algorithms implemented in each subsystem. The results will be changes in the positions of the stepper motors associated with the vision system and the commands for the mobile robot to maintain the desired system state.

The movements executed by the mobile platform are based on two motors associated with each of the driving wheels (rear axle) and are essential to make the compensation for the error in the distance

5 Conclusions

In this paper we have shown that by using the concept of *purposive behavior* it is possible to implement real-time robotic systems performing useful tasks. The concept is essential for the design of the system architecture, if autonomy is a major design goal. *Autonomy* implies that only the system itself will take all the decisions required by the course of its operation. No external entity will interfere. In our case we not only achieved this type of autonomy, but also physical autonomy in the sense that all the processing power was on board. All the computers required for the operation of the system were on-board implying that the system was fully autonomous. The integration, the system architecture, the information processing modules, and the motor control processes were all designed taking into account the tasks and behavior of the system.

ACKNOWLEDGEMENT

This work has been partially supported by Junta Nacional de Investigação Científica e Tecnológica (JNICT) project VARMA.

References

1. Aloimonos, Y.: Purposive and qualitative active vision. *Proc. Image Understanding Workshop* (1990) 816–828
2. Aloimonos, Y., Weiss, Y., Bandopadhay, A.: Active Vision. *Intern. J. Comput. Vision* **7** (1988) 333–356
3. McFarland, D., Bosser, T.: *Intelligent Behavior in Animals and Robots.* MIT Press (1993)
4. Meystel, A.: *Autonomous Mobile Robots.* World Scientific (1991)
5. Aloimonos, Y.: What I Have Learned. *CVGIP: Image Understanding* **60** No.1 (1994) 74–85
6. Ho, Y.: Dynamics of Discrete Event Systems. *Proc. IEEE* **77** (1989) 3–7
7. Sloman, A.: On Designing a Visual System. *J. Exper. and Theor. Artif. Intell.* **1** (1989) 289–337
8. Horridge, G.: The evolution of visual processing and the construction of seeing systems. *Proc. Roy. Soc. London B* **230** (1987) 279–292

9. Batista, J., Dias, J., Araújo, H., Almeida, A. The ISR Multi-Degrees-of-Freedom Active Vision Robot Head: design and calibration. *M2VIP'95–Second International Conference on Mechatronics and Machine Vision in Pratice* Hong–Kong, September, (1995)
10. Pahlavan, K.: Active Robot Vision and Primary Ocular Processes. *PhD Thesis*, CVAP, KTH, (1993), Sweden.
11. Dias, J., Paredes, C., Fonseca, I., Batista, J., Araújo, H., Almeida, A.: Simulating Pursuit with Machines: Experiments with Robots and Artificial Vision. *Proc. IEEE Int. Conf. on Rob. and Auto.* 472–477 May 21–27, (1995), Nagoya, Japan
12. Brown,C.: Gaze controls with interaction and delays. *IEEE Transactions on Systems, Man and Cybernetics* **20**, May, (1990), 518–527.
13. Almeida, A., Araújo, H., Nunes, U., Dias, J.: Multi-Sensor Integration for Mobile Robot Navigation. In *Artificial Intelligence in Industrial Decision Making, Control and Automation* Spyros Tzafestas and Henk Verbruggen (Eds.), Kluwer Academic Publishers, (1995).
14. Burt,P., Bergen, J., Hingorani, R., Kolczynski, R., Lee, W., Leung, A., Lubin, J., Shvaytser, H.: Object tracking with a moving camera. *Proc. IEEE Workshop Visual Motion*, Irvine, 1989.

This article was processed using the LaTeX macro package with LLNCS style

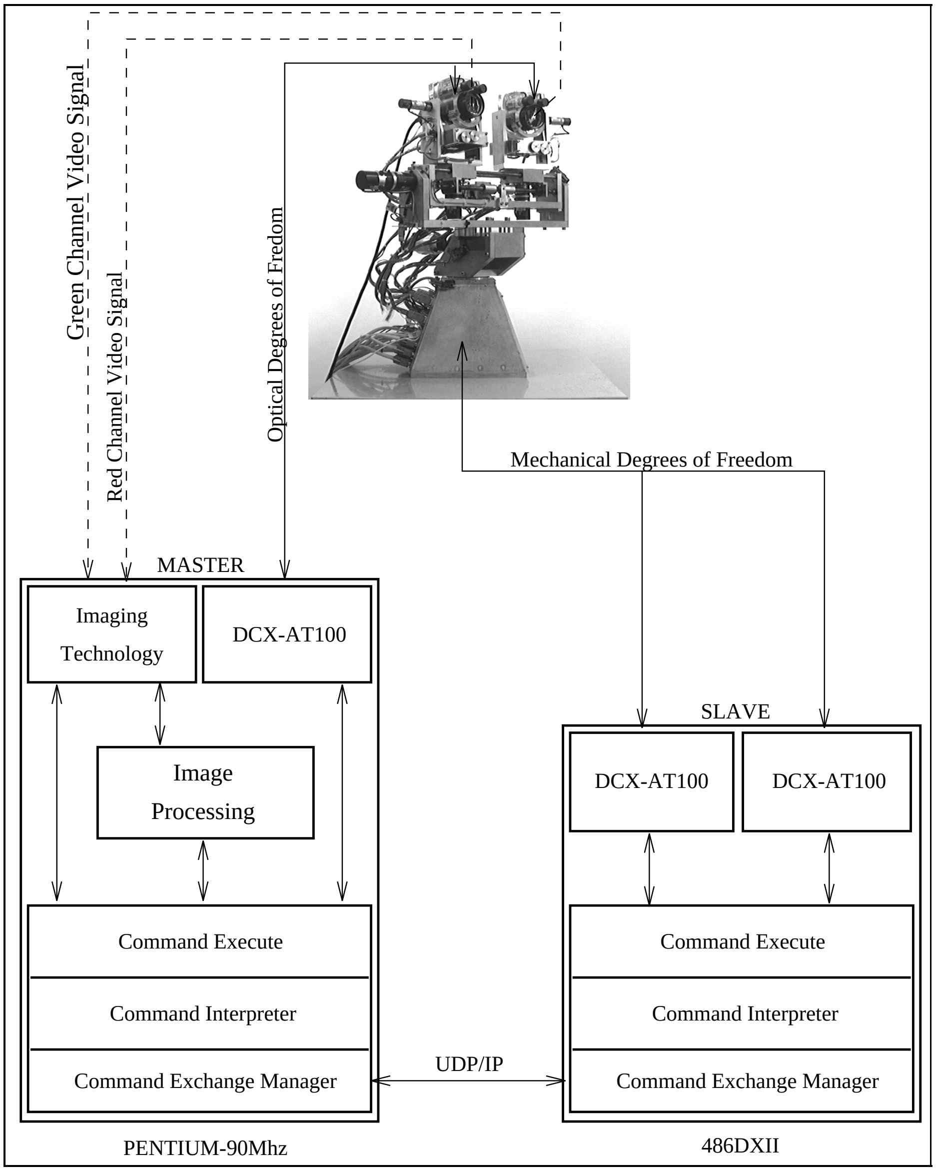

Fig. 2. The ISR MDOF system Architecture

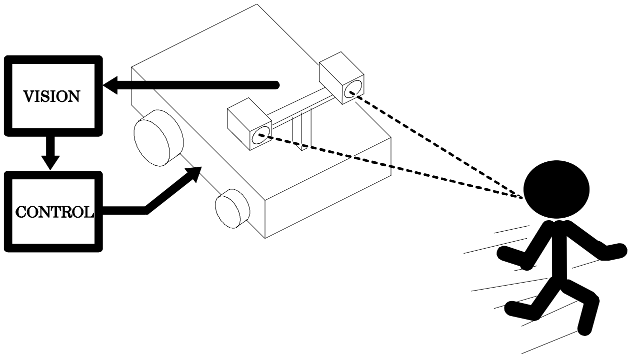

Fig. 3. The information provided by the *active vision system* is used to control the mobile robot for *pursuit* a target in $real - time$.

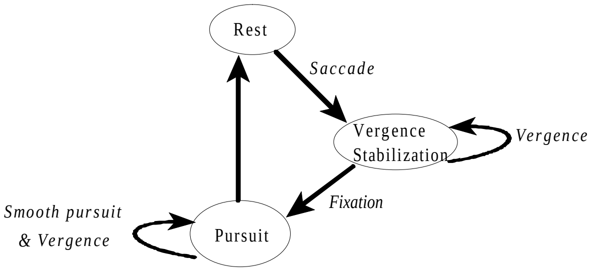

Fig. 4. State diagram of the *pursuit* process.

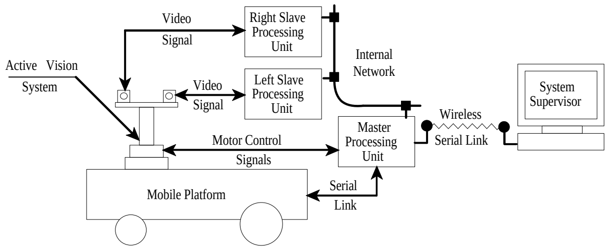

Fig. 5. System Architecture.

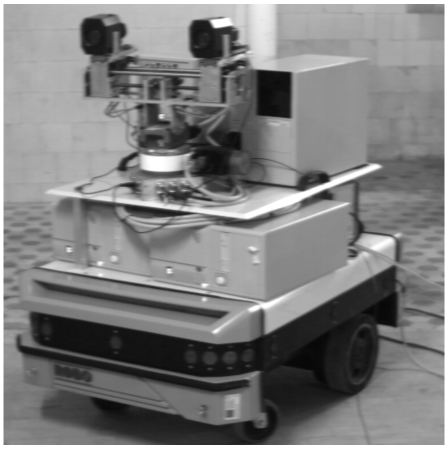

Fig. 6. The active vision system and the mobile robot.

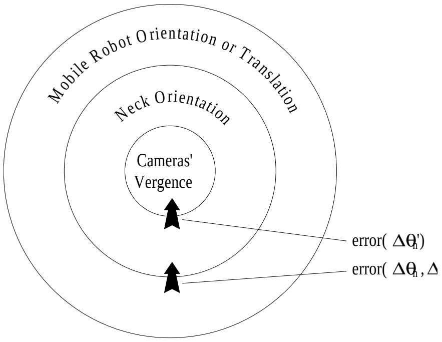

Fig.7. Graphic scheme of the principle used for the control of the system.