Active Motion Detection and Object Tracking

Joachim Denzler and Dietrich W. R. Paulus

denzler,paulus@informatik.uni-erlangen.de

The following paper was published in the
Proceedings on the 1^{st} International Conference
on
Image Processing

Austin, Texas, November $13^{th} - 16^{th}$, 1994
Volume 3

ACTIVE MOTION DETECTION AND OBJECT TRACKING

Joachim Denzler and Dietrich W. R. Paulus

Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, D–91058 Erlangen
Tel.: +49–9131–85-7894, FAX: +49–9131–303811
email: {denzler,paulus}@informatik.uni-erlangen.de

ABSTRACT

In this paper we describe a two stage active vision system for tracking of a moving object which is detected in an overview image of the scene; a close–up view is then taken by changing the frame grabber's parameters and by a positional change of the camera mounted on a robot's hand. With a combination of several simple and fast working vision modules, a robust system for object tracking is constructed. The main principle is the use of two stages for object tracking: one for the detection of motion and one for the tracking itself.

Errors in both stages can be detected in real time; then, the system switches back from the tracking to the motion detection stage.

Standard UNIX interprocess communication mechanism are used for the communication between control and vision modules. Object–oriented programming hides hardware details.

1. INTRODUCTION

Real time image processing and analysis is essential for robot control based on visual information. Since today's general purpose computers still lack the required transmission rates for continuous image input and processing, usually special hardware is applied for image processing using dedicated software or firmware for these devices. Detection and tracking of moving objects in the scene remains one of the basic problems to be solved. Various progress reports on this subject may be found in the literature e. g. in [1, 11].

In this article we demonstrate how standard Unix workstations and a general software architecture can be utilized for real time motion detection and object tracking in a closed loop of sensor and action. A wide–angle view of the scene is recorded to detect the moving object; the parameters of the frame grabbing device are changed to obtain a detail view in a higher resolution; by a change of the camera position the detail view is kept in the center. The key to success is adaptivity and selectivity of processing in space and time and the combination of several simple and fast vision modules for subtasks needed for object tracking [2]; this is commonly referred to as *active vision*.

For motion detection we present a fast algorithm based on difference images, obtained from a fixed (stationary) camera position. In the second stage, active contour models are applied for object tracking [6, 7, 11]. The snake is initialized automatically from the first image using a real time method presented in [3]. Actually, there is no proof in the literature for the robustness of real time tracking based on snakes in an non–synthetical environment. We present snake features which can be used for the detection of errors during the tracking. Thus tracking become more robust.

Two computers in parallel perform the image processing and robot control tasks. All processes can run on a single machine, alternatively.

In Sect. 2, we will give a short overview of the software and hardware architecture. Sect. 3 summarizes the principles of active vision. The two

stage object tracking will be described in Sect. 4, where we will present features, which have to be computed for detecting errors during the tracking. After an introduction of the ideas underlying the robot's control, we will show in Sect. 6 the robustness of our two stage approach during experiments in a quantitative manner. We conclude with an outlook on future extensions of the system.

2. SOFTWARE AND HARDWARE ARCHITECTURE

In our approach the general modular software architecture for knowledge based pattern analysis [8] is extended to to the active vision paradigm. An object–oriented implementation encapsulates hardware features of the used computers (HP 735) and video equipment (RasterOps Videolive card and Sony CCD RGB cameras); also, flexible segmentation is provided. Images are assumed to reside in main memory, i.e. not on the frame grabber device.

Visual tasks and robot control are separated into two communicating processes. Communication is based on the `pvm` interprocess communication library [5]. Only few bytes have to be exchanged per cycle. Since continuous image transfer from the frame grabber card to main memory uses up to 40% percent of the CPU power, experiments show better performance if control and vision are running in parallel on separate computers.

3. VISION MODULES IN ACTIVE VISION

Swain [10] gives a summary of active vision methods for image processing. Amongst active changes in the image capturing devices, selectivity of the algorithms *or* the hardware is mentioned.

In [2] the use of vision modules is specified. Various simple modules, each performing a single and dedicated task, communicate with each other. By integration of results from all vision modules, a more complicated vision task like object tracking may be executed.

Several modules are important for object tracking in a closed loop of sensor an action: The motion detection module, the tracking module itself, the module for the robot control and an attention module for detection of certain events during tracking (see Figure 1).

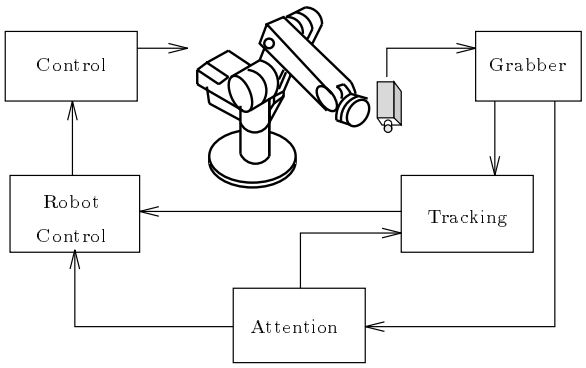

Figure 1: The principle of vision modules for object tracking

To reduce the amount of data which must be processed, one principle of active vision is the reduction of data by selectivity in space, time and resolution [10]. In the tracking stage selectivity should be done in space; in the attention module selectivity means looking at fixed time intervals at the whole image to recognize some events, like the entry of new moving object; this is selectivity in time. Finally, selectivity in resolution should be used by the motion detection stage.

4. TWO STAGE ACTIVE OBJECT TRACKING

We will now give a detailed description of our approach for object tracking using a connection of four simple vision modules: motion detection, motion tracking, attention module, and robot control.

At present, the first three modules are implemented in one single process. The second process, the robot control, will be described in the next section.

First one has to detect motion, i.e. the moving object in the scene; we assume a static camera, i. e. we keep the position of the robot fixed. A difference image between neighbouring images is computed. After a threshold operation we get a region of interest (ROI). Small gaps are closed and then the contour of the ROI is computed and

passed to the second module, the object tracking module.

For object tracking we use active contour models. In most other approaches the active contour is interactively initialized on the first image of an image sequence of course, this is impracticable in closed loop processing. Therefore we assume, that the ROI covers the moving object and make additionally make use of a property of non-rigid snakes: In the case of low or zero external energy – that means, no edges are near the snake – the snake will collapse to one point, and therefore it is sufficient to do a coarse initialization around the object's contour. The snake will collapse until it reaches the contour of the moving object.

Now, we use the contour of the ROI as an initialization for the active contour. Experiments show that this approach is sufficient for an automatic initialization of the snake. Errors may occur if there are strong background edges near the object, or if the ROI only partially covers the moving object. In combination with error detection by the attention module we get a robust initialization, which is proven by the results of the experiments described in Sect. 6.

An overview of the system for object tracking is given in Figure 2. A complete description may be found in [3, 4]. Motion detection is done on a low resolution (128×128) of the complete camera image. After initialization of the snake on the object's contour we switch to a higher resolution subimage containing the moving object.

The single steps can be computed very fast because of their simplicity. On the other hand, simplicity may be the source of errors. For example, if the ROI does not contain the moving object, the snake cannot extract it or the snake may loose the object during tracking. Thus, another important aspect is the detection of errors during object tracking. The attention module, or any other module in the vision system, should thus be able to decide, whether tracking of a moving object is no longer possible; for example, if the object will be occluded by background objects, or the tracking module has lost the object. Then it has to stop tracking, send a message to the motion detection module (stage 1) and wait for a result to start tracking again. This can be seen by the two paths back from stage 2 to stage 1 in Figure 2.

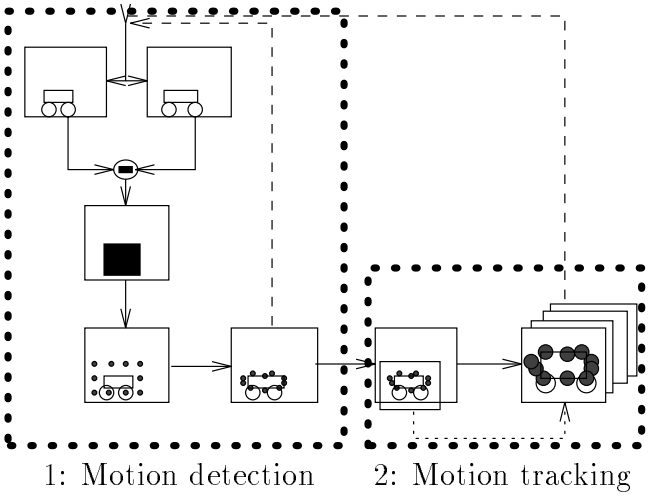

Figure 2: Overview of the two stages of the object tracking. Left side: stage 1 (motion detection). Right side: stage 2 (motion tracking).

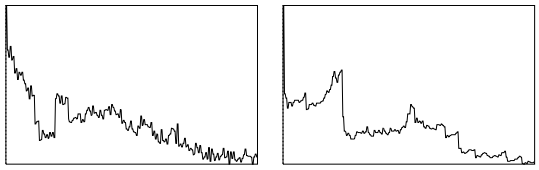

Figure 3: Features for detection of errors during tracking: the x- and y-moment of the active contour.

Typically, three main errors ocurr: The snake collapses to one point, if there is no edge near the snake [7], the snake extracts a static background object, or the fixation of some snake elements on strong background edges near the object (see Figure 3). To detect such errors the following features are extracted from the active contour: correlation, regression, moments and motion of the the snake. In Figure 3 the plots of the x-moment and y-moment of the active contour are shown during a sequence of images. The reason for the rapid change of the x- and y-moments is that three snake elements extract the rail instead of the train itself for 10 images. A more detailed discussion of the feature based detection of errors may be found in [4].

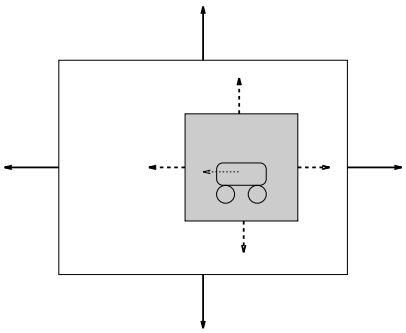

Figure 4: Two methods used for changing the view on the scene: dashed lines by changing the parameters of the grabbing device; solid lines by the motion of the robot. Only the grey area is digitized on the board.

5. CLOSED LOOP ROBOT CONTROL

The fourth part of our closed loop object tracking system is the module for robot control. There exist many approaches in the literature for robot control [9]. Since in a closed loop system the slowest module constraints the overall system performance, robot control should be kept as simple as possible. Therefore we use only four signals for each direction, in which the robot can move: `UP` or `DOWN` and `FASTER` or `SLOWER`. To cause a smooth robot motion we use two different methods for changing the grabbed image. First, we move the robot itself. Second, we change the parameters of the grabbing device such, that the tracked object is kept always in the middle of the subimage (see Figure 4). The motion, indicated by the dashed line is done by changing the area, which the grabbing device will digitize. The motion of the whole image is indicated by the solid lines and is achieved by the robot's motion.

6. EXPERIMENTS

Various experiments show the stability and speed of our system. Presently, tracking of a moving toy locomotive with an approximate speed of 2 cm/sec is possible in real time. The distance between the object and the camera is approximately 1.5 m.

Figure 5 shows the subimage captured by the robot's camera (compare Figure 4) in the first row. The corresponding snakes are printed below. The third row gives an overview of the whole experiment from a background camera.

In order to give quantitative results we have tested and evaluated the system with five test runs, which is equivalent to 3000 images and to a time period of 14 minutes. Three times the object was lost and the system had to switch back to the motion detection stage. 66 % of all initializations of the snake were sufficient for the follwing tracking. Judging the quality of the tracking algorithm, we have measured the distance of the object's center of gravity (COG) from the middle of the image. We define, that the object is in the center of the image, i.e. the active contour has accurately extracted the object, if the this distance is less than 20 pixels. Using this criterion, in 83 % the object is in the middle of the image. Presently, no prediction is done for the position of the object in the next image. Using a prediction step the performance of the system should be improved [11].

7. CONCLUSION

The motion detection and tracking module is allowed to make mistakes, because the detection of such mistakes is possible in real time and so the error can be fixed. Thus the system operates fault tolerant. A simple and fast initialization of the snakes can therefore be used. The results prove, that the combination of simple vision modules results in a very fast and robust system. In future work, we will extend the tracking stage by a prediction module. Robustness of the attention module and an extension of the automatic initialization are subject to further improvements. Furthermore, the robot control has to be extended to allow the investigation of the robot's environment. This will result in an active detection of moving objects by a closed loop saccade control.

8. REFERENCES

[1] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Trajectory filtering and prediction for automated tracking and grasping of a moving object. In *Image Understanding Workshop*, pages 1019–1033, San Diego, 1993.

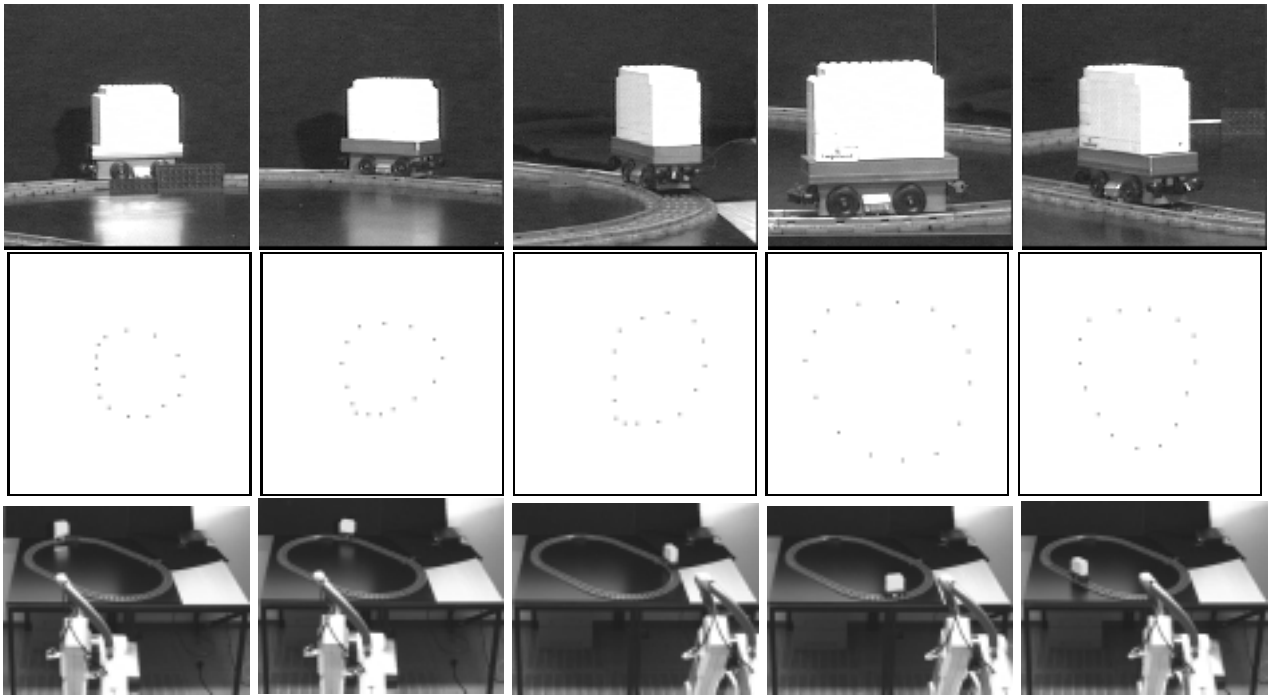

Figure 5: Images 13, 84, 207, 329, 424 of one experiment. First row: The subimage containing the tracked object. Second row: the corresponding snakes. Third row: overview of the whole experiment.

[2] J. Aloimonos and D. Shulman. *Integration of Visual Modules*. Academic Press, Boston, San Diego, New York, 1989.

[3] J. Denzler, R. Beß J. Hornegger, H. Niemann, and D. Paulus. Learning, tracking and recognition of 3D objects. In V. Graefe, editor, *International Conference on Intelligent Robots and Systems – Advanced Robotic Systems and Real World*, to appear September 1994.

[4] J. Denzler and H. Niemann. A two-stage real time object tracking system. In *German-Slowenia Workshop on Image Processing*, to appear 1994.

[5] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM 3.0 user's guide and reference manual. Technical Report ORNL/TM-12187, Engeneering Physics and Mathemantics Division, Oak Ridge National Laboratory, Tennessee, 1993.

[6] M. Kass, A. Wittkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 2(3):321–331, 1988.

[7] F. Leymarie and M.D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634, 1993.

[8] H. Niemann. *Pattern Analysis and Understanding*. Springer, Berlin, 1990.

[9] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1):14–34, 1993.

[10] M.J. Swain and M. Stricker. Promising directions in active vision. Technical Report CS 91-27, University of Chicago, 1991.

[11] D. Terzopoulos and R. Szeliski. Tracking with kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. MIT Press, 1992.