Steroscopic Camera for Autonomous Mini-Robots Applied in KheperaSot League

Teerapat Chinapirom, Ulf Witkowski, and Ulrich Rückert
System and Circuit Technology
Heinz Nixdorf Institute, University of Paderborn,
Fuerstenallee 11, 33102 Paderborn, Germany
{chinapir,witkowski}@hni.upb.de

Abstract

This paper presents a stereoscopic vision system for the mini-robot Khepera. The vision system performs objects detection by using the stereo disparity and stereo correspondence. The stereoscopic vision system enhances robot's visual perception ability by grabbing stereo images and analysis 3D objects, while the robot doesn't need to move. The simple principle of our stereo vision is the less displacement of correspondence pixels shows that the pixels object is far away. To realize the stereo vision and its calculation algorithms, the mini robot needs a powerful FPGA and micro-controller module as well as 2D color cameras. An application of Khepera equipped with the stereoscopic camera is robot soccer in the KheperaSot league. In the match, the robot has to be able to detect its environment, i.e. the ball, walls, goals and its opponent.

1. Introduction

In the KheperaSot league [1] of FIRA Robot World Cup Soccer Tournament, the linear camera turret for Khepera [2] is generally used for the vision perception, which provides fairly poor image data with 64 gray level pixels. This data is used to distinguish the three intensity levels (three objects): black goal, gray wall and yellow tennis ball. For identification of opponent, the robot detects opponent's black and white striped dress. The robots compete in the soccer pitch dimension 1050 mm × 680 mm, illustrated in Figure 1. The vision perception of the linear camera can sometimes fail because of wrong color/brightness-tone, lighting spread and disturbance by shadows. Using of the 2D-color camera system can reduce these problems, and lead to reliable object detection [3].

To provide the 3D perception ability of the mini-robot, we have developed the stereoscopic camera module. The module can be used for various stereoscopic image processing applications, for example the distance estimations of 3D objects, the obstacles detection for navigation and the visual reconstruction approach of unknown environment.

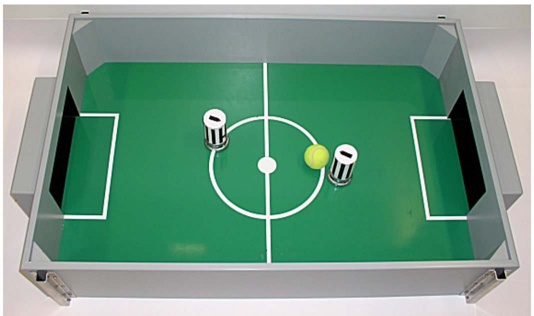


Figure 1: The KheperaSot competition with two robots, one in each team

Now, many robots are equipped with a camera for 2D image perception. The most robots use their camera for grabbing a snap shot of their environment. Only some robots are able to detect the depth dimension either with single camera by using optical flow [4]; or with stereo camera by using epipolar constraint [5].

The single camera solution can provide the depth image information, only while the robot or the observing object is moving. In contrast, the stereoscopic camera solution can provide the third dimension by grabbing simultaneously the left and right cameras, like as humans' binocular vision. To realize the stereoscopic vision algorithms in a mobile embedded system for real time applications, requires generally a very fast processor or many processors for acquisition and computation of the huge amount of image information. Some research groups therefore have decided to use FPGAs in their embedded systems for image processing approaches, because of some advantages in the parallelism performance and the flexibilities.

Our approach is to enable the 3D vision perception to the mobile autonomous mini-robot by using the stereo camera. In this paper the mini-robot, equipped with our stereoscopic vision module and the information processing FPGA module has to capture stereoscopic images and then evaluate the disparities of corresponding points in left and right images. A simple environment platform for the experiments the vision module and testing some stereo image processing algorithms is the KheperaSot pitch.

The paper is organized as follows: The next section describes some essential notation and theory of stereopsis, including the transformation of 3D objects to the image plane; the epipolar geometry and some disparity and correspondence analysis. Section 3 presents the hardware platform of stereo vision system of Khepera for our implementation. Some experiments and implementation details are discussed in section 4. Finally, in section 5 the paper conclusion is presented.

2. Stereoscopic vision

Two cameras take pictures of a same object from different location. The 2D images on the plane of projection represent the object from camera view. These two images contain some encrypted information, e.g. the image-depth of each other. This information is the third dimension of 2D images. Therefore the object distance and its depth can be determined by using the two cameras (often called stereo camera)

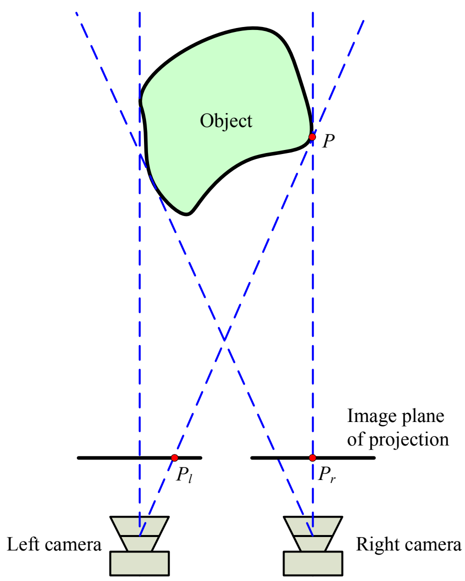


Figure 2: the positions of two cameras and their image planes of projection

The principle of computer stereo vision is that the left and right cameras see the scene from different positions (side by side), which are similar to human's eyes. The computer then compares the images while shifting the two images together over top of each other to find the parts that match or are similar. The shifted amount is called "*disparity*", which relates to the object distance. The higher disparity of object pixel means that the object is closer to the cameras. The less disparity means the object is far from the cameras. And, if the object is very far away, the disparity is zero that means the object on the left images is the same pixel location on the right image.

Figure 2 depicts the geometrical basis for stereoscopic image by using two identical cameras. These cameras are set up on the same plane and turned in the same direction, known as *parallax sight*. The position of the both cameras is different in *X* axis. The image planes are presented in front of the cameras for ease to model the projection (like as the model of pinhole camera).

Consider the point (P) on the object, whose perspective projections on the image planes locate at P_l and P_r from left and right cameras respectively. These perspective projections are constructed by drawing straight lines from the point to the center lens of the left and right cameras. The intersection of the line and image plane is the projection point. The left camera's projection point (P_l) is shift from the center, while the right camera's projection point (P_r) is at center. This shift of the corresponding point on left and right camera can be computed to get the depth information of the object.

2.1. Statement of the stereo problems

The main purpose of stereoscopic vision is to recover the depth of stereo images, so the problems are how to find out the depth and which models of constraints can solve the problems. There are two main steps to solve the stereo problems: First step is to determine corresponding points in the images. The second step is to compute the depth from the corresponding points, which is easier than the first step.

To find out the correspondence, some stereo image constraint must be assumed before, as following:

- *Uniqueness*: Each point has at most one match in the other image
- *Similarity*: Each intensity /color area matches a similar intensity /color area in the other image
- *Ordering*: The order of points in two images is usually same.
- *Continuity*: Disparity changes vary slowly across a surface, except at depth edges
- *Epipolar constraint*: Given a point in the image of one eye, the matching point in the image for the other eye must lie along a single line

Some of the constraints are strong conditions, which must be always considered. While some of the constraints are weak conditions, which may be considered depending on the cases. The methods for matching the correspondence of stereo images can be classified in two groups:

1. Area based matching: e.g. intensity and color of the area block
2. Feature based matching: e.g. location, strength and orientation of the edges

To evaluate of the area based and feature base matching some well known image processing algorithms are needed, for example block matching by using sum absolute difference (SAD) or sum square difference (SSD), edge detection by using sobel filter. The constrains of correspondence for stereoscopic vision are mostly related with epipolar geometry

2.2. Epipolar geometry

The epipolar geometry of stereoscopic camera is illustrated in Figure3. This simple stereo model shows two different perspective views of an object point (*P*) from two identical cameras centers (F_l and F_r.), which separate only in *x* direction by a baseline distance. The point P_l and P_r in the image plane are the perspective projections of P in left and right view, which are called a *conjugate pair*. The plane passing through the camera centers and the object point in the scene is called the *epipolar plane*. The intersection of the epipolar plane with the image plane is called *epiolar line.* By referring the epipolar geometry, correspondences at point P_l and P_r must lie on the epipolar line

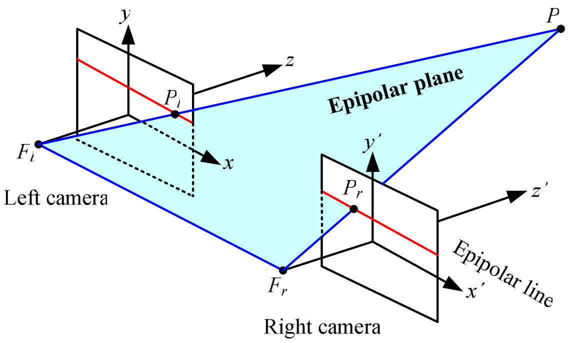


Figure 3: The epipolar geometry of stereoscopic vision

The epipolar geometry is determined by the sufficient point correspondences. Selection and matching of the point features in the two views are the standard procedure for the depth recovery. The depth information can be evaluated by using the triangle similarity algorithms.

The epipolar geometry can be represented in the top view as shown in Figure 4. In the illustration, the baseline distance (*T*) and the focus length (*f*) of both cameras are known. The perspective projection P_l and P_r on epipolar line is shift from their center by distance x_l and x_r respectively. The distance from the object to the base line (*Z*) can be determined by comparing the similar triangles PF_lF_r and PP_lP_r. The following equations presenting the object's distance evaluation:

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

then
$$Z = f\frac{T}{x_r - x_l} = f\frac{T}{d}$$

where: disparity is $d = x_r - x_l$

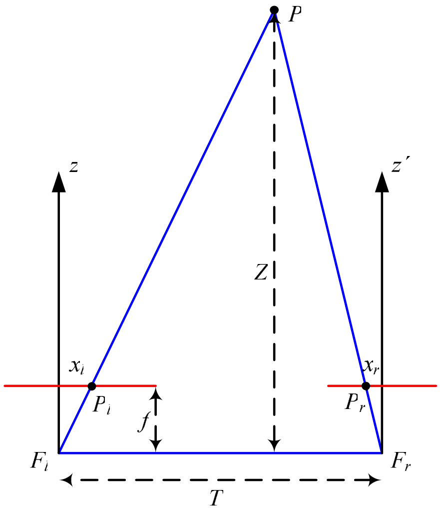


Figure 4: The disparity is the displacement between the locations of the two projection points on image plane

From the epiplar geometry, the projection point can be computed for each view and matched image points, which then can be backprojected to give 3D structure.

2.3. Correspondence by block matching

To evaluate photometric correspondence of stereo images, there are some simple standard algorithms by using block matching and matching criteria, such as SAD, SSD and cross correlation [6]. The blocks are usually defined on epipolar line for matching ease. Each block from the left image is matched into a block in the right image by shifting the left block over the searching area of pixels in right image as shown in Figure 5.

At each shift, the sum of comparing parameter e.g. intensity or color of the two blocks is computed and saved. The sum parameter is called *"match strength"*. The shift which gives a best result of the matching criteria is considered as the best match or correspondence. The block matching of stereo images is responsive to brightness, noise of the left and right cameras.

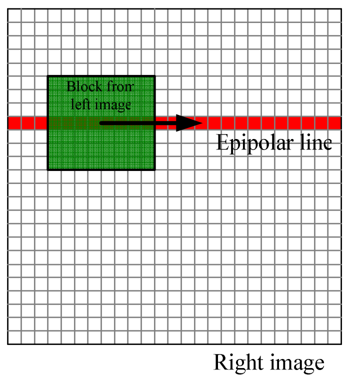


Figure 5: The block matching algorithm, computing each point of the left image block for every position through the corresponding epipolar line in the right image

The following equations are often implemented for block matching algorithm:

- Sum absolute difference (SAD) equation

$$SAD(p_l, p_r) = \sum_{j=-n}^{n} \sum_{i=-m}^{m} | I_l(x_l + i, y_l + j) - I_r(x_r + i, y_r + j) |$$

- Sum square difference (SSD) equation

$$SSD(p_l, p_r) = \sum_{j=-n}^{n} \sum_{i=-m}^{m} (I_l(x_l + i, y_l + j) - I_r(x_r + i, y_r + j))^2$$

- Cross correlation equation

$$CORR(p_l, p_r) = \sum_{j=-n}^{n} \sum_{i=-m}^{m} I_l(x_l + i, y_l + j) * I_r(x_r + i, y_r + j)$$

where: I_l, I_r are the intensity of pixel in left – right image.
p_l, p_r are the comparing point *(x, y)* of image

The minimum value of SAD and SSD is considered the best match. While the maximum value of cross correlation is considered the best match

3. Hardware platform for implementation

The autonomous mini-robot for the implementation of stereoscopic vision is Khepera, which is widely used in research and edutainment. The robot's base module possesses the processing platform from a Motorola 68331 microcontroller, running with 25MHz clock frequency, 512 kB RAM and 512 kB Flash. The base has two independently driven wheels with incremental encoders for the odometry, For close distance and ambient-light measurement Khepara use 8 integrated IR sensors. The additional battery module provides extension energy to the mini-robot for the long-run experiments or the robot soccer game. Khepera can be equipped with the additional turret (illustrated in Figure 6), for example, the stereo vision module for capturing stereoscopic image, the FPGA module for running real-time or parallel tasks such as the image processing task.

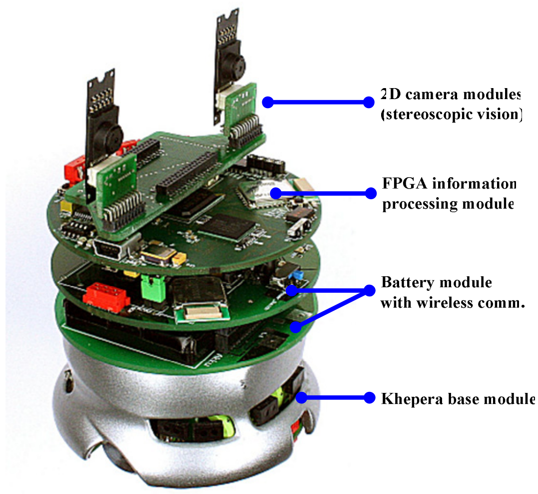
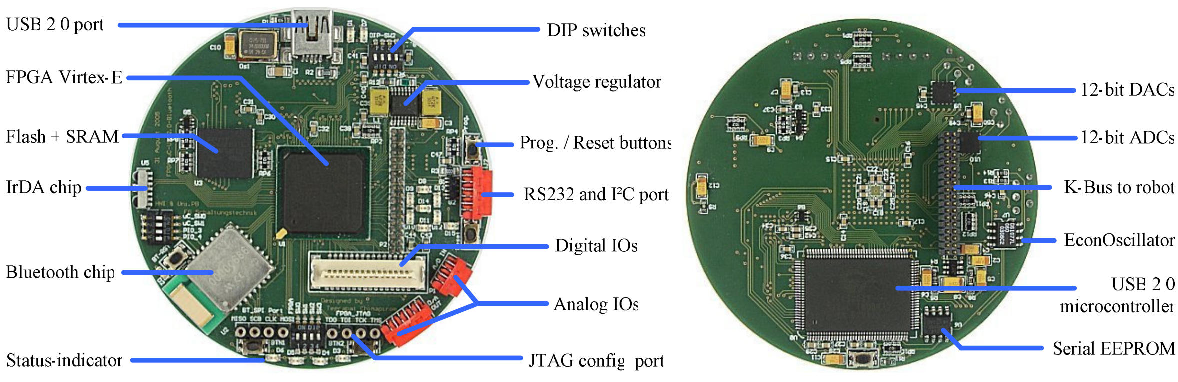


Figure 6: The mini-robot Khepara equipped with the stereo camera and the FPGA-MCU module for stereoscopic vision application

Our stereoscopic camera module uses two miniature 2D color CMOS image sensors TC5740MB24B from TransChip Inc., which have integrated color processing and JPEG codec. The camera can provide VGA resolution color image (max. 648×488 pixel), 10 bit/pixel and up to 40 fps (frames per second) at QVGA format. The integrated image processor enhances the flexibility of the camera for image format (e.g. RAW, YUV, RGB) and automat setting (e.g. exposure control, white balance and digital zoom function). The power consumption of the camera is low about 85 mW at 2.8 V.

Due to the robot's base processing module runs the game strategy task and motor controlling for navigation, the processing module isn't sufficient to run 2D image acquisition and image processing. To solve this problem of processing performance, we have designed a co-processor universal FPGA-microcontroller module [7] for Khepera, illustrated in Figure 7. This module extends the robot's processing performance and provides the communication option for robot to host and inter-robot communication in cooperative robotics.



Top side **Bottom side**

Figure 7: The top and bottom view of the universal FPGA – MCU module for the mini robot

The universal FPGA Microcontroller module consists of the main following components:

- FPGA Virtex-E XCV300E-6 from Xilinx with 9612 logic blocks, and flexible I/Os, will be responsible for image acquisition and concurrently processing of image data to transmit symbolic data to the mini-robot's base. The FPGA is reconfigured to change its tasks during run-time and also supports partial reconfiguration.
- Cypress EZ-USB FX2 USB microcontroller, which provides fast serial communication USB2.0 interface and flexible programmable processor 8051 core with adequate peripheral interfaces and I/Os; responsible for FPGA reconfiguration and downloading of FPGA configuration file.
- Multichip memory package contains 128MBit simultaneous Flash and 32 MBit pseudo SRAM, which has sufficient capacity to store up to 32 full bitstream files.
- Dual 12 bits channel ADC and Dual 12 bits channel DAC are the fast converters with ultra low power consumption using for sound applications, analog signal acquisition applications and analog control system (e.g. the back light of LCD).
- Bluetooth WML-C19 module from Mitsumi contains BlueCore2-External [8], integrated with a 16bit RISC microcontroller, serial peripheral interface and IOs. Ac-hoc, Piconets and point-to-point connections are supported by our firmware.

4. Experiments and implementation

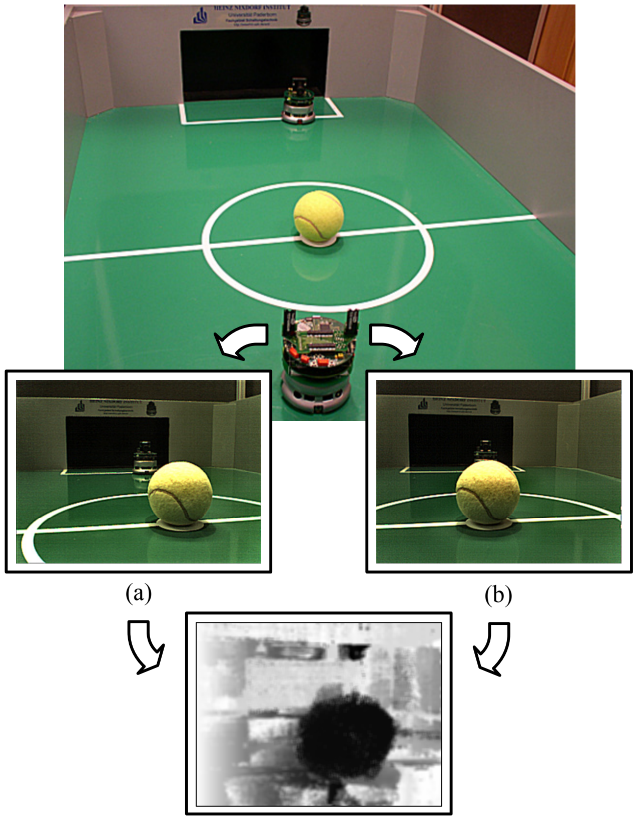
Before the implementation of the stereoscopic vision on the Khepera mini-robot, we have simulated our stereo processing algorithms on PC with Matlab software to optimize some setting parameter and see the visual image result. In the implementation on the robot we used a simple and fast algorithm to detect the ball, the goal and the opponent.

4.1. Simulation with Matlab

Our stereoscopic vision module can send the real time stream image to PC via USB 2.0 port. This provides the raw image data to save on PC for the simulation. The raw data is the same data that robot percept at run time. Therefore, what the simulated result comes out from Matlab simulation will be the identical result from the robot. By using the simulation we can see the visual output image and know the result tendency, when we change some parameters.

The stereoscopic images sized 640 x 480 pixels from the left and right camera, which are shown in Figure 8 (a) and 8 (b) respectively, are reduced the size to 320 x 240 pixels. The size of the matching block is set at 8 x 8 pixels, and the searching area is 80 x 80 pixels around the matching block. We used the intensity and color block matching algorithm. The color transformation of RGB to grey and RGB to HSV are available in Matlab library. The SAD block matching is used in this simulation because of ease to implementation on the mini – robot.

The simulated image output is shown in Figure 8 (c). The ball can be detected as well as the goal. The opponent is hardly detected because it is hidden behind the ball in right image. However the head of opponent, which is higher than the ball, can be detected. In the image the black color shows disparity of the ball. The light gray rectangle form, behind the ball locates the goal. We can see some parts in the image are not clear, such as the circle line and the floor, because of the reflections on the ground.



(a) (b) (c)

Figure 8: Khepera equipped with stereoscopic camera on the soccer pitch, (a) the scene from left camera, (b) the scene from right camera, (c) the simulated image by using the SAD block matching algorithm using Matlab

4.2. Implementation on Khepera

For implementation of stereoscopic vision for the object detection approach on Khepera we used following steps: First, finding the interesting objects from their features in the left and right images (e.g yellow color means ball, black color means goal), then computing the disparity of the interesting object. There are three objects to be found (goal, ball and opponent) in this implementation. The RGB- format image sized 320 x 240 pixels is captured by each camera. The raw image data is saved in onboard SRAM and only some necessary data sections will be loaded into internal Block-RAM of the FPGA for faster access time while computing the interesting line

Goal detection

From practical test we know that the horizontal 60th line form top can be epipolar line (the red line illustrated in Figure 9) used to detect the goal. First, the both edges of the goal in the right image must be detected on the epipolar line. The matching blocks sized 8 x 8 pixels at these edges are defined and then compared in the left image on the epipolar line by using SAD block matching. The minimum intensity difference of SAD shows the best match. The goal in right image locates between (42, 60) and (177, 60). The disparity results are 23 pixels for the left goal edge and 22 pixels for the right goal edge

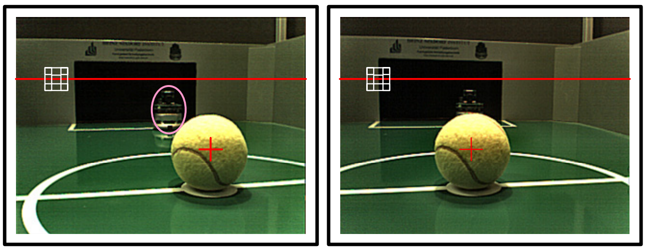


Figure 9: the illustration of the implementation algorithm for detection of the ball, the goal and the opponent robot

Ball detection

We used the yellow color feature to determine the ball location. The image procedure begins with the color conversion (RGB → HSV). We then determine the yellow region. The center point of the ball in the both views can be also evaluated by finding the biggest yellow strip in low and column. We have implemented this ball detection [3]. The method is fast, accurate and robust. We got the center point of the ball from left view and right view at point (215, 146) and (146, 148) respectively shown in Figure 9. The ball disparity in the stereoscopic images is [-70, 2].

Opponent detection

To locate the opponent robot, the algorithm is to find the black and white strip in the horizontal 80th line defined to epipolar line. The SAD block matching algorithm is implemented as same as the goal detection. The matching block is located at the center of robot and compared in the left image. The location of the opponent in the right image is at (143, 80), the disparity of the point is 26 pixels.

For the disparity result of these three objects in soccer pitch, the robot can knows their position and distance. The ball is the nearest object because of the highest disparity. The goal is the farthest object with lowest disparity.

5. Conclusions

The stereoscopic vision principle and some theoretical background (e.g. epipolar geometry, disparity, stereo image constrain and block matching algorithms) have be presented in this paper. Two main steps for depth recovery of stereoscopic image are searching the correspondences in the images; and computing the depth from the corresponding points.

We have realized the hardware platform of the stereoscopic vision for Khepera mini-robot. The simulation with Matlab can prove our image processing algorithms before we implement the right algorithm into the hardware platform. The experiments in a simple environment, such as KheperaSot soccer pitch, shows that our stereoscopic vision module enables the robot to perceive stereo image. The FPGA-MCU module performs dual camera controller for the simultaneous image acquisition, and then computes the image data. In the experiment the mini-robot can locate the three interesting objects (e.g. the goal, the ball and the opponent) and evaluate accurately their disparities, so that the robot can estimate the distance to these object.

6. References

[1] FIRA Executive Committee, FIRA Khepera Game Rules, www.fira.net/soccer/kheperasot/KhepearSot.pdf.

[2] K-Team S.A., Mobile Minirobots Khepera II, Lausanne, Switzerland, www.k-team.com, 2007.

[3] T. Chinapirom, T. Kaulmann, U. Witkowski and U. Rückert, Visual Object Recognition by 2D-Color Camera and On-Board Information Processing for Minirobots, *Proceedings of the FIRA Robot World Congress*, Busan, Korea

[4] D. H Ballard, C. M Brown, *Computer Vision*, Optical Flow, Prentice-Hall, 1982.

[5] D. A. Forsyth, J. Ponce, *Computer Vision A Modern Approach,* Stereopsis, Prentice Hall Inc., 2003.

[6] E. Trucco, A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, Inc., 1998.

[7] T. Chinapirom, U. Witkowski and U. R¨uckert, Universal FPGA-Microcontroller-Module for Autonomous Minirobots, *Autonomous Minirobots for Research and Edutainment (AMiRA 2005)*, Fukui, Japan, September 2005.

[8] Cambridge Silicon Radio (CSR), *BlueCore2 External Product Data Book*, www.csrsupport.com, 2004.