S & M 1547

Visual Servo Strategy for Robot Soccer Systems

Ming-Yuan Shieh*

Department of Electrical Engineering, Southern Taiwan University of Science and Technology,
No. 1, Nan-Tai St., Yung-Kang Dist., Tainan City, 71005, Taiwan (R.O.C.)

(Received October 31, 2017; accepted April 2, 2018)

Keywords: visual path planning, android soccer system, self-organizing map neural network, A* algorithm, FIRA AndroSot game

In this paper, we propose a visual servo strategy scheme for robot soccer systems. All the controls of androids are based on the image processing of top-view screens captured by an overhead CCD camera upon the game field. To separate the targets from the background, a self-organizing map neural network (SOMNN)-based image processing scheme is proposed, in which the gravity of every color is calculated using the algorithms of image filtering, division, and morphology. It provides the coordinates and heading angle of every soccer android using the information of color pitches. The decision-making subsystem adopts the A* algorithm to determine a collision-free, direct, and optimal path so that the androids can trace a ball faster and kick well. The experimental results in the real FIRA AndroSot games demonstrate the feasibility of the proposed visual control system.

1. Introduction

An android is a conspicuous type of robot because of its similar behavior and features to those of a human. It behaves as one of the most suitable robots to live with people in a human environment. In the past few years, many real androids have been developed; however, most of them still do not reach human standards. For example, the android “Asimo”,(1) which is regarded as one of the world’s most advanced androids, only can perform several simple preset operations and communicate with humans by limited conversation and stiff emotional expression using its hands and/or body. The ideal android in people’s mind, which should show automatic movements and human-android interaction, have a pleasing appearance and the ability to follow, guide, perceive, recognize, act, and interact with people or the environment.

Professor Alan Mackworth first mentioned the idea of robots playing soccer in his paper(2) in 1993. Some roboticists expect that a team of android soccer players will win a match against the world’s best humans by the year 2050. Since soccer is a sport that involves individual motion controls and group cooperation, the robot soccer game is a suitable platform to catalyze the development of robotics. A successful soccer match needs good group cooperation, which depends on clear matching situation assessments and smart strategy decisions. Many

*Corresponding author: e-mail: myshieh@stust.edu.tw

http://dx.doi.org/10.18494/SAM.2018.1792

ISSN 0914-4935 © MYU K.K.
http://myukk.org/

approaches[3–5] focus on developing learning algorithms for soccer robotic group cooperation.

In android soccer games, not only the androids' motion and gait controls but also real-time visual and game strategy servo controls are needed. Therefore, to simplify the problems that exist in an android soccer game, the Federation of International Android-soccer Association (FIRA, http://www.fira.net/) intends to lead researchers to focus first on the development of visual servo control strategies rather than on an all-in-one design for a complete android. FIRA began to promote the AndroSot game in 2007. The AndroSot game aims to let the contestants design their own android soccer system based on a top-view vision system, a PC for decision-making, strategies, image processing, wireless communication modules, and autonomous androids. These androids execute the strategic commands received from the PC, where all strategies are based on the visual processing of real-time field sequence images captured by a camera installed above the game field.

In an android soccer system, high-performance servo controls, reliable sensory and processing systems, wireless communication network, and strategy-making systems must be integrated. Behnke *et al.*[6,7] proposed fundamental design schemes for humanoid robots playing soccer games. Baltes *et al.* designed a small humanoid robot to balance itself actively via a gyroscope servo control.[8] Tang *et al.*[9] adopted zero moment point (ZMP) trajectories as control conditions to provide feedback and real-time control of the android's gaits during a soccer game. Wong *et al.*[10] proposed a vision-based fuzzy obstacle avoidance method, and Chen *et al.*[11] discussed multirobot segmentation and identification problems using HIS color space.

In a soccer game, a good matching situation assessment is based on a clear visual recognition of objectives. Matching situation assessment will affect group strategy making such as localization, obstacle avoidance, path planning, and the control of all movements in android soccer games. In this paper, the authors propose a self-organizing map neural network (SOMNN) to perform the real-time image segmentation processing for an android soccer system. The SOMNN scheme aims for not only the real-time, but also stable and accurate, performance of the decision-making system of optimal motion controls for a real android soccer game.

The SOMNN in the proposed android soccer system is used to segmentalize the objectives from the background in a dynamic image. The first is to update the weighting values in accordance with the change in background illumination. Next, it makes the objectives clearer by filtering out the background. Afterward, one can use an image filter, color model transform, and image morphograph methods for object identification. Actual experimental results reveal that the SOMNN has improved visual sensing and recognition for actual android soccer games.

2. System Configuration

AndroSot games in the FIRA RoboWorld Cup can be illustrated simply as in Fig. 1. In such a competition, each team mounts their camera above the field to take top-view image sequences continuously, which are sent to the host computer for visual servo and path planning strategy computations. Then, control commands are transferred by wireless communication to androids playing the soccer game. From the above processes, the android soccer system may consist of

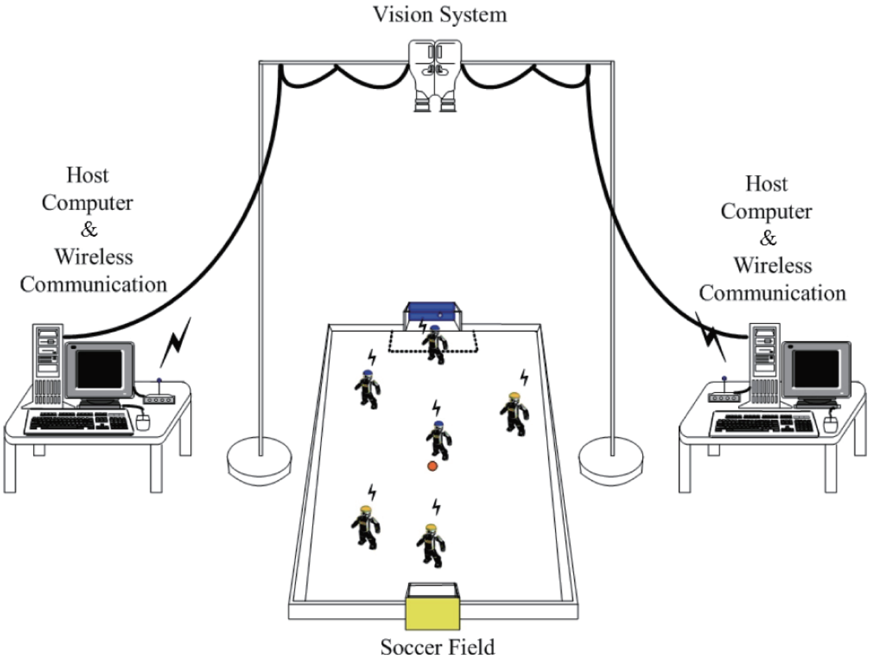

Vision System
Host Computer & Wireless Communication
Host Computer & Wireless Communication
Soccer Field

Fig. 1. (Color online) System configuration of FIRA AndroSot games.

three subsystems, namely, the visual subsystem, the strategy-making subsystem, and the motion control subsystem.

The flowchart in Fig. 2 illustrates the signal processing that exists in android soccer games. While a game is in progress, the CCD camera first captures the top-view image sequences of the field; then the objective features of the androids, the ball, and the environmental objects such as the door are determined by digital image processing procedures. These objective features will be provided for the strategy-making subsystem to determine path planning, game strategy decision, and movement controls. The androids will execute the commands received through Zigbee modules to respond to appropriate game actions such as tracking the ball, kicking the ball, and moving for defense or attack.

2.1 Visual subsystem

The visual subsystem in the proposed system mainly consists of a CCD camera, an image grabber card, attached modules, and digital image processing programs. The key goal is to determine the features of every object in the game field, such as the locations of the ball, the androids, the door, and the sidelines.

In this study, as shown in Fig. 3, sequences of image processing procedures are used in the proposed visual subsystem. The images are captured in 30 frames, with a resolution of 640 × 480 in full red–green–blue (RGB) color using the camera CV-M77 produced by JAI Corp. Note that the color values in RGB are very sensitive to intensity, however, the color values in HSV, YIQ, NCC, or YCbCr color models are less sensitive than those in RGB. Generally, converting

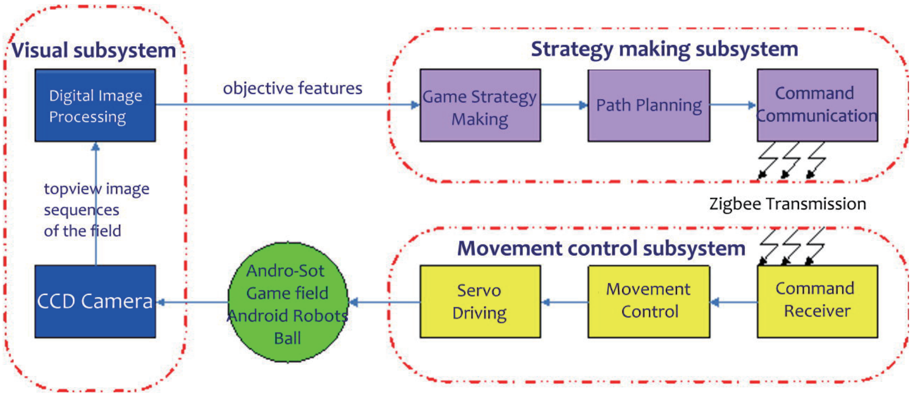


Fig. 2. (Color online) Flowchart of the signal processing in android soccer games.

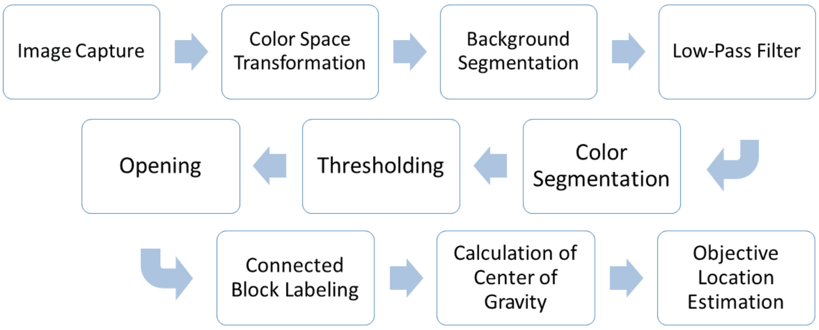


Fig. 3. (Color online) Flowchart of the visual subsystem.

the original RGB images to YCbCr ones will reduce the intensity sensitivity. In the YCbCr color model, the parameter *Y* can be regarded as an intensity factor; the parameters *Cb* and *Cr* respond to the chrome. A popular equation for color space conversion is depicted as

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2549 & 0.5059 & 0.0980 \\ -0.1451 & -0.2902 & 0.4392 \\ 0.4392 & -0.3647 & -0.0706 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}. \tag{1}$$

For real-time competition, the objective features of the ball, the androids of opposite and host sides, and the environmental objects must be determined, especially their locations. To estimate the location, the background segmentation should first be considered. Fortunately, since the background of the field for a soccer game is mostly black and with several simple types of lines, rectangles, curves, and circles in white, the background could be easily segmented from the images. In this work, the targets are segmented from the background by using the SOMNN-based image processing algorithm.

Since there are six androids that exist in a 3 on 3 AndroSot game, segmenting these robotic blocks from the top-view image is difficult if there is no additional or uniform mark on the androids. The FIRA AndroSot game rules stipulate that each team must assign its team patch as either blue or yellow, as shown in Fig. 4. In a match, the color assigned to each team is unchanged. For a fully autonomous competition, the team color patch is between 35 × 35 to 70 × 70 mm^2 in size to make it visible from any side. Other colors can be used as robotic colors on the patches, except yellow, blue, black, white, or other similar color.

Among the preparation procedures of a competition, calibrating the field image is the first step. The calibration procedure shown in Fig. 5(a) is proposed. There are arrays of black points spaced equidistantly as datum points. To determine the interval between every two points on the image plane, one can transfer it with a fixed formula to obtain a proper correction. Moreover, since the illumination throughout the field is inconsistent, there are different illumination values that exist in different areas, as shown in Fig. 5(b). The phenomenon usually results in two distinct RGB values revealed by the same color pitches in different locations. If this phenomenon is very conspicuous, the image discrimination will be poor. Adopting a wider RGB range of one color can enable immediate color discrimination, reducing the error in image processing.

2.2 Strategy-making subsystem

The proposed strategy of a 3-on-3 android soccer game consists of two stages: the principal strategy and the immediate strategy. The principal strategy of every android is chosen in accordance with the region in which the ball is. Then, the immediate strategy provides the

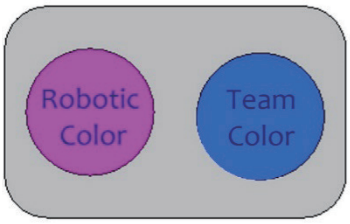
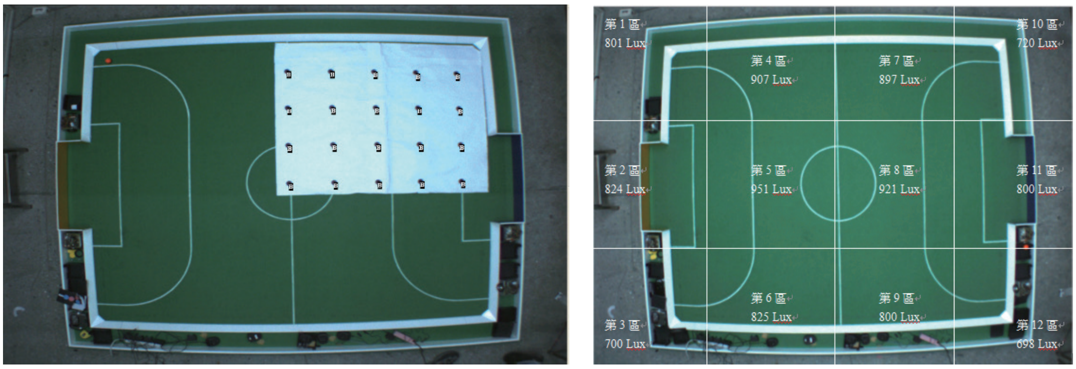


Fig. 4. (Color online) One possible team patch on an android.



(a) (b)

Fig. 5. (Color online) Calibration and partition of image plant. (a) Calibration of field image. (b) Partition for immediate color discrimination.

integrated control to every android. Taking the consideration of the android's mobility, the immediate strategy aims to make the androids move cooperatively. The role of each android is strict and clear, that is, they all have a duty region. They will not rush, dispute or contend; in fact, they will move in order and obey a clear but harmonious strategy.

In this work, the decision-making subsystem mainly infers the locations of the ball and the soccer androids using the strategy rules listed in Table 1. For simplification, it defines the center of the team logo as the center of the android. Let us define A (q_{rx}) as the angle between the heading and the x axis, and B (q_{rb}) as the angle between the ball and the x axis. The relationship can be explained as in Figs. 6(a) and 6(b).

The relative angle between the android and the ball, θ_{rb}, can be used to decide what action mode should be adopted, and then to choose the turn angle of the android. Different situations are suitable for different controls, as summarized in Table 1.

For the strategic planning of soccer android games, there are two types of strategy proposed in this paper. At the beginning of a competition, the main strategy of the team will be decided in accordance with the location of the ball. For example, if the ball is in the middle field, the team will adopt the normal strategy in which there are two strikers, a center, a back-fielder and a goalkeeper. On the other hand, if the ball stays in the front half-field, the main strategy will be changed to an offensive one with a formation of three strikers but no center. Similarly, if the ball is under the control of opponents, the main strategy must be set to a defensive one and the number of back-fielders should be increased to two.

Table 1
Rule of dual-direction movement.

Quadrant of ball	Quadrant of robotic heading	Condition	Action
1st	1st	0° ≤ A − B < 90°	Slow right turn
	2nd	90° ≤ A − B < 180°	Normal right turn
	3rd	180° ≤ A − B < 270°	Fast right turn
	4th	270° ≤ A − B < 360°	Reverse right turn
2nd	1st	270° ≤ B − A < 360°	Reverse left turn
	2nd	0° ≤ B − A < 90°	Slow left turn
	3rd	90° ≤ B − A < 180°	Normal left turn
	4th	180° ≤ B − A < 270°	Fast left turn

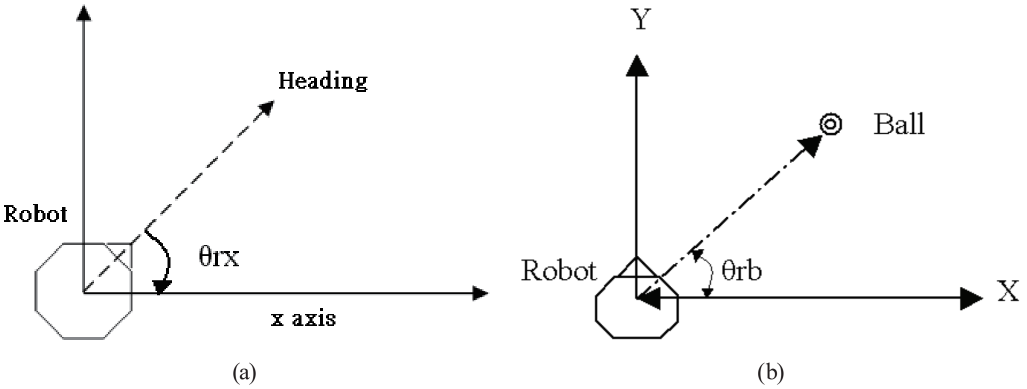


Fig. 6. Relative angles of (a) robotic heading and (b) ball.

Another strategic mode is the real strategy, which is used to position each android. The relative locations of the ball, the opponents, and the teammates will determine what role each android should play. For instance, if the ball stays in the front half and far from the boundaries, the attacker should move toward the ball and try to shoot it towards the goal. On the other hand, even though the ball is in the front half field, if it stays very near the boundaries, the attacker should rotate to kick the ball away from the boundaries. In consideration of ball control and further shooting, the rotating kick should be performed when the ball is near the lines.

Sometimes the android might be closer to the goal than the ball. In this situation, the android must adjust its position and aspect, because incorrect directional shoots will enable the opponent to score. Fortunately, the robotic mechanism includes the function of dual-direction movement; it allows the android to turn about directly rather than to circle around.

The duty of strikers is to earn points. When the ball falls into the duty area of strikers, strikers will try to move close to the ball first, then to control it, and finally, to shoot it towards the goal. The hitting and shooting actions must be carried out as fast as possible. We believe that to control the ball is to dominate the game. Dual-direction movement and fast hitting increase the mobility and scoring efficiency.

The center is assigned to patrol the middle field. Its job is not only to prevent the ball from entering the home area but also to kick the ball toward the front half-field. The center does not need to shoot the ball but sometimes needs to kick the ball outside the field if the team is in a difficult situation. More serious defensive tasks will be carried out by back-fielders. Their duty is just to kick the ball away and prevent opponents from scoring. If necessary, the number of back-fielders will be increased to three. Moreover, proper interference is allowed as necessary. The goalkeeper moves in accordance with the y axis of the ball. This movement can prevent the opponent androids from scoring. If the goalkeeper is held up so that it cannot move, the defenders will change roles to help the goalkeeper. The goalkeeper and defenders will cooperate with each other.

2.3 Movement control subsystem

The responding efficiency of androids will dominate the control of the ball and raise the possibility of winning. Generally, a fine design of a robotic mechanism and powerful servo controls result in faster and more stable movements for androids. In addition, the robotic movement can also be accelerated by simple and direct control strategies if the hardware and mechanism are limited. However, it does not work all the time.

In this work, the proposed androids were originally made by Robotics Corp. The authors endeavored to increase the force of the knee of the robotic legs by changing the high-torque servomotors of the knee joints. Moreover, the mechanisms of the androids are all remade using lighter material, which resulted in faster and more powerful movements.

The movement controller is based on the locations of each android and the ball. The controller analyzes all the distances, directions, and relationships among the ball, doors and androids. For simplification, the decision can only depend on which android can kick the ball with highest possibility every time. The highest possibility means that both the velocity of each android and the distance between the android and the ball should be considered. The reaching time is defined as the distance to the ball divided by the velocity. The android with the shortest time will be the most possible kicker at that moment. On the basis of this concept, the controller should decide to attack or defend depending on who is the most possible kicker and who belongs to the host or the visiting team.

3. Visual Proceeding Procedures

3.1 SOMNN-based image segmentation processing

Before the segmentation procedure, the initial background model must be set up. Since the first frame of the image sequence is indeed a good initial background, for each pixel, the corresponding weight vectors are initialized with the pixel value.

After initialization, the following samples are sent to the SOMNN. Each input pixel $\mathrm{p_t}$ of the ith frame $\mathrm{I_t}$ is compared with the current pixel model C to determine if there exists a weight vector $W[i, j]$ that best matches it. If a best-matching weight vector c_m is found, it indicates that p_t fits to the background and it is adopted as the pixel encoding approximation, and the best-matching weight vector, together with its neighborhood, is reinforced. Otherwise, if no acceptable matching weight vector exists, we determine whether p_t is in the shadow cast by some object or not. In the first case, p_t should still be considered as background, but it should not be used to update the corresponding weight vectors; in the latter case, p_t is detected as belonging to a moving object (foreground).

The proposed architecture is shown in Fig. 7. The background subtraction and update procedure for each pixel can be summarized as in the proposed SOMNN algorithm. In the algorithm, the inputs are the pixel values p_t in the frame I_t, $t = 0, 1, ...,$ lastframe; the outputs are the background or foreground binary mask value $B_t(p_i)$.

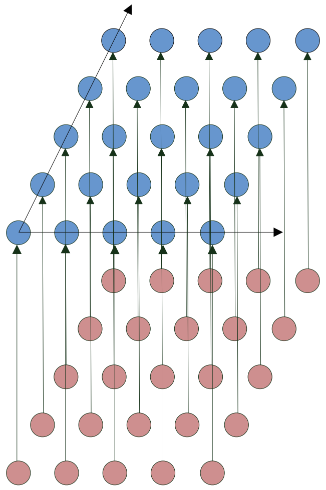


Fig. 7. (Color online) Architecture of SOMNN for segmentation (red: inputs, blue: outputs).

Step1: Initialize model C for pixel p_0 and store it into A.

Step2: Find the best match c_m in C to the current sample p_i. The nearest Cb/Cr color pair is regarded as the best match, i.e., the *Net*, given by Eq. (2), is used for comparison routing to find the smallest one among all samples. In Eq. (2), $Cb[i, j]$ and $Cr[i, j]$ denote the current samples Cb and Cr color values of current samples, respectively, and $W[i, j, Cb]$ and $W[i, j\ Cr]$ denote the corresponding weight vector values of Cb and Cr color values respectively.

$$Net = \sqrt{Cb\_D + Cr\_D} \tag{2}$$

Here, $Cb\_D = (Cb\,[i, j] - W\,[i, j, Cb])^2$ and $Cr\_D = (Cr\,[i, j] - W\,[i, j, Cr])^2$.

Step3: If *Net* is smaller than the threshold, $B_t(i, j)$ is given as 0; otherwise, output the current values.

Step4: Update A in the neighborhood of c_m using

$$B_t(i, j) = (1 - \alpha)B_{t-1}(i, j) + \alpha I_t(i, j), \tag{3}$$

where $B_t(i, j)$ is the new updated neighborhood, $B_{t-1}(i, j)$ is the old one, and $I_t(i, j)$ is the current input and denotes the learning rate.

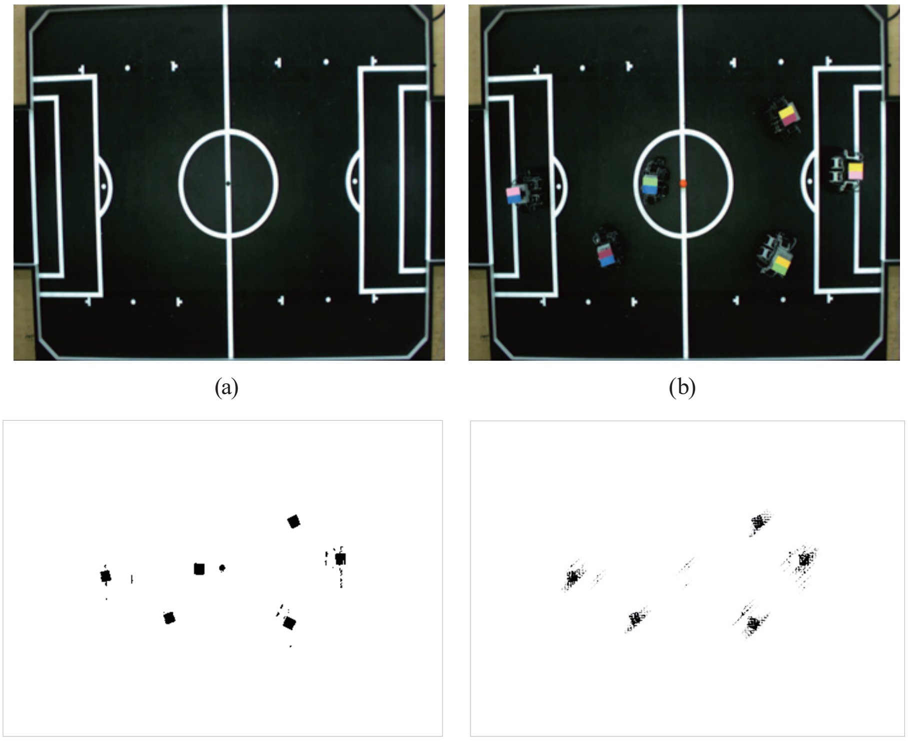
The aim of object segmentation is to identify the objects that are usually regarded as the foreground from the background. Moreover, it also plays the role of a color filter whenever color is adopted as the distinguished factor in image segmentation. The SOMNN is a learning algorithm that can be used to perform the real-time image segmentation processing for the android soccer system. In such an algorithm, the threshold A will be updated on the basis of the changing $B_t(i, j)$ in Eq. (3). It is useful to overcome these inconsistent color values under varying illumination.

The segmentation results given by SOMNN algorithms are illustrated in Figs. 8(a)–8(c). The initial background frame is shown in Fig. 8(a); the current frame is shown in Fig. 8(b); the segmentation result based on the proposed scheme is shown in Fig. 8(c); and the segmentation result by direct image subtraction is shown in Fig. 8(d). From the result in Fig. 8(d), it is seen that the moving objects in the center area are missing after the segmentation. The result in Fig. 8(c) demonstrates the good performance of the proposed algorithms.

3.2 Objective data processing

Since the image is captured from the top of the field, most of the posture data of the androids are unknown, except the information of the patch at the top of the robotic head. It also means that the strategy decided upon only can yield which location is the android, which color the robotic patch is in, and which direction is shown in the patch.

As shown in Fig. 9, the direction that the android faces can be chosen relative to the center line between the two centers of marks. Note that the direction is forward while the team logo (blue) is on the right-hand side of the member mark (purple or orange). Figure 9 also illustrates



(a) (b)

(c) (d)

Fig. 8. (Color online) Segmentation result given by the SOMNN algorithm. (a) Initial background frame, (b) current frame, (c) segmentation result (proposed scheme), and (d) result of direct image subtraction.

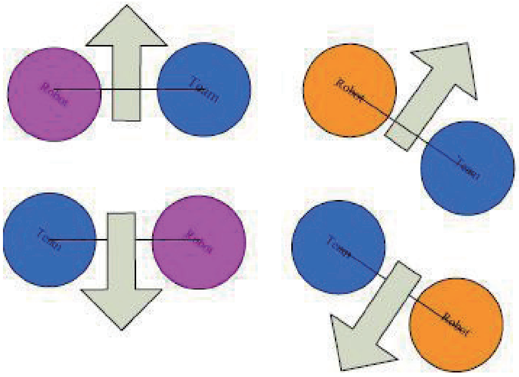


Fig. 9. (Color online) Geometric relationship of patches.

the direction θ, expressed as Eq. (4), that the android faces and the location of the android, that is, the center of two centers of the marks, shown in Eqs. (4a) and (4b):

$$\theta = \tan^{-1}\left(\frac{CenterY_{team} - CenterY_{robot}}{CenterX_{team} - CenterX_{robot}}\right), \tag{4}$$

where
$$CenterX_{android} = \frac{|CenterX_{team} - CenterX_{robot}|}{2}, \tag{4a}$$

$$CenterY_{android} = \frac{|CenterY_{team} - CenterY_{robot}|}{2}. \tag{4b}$$

In Eq. (4), the center coordinates of the team logo are denoted as ($CenterX_{team}$, $CenterY_{team}$); the center coordinates of the robot mark are denoted as ($CenterX_{robot}$, $CenterY_{robot}$); and the center coordinates of the android are denoted as ($CenterX_{android}$, $CenterY_{android}$).

The resolution of field images used in the global scan is 1024 × 768. By taking the image resolution in comparison with the size of the real field area, an object such as the ball of 4 cm diameter is displayed in the image of 25 pixel diameter. Thus, the index taken by the global image scan should be limited to 25 pixels. In our study, to increase the efficiency of the scan, an index of 2 pixels is used. From the illustration in Fig. 10, the search time will be reduced by 4 times in comparison with that of the original scan.

When the image searching steps are finished, four resulting parameters are then sent to the androids for motion control using wireless modules. The parameters received in each android are the member number, the mode of action, the speed, and the direction of the robotic movement. The first code, i.e., the member number of the android, indicates the role of the android. The second code, i.e., the mode of action, is given by the task assignment. The last two codes tell the android how and what it should move to.

3.3 Path planning processing

During the strategy-making procedures, the decision-making scheme always calculates only the direction of the movement and ignores whether there are obstacles along the way. In such a common situation as shown in Fig. 11, there is an obstacle (the red object) that exists between the origin and the destination. Here, the authors intend to execute path planning by adopting the A* algorithm, which is used for the optimal shortest problem. Figures 12 and 13 show the results of path planning using the A* algorithm. From the simulation results shown in Fig. 13, one can see that the path is really collision-free and optimally the shortest one.

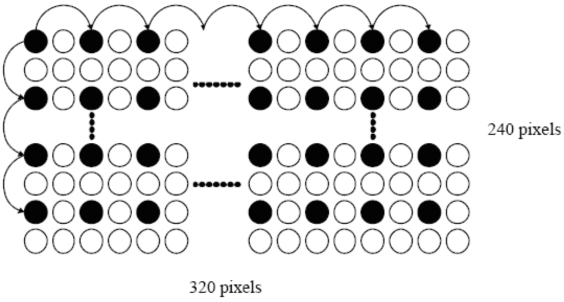


Fig. 10. Global image indexing scan.

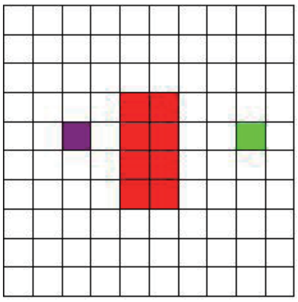


Fig. 11. (Color online) Case of an obstacle (red) existing between the origin (purple) and the destination (green).

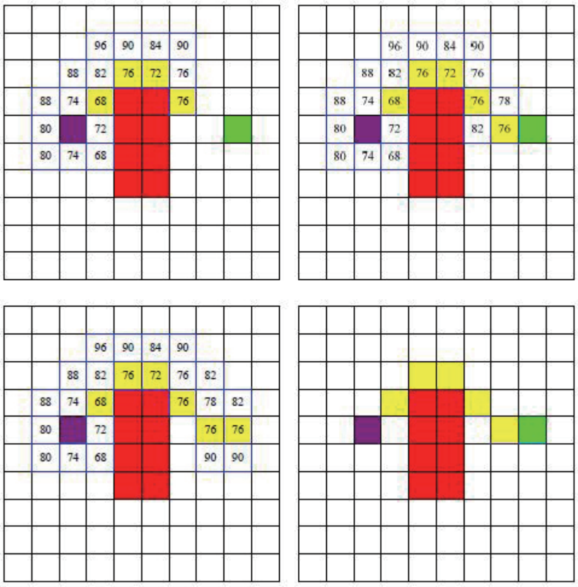


Fig. 12. (Color online) Result of path planning using the A* algorithm.

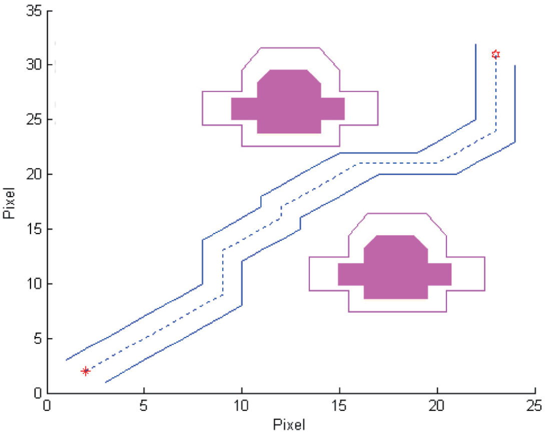
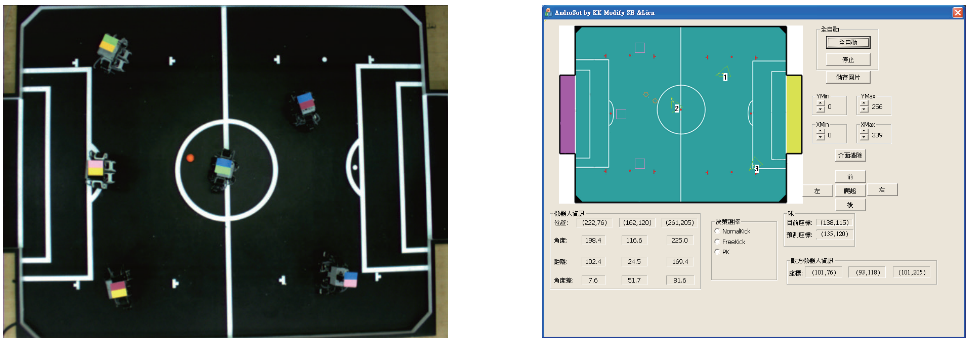


Fig. 13. (Color online) Simulation result of path planning in the case of two obstacles, using the A* algorithm.



YMin 0
YMax 256
XMin 0
XMax 339

機器人資訊

位置:	(222,76)	(162,120)	(261,205)
角度:	198.4	116.6	225.0
距離:	102.4	24.5	169.4
角度差:	7.6	51.7	81.6

決策選擇
○ NormalKick
○ FreeKick
○ PK

球
目前座標: (138,115)
預測座標: (135,120)

敵方機器人資訊
座標: (101,76) (93,118) (101,205)

(a) (b)

Fig. 14. (Color online) Interface of experiments. (a) Real-time top-view screen of field and (b) display of the interface.

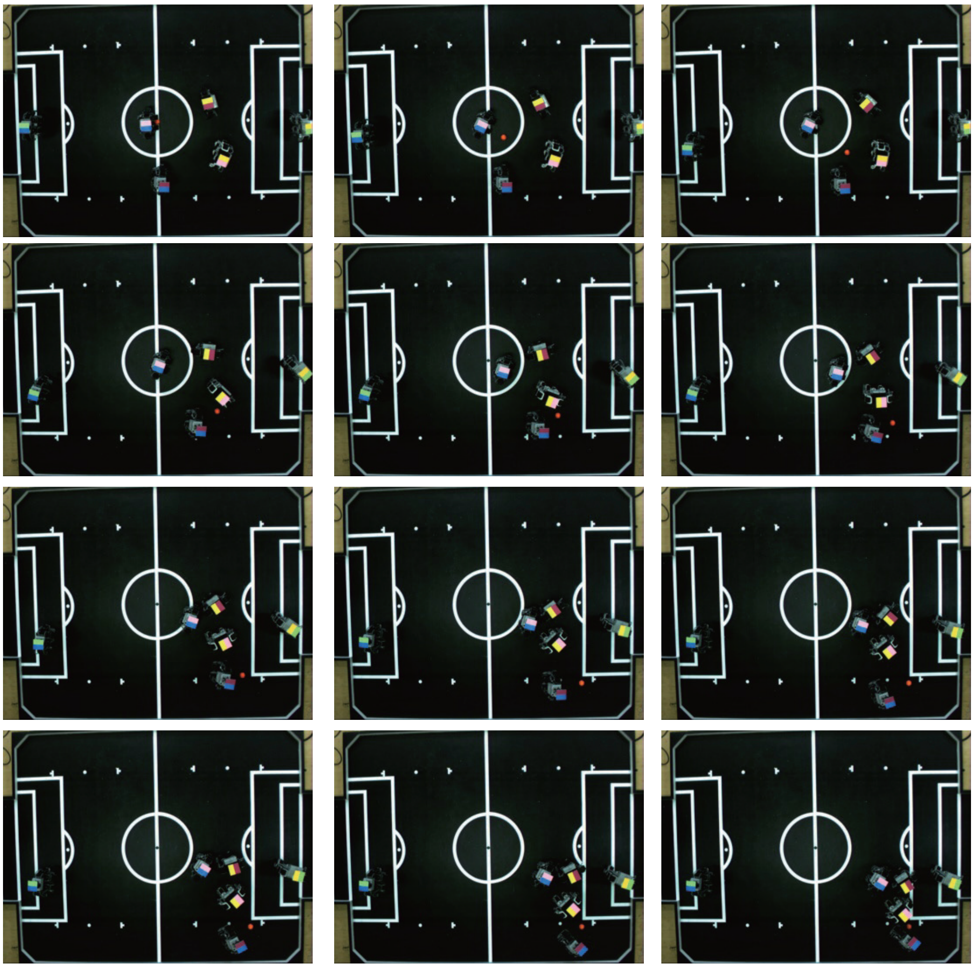


Fig. 15. (Color online) Experimental results of SOMNN and A*-based scheme in a real game.

4. Experimental Results

For real-time observation, the proposed system consists of an interface, as shown in Fig. 14(a) showing the immediate image of the field, and the estimated results, as shown in Fig. 14(b) after strategy-making procedures. The image sequences shown in Fig. 15 illustrate the actual experiments of a real game. The androids are under visual servo control to trace the ball and to move collision-free and fast. The results demonstrate that the SOMNN segmentation algorithm helps to obtain clear and exact objectives and the A* algorithm provides better path planning for androids to move more efficiently.

5. Conclusions

A vision-based android soccer system was implemented in an actual FIRA AndroSot game. A visual subsystem was designed for the recognition of objective features on the basis of a SOMNN visual processing system. The host computer performs image processing, strategy making, and control command decision, and provides appropriate commands to each android to kick the ball, trace, or defend. The A* algorithm is used to improve path planning efficiency. The experimental results revealed the feasibility of the proposed system.

Acknowledgments

The author would like to express their sincere thanks to the reviewers for their invaluable comments and suggestions. This study was supported by the Ministry of Science and Technology, Taiwan, under Grant numbers MOST105-2221-E-218-019 and MOST106-2221-E-218-003.

References

1 Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura: Proc. 2002 IEEE Intelligent Robots and Systems Conf. **3** (IEEE, Switzerland, 2002) 2478.
2 A. Mackworth: Computer Vision: Systems, Theory and Applications, Eds. A. Basu and X. Li (World Scientific Press, Singapore, 1993) Chap. 1.
3 K.-S. Hwang, S.-W. Tan, and C.-C. Chen: IEEE Trans. Fuzzy Syst. **12** (2005) 569.
4 H. S. Kim, H. S. Shim, M. J. Jung, and J. H. Kim: Proc. IEEE Int. Symp. Robotics and Automation (IEEE, California, 1997) 390.
5 Y. Wang and C. W. Silva: Eng. Appl. Artif. Intell. **21** (2006) 470.
6 S. Behnke, M. Schreiber, J. Stückler, R. Renner, and H. Strasdat: Proc. 6th IEEE-RAS Int. Conf. Humanoid Robots (IEEE, Italy, 2006) 497.
7 S. Behnke: Proc. 2006 IEEE Workshop Benchmarks in Robotics Research (IEEE, Jeju, 2006) 231.
8 J. Baltes, S. McGrath, and J. Anderson: Proc. 2nd Int. Conf. Autonomous Robots and Agents (New Zealand, 2004) 470.
9 Z. Tang, C. Zhou, and Z. Sun: Proc. 2003 IEEE Robotics and Automation Conf. **2** (IEEE, Taipei, 2003) 2455.
10 C.-C. Wong, C. L. Hwang, K. H. Huang, Y. Y. Hu, and C. T. Cheng: Int. J. Fuzzy Syst. **13** (2011) 45.
11 F.-D. Chen, B. R. Hong, and Y. Zhu: J. Harbin Inst. Tech. **7** (2004) 24.

About the Author

Ming-Yuan Shieh received his B.S. degree in 1992 and his Ph.D. degree in 1998 from National Cheng-Kung University, Taiwan. From 2000 to 2008, he was an assistant professor at Southern Taiwan University of Science and Technology, Taiwan. Since 2008, he has been an associate professor at Southern Taiwan University of Science and Technology, Taiwan. His research interests are in intelligent control, digital image processing, and humanoid robotics.