FPGA-BASED UBIQUITOUS COMPUTING INTELLIGENCE FOR ROBOTIC FORMATION CONTROL

Ying-Hao Yu, S. Kodagoda and Q. P. Ha
School of Electrical, Mechanical and Mechatronic Systems
University of Technology, Sydney
Broadway, NSW 2007, Australia
Email: {YingHao.Yu|sakoda|quangha}@eng.uts.edu.au

Abstract

Ubiquitous computing (UC) has been an important stream in distributed computing, which is nowadays recognized as an attractive vision in ambient intelligence (AmI) technologies and home robotic systems. This paper, focusing on our work on the application of field programmable gate array (FPGA) in mobile robotics, presents a prototypical computing node of a ubiquitous robot (Ubibot) for indoor multiple robot coordination. The hardware based FGPA designs such as colour discrimination, object tracking, relative distance estimation, and robotic steering manoeuvre are integrated into a single chip. This hardware design, based on the system-on-programmable chip concept, will demonstrate the feasibility of an AmI environment for real-time processing with lower power consumption.

KEYWORDS: Ubiquitous computing, FPGA, robotic formation.

INTRODUCTION

The concept of ubiquitous computing is referred to as a distributed, embedded, context-aware, and unobtrusively computing system, first coined in 1991, by Mark Weiser in *Scientific American*. In his vision, human's society will be surrounded by intelligent services with computer technologies in the future. For this reason, the centralized computing control will be replaced by the distributed nodes (called UC nodes), a virtual creature will perform the role of user interface, and the embedded system in UC will be much more proper than a general purpose computer (Mühlhäuher and Gurevych, 2008). The most salient features of UC include network and application scalability, wireless network connectivity, adaptability and context-aware computing, information technology security and liability, and human-computer interaction. Capturing all these advantages, the ultimate objective of AmI is to provide the information-based intelligent environment in working and living places, whereby lots of embedded systems collaborate with the infrastructure, portable devices, artificial intelligence technologies, wireless communication, sensing technologies, and mobile agents will satisfy people's living essentiality and entertainment (Dopico *et al*., 2009).

In robotics, the ubiquitous robot (Ubibot) is impersonating the mobile abilities in AmI environments. Different components of Ubibot are classified into three types: Sobot, Mobot, and Embot (Kim, 2006). Sobot is a software-based 3D virtual pet, performing the user interface to communicate with customers, UC nodes, and the system manager; Mobot is a mobile platform bringing services to customers, while Embot covers all sensory functions.

Sensors such as digital camera, microphone, and laser range finder are used for vision, speech, and distance measurement, and the wearable devices and underlying active or passive RFIDs are useful for robotic localization (Koch *et al*., 2007). Ubibot concepts have been realized in warehouse automation with service robots called the Kiva's. Staff in a working place can command them to cooperatively pick up various inventories from distributed nodes (D'Andrea *et al*., 2008). The assigned robots track the bar codes on the floor to reach their destinations then come back to staff. Every robot's location in the Kiva system is well supervised by computers through wireless communication. The only problem in such system may perhaps rest with bar codes that are smeared by foot print, greasiness, or dust.

Since using the markers is currently a popular and realistic scheme to localize the robots in an automation system, implementing the markers on robots' body seems to have more flexibility without dramatically modifying the working or living place, particularly when installing the markers on the top of robots, where they can be easily tracked by the portable global cameras in an indoor environment (Stubbs *et al*., 2006). However, since high pixel resolution image processing is still a tough requirement for real-time image processing, many external computers may be needed to collaborate with each camera.

In this paper, based on our recent work on FPGA hardware designs (Yu *et al*. 2009a), we propose a chip-based real-time UC node for indoor environments. As shown in Figure 1, a FPGA chip performs the roles of Mobot and Embot, and a global camera is working as the sensory function which tracks three miniature robots. The UC node functions are all realized on a FPGA platform Altera DE2-70 with Cyclone II chip, shown in Figure 2(a). It receives the external instructions (e.g. UC server) then performs image processing, robotic tracking, relative distance estimation, and steering control in pure register transfer level (RTL) circuits. For vision sensing, we use a Terasic 5M pixels digital camera module set for 1024x1280 pixels resolution with 34 fps, as shown in Figure 2(b). The robots used for demonstrating formation control are the Eyebots, shown in Figure 2(c).

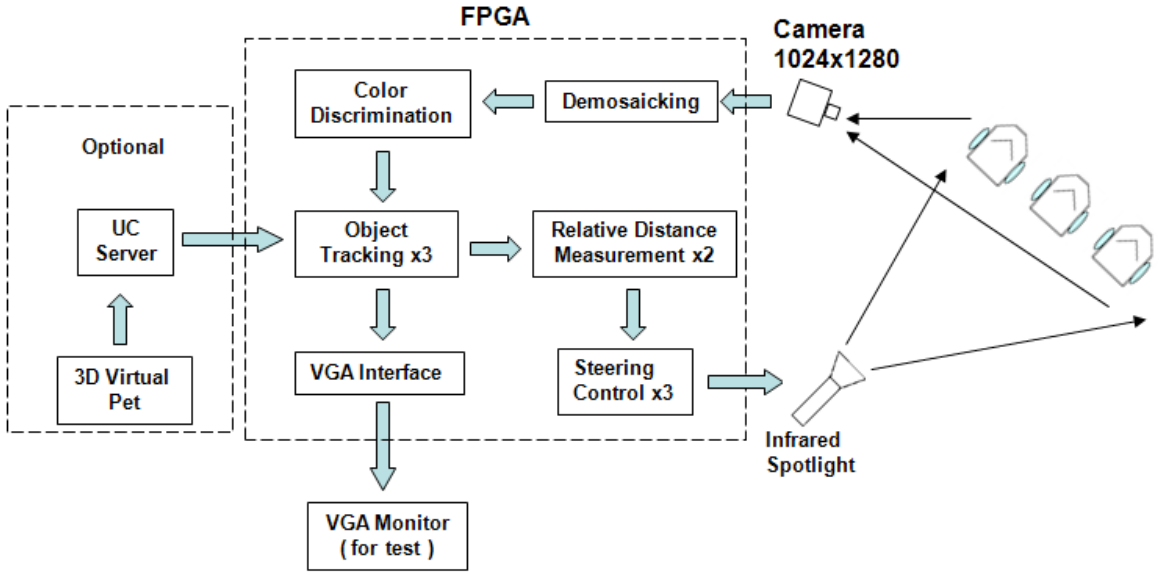
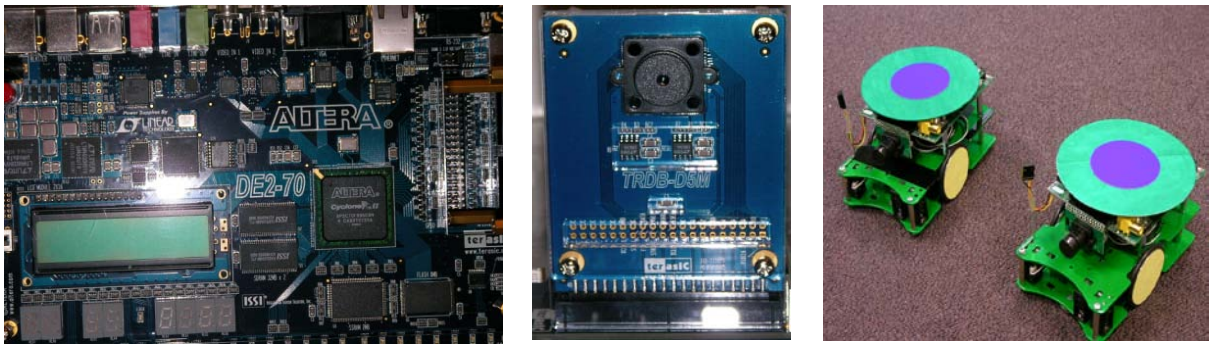


Figure 1: Proposed UC node with FPGA designs.

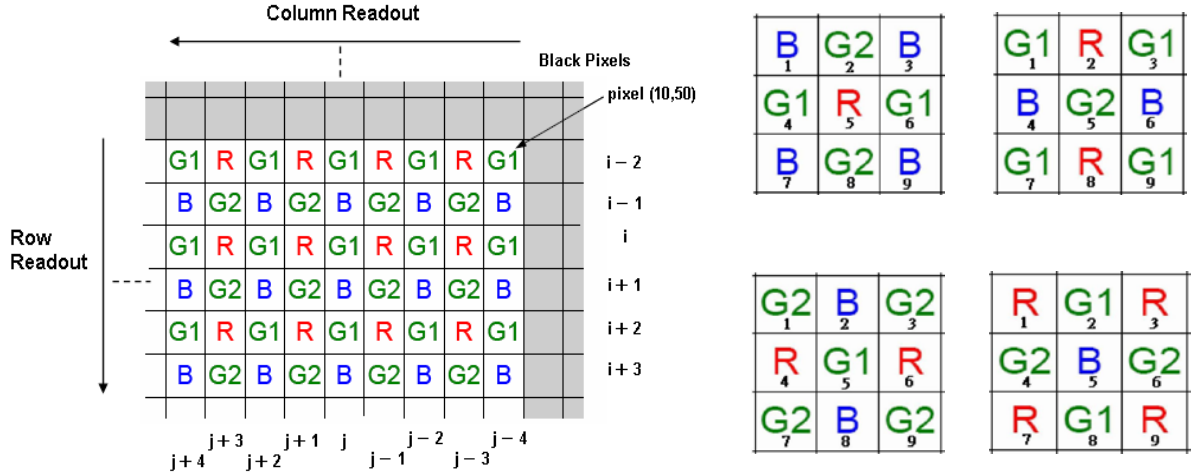


a. FPGA developing platform.

b. 5M pixels camera module.

c. Eyebots with bull eye labels.

Figure 2: FPGA-based Ubibot system.



a. Bayer pattern.

b. Four possible combinations for 3x3 pixels arrangement.

Figure 3: Bayer pattern arrangements.

DESIGN METHODOLOGIES

Raw Image Demosaicking

The digital image sensor array in the adopted camera module is arranged in a Bayer pattern (Lukac *et al.*, 2005), see Figure 3(a). Every pixel in such pattern only detects in monochromatic with red (R), green (G), or blue (B) colour. That makes the output image looks as a mosaic-like picture then named raw image. For converting the raw image into the full colour image, or demosaicking, the missed colours of every central pixel should be interpolated by deliberating the strength of neighbouring pixels (Wang *et al.*, 2005). The simplest demosaicking procedure is to pick up the absent pixels from the nearest vicinity. Such method does not spend any calculation effort and suitable for a required high-speed system. Unfortunately, it will yield intolerable saw-toothed artifacts. Thus, by considering the

reasonable image quality, logic gate usage, and real-time image processing, the bilinear interpolation seems to be a moderate scheme to ease off the saw-toothed artifacts (Acharya and Ray, 2005). This algorithm is aimed to interpolate colours by averaging pixels' strength from neighbourhood. To perform a bilinear 3x3 arrangement, the three lines shift registers are implemented as shown in Figure 3(b), where the fifth pixel is the interpolated and output pixel for every scanning instant. By extracting the essence of bilinear algorithm and considering implementation feasibility, we suggest an efficient linear algorithm as given Table 1:

Table 1: Linear Demosaicking

Interpolated Colour	Neighbourhood formation	Algorithm
G_{P5}	Rhombus	$\frac{(G_{P4}+G_{P8})}{2}$
R_{P5} (B_{P5})	Column	$\frac{(R_{P2}+R_{P8})}{2}$
R_{P5} (B_{P5})	Line	$\frac{(R_{P4}+R_{P6})}{2}$
R_{P5} (B_{P5})	Square	$\frac{(R_{P7}+R_{P3})}{2}$

It has been verified in our results that one operation of addition (+) in the linear interpolation only consumes 0.2 pixel clock, but one operation of division (/) will consume 1.2 pixel clocks in FPGA. Due to the edge latch design, the output colours are updated for every two pixels, where the effect of artifacts can be mitigated by using the proposed interpolation algorithm (Yu *et al*., 2009b).

Colour Discrimination and Object Tracking

For multiple robot coordination, tracking by colour is a feasible option as it has less computing effort for implementing on an embedded system. In this regard, the colour of markers remains a popular way for identification. Accordingly, we installed the bull-eye labels on the top of the Eyebots, Figure 2(c), with a green colour for the outer ring and blue colour in the inner area. The outer ring is designated to isolate the interference from the background to enable the initialization for the detection procedure, and the inner area is the interested label. In practice, colour tracking mechanisms are subject to the interferences from reflection and shadow. The former can be classified into diffuse and specular reflections. In diffuse reflection the light penetrating an object's surface reflects in diverse directions, and the behaviour of the specular reflection is referred to as the mirror phenomenon (Ren and Wang, 2008). Similarly, shadow reflection may be also categorised in self and cast shadows

(Salvador *et al.*, 2001). The self shadow is attached to an object that is not illuminated. In contrast, the cast shadow is projected on the ground or on other objects. Those interferences make colour tracking not a simple task for an intrinsic colour. Now a full colour I on every pixel can be composed with different strengths of R-G-B essential colours:

$$I(i, j) = m_{R(i,j)}R_{sat} + m_{G(i,j)}G_{sat} + m_{B(i,j)}B_{sat}, \quad (1)$$

where R_{sat}, G_{sat}, and B_{sat} denote the colour strength saturation, and the magnitudes m_R, m_G, and m_B are varying with different lighting conditions such as colour, illumination, reflections, and shadows. Here we assume the lighting colour and illumination are controlled in an indoor environment. Hence the interferences can be considered as moderate variations from the intrinsic colour. An example for discriminating green colour can thus be proposed as:

$$G \Leftrightarrow \left(G_{n(i,j)} - g_n + t_n\right) \geq \left(R_{n(i,j)} \cap B_{n(i,j)}\right), \quad (2)$$

where g_n is the tolerance of intrinsic green colour, n is the number of threshold levels in pixel strength. Once the system detects a pixel with green strength by ($G_{n(i,j)}$ - g_n) stronger than red and blue, a range of interested (ROI) area is then created for the following pixels (Ren and Wang, 2008). Here we assume the blue colour is of least strength in RGB which can be ignored. For a flexible discriminating boundary, variable t_n is implemented as an adjustable parameter (Yu *et al.*, 2009b). Accordingly, with the initial value $t_n^{(0)} = 0$, t_n increases for next pixel (l+1) to:

$$t_n^{l+1} = \left(t_n^* - G_{n(i,j)}^l\right) + \left(R_{n(i,j)}^l + g_n\right), \quad (3)$$

if $G_{n(i,j)}^l \leq \left(R_{n(i,j)}^l + g_n\right)$, and $G_{n(i,j)}^l + t_n^l > \left(R_{n(i,j)}^l + g_n\right)$, i.e., when the red colour is stronger than green. The threshold t_n will be reduced to a given strength threshold level t_n^* while the red colour is weaker than green:

$$t_n^{l+1} = t_n^*, \text{ if } G_{n(i,j)}^l > \left(R_{n(i,j)}^l + g_n\right). \quad (4)$$

Finally, the output noise filter is obtained by means of the shift register with the less continuity pixels being eliminated as noise while the integrity green image can entirely pass through the filter:

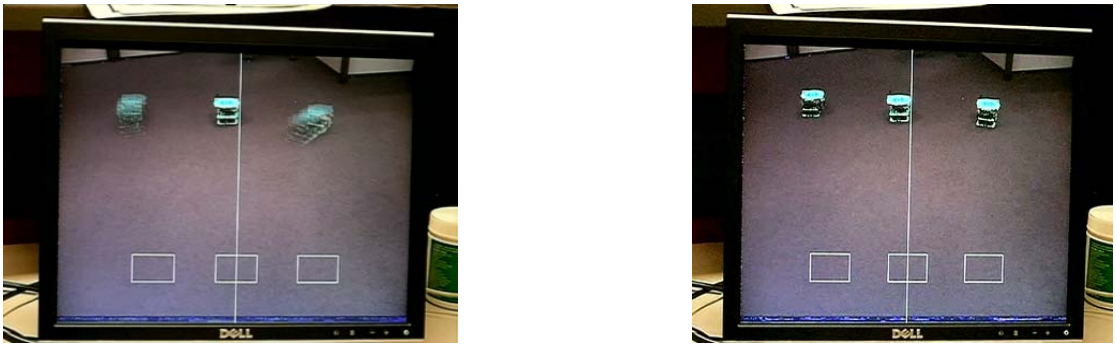
$$G\langle 1 \mid 0 \rangle = G_{(i-1,j)} \cap G_{(i-2,j)} \cap G_{(i-3,j)}. \quad (5)$$

We have verified the colour discrimination quality using the proposed adjustable threshold by demonstrating that the single threshold algorithm yielded a big black hole in the central area of a rolled up green cloth while the proposed discriminating methodology could fill most of its inner areas (Yu *et al.*, 2009b). The same algorithm is also applied for the blue label.

Finally, three docking areas are designated as depicted in the test scenarios of Figure 4, for which the steering methodologies will be briefly presented in the next section of this paper. If any Eyebot moves into the docking area, it will be immediately tracked. It follows that the blue pixels number will exceed the specified threshold, and an interested window (square mark) will then be set with the Eyebot's label. This window will be dynamically updated with

the mobile central point of a blue circle. To maintain the robot formation, the inter-robot distance measurement is important and is addressed in the following section.

a. Initial location. b. First arc turning. c. Driving straight.



d. Second arc turning. e. Arrive destination.

Figure 4: Test scenarios with the proposed hardware design.

Relative Distance Estimation between Robots

When using the digital camera in a dynamic robotic system for distance measurement, one has to resolve the need to tilt the camera, to adjust its focal length for better image quality, and to calibrate extrinsic and intrinsic parameters of camera. For possible implementation on a programmable chip, we have developed a convenient algorithm named Perspective Projection Image Ratio (PPIR) using the perspective images on a TFT-LCD (Thin Film Transistor–Liquid Crystal Display) monitor to estimate the relative distances between 2D labels of the robots. The idea is to derive the relative distance between two robots (labels) in longitudinal and lateral directions. In the leader-follower formation (Nguyen *et al*. 2008), by defining y_1 the real longitudinal length of a leader robot's label driving at front (further to camera), y_0 the follower robot's label length is (closer to camera), and d the real relative distance between two labels, the estimated distance between two disc labels on the flat floor is derived from the ratio:

$$\frac{d}{y_0} = R_v \cong d'\left(\frac{y_0' + y_1'}{2y_0'y_1'}\right)\cdot\delta_v, \tag{6}$$

where $y_0 = y_1$, and y_1', y_0', and d' are the corresponding perspective values directly measured on TFT-LCD monitor. Given $\delta_y = y_1' / y_0'$ and by noticing the bounds of δ_v, the PPIR in the longitudinal direction is determined of index n in the following approximation:

$$\delta_v \cong \delta_y + \sum_{n=1}^{n=\infty} \frac{1}{2^n}(1-\delta_y) \cong \frac{(2^n - 1)y_0' + y_1'}{2^n y_0'}, \text{ and } \delta_y \le \delta_v \le 1. \quad (7)$$

The lateral relative ratio, R_h, on the flat floor can be also derived similarly using the PPIR algorithm where index n is chosen between 3 and 4. Finally the estimated relative distance R_s can be derived by using Pythagoras formula as:

$$R_s = \sqrt{R_v^2 + R_h^2}. \quad (8)$$

According to our tests, the PPIR algorithm can efficiently keep the estimation error under 5% of the inter-robot distance, even with different camera poses and focal lengths (Yu *et al.*, 2010) and may have further applications involving estimation of the relative distance between moving objects.

Slope-Based Point Pursuing Manoeuvres

In an indoor environment, the wheeled robot's driving speed is constrained by the space dimension. Unlike a high speed vehicle which needs to consider the lateral sliding factors, the indoor robot can safely model its steering manoeuvers by geometry (Lamiraux and Laumond, 2001). By using the PPIR relative distance estimation, a slope-based arc-line-arc (SBALA) algorithm has been developed for robot steering in an obstacle-free environment. The SBALA is composed of seven critical points on 2D plane, as typically shown in Figure 5. The expected destination is denoted as point *A*, and the path orientation of point *A* is determined via 2D slope *m*. Point *B* represents the central point of robot. The line $\overline{AC}$ is decomposed into four sections as depicted. From B, the robot is steered anticlockwise (L) to make a semi-circle with radius $\overline{AC}/4$, driving straight (S) on $\overline{DG}$ to another lateral distance $\overline{AC}/4$ to point *E*, turning clockwise (R) on a quarter-circle with radius $\overline{AF} = \overline{EF}$, and reaching point *A* with the desired orientation.

DISCUSSION

Applying the SBALA algorithm, two follower robots are firstly locked from the docking areas at the bottom of monitor, then the FPGA chip incorporating the PPIR algorithm calculates the 2D coordinates and expected destination. Finally, the chip successfully leads two follower robots to arrive their destinations. The test results are shown respectively in Figure 4(a) to (e). All the proposed algorithms for colour demosaicing, object detection, distance estimation and steering manoeuvres are integrated in a Ubibot system by using just basic calculations. To avoid the floating point and trigonometric operations in FPGA, the numbers in PPIR and SBALA algorithms can be simply re-scaled by multiplying 100 with a reasonable accuracy. Even for the operation of square rooting in (8), there is a useful Register Transfer Level approximation (Chu, 2006) as:

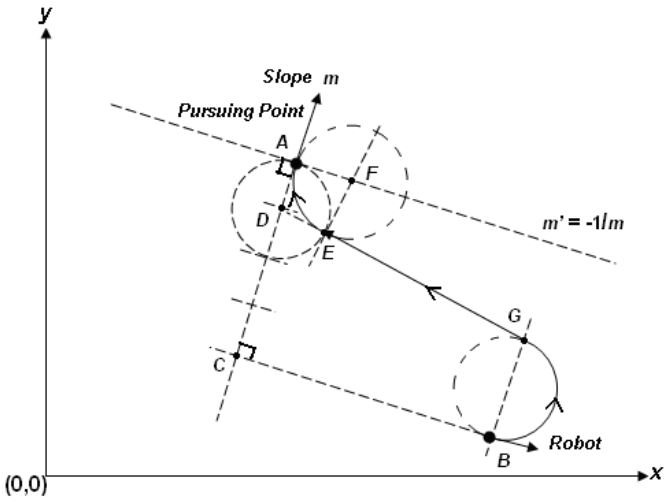


Figure 5: Steering using SBALA algorithm.

$$\sqrt{R_v^2 + R_h^2} \cong \max\left(\left(\left(a - 0.125a\right) + 0.5b\right), a\right), \tag{9}$$

where a = max ($|R_v|$, $|R_h|$), and b = min ($|R_v|$, $|R_h|$). The floating point values such as 0.5 and 0.125 in (9) represent the operations of right shift for 1 and 3 bits. Consequently, unlike the architecture of PCs, the prototypical FPGA based real-time UC node design can be realised in lower speed hardware circuits without a significant time delay.

CONCLUSION

In this paper, we have presented a prototype of ubiquitous computing node for robotic formation, designed with pure hardware circuit designs. The elaborate algorithms can artfully avoid the complicated functions. That makes the proposed FPGA chip integrate more flexible functions than in traditional techniques for identifying, locating and coordinating the robots. Compared to the PCs, such FPGA designs have the advantages of resulting in lower power and efficient real-time computing. These characteristics totally fulfil the requirements in ubiquitous computing and AmI environments. In addition, by implementing extra interfaces, the FPGA chip also provides the possibility to communicate with an external UC server. Due to the critical role of embedded systems in UC, it can be predicted that the FPGAs will be the key technologies to a successful UC society for home robotics in the future.

REFERENCES

Acharya, T and Ray, A. K (2005), *Image Processing: Principles and Application*. Canada: John Wiley & Sons.

Chu, P.P (2006), *RTL Hardware Design Using VHDL*. Canada: John Wiley & Sons, Canada.

D'Andrea, R and Wurman, P (2008), "Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities," *Proc. IEEE Intl. Conf. Technologies for Practical Robot Application*, Woburn, USA, pp. 80-83.

Dopico, J. R. R., Calle, J. D., and Sierra, A. P (2009), *Encyclopedia of Artificial Intelligence*. USA: Information Science Reference.

Lukac R., Plataniotis K.N. and Htazinakos D. (2005). "Color Image Zooming on the Bayer Pattern," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 15, No. 11, pp. 1475-1492.

Kim, J-H (2006), "Ubiquitous Robot: Recent Progress and Development," *Proc. IEEE/ SICE Intl. Conf on Digital Object Identifier*, Busan, Korea, pp. 25- 30.

Koch, J., Anastasopoulos, M., and Berns, K. (2007), "Using the Modular Controller Architecture (MAC) in Ambient Assisted Living," *Proc. 3rd IET International Conference on Intelligent Environments (IE07)*, Ulm, Germany, pp. 411-415.

Lamiraux, F and Laumond, J.-P. (2001) "Smooth Motion Planning for Car-Like Vehicles," *IEEE Trans. on Robotics*, vol. 17, pp. 498-502.

Mühlhäuher, M. and Gurevych, I (2008), *Ubiquitous Computing Technology for Real-Time Enterprises*. USA: Information Science Reference.

Nguyen, A.D., Ngo, V.T., Ha, Q.P. and Dissanayake, G. (2008) "Robotic Formation: Initialisation, Trajectory Planning, and Decentralised Control," *International Journal of Automation and Control*, Vol. 2, No. 1, pp. 22-45.

Remagnino, P and Foresti, G. L. (2005) "Ambient Intelligence: A New Multidisciplinary Paradigm," *IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, pp. 1-6.

Ren, S. and Wang, Y. (2008) Separating Reflection Components of Smooth Metallic Surface Using Special Random Sampling Method, In *Proc. IEEE Intl. Conf. on Innovative Computing Information and Control*, pp. 527-532, Kaohsiung, Taiwan, 2008.

Salvador, E., Cavallaro, A., and Ebrahimi, T (2001) "Shadow Identification and Classification Using Invariant Color Models," *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing*, Salt Lake, USA, pp. 1545-1548.

Stubbs, A., Vladimerou, V., Fulford, A. T., King, D., Strick, J., and Dullerud, G. E (2006) "Multivehicle Systems Control over Networks: a hovercraft testbed for networked and decentralized control", *IEEE Control System Magazine*, Vol. 26. No. 3, pp. 56-69.

Yu, Y.-H., Kwok, N. M and Ha, Q. P. (2009a) "FGPA-based real-time color tracking for robotic formation control," *Proc. Intl. Symp. Automation and Robotics in Construction*, Austin, Texas, US, 24-27 June 2009, pp. 252-258.

Yu, Y.-H., Ha, Q. P., and Kwok, N. M. (2009b) "FPGA Based Real-time Colour Discrimination Design for Ubiquitous Robots," *Proc. of the* 2009 *Australasian Conference on Robotics and Automation*, Sydney, Australia, 2-4 December 2009, 6 pages.

Yu, Y.-H., Kodagoda, S., and Ha, Q. P. (2010) "FPGA-Based Relative Distance Estimation for Indoor Robot Control Using Monocular Digital Camera," *Journal of Advanced Computational Intelligence & Intelligent Informatics*, to appear.

Wang, X., Lin, W., and Xue, P (2005) "Demosaicing with Improved Edge Direction Detection," *Proc. IEEE Intl. Symp. on Circuits and Systems*, , Kobe, Japan, pp. 2048-2051.