FPGA Based Real-time Color Discrimination Design for Ubiquitous Robots

Ying-Hao Yu[†], **Q.P. Ha**[†]
[†]School of Electrical, Mechanical
and Mechatronic Systems
University of Technology, Sydney
Broadway, NSW 2007, Australia
{YingHao.Yu, quangha}@eng.uts.edu.au

N.M. Kwok[‡]
[‡]School of Mechanical
and Manufacturing Engineering
The University of New South Wales
Sydney, NSW 2052, Australia
nmkwok@unsw.edu.au

Abstract

Detecting objects by their color is essential in robotic sensing. A simple way is probably to capture images with a digital camera and then using algorithms to determine colors of an input image. However, this approach is always affected by some issues such as light reflection and shadows on the object's surface. Although some sensory methodologies for color detection are available, they might not be a good choice in a ubiquitous robotic system, whereby real-time computation ability and low power consumption are of utmost importance. Too complicated image processing techniques will cause some practical problems when implemented in an embedded system. Here, a real-time algorithm is presented for color detection which can be used with the field programmable gate array (FPGA) technology. By utilizing the proposed approach implemented in circuit designs, real-time color discrimination can be achieved with good performance as demonstrated in this paper.

1 Introduction

The ubiquitous robot (Ubibot) concept [Kim, 2006] is based on ambient intelligence with pervasive computing or ubiquitous computing (UC) proposed in 1991 by Mark Weiser [Weiser *et al.*, 1991]. Accordingly, the human society in the future will be surrounded by the services of computer systems that are smaller in size, more connected and integrated into the environment, to help people carry out daily activities and routines in an easy and natural way using intelligent interfaces [Muhlhauser and Gurevych, 2004]. Unlike controlled by a conventional central computer system, the user in the future can have services from a network of computing nodes. Identification, location, sensing and connectivity are the key elements for individual ubiquitous computing nodes in a pervasive computing environment. Realising this blueprint will rely on a lot of small embedded systems with lower power consumption and more versatile real-time abilities than with the general purpose computer. This novel technology has been increasingly applied to tackle problems in the ambient intelligence context [Basten *et al.*, 2004].

Ubiquitous robots serve as a good demonstrator for the UC society. The mobile platform carrying multi-sensory functions and intelligent information can respond to different clients in a broader navigation range than the stationary server system [Kim, 2006]. However, in the contrast, Ubibot has more critical requirements to be satisfied in a battery-powered system than the stationary computer. The solution to the limited power capacity problem has become the first priority before the sensory devices and computing units are being designed [Basten *et al.*, 2004].

For the robot vision subsystem, color discrimination is one of the essential perception abilities for a robot. It is not only helping the robot to recognize the objects by a specific color but also helps the central surveillance system to track robots by their color labels [Vladimeros *et al.*, 2006]. The challenge to track objects by their color arises from the non-ideal representation of color which are always corrupted with different interferences from reflection and shadow [Ren and Wang, 2008]. The reflection can be estimated from the physical phenomenon of diffusion and specular reflection. The diffusion is from the light penetrating the object's surface and re-reflected in multiple directions, and the specular reflection is known as the mirror-like reflection on the surface. The shadow can be also be classified into self and cast shadows [Salvador *et al.*, 2001]; the self shadow is the

shadow attached on the object's body that is not illuminated, while the cast shadow is projected by the object on the ground or other objects. Consequently, in real world practice, the color discrimination function becomes a complicated task by the occurrence of various surface color with interferences.

A survey of the literature on reflection and shadow removing technologies reveals that the earlier reflection removing methodology can be traced back to [Shafer, 1985]. The work reported uses the dichromatic reflection model to separate the spectral property, while an improved methodology with photometric linearization applied to remove the reflection from a smooth metallic surface can be found in [Ren and Wang, 2008]. The other methodologies for removing reflection and shadow images include comparing different statistic measures of edges between shadow and reflection [Tappen *et al.*, 2002], mitigating overlapped edges inside the reflection image via edge detection and comparing templates in a database [Levin *et al.*, 2004], extracting reflection and shadow from a sequence of moving pictures [Szeliksi *et al.*, 2000], or using a polarizing filter with kernel independent component analysis (KICA) to reduce reflection [Yamazaki *et al.*, 2006]. These complicated methodologies are based on the analysis with stationary pictures on a computer monitor that might not meet the real-time requirements for a mobile ubiquitous robotic system unless the embedded system's speed is tremendously increased.

When robotic vision is incorporated in a cooperative control framework, the performance of the overall system will rely critically on the high speed digital camera and the subsequent image processing procedure. However, high speed image processing always means more power consumption, so a feasible and effective algorithm is proposed in this paper to perform the real-time color discrimination function with hardware circuits designed using a field programmable gate array (FPGA) chip. Unlike our previous approach [Yu *et al.*, 2009], here the color discrimination quality is enhanced using an additional tuning parameter, implemented for dynamic adjustment of boundaries of light reflection and shadows. The rest of the paper is organized as follows. In Section 2, previous work on the demosaicing algorithm employed for Bayer pattern is reviewed. The color discrimination algorithm is presented in Section 3. The test result is presented in Section 4 and followed by the discussion in Section 5. Finally, a conclusion is drawn in Section 6.

2 Demosaicing

The digital image sensor array in our digital camera is composed of red, green, and blue (RGB) essential colors using a mosaic-like Bayer pattern [Lukac *et al.*, 2005], see Figure 1.

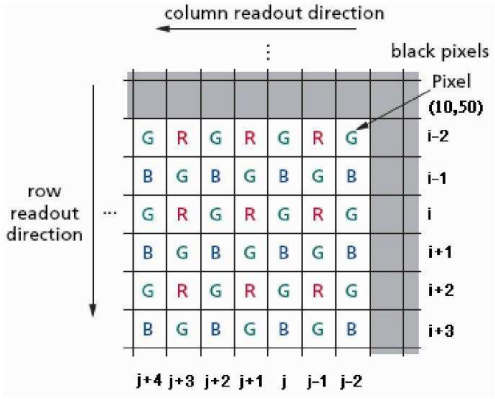


Figure 1: Bayer pattern formation and Scanning sequence of digital sensor array.

Based on the shift register design in the FPGA, a 9×9 raw image matrix for every image detection instant is then obtained with the four possible combinations, as shown in Figure 2, where the central pixels ($P5$) are read-out data during every operation instance.

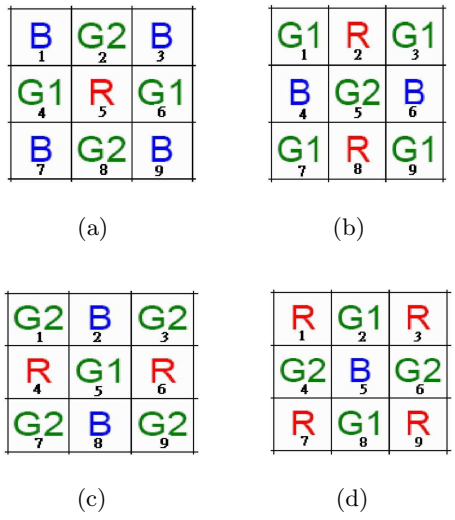


Figure 2: RGB combinations of a raw image by Bayer pattern.

Apparently, while every pixel on the digital image sensor array is only detecting a monochromatic color (red or green or blue), the full RGB colors of the central pixel should be interpolated from its neighboring pixels. This is known as demosaicing [Wang *et al.*, 2005].

To avoid a large power consumption caused by the

high speed transient signals, the demosaicing design is only synchronized with the pixel clock from the digital camera, and the demosaic procedure should be finished within every pixel clock period in order to meet the real-time requirement. For that, an easy technique is via fetching the missed color directly (randomly) from neighboring pixels. Unfortunately, this kind of real-time design will cause a saw-toothed distortion at the edge of the image. Therefore we suggest avoiding such distortion by averaging a pixel strength from its neighboring pixels. That is, we compute

$$G_{P5} = (G_{P4} + G_{P8})/2 \tag{1}$$

if the neighboring green pixels are in a rhombus formation and

$$R_{P5} = (R_{P2} + R_{P8})/2 \tag{2}$$

if the neighboring red pixels are in a column formation or else the red pixels are in a line formation:

$$R_{P5} = (R_{P4} + R_{P6})/2 \tag{3}$$

and

$$R_{P5} = (R_{P7} + R_{P3})/2 \tag{4}$$

for a square formation. Similarly, the blue color component has the same demosaicing rules as the red color.

This algorithm has least operations and takes into account the different light projection directions on the Bayer pattern. According to the simulated timing sequence results, an addition operator only spends 1/10 period of our camera's pixel clock, but it will need 1.5 pixel clocks for a division operation. Thus any averaged color strength from neighboring pixels will be updated for every two pixel clocks due to the fact that the digital circuit is designed for edge triggering.

3 Color Discrimination

Since the demosaiced real image I is from the combination of RGB essential colors, any pixel's full color strength can be represented by the magnitudes of $m_{R(i,j)}, m_{G(i,j)}$, and $m_{B(i,j)}$ as:

$$I_{(i,j)} = m_{R(i,j)} R_{sat} + m_{G(i,j)} G_{sat} + m_{B(i,j)} G_{sat} \tag{5}$$

where R_{sat}, G_{sat} , and B_{sat} denote the saturation color strength, i and j are the coordinates on a Bayer pattern, see Figure 1. In real practice, magnitudes m_R, m_G, and m_B are varied by different lighting conditions such as illumination and color of the light source, reflection, and shadow. Consequently, a surface with the same intrinsic color will be represented in several different RGB strengths.

With regard to the interference factors, the illumination and color of a light source can be determined easily by an appropriate sensing technology, and the specular reflection strength can be attenuated in the case of a rough surface. Therefore, if lighting conditions are controlled, the reflection and shadow can be considered as some variations of intrinsic colors. Moreover, for real-time image processing, the magnitude parameters in multiplication operations can be approximated by simpler addition and subtraction operations in different light strength thresholds.

In this paper, we provide an example to illustrate the detection of an object by using the green color. According to the demosaicing algorithm, the green color can be defined as:

$$G \triangleq \left(G_{(i,j)} - g_n + t_n\right) \geq \left(R_{(i,j)} \cap B_{(i,j)}\right) \tag{6}$$

where n is the number of threshold levels and g_n is the tolerance for the intrinsic green color. If any strength of $G_{(i,j)} - g_n$ is stronger than red and blue colors, a range of interested (ROI) green pixel is then decided [Ren and Wang, 2008], and a certain range of the green color strength taking into account the ROI pixel can be discriminated by using an adjustable parameter t_n.

Equation (6) reveals the ability to mitigate the disturbances of shadow and reflection if g_n and t_n are properly selected. However, the fixed boundary of the green color may be covered by stronger reflection and shadow. Thus, in order to reduce the discrimination error, a tuning scheme is proposed for parameter t_n to dynamically adjust the green boundary, which should be simple enough for implementation with the FPGA circuit design. Here, we assume that the property of the red color is stronger than the blue color in our light source. With $t_n^{(0)} = 0$ and for a given strength threshold level t_n^*, the tuning algorithm for t_n at the next pixel $(l+1)$ is then defined as:

$$t_n^{(l+1)} = \begin{cases} \left(t_n^{(l)} - G_{(i,j)}\right) + \left(R_{(i,j)} + g_n\right), \\ \qquad G_{(i,j)} \leq \left(R_{(i,j)} + g_n\right) \\ t_n^*, \qquad G_{(i,j)} > \left(R_{(i,j)} + g_n\right) \end{cases} \tag{7}$$

Apparently, t_n will enhance the discrimination accuracy if the boundaries of reflection and shadow do not change too abruptly.

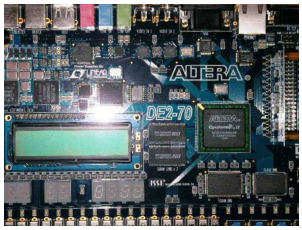
For the random green pixels on the undesired area, an additional simple noise filter can be implemented to exclude these random green pixels from a non-green color area. The following algorithm is proposed, which can be designed in the FPGA shift register structure:

$$G\langle 1|0\rangle = G_{(i+1,j)} \cap G_{(i+2,j)} \cap G_{(i+3,j)} \tag{8}$$

where $G\langle 1|0\rangle$ is a logic variable taking either value 1 or 0. If the green pixels appear at three successive row positions in the same column j, a logical true (1) will be set and one green pixel will be output.

4 Test Results

The proposed color discrimination design is implemented on the FPGA developing platform DE2-70 with Cyclone II FPGA from Altera, see Figure 3(a). The Logic Elements (LEs) in the Cyclone II chip has 68416 units, and the inbound RAM space is $1.1Mbits$. The utilized digital camera module, Figure 3(b), is of maximum $5M-pixel$ resolution assembled from Terasic.



(a)

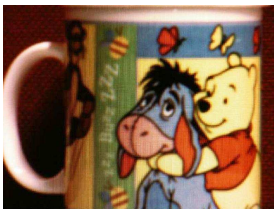
(b)

Figure 3: Test setup: (a) FPGA developing platform and (b) $5Mpixel$ camera.

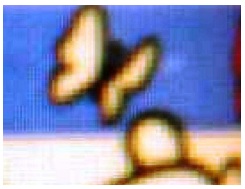
In this paper, we design the color discrimination circuit by setting a resolution of $1024 \times 1280(1.3M)$ pixels, and adjust the picture frame rate to reach 34 frames per second (fps) with a pixel clock of $77MHz$. For the surveillance purpose, an additional VGA interface to the FPGA is implemented, and two external on-board SDRAMs memory with $166MHz$ clock rate are then utilized as the image buffer for displaying images on the computer monitor. With this design, the total resource usages of LEs are about 4% of its maximum capacity.

The demosaicing test results are shown in Figure 4. The center pixel ($P5$) is updated at every pixel clock, but the averaged pixel strength at every two pixel clocks. Since the missed colors are interpolated directly from the neighboring pixels, saw-toothed distortions appears as can be seen in Figure 4(b). After using the proposed algorithm for neighboring pixels, the effect of saw-toothed distortion is reduced. Figure 4(c) shows that the same edges of the image are smoothed by averaging colors strength with neighboring pixels, as expected.

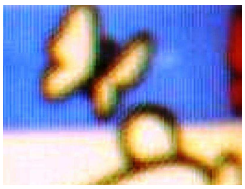
We further compare the green color discrimination ability by using the single threshold algorithm and the multi-threshold one in (6) and (7) with dynamic tuning. The test scenarios are set, respectively, with a rough green cloth and a smooth green miniature robot depicted in Figure 5(a) and 6(a). The detected green color area in these pictures are marked by white dots (pixels). In Figure 5, a green cloth is folded in a spiral shape, so the color inside the spiral cloth will be mixed with its own shadow and some diffuse reflections. The smooth



(a)



(b)



(c)

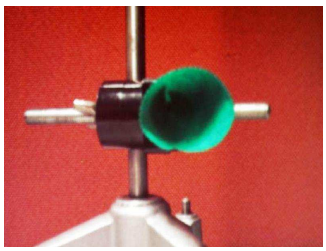
Figure 4: Demosaicing effect.

miniature robot shown in 6(a) is used to test with the specular reflection and cast shadow.

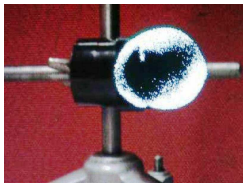
Figure 5(b) shows the worst case of color discrimination by using the single threshold algorithm including noise filter. As can be seen, a large black patch is spreading from the center of the spiral area. Meanwhile, the multi-threshold algorithm can discriminate the green color in different illumination levels, so the black patch mentioned is mostly filled by white pixels, as shown in Figure 5(c).

The different color discrimination abilities are also compared and results are illustrated in Figure 6. The green mobile robot has its smooth surface resulting in specular reflections from the background (red color) and metal posts on the top. The white marks shown in Figure 6(b) look loose when using the single threshold algorithm and noise filter. In contrast, the white pixels with the multi-threshold algorithm appear more solid, as shown in Figure 6(c).

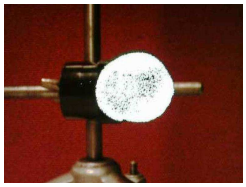
Finally, similar results between single and multi-threshold algorithms for the cast shadow and specular reflection case are shown in Figures 6(d) and 6(e). The light projection is blocked from the top with the blue object. Some weak specular reflection can also be observed, coming from the desk, top object, and background. The white marks when using the single threshold algorithm nearly vanish while the white color portions can still spread on the vehicle's body in the multi-threshold algorithm case.



(a)

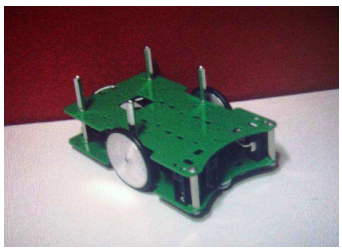


(b)



(c)

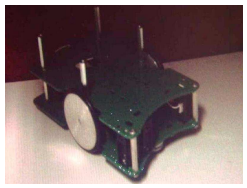
Figure 5: Color discrimination in the self shadow and diffuse reflection cases.



(a)

(b)

(c)



(d)

(e)

Figure 6: Color discrimination for specular reflection and cast shadow tests.

5 Discussion

By observing the demosaicing performance from the Bayer pattern, we find that although the division operations causes some time delay and requires updating the output pixel strength for every two pixel clocks, the interpolation algorithm with simple averaging can still provide satisfactory performance to smooth the saw-toothed distortion. Better demosaicing performance can be achieved by improving the time delay effect by using a higher speed circuitry.

Secondly, when proper values for the tolerance g_n and tuning parameter t_n have been chosen, the multi-threshold algorithm can track the desired color with a smaller effect of shadow and reflection on the object's surface. In addition, the real-time noise filter is designed and implemented by a shift register in the FPGA structure. These will improve the surveillance quality, i.e. to remove the random green pixels from the undesired color area in our case study.

Thirdly, the color discrimination with the proposed multi-threshold algorithm is shown to be capable of efficiently detecting in real time the desired color under different light illumination conditions. For the real-time image processing requirement, the multiplication operation given in equation 5 is replaced by addition and subtraction operations, which can be implemented in small and parallel computation units without an additional delay, compared to using a finite state machine approach in general. Hence, the color discrimination performed can be realized in every pixel clock period. Notably, this design helps to avoid additional power dissipation from high speed transient signals.

Finally, it is indicated through the observation of the overlapped images between green and white marked pixels mentioned above that by using the dynamic threshold algorithm, the pixels of a particular color can be accurately detected by following the ROI pixels unless the region is not for the desired color or shifted too much from an intrinsic color.

6 Conclusion

In this paper, we have proposed an FPGA based image processing design for demosaicing, color discrimination using a multi-threshold algorithm with dynamic tuning and noise filtering. With traditional methodologies, the real-time reflection and shadow removal with high pixel resolution camera and picture frame rate usually rely on the high speed computing system. In the proposed embedded system with a limited computing ability, comparable performance can be achieved with satisfactory real-time requirements, particularly for ambient intelli-

gence systems, where low power consumption is a constraint. The test results have demonstrated feasibility of the proposed technique for color discrimination with promising application to ubiquitous robots in the future.

7 Acknowledgement

Support from the Center of Excellence for Autonomous Systems, funded by the ARC and the NSW Government, is gratefully acknowledged. The first and second authors would like to acknowledge also support from UTS Research Grant 20006000848.

References

[Basten *et al.*, 2004] T. Basten, M. Geilen, and H de. Groot. *Ambient Intelligence: Impact on Embedded System Design*. Kluwer Academic, USA, 2004.

[Kim, 2006] J. H. Kim. Ubiquitous robot: Recent progress and development. In *Proc. IEEE/SICE Intl. Conf on Digital Object Identifier*, pages 25–30, Busan, Korea, 2006.

[Levin *et al.*, 2004] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 306–313, Washington, USA, 2004.

[Lukac *et al.*, 2005] R. Lukac, K. N. Plataniotis, and D. Htazinakos. Color image zooming on the bayer pattern. *IEEE Trans. on Circuits and Systems for Video Technology*, 15(11):1475–1492, 2005.

[Muhlhauser and Gurevych, 2004] M. Muhlhauser and I. Gurevych. Ubiquitous computing technology for real time enterprise. *Information Science Reference*, pages 1–20, 2004.

[Ren and Wang, 2008] S. Ren and Y. Wang. Separating reflection components of smooth metallic surface using special random sampling method. In *Proc. IEEE Intl. Conf. on Innovative Computing Information and Control*, pages 527–532, Kaohsiung, Taiwan, 2008.

[Salvador *et al.*, 2001] E. Salvador, A. Cavallaro, and T Ebrahimi. Shadow identification and classification using invariant color models. In *Proc. IEEE Intl. Conf. on Acoustic, Speech, and Signal Processing*, pages 1545–1548, Salt Lake, USA, 2001.

[Shafer, 1985] S. Shafer. Using color to separate reflection components. *Color Research and Application*, 10(4):210–218, 1985.

[Szeliksi *et al.*, 2000] R. Szeliksi, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 246–253, 2000.

[Tappen *et al.*, 2002] M. Tappen, W.T. Freeman, and E.H. Adelson. *Recovering intrinsic images from a single image*, volume 15 *Advances in Neural Information Processing Systems*, 2002.

[Vladimeros *et al.*, 2006] V. S. Vladimeros, A. T. Fulford, D. King, J. Strick, and G. E. Dullerud. Multivehicle systems control over networks: a hovercraft testbed for networked and decentralized control. *IEEE Control System Magazine*, 26(3):56–69, 2006.

[Wang *et al.*, 2005] X. Wang, W. Lin, and P. Xue. Demosaicing with improved edge direction detection. In *Proc. IEEE Intl. Conf. on Circuits and Systems*, pages 2048–2051, Kobe, Japan, 2005.

[Weiser *et al.*, 1991] M. Weiser. The Computer for the 21st century. *Scientific American*, 265(3):94–101, 1991.

[Yamazaki *et al.*, 2006] M. Yamazaki, Y. Chen, and G. Xu. Separating reflections from images using kernel independent component analysis. In *Proc. IEEE Intl. Conf. on Pattern Recognition*, pages 194–197, 2006.

[Yu *et al.*, 2009] Y.-H. Yu, Q.P. Ha, and N.M. Kwok. Chip-Based Design for Real-time Moving Object Detection using a Digital Camera Module. In *Proc. 2nd Intl. Congress on Image and Signal Processing*, Tianjin China, pages 1993–1997, 2009.