Introduction and Initial Exploration to an Automatic Tennis Ball Collecting Machine

Jialiang Zhao, Hongbin Ma*, Jiahui Shi , Yunxuan Liu

Abstract— **This paper describes a preliminary development to an automatic tennis ball collecting machine. The framework adopts a roller to collect tennis balls when moving. Tennis balls are detected and tracked automatically by computer vision while hardware acceleration is used to improve efficiency. For navigation, fuzzy control and 6 ultrasonic sensors are used to avoid obstacles when approaching the targets. A web-based monitoring and teleoperation system is added for human-robot interaction.**

I. INTRODUCTION

There are more and more people start playing tennis nowadays [1]. According to statistics [2], there are more than 17 million tennis players in US. When playing tennis, a player commonly spend hours in practicing, however, during which collecting balls all around the tennis court can be a very tedious and time-consuming work. Some tools are invented to alleviate this problem.

A. Prior Work

There are some passive tennis ball collecting devices that can help players to pick tennis balls up. The most common tennis ball collecting tools can be seen in the market are passive tube-like gadgets, commonly called Tennis Tube. This kind of device is composed of a tube, used as the handle, which is also a container of tennis balls, and an end cap installed in the front end of the tube, used to collect balls. When working, users should hold the tube, approach a tennis ball, and push the tube towards the tennis ball and force it to be squeezed into the tube via the end cap. This kind of device is convenient and cheap, however, drawbacks of this design are obvious. For convenience and portability, its capacity is too limited, thus users should regularly empty the tube manually. What's more, users need to approach every tennis ball in the field, while there may be hundreds of them during practice. Thus this design can only free users from stooping down, but can't thoroughly solve this problem.

Mobile gadgets generally are more efficient than tube-like gadgets. This kind of device has a roller in its front end, and a handle is used to manipulate. The roller is used to collect tennis balls, one common design of which is shaped like a squirrel cage. The distance between two neighboring beams is slightly smaller than the diameter of a tennis ball. During working, user pushes this mobile gadgets, and the roller starts rolling because of friction of the ground. Tennis balls in front of the roller will be squeezed into it and will not go out because of the beams. The drawback of this design is that, users still need to approach every tennis ball, and it's still a time-consuming process, although much more efficient that tube-like gadgets. Thus the problem still remains unsolved.

There are also some research on autonomous or semi-autonomous electrical active tennis ball collectors, like RapidBallBoy, Har-Tru Ball Mower, and TENNIBOT. RapidBallBoy and Har-Tru Ball Mower both have electrical collectors. When it approaches a tennis ball, these devices will collect it automatically. Users still need to push these devices around the court. TENNIBOT is based on Computer Vision. It can automatically detect a tennis ball, approach it and finally pick it up. The main problem is that this robot doesn't have the ability to avoid obstacles and navigate itself.

B. Our Contribution

In this paper, we present a novel design of a fully autonomous tennis ball collecting machine, which can automatically detect, track, collect tennis balls and navigate itself in tennis court. The visual part is achieved by a RGB camera. The navigation part is achieved by 6 ultrasonic distance sensors. A web sever is built for human-robot interaction.

The framework of this robot is described in Section II. Software part, including the visual algorithms and the web server, is described in Section III. Control strategy is presented in Section IV. Performance and tests of different configurations are discussed in Section V.

Arch Linux is chosen as the operating system. The design approach of the development team follows the KISS principle ("keep it simple, stupid") as the general guideline, and focuses on elegance, code correctness, minimalism and simplicity [3]. It is easy for developers to customize their system to the maximum [4].

The open source robotic framework, Robot Operating System (ROS) [5], is adopted as the communication tool in this system. This framework provides a low-latency communication platform, many useful tools and protocols [6]. Indigo Igloo is the first Long-term Support (LTS) distribution of ROS, thus it is chosen for stability.

This research was partially supported by the National Natural Science Foundation of China under Grant No. 61473038, and the Training Program of the Major Research Plan of the National Natural Science Foundation of China No. 91648117

To whom all correspondences should be addressed. Email: mathmhb@bit.edu.cn

The authors are with School of Automation, Beijing Institute of Technology, Beijing, China, 100081. Hongbin Ma is also with State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing, China, 100081.

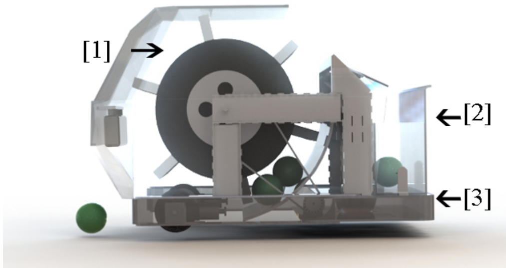


Fig. 1: Schematic of the tennis ball collecting machine. Part [1] is the roller, which is used to collect tennis balls. Part [2] is the ball container, which has a capacity of about 20 tennis balls. Part [3] is the chassis. During working, the roller starts rolling when it approaches a tennis ball, and the brushes installed on it will sweep the balls into the container. The outer shell is made by acrylic. A camera is installed on the front of it, and 6 ultrasonic sensors are installed around it.

II. Framework

A. Mechanical Structure

Carbon fibers (CF) are popular in mechanical structures for their high stiffness, high tensile strength, low weight, and high temperature tolerance [7]. Thus the main mechanical framework is based on CF.

The framework consists of 3 main components, a roller, a chassis, and a tennis ball container.

1) Roller: The roller illustrated in Fig.2 is used to collect tennis balls. There are 6 brushes, made by nylon, installed on it, which are used to sweep tennis balls. When the robot approaches a tennis ball, the roller starts rolling. The brushes will sweep the target into the container.

The roller is driven by a 12V DC 60RPM motor installed on the chassis. They are connected via a pair of pulleys and a drive belt.

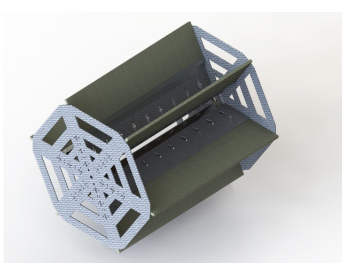


Fig. 2: The roller has a length of 400mm, and a diameter of 350mm. In order to guarantee that the tennis ball won't be missed, the front end of the bottom brush touches the ground when it's in a position.

2) Chassis: The chassis is made by two CF plates, which are connected to each other with some copper pillars. It is shaped like a character *U*. The *mouth* is used to install the roller. The two side bars of the *mouth* are used to place the two front motors. These two bars are narrow, in order to make a larger *mouth*, which will improve efficiency during working. Dynamic system, control boards and batteries are placed between the two plates.

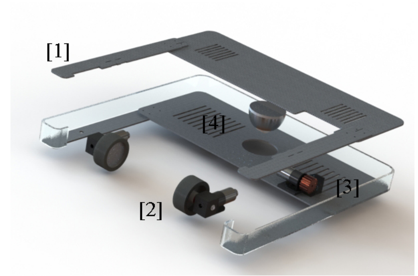


Fig. 3: The chassis. Part [1] is the U-shaped CF plate. Part [2] is the dynamic set. Part [3] is a motor used to drive the roller. Part [4] is an universal driven wheel.

3) Container: The container is made by acrylic, which is lite, cheap and beautiful. It is installed behind the roller and on the chassis. When a ball is collected, it will go into the container with the force of the brushes.

B. Dynamic System

The robot is powered by two T-shaped 12V DC motors. This kind of motor is widely used. Some manufactures, like TAMAGAWA^{TM}, maxon motor^{TM}, and CM motors^{TM} produce them. In this robot, 31ZY from CM motors^{TM} are used. Two encoders are installed in each motor for velocity measurement. These two electrical motors are separately installed on each of the side bar of the chassis. In the rear end of the chassis, a driven wheel is installed. Thus both direction and speed are determined by the two front driving wheels.

The physical robot is shown in Fig.4.

Fig. 4: The physical robot.

III. Software

This section consists two parts, the visual detection part and the web-based human-robot interaction part.

A. Visual Detection

In order to detect tennis balls and find their position, a high-speed and low-latency tennis ball detecting system is needed. What's more, this system should be able to accurately locate several tennis balls at a time.

Tennis balls can be detected by a variety of methods, e.g. wireless modules that are inserted in tennis balls in order to send coordinates; however, this kind of modules need power, and charging them frequently is unrealistic. Also, installing this kind of module in every tennis ball costs much. There are also researchers who let the collecting machine run randomly to traverse the target field, which is of low efficiency. Especially in tennis courts, traversal costs too much time [8].

Vision-based detecting system needs only a processor and a camera, and an efficient set of algorithms can locate tennis balls immediately. With the rapid development of CV, more and more efficient algorithms arise.

Before the recognition algorithm, some pre-processing procedures are necessary.

1) Convert Color Space: The native color space of the video stream captured by camera is the RGB color space. Because the color of a tennis ball is one of its most significant features, we need to do color splitting in the following step. HSV color space is more efficient in extracting a color block's hue, and less sensitive to different light conditions [9], so we convert the video stream from RGB to HSV.

After that, three thresholds are set to Hue, Saturation and Value respectively. It's obvious that the performance of detection is hugely influenced by these color thresholds. Thresholds with larger range will provide better performance in different environments with different light intensities, but will decrease the accuracy.

2) Blur: After picking the possible color blocks out, we obtain a gray-scale mask. In this mask, shade of a block indicates the possibility of this block to be a component of a tennis ball. Because of the uncontrollable quality of cameras, light conditions, and mottle on a tennis ball, a noise elimination procedure is needed. We add a median blur filter into the mask. Median blur can protect the image edges and is very effective in smoothing sharp noise [10].

3) Shape Identification: A pixel is treated to be a possible block if it's white, while dark means it belongs to background. Next we determine whether a cluster of possible blocks indicates a tennis ball by measuring its shape. Two algorithms are adopted in this procedure: Suzuki85 and Hough Transform.

- *Suzuki85:* Suzuki85 algorithm determines the surroundness relations among the borders [11] of a binary image. Firstly we use this algorithm to find every contour of the mask image. A result is shown in Fig.5.

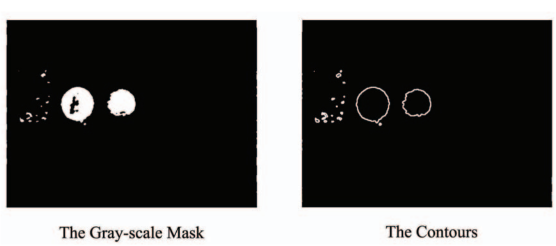


Fig. 5: Contours extracted from the mask.

Then we calculate the area of every contour picked from the mask. In this procedure, Green's Theorem is used. Green's Theorem gives Eq.1.

$$\oint_C (Ldx + Mdy) = \iint_D (\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y})\, dx\, dy \qquad (1)$$

where C is a simple close curve, D is a double integral over the plane region, L and M are functions of (x, y) defined on D that have continuous partial derivatives [12]. From Green's Theorem, the area of C is given by Eq.2.

$$S = \oint_C (Ldx + Mdy) \qquad (2)$$

when $\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} = 1$ [13]. Ideally, the contour of a tennis ball is a circle whatever the shooting angle is, so the area and the perimeter have a relationship of Eq.3;

$$\frac{C}{2\pi} = \sqrt{\frac{S}{\pi}} \qquad (3)$$

where C denotes perimeter and S denotes area. Algorithm.1 is adopted to identify a contour.

Algorithm 1 Contour Identification

Input: sample contours $\{\xi_k\}$ of a total of n contours
Output: $Con\{\xi_k\}$ that indicates whether $\{\xi_k\}$ is a circle

```
for k = 1 to n do
    if C{ξ_k} and S{ξ_k} is consistent with Eq.3 then
        Con{ξ_k} = true
    else
        Con{ξ_k} = false
return Con{ξ_k}
```

- *Hough Transform:* Hough Transform is another algorithm to determine a contour's shape. It is a method to detect curves by exploiting the duality between points on a curve and parameters of that curve [14]. We use function from OpenCV to apply this algorithm.

Compared with Hough Transform, Suzuki85 algorithm runs faster but is less accurate. We use Suzuki85 first, and when a contour is detected to be a circle, we use Hough Transform on this small region to confirm it. One result in medium light intensity is shown in Fig.6, and the target tennis ball is labeled.

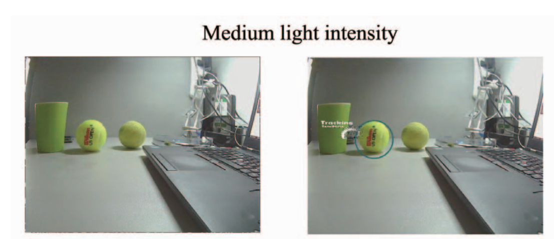


Fig. 6: The closest (biggest) tennis ball is marked, while the green cup is ignored.

B. Hardware Acceleration

Our image processing procedure takes lots of computing resources. This procedure is highly repeatable, thus parallel computing can significantly improve the performance [15]. Open Computing Language (OpenCL) is used in this system. OpenCL is a framework for writing programs that are executed on across heterogeneous platforms [16]. We use it to run some of our image processing algorithms in Graphics Processing Units (GPUs).

Several computing kernels are set up to compute independently. Hough Transform and median blur are computing with hardware acceleration.

After hardware acceleration, the performance is significantly improved compared with computing only with CPUs. Some experiment results are listed in Table I. With each computing method, three different scenes are tested. The test environment is 64-bit Intel Atom E3826 (dual-core, 1.46 GHz) with Intel HD Graphics Integrated, 2GB DDR3 RAM. OpenCL 2.0 and OpenCV 2.4.13.

TABLE I: Results of Hardware Acceleration

Scene	No.	CPU	GPU	FPS
Scene 1	1	√	×	73
	2	√	√	137
Scene 2	1	√	×	47
	2	√	√	110
Scene 3	1	√	×	24
	2	√	√	89

C. Web-based Human-robot Interaction

To have better interactions with this robot, we designed a web-based interface to control and monitor it. A python micro web framework, namely flask, is chosen to build the server for it's light and convenient.

The remote monitoring system should be real-time and of low-latency, to achieve that, we choose Socket.IO as the transmission media. Socket.IO is the cross-browser WebSocket for real-time applications, it enables real-time bidirectional event-based communication [17]. By using Socket.IO the system will be able to communicate between front-end and back-end in real-time. The whole process is shown in Fig.7. Based on C/S framework, Socket.IO is driven by event.

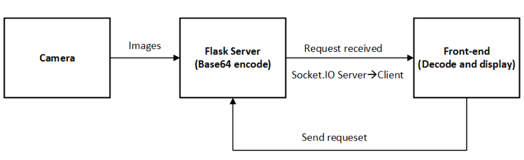


Fig. 7: Remote Monitoring System

In our system, when users open the web the first time, a request for images will be sent from front-end to back-end. When receiving the request, the back-end server will collect images from the camera. After encoding them in Base64 format, image codes will be sent to the front-end

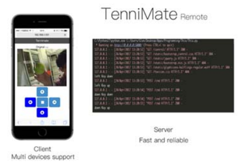


Fig. 8: Image Streaming

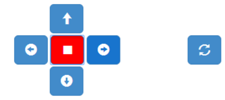


Fig. 9: Controlling Buttons

by Socket.IO. Then the Base64 codes will be decoded and displayed at the front-end, and a new request for next frame will be sent to server. The whole communication process will last until users close the web. The request-reply design makes full use of the characteristics of Socket.IO, which guarantee the low latency in transmission. Another advantage of this design is that the server sends images only if users visit the web. The transmission will be shut down when users close the web.

Based on the real-time images, users can teleoperate robots via buttons (shown in Fig.9) on web. The left five buttons send moving commands (in four directions) and stop command. The right button is used to start or stop the collecting roller. When a button is pressed, the corresponding color will be changed and a "post" request will be sent from the front-end and detected at the back-end. Commands will then be sent to ROS and robots will move accordingly.

IV. Navigation

We adopt 6 ultrasonic sensors and fuzzy control algorithm to navigate.

A. Sensors

One of the major challenges of the autonomous navigation for mobile robots is to detect obstacles during the robotic navigation task. Considering the main obstacles are humans and chairs on tennis court, we choose to use ultrasonic sensors because they are cheap and accurate enough in this specific environment. We use 6 ultrasonic sensors separated by 36° to get the real-time distance information. For each sensor, it can measure distance between 2cm to 4m with 15° measuring angle. The distribution of ultrasonic sensors is shown in Fig.10.

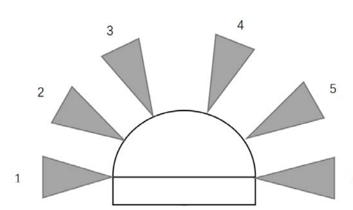


Fig. 10: Sensors Distributions

B. Fuzzy Control

In real-time navigation systems, controllers must be robust enough to deal with different situations, fuzzy control is therefore introduced in our system. Fuzzy control systems are rule-based or knowledge-based systems containing a collection of fuzzy IF-THEN rules based on the domain knowledge or human experts [18]. The process of fuzzy logic is shown in Fig.11.

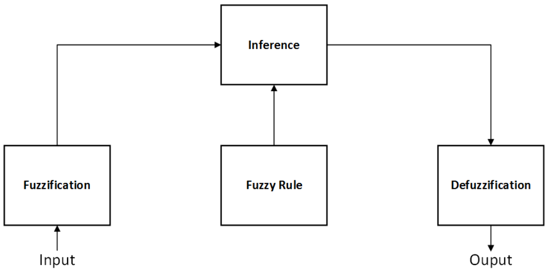


Fig. 11: Fuzzy Controller Structure

1) Fuzzyfication: In our fuzzy control system, variables are shown in Table II. The fuzzyfication process is to transform the continuous values variables into linguistic variables, which is implemented by member functions shown in Fig.12. TableIII shows the meaning of each linguistic variables.

TABLE II: Variables defined in fuzzy control

Variable	Input/Output	Description
LD	Input	Distance from left
FD	Input	Distance from front
RD	Input	Distance from right
θ	Input	Angle to destinations
LV	Output	Output velocity for left wheel
RV	Output	Output velocity for right wheel

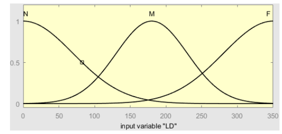
TABLE III: Linguistic Variables

Continuous Value Variables	Linguistic Variables	Description
Distance	N	Distance is near
	M	Distance is middle
	F	Distance is far
Angle	P	Angle is from 0° to 60°
	Z	Angle is from 60° to 120°
	N	Angle is from 120° to 180°
Velocity	S	Velocity is very slow
	MS	Velocity is slow
	MF	Velocity is fast
	F	Velocity is very fast

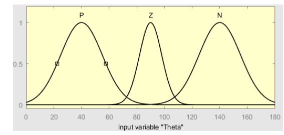
2) Fuzzy Rules: Fuzzy rules are built based on knowledge. To simplify the control rules, we make a transformation to the sensor information.

$$\begin{aligned} LD &= min(Sensor[1], Sensor[2]) \\ FD &= min(Sensor[3], Sensor[4]) \\ RD &= min(Sensor[5], Sensor[6]) \end{aligned} \tag{4}$$

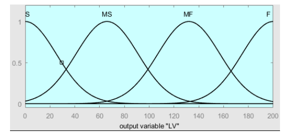
For each input variable, it is divided into three parts, so the total number of rules can be reduced to $3^4 = 81$. One part of rules is shown in Table IV. Taken rule1 as an example, it can be expressed as *"If (LD is F) and (FD is F) and (RD is F) and (Theta is N) then (LV is MS)(RV is F)"* in fuzzy logic rules, which means if distance from 3 directions are all



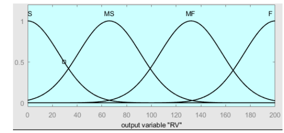
Distance MF



Theta MF



Left Velocity MF



Right Velocity MF

Fig. 12: Member Function

big enough and the destination angle lies between 120° and 180 °, we will set LV to be slow and RV to be very fast, so the robot will turn left to reach the destination.

TABLE IV: Variables defined in fuzzy control

Number	LD	FD	RD	θ	LV	RV
1	F	F	F	N	MS	F
2	F	F	F	Z	F	F
3	F	F	F	P	F	MS
4	F	F	M	N	MS	F
5	F	F	M	Z	F	F
6	F	F	M	P	MF	F
...	...	...	...	...	...	...
81	N	N	N	P	MS	S

C. Defuzzification

We use centroid method [19] to defuzzify the member functions for outputs, the output velocities are defined by

$$v = \frac{\int_0^{200} v\mu_D(v)dv}{\int_0^{200} \mu_D(v)dv} \tag{5}$$

where $\mu_D(v)$ is the output member function.

V. Experimental Results

We tested different sets of configuration in the detecting system and the navigation system. Stability and regulating speed are used as criterion during tests.

A. Detecting System

The sensitivity of the detecting system is fundamentally determined by the color threshold and the area threshold. As mentioned early, the color threshold is used to determine whether a color block is a part of a tennis ball. The wider color threshold will show better performance in different conditions with different light intensities. As a result, the detecting accuracy will decrease. The area threshold indicates the minimum area of a possible contour. Smaller area threshold will show better performance when a tennis ball is far away from the robot. However, if this threshold is too small, noises may be erroneously treated as tennis balls. We tested different sets of these two parameters. The best parameters are shown in Table V

TABLE V: Values of HSV threshold

Parameter	H_{th_sub}	H_{th_sup}	S_{th_sub}	S_{th_sup}
Value	43	55	107	240
Parameter	V_{th_sub}	V_{th_sup}	$Area_{th\_sub}$	$Area_{th\_sup}$
Value	30	250	200	1800

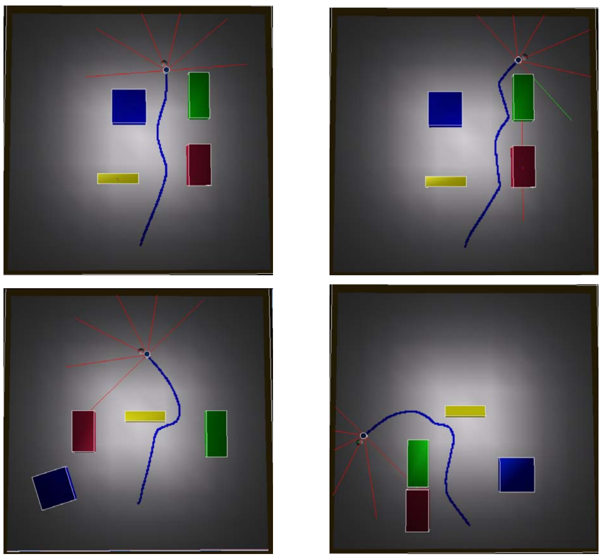


Fig. 13: Simulation Results

B. Navigation System

Webot Pro simulation software is used to test the controller. We build a world and a model for this robot, which is equipped with 6 ultrasonic sensors ranging from 0 to 350mm. Control program is written in MATLAB^{TM} to drive the simulation.

Fig.13 lists some of the most common situations on tennis court, the gray ball is set as destination and the blue footprint is the track of robot's movement. Red lines on robot is the range for ultrasonic sensors. We can see that this robot can collect the ball successfully without running into the obstacles.

VI. Future Work

This machine can now collect tennis balls by one camera. In the future, we are going to add more cameras and extend the collecting functions to other balls.

A. Collecting in Various Ball Games

This vision-based automatic control system can be applied into different ball games, like table tennis, golf and badminton. Theory behind these different functions are similar, thus we are going to make the image processing node able to detect other balls.

B. Multi-camera Support

Now this collecting machine only has one camera installed, and only the scenes in front of this machine can be observed. We are going to add multi-camera support, for example, one camera in each direction, to improve the efficiency. If the collecting machine can observe scenes from every direction at a time, its decision will be more intelligent and the collecting process will be faster.

VII. Conclusions

In this work, we have established a basic vision-based automatic tennis ball collecting system. This system is able to detect tennis balls and control the collecting machine to move towards them and then pick them up. Hardware acceleration and timing synchronization are applied to improve the performance. A web-based remote monitoring and controlling system is used for users to monitor the collecting machine's working state and environment. This web page also provides a method of manual intervention. Such a system may help tennis participants to collect tennis balls and save their time. In future work, we will make this system capable of collecting more balls rapidly in various ball games and add multi-camera support to improve efficiency.

References

[1] E. Wilson, *Love Game: A History of Tennis, from Victorian Pastime to Global Phenomenon*. University of Chicago Press, 2016.

[2] Statista. (2016) Number of participants in tennis in the united states from 2006 to 2014 (in millions). [Online]. Available: http://www.statista.com/statistics/191966/participants-in-tennis-in-the-us-since-2006/

[3] (2009) The arch way. [Online]. Available: http://wiki.archlinux.org

[4] I. Devolder, *Arch Linux Environment set-up How-To*. Packt Publishing, 2012.

[5] (2014) Robot operating system(ROS). [Online]. Available: http://wiki.ros.org

[6] S. Edwards and C. Lewis, "Ros-industrial: applying the robot operating system (ros) to industrial applications," in *IEEE Int. Conference on Robotics and Automation, ECHORD Workshop*, 2012.

[7] J.-B. Donnet and R. C. Bansal, *Carbon fibers*. CRC Press, 1998.

[8] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," in *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1. IEEE, 1994, pp. 610–617.

[9] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with hsv color information," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 334–339.

[10] D. Zhang, Y. Xue, X. Ye, and Y. Li, "Research on chipsdefect extraction based on image-matching," *International Journal on Smart Sensing and Intelligent Systems*, vol. 7, no. 1, pp. 321–336, 2014.

[11] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[12] G. Green, *An essay on the application of mathematical analysis to the theories of electricity and magnetism*. author, 1828.

[13] J. Stewart, *Calculus*. Cengage Learning, 2015.

[14] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[15] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to parallel computing: design and analysis of algorithms*. Benjamin/Cummings Redwood City, CA, 1994, vol. 400.

[16] A. Munshi, "The opencl specification," in *2009 IEEE Hot Chips 21 Symposium (HCS)*. IEEE, 2009, pp. 1–314.

[17] R. Rai, *Socket. IO Real-time Web Application Development*. Packt Publishing Ltd, 2013.

[18] S. H. Tang, D. Nakhaeinia, and B. Karasfi, *Application of Fuzzy Logic in Mobile Robot Navigation*. InTech, 2012.

[19] L.-X. Wang, "Fuzzy systems are universal approximators," in *Fuzzy Systems, 1992., IEEE International Conference on*. IEEE, 1992, pp. 1163–1170.