

การพัฒนาระบบคอมพิวเตอร์วิทัศน์สำหรับแขนหุ่นยนต์ป้อนอาหารคนพิการ

Development of Computer Vision System for Food Feeding Robot Arm for the Disabled Person

สยาม เจริญเสียง และ บพิตร ช่วยคง

สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

126 ถ.ประชาธิปไตย แขวงบางมด เขตทุ่งครุ กทม. 10140

โทรศัพท์ 0-2470-9715 โทรสาร 0-2470-9714 E-mail : siam@fibo.kmutt.ac.th, bopid@hotmail.com

ABSTRACT - This research presents the development of computer vision to determine the locations of a spoon, a food plate, and a water glass along with the orientation of a spoon on the tray. The tracked information will be sent to a robot arm to feed food to the disabled person. The SIFT feature technique is applied to find the recognized objects. It can take care of lighting and image's rotation with noises. The proposed system can find the spoon within 1.7 seconds with accuracy of tracked location with less than ± 14 mm. In addition, the computer vision can send all tracked information to the robot arm in order to pick up the targeted object automatically.

KEYWORDS - SIFT Feature , Image Processing, Object Recognition

บทคัดย่อ - งานวิจัยนี้เป็นการพัฒนาระบบคอมพิวเตอร์วิทัศน์เพื่อใช้ในการหาตำแหน่งของช้อน จาน แก้วน้ำ และมุมที่ทำกับแนวนราบของด้ามช้อนบนถาดอาหาร เพื่อส่งข้อมูลดังกล่าวให้แก่แขนหุ่นยนต์สำหรับป้อนอาหารให้แก่คนพิการ โดยใช้หลักการ SIFT Feature ซึ่งสามารถประยุกต์ใช้งานได้ดีในระบบนี้ ซึ่งต้องมีการค้นหาวัดด้วยภาพที่มีการเปลี่ยนแปลงสภาพแสง โดยวัตถุมีการหมุนและมีจุดรบกวนภายในภาพ จากผลการทำงานของระบบพบว่าสามารถตรวจหาตำแหน่งช้อน ได้เฉลี่ยภายในเวลาไม่เกิน 1.7 วินาที และมีความถูกต้องของตำแหน่งพิกัดของวัตถุ วัดได้เฉลี่ยไม่เกิน ± 14 มิลลิเมตร นอกจากนี้ระบบคอมพิวเตอร์วิทัศน์ยังสามารถส่งตำแหน่งและมุมดังกล่าวเพื่อช่วยให้แขนหุ่นยนต์หยิบจับวัตถุที่ต้องการได้โดยอัตโนมัติ

คำสำคัญ - SIFT Feature การประมวลผลภาพ การจดจำวัตถุ

1. บทนำ

จากการสำรวจพบว่าในปี พ.ศ.2545 [1] มีจำนวนคนพิการประมาณ 1,098,000คนซึ่งจะเห็นได้ว่าคนจำนวนไม่น้อยที่ต้องทนยากลำบากในการดำรงชีวิตประจำวัน ต่อความ ด้วยเหตุนี้ผู้พิการต้องการความช่วยเหลือจากบุคคลอื่นหรืออุปกรณ์เข้ามาช่วยให้สามารถดำเนินชีวิตได้สะดวกขึ้นและมีคุณภาพชีวิตที่ดีขึ้น ได้มีการพัฒนาอุปกรณ์เพื่อช่วยอำนวยความสะดวกให้แก่คนพิการอย่าง

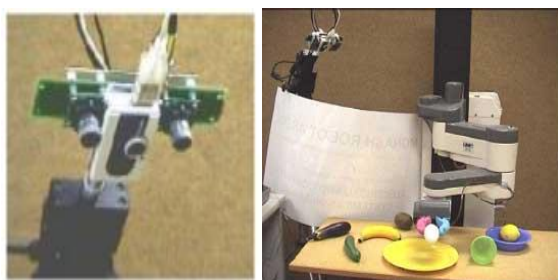
ต่อเนื่อง รวมทั้งมีการนำระบบหุ่นยนต์เข้ามาช่วยทำให้คนพิการสามารถดำรงชีวิตได้อย่างสะดวกขึ้น ซึ่งโดยทั่วไปได้มีการนำแขนหุ่นยนต์มาช่วยป้อนอาหารคนพิการแต่ส่วนใหญ่จะไม่เป็น

แบบระบบอัตโนมัติ โดยจะต้องมีการระบุตำแหน่งการเคลื่อนที่ให้แก่หุ่นยนต์ไว้ก่อนแล้ว หรืออาจจะเป็นแบบการควบคุมด้วยมือเพื่อสั่งงานแขนหุ่นยนต์โดยตรง ทำให้ขาดความสะดวกแก่ผู้พิการทางแขนที่ไม่สามารถใช้งานได้ต้องอาศัยผู้อื่นคอยช่วยเหลือ งานวิจัย

นี้จึงนำเสนอ การพัฒนาระบบคอมพิวเตอร์วิชั่นมาช่วยหาตำแหน่งของวัตถุ แล้วส่งข้อมูลดังกล่าวไปให้แขนหุ่นยนต์จะทำให้ระบบการป้อนอาหารคนพิการสามารถทำงานได้เองอย่างอัตโนมัติ

2. การทบทวนงานวิจัยที่เกี่ยวข้อง

ในปี ค.ศ.2004 คณะวิจัยของ Shahram Jafari และ Ray Jarvis จาก Monash University Australia ที่ศูนย์วิจัย Intelligent Robotic Research Centre (IRRC) ได้พัฒนาระบบ Eye-To-Hand Coordination [2] เป็นการศึกษา การทำงานระหว่างกล้องถ่ายภาพแบบสเตอริโอ ร่วมกับแขนหุ่นยนต์ชื่อ COERSU เพื่อให้สามารถหยิบจับสิ่งของที่มีลักษณะอ่อนนุ่มเช่น กล้วย แดงกวา บนโต๊ะได้ มีมือจับที่ติดเซนเซอร์รับแรง (Tactile Sensor) และหุ้มด้วยยางเพื่อช่วยให้มีความอ่อนนุ่มในการหยิบ มีการทำงานร่วมกันจาก การวิเคราะห์ภาพ 3 มิติ แบบสเตอริโอวิชั่นร่วมกับพีซีลอจิกมาช่วยหาดำแหน่งแล้วสามารถหยิบของได้



รูปที่ 1. หุ่นยนต์ COERSU กำลังหยิบวัตถุ [2]

หุ่นยนต์ COERSU ดังรูปที่ 1 สามารถหยิบจับวัตถุที่มีลักษณะยาว ได้ เช่น แดงกวา กล้วย ข้อมูลภาพจะถูกส่งเข้าไปให้คอมพิวเตอร์ประมวลผลภาพและหาดำแหน่งของวัตถุซึ่งเป็นส่วนที่ยาก เพื่อให้หุ่นยนต์สามารถหยิบวัตถุให้ได้ โดยเริ่มจากการวางวัตถุเป้าหมายให้กล้องจับภาพหาดำแหน่งให้ได้ เพื่อให้สามารถสร้างแผนที่วัตถุบนภาพได้ แสดงดังรูปที่ 2



รูปที่ 2. แสดงการวิเคราะห์ภาพและการจับตำแหน่งของวัตถุ [2]

หลังจากนั้นใช้การคำนวณหา Kinematics ของแขนหุ่นยนต์เพื่อใช้ในการเคลื่อนที่ไปหยิบวัตถุได้อย่างถูกต้อง โดยวิธีของ Jacobian Matrices การหยิบจะใช้ที่จับแบบคิปล 2 อัน หยิบตรงกลางของวัตถุที่ทำการประมวลผลภาพไว้



รูปที่ 3. แสดงรูปแบบการกำหนดจุดที่ต้องการหยิบ [2]

โดยหุ่นยนต์สามารถหยิบจับวัตถุได้ตามตำแหน่งที่ระบบประมวลผลภาพหาดำแหน่งเอาไว้ให้ โดยจะเน้นวัตถุที่เป็นลักษณะยาว เป็นตัวหลัก แสดงดังรูปที่ 3

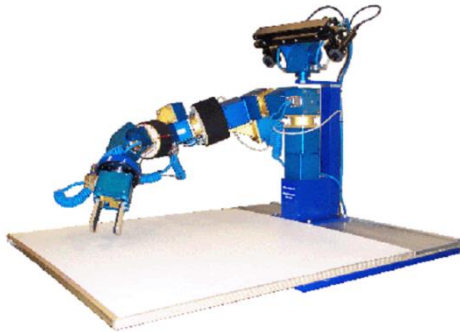
ในปี ค.ศ. 2006 Andrew Dankers ได้พัฒนาระบบ Realtime Stereo Active Vision [3] ซึ่งระบบสเตอริโอวิชั่นส่วนใหญ่จะเป็นแบบสเตติก-สเตอริโอวิชั่น (Static Stereo Vision) โดยในงานวิจัยนี้จะเป็นการพัฒนาแบบแอคทีฟ (Active) เป็นหลักและยังสามารถทำงานแบบสเตติก-สเตอริโอวิชั่นได้ด้วย ระบบแบบแอคทีฟคือกล้องสามารถเคลื่อนที่ขึ้น-ลง ซ้าย-ขวา เพื่อมองหากภาพได้ เริ่มจากการพัฒนาระบบที่ชื่อ CeDAR แสดงดังรูปที่ 4



รูปที่ 4. แสดงระบบ CeDAR [3]

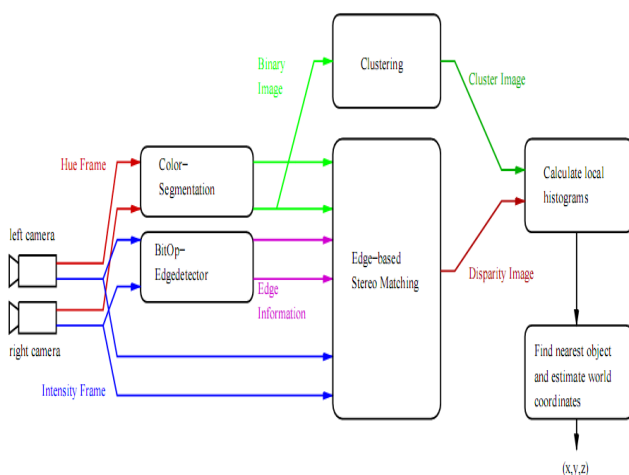
แนวคิดคือการใช้เทคนิคซ้อนภาพแบบเรียลไทม์ (Real-Time Active Rectification Technique) ในขณะที่กล้องกำลังเคลื่อนที่ เพื่อให้ภาพกลมกลืนกัน แล้วจะถูกประมวลผลอย่างต่อเนื่องจนกลายเป็นระบบแบบแอคทีฟ ปัจจุบันระบบ CeDAR สามารถตรวจหาวัตถุในภาพที่มีการเคลื่อนไหวได้ โดยกล้องจะเคลื่อนที่ตามวัตถุไปในเฟรมภาพที่เคลื่อนที่ได้อย่างรวดเร็ว

ในปี ค.ศ. 2002 Ioannis Lossifidis จาก Bochum University Germany ได้พัฒนาระบบ An Anthropomorphic Robot Assistant [4] โดยมีหุ่นยนต์ CORA ซึ่งเคลื่อนที่ไม่ได้ โดยจะยึดติดอยู่กับโต๊ะ แสดงดังรูปที่ 5



รูปที่ 5. แสดงหุ่นยนต์ CORA มีส่วนแขน 7 DOF และส่วนหัว 2 DOF [4]

มีการทำงานได้ตอบกับมนุษย์ที่นั่งอยู่ตรงข้ามกันบนโต๊ะ ส่วนหัวหุ่นยนต์จะมีการเคลื่อนที่ได้ 2 DOF แบบ Pan – Tilt ซึ่งส่วนหัวจะบรรจุกล้องสเตอริโอไว้ 2 ตัว เพื่อทำเป็นระบบสเตอริโอวิชั่นสามารถจดจำวัตถุบนโต๊ะได้และยังจำท่าทางของมือที่ให้อาหารได้ นอกจากนี้ยังสามารถวิเคราะห์ตำแหน่งวัตถุในระบบ 3 มิติได้ กล้องจะมีระยะห่างระหว่างกล้องถึง 300 มิลลิเมตร ซึ่งกว้างกว่าตาของมนุษย์ ทำให้สามารถเพิ่มความละเอียดในมุมมองได้มากกว่า มีความสามารถในการติดตามวัตถุ โดยถ้าพบมือ กล้องก็จะเคลื่อนที่ตามมือ แล้วล็อกตำแหน่งนั้นไว้ ข้อมูลการเคลื่อนที่ของมือก็จะถูกคำนวณส่งไปยังอุปกรณ์ควบคุมความเร็วและการเคลื่อนที่ของกล้อง ขบวนการทำงานของระบบดังกล่าวแสดงได้ดังรูปที่ 6



รูปที่ 6. แสดงขบวนการประมวลผลภาพของระบบ [4]

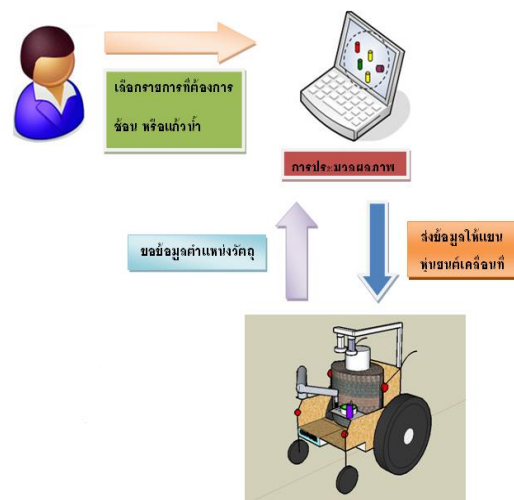
สิ่งแรกที่หุ่นยนต์จะต้องทำคือการจดจำวัตถุที่มีทั้งหมดก่อนเปรียบเสมือน เป็นการเรียนรู้ของหุ่นยนต์ เช่นการจำสี รูปร่าง ขนาด แล้วเอาเข้ามาคำนวณร่วมกับมุมมอง เป็นเรื่องเกี่ยวกับเรขาคณิต มุมของกล้องที่มองลงมาแบบ Bird's Eye View ตัวอย่างของการสั่งงานโดยใช้ท่าทางมือแสดงได้ดังรูปที่ 7



รูปที่ 7. แสดงการสั่งให้หุ่นยนต์หยิบของตามคำสั่งมือ [4]

3. ภาพรวมของระบบวิชั่น

ระบบการประมวลผลภาพจะทำงานร่วมกับแขนหุ่นยนต์ป้อนอาหารคนพิการ เพื่อหาตำแหน่งและทิศทางของวัตถุที่ต้องการจากภาพระบบสเตอริโอวิชั่น คือ ซ้อน งานข้าว และแก้วน้ำ เมื่อได้พิกัดของตำแหน่งซ้อน งานข้าว และแก้วน้ำ แล้วจะส่งข้อมูลผ่านทางซ็อกเก็ต (Socket Communication) ไปยังชุดโปรแกรมของแขนหุ่นยนต์ โดยให้ส่วนของ การประมวลผลภาพเป็นเซิร์ฟเวอร์ (Server) หรือตัวแม่ข่าย แล้วส่วน โปรแกรมของแขนหุ่นยนต์จะเป็นตัวลูกข่าย (Client) เพื่อให้แขนหุ่นยนต์ทำงานตามคำสั่งของผู้ใช้งานที่ต้องการให้ระบบเข้ามาหยิบจับแบบอัตโนมัติ ภาพรวมของระบบวิชั่นสามารถแสดงได้ดังรูปที่ 8

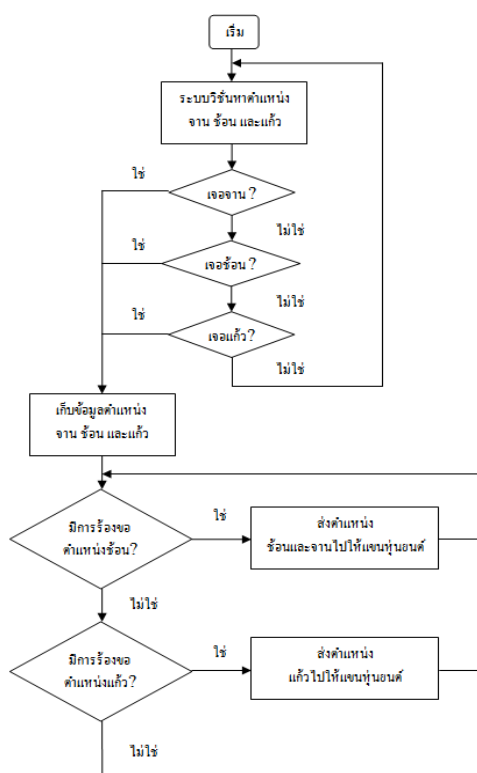


รูปที่ 8. แสดงภาพรวมของระบบ

การทำงานของระบบคอมพิวเตอร์วิชั่น เริ่มจากระบบจะประมวลผลภาพเพื่อหาตำแหน่งของจาน ช้อน และแก้ว ถ้าเจอจาน ช้อน และแก้ว ก็จะเก็บข้อมูลตำแหน่งเอาไว้ เพื่อรอการร้องขอตำแหน่งจากโปรแกรมแขนหุ่นยนต์ ถ้าขอตำแหน่งช้อนมา ก็จะส่งตำแหน่งช้อนและจานไปให้ ถ้าขอตำแหน่งของแก้วมา ระบบวิชั่นก็จะส่งข้อมูลตำแหน่งแก้วไปให้โปรแกรมแขนหุ่นยนต์โดยโปรแกรมการทำงานของระบบแสดงได้ดังรูปที่ 10



รูปที่ 9. แสดงกล้อง 2 ตัวที่ใช้ทำ สเตอริโอวิชั่น



รูปที่ 10. แสดงโปรแกรมการทำงานของระบบ

3.1 ส่วนประกอบด้านฮาร์ดแวร์

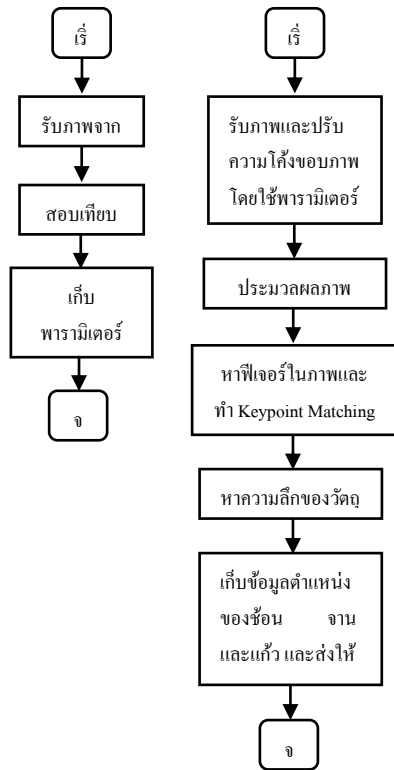
ส่วนประกอบที่สำคัญด้านฮาร์ดแวร์คือกล้อง 2 ตัว แสดงดังรูปที่ 9 ถูกติดตั้งอยู่บนรถเข็นคนพิการสำหรับสร้างภาพแบบสเตอริโอ กล้องมีระยะห่างระหว่างกัน 12 เซนติเมตร มีความยาวโฟกัส 3.7 มิลลิเมตร และใช้คอมพิวเตอร์ 1 เครื่อง ที่มีชิพ Intel Duo T9400 2.4 GHz และการ์ดจอรุ่น ATI Radeon HD3650 ติดอยู่หลังเบาะของเก้าอี้ เพื่อควบคุม การประมวลผลภาพ การติดตั้งของอุปกรณ์ด้านฮาร์ดแวร์ แสดงในรูปที่ 11



รูปที่ 11. แสดงอุปกรณ์ที่ติดตั้งบนรถเข็นคนพิการ

3.2 ส่วนประกอบด้านซอฟต์แวร์

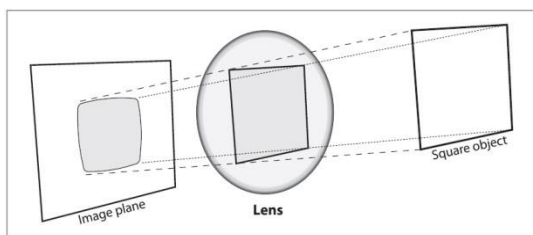
โปรแกรมภาษาซีพลัสพลัสของ Microsoft Visual Studio 2008 พร้อมด้วยซอฟต์แวร์ไลบรารี OpenCV 2.1 ได้ถูกนำมาใช้พัฒนาเพื่อ การประมวลผลภาพจากกล้อง โดยระบบจะต้องทำการสอบเทียบ ภาพจากกล้องก่อนแล้วนำพารามิเตอร์จากการสอบเทียบไปใช้ในการ ปรับขอบภาพให้ถูกต้องและทำการประมวลผลภาพเบื้องต้นเพื่อให้ ได้ภาพที่เหมาะสมขึ้น หลังจากนั้นระบบจะใช้หลักการ SIFT เพื่อหา ฟิเจอร์ในภาพและทำการเปรียบเทียบกับส่วนที่เหมือนกันใน ข้อมูลภาพต้นแบบ แล้วทำสเตอริโออิมเมจ เพื่อหาระยะความลึกของ วัตถุในภาพ หลังจากนั้นระบบวิชั่นจะเก็บตำแหน่งของจาน ช้อน และแก้ว เพื่อรอการร้องขอจากโปรแกรมควบคุมแขนกลต่อไป โปรแกรมการทำงานแสดงได้ดังรูปที่ 12



รูปที่ 12. แสดงโฟลว์ชาร์ตการทำงานของการทำงานประมวลผลภาพ

3.2.1. การสอบเทียบภาพจากกล้อง

ก่อนการประมวลผลภาพจากกล้องแบบสเตอริโอจะต้องทำการสอบเทียบภาพจากกล้องก่อน ด้วยการถ่ายภาพตารางหมากรุก เพื่อปรับแก้ค่าความโค้งของเลนส์ให้ถูกต้องก่อนดังรูปที่ 13 แล้วจึงนำค่าตัวแปรที่ได้ไปใช้ปรับแก้ภาพเพื่อใช้ในการประมวลผลภาพในขั้นถัดไป

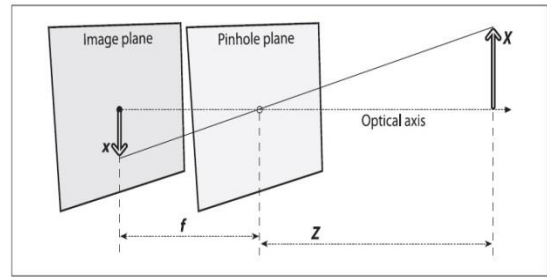


รูปที่ 13. แสดงให้เห็นภาพที่บิดเบี้ยวจากความโค้งของเลนส์ [8]

โดยฟังก์ชันการวางของระนาบบนฉากรับภาพสามารถแสดงได้ดังรูปที่ 14

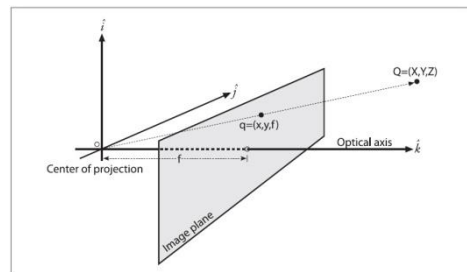
จากรูปที่ 15 ภาพที่ได้จากกล้องรูเข็มจะใช้เป็นโมเดลเริ่มต้น จากนั้นจึงแปลงไปอยู่ในรูปที่ให้จุด Center of Projection อยู่ด้านหลัง Image Plane เป็นการย้ายจุด Pin Hole ไปไว้ด้านหลัง หรือ

การสลับที่กันระหว่าง Pinhole กับ Image Plane เพื่อให้เข้าใจง่ายขึ้น และสามารถคำนวณหาระยะต่างๆ ได้แสดงได้ดังรูปที่ 15



รูปที่ 14. แสดงภาพที่ได้จากกล้องรูเข็ม (Pin Hole) ที่ปรากฏบนฉากรับภาพ (Image Plane)

f คือ ระยะโฟกัส Z คือ ระยะจากวัตถุถึงรูกล้อง [8]



รูปที่ 15. แสดงภาพที่ได้จากการสลับที่กันของภาพจากกล้องรูเข็ม กับฉากรับภาพจะได้เป็น ต้นแบบภาพ (Camera Model) ที่จะใช้ในการสอบเทียบ [8]

โดย Q คือจุดบนวัตถุ และ q คือจุดบนภาพที่ Image Plane เมื่อมีการลากเส้นจาก Q ไปตัดแกน Optical Axis ความสัมพันธ์ระหว่างจุดตัดบนวัตถุ $Q(X,Y,Z)$ และจุดตัดบนภาพ $q(x,y,z)$ บน Image Plane ได้ดังสมการที่ 1 และ 2

$$x = f * (X/Z) \quad (1)$$

$$y = f * (Y/Z) \quad (2)$$

จากความสัมพันธ์ข้างต้น จะใช้เมตริก Projective Transform เพื่อนำมาปรับแก้ความบิดเบี้ยวจากเลนส์โดยใช้พารามิเตอร์ Camera Intrinsic Matrix และ Distortion Intrinsic Matrix โดย Camera intrinsic matrix จะอยู่ในรูปเมตริก 3×3 ซึ่งแทนด้วย M มีส่วนประกอบด้วย 2 ส่วน คือ Focal Length (f_x, f_y) โดยที่ f_x, f_y คือ focal length หน่วยพิกเซล (pixel) และ Principle Point (C_x, C_y) ตามสมการที่ 3 ในความเป็นจริงจุด Principle Point ไม่ได้อยู่บน

Optical Axis ดังนั้นจึงต้องสร้างตัวแปรใหม่ ตัว คือ $2C_x$, C_y เพื่อสร้างจุดที่เคลื่อนที่ออกจาก Optical Axis ตามสมการที่ 4

$$x_{\text{screen}} = f_x \left(\frac{X}{Z} \right) + c_x, \quad y_{\text{screen}} = f_y \left(\frac{Y}{Z} \right) + c_y \quad (3)$$

$$q = MQ, \text{ where } q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4)$$

ส่วน Distortion Intrinsic Matrix จะใช้ในเรื่องของการแก้ความโค้งของเลนส์ ซึ่งประกอบด้วย 2 ส่วนคือ Radial Coefficients และ Tangential Coefficients โดย Radial Coefficients คือ k_1, k_2, k_3 ซึ่ง k_3 จะใช้ในกรณีของเลนส์ตาปลา (Fish Eye) และสามารถปรับค่าความโค้งให้ถูกต้องตามสมการที่ 5 และ 6

$$x_{\text{corrected}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (5)$$

$$y_{\text{corrected}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (6)$$

ส่วน Tangential Coefficients คือ P_1 และ P_2 จะใช้ในการปรับค่าความเอียงที่เกิดจากการติดตั้งเลนส์ของกล้อง

$$x_{\text{corrected}} = x + [2p_1 y + p_2(r^2 + 2x^2)] \quad (7)$$

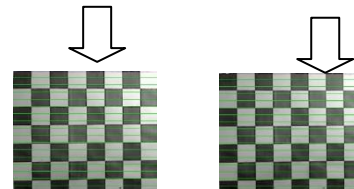
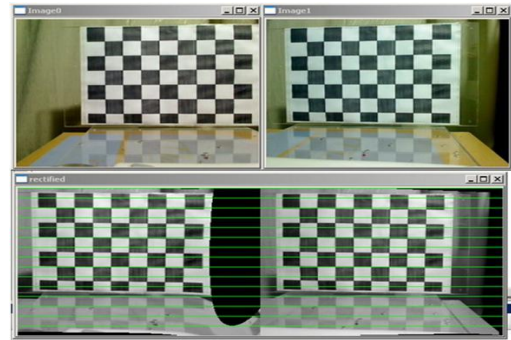
$$y_{\text{corrected}} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \quad (8)$$

พารามิเตอร์ทั้ง 5 ตัว ของ Distortion Intrinsic Matrix จะนำมาใช้ร่วมกันในการแก้ความโค้งของเลนส์ การทดลองเขียนโปรแกรมเพื่อหาค่าพารามิเตอร์ต่างๆ และนำไปใช้ในการปรับแก้ความโค้งของเลนส์โดยจะต้องเตรียมภาพตารางหมากรุกสำหรับการสอบเทียบกล้องแสดงได้ดังรูปที่ 16



รูปที่ 16. แสดงการหาจุดตัดบนตารางหมากรุกเพื่อใช้ในการสอบเทียบกล้อง

คำสั่งที่ใช้ในการสอบเทียบกล้องของ OpenCV คือ `cvCalibrateCamera2()` และ `cvInitUndistortMap()` ก็นำไปใช้เขียนในฟังก์ชัน `cvStereoCalibrate()` จะได้พารามิเตอร์ที่อยู่ในรูป .xml เมื่อได้พารามิเตอร์มาแล้วจะนำไปใช้ในการแก้ความโค้งของเลนส์ ดังแสดงในรูปที่ 17



รูปที่ 17. แสดงภาพที่ปรับแก้ความโค้งของเลนส์จะเห็นว่าภาพข้างบนก่อนปรับแก้ความโค้งของเลนส์บริเวณริมขอบภาพจะมีความโค้งอยู่ หลังปรับแก้แล้วจะทำให้ภาพมีส่วนโค้งลดลง

3.2.2. การรับข้อมูลภาพเข้ามาเพื่อทำการประมวลผลภาพเบื้องต้น ภาพที่ได้มาจากกล้อง 2 ตัว เมื่อปรับแก้ความเอียงเรียบร้อยแล้ว ข้อมูลภาพจะแยกภาพด้านซ้ายกับภาพด้านขวา ภาพแต่ละด้านจะถูกนำไปปรับเป็นภาพเทา-ดำ ก่อน แล้วจึงปรับแต่งเรื่องสภาพแสงที่เรียกว่าการกรองขั้นต่ำ(Threshold) เพื่อให้มีความเหมาะสมก่อน

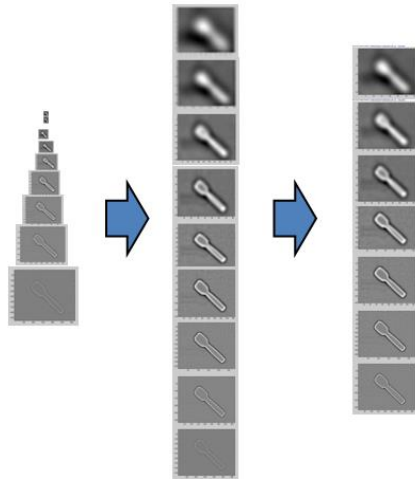
3.2.3. หาพีเจอร์ในภาพและทำการเปรียบเทียบกับส่วนที่เหมือนกันในข้อมูลภาพต้นแบบ

กระบวนการนี้เป็นการจดจำวัตถุในภาพ (Object Recognition) โดยใช้อัลกอริทึม SIFT Feature [5] [6] ในการหาวัตถุ ซึ่งมีคุณสมบัติที่ดีต่อการทนการเปลี่ยนแปลงของสภาพแสง ทนต่อการย่อ-ขยาย ทนต่อการหมุนของภาพได้ดีกว่า ซึ่งหลักการทำงานมี 4 ขั้นตอนหลักคือ

1. Scale-Space Extrema Detection
2. Keypoint Localization
3. Orientation Assignment
4. Generation of Keypoint Descriptor

ขั้นตอนที่ 1 : Scale-Space Extrema Detection

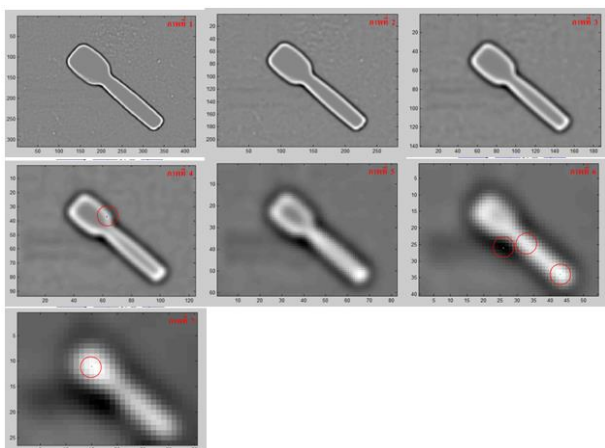
เป็นการนำวิธีของ Burt&Adelson ที่เรียกว่า DoG:Difference Gaussian of Pyramid มาพัฒนาเพิ่มเติม มีหลักการคือการนำภาพมาทำให้เบลอด้วย โฟกัสฟิลเตอร์ชื่อว่า Gaussian-Blurred ของภาพแต่ละขนาดและหาความแตกต่างของภาพโดยตัวอย่างแสดงดังรูปที่ 18



รูปที่ 18. แสดงการลดขนาดภาพของซ็อนลงไปเรื่อยๆ เป็นการทำให้เรียกว่าปิรามิด ตามอัตราส่วนที่กำหนดไว้

ขั้นตอนที่ 2 : Keypoint Localization

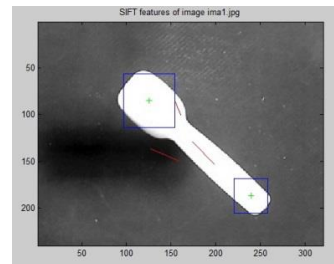
ขั้นตอนนี้จะเป็นการนำจุดที่สนใจที่ได้จากรูปที่ 18 มาไว้ในรูปเดียวกัน เพื่อรวมเป็นลักษณะของฟีเจอร์ในภาพ ผลของการหา Keypoint ดังแสดงในรูปที่ 19



รูปที่ 19. แสดงผลการทำ Keypoint Localization

ขั้นตอนที่ 3 : Orientation Assignment

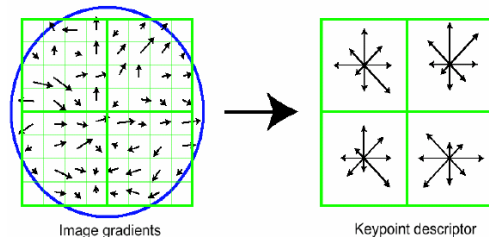
ขั้นตอนนี้เป็นการหาทิศทางผลรวมของจุดที่สนใจ จุดที่สนใจของซ็อนจะเห็นชัดที่บริเวณที่ดกของซ็อน และปลายสุดของหางซ็อน จะได้ตำแหน่ง (x,y) ของทั้ง 2 จุดออกมา ดังแสดงในรูปที่ 20



รูปที่ 20. แสดงการทิศทางของฟีเจอร์ที่ได้

ขั้นตอนที่ 4 : Generation of Keypoint Descriptor

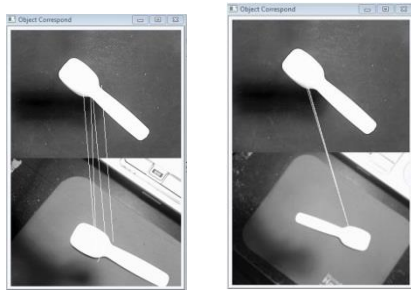
เมื่อได้จุดที่สนใจจากขั้นตอนที่ 3 แล้ว ขั้นตอนนี้จะสร้างตัวค้นหาเป็นกรอบเมตริก 4x4 หรือ 8x8 หรือมากกว่า เพื่อใช้เป็นฟีเจอร์มาตรฐาน ในการนำไปค้นหากรอบเมตริกที่มีความเหมือนกันในภาพที่ต้องการ ยิ่งกรอบเมตริกมากก็ยิ่งละเอียดมาก แต่จะใช้เวลาในการคำนวณหามากขึ้นด้วยตัวอย่างของการทำงานดังแสดงในรูปที่ 21



รูปที่ 21. แสดงวิธีการสร้างตัวค้นหา จากภาพซ้าย ขนาด 8x8 แบ่งย่อยภายในเป็น 4x4 โดยนำเวกเตอร์ที่มีทิศทางเดียวกันรวมกันแล้วให้มีขนาดพิกเซลเหลือเป็น 2x2 [6]

3.2.4 Keypoint Matching

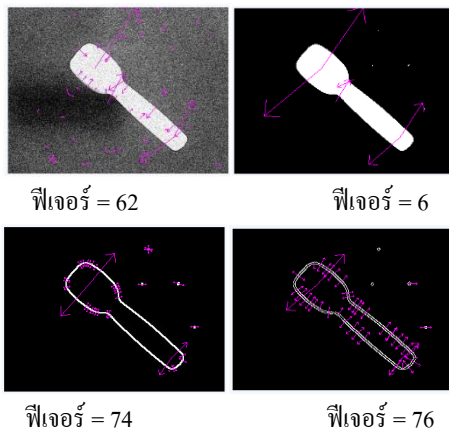
การทำการจับคู่จุดที่สนใจ (Keypoint Matching) เบื้องต้นจะใช้วิธีการของ Nearest Neighbor มาคำนวณหาระยะทางยูคลิเดียน (Euclidean Distance) ของแต่ละจุดที่สนใจ โดยจุดที่สนใจต้องมีฟีเจอร์อย่างน้อย 3 ฟีเจอร์ถึงจะใช้ได้ ตัวอย่างการทำงานดังแสดงในรูปที่ 22



รูปที่ 22. แสดงการการจับคู่ของพีเจอร์จากภาพต้นแบบไปยังวัตถุที่
ต้องการ

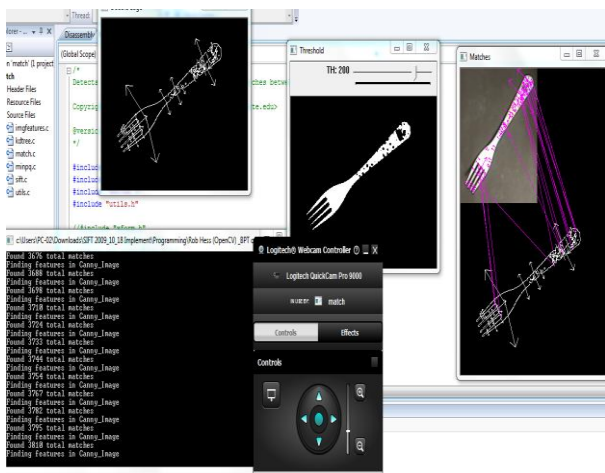
3.2.5. การเพิ่มจำนวนพีเจอร์ให้มากขึ้น

วิธีการเพิ่มจำนวนพีเจอร์ให้มากขึ้นสามารถทำได้โดยการทำ
ขอบภาพ (Edge) ซ้อน 2 ชั้นจะช่วยทำให้ พีเจอร์เพิ่มขึ้น อย่างน้อย 3-
5 เท่า ดังแสดงในรูปที่ 23



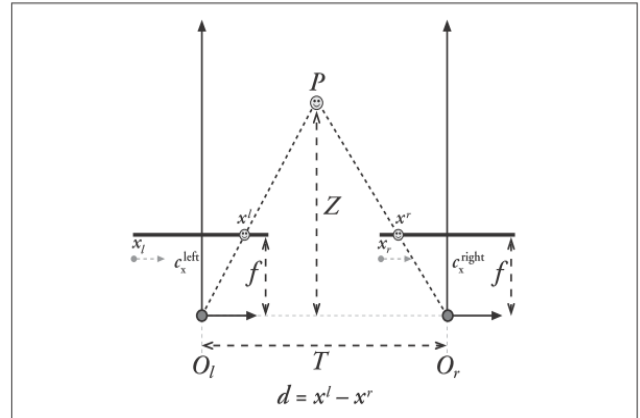
รูปที่ 23. แสดงพีเจอร์ที่เพิ่มขึ้นจากการทำขอบภาพ 2 ชั้น

ส่วนรูปที่ 24 แสดงภาพการหาพีเจอร์ของวัตถุที่มีลักษณะคล้ายกัน
เช่นส้อมจะสามารถหาได้หลังจากเพิ่มจำนวนพีเจอร์แล้ว



รูปที่ 24. แสดงภาพการหาพีเจอร์ของวัตถุที่มีลักษณะคล้ายกัน

3.2.6. การทำสเตอริโออิมเมจ เพื่อหาระยะความลึกของวัตถุในภาพ
การหาความลึกของวัตถุสามารถหาได้โดยใช้หลักการสามเหลี่ยม
คล้ายดังแสดงในรูปที่ 25 [7]

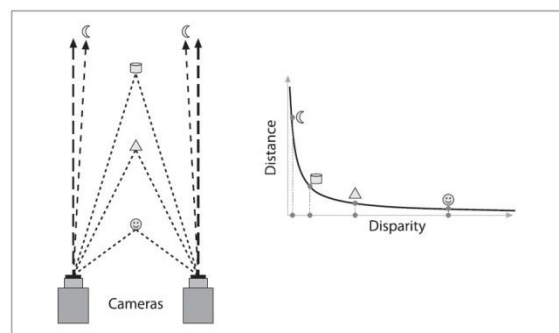


รูปที่ 25. แสดงหลักการสามเหลี่ยมคล้ายที่ได้จากกล้อง 2 ตัว [7]

T คือระยะระหว่างกล้อง จากภาพแสดงรูป 2 มิติ มี Cl กับ Cr จะเกิด
กรณีเลนส์มีความโค้ง (Lens Distortion) เราสามารถคำนวณจุด P ได้
จากภาพซ้าย ที่ จุด Pl และ Pr ซึ่งมี x^l และ x^r อยู่ โดยใช้สามเหลี่ยม
คล้ายหาความลึก z ตามสมการ 9

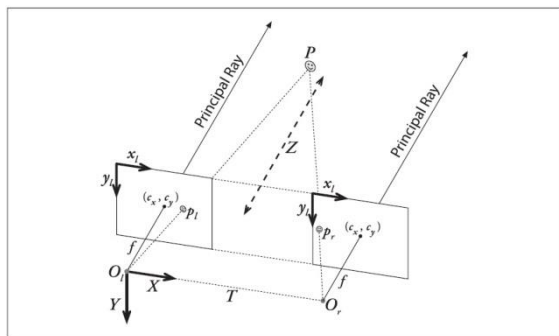
$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^l - x^r} \quad (9)$$

สำหรับระยะวัตถุที่ไกลขึ้นจะทำให้ภาพทางซ้ายและขวามีค่า
เหมือนกันน้อยลงดังแสดงในรูปที่ 26



รูปที่ 26. แสดงให้เห็นว่าถ้าระยะวัตถุไกลขึ้นค่าความเหมือนกัน
(Disparity) ของภาพทั้ง ซ้ายและขวาจะยิ่งน้อยลง [7]

ภาพในรูปแบบ 3 มิติ สำหรับใช้ในการคำนวณหาระยะความลึกดัง
แสดงในรูปที่ 27



รูปที่ 27. แสดงภาพแบบ 3 มิติ เพื่อใช้ในการหาระยะความลึก [7]

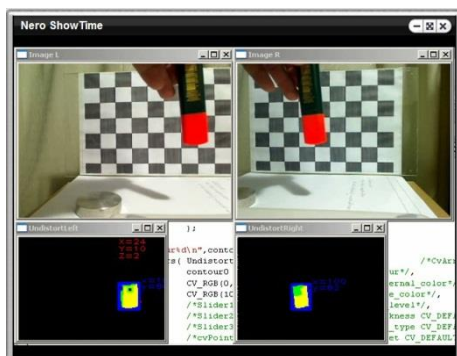
สูตรสมการ 10 11 และ 12 ที่ใช้ในการคำนวณหาระยะใน 3 มิติ

$$Z_actual = (b * focal_length) / (x_l - x_r) \quad (10)$$

$$X_actual = x * Z_actual / focal_length \quad (11)$$

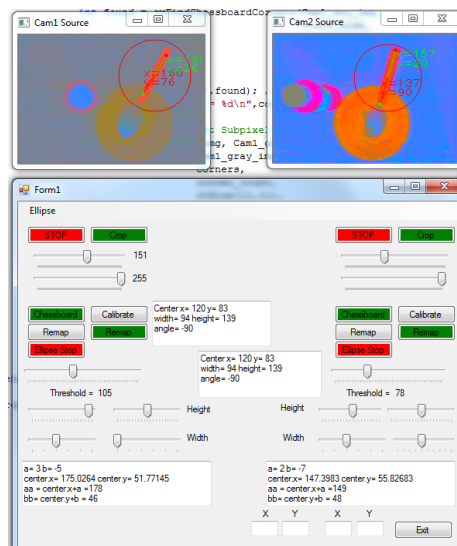
$$Y_actual = y_l * Z_actual / focal_length \quad (12)$$

จากรูปที่ 28 สำหรับทดสอบหาตำแหน่งของวัตถุในภาพ 3 มิติ จะเห็นว่าภาพด้านล่างจะมีบางส่วนใช้งานไม่ได้ จะต้องมีการตัดออกไปหลังสอบเทียบกล้อง จากนั้นก็สามารถหาระยะความลึกของวัตถุได้



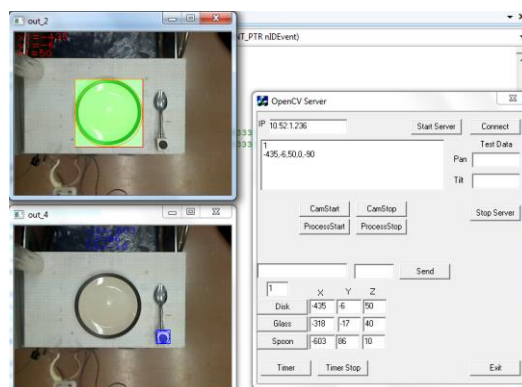
รูปที่ 28. แสดงการทดลองหาระยะความลึกของวัตถุในภาพ 3 มิติ

จากการทดลอง เราสามารถจำแนกวัตถุที่เป็นถ้วย และแก้ว ออกไปได้ก่อนและมีความแม่นยำสูง แล้วจึงใช้ SIFT Matching มาหาตัวซ้อนเพื่อหาคอordinateรอบวัตถุที่เป็นซ้อน แล้วจึงหาลักษณะเด่นของซ้อนเพิ่มเติมคือสี ความยาว และความกว้างของค้ำ เพื่อหาตำแหน่งของค้ำซ้อนดังแสดงในรูปที่ 29



รูปที่ 29. แสดงฟังก์ชันค้ำซ้อนของภาพซ้ายและขวา

การทดสอบการทำงานของโปรแกรมระบบวิชั่นที่เป็นเซิร์ฟเวอร์ สำหรับใช้รอการติดต่อมาจากโปรแกรมของแขนหุ่นยนต์เพื่อร้องขอค่าตำแหน่งของวัตถุที่ต้องการดังแสดงในรูปที่ 30



รูปที่ 30. แสดงภาพการทดสอบการทำงานของโปรแกรมในการหา

วัตถุที่จดจำไว้คือซ้อนจาน และ แก้ว ในรูปตำแหน่งทางซ้อนคือ (603, 86, 10) วัตถุตำแหน่งจริงได้ (594,90,22) ผิดพลาดไป $x=-9$ $y=4$ $z=12$ มิลลิเมตร

4. ผลการวิจัย

การทดสอบการทำงานของโปรแกรมระบบวิชั่นส่งค่าตำแหน่งของแก้ว ไปให้แขนหุ่นยนต์เข้ามาหยิบป้อนผู้ทดลองดังแสดงในรูปที่ 31



รูปที่ 31. แสดงการหยิบจับวัตถุของแขนหุ่นยนต์ร่วมกับระบบประมวลผลภาพ

จากการบันทึกผลการทำงานของระบบพบว่าสามารถตรวจหาตำแหน่งชิ้น ตำแหน่งงาน และตำแหน่งแก้ว ได้เฉลี่ยภายในเวลาไม่เกิน 1.7 วินาที มีความถูกต้องของตำแหน่งพิกัดของวัตถุ วัดได้เฉลี่ยไม่เกิน ± 14 มิลลิเมตร ส่วนการส่งข้อมูลไปให้แขนหุ่นยนต์จะส่งทางซ็อกเก็ต โดยผู้ใช้กดปุ่มขวาด้วยคานข้างของหัวไหล่เพื่อเลือกจะให้แขนหุ่นยนต์หยิบอะไร เมื่อโปรแกรมแขนหุ่นยนต์รับคำสั่งก็จะส่งรหัสไปยังโปรแกรมประมวลผลภาพเพื่อขอให้ส่งข้อมูลตำแหน่งของวัตถุที่จะหยิบเข้ามา ส่วนการกดปุ่มซ้ายจะเป็นการสั่งให้แขนหุ่นยนต์หยุด

5. สรุปและแนวทางการพัฒนา

งานวิจัยนี้เป็นการนำเทคโนโลยีทางด้านคอมพิวเตอร์วิชั่นมาประยุกต์ใช้ในการจดจำวัตถุในภาพ เพื่อช่วยหาตำแหน่ง $x y z$ และมุมที่กระทำกับแนวระนาบ ของวัตถุในระบบภาพ 3 มิติ เพื่อส่งข้อมูลให้แขนกลเข้ามาหยิบจับแก้วน้ำและค้ำขึ้นตามทิศทางดังกล่าว จากการทดลองพบว่าผล การทำงานของระบบประมวลผลภาพความเร็วอยู่ในเกณฑ์ดี และการหาวัตถุซ้ำที่สุดไม่เกิน 2 วินาที เนื่องจากบางครั้งสภาพแสงไม่คงที่ ทำให้

การประมวลผลภาพยากขึ้น แต่ก็ยังหาวัตถุได้ โดยสามารถหาตำแหน่งของค้ำขึ้นได้และส่งข้อมูลให้แขนกลเข้ามาหยิบจับค้ำขึ้นได้ถูกต้อง แต่ยังต้องทำงานในสภาพแสงคงที่เช่นในห้อง ที่มีแสงจากหลอดฟลูออเรสเซนต์ เป็นตัวหลัก การส่งข้อมูลไปให้แขนหุ่นยนต์โดยใช้สถาปัตยกรรมแม่ข่าย-ลูกข่าย สามารถรับส่งข้อมูลกันได้เป็นอย่างดี

งานวิจัยนี้สามารถพัฒนาต่อให้ระบบวิชั่นสามารถตรวจจับปริมาณของอาหารบนจานได้ว่าเมื่ออาหารใกล้หมดหรือหมดแล้ว ก็จะส่งข้อมูลไปสั่งให้หุ่นยนต์หยุดการป้อนอาหาร

เอกสารอ้างอิง

- [1] พวงแก้ว กิจธรรม, การศึกษาสำหรับคนที่มีความบกพร่องทางร่างกายหรือสุขภาพ, Available from: <http://www.tddf.or.th/tddf/library> [20 ธันวาคม 2548].
- [2] Shahram Jafari and Ray Jarvis, "Eye-To-Hand Coordination: A Coarse-Calibrated Approach," presented at The 2004 Australasian Conference on Robotics and Automation (ACRA), Canberra, Australia, Dec. 2004.
- [3] Andrew Dankers, Nick Barnes, and Alex Zelinsky, "Primate Structures in Synthetic Dynamic Active Visual Saliency," 6th International Conference on Epigenetic Robotics (Epirob 2006): Modeling Cognitive Development in Robotic Systems. Paris, France, 2006.
- [4] Ioannis Iossifidis, Carsten Bruckhoff, Christoph Theis, Claudia Grote, Christian Faubel and Gregor Schöner, "CORA: An anthropomorphic robot assistant for human environment," In Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop, Berlin, Germany, Sep. 2002, pp. 392-398.
- [5] David G. Lowe, "Object Recognition from Local Scale-invariant Feature", International Conference on Computer Vision, Corfu, Greece, Sep.1999, pp. 1-10.
- [6] David G. Lowe, "Distinctive Image Features from Scale-invariant Key-point", International Journal of Computer Vision, vol. 60, pp. 91-110, January 2004.
- [7] Gary Bradski and Adrian Kaebler, Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Inc., 2008.