การบูรณาการ MATLAB กับ ระบบพีแอลซี S7-300 กรณีศึกษาการตรวจจับขวด

Integration MATLAB and S7-300 PLC System; Case study in Bottle Detecting

นายสะกล คำแผ่น[1], ไสว พงศ์สวัสดิ์[2]

คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขที่ 1 หมู่1 แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพฯ 10520

[1] Sakol2499@hotmail.com

[2] klsawai@kmitl.ac.th

บทคัดย่อ

ในบทความนี้ นำเสนอการประยุกต์ใช้กล้องถ่ายภาพขวดที่อยู่บนสายพานลำเลียง มาทำการวิเคราะห์หาจำนวนและความหนาแน่นของจำนวนขวดต่อพื้นที่สายพานลำเลียงล่วงหน้าแทนการใช้ตัวเซนเซอร์หรือลิมิตสวิตช์ที่จะรับรู้ข้อมูลเมื่อขวดต้องมาถึงตำแหน่งตรวจจับ โดยข้อมูลที่ได้จากการประมวลผลภาพจะถูกนำไปใช้งานร่วมกับโปรแกรมควบคุมการขับเคลื่อนสายพานลำเลียง เพื่อทำการปรับความเร็วของสายพานให้สอดคล้องกับปริมาณขวด ซึ่งในบทความได้ใช้โปรแกรม MATLAB ในการวิเคราะห์ข้อมูลภาพ เพื่อให้ได้ข้อมูลจำนวนขวดในบริเวณพื้นที่ ที่กำหนดและนำส่งข้อมูลผ่าน OPC (OLE for Process Control) ให้กับโปรแกรมควบคุม ในระบบพีแอลซี S7-300 เพื่อทำการควบคุมระบบขับสายพานลำเลียง ผลของการทดลองการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมดังกล่าวผ่าน OPC แสดงให้เห็นว่าระบบสามารถทำการแลกเปลี่ยนข้อมูลปริมาณขวดได้และสามารถทำงานร่วมกันได้จริง

คำสำคัญ: สายพานลำเลียง OPC พีแอลซี เซนเซอร์ ความหนาแน่นของจำนวนขวด

Abstract

This paper describes an implementation of using camera to capture image of bottle on conveyor belt. The image analysis will be done in order to determine a density of number of bottles (DNB) instead of using sensor or limit switch which activate when bottles arrived to detecting point of the sensor. The data from the image processing to be used in conjunction with the drive control system in order to adjust the belt speed in accordance with the amount of bottles in assigned zone on the conveyor. The image analysis will be done by MATLAB to obtain information on the number of bottles in the specific area. The data from MATLAB will be sent via OPC to S7-300 PLC for the motor speed control system. The experiment result shows that the system can exchange information and can actually work together.

Keywords: Conveyor belt, OPC, PLC, Sensor, DNB

1. คำนำ

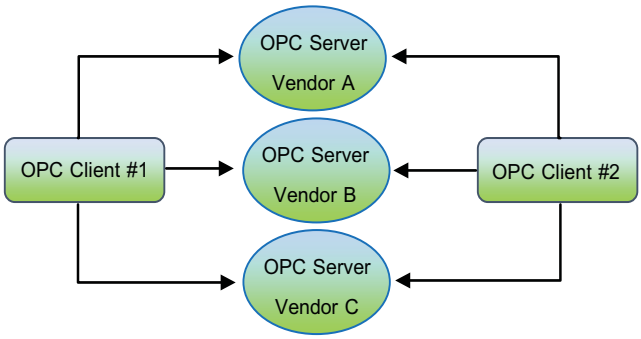
ในสายการผลิตที่ใช้ขวดเป็นบรรจุภัณฑ์และใช้โฟโต้อิเล็กทริคเซนเซอร์หรือลิมิตสวิตช์ เป็นตัวตรวจจับขวดที่เข้ามาในสายการผลิต หรือ สายพานลำเลียง ซึ่งบางครั้งในสายการผลิตที่ใช้ขวดเป็นบรรจุภัณฑ์ จะต้องนำผลจากตัวตรวจจับดังกล่าว เป็นตัวแสดงผลการนับจำนวนขวดและจับความหนาแน่นของจำนวนขวดที่อยู่บนสายพานลำเลียง เพื่อปรับความเร็วของมอเตอร์ขับเคลื่อนสายพาน ผลของการตรวจจับที่ต้องให้ขวดสัมผัสกับลิมิตสวิตช์ อาจจะทำให้ขวดเกิดการเบียดกับตัวสวิตช์แล้วทำให้ขวดผิดรูปไป อีกทั้งเกิดความล่าช้าในการที่จะตรวจจำนวนขวดในสายพานลำเลียงเนื่องจากต้องรอให้ขวดเคลื่อนที่มาถึงตำแหน่งตรวจจับก่อน ตัวตรวจจับหรือลิมิตสวิตช์จึงจะสามารถตรวจจับได้ เป็นผลให้บางครั้งขวดเกิดการเบียดกันจนผิดรูปหรือล้ม หรือถ้าเป็นขวดแก้วก็อาจทำให้ขวดแตกได้ การประยุกต์ใช้การสื่อสารระหว่างกล้อง กับกล่องประมวลผลรูปภาพและพีแอลซี สามารถช่วยควบคุมจำนวนและความหนาแน่นของจำนวนขวดปรกติบนสายพานลำเลียงได้ 90% และ ช่วยแก้ปัญหาเรื่องขวดแตกได้ [1] โดยในบทความนี้นำเสนอวิธีการประยุกต์ใช้ซอฟต์แวร์ที่มีอยู่ในปัจจุบันมาประยุกต์ทำงานร่วมกัน ซึ่งสะดวกและไม่จำเป็นต้องสร้างฮาร์ดแวร์ขึ้นมาใหม่ โดยการใช้กล้องที่มีความละเอียด 2592 x 1936 พิกเซล มาทำการถ่ายภาพขวดที่อยู่บนสายพานลำเลียง มาวิเคราะห์หาจำนวนและความหนาแน่นของจำนวนขวดต่อพื้นที่สายพานลำเลียงแล้วนำผลที่ได้จากการวิเคราะห์ไปใช้งานร่วมกับโปรแกรมควบคุมการขับเคลื่อนของสายพานลำเลียง เพื่อทำการปรับความเร็วของสายพานให้สอดคล้องกับปริมาณขวด โดยในบทความได้ใช้โปรแกรม MATLAB ในการวิเคราะห์ข้อมูลภาพเพื่อให้ได้ข้อมูลจำนวนขวดในบริเวณพื้นที่ ที่กำหนดและนำส่งข้อมูลผ่าน OPC ให้กับโปรแกรมควบคุม ระบบพีแอลซี S7-300 เพื่อทำการควบคุมระบบขับสายพานลำเลียงให้สอดคล้องกับปริมาณหรือความหนาแน่นของจำนวนขวดที่กำหนด ผลของการทำลองการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมดังกล่าวผ่าน OPC แสดงให้เห็นว่าระบบสามารถทำงานได้จริงและสามารถใช้เป็นแนวทางเพื่อนำข้อมูลที่ได้ ไปประยุกต์ใช้เพื่อแทนการ ใช้งานหรือลดการใช้ตัวตรวจจับ อีกทั้งข้อมูลที่ผ่าน OPC สามารถนำไปเชื่อมโยงกับโปรแกรม Excel เพื่อจัดทำข้อมูลหรือรายงานต่างๆ ต่อไปได้

2. รูปแบบการสื่อสาร

บทความนี้จะใช้ OPC เป็นจุดศูนย์กลางในการเชื่อมต่อข้อมูลระหว่างโปรแกรมและอุปกรณ์ต่างๆ

2.1 ภาพรวม OPC

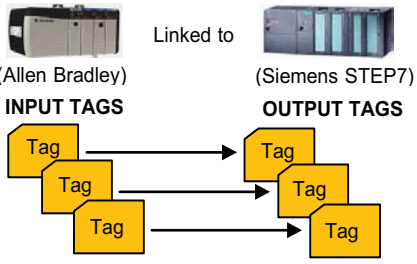
OPC Foundation ตั้งขึ้นในปี ค.ศ 1994 เป็นสมาคมอุตสาหกรรมที่สร้างและรักษา มาตรฐานสำหรับการเชื่อมต่อแบบเปิด ของอุปกรณ์ในระบบอุตสาหกรรมอัตโนมัติ ในปี ค.ศ 1996 ได้เริ่มพัฒนา OPC (OLE for Process Control) โดยการร่วมมือกับบริษัทเอกชนในกลุ่มอุตสาหกรรมอัตโนมัติ และต่อมาในปี ค.ศ. 2011 ได้ให้ความหมายตัวย่อใหม่ของ OPC เป็น รูปแบบการสื่อสารแบบเปิด (Open Platform Communications) การเชื่อมต่อระหว่าง OPC เซิร์ฟเวอร์และ OPC ไคลเอนต์ สามารถที่จะเชื่อมต่อได้มากกว่าหนึ่งการเชื่อมต่อ ที่มาจากต่างผู้ผลิตกัน [2] ดังรูปที่ 1



รูปที่ 1 การเชื่อมต่อระหว่าง OPC เซิร์ฟเวอร์ และ ไคลเอนต์

2.2 OPC ที่ใช้ในบทความนี้

บทความนี้จะใช้ KEPware OPC ซึ่งจะมีฟังก์ชันการสื่อสารข้อมูลระหว่าง ไคลเอนต์ เซิร์ฟเวอร์ และ อุปกรณ์ โดยใช้แท็กขั้นสูง (Advance Tags) ที่เป็นลิงค์แท็ก (Link Tag) ซึ่งมีทั้งการเชื่อมต่อแบบธรรมดา และการเชื่อมต่อแบบมีเงื่อนไข

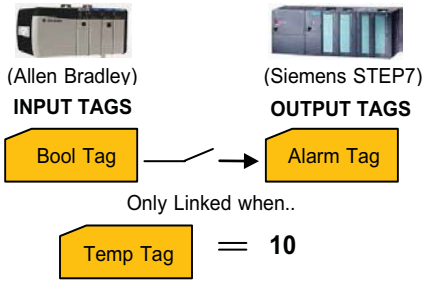


Setting: Update Rate 1000 ms

Conduct link on: Input value Change

รูปที่ 2 การส่งข้อมูลด้วย Tag ระหว่างอุปกรณ์

จากรูปที่ 2 กลุ่มข้อมูลจากอุปกรณ์ต้นทาง จะถูกส่งไปยังอุปกรณ์ปลายทาง ทุกๆ 1000 ms หากค่าอินพุทมีการเปลี่ยนแปลงจากค่าก่อนหน้า



Setting: Scan Rate of Monitored Tag 1000 ms

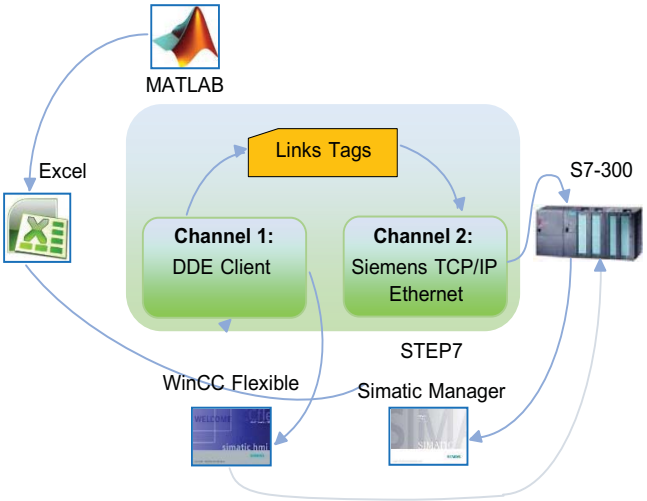
Conduct link on: Temp Tag = 10

รูปที่ 3 การส่งข้อมูลด้วย Tag แบบมีเงื่อนไข

จากรูปที่ 3 ข้อมูลจากอุปกรณ์ต้นทาง จะถูกส่งไปยังอุปกรณ์ปลายทาง ทุกๆ 1000 ms หากตรงตามเงื่อนไขที่กำหนด คือ Temp Tag=10

3. การออกแบบระบบและการสื่อสาร

บทความนี้ได้ออกแบบโครงสร้างการสื่อสารดังรูปที่ 4 ที่มีรูปแบบการส่งข้อมูลจาก MATLAB ไปยัง MS Excel ก่อนที่จะผ่าน OPC เพื่อลิงค์แท็ก ไปยังระบบพีแอลซี S7-300 ในขณะเดียวกันก็สามารถนำข้อมูลที่ได้ไปแสดงยังหน่วยแสดงผล WinCC Flexible ซึ่งมี OPC Client Driver เดียวกับ MS Excel จึงไม่จำเป็นต้องผ่านลิงค์แท็ก



รูปที่ 4 โครงสร้างการสื่อสารที่ออกแบบ

3.1 การส่งข้อมูลจาก MATLAB ไปยัง OPC

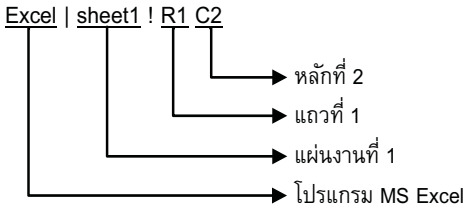
รูปแบบมาตรฐานการส่งข้อมูลจาก MATLAB ไปยัง OPC สามารถทำได้โดยการใช้เครื่องมือใน MATLAB OPC Toolbox [3,9] โดยบทความนี้ออกแบบให้ส่งออกข้อมูลจาก MATLAB ไปเก็บไว้ใน MS Excel ก่อนที่จะส่งต่อไปในส่วนอื่นๆ

3.2 การรับและส่งข้อมูลผ่าน OPC เซิร์ฟเวอร์

ช่องทางที่1 : ข้อมูลจาก Excel จะถูกเขียนลงไปในแท็กรับข้อมูลโดย DDE Client Server ซึ่งมีรูปแบบมาตรฐาน (Standard DDE Addressing) ดังนี้

<DDE service name>|<Topic name>!<Item name>

เช่น excel|sheet1!R1C2 อธิบายตามรูปที่ 5



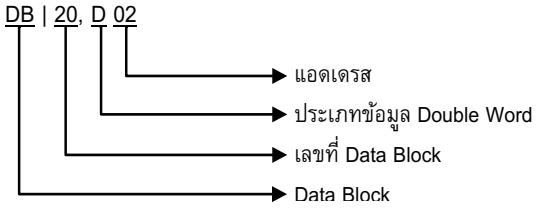
รูปที่ 5 รูปแบบการสื่อสารระหว่าง OPC กับ Excel

จากนั้นข้อมูลจะถูกส่งไปยังช่องทางที่2 ผ่านลิ้งแท็ก ในขณะเดียวกันก็สามารถส่งข้อมูลไปยังหน้าจอแสดงผล WinCC Flexible

ช่องทางที่2 : รับข้อมูลจากช่องทางที่1 ผ่านลิ้งแท็ก ก่อนที่จะส่งไปยังพีแอลซีโดยไดรเวอร์ Siemens TCP/IP Ethernet [4] ซึ่งมีรูปแบบมาตรฐาน (DB Memory Type – Standard S7300/400 Item Syntax) ดังนี้

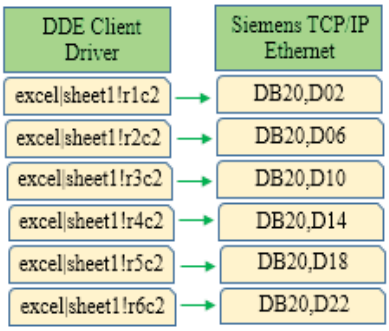
DB<num>,<S7 data type><address> อธิบายตามรูปที่ 6

เช่น DB20,D02



รูปที่ 6 รูปแบบการสื่อสารระหว่าง OPC กับ S7-300

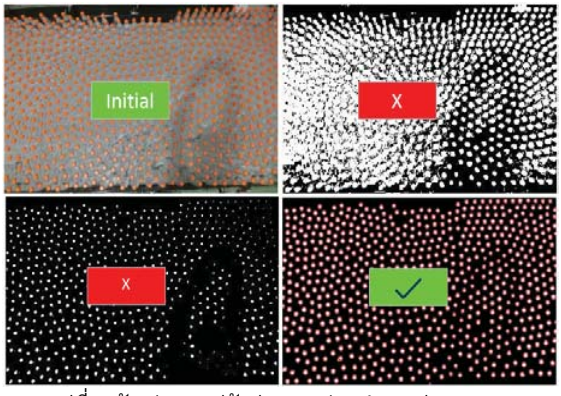
บทความนี้จะกำหนดข้อมูลขึ้นมาทั้งหมดหกชุดตามรูปที่ 7 เพื่อแทนปริมาณความหนาแน่นของจำนวนขวดบนสายพานลำเลียงทั้งหกโซนซึ่งจะกล่าวถึงรายละเอียดในหัวข้อการวิเคราะห์รูปภาพ



รูปที่ 7 การเชื่อมต่อระหว่าง Device Driver โดยใช้ ลิงค์แท็ก

4. กรณีศึกษาการตรวจจับขวด

บทความนี้จะนำเสนอการใช้ MATLAB ประมวลผลหาความหนาแน่นของจำนวนขวดบนสายพานลำเลียง ซึ่ง MATLAB มีเครื่องมือหลากหลายในการสื่อสารกับกล้องถ่ายรูปเพื่อตรวจจับภาพโดยไม่จำเป็นต้องสร้างฮาร์ดแวร์และโปรแกรมมาใหม่ และ มีเครื่องมือวิเคราะห์รูปภาพมากกว่า 2,000 เครื่องมือ ครอบคลุมทั้งขั้นตอนวิธีการอ้างอิงฟังก์ชันมาตรฐาน แอพพลิเคชั่นสำหรับการประมวลผลภาพ การวิเคราะห์ การแสดง และการพัฒนา อัลกอริทึม ซึ่งได้อธิบายไว้ใน [5,6] ความท้าทายในการประมวลผลภาพทั่วไปก็จะมีทั้ง ขนาดของภาพ ความสว่าง รูปร่างของวัตถุ ขนาดของวัตถุ และสีของวัตถุที่ต้องการตรวจจับ ดังแสดงใน รูปที่ 8 จะเห็นได้ว่าภาพความสว่างของพื้นหลังไม่สม่ำเสมอ จึงเป็นอุปสรรคสำหรับการประมวลผล ดังนั้นจึงจำเป็นต้องปรับปรุงคุณภาพโดยการขจัดสัญญาณภาพส่วนที่เกิน และเพิ่มความคมชัดของภาพในส่วนที่ต้องการประมวลผล ซึ่งในบทความนี้ก็คือฝาขวด จะอธิบายรายละเอียดในแต่ละขั้นตอนต่อไป



รูปที่ 8 ตัวอย่างการปรับปรุงภาพก่อนทำการประมวลผล

4.1 ขนาดของวัตถุ

จากการทดลองถ่ายภาพขวดในแนวตั้งฉากที่ระยะห่างระหว่างกล้องกับขวดต่างๆ กันแล้ววัดขนาดของฝาขวดโดยใช้ฟังก์ชัน ‘ imtool ’ จะได้ขนาดของฝาขวดตามตารางที่1 จะเห็นได้ว่าความละเอียดของภาพจะลดลง หรือ ขนาดของพิกเซลจะมีขนาดใหญ่ขึ้นเมื่อระยะห่างระหว่างกล้องกับขวดเพิ่มมากขึ้น [7]

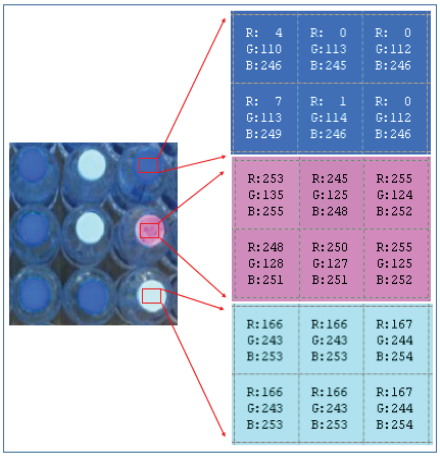
ตารางที่1 เปรียบเทียบขนาดฝาขวดกับระยะห่างระหว่างกล้อง

ระยะห่างระหว่างกล้องกับขวด (มิลลิเมตร)	ค่าเฉลี่ยขนาดฝาขวด (พิกเซล)
500	56
1,000	27
1,500	18
1,800	14
2,000	11
2,500	8
3,000	6

ข้อมูลขนาดฝาที่ได้จะเอาไปประมวลผลตรวจจับในขั้นตอนต่อไป

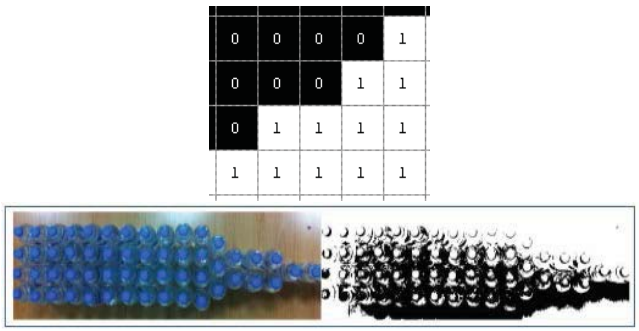
4.2 ความสว่างของวัตถุ

ภาพที่ได้จากอุปกรณ์ถ่ายภาพ บางครั้งจะได้ภาพที่มีความส่องสว่างไม่สม่ำเสมอ ขนาดรูรับแสงเลนส์ก็มีผลกับความความสว่างของภาพเช่นกัน [8] MATLAB จะมีฟังก์ชันในการวัดค่าความสว่างของสีแดง สีเขียว และ สีน้ำเงิน (RGB : Red Green and Blue) บทความนี้ได้ทดลองวัดค่า RGB สีของฝาขวดสามสีคือ สีน้ำเงิน สีชมพู และ สีขาว ซึ่งจะได้ค่าตามรูปที่ 9



รูปที่ 9 ค่าความสว่างของภาพ RGB

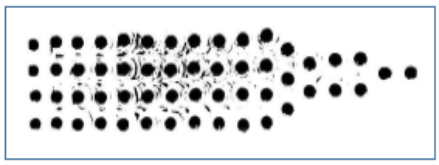
ข้อมูลที่ได้จะถูกนำมาแปลงเป็น ข้อมูลภาพ ขาว-ดำ (Binary image) ดังรูปที่ 10 ก่อนทำการประมวลผลนับจำนวนขวด ซึ่งภาพ ขาว-ดำ นี้จะมี 1 บิต ต่อ พิกเซล โดยค่าสีจะมีแค่สองค่าคือ 0 หรือสีดำ และ 1 หรือสีขาว โดยใช้รูปแบบฟังก์ชัน “ im2bw(I, level) ” เพื่อทำให้เป็นภาพไบนารี่ เมื่อ “ I ” คือภาพที่เราจะนำมาทำการประมวลผล และ “ level ” ค่าที่ใช้ปรับระดับความสว่าง



รูปที่ 10 ค่าข้อมูลแบบไบนารี่ที่ได้จากการประมวลผลภาพ

4.3 การปรับละเอียดระดับสี ขาว-ดำ

จากขั้นตอนก่อนหน้านี้อาจจะยังได้ภาพที่ไม่สมบูรณ์ ซึ่งเกิดจากเงาที่มีอยู่ในภาพเริ่มต้น จะเห็นได้ว่าพื้นหลังของภาพมีความสว่างไม่สม่ำเสมอ จึงทำให้ภาพหลังจากที่ทำให้เป็นภาพ ขาว-ดำ มีภาพรบกวนวัตถุที่ต้องการนับ เราจึงทำการปรับละเอียดอีกรอบโดยระบุค่าระดับ (level) ให้เหมาะสมกับความสว่างของภาพ เพื่อให้ได้ภาพที่มีคุณภาพมากขึ้นดังรูปที่ 11

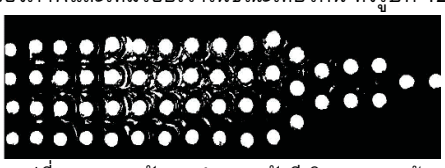


รูปที่ 11 ภาพหลังจากปรับค่าระดับความสว่าง

การแปลงภาพระดับสีเทาเป็นภาพไบนารี ภาพจะแทนที่พิกเซลทั้งหมดในภาพที่นำเข้าด้วยลอจิก 1 (สีขาว) และ 0 (สีดำ) Level จะช่วยปรับละเอียดอยู่ในช่วง (0,1) ช่วงนี้จะสัมพันธ์กับระดับสัญญาณของภาพ ดังนั้นค่าระดับ 0.5 เป็นค่ากึ่งกลางระหว่างสีดำและสีขาวโดยไม่คำนึงถึงระดับมาตรฐานของภาพไบนารี

4.4 การเติมเต็มรูปภาพวัตถุ

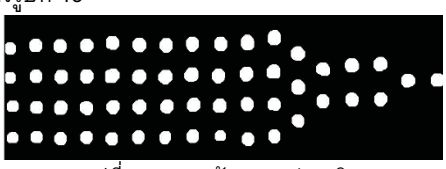
ภาพวัตถุที่ได้จากขั้นตอนการปรับละเอียดก่อนหน้านี้อาจมีรอยเว้าหรือฉีกและไม่สมบูรณ์ จึงจำเป็นต้องดำเนินการในขั้นตอนต่อไป โดยจะแทนที่สีขาวด้วยสีดำทั้งหมด โดยใช้ฟังก์ชัน “ imcomplement ” เพื่อคำนวณคุณสมบัติของภาพ ซึ่งภาพสามารถเป็นไบนารี สีเทา หรือภาพ RGB ก็ได้ ถ้าเป็นภาพไบนารี่ 0 จะกลายเป็น 1 และ 1 จะกลายเป็น 0 สีดำและสีขาวจะสลับกัน แต่ถ้าเป็นภาพ RGB ค่าแต่ละพิกเซลจะถูกหักออก จากค่าพิกเซลสูงสุดที่สนับสนุนและความแตกต่างถูกนำมาใช้เป็นค่าพิกเซล บริเวณที่มืดกลายเป็นสว่างและพื้นที่สว่างจะกลายเป็นสีเข้ม จากนั้นทำการเพิ่มส่วนที่ขาดหายโดยใช้ฟังก์ชัน “ imfill ” หลังจากทำการสลับสีของภาพและเติมรอยเว้าในขณะเดียวกัน ดังรูปที่ 12



รูปที่ 12 ภาพหลังจากทำการสลับสีเติมและรอยเว้า

4.5 การตัดภาพรบกวน

จากขั้นตอนก่อนหน้านี้จะเห็นได้ว่ายังมีภาพรบกวนอยู่ ฟังก์ชัน “ strel ” และ “ imopen ” จะสร้างองค์ประกอบโครงสร้าง ประเภทและขนาดของวัตถุที่ระบุโดยรูปร่าง Strel ยังสามารถใช้กับพารามิเตอร์เพิ่มเติมได้ เช่น ต้องการเฉพาะวัตถุที่มีลักษณะเป็นแผ่นกลมที่มีรัศมีตั้งแต่ 10 พิกเซลขึ้นไป วัตถุที่มีรัศมีน้อยกว่า 10 พิกเซลจะถูกลบออกจากภาพ ดังรูปที่ 13

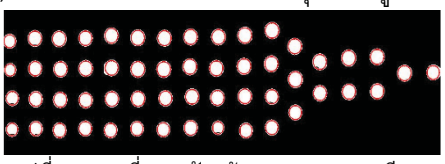


รูปที่ 13 ภาพหลังจากลบส่วนเกิน

4.6 การประมวลผลตรวจจับภาพ

มาถึงขั้นตอนนี้ เราได้ภาพที่สมบูรณ์และพร้อมที่จะหาฝาซึ่งเป็นวัตถุวงกลม โดยใช้ฟังก์ชัน “imfindcircles” ที่มีรัศมีอยู่ในช่วงที่เราต้องการตรวจสอบ ช่วงของรัศมีสำหรับวัตถุทรงกลมที่ต้องการกาตรวจสอบจะระบุเป็นเวกเตอร์ สององค์ประกอบ [Rmin Rmax] ของจำนวนเต็มใดๆ

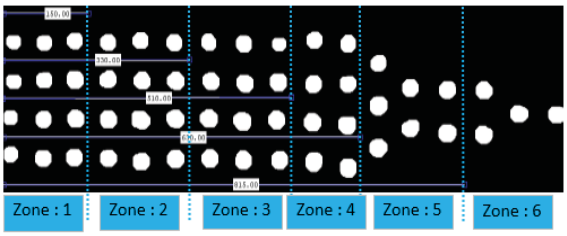
ในขณะเดียวกันเมื่อตรวจสอบพบแล้วก็ให้เขียนวงกลมลงไปในฝานั้นๆ (Viscircles) เพื่อให้มั่นใจว่าตรวจสอบฝาครบทุกฝา ดังรูปที่ 14



รูปที่ 14 ภาพที่ตรวจนับแล้วจะถูกวาดวงกลมสีแดง

4.7 คำนวนหาความหนาแน่นของจำนวนขวดในแต่ละโซน

ข้อมูลที่ได้ในขั้นตอนการตรวจจับภาพนี้คือตำแหน่งจุดศูนย์กลาง X,Y ของฝาแต่ละฝา ดังนั้นเราจึงสามารถประมวลผลหาความหนาแน่นของจำนวนขวดในแต่ละโซนได้โดยกำหนดแนวแกน X เป็นหลัก จากขั้นตอนก่อนหน้านี้จำนวนขวดทั้งหมดที่นับได้คือ 55 ขวด และจำนวนขวดในแต่ละโซนดังแสดงรูปที่ 15 ในหัวข้อผลการทดลอง

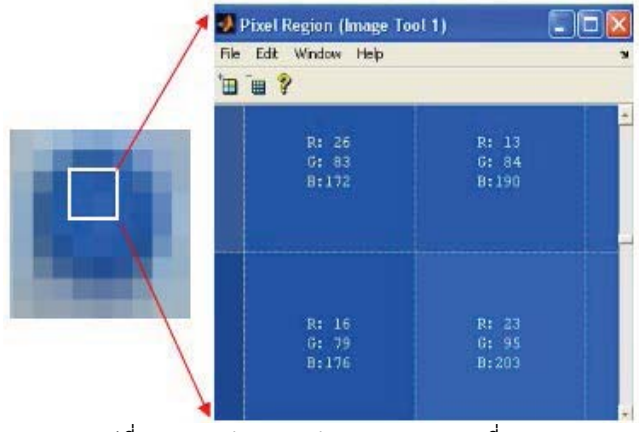


รูปที่ 15 กำหนดขอบเขตแต่ละโซนในแนวแกน X

5. ผลการทดลอง

จากรูปที่ 16 และ ตารางที่ 2 จะแสดงความคมชัดของภาพที่ระยะห่างระหว่างกล้องกับขวด ตั้งแต่ 500, 1,000, 1,500, 1,800, 2,000, 2,500, และ 3,000 มิลลิเมตร(มม.) ตามลำดับ ซึ่งจากการวัดขนาดของฝาและเปรียบเทียบค่าพิกเซลก่อนและหลังการปรับปรุงภาพ จะเห็นได้ว่าเปอร์เซ็นผิดพลาดจะสูงขึ้นที่ระยะห่างมากขึ้นเนื่องจากความละเอียดค่าพิกเซลของภาพลดลง

รูปที่ 16 แสดงความคมชัดของภาพที่ระยะห่าง 500-3,000 มม.



รูปที่ 17 ความค่าความสว่าง RGB ของภาพที่เบลอ

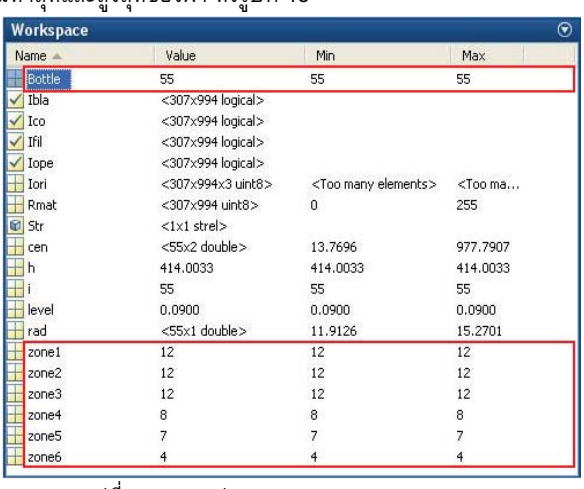
จากรูปที่ 17 ค่าความสว่าง RGB ของแต่ละพิกเซลจะแตกต่างกันอย่างเห็นได้ชัดที่ระยะห่างระหว่างกล้องกับวัตถุมากๆ

ตารางที่ 2 เปรียบเทียบค่าเปอร์เซ็นความผิดพลาดของขนาดฝา ก่อน และหลังการประมวลผลภาพ

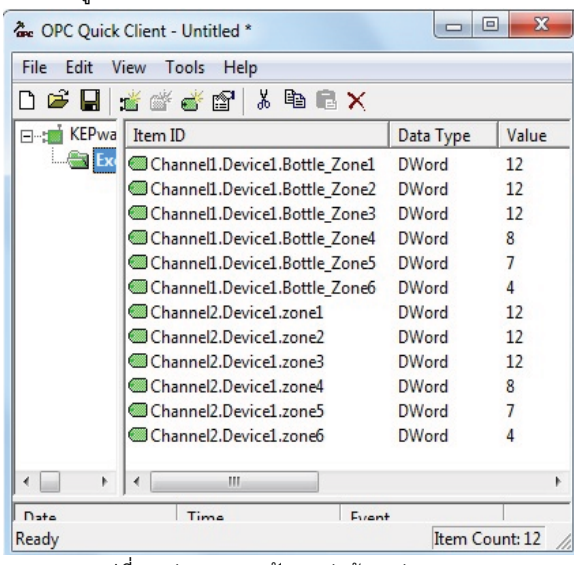
ระยะห่างระหว่างกล้องกับขวด (มิลลิเมตร)	เปรียบเทียบขนาดฝาก่อนและหลังการปรับปรุง (พิกเซล)		เปอร์เซ็นผิดพลาด
	ก่อน	หลัง	
500	56	58	3.57%
1,000	27	28.72	6.37%
1,500	18	18.54	3.00%
1,800	14	16.5	17.86%
2,000	11	13.38	21.64%
2,500	8	12.50	56.25%
3,000	6	-*	100%*

* ที่ระยะห่าง 3,000 มม. ภาพมีขนาดเพียงแค่ 6 พิกเซล ซึ่งไม่ละเอียดเพียงพอ (ภาพเบลอ) จึงไม่สามารถประมวลผลตรวจจับฝาขวดได้

ข้อมูลที่ได้จากการประมวลผล นอกจากจำนวนขวดแล้ว ก็จะมีข้อมูลค่าเฉลี่ยต่างๆ ค่าต่ำสุด และค่าสูงสุด เช่น จุดศูนย์กลาง X,Y และ ค่ารัศมีต่ำสุดและสูงสุดของฝา ดังรูปที่ 18

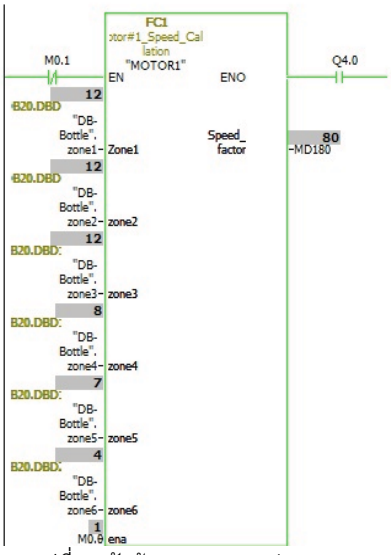


รูปที่ 18 ผลการประมวลผล MATLAB Workspace



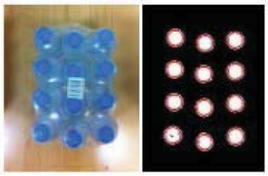
รูปที่ 19 ช่องทางการรับและส่งข้อมูลผ่าน OPC

จากรูปที่19 ช่องทางที่1 (Channel1.Device1) จะสื่อสารกับ Excel ที่รอบการสแกนส์ที่ 500 ms ส่วนช่องทางที่2 (Channel2.Device2) จะสื่อสารกับ พีแอลซี S7-300 ที่รอบการแสกนส์ที่ 1,000 ms โดยข้อมูลที่ได้จะถูกส่งผ่าน DB ไปประมวลผลฟังก์ชัน FC1 ที่เป็นโปรแกรมเพื่อควบคุมการขับเคลื่อนความเร็วของระบบสายพานลำเลียง ซึ่ง FC1 ที่พัฒนามาใช้งานแสดงได้ดังรูปที่ 20



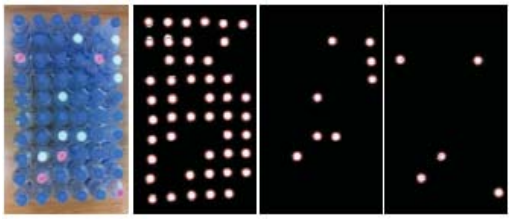
รูปที่ 20 รับข้อมูลจาก OPC ผ่าน DB20

นอกจากการนับจำนวนขวดบนสายพานลำเลียงขวดแล้ว ข้อมูลภาพที่ได้ยังสามารถประยุกต์ใช้เพื่อตรวจนับจำนวนขวดต่อหนึ่งแพ็คว่ามีครบหรือไม่ เพื่อแทนการใช้งานปัจจุบันที่ยังใช้วิธีการชั่งน้ำหนัก ดังรูปที่ 21



รูปที่ 21 ตรวจนับจำนวนขวดในแพ็ค

ในบางครั้งผู้ผลิตอาจมีผลิตภัณฑ์มากกว่าหนึ่งชนิดบนสายพานลำเลียงเดียวกัน ซึ่งอาจมีสีของฝาต่างกัน ในกรณีนี้เราสามารถประมวลผลนับฝาทีละสี ฟ้า ขาว และ ชมพู แล้วค่อยรวมจำนวนฝาทั้งสามสี ดังรูปที่ 22



รูปที่ 22 ตรวจนับฝาขวดที่มีสีแตกต่างกัน

6. สรุป

การบูรณาการ MATLAB กับ ระบบพีแอลซี S7-300 สามารถประมวลผลตรวจจับจำนวนขวดในบริเวณพื้นที่ ที่กำหนดโดยใช้ซอฟต์แวร์ที่มีอยู่ในปัจจุบันมาประยุกต์ทำงานร่วมกันซึ่งสะดวกและไม่จำเป็นต้องสร้างฮาร์ดแวร์ขึ้นมาใหม่ ซึ่ง MATLAB มีเครื่องมือที่หลากหลายในการสื่อสารกับกล้องถ่ายรูปเพื่อจับภาพและประมวลผลภาพ และนำส่งข้อมูลผ่าน OPC ให้กับโปรแกรมควบคุม ระบบพีแอลซี S7-300 เพื่อทำการควบคุมระบบขับสายพานลำเลียง ผลของการทดลองการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมดังกล่าวผ่าน OPC แสดงให้เห็นว่าระบบสามารถทำการแลกเปลี่ยนข้อมูลปริมาณขวดได้และสามารถทำงานร่วมกันได้จริง

เอกสารอ้างอิง

[1] Pieter P. Jonker, “Image Processing Hardware for Counting Massive Object Streams”, IV Conference D: Architectures for Vision and Pattern Recognition (IAPR), International Conference, 1992, page(s) 31-33.

[2] OPC Foundation, OLE for Process Control Data Access Custom Interface Standard Version 2.5

[3] OPC Toolbox™ User's Guide 2004–2015 by The MathWorks, Inc., http://cn.mathworks.com/help/pdf_doc/opc/opc.pdf

[4] Siemens TCP/IP Ethernet Driver Help © 2015 Kepware Technologies;http://www.kepware.com/support/manuals/siemens-tcp-ip-ethernet-manual.pdf

[5] Image Processing Toolbox™ User's Guide © COPYRIGHT 1993–2014 by The MathWorks, Inc.

[6] Image Acquisition Toolbox™ User's Guide © COPYRIGHT 2013–2015 by The MathWorks, Inc.

[7] Jae-Ho Kim, “Look-up table reduction using the relationship between pixel pitch and reconstructed distance”, Information and Communication Technology Convergence (ICTC), 2010, page(s) 205-204.

[8] L.N. Wayne, “The Automated Inspection of Moving Webs using Machine Vision” IEE Colloquium in Application of Machine Vision, 1995, page(s) 3/1-3/8.

[9] Zhang Lieping. “On Remote Real-time Communication between MATLAB and PLC Based on OPC Technology”. Control Conference Chinese (CCC), 2007, page(s) 545-548.