

การวิเคราะห์การเคลื่อนที่ของแขนมนุษย์ด้วยการประมวลผลภาพบน

ระบบปฏิบัติการลินุกซ์

Human Arm Motion Analysis Using Image Processing Under Linux

ศักดิ์สิริ ศิริสัมพันธ์ สถาพร ลักษณะเจริญ

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ เขตบางซื่อ กรุงเทพฯ 10800

E-mail: STL@kmitnb.ac.th

Saksiri Sirisampan Sathaporn Laksanacharoen

Department of Mechanical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology North
Bangkok, Bangsue, Bangkok 10800

Email: STL@kmitnb.ac.th

บทคัดย่อ

การวิเคราะห์การเคลื่อนที่ของแขนมนุษย์เป็นเครื่องมือที่สร้างขึ้นมาเพื่อศึกษาพื้นฐานการใช้งานการประมวลผลภาพสำหรับงานทางด้านวิศวกรรม โดยการพัฒนาโปรแกรมอย่างง่ายบนระบบปฏิบัติการลินุกซ์ ให้สามารถค้นหาพิกัดของจุดสีบนภาพได้ 3 จุด โดยอาศัยหลักการของการเปรียบเทียบข้อมูลทางสถิติของจุดสีบนภาพ โดยภาพจากการจับภาพวิดีโอได้ติดตั้งบนเครื่องคอมพิวเตอร์แบบโอเพ่นซอร์สทำงานด้วยระบบปฏิบัติการลินุกซ์ทะเล 7.0 ในเบื้องต้นโปรแกรมประมวลผลภาพสามารถค้นหาจุดสีที่ติดอยู่บนแขนมนุษย์ได้ที่ระดับค่าความผิดพลาดต่ำสุดประมาณ 0.29 เปอร์เซ็นต์

คำหลัก ประมวลผลภาพ ลินุกซ์ คอมพิวเตอร์

Abstract

Human arm motion analysis using image processing under linux was constructed to study the image processing. The program was to find the coordinate of three different points using the comparison method in statistic. The image was captured by a video capture card installed on desktop PC running LinuxTLE 7.0. The experiment results showed that the program can find three color points attached on a human arm with the minimum error of 0.29 percent.

Keywords: image processing, linux, motion capture

1. บทนำ

ในปัจจุบันได้มีการนำเอาการประมวลผลภาพเข้ามาใช้งานทางด้านวิศวกรรมและอุตสาหกรรมแขนงต่างๆ กันอย่างแพร่หลาย ด้วยเหตุผลทางด้านความรวดเร็วและแม่นยำในการทำงานเป็นหลัก

การประมวลผลภาพได้เข้ามามีบทบาทในอุตสาหกรรมในหลายๆ

แขนง อาทิเช่น ในอุตสาหกรรมอาหารได้มีการนำระบบประมวลผลภาพเข้ามาช่วยในการตรวจสอบและบันทึกวันและเวลาที่ใช้ในการผลิต ในวงการอุตสาหกรรมอิเล็กทรอนิกส์ก็เช่นเดียวกัน ได้มีการใช้งานการประมวลผลภาพมาเป็นเวลานานเนื่องจากข้อจำกัดในเรื่องขนาดของผลิตภัณฑ์ที่มีขนาดเล็กมากจนไม่สามารถใช้เครื่องมืออื่นใดเข้าไปจัดการได้ นอกจากภาพจากกล้องจุลทรรศน์เท่านั้น แต่ที่เห็นได้เด่นชัดและเข้าถึงผู้บริโภคได้มากที่สุดเห็นจะเป็นการใช้งานการประมวลผลภาพในการอุตสาหกรรมภาพยนตร์ ในอุตสาหกรรมภาพยนตร์นั้นได้นำเอาการประมวลผลภาพเข้ามาช่วยในการสร้างเทคนิคพิเศษให้กับภาพยนตร์เรื่องต่างๆมากมาย จนกลายเป็นเครื่องมือหลักที่ขาดไปเสียไม่ได้ในวงการอุตสาหกรรมภาพยนตร์

เครื่องมือชนิดหนึ่งซึ่งกำลังเป็นที่นิยมในวงการภาพยนตร์และถูกนำมาใช้ในภาพยนตร์หลายๆเรื่องโดยเฉพาะภาพยนตร์การ์ตูนแอนิเมชันได้แก่เครื่องมือที่เรียกกันว่าเครื่องบันทึกการเคลื่อนไหว(Motion Capture System) เครื่องมือดังกล่าวนี้มีความสามารถในการบันทึกการเคลื่อนไหวของสิ่งที่กำหนดไว้ในรูปแบบของพิกัดจุดในแกน 3 มิติ ที่เวลาต่างๆ ได้โดยอาศัยการคำนวณจากภาพที่ได้จากกล้องถ่ายภาพเป็นหลัก [1]

การเคลื่อนที่ที่ถูกบันทึกได้จากเครื่องมือดังกล่าวนี้ได้ถูกนำไปใช้ในการสร้างการเคลื่อนไหวให้กับตัวละครภายในภาพยนตร์ให้เกิดความสมจริงได้อย่างง่ายดายด้วยการบันทึกท่าทางการเคลื่อนไหวของนักแสดงเอาไว้ แล้วจึงนำเอาไปสั่งให้ตัวละครต่างๆภายในภาพยนตร์ซึ่งถูกสร้างโดยคอมพิวเตอร์ เคลื่อนที่ตามการเคลื่อนไหวที่ได้บันทึกเอาไว้แล้ว ส่งผลให้ตัวละครในภาพยนตร์เคลื่อนที่ได้สมจริงราวกับใช้นักแสดงจริง เทคนิคดังกล่าวนี้จึงเป็นที่นิยมและถูกนำมาใช้ในภาพยนตร์ต่างประเทศหลายต่อหลายเรื่อง ซึ่งนับว่าเป็นเทคโนโลยีที่ได้รับการพัฒนาให้มีความสามารถสูงขึ้นทุกวันนี้ในประเทศ

แต่สำหรับในประเทศไทยยังมีผู้คิดริเริ่มใช้งานหลักการดังกล่าวในวงจำกัด อาจเป็นเพราะว่ายังขาดความรู้พื้นฐานอยู่มาก จึงได้เกิดแนวคิด

ที่จะสร้างโครงการที่เป็นารริเริ่มการใช้งานการประมวลผลภาพในการเก็บบันทึกการเคลื่อนไหวของร่างกายมนุษย์เพื่อนำไปประยุกต์ใช้งานในด้านต่างๆได้ จึงได้เกิดเป็นโครงการเครื่องวิเคราะห์การเคลื่อนที่ของแขนมนุษย์นี้ขึ้นมาโดยมีวัตถุประสงค์เพื่อศึกษาพื้นฐานการสร้างระบบการบันทึกการเคลื่อนไหว โดยอาศัยการประมวลผลภาพจากกล้องถ่ายภาพเป็นหลัก

นอกจากความต้องการในการศึกษาการประยุกต์ใช้งานการประมวลผลภาพในเชิงวิศวกรรมดังที่ได้กล่าวไปแล้วนั้นงานวิจัยชิ้นนี้ยังมีวัตถุประสงค์แฝงอื่นไว้ด้วย นั่นคือการสาธิตการใช้งานระบบปฏิบัติการลินุกซ์ในงานเชิงวิศวกรรม เป็นการแสดงให้เห็นว่าระบบปฏิบัติการตัวนี้เหมาะสมที่ส่งเสริมให้มีการใช้งานในงานทางด้านวิศวกรรมให้มากขึ้น เนื่องจากสามารถใช้ระบบปฏิบัติการตัวนี้ในงานวิจัยได้อย่างสมบูรณ์โดยไม่ต้องเสียค่าใช้จ่ายทางด้านซอฟต์แวร์ใดๆทั้งสิ้น อีกทั้งยังมีประสิทธิภาพในการทำงานเป็นอย่างดีไม่แพ้ระบบปฏิบัติการราคาแพงตัวอื่นๆ ซึ่งงานวิจัยนี้ตั้งแต่ต้นจนจบ รวมถึงบทความนี้ได้ถูกสร้างขึ้นโดยใช้โปรแกรมแบบโอเพินซอร์ส (Open Source) บนระบบปฏิบัติการลินุกซ์ทั้งสิ้น จึงเป็นหลักฐานอย่างดีถึงประสิทธิภาพการทำงานของระบบปฏิบัติการตัวนี้ [2],[3],[4],[5],[6],[7]

2. การรับภาพจากกล้องบันทึกภาพ

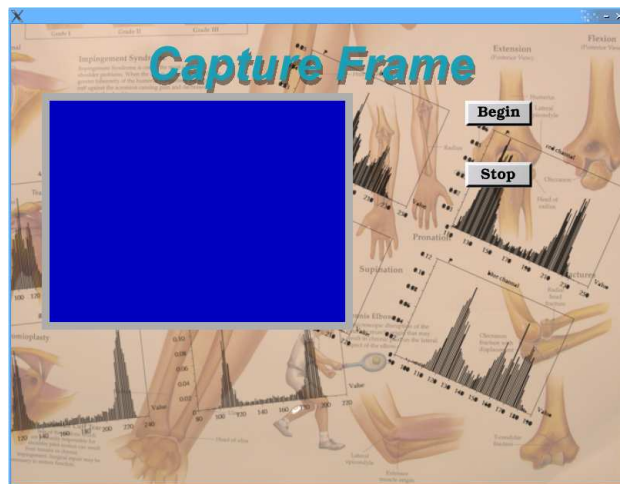
การรับภาพจากกล้องวิดีโอเข้าสู่เครื่องคอมพิวเตอร์นั้น จะต้องมอุปกรณ์ที่เรียกว่า การ์ดจับภาพวิดีโอดังที่ได้แสดงไว้รูปที่ 1 ตั้งอยู่บนเครื่องคอมพิวเตอร์ด้วย

การ์ดจับภาพวิดีโอมีหน้าที่แปลงสัญญาณภาพซึ่งเป็นสัญญาณทางไฟฟ้า ให้กลายเป็นข้อมูลภาพเก็บไว้ในหน่วยความจำของการ์ดหรือหน่วยความจำของคอมพิวเตอร์ ผู้ใช้งานคอมพิวเตอร์สามารถนำภาพที่ได้จากการ์ดจับภาพวิดีโอไปใช้งานได้โดยการอ่านข้อมูลภาพออกจากหน่วยความจำของคอมพิวเตอร์ อาจจะใช้โปรแกรมสำเร็จรูปที่เรียกว่าโปรแกรมจับภาพวิดีโอ หรือจะใช้การเขียนโปรแกรมด้วยตัวเองเพื่อดึงข้อมูลออกจากหน่วยความจำก็ได้



รูปที่ 1 การ์ดจับภาพวิดีโอ

สำหรับงานวิจัยชิ้นนี้ได้ใช้การเขียนโปรแกรมเพื่อดึงข้อมูลภาพออกจากหน่วยความจำ โดยผ่านทาง API (Application Programming Interface) ที่ชื่อว่า Video 4 Linux โดยหน้าตาของโปรแกรมที่เขียนขึ้นมาได้แสดงไว้แล้วในรูปที่ 2

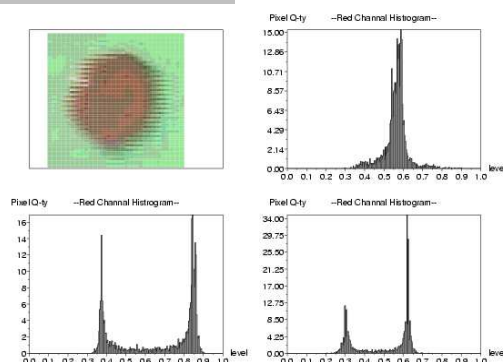


รูปที่ 2 โปรแกรมจับภาพวิดีโอที่เขียนขึ้นมา

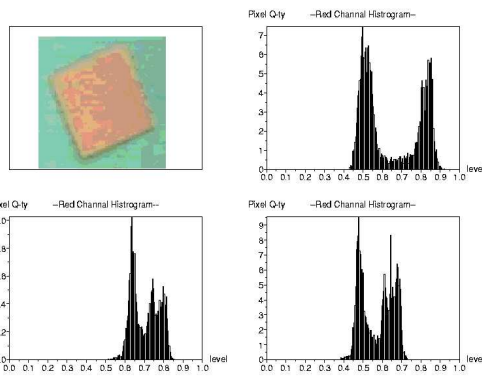
3. หลักการค้นหาลักษณะด้วยการเปรียบเทียบข้อมูลภาพ

เมื่อพิจารณาลักษณะทางโครงสร้างของข้อมูลคอมพิวเตอร์ของภาพแบบต่างๆภายในคอมพิวเตอร์ จะพบว่าแท้จริงแล้วภาพที่บันทึกลงในคอมพิวเตอร์ก็คือข้อมูลดิบ 3 ชุด ซึ่งข้อมูลแต่ละชุดเป็นค่าความสว่างของแม่สีทางแสง 3 สีคือ สีแดง สีเขียว และ สีน้ำเงิน ในแต่ละพิกเซล ดังนั้นจึงสามารถที่จะมองได้ว่าภาพ 1 ภาพก็คือข้อมูล 3 ชุดนั่นเอง เมื่อเป็นเช่นนั้นจึงสามารถที่จะนำเอาระบบการทางสถิติหรือการวิเคราะห์ข้อมูลมาประยุกต์ใช้ได้ [8][9]

จากรูปที่ 3 และ รูปที่ 4 แสดงแผนภาพฮิสโตแกรมของแม่สีทั้ง 3 สี ของภาพจุดสีบนพื้นเขียวที่มีสีแดงแตกต่างกันอยู่ โดยรูปที่ 3 นั้นเป็นภาพของจุดสีแดง และ รูปที่ 4 เป็นภาพของจุดสีส้ม โดยภาพทั้ง 2 นี้มีขนาดใกล้เคียงกัน และเป็นภาพที่ได้จากการจับภาพจากกล้องวิดีโอตัวเดียวกัน เมื่อพิจารณาที่ฮิสโตแกรมจะพบว่า รูปร่างของฮิสโตแกรมของข้อมูลแม่สีทั้ง 3 ช่องสีนั้นแตกต่างกันอย่างเห็นได้ชัดจึงเป็นเครื่องบ่งชี้ได้อย่างคร่าวๆว่าถ้าข้อมูลไปวิเคราะห์ด้วยหลักการทางสถิติอื่นๆก็น่าจะได้ผลวิเคราะห์ที่แตกต่างกันอย่างมาด้วยเช่นกันและจะสามารถนำความแตกต่างของผลการวิเคราะห์ดังกล่าวนี้มาใช้ในการแยกแยะความแตกต่างของจุดสีที่อยู่บนภาพแต่ละภาพได้



รูปที่ 3 ฮิสโตแกรมของภาพจุดสีแบบที่ 1



รูปที่ 4 ฮิสโตแกรมของภาพจุดสีแบบที่2

แต่การนำเอาภาพฮิสโตแกรมมาเปรียบเทียบกันโดยตรงเพื่อแยกความแตกต่างของจุดสีนั้นทำได้ยากและมีขั้นตอนที่ซับซ้อนไม่เหมาะที่จะนำมาใช้ในการออกแบบโปรแกรมขั้นพื้นฐาน จึงได้มีการนำเอาหลักการทางสถิติพื้นฐานเข้ามาใช้ก่อนในเบื้องต้นซึ่งเครื่องมือดังกล่าวได้แก่การหาค่าเฉลี่ย และการหาค่าส่วนเบี่ยงเบนมาตรฐาน ดังแสดงไว้ในสมการที่ (1) และ (2)

$$\bar{x} = \frac{\sum_{i=0}^N x_i}{N} \quad (1)$$

$$SD = \sqrt{\frac{\sum_{i=0}^N x_i^2}{N} - \bar{x}^2} \quad (2)$$

โดยที่

$\bar{x}$ คือ ค่าเฉลี่ยของข้อมูล

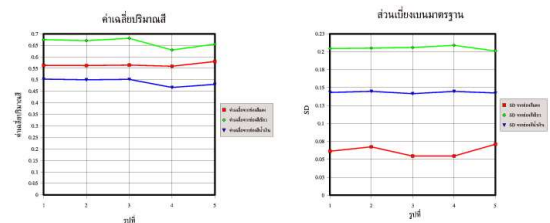
x_i คือ ค่าของข้อมูลแต่ละตัว

N คือ จำนวนข้อมูลทั้งหมด

SD คือ ค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูล

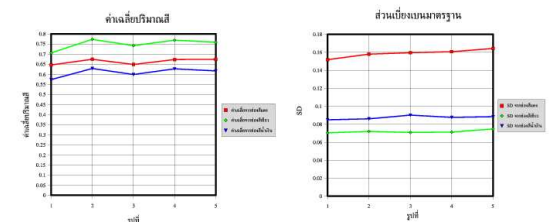
จากสมการที่ (1) และ (2) เป็นสมการสำหรับการวิเคราะห์ทางสถิติเบื้องต้นคือการวิเคราะห์ค่าเฉลี่ย และค่าส่วนเบี่ยงเบนมาตรฐานในงานวิจัยได้ใช้สมการทั้ง 2 ทดลองคำนวณหาค่าเฉลี่ยและค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูลแม่สี ของภาพตัวอย่าง 4 ชุด ชุดละ 5 ภาพและได้ผลการทดลองดังในรูปที่ 5 ถึง รูปที่ 8

รูปที่	ค่าเฉลี่ยของสีแดง	SD ของสีแดง	ค่าเฉลี่ยของสีเขียว	SD ของสีเขียว	ค่าเฉลี่ยของสีน้ำเงิน	SD ของสีน้ำเงิน
1	0.562868	0.061304	0.674798	0.204397	0.502994	0.143247
2	0.562021	0.067461	0.670128	0.205041	0.502071	0.144786
3	0.56397	0.054536	0.680136	0.205854	0.502031	0.141537
4	0.559023	0.054624	0.62947	0.208945	0.466858	0.144699
5	0.575605	0.071064	0.655053	0.201041	0.480355	0.142478
ค่าเฉลี่ย	0.5654974	0.0617982	0.6619952	0.2050556	0.4904816	0.1433484



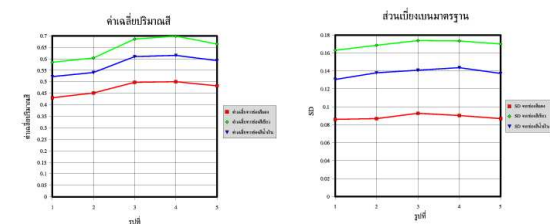
รูปที่ 5 ผลการคำนวณค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของภาพชุดที่ 1

รูปที่	ค่าเฉลี่ยของสีแดง	SD ของสีแดง	ค่าเฉลี่ยของสีเขียว	SD ของสีเขียว	ค่าเฉลี่ยของสีน้ำเงิน	SD ของสีน้ำเงิน
1	0.645746	0.151869	0.705305	0.070479	0.573122	0.084997
2	0.673914	0.157992	0.772731	0.072051	0.628296	0.086036
3	0.648273	0.159566	0.741614	0.070883	0.598463	0.090176
4	0.672807	0.160726	0.768697	0.071289	0.626612	0.087697
5	0.673927	0.164274	0.758626	0.074766	0.616193	0.088409
ค่าเฉลี่ย	0.6629334	0.1588896	0.7493646	0.0718956	0.6085372	0.0874636



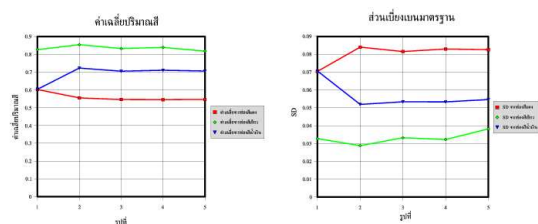
รูปที่ 6 ผลการคำนวณค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของภาพชุดที่ 2

รูปที่	ค่าเฉลี่ยของสีแดง	SD ของสีแดง	ค่าเฉลี่ยของสีเขียว	SD ของสีเขียว	ค่าเฉลี่ยของสีน้ำเงิน	SD ของสีน้ำเงิน
1	0.430768	0.086009	0.584901	0.162985	0.522627	0.130454
2	0.451613	0.086821	0.603692	0.168604	0.540464	0.137932
3	0.497813	0.092765	0.685854	0.17376	0.609643	0.140767
4	0.502611	0.090273	0.698763	0.173457	0.614936	0.143628
5	0.48255	0.086832	0.664362	0.170008	0.592369	0.137189
ค่าเฉลี่ย	0.472851	0.08854	0.6475144	0.1697666	0.5760116	0.137996



รูปที่ 7 ผลการคำนวณค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของภาพชุดที่ 3

รูปที่	ค่าเฉลี่ยจากช่องสีแดง	SD จากช่องสีแดง	ค่าเฉลี่ยจากช่องสีเขียว	SD จากช่องสีเขียว	ค่าเฉลี่ยจากช่องสีน้ำเงิน	SD จากช่องสีน้ำเงิน
1	0.602032	0.070401	0.825701	0.032754	0.603134	0.070647
2	0.555037	0.063999	0.853717	0.028797	0.722502	0.051916
3	0.545779	0.061532	0.832125	0.033253	0.704816	0.05337
4	0.545227	0.082958	0.838408	0.032268	0.710119	0.053334
5	0.546096	0.082603	0.817477	0.036194	0.705637	0.054636
ค่าเฉลี่ย	0.5586342	0.0802966	0.8334855	0.0330532	0.699282	0.0567816



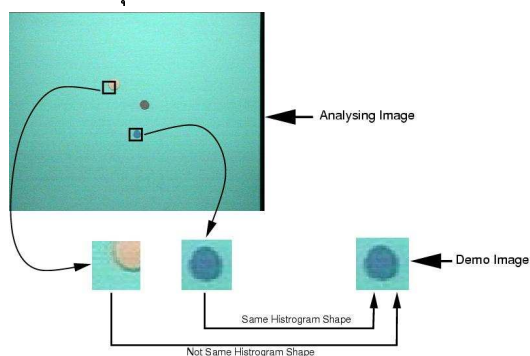
รูปที่ 8 ผลการคำนวณค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของภาพชุดที่ 4

จากผลการทดลองข้างต้นภาพ 1 ภาพจะมีค่าที่คำนวณออกมาได้ทั้งหมด 6 ค่าได้แก่ ค่าเฉลี่ยของค่าความสว่างของแม่สีแดง, ค่าเฉลี่ยของค่าความสว่างของแม่สีเขียว, ค่าเฉลี่ยของค่าความสว่างของแม่สีน้ำเงิน, ค่าส่วนเบี่ยงเบนมาตรฐานของค่าความสว่างของแม่สีแดง, ค่าส่วนเบี่ยงเบนมาตรฐานของค่าความสว่างของแม่สีเขียว, และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความสว่างของแม่สีน้ำเงิน สำหรับภาพที่มีลักษณะเหมือนหรือใกล้เคียงกัน เช่น เป็นภาพที่มีจุดสีแดงบนพื้นสีเขียวเหมือนกันและมีขนาดภาพใกล้เคียงกันพบว่าค่าที่คำนวณได้ทั้ง 6 ค่านั้นจะมีค่าใกล้เคียงกันด้วย

นอกจากนี้ยังพบว่าในกรณีที่ภาพมีจุดสีบนภาพแตกต่างกันค่าที่คำนวณได้จากภาพ 6 ค่านั้นจะต้องมีอย่างน้อย 1 ค่าที่มีความแตกต่างกันอย่างชัดเจน ดังนั้นจึงสามารถใช้หลักการนี้มาสร้างระเบียบวิธีการเขียนโปรแกรมใช้สำหรับเปรียบเทียบความเหมือนหรือต่างของจุดสีบนภาพได้ โดยกำหนดให้โปรแกรมคำนวณและเปรียบเทียบค่าทั้ง 6 ค่าดังกล่าว ถ้าผลการเปรียบเทียบพบว่ามีค่าแตกต่างกันเกิน 1 ค่าใน 6 ค่านั้นจะสรุปได้ว่ารูปที่นำมาเปรียบเทียบกับนั้นแตกต่างกันโดยสิ้นเชิง แต่ถ้าทั้ง 6 ค่าดังกล่าวมีค่าใกล้เคียงกัน ก็จะสรุปได้ว่ารูปที่นำมาเปรียบเทียบทั้ง 2 มีความคล้ายกันหรือเหมือนกัน

4. การทำงานโปรแกรมประมวลผลภาพ

จากหลักการการเปรียบเทียบความเหมือน ความแตกต่างของภาพด้วยข้อมูลทางสถิติตามที่กล่าวไว้ในหัวข้อที่แล้วนั้น สามารถนำมาสร้างหลักการค้นหาจุดสีจากภาพได้



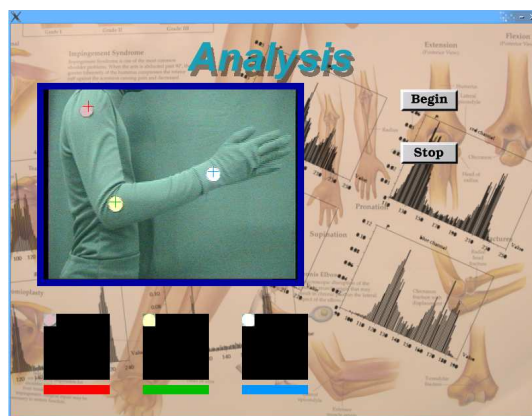
รูปที่ 9 หลักการค้นหาจุดสีจากภาพ
จากรูปที่ 9 แสดงหลักการของการค้นหาตำแหน่งของจุดสีจากภาพ

โดยการค้นหาตำแหน่งของจุดสีเริ่มจากการเตรียมตัวอย่างของจุดสีที่ต้องการค้นหาเอาไว้ก่อน เรียกว่า Demo Image โดยตัวอย่างภาพนี้จะต้องทราบขนาดความกว้างและความสูงของภาพ จากรูปที่ 9 ภาพตัวอย่างจุดสีเป็นภาพจุดสีวงกลมสีน้ำเงินบนพื้นสีเขียว การค้นหาเริ่มด้วยการตัดภาพขนาดเท่ากับภาพตัวอย่าง โดยเริ่มจากมุมบนด้านซ้ายก่อน จากนั้นจึงคำนวณค่าเฉลี่ย และ ค่าส่วนเบี่ยงเบนมาตรฐานจากช่องสี ทั้ง 3 ช่องจากภาพที่ตัดออกมา และภาพตัวอย่าง รวมทั้งหมด 12 ค่า (ภาพละ 6 ค่า) มาเปรียบเทียบกันตัวต่อตัว (ค่าเฉลี่ยจากช่องสีแดงของภาพตัวอย่างเทียบกับค่าเฉลี่ยจากช่องสีแดงของภาพที่ตัดออกมา เป็นต้น) ถ้าผลการเปรียบเทียบพบว่า ค่าทั้ง 6 คู่ที่คำนวณได้นั้นเหมือนหรือใกล้เคียงกัน แสดงว่าบริเวณที่ตัดภาพมานั้นเป็นภาพจุดสีน้ำเงินบนพื้นสีเขียวเหมือนกัน แต่ถ้าผลออกมาปรากฏว่าค่าทั้ง 6 คู่ไม่มีแม้แต่คู่ใดคู่หนึ่ง ต่างกันโดยสิ้นเชิง จะสรุปได้ว่าภาพบริเวณที่ตัดออกมานั้นไม่มีจุดสีน้ำเงินบนพื้นสีเขียวอยู่ ให้ตัดภาพจากพื้นที่อื่น ๆ มาวิเคราะห์ต่อไป

5. การทดสอบโปรแกรมประมวลผลภาพ

จากหลักการการประมวลผลภาพดังที่ได้กล่าวไปแล้วในหัวข้อที่แล้ว ได้มีการสร้างโปรแกรมประมวลผลภาพขึ้นมาดังแสดงไว้ในรูปที่ 10 พบว่าโปรแกรมสามารถค้นหาจุดสีที่ปะอยู่บนขนของมนุษย์ที่เคลื่อนไหวอยู่ได้ แต่พบว่าในหลายๆครั้ง โปรแกรมมีการค้นหาผิดพลาดหรือค้นหาไม่ได้เลย

จึงได้มีการทดสอบหาความสามารถในการประมวลผลภาพของโปรแกรมที่สร้างขึ้นมาด้วยการทดลองประมวลผลภาพ จากภาพที่ไม่มีการเคลื่อนไหว เพื่อให้ทราบความสามารถสูงสุดของโปรแกรม



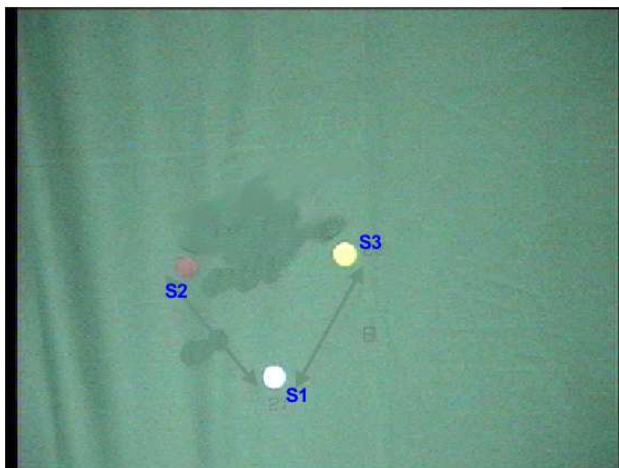
รูปที่ 10 หน้าต่างโปรแกรมประมวลผลภาพที่สร้างขึ้น

5.1 วิธีการทดสอบ

การทดสอบโปรแกรมประมวลผลภาพใช้การถ่ายภาพจุดสี 3 จุดที่ทราบระยะห่างระหว่างจุดสีอยู่ก่อนแล้ว ดังที่ได้แสดงไว้ในรูปที่ 11 จากรูปดังกล่าว ระยะจากจุด s1 ไปยังจุด s2 มีค่าเท่ากับ 190 มิลลิเมตร ระยะจากจุด s2 ไปยังจุด s3 มีค่าเท่ากับ 225 มิลลิเมตร และระยะระหว่างจุด s3 ไปยังจุด s1 มีค่าเท่ากับ 195 มิลลิเมตร

โดยจะใช้โปรแกรมจับภาพที่เขียนขึ้นมาก่อนหน้านั้นที่ภาพจุดสีนี้แล้วเก็บไว้ในคอมพิวเตอร์เป็นจำนวนทั้งหมด 108 ภาพ จากนั้นจึงนำ

เอาภาพทั้ง 108 ภาพนี้ มาวิเคราะห์โดยใช้โปรแกรมประมวลผลภาพที่สร้างขึ้นมา โดยโปรแกรมจะค้นหาจุดสี สีต่างๆตามที่กำหนดไว้จาก หลังจากที่ได้ค้นหาจนพบแล้ว โปรแกรมก็จะบันทึกตำแหน่งของจุดสีเทียบกับมุมบนด้านซ้ายของภาพ หน่วยเป็นมิลลิเมตรลงสู่คอมพิวเตอร์

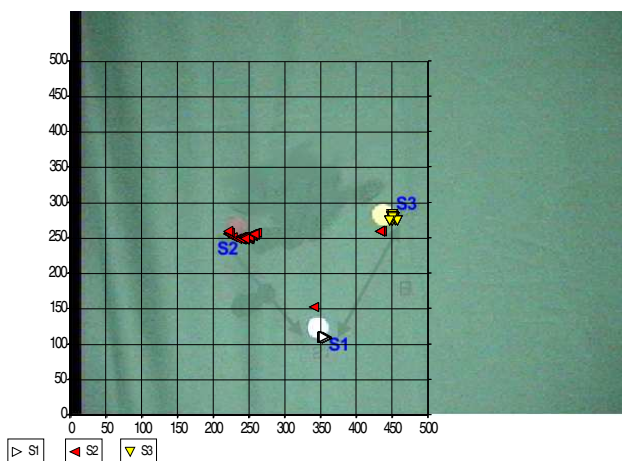


รูปที่ 11 ภาพสำหรับใช้ทดสอบโปรแกรม

เมื่อได้พิกัดของจุดสีต่างๆแล้ว ก็นำเอาพิกัดเหล่านั้นมาคำนวณหาระยะทางระหว่างจุดสีต่างๆ แล้วนำเอามาเปรียบเทียบกับค่าจริง

5.2 ผลการทดสอบ

เมื่อโปรแกรมประมวลผลภาพได้ค้นหาจุดและบันทึกพิกัดลงสู่คอมพิวเตอร์เรียบร้อยแล้ว จึงได้มีการนำเอาค่าพิกัดเหล่านั้นมาหาผลลงบนภาพจริงดังแสดงไว้ในรูปที่ 12



รูปที่ 12 ผลการทดลอง

จากรูปที่ 12 สามเหลี่ยมสีขาวแสดงพิกัดที่ค้นหาได้ของจุด s1 สามเหลี่ยมสีแดงแสดงพิกัดที่ค้นหาได้ของจุด s2 และสามเหลี่ยมสีเหลืองแสดงพิกัดของจุด s3 และพบว่า โปรแกรมสามารถค้นหาจุดสีขาวได้ช้าที่สุด รองลงมาคือ จุด s3 สำหรับจุด s2 นั้นได้ผลการค้นหาที่ต่ำที่สุด กล่าวคือนอกจากการค้นหาจุดในแต่ละครั้งจะไม่ใกล้เคียงกับจุดสีแดง

แล้ว ยังมีหลายครั้งที่พิกัดที่ได้จากการค้นหาอยู่ห่างจากจุดสีแดงมาก

เมื่อนำเอาพิกัดของจุดที่ค้นหาได้ทั้ง 108 ภาพ มาคำนวณหาระยะทางจุดสีได้ผลโดยสรุปดังตารางที่ 1

ตารางที่ 1 ผลการคำนวณระยะทางระหว่างจุดสีโดยสรุป (หน่วยเป็นมิลลิเมตร)

	ระยะ s1-s2	ระยะ s1-s3	ระยะ s1-s2
ค่าเฉลี่ย	176.410	195.562	198.864
ค่าสูงสุด	200.377	198.408	232.625
ค่าต่ำสุด	44.1761	188.984	24.2436
ค่าจริง	190.000	195.000	225.000
ค่าผิดพลาดจากค่าเฉลี่ย	13.5892	-0.562847	26.1350
เปอร์เซ็นต์ผิดพลาด (%)	7.15224	0.288639	11.6155

จากตารางที่ 1 และการวาดพิกัดของจุดลงบนภาพต้นฉบับจะพบว่า โปรแกรมประมวลผลภาพสามารถค้นหาจุดสีขาวได้ถูกต้องแม่นยำที่สุด รองลงมาคือสีเหลือง และต่ำสุดคือสีแดง ส่งผลให้ความแม่นยำของการคำนวณระยะระหว่างจุดสีขาว กับจุดสีเหลือง สูงที่สุด อยู่ที่ประมาณ 0.3 เปอร์เซ็นต์

6. บทสรุป

จากที่ได้ดำเนินการวิจัยมาทั้งหมดสรุปได้ว่าโปรแกรมประมวลผลภาพเพื่อใช้สำหรับการวัดการเคลื่อนที่ของแขนมนุษย์จากภาพนั้น ยังไม่สามารถทำงานได้อย่างสมบูรณ์ สามารถที่จะค้นหาจุดสีจากภาพได้บางช่วงเท่านั้น เมื่อแขนเคลื่อนที่ไปในบางครั้งก็อาจเกิดการบิดขึ้นทำให้จุดสีที่ติดไว้ที่แขนไม่อยู่บนระนาบ ทำให้โปรแกรมไม่สามารถค้นหาจุดสีที่ติดไว้ได้

นอกจากนี้ จากการทดสอบโดยใช้จุดสีที่ติดอยู่กับฉากที่อยู่กับที่ ยังพบว่าการค้นหาจุดสียังมีความผิดพลาดได้ โดยเฉพาะจุดสีแดงนั้นให้ผลการค้นหาอยู่ในระดับที่ต่ำที่สุดทั้งนี้เนื่องจากคุณภาพของภาพที่ได้จากการจับภาพวิดีโออยู่ยังอยู่ในระดับต่ำ มีความเพี้ยนของสี ทำให้จุดสีกลืนไปกับฉาก ทำให้โปรแกรมประมวลผลภาพมีความแม่นยำในการค้นหาจุดสีได้ค่อนข้างต่ำ

แต่อย่างไรก็ตามการวิจัยนี้ได้มีการศึกษาค้นคว้าวิธีการจับภาพบนระบบปฏิบัติการลินุกซ์ และการประมวลผลภาพเบื้องต้น ซึ่งวิธีการและหลักการเหล่านี้จะเป็นข้อมูลพื้นฐานที่ดีในการพัฒนา และต่อยอดงานวิจัยนี้ให้สามารถใช้งานได้จริงต่อไปในอนาคต

เอกสารอ้างอิง

- [1] Behind the scene of the Matrix. Available online at www.thematrix.com
- [2] Bill Ball, Stephen Smoogen. Tech Yourself Linux in 24 Hours. Indiana USA : Sams Publishing and Red Hat Press, 1998
- [3] GNU General Public License. Available online at www.gnu.org
- [4] K. S. Fu, R. C. Gonzales. Robotics Control, Sensing, Vision, and Intelligence. New York USA : McGraw-Hill Inc, 1987

- [5] Linux TLE 7.0. Available online at
www.opentle.org
- [6] Loki Software Inc. Linux Game Programing. San
Francisco USA : No Starch Press Inc, 2001
- [7] Mark Mitchell, Jeffrey Oldham and Alex Samuel.
Advanced Linux Programming. Indiana USA :
New Riders Publishing, 2001
- [8] Sensors Vision and Measurement 2002-2003.
Japan : KEYENCE COPERATION, 2002
- [9] Sonka, Milan. Image processing, analyis, and
machine vision. USA : BROOKS/COLE
PUBLISHING COMPANY, 1998