

สถาปัตยกรรมประสิทธิภาพสูงเพื่อตรวจจับและบรรยายจุดเด่นของภาพบนพื้นฐานของขั้นตอนวิธีไออาร์บี



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

High-Efficient Architecture for Image Feature Detection and Description based on ORB
algorithm



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	สถาปัตยกรรมประสิทธิภาพสูงเพื่อตรวจจับและบรรยาย จุดเด่นของภาพบนพื้นฐานของขั้นตอนวิธีไออาร์บี
โดย	น.ส.ทักษพร อิ่มแสงสุข
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	
.....	ประธานกรรมการ
(รองศาสตราจารย์ ดร.วันเฉลิม โปธา)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.เอกชัย ลีลาวัณย์)	

วิทยาลัย
CHULALONGKORN UNIVERSITY

ทักษพร อิ่มแสงสุข : สถาปัตยกรรมประสิทธิภาพสูงเพื่อตรวจจับและบรรยายจุดเด่นของ
 ภาพบนพื้นฐานของขั้นตอนวิธีโออาร์บี. (High-Efficient Architecture for Image
 Feature Detection and Description based on ORB algorithm) อ.ที่ปรึกษาหลัก
 : ผศ. ดร.สุรีย์ พุ่มรินทร์

วิทยานิพนธ์ฉบับนี้นำเสนอสถาปัตยกรรมประสิทธิภาพสูงเพื่อตรวจจับและบรรยาย
 จุดเด่นบนพื้นฐานของขั้นตอนวิธีโออาร์บี สถาปัตยกรรมถูกบรรยายด้วยภาษาวีเอสดีแอลและถูก
 ออกแบบให้สามารถประมวลผลบนเอฟพีจีเอได้ ขั้นตอนวิธีการที่ นำไปใช้ออกแบบสถาปัตยกรรม
 นั้นถูกพัฒนาขึ้นด้วยภาษาไพทอนเพื่อลดความซับซ้อนและปรับปรุงขั้นตอนวิธีให้เหมาะสมกับการ
 ทำงานของฮาร์ดแวร์ โดยประยุกต์ใช้การประมวลผลแบบขนาน, การจัดระบบงานแบบสายท่อ
 (parallel and pipeline techniques) และเปลี่ยนวิธีในการประมวลผลบางขั้นตอนให้สามารถ
 คำนวณได้โดยไม่ต้องอาศัยวงจรคำนวณเชิงคณิตศาสตร์เพิ่มเติมในการเร่งการทำงานของระบบ
 จากผลการทดสอบสถาปัตยกรรมด้วยโปรแกรม ModelSim พบว่าสถาปัตยกรรมสามารถรองรับ
 สัญญาณภาพขนาด 1920x1080 จุดภาพอัตรา 30 เฟรมต่อวินาทีได้ โดยยังคงค่าความต้องกันของ
 จุดมุ่มและความแม่นยำเมื่อเปรียบเทียบกับผลที่ได้จากฟังก์ชัน cv.ORB() ใน OpenCV library

จุฬาลงกรณ์มหาวิทยาลัย
 CHULALONGKORN UNIVERSITY

สาขาวิชา วิศวกรรมไฟฟ้า
 ปีการศึกษา 2563

ลายมือชื่อนิสิต
 ลายมือชื่อ อ.ที่ปรึกษาหลัก

6270102221 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Feature detection and description, ORB algorithm, FPGA, Image Processing

Taksaporn Imsaengsuk : High-Efficient Architecture for Image Feature Detection and Description based on ORB algorithm. Advisor: Asst. Prof. SUREE PUMRIN, Ph.D.

This thesis proposes a high-efficient architecture for feature detection and description based on the ORB algorithm. The proposed architecture is described in VHDL and designed for Field Programmable Gate Arrays. The algorithm is developed by Python to reduce the operation of the process and be beneficial for designing FPGA architecture. The proposed algorithm adopts parallel and pipeline techniques and changes some steps to not including complex mathematical circuits for accelerating the system. The ModelSim simulation results show that the proposed architecture can process the 1920x1080 pixels and support 30 frames per second video source. The architecture could maintain the consistency of the corner numbers and the accuracy compared with the results obtained by cv.ORB() function in OpenCV library.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Electrical Engineering

Student's Signature

Academic Year: 2020

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์ อาจารย์ที่ปรึกษา ที่ให้คำปรึกษาแนวทางการดำเนินงานวิจัยและแนวทางในการคิดทั้งในเรื่องวิชาการ การศึกษาต่อและการดำเนินชีวิต

ขอขอบคุณโครงการศิษย์ก้นกุฏิ ระดับปริญญาโท (EECU Master Honors Program Scholarship) ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้การสนับสนุนค่าลงทะเบียนเรียน ค่าใช้จ่ายประจำเดือน เงินสนับสนุนการไปเสนอผลงานวิจัยในที่ประชุมวิชาการ และโอกาสในการเป็นผู้ช่วยสอน ซึ่งช่วยให้เห็นแนวคิดใหม่ ความคิดสร้างสรรค์จากน้องๆนิสิตในรายวิชาที่ได้ช่วยสอน

ขอขอบคุณอาจารย์ พี่และเพื่อนในห้องปฏิบัติการวิจัยการออกแบบและประยุกต์วงจรรวมที่ให้การสนับสนุน รับฟังปัญหาและช่วยให้คำปรึกษาทั้งในเรื่องวิชาการและเรื่องการดำเนินชีวิต

ขอขอบคุณ ดร.วิวัฒน์ วงศ์วรวิภัทร์, คุณธนพร แสงไพฑูรย์, คุณภทธีรา อุทัยชนะ และพี่ๆ บริษัท Design gateway ที่ให้คำปรึกษาและแนวทางในการดำเนินงานวิจัยที่เกี่ยวข้องกับการออกแบบวงจรดิจิทัล

สุดท้ายนี้ข้าพเจ้าขอขอบคุณครอบครัวของข้าพเจ้า คุณพ่อ คุณแม่และน้องๆ ที่เป็นกำลังใจดูแลเอาใจใส่และให้การสนับสนุนในทุกการตัดสินใจของข้าพเจ้าตลอดมา

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ทักษพร อิ่มแสงสุข

สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ฅ
สารบัญรูปภาพ.....	ญ
1. บทนำ	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตของวิทยานิพนธ์.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ขั้นตอนวิธีโออาร์บี (Oriented FAST and Rotated BRIEF).....	3
2.1.1 การหาลักษณะด้วยวิธีการโอฟาสต์ (oFAST Feature Detection).....	3
2.1.2 การสร้างพีระมิดรูปภาพ	5
2.1.3 การขจัดค่าที่ไม่มากที่สุด (Non maxima suppression)	5
2.1.4 การคำนวณทิศทางของจุดเด่น (Orientation Computing).....	5
2.1.5 อาร์บรีฟ (rBRIEF : rotated BRIEF descriptor)	6

2.2 การประเมินคุณภาพ	6
3. สถาปัตยกรรมที่ออกแบบ	9
3.1 การสร้างพีระมิดรูปภาพ.....	11
3.2 วินโดว์บัพเฟอร์ขนาด 7x7.....	13
3.3 การตรวจจับลักษณะภาพด้วยขั้นตอนวิธีฟาสต์และการคำนวณคะแนน	14
3.4 การขจัดจุดเด่นที่มีค่าคะแนนความเป็นมุมไม่มากที่สุด	15
3.4 ตัวจัดเรียงลำดับจุดมุมตามค่าความเป็นมุมด้วยขั้นตอนวิธีแบบฮีป	16
3.6 ตัวกรองแบบทวินาม	17
3.7 การส่งข้อมูลแผ่นภาพขนาด 31x31 จุดภาพ.....	18
3.8 การคำนวณโมเมนต์ของจุดมุม.....	18
3.9 การประมาณทิศทางของจุดเด่น	20
3.7 การหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุม	20
3.8 การสร้างตัวบรรยายแบบบริฟ.....	22
4. ผลการทดลอง.....	23
4.1 การปรับปรุงขั้นตอนวิธีการด้วยภาษาไพทอน.....	23
4.2 การจำลองการทำงานด้วยโปรแกรม ModelSim	27
4.3 ผลการสังเคราะห์วงจรและการใช้ทรัพยากร.....	30
5. สรุปผล.....	31
5.1 สรุปผล.....	31
5.2 ข้อเสนอแนะ	31
ภาคผนวก.....	33
บทความที่ได้ตีพิมพ์ใน iEECON 2021	33
บรรณานุกรม.....	2
ประวัติผู้เขียน.....	5



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

	หน้า
ตารางที่ 1 ความซับซ้อนและแพลตฟอร์มของขั้นตอนวิธีจัดเรียง	4
ตารางที่ 2 ความต้องกันของจุดมุมและความแม่นยำที่ได้จากขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับผลที่ได้จากฟังก์ชัน cv.ORB() ใน OpenCV library	26
ตารางที่ 3 การเปรียบเทียบการใช้ทรัพยากรและผลการทำงานระหว่างวิทยานิพนธ์และงานวิจัยที่เกี่ยวข้อง	30



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูปร่างภาพ

	หน้า
รูปที่ 1 จุดภาพที่ใช้พิจารณาความเป็นมุมของจุดภาพ	3
รูปที่ 2 ภาพรวมระบบการตรวจจับลักษณะและสร้างตัวบรรยายของภาพ.....	9
รูปที่ 3 สถาปัตยกรรมสำหรับระบบการตรวจจับจุดเด่นที่เป็นมุมและสร้างตัวบรรยายของภาพ.....	10
รูปที่ 4 พีระมิดรูปร่างภาพที่ประกอบด้วยผลการลดการสุมตัวอย่างแบบหลักและแบบย่อย.....	11
รูปที่ 5 สถาปัตยกรรมของการลดการสุมตัวอย่างแบบหลัก.....	12
รูปที่ 6 สถาปัตยกรรมของการลดการสุมตัวอย่างแบบย่อย.....	13
รูปที่ 7 ตำแหน่งจุดภาพที่ใช้ทดสอบสำหรับขั้นตอนวิธีแบบฟาสต์	14
รูปที่ 8 สถาปัตยกรรมสำหรับขั้นตอนวิธีฟาสต์และตัวคำนวณคะแนนความเป็นมุม	15
รูปที่ 9 ไลน์บัฟเฟอร์สำหรับการขจัดจุดเด่นที่มีค่าคะแนนความเป็นมุมไม่มากที่สุด	15
รูปที่ 10 สถาปัตยกรรมสำหรับขั้นตอนวิธีเรียงลำดับแบบฮีป.....	16
รูปที่ 11 สถาปัตยกรรมของตัวกรองทวินาม	17
รูปที่ 12 วงจรคุณค่าความเข้มของจุดภาพกับค่าประจำหลัก 1 ถึง 15.....	19
รูปที่ 13 วงจรคำนวณค่าโมเมนต์ของจุดเด่น	19
รูปที่ 14 สถาปัตยกรรมการประมาณทิศทางจุดเด่น	20
รูปที่ 15 สถาปัตยกรรมสำหรับหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุม	21
รูปที่ 16 สถาปัตยกรรมสำหรับสร้างตัวบรรยายแบบบริฟ.....	22
รูปที่ 17 ตัวอย่างภาพทดสอบ	23
รูปที่ 18 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ	24
รูปที่ 19 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ	25
รูปที่ 20 ผลการจำลองการทำงานของสถาปัตยกรรมส่วนตรวจจับจุดเด่นด้วยโปรแกรม ModelSim	28

รูปที่ 21 ผลการจำลองการทำงานของสถาปัตยกรรมส่วนสร้างตัวบรรยายด้วยโปรแกรม ModelSim 29

รูปที่ 22 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ 30



1. บทนำ

1.1 ที่มาและความสำคัญ

ภาพถูกนำมาใช้อย่างแพร่หลายในการนำมาประมวลผลเพื่อหาข้อมูลต่างๆ ในบางครั้งภาพที่ต้องการเก็บเพื่อใช้ในการประมวลผลมีขนาดใหญ่จนเซ็นเซอร์หรือกล้อง 1 ตัวไม่สามารถเก็บข้อมูลได้หมด จำเป็นต้องใช้เซ็นเซอร์หลายตัวตรวจจับภาพในแต่ละตำแหน่งและนำมาต่อกัน การนำภาพมาต่อกัน เรียกว่า การเรียงภาพย่อยเพื่อประกอบเป็นภาพใหญ่ (image registration and stitching) การเรียงภาพย่อยเพื่อประกอบเป็นภาพใหญ่ถูกนำมาใช้อย่างหลากหลาย เช่น ในการสร้างภาพทางการแพทย์ของเครื่องเอ็มอาร์ไอ (MRI), ในการสร้างแผนที่ด้วยภาพถ่ายจากดาวเทียมหรือภาพถ่ายทางอากาศ เป็นต้น

การนำภาพย่อยมาเรียงต่อกันนั้น ต้องหาตำแหน่งที่เหมือนกันของภาพแต่ละภาพมาเรียงซ้อนกันในแต่ละตำแหน่งที่ถูกต้อง ขั้นตอนวิธีที่ใช้ในการรวมภาพย่อยเป็นภาพใหญ่ เริ่มจากการเลือกลักษณะ (feature) ที่จะใช้ในการหาจุดเด่น (keypoint) ในแต่ละภาพ สร้างตัวบรรยาย (descriptor) เพื่อบรรยายลักษณะของแต่ละจุดเด่นเหล่านั้น จากนั้นจึงจับคู่จุดเด่นที่มีตัวบรรยายเหมือนหรือใกล้เคียงกันมากที่สุดจะถือเป็นจุดเดียวกัน การตรวจจับจุดเด่นและการบรรยายจุดเด่นมีการพัฒนาขั้นตอนวิธีการมาอย่างต่อเนื่อง ตั้งแต่ ขั้นตอนวิธีเอสไอเอฟที (Scale-Invariant Feature Transform : SIFT) [1] มิมุม (corner) เป็นลักษณะเด่น และเป็นการหาจุดเด่นโดยไม่ขึ้นกับการขยาย (scale-invariance) ขั้นตอนวิธีการเอสยูอาร์เอฟ (Speeded-Up Robust Features : SURF) [2] ขั้นตอนวิธีการฟาสต์ (Feature from Accelerated Segment Test : FAST) [3] ขั้นตอนวิธีการบริฟ (Binary Robust Independent Elementary Features : BRIEF) [4] หรือ ขั้นตอนวิธีโออาร์บี (Oriented FAST and Rotated BRIEF : ORB) [5] สำหรับขั้นตอนวิธีแบบโออาร์บี เป็นขั้นตอนวิธีที่นิยมในงานวิจัยและการนำมาประยุกต์ใช้กับต่อภาพถ่ายดาวเทียมและการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) เนื่องจากเป็นขั้นตอนวิธีที่ไม่ซับซ้อน และมีการปรับปรุงให้ไม่ขึ้นกับการขยายขนาด (scale-invariance) และไม่ขึ้นกับการหมุน (rotation-invariance)

สำหรับภาพที่มีความละเอียดสูงและขนาดใหญ่ การตรวจจับจุดเด่นและสร้างตัวบรรยายของจุดเด่นแต่ละจุดต้องอาศัยเวลาในการประมวลผลมาก หรือหากต้องการให้ระบบเป็นระบบในเวลาจริง (real-time system) จำเป็นต้องใช้พลังในการประมวลผล (computing power) มากตามไปด้วย ถือเป็นคอขวดของระบบ วิธีหนึ่งในการเร่งความเร็วในการประมวลผล คือ การใช้อุปกรณ์ลอจิกแบบโปรแกรมได้หรือเอฟพีจีเอ (Field Programmable Gate Arrays : FPGAs) ซึ่งสามารถประมวลผลแบบขนานได้ (parallel processing) ทำงานในส่วนการหาจุดเด่นและสร้างตัวบรรยายทดแทนการใช้ตัวประมวลผลกลาง (CPU) ที่ทำงานแบบลำดับ (sequential processing) อีกทั้งขั้นตอนย่อยในขั้นตอนวิธีการตรวจจับและสร้างตัวบรรยายแบบโออาร์บีบางขั้นตอน มีความซับซ้อนไม่เหมาะกับการนำไปใช้กับฮาร์ดแวร์อย่างเอฟพีจีเอ ซึ่งถือเป็นอีกคอขวดหนึ่งที่ทำให้ใช้ความสามารถของฮาร์ดแวร์ได้ไม่เต็มที่

ในวิทยานิพนธ์นี้ ขั้นตอนวิธีการและสถาปัตยกรรมเพื่อตรวจจับจุดเด่นที่เป็นมุมและบรรยายจุดเด่นของภาพจะถูกออกแบบบนพื้นฐานของขั้นตอนวิธีโออาร์บีให้มีความซับซ้อนในการคำนวณน้อยลง ลักษณะการคำนวณสอดคล้องกับหลักการทำงานและการใช้ทรัพยากรของเอพีพีจีเอมากขึ้น ทำให้สามารถเพิ่มความสามารถในการคำนวณและประหยัดทรัพยากรได้

1.2 วัตถุประสงค์

1. เพื่อนำเสนอขั้นตอนวิธีการตรวจจับจุดเด่นที่เป็นมุมและสร้างตัวบรรยายบนพื้นฐานของขั้นตอนวิธีการโออาร์บีที่มีความซับซ้อนในการประมวลผลน้อยลง สามารถประมวลผลแบบขนานได้
2. เพื่อนำเสนอสถาปัตยกรรมของขั้นตอนวิธีการตรวจจับมุมและสร้างตัวบรรยายที่นำเสนอบนฮาร์ดแวร์เอพีพีจีเอ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเพิ่มประสิทธิภาพในการใช้ทรัพยากรของฮาร์ดแวร์และลดเวลาที่ใช้ในการตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยายบนพื้นฐานของขั้นตอนวิธีแบบโออาร์บี
2. ผู้ใช้สามารถนำสถาปัตยกรรมไปประยุกต์ใช้เพิ่มประสิทธิภาพในระบบประมวลผลภาพที่ต้องการตรวจจับจุดเด่นที่เป็นมุมและสร้างตัวบรรยายแบบไบนารี

1.4 ขอบเขตของวิทยานิพนธ์

1. ออกแบบขั้นตอนวิธีการตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยายไบนารีขนาด 256 บิตบนพื้นฐานของขั้นตอนวิธีโออาร์บีที่มีความต้องการของจุดมุมและความแม่นยำไม่ต่ำกว่า 70 เปอร์เซ็นต์
2. นำเสนอสถาปัตยกรรมเพื่อตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยายไบนารีบนพื้นฐานของขั้นตอนวิธีโออาร์บีที่รองรับสัญญาณภาพความละเอียดไม่เกิน 1920x1080p ที่อัตรา 30 เฟรมต่อวินาที

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษางานวิจัยที่เกี่ยวข้องกับขั้นตอนวิธีการตรวจจับจุดเด่นและการบรรยายลักษณะของจุดเด่นแบบต่างๆ
2. ศึกษาหลักการและงานวิจัยที่เกี่ยวข้องกับขั้นตอนวิธีตรวจจับมุมและบรรยายจุดเด่นด้วยขั้นตอนวิธีโออาร์บี
3. ออกแบบขั้นตอนวิธีในการตรวจจับมุมและบรรยายจุดเด่นบนพื้นฐานของขั้นตอนวิธีโออาร์บีด้วยภาษาไพทอน (Python)
4. ทดสอบการทำงานและความแม่นยำของขั้นตอนวิธีที่นำเสนอด้วยภาษาไพทอน โดยเปรียบเทียบกับผลที่ได้จากฟังก์ชัน cv.ORB() ใน OpenCV library
5. ออกแบบและจำลองผลการทำงานสถาปัตยกรรมของขั้นตอนวิธีการที่นำเสนอด้วยภาษาวีเอชดีแอล (VHDL: The Very High Speed Integrated Circuit (VHSIC) Hardware Description Language)
6. ประเมินคุณภาพและพัฒนาสถาปัตยกรรมขั้นตอนวิธีการที่นำเสนอด้วยการจำลองผ่านโปรแกรม ModelSim
7. สรุปผลการและเขียนวิทยานิพนธ์

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ขั้นตอนวิธีโออาร์บี (Oriented FAST and Rotated BRIEF)

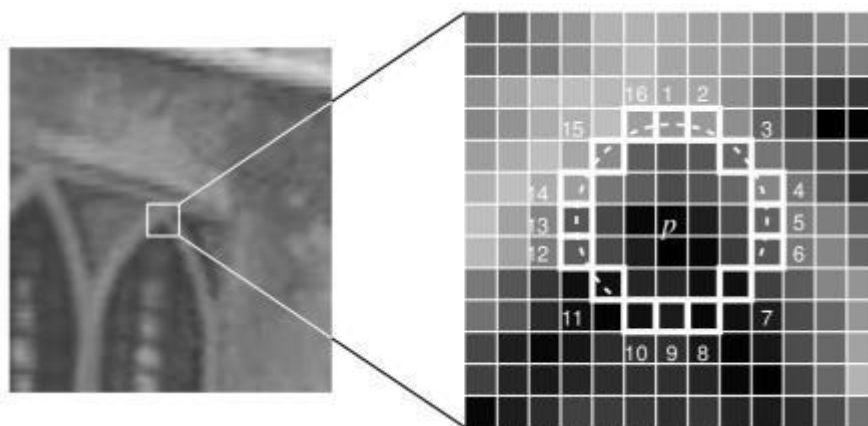
ขั้นตอนวิธีโออาร์บี เป็นขั้นตอนวิธีการหาจุดเด่นโดยพิจารณาจากมุมที่ปรากฏในภาพ และสร้างตัวบรยายแต่ละจุดเด่นแบบไบนารี (Binary descriptor) การหาจุดเด่น ใช้ขั้นตอนวิธีการแบบฟาสต์ แต่พิจารณาการวางทิศทางของมุม (Orientation) ด้วย เรียก โอฟาสต์ (Oriented FAST : oFAST) การวางทิศทางนี้จะถูกนำไปใช้ในขั้นตอนการสร้างตัวบรยายร่วมกับขั้นตอนวิธีการแบบบริฟ รวมเรียก อาร์บริฟ (Rotated BRIEF : rBRIEF) โดยมีรายละเอียดดังต่อไปนี้

2.1.1 การหาลักษณะด้วยวิธีการโอฟาสต์ (oFAST Feature Detection)

ขั้นตอนวิธีนี้ใช้การตรวจจับจุดเด่นจากลักษณะการเป็นมุม โดยจุดภาพแต่ละจุดภาพจะถูกจัดว่าเป็นมุมจากการพิจารณาความเข้มของจุดภาพที่อยู่บนวงแหวนรอบจุดที่สนใจเป็นรัศมี r จุดภาพ จุดภาพที่อยู่บนวงกลมจะถูแบ่งออกเป็น 3 ประเภท ได้แก่

1. จุดภาพที่สว่างกว่าจุดภาพศูนย์กลาง คือ จุดภาพที่มีความเข้มมากกว่าความเข้มที่จุดศูนย์กลางที่กำลังพิจารณาอยู่มากกว่าหรือเท่ากับค่าขีดแบ่ง (threshold : t)
2. จุดภาพที่ความสว่างเท่ากับจุดศูนย์กลาง คือ จุดภาพที่มีความเข้มของจุดภาพมากกว่าค่าความเข้มของจุดศูนย์กลางไม่เกินค่าขีดแบ่งหรือมีความเข้มน้อยกว่าจุดศูนย์กลางไม่เกินค่าขีดแบ่ง
3. จุดภาพที่มืดกว่าจุดศูนย์กลาง คือ จุดภาพที่มีความเข้มน้อยกว่าความเข้มที่จุดศูนย์กลางที่กำลังพิจารณาอยู่มากกว่าหรือเท่ากับค่าขีดแบ่ง

จุดภาพศูนย์กลางใดที่มีจุดภาพที่อยู่บนวงกลมสว่างกว่าหรือมืดกว่าจุดศูนย์กลางเรียงต่อกันมากกว่า N จุด จะถูกพิจารณาเป็นจุดมุม ดังรูป



รูปที่ 1 จุดภาพที่ใช้พิจารณาความเป็นมุมของจุดภาพ

https://docs.opencv.org/master/df/d0c/tutorial_py_fast.html

จากรูปที่ 1 ความเป็นมุมของจุดภาพ p พิจารณาจากจุดภาพที่อยู่ล้อมรอบเป็นรัศมี $r = 3$ ซึ่งมีทั้งหมด 16 จุดภาพ จุดภาพที่เส้นประลากผ่านเป็นจุดภาพที่มีความเข้มแสงมากกว่าความเข้มที่จุดศูนย์กลางที่กำลังพิจารณาอยู่จึงพิจารณาว่าเป็นจุดสว่าง และเนื่องจากจุดสว่างเหล่านี้เรียงตัวติดต่อกัน 12 จุดภาพ หากตั้งค่าขีดแบ่งของจำนวนจุดภาพที่เรียงต่อกัน $N = 12$ จะได้ว่า จุดภาพ p นี้เป็นจุดภาพที่เป็นมุม

การหาจุดเด่นด้วยวิธีข้างต้น พบว่านอกจากจุดที่เป็นมุมแล้วตัวตรวจจับยังตรวจจับขอบที่ปรากฏในภาพอีกด้วย จุดเด่นแต่ละจุดจึงถูกคำนวณคะแนนความเป็นมุมซึ่งพิจารณาได้จาก

1. ผลตอบสนองแฮริส (Harris response)
2. ผลรวมค่าสมบูรณ์ของผลต่างความเข้มของจุดภาพ (Sum of absolute intensity differences

algorithm : SAD algorithm)

โดยจุดเด่นที่มีค่าคะแนนความเป็นมุมมากที่สุด 2^n จุดแรกจะถูกพิจารณาว่าเป็นจุดมุมจริงๆ การเรียงจุดเด่นตามค่าคะแนนความเป็นมุมสามารถทำได้หลายขั้นตอนวิธี [6], [7] โดยขั้นตอนวิธีที่นิยมเป็นไปตามตารางที่ 1

ตารางที่ 1 ความซับซ้อนและแพลตฟอร์มของขั้นตอนวิธีจัดเรียง

ขั้นตอนวิธีการ	ความซับซ้อน (Complexity)	แพลตฟอร์ม
Bubble Sort	$O(n^2)$	CPU, FPGA
Insertion Sort	$O(n^2)$	CPU, FPGA
Selection Sort	$O(n^2)$	CPU, FPGA
Quick Sort	$O(n \log(n))$	CPU, FPGA
Heap Sort	$O(n \log(n))$	CPU, FPGA
Merge Sort	$O(n \log(n))$	CPU, GPU, FPGA
Shell Sort	$O(n(\log(n))^2)$	CPU, FPGA
Tim Sort	$O(n \log(n))$	CPU, FPGA

จากตารางที่ 1 จะพบว่าขั้นตอนวิธีในการเรียงแต่ละประเภทนั้นมีความซับซ้อนที่ต่างกัน จำนวนข้อมูลที่นำมาเรียงจึงส่งผลต่อการเลือกขั้นตอนวิธีมาใช้ หากข้อมูลมีจำนวนมาก การใช้ขั้นตอนวิธีที่ง่าย เช่น Bubble Sort, Insertion Sort หรือ Selection Sort อาจไม่เหมาะสม เนื่องจากความซับซ้อนจะสูงขึ้นตามจำนวนข้อมูลจนทำให้ใช้เวลาในการประมวลผลนานกว่าขั้นตอนวิธีอื่นๆ

จากการหาจุดเด่นที่เป็นมุมด้วยวิธีข้างต้นเพื่อให้การตรวจจับมุมไม่ขึ้นอยู่กับขนาดของภาพ จึงมีการใช้ขั้นตอนวิธีพาสต์กับพีระมิดรูปภาพตรวจจับมุมในรูปภาพเดิมแต่มีการลดขนาดภาพลงเรื่อยๆ มุมบางมุมในภาพอาจโค้งมนเกินไปทำให้ไม่สามารถตรวจจับมุมได้ในภาพขนาดปกติ เมื่อทำการลดขนาด มุมโค้งมนนั้นจะมีลักษณะที่แหลมขึ้น และสามารถตรวจจับว่าเป็นมุมได้ในภาพขนาดที่เล็กลงมา หรือหากนำไปตรวจจับมุมของวัตถุที่มีการ

เคลื่อนที่เข้าใกล้หรือออกห่างกล้องรับภาพ ซึ่งจะทำให้ขนาดของมูมเดิมเกิดการเปลี่ยนแปลง การประยุกต์ใช้พีระมิดรูปภาพกับขั้นตอนวิธีการฟาสต์จะทำให้ยังสามารถตรวจจับมูมที่มีขนาดเปลี่ยนไปนี้เป็นมูมเดิมได้

2.1.2 การสร้างพีระมิดรูปภาพ

ใช้หลักการลดการสุ่มตัวอย่าง (down-sampling) เพื่อสร้างรูปภาพที่มีความละเอียดน้อยลง การลดการสุ่มตัวอย่างนี้ไวต่อสิ่งรบกวน จึงใช้ตัวกรองแบบเกาส์เซียน (Gaussian filter) เพื่อกรองสัญญาณรบกวนออกก่อน จึงค่อยลดการสุ่มตัวอย่าง เพื่อลดความซับซ้อน ตัวกรองแบบเกาส์เซียนสามารถประมาณได้ด้วยตัวกรองแบบทวินาม (binomial filter) [4, 8] ที่สามารถเปลี่ยนเป็นตัวดำเนินการตามบิตได้ง่าย (Bit-wise operation)

2.1.3 การขจัดค่าที่ไม่มากที่สุด (Non maxima suppression)

การประยุกต์ใช้ขั้นตอนวิธีการฟาสต์กับพีระมิดรูปภาพ อาจทำให้ตรวจจับจุดเด่นที่เป็นจุดเด่นเดียวกันได้หลายตำแหน่งที่ใกล้เคียงกัน จุดที่ถือเป็นตำแหน่งจุดมูมของรูปภาพต้องเป็นจุดที่ให้ค่าคะแนนความเป็นมูมสูงสุด เฉพาะที่ การใช้การขจัดค่าที่ไม่มากที่สุดจะทำให้จุดเด่นที่ได้จากขั้นตอนวิธีฟาสต์ที่มีตำแหน่งใกล้เคียงกันเหลือเป็นตำแหน่งตัวแทนแค่ตำแหน่งเดียว

2.1.4 การคำนวณทิศทางของจุดเด่น (Orientation Computing)

เพื่อให้การตรวจจับจุดเด่นและสร้างตัวบรรยายลักษณะ ไม่ขึ้นกับผลจากการหมุนรูปภาพ (rotation invariance) ทิศทางของมูมจะถูกคำนวณเพื่อใช้ประกอบการสร้างตัวบรรยายลักษณะ โดยพิจารณาจากผลรวมแบบถ่วงน้ำหนักของความเข้มจุดภาพที่อยู่โดยรอบจุดเด่น หรือ โมเมนต์ ดังสมการที่ 2.1

$$\begin{cases} m_x = \sum_{x=-r}^r xI(x, y) \\ m_y = \sum_{y=-r}^r yI(x, y) \end{cases} \quad (2.1)$$

เมื่อ $I(x, y)$ คือ ความเข้มแสงของจุดภาพในตำแหน่งรอบจุดเด่นในรัศมี r

m_x คือ โมเมนต์ของจุดภาพในแนวนอน

m_y คือ โมเมนต์ของจุดภาพในแนวตั้ง

ทิศทางของมูม หรือ มูมของจุดเด่น พิจารณาจาก

$$\theta = \arctan\left(\frac{m_y}{m_x}\right) \quad (2.2)$$

เมื่อ θ คือ มุมแสดงทิศทางของมูมที่เป็นจุดเด่น

2.1.5 อาร์บรีฟ (rBRIEF : rotated BRIEF descriptor)

การสร้างตัวบรรยายจุดเด่นด้วยขั้นตอนวิธีอาร์บรีฟ [4], [9] เป็นขั้นตอนวิธีที่ปรับปรุงมาจากขั้นตอนวิธีบรีฟ โดยเริ่มการพิจารณาทิศทางของมุมก่อนสร้างตัวบรรยาย กลุ่มของจุดภาพที่จะนำมาสร้างตัวบรรยายนั้นจะถูกหมุนกลับด้วยมุมขนาดเท่ากับทิศทางของจุดเด่นนั้นๆ และสร้างตัวบรรยายแบบไบนารีแต่ละบิตด้วยการเปรียบเทียบค่าความเข้มของคู่จุดภาพซึ่งอยู่โดยรอบจุดเด่น ทำให้การสร้างตัวบรรยายด้วยวิธีนี้ไวต่อสัญญาณรบกวน ก่อนนำมาเปรียบเทียบจึงต้องลดสัญญาณรบกวนก่อนด้วยตัวกรองแบบเกาส์เซียนแล้วเลือกคู่จุดภาพมาเปรียบเทียบกันดังสมการ 2.3

$$\tau(p; a, b) = \begin{cases} 1, & p(a) < p(b) \\ 0, & p(a) \geq p(b) \end{cases} \quad (2.3)$$

เมื่อ $\tau(p; a, b)$ คือ การทดสอบแบบไบนารีที่จุดภาพตำแหน่ง a และตำแหน่ง b

p คือ แผ่นภาพที่ทำการลดสัญญาณรบกวนแล้ว (patch)

$p(a), p(b)$ คือ ความเข้มของจุดภาพ ณ ตำแหน่ง a และ b

และตัวบรรยายแบบบรีฟขนาด n บิตของแต่ละแผ่นภาพ p สร้างจาก

$$f_n(p) = \sum_{i=1}^n 2^{i-1} \cdot \tau(p; a_i, b_i) \quad (2.4)$$

เมื่อ $f_n(p)$ คือ ตัวบรรยายจุดภาพแบบบรีฟขนาด n บิตของแต่ละแผ่นภาพ p

จุฬาลงกรณ์มหาวิทยาลัย

2.2 การประเมินคุณภาพ

ในการประเมินคุณภาพของตัวตรวจจับของขั้นตอนวิธีการที่นำเสนอ Farzin และคณะ [10] ได้เสนอตัวชี้วัด 2 ตัวชี้วัด ได้แก่ ความต้องกันของจำนวนจุดมุม (Consistency of Corner Numbers : CCN) และ ความแม่นยำ (Accuracy : ACU) ในการประเมินคุณภาพของตัวตรวจจับมุม

ความต้องกันของจำนวนจุดมุม หมายถึง จำนวนจุดมุมของภาพที่ตรวจจับได้จะต้องไม่ขึ้นกับสัญญาณรบกวน, การหมุนและการแปลงแบบสัมพรรค (affine transformation) โดยพิจารณาจากสมการ 2.5

$$CCN = 100 \times 1.1^{-|N_t - N_o|} \quad (2.5)$$

เมื่อ CCN คือ ค่าความต้องกันของจำนวนจุดมุม

N_o คือ จำนวนจุดมุมที่ตรวจจับได้จากภาพต้นฉบับ

N_t คือ จำนวนจุดมุมที่ตรวจจับได้จากภาพที่ผ่านการแปลงแบบต่างๆ

สำหรับความแม่นยำ พิจารณาจากสมการ 2.6

$$ACU = 100 \times \frac{\frac{N_m}{N_d} + \frac{N_m}{N_g}}{2} \quad (2.6)$$

เมื่อ ACU คือ ความแม่นยำ

N_d คือ จำนวนจุดมุ่มที่ตรวจจับได้โดยตัวตรวจจับมุ่มที่ต้องการทดสอบ

N_g คือ จำนวนจุดมุ่มจากข้อมูลที่เป็นจริง (ground truth)

N_m คือ จำนวนจุดมุ่มซึ่งตรวจจับได้จากตัวตรวจจับที่สอดคล้องกับข้อมูลที่เป็นจริง

การพิจารณาความแม่นยำด้วยสมการดังกล่าวจะครอบคลุมการพิจารณา ผลบวกเท็จ และ ผลลบเท็จ รวมอยู่ในตัวชี้วัดตัวเดียวกัน

ขั้นตอนวิธีโออาร์บีได้รับความนิยมในการนำมาประยุกต์ใช้เพื่อหาจุดเด่นที่เป็นมุ่มและสร้างตัวบรรยายแบบไบนารีโดยเฉพาะกับระบบที่ต้องการความเร็วในการประมวลผล เช่น การระบุตำแหน่งพร้อมกับการสร้างแผนที่สำหรับรถยนต์อัตโนมัติหรือหุ่นยนต์ มีการพัฒนาสถาปัตยกรรมและเทคนิคต่างๆเพื่อให้สามารถตรวจจับจุดเด่นและสร้างตัวบรรยายด้วยขั้นตอนวิธีแบบโออาร์บีและวิธีใกล้เคียงอย่างต่อเนื่อง W.Fang และคณะ [11] นำเสนอสถาปัตยกรรมสำหรับขั้นตอนวิธีโออาร์บีบนฮาร์ดแวร์ Altera Stratix V FPGA โดยสร้างพีระมิดรูปภาพด้วยอัตราส่วน 1.25 เพื่อให้สามารถออกแบบสถาปัตยกรรมที่มีตัวดำเนินการแบบบวกและการเลื่อนบิตเท่านั้นได้ และมีการใช้เทคนิคประสานผสมระหว่างการประมวลผลแบบลำดับและแบบขนาน (Serial-Parallel Hybrid Workflow) กล่าวคือ ในการตรวจจับจุดเด่นและสร้างตัวบรรยายซึ่งประมวลผลแบบลำดับของพีระมิดรูปภาพในแต่ละลำดับชั้นทำงานขนานกันไป ทำให้ลดเวลาในการประมวลผลโดยภาพรวมได้ สำหรับการสร้างพีระมิดรูปภาพ J. Weberruss และคณะ [12] ได้เสนอการสร้างลำดับชั้นย่อยระหว่างการลดการสุ่มตัวอย่าง และได้ทดลองเปรียบเทียบจำนวนระดับชั้นของพีระมิดรูปภาพกับความแม่นยำพบว่า การเพิ่มระดับชั้นย่อย 2 ชั้นระหว่างการลดการสุ่มตัวอย่างของพีระมิดรูปภาพช่วยเพิ่มความแม่นยำในการตรวจจับและจับคู่จุดเด่น

เพื่อลดทรัพยากรในการประมวลผล R. Sun และคณะ [13] ใช้เทคนิคลดความยาวคำ (word length) ที่ใช้แสดงค่าโมเมนต์ของจุดเด่นในขั้นตอนวิธีอาร์โอบี ด้วยการลดจำนวนบิตศูนย์ที่มีความสำคัญน้อยออก R. Liu และคณะ [14] แบ่งมุ่มในช่วง $[0, 2\pi)$ ออกเป็น 32 ส่วนเพื่อแสดงทิศทางของจุดมุ่ม การแบ่งมุ่มในลักษณะนี้ทำให้ระบบสามารถคำนวณแบบรูปที่หมุนแล้วของคู่จุดสนใจที่จะนำมาสร้างตัวบรรยายแบบบริฟ (rotated BRIEF pattern) ได้ ก่อนล่วงหน้า J. Huang และคณะ [15] นำเสนอสถาปัตยกรรมบนเอฟพีจีเอที่ช่วยลดพื้นที่เก็บข้อมูลของระบบ โดยให้องค์ประกอบที่จำเป็นต้องใช้ข้อมูลความเข้มแสงในแผ่นภาพเดียวกัน ทำงานพร้อมกัน ระบบจะได้ไม่ต้องเสียเวลาเข้าถึงหน่วยความจำซ้ำหลายครั้งซึ่งใช้เวลามากกว่ามาก

นอกจากฮาร์ดแวร์เอฟพีจีเอที่นำมาช่วยเร่งความเร็วในการตรวจจับจุดมุ่มและสร้างตัวบรรยายไบนารี ตัวประมวลผลสัญญาณดิจิทัล (Digital Signal Processor: DSP) ยังถูกนำมาใช้ในการประมวลผลขั้นตอนวิธีโออาร์บีใน

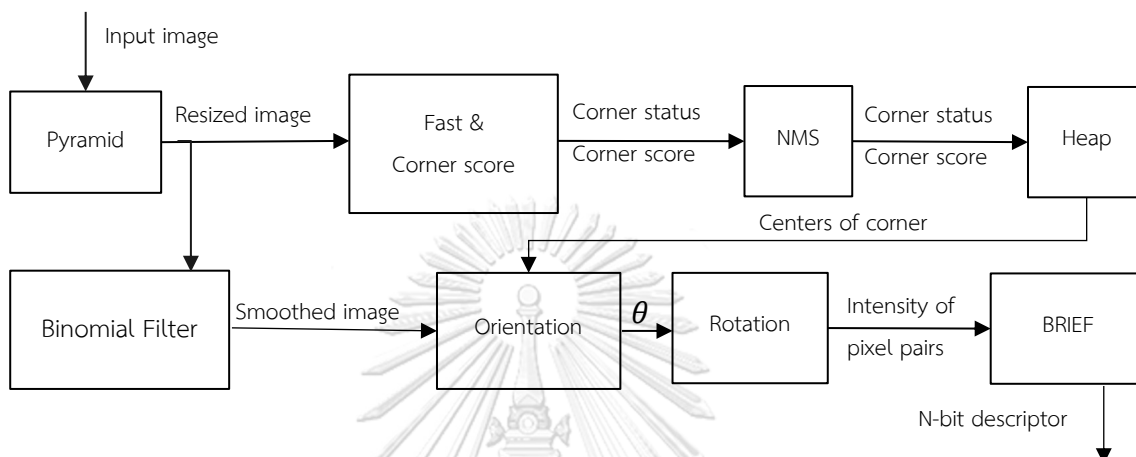
งานวิจัยของ G. Wang และคณะ [16] และเพิ่มขั้นตอนการผสมภาพด้วยเทคนิค α -blending หลังจากตรวจจับจุดเด่นและสร้างตัวบรรยาย โดยค่าความเข้มของจุดภาพที่จะนำมาผสมกันของแต่ละภาพ จะถูกถ่วงน้ำหนักตามระยะทางจากขอบจนถึงตำแหน่งของจุดนั้น เพื่อให้รูปภาพที่ได้จากการผสมที่กลมกลืนมากขึ้น

การตรวจจับจุดเด่นและการสร้างตัวบรรยายด้วยขั้นตอนวิธีโออาร์บีสามารถเพิ่มความเร็วได้ด้วยการออกแบบสถาปัตยกรรมที่เหมาะสมกับทรัพยากรและลักษณะการประมวลผลแบบขนาน เพื่อให้สามารถเพิ่มประสิทธิภาพในการตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยาย วิทยานิพนธ์นี้จะปรับปรุงขั้นตอนวิธีการและออกแบบสถาปัตยกรรมบนพื้นฐานของขั้นตอนวิธีโออาร์บีให้สอดคล้องกับการทำงานของฮาร์ดแวร์มากขึ้น ดังรายละเอียดในหัวข้อถัดไป



3. สถาปัตยกรรมที่ออกแบบ

ขั้นตอนวิธีสำหรับตรวจจับและบรรยายจุดเด่นของภาพบนพื้นฐานของขั้นตอนวิธีโออาร์พีถูกปรับปรุงให้เหมาะกับการนำไปใช้กับฮาร์ดแวร์ ลดความซับซ้อนในการประมวลผล โดยมีรายละเอียดโดยมีภาพรวมของระบบดังรูปที่ 2

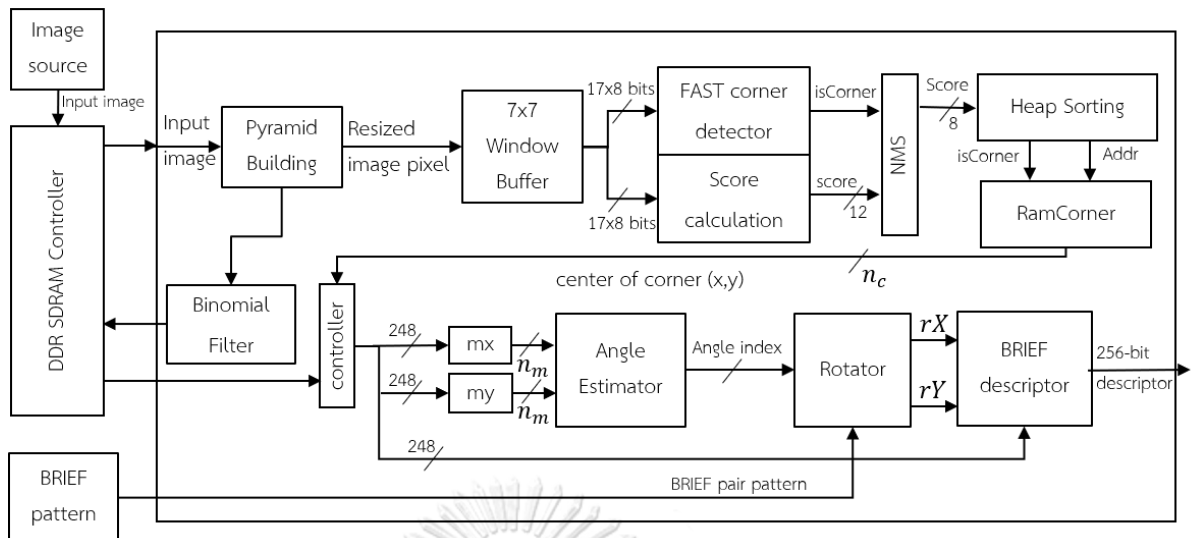


รูปที่ 2 ภาพรวมระบบการตรวจจับลักษณะและสร้างตัวบรรยายของภาพ

ระบบตรวจจับและสร้างตัวบรรยายของภาพมีองค์ประกอบหลัก 8 ส่วนหลัก ได้แก่

1. การสร้างพีระมิดรูปภาพ
2. การตรวจจับลักษณะภาพด้วยขั้นตอนวิธีฟาสต์และการคำนวณคะแนน
3. การขจัดจุดเด่นที่มีค่าคะแนนความเป็นมุมไม่มากที่สุด
4. ตัวจัดเรียงลำดับจุดมุมตามค่าความเป็นมุมด้วยขั้นตอนวิธีแบบฮีป
5. ตัวกรองแบบทวินาม
6. การประมาณทิศทางของจุดเด่น
7. การหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุม
8. การสร้างตัวบรรยายแบบบริฟ

เพื่อลดเวลาในการประมวลผลและปรับปรุงขั้นตอนวิธีการให้สามารถประมวลผลแบบขนานได้ วิทยานิพนธ์นี้ได้ออกแบบสถาปัตยกรรมสำหรับตรวจจับจุดเด่นที่เป็นมุมและสร้างตัวบรรยายแบบไบนารีที่มีภาพรวมของระบบเป็นดังรูปที่ 3



รูปที่ 3 สถาปัตยกรรมสำหรับระบบการตรวจจับจุดเด่นที่เป็นมุมและสร้างตัวบรรยายของภาพ

จากรูปที่ 3 ภาพต้นฉบับจะถูกเก็บในบัฟเฟอร์ของตัวสร้างพีระมิดรูปภาพและลดการสุ่มตัวอย่างแบบหลักและแบบย่อย ตามลำดับชั้น ภาพผลลัพธ์ที่ได้จากการสร้างพีระมิดภาพจะถูกส่งไปยังวินโดว์บัฟเฟอร์ขนาด 7x7 จุดภาพ เพื่อส่งไปตรวจจับจุดมุมด้วยตัวตรวจจับแบบฟาสต์และคำนวณคะแนนความเป็นมุม ในขณะเดียวกันผลลัพธ์จากการสร้างพีระมิดจะส่งไปยังตัวกรองแบบทวินามเพื่อสร้างภาพลดทอนสัญญาณรบกวน และไปเป็นภาพต้นแบบในลำดับชั้นถัดไป ทำงานขนานกัน

จุดภาพที่จัดเก็บในวินโดว์บัฟเฟอร์ขนาด 7x7 จะถูกใช้ในการพิจารณาความเป็นมุมของจุดภาพและคำนวณค่าคะแนนความเป็นมุมไปพร้อมกัน ค่าความจริงของความเป็นมุมและค่าคะแนนความเป็นมุมจะเข้าสู่ตัวจัดเรียงแบบฮีปเพื่อเรียงลำดับเลือกเฉพาะ N จุดแรกที่มีค่าคะแนนความเป็นมุมสูงสุด ข้อมูลตำแหน่งของจุดภาพที่เป็นจุดเด่นจะถูกส่งไปยังตัวควบคุมหน่วยความจำที่เก็บภาพจากตัวกรองแบบทวินาม ตัวควบคุมจะเข้าถึงกลุ่มของจุดภาพที่มีตำแหน่งจุดศูนย์กลางตามพิกัดที่ตัวจัดเรียงแบบฮีปส่งมาและส่งข้อมูลแต่ละแถวซึ่งมีขนาด 31 จุดภาพไปเก็บยังวินโดว์บัฟเฟอร์ขนาด 31x31 จุดภาพ เพื่อนำไปคำนวณค่าโมเมนต์ของจุดมุมในแนวแกนนอนและแนวแกนตั้งตามลำดับ

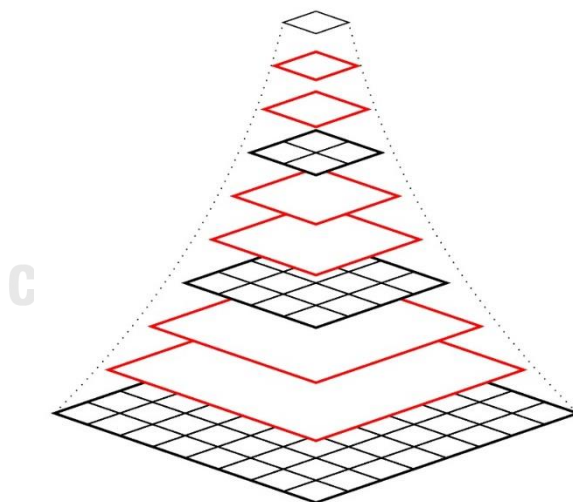
โมเมนต์ของจุดภาพขนาด n_m บิตจะเข้าสู่ขั้นตอนประมาณทิศทางของจุดมุมเพื่อหาช่วงมุมที่ใกล้เคียงที่สุดที่ทำให้ผลคูณระหว่างโมเมนต์ในแนวแกนนอนกับค่าฟังก์ชันแทนเจนต์ของมุนั้นใกล้เคียงกับค่าโมเมนต์ในแนวแกนตั้งมากที่สุด ส่งไปยังตัวหมุนแผ่นภาพเพื่อใช้หมุนรูปแบบคู่จุดสำหรับสร้างตัวบรรยายแบบบริฟ (BRIEF pair pattern) คู่จุดแต่ละคู่จะสร้างเป็นตัวบรรยายแต่ละบิตนำมาเรียงต่อกันและส่งออกเป็นผลลัพธ์ตัวบรรยายขนาด N บิต

ในแต่ละชั้นของการประมวลผลจะประกอบด้วยองค์ประกอบที่ทำงานเหมือนกัน แต่ต่างเพียงภาพต้นแบบที่ได้รับในแต่ละชั้นเท่านั้น โดยขั้นตอนวิธีการตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยายไบนารีบนพื้นฐานขั้นตอนวิธีโออาร์บีที่ปรับปรุงใหม่และสถาปัตยกรรมสำหรับแต่ละองค์ประกอบมีรายละเอียดดังต่อไปนี้

3.1 การสร้างพีระมิดรูปภาพ

การใช้พีระมิดรูปภาพช่วยให้สามารถตรวจจับลักษณะเฉพาะเดียวกันแต่มีขนาดต่างกันได้ การสร้างพีระมิดภาพจะลดขนาดของรูปภาพในแต่ละชั้นเรื่อยๆ โดยพีระมิดในลำดับสูงกว่าจะมีขนาดเล็กกว่าลำดับล่างกว่า ในการลดการสุมตัวอย่างจะทำให้ภาพเรียบ (smoothing) ไปพร้อมกัน เพื่อลดขั้นตอนในการประมวลผลและลดขนาดของหน่วยความจำที่ใช้เก็บรูปที่เกิดจากการทำให้ภาพเรียบ โดยใช้แนวคิดของการประมาณค่าในช่วงเชิงเส้นคู่ (bilinear interpolation) โดยทั่วไปการสร้างพีระมิดรูปภาพ จะลดขนาดลงลำดับชั้นละ 2 เท่าจากภาพเดิม เพื่อไม่ให้ลักษณะเฉพาะที่เป็นมุมบางมุมหายไปจากการลดการสุมตัวอย่างที่รวดเร็วเกินไป [12] จะมีการลดการสุมตัวอย่างในอัตราส่วน 1.25 เท่าของภาพเดิมด้วย ซึ่งใกล้เคียงกับค่าเริ่มต้น 1.2 ของฟังก์ชัน ORB ใน openCV library

ในวิทยานิพนธ์นี้ การลดการสุมตัวอย่างจะถูกแบ่งเป็น 2 แบบ คือ การลดการสุมตัวอย่างด้วยอัตราส่วน 2 เท่า เรียก การลดการสุมตัวอย่างหลัก และ การลดการสุมตัวอย่างในอัตราส่วน 1.25 เท่า เรียก การลดการสุมตัวอย่างย่อย ดังแสดงในรูปที่ 4



รูปที่ 4 พีระมิดรูปภาพที่ประกอบด้วย การลดการสุมตัวอย่างแบบหลักและแบบย่อย
(แผ่นสีดำแทนแบบหลัก และ แผ่นสีแดงแทนแบบย่อย)

ข้อมูลจุดภาพแบบระดับสีเทา (gray scale) ที่หน่วยรับภาพนำมาเก็บหน่วยความจำ DDR3 จะถูกส่งให้มอดูลพีระมิดรูปภาพทีละ 1 จุดภาพขนาด 8 บิต โดยไล่ไปตามแถวในแนวนอน ทำให้ในการสุมตัวอย่างจะไล่ทำทีละแกนภาพโดยเริ่มจากแกนอนก่อน จากนั้นนำผลลัพธ์จากแนวนอนมาเป็นภาพต้นแบบในการทำแนวตั้งต่อไป การ

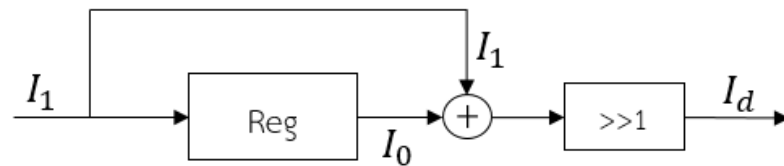
ลดการสุมตัวอย่างเพื่อสร้างพีระมิดรูปภาพนี้จะใช้ผลลัพธ์จากการสุมตัวอย่างในระดับชั้นก่อนหน้า ยกเว้นการลดการสุมตัวอย่างหลักที่ใช้ผลลัพธ์จากการสุมตัวอย่างหลักระดับก่อนหน้าเท่านั้น สำหรับการลดการสุมตัวอย่างแบบหลักในวิทยานิพนธ์นี้ เลือกใช้เฉพาะการดำเนินการบวก และ การเลื่อนบิตดังนี้

$$I_d = (I_0 + I_1) \gg 1 \quad (3.1)$$

เมื่อ I_d คือ ความเข้มของจุดภาพที่ได้หลังการลดการสุมตัวอย่าง

I_0, I_1 คือ ความเข้มของจุดภาพจากภาพต้นทาง

ซึ่งสามารถออกแบบสถาปัตยกรรมภายในได้เป็นดังรูป



รูปที่ 5 สถาปัตยกรรมของการลดการสุมตัวอย่างแบบหลัก

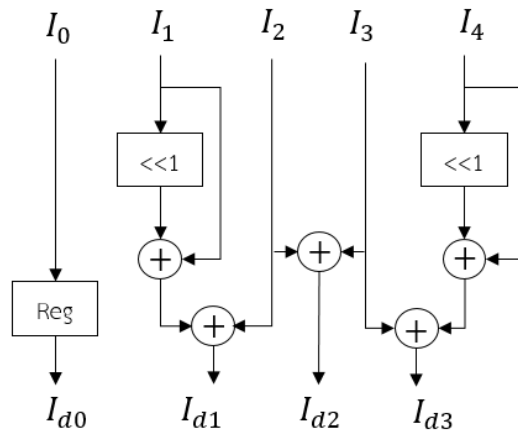
สำหรับการลดการสุมตัวอย่างแบบย่อยในวิทยานิพนธ์นี้ได้เลือกใช้เฉพาะการดำเนินการบวกและเลื่อนบิตได้ดังนี้

$$\begin{cases} I_{d0} = I_0 \\ I_{d1} = ((I_1 \ll 1) + I_1 + I_2) \gg 2 \\ I_{d2} = (I_2 + I_3) \gg 1 \\ I_{d3} = (I_3 + (I_4 \ll 1) + I_4) \gg 2 \end{cases} \quad (3.2)$$

เมื่อ $I_{d0}, I_{d1}, I_{d2}, I_{d3}$ คือ ความเข้มของจุดภาพที่ได้หลังการลดการสุมตัวอย่าง

I_0, I_1, I_2, I_3, I_4 คือ ความเข้มของจุดภาพจากภาพต้นทาง

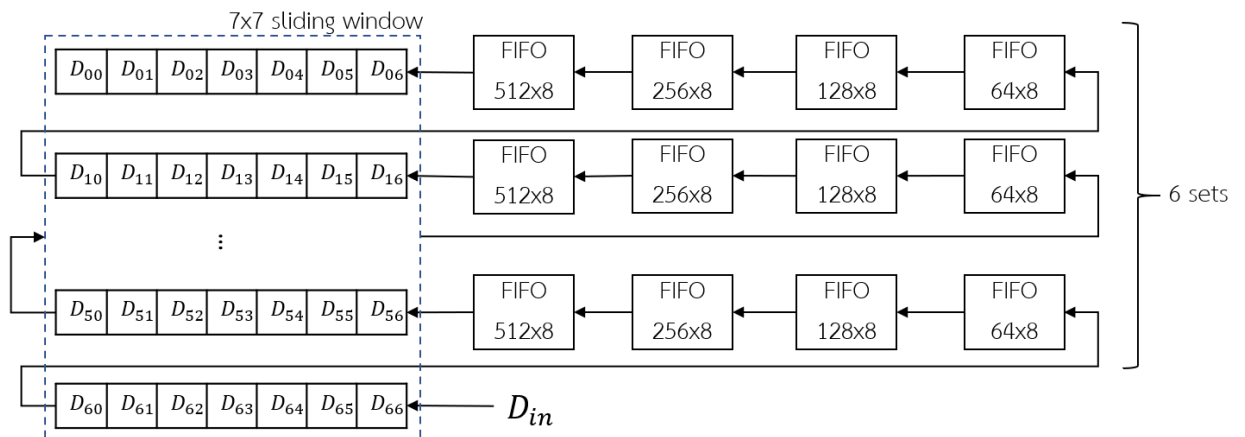
สามารถออกแบบสถาปัตยกรรมภายในได้ดังรูป



รูปที่ 6 สถาปัตยกรรมของการลดการคูณตัวอย่างแบบย่อ

3.2 วินโดว์บัฟเฟอร์ขนาด 7x7

จุดภาพผลลัพธ์หลังการคูณตัวอย่างจะถูกส่งไปยังมอดูลวินโดว์บัฟเฟอร์ขนาด 7x7 ซึ่งประกอบด้วย หน่วยความจำแบบ เรจิสเตอร์ (register) ขนาด 7x8 บิต, ไฟโฟ (FIFO : First-In-First-Out) ขนาด 512x8 บิต, 256x8 บิต, 128x8 บิต และ 64x8 บิต ต่ออนุกรมกัน ตามลำดับ จำนวน 6 ชุด และต่อท้ายด้วยเรจิสเตอร์ ขนาด 7x8 บิตอีกครั้ง ดังรูปที่ 7

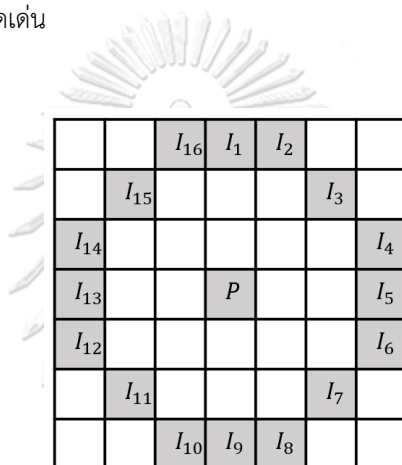


รูปที่ 7 สถาปัตยกรรมสำหรับวินโดว์บัฟเฟอร์ขนาด 7x7

จะเห็นว่าข้อมูลจุดภาพจะถูกส่งต่อแบบเรียงลำดับไปในหน่วยความจำที่ต่ออนุกรมกัน โดยมีส่วนสัญญาณควบคุมให้เก็บข้อมูลในไฟโฟตัวก่อนหน้าให้เต็มก่อน จากนั้นค่อยสร้างสัญญาณอ่านข้อมูลเพื่อเขียนไปยังไฟโฟตัวถัดไป ยกเว้นไฟโฟตัวที่ติดกับเรจิสเตอร์ที่จะไม่เก็บข้อมูลจนเต็ม แต่จะกันข้อมูลจำนวน 7 จุดภาพให้เก็บในหน่วยความจำเรจิสเตอร์แทน ข้อมูลในเรจิสเตอร์ทั้งหมดคือจุดภาพที่อยู่ในแผ่นภาพขนาด 7x7 ซึ่งมอดูลวินโดว์บัฟเฟอร์ขนาด 7x7 จะอ่านค่าและส่งต่อไปยังมอดูลตรวจจับลักษณะภาพแบบฟาสต์และมอดูลคำนวณคะแนนต่อไป

3.3 การตรวจจับลักษณะภาพด้วยขั้นตอนวิธีฟาสต์และการคำนวณคะแนน

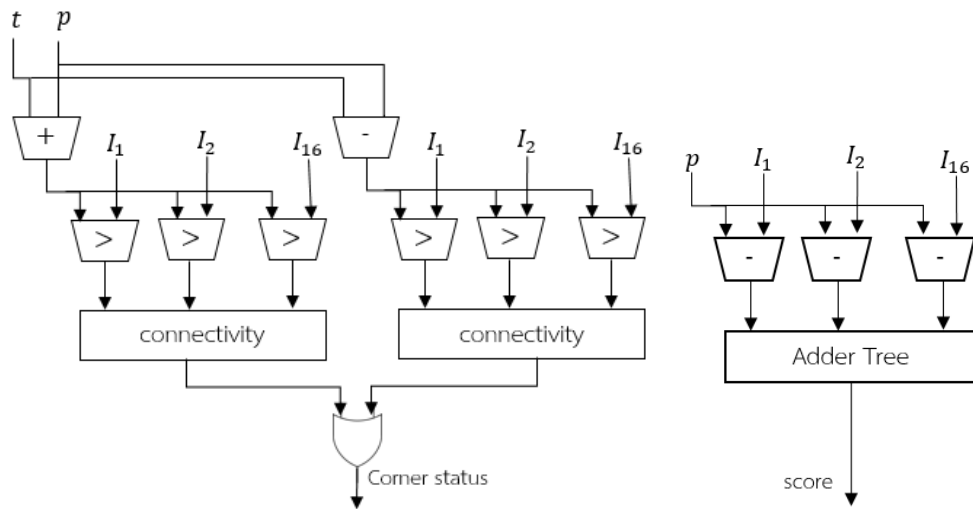
ในวิทยานิพนธ์นี้เลือกทดสอบความเป็นมุมด้วยขั้นตอนวิธีฟาสต์และคำนวณคะแนนความเป็นมุมจากผลรวมค่าสมบรูณ์ของผลต่างความเข้มของจุดภาพ ทดแทนการใช้ผลตอบสนองแฮริส เนื่องจากมีขั้นตอนวิธีคำนวณที่ซับซ้อนน้อยกว่ามาก โดยเลือกทดสอบจุดที่คาดว่าจะจุดเด่นด้วยจุดภาพที่อยู่รอบจุดนั้นเป็นรัศมี 3 จุดภาพ จุดภาพของภาพต้นแบบที่ถูกบรรจุในเรจิสเตอร์จากมอดูลวินโดว์บัพเฟอร์ตั้งรูปที่ 7 ตำแหน่ง p เป็นตำแหน่งของจุดภาพที่กำลังพิจารณาและจุดภาพทดสอบจะอยู่ในแนวเส้นรอบวงมีทั้งหมด 16 จุดภาพ จุดทดสอบทั้ง 17 จุดภาพจะถูกส่งมาจากมอดูลวินโดว์บัพเฟอร์ 7×7 มาแบ่งเป็น 3 ประเภทตามค่าความเข้มของจุดภาพนั้นๆ สำหรับวิทยานิพนธ์นี้ หากจุดภาพที่คาดว่าจะจุดเด่นมีจุดทดสอบที่อยู่โดยรอบเป็นจุดประเภทสว่างกว่าหรือมืดกว่า 12 จุดติดต่อกัน จะถือว่าจุดนั้นเป็นจุดเด่น



รูปที่ 7 ตำแหน่งจุดภาพที่ใช้ทดสอบสำหรับขั้นตอนวิธีแบบฟาสต์

การพิจารณาประเภทของจุดภาพต้องอาศัยค่าขีดแบ่งในการจำแนก เพื่อลดการสร้างวงจรวกหรือลบจากนำค่าความเข้มของจุดภาพทดสอบ 16 จุดมาบวกหรือลบกับค่าขีดแบ่งเพื่อเทียบกับจุดภาพ p ที่กำลังพิจารณา วิทยานิพนธ์นี้จึงนำค่าความเข้มของจุด p มาบวก-ลบกับค่าขีดแบ่งเพียงครั้งเดียวแทน และนำผลลัพธ์นั้นไปเปรียบเทียบกับค่าความเข้มของทั้ง 16 จุดภาพแทน ดังรูปที่ 8

เนื่องจากการพิจารณามุมด้วยวิธีนี้มีโอกาสที่ตัวตรวจจับจะตรวจจับมุมเดียวกันแต่อยู่ในหลายตำแหน่งใกล้กันได้ จึงต้องมีการคำนวณคะแนนค่าความเป็นมุมของแต่ละจุดเด่นที่ได้จากขั้นตอนวิธีฟาสต์ โดยการพิจารณาจากผลรวมค่าสมบรูณ์ของผลต่างความเข้มของจุดภาพที่คาดว่าจะจุดเด่นกับจุดภาพที่อยู่โดยรอบ โดยเลือกรัศมีที่พิจารณาเป็น 3 จุดภาพเช่นเดียวกับที่ใช้ในขั้นตอนวิธีฟาสต์ ดังนั้น เพื่อลดจำนวนครั้งในการเข้าถึงหน่วยความจำและประหยัดเวลาในการประมวลผล ในวิทยานิพนธ์นี้จะทำการคำนวณคะแนนความเป็นมุมของแต่ละจุดที่กำลังพิจารณาไปพร้อมกับการพิจารณาการเป็นจุดเด่นด้วยขั้นตอนวิธีฟาสต์ และให้ผลลัพธ์สถานะหรือค่าความจริงของการเป็นจุดมุมของจุดที่คาดว่าจะจุดเด่นพร้อมกับคะแนนความเป็นมุม ดังรูปที่ 8

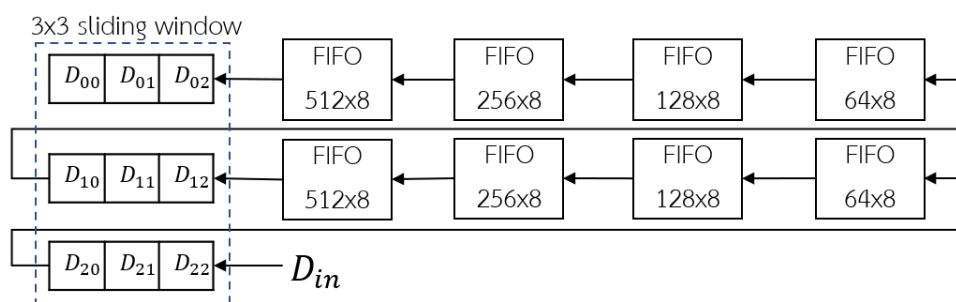


รูปที่ 8 สถาปัตยกรรมสำหรับขั้นตอนวิธีฟาสต์และตัวคำนวณคะแนนความเป็นมุม

3.4 การขจัดจุดเด่นที่มีค่าคะแนนความเป็นมุมไม่มากที่สุด

จุดที่คาดว่าเป็นจุดเด่นจุดหนึ่งมีโอกาสตรวจจับได้ในตำแหน่งที่ต่างกันแต่อยู่บริเวณใกล้เคียงกัน เพื่อลดความซ้ำซ้อนของจุดเด่น จุดที่คาดว่าจะจะเป็นจุดเด่นที่อยู่ใกล้กันระยะ 3x3 จุดภาพ จุดภาพเดียวเท่านั้นที่คะแนนความเป็นมุมสูงสุดจะถูกพิจารณาเป็นจุดภาพตัวแทนของจุดภาพกลุ่มนั้น

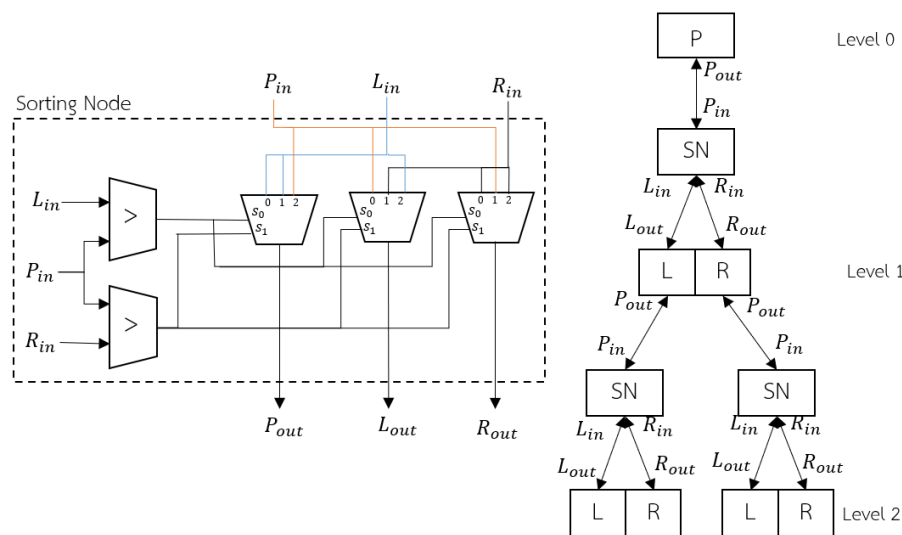
มอดูลขจัดจุดที่มีค่าความเป็นมุมไม่มากที่สุดในวิทยานิพนธ์นี้มีองค์ประกอบคล้ายกับมอดูลวินโดว์บัฟเฟอร์ขนาด 7x7 กล่าวคือ ประกอบด้วยเรจิสเตอร์ขนาด 3x8 บิต, ไฟโฟขนาด 512x8 บิต, 256x8 บิต, 128x8 บิต และ 64x8 บิต ต่ออนุกรมกัน ตามลำดับ จำนวน 2 ชุด และต่อท้ายด้วยเรจิสเตอร์ขนาด 3x8 บิตอีกครั้ง เพื่อให้สามารถรับค่าคะแนนที่มารั้งละ 1 จุดภาพมาเรียงกันให้สามารถพิจารณาครั้งละ 3x3 จุดภาพได้ ดังรูปที่ 9 และเพิ่มส่วนคำนวณค่าคะแนนโดยเปรียบเทียบคะแนนความเป็นมุมของจุดภาพศูนย์กลางคู่กับอีก 8 จุดภาพโดยรอบ หากจุดภาพศูนย์กลางมีค่าคะแนนความเป็นมุมมากกว่าทุกจุดโดยรอบ ค่าคะแนนความเป็นมุมจะคงเดิม แต่มีจุดโดยรอบมีคะแนนความเป็นมุมมากกว่าจุดภาพศูนย์กลางอย่างน้อย 1 จุด คะแนนความเป็นมุมของจุดศูนย์กลางนั้นจะถูกปรับเป็นศูนย์โดยทันที เช่นเดียวกับจุดภาพที่ไม่ใช่จุดมุมก็จะมีคะแนนความเป็นมุมเป็นศูนย์



รูปที่ 9 ไลน์บัฟเฟอร์สำหรับการขจัดจุดเด่นที่มีค่าคะแนนความเป็นมุมไม่มากที่สุด

3.4 ตัวจัดเรียงลำดับจุดมุมตามค่าความเป็นมุมด้วยขั้นตอนวิธีแบบฮีป

เนื่องจากการตรวจจับจุดเด่นด้วยขั้นตอนวิธีฟาสต์ มักตรวจจับเจอเส้นขอบด้วย วิทยานิพนธ์นี้จึงเลือกนำจุดที่คาดว่าเป็นจุดเด่นที่มีค่าคะแนนความเป็นมุมสูงสุด N จุดเด่นแรกเท่านั้น มาดำเนินการหาค่าทิศทางและหาตัวบรรยายลักษณะต่อไป จุดที่คาดว่าเป็นจุดเด่นแต่ละจุดจะถูกนำมาเรียงตามคะแนนความเป็นมุมเป็นลำดับไม่เพิ่มด้วยขั้นตอนวิธีแบบฮีป ในแต่ละรอบของการเรียงลำดับ จะมีการนำจุดเด่นที่มีค่าคะแนนความเป็นมุมน้อยสุดออกและรับจุดเด่นใหม่เข้ามา ซึ่งทำให้สามารถกำหนดจำนวนจุดเด่นที่ต้องการเก็บไว้เพื่อนำไปคำนวณทิศทางและตัวบรรยายต่อได้จากจำนวนระดับของต้นไม้ สถาปัตยกรรมสำหรับจัดเรียงลำดับจุดมุมตามค่าความเป็นมุมด้วยขั้นตอนวิธีแบบฮีปสามารถออกแบบได้ดังรูปที่ 10



รูปที่ 10 สถาปัตยกรรมสำหรับขั้นตอนวิธีเรียงลำดับแบบฮีป

มอดูลจัดเรียงแบบฮีปประกอบด้วยส่วนจัดเรียง (Sorting node) และส่วนจัดข้อมูลซึ่งเก็บค่าคะแนนและตำแหน่งจุดภาพ ดังรูปที่ 10 เมื่อเริ่มได้รับข้อมูลคะแนนความเป็นมุมที่ผ่านมอดูลจัดค่าที่ไม่มากที่สุดมาครั้งละจุดภาพ ตัวนับที่อยู่ภายในมอดูลจัดเรียงแบบฮีปจะนับขึ้นครั้งละ 1 เพื่อใช้เป็นค่าบอกตำแหน่งของจุดภาพที่กำลังพิจารณาอยู่และสร้างเป็นสัญญาณใหม่ซึ่งมีคะแนนความเป็นมุมอยู่ใน 8 บิตแรกและตำแหน่งจากตัวนับอยู่ใน 20 บิตหลัง สัญญาณที่สร้างขึ้นใหม่นี้จะถูกสับเปลี่ยนไปเก็บในส่วนเก็บข้อมูล โดยมีส่วนจัดเรียงเป็นตัวควบคุม ตัวจัดเรียงจะจัดเรียงให้ส่วนเก็บข้อมูลที่อยู่ด้านบนมีค่าคะแนนความเป็นมุมน้อยสุด และข้อมูลในส่วนเก็บข้อมูลด้านล่างมีค่าความเป็นมุมน้อยกว่าข้อมูลในส่วนเก็บข้อมูลด้านขวา และหากส่วนเก็บข้อมูลด้านล่างซ้ายและด้านล่างขวาของระดับชั้นต้นไม้นั้นมีระดับชั้นต้นไม้อื่นต่อลงมา ส่วนเก็บข้อมูลด้านล่างซ้ายและด้านล่างขวาก็จะทำหน้าที่เป็นส่วนเก็บข้อมูลด้านบนของระดับชั้นต่อไป

ในแต่ละครั้งที่ข้อมูลคะแนนความเป็นมุมของจุดภาพใหม่เข้ามา หากเป็นจุดมุม มอดูลจัดเรียงแบบฮีปจะส่งสัญญาณเข้าไปเขียนแรมเก็บข้อมูลจุดมุม (RamCorner) ให้ค่าในแรม ณ แอดเดรส (Address) เดียวกับตำแหน่งที่ได้จากตัวนับ มีค่าเป็น 1 นั้นแสดงว่ามีสถานะเป็นจุดมุมโดยปริยายก่อน เมื่อมีการจัดเรียงจนได้จุดมุมที่ต้องนำออก เนื่องจากมีค่าความเป็นมุมน้อยสุดไม่สอดคล้องกับเงื่อนไขของส่วนจัดเรียง มอดูลจัดเรียงแบบฮีปจะอ่านค่าตำแหน่งที่อยู่ซึ่งอยู่ที่ 20 บิตหลังของข้อมูลและส่งสัญญาณเข้าไปเขียนแรมเก็บข้อมูลจุดมุมให้ค่าในแรม ณ แอดเดรสนั้น เปลี่ยนค่าเป็น 0 แสดงถึงสถานะไม่ใช่จุดมุมที่จะพิจารณาสำหรับภาพนี้

เมื่อมอดูลได้รับข้อมูลคะแนนความเป็นมุมครบทุกจุดภาพและการจัดเรียงเสร็จสิ้นส่งสัญญาณ heapReady = '1' เพื่อให้ตัวควบคุมหน่วยความจำที่เก็บภาพจากตัวกรองแบบทวินามเริ่มประมวลผล

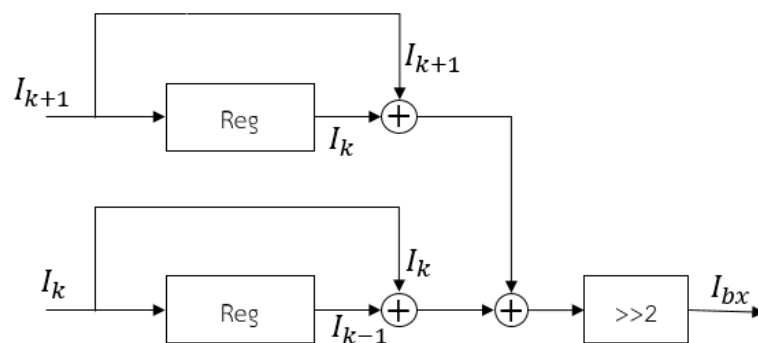
3.6 ตัวกรองแบบทวินาม

ตัวกรองแบบทวินามถูกนำมาใช้ในการลดสัญญาณรบกวนในภาพ ทดแทนการใช้ตัวกรองแบบเกาส์เซียน เนื่องจากสามารถออกแบบให้ใช้เพียงตัวดำเนินการแบบบวกและการเลื่อนบิตเท่านั้น โดยค่าความเข้มของภาพหลังการลดสัญญาณรบกวนพิจารณาได้ดังสมการ 3.3

$$I_{bk} = [(I_{k-1} + I_k) + (I_k + I_{k+1})] \gg 2 \quad (3.3)$$

เมื่อ I_{bk} คือ ความเข้มของจุดภาพหลังผ่านตัวกรองทวินามตำแหน่ง k
 I_k คือ ความเข้มของจุดภาพของภาพต้นแบบตำแหน่ง k

และสามารถออกแบบสถาปัตยกรรมได้ดังรูป



รูปที่ 11 สถาปัตยกรรมของตัวกรองทวินาม

โดยจะใช้ตัวกรองแบบทวินามนี้ทีละแนวแกนของภาพ โดยเริ่มจากแนวนอนไล่ไปตามแต่ละแถวก่อน และใช้ตัวกรองทวินามนี้กับผลลัพธ์ที่ได้ในแนวตั้ง ผลลัพธ์สุดท้ายของตัวกรองนี้จะถูกส่งไปเก็บในหน่วยความจำ DDR3 สำหรับใช้เรียกเมื่อต้องการค่าจุดภาพสำหรับคำนวณทิศทางของจุดมุมและสร้างตัวบรรยายแบบไบนารีต่อไป

3.7 การส่งข้อมูลแผ่นภาพขนาด 31x31 จุดภาพ

เมื่อคัดลอกจุดมุมที่ต้องการพิจารณาซึ่งเป็นผลลัพธ์จากมอดูลจัดเรียงแบบฮีป ตัวควบคุมหน่วยความจำที่ได้จากตัวกรองทวินามจะไล่อ่านค่าข้อมูลในแต่ละแอดเดรสของแรมเก็บข้อมูลจุดมุม (RamCorner) หากที่แอดเดรสใดมีข้อมูลเป็น 1 แสดงว่าต้องพิจารณาเป็นจุดศูนย์กลางของแผ่นภาพขนาด 31x31 จุดภาพ ตัวควบคุมจะสื่อสารกับ DDR3 ด้วยโปรโทคอล Avalon เพื่อเรียกข้อมูลจุดภาพที่อยู่ภายในแผ่นภาพขนาด 31x31 จุดภาพ และจัดเรียงค่าความเข้มของจุดภาพที่ผ่านตัวกรองแบบทวินามแล้ว เป็นแถว แถวละ 31 จุดภาพ ส่งต่อไปยังมอดูลคำนวณโมเมนต์เพื่อประมาณทิศทางของจุดมุม และส่งไปยังมอดูลสร้างตัวบรรยายแบบบริฟเพื่อใช้สร้างตัวบรรยายแบบไบนารีต่อไป

3.8 การคำนวณโมเมนต์ของจุดมุม

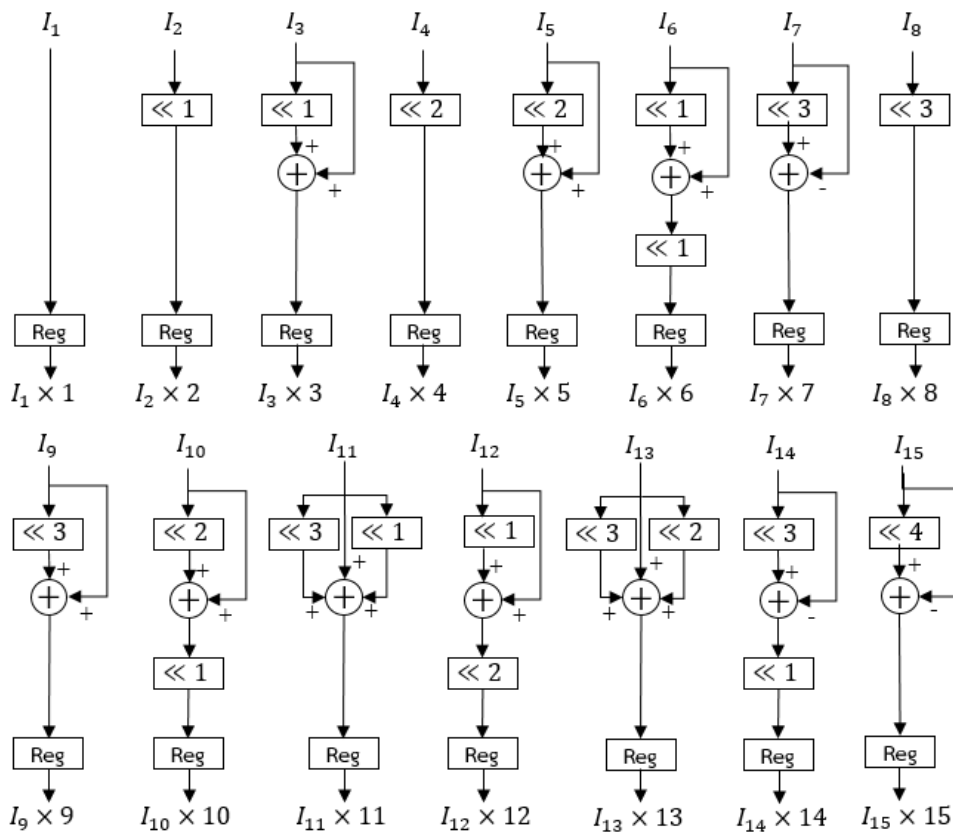
จุดเด่นที่มีค่าคะแนนความเป็นมุมมากที่สุด N อันดับแรก จะถูกคำนวณทิศทางของมุมเพื่อใช้หมุนกลุ่มของจุดภาพรอบจุดเด่นนั้น ก่อนการสร้างตัวบรรยายแบบบริฟ โดยทิศทางของมุมจะพิจารณาจากโมเมนต์รอบจุดเด่นในรัศมี r ในแต่ละแกนรอบจุดเด่น [4] เป็นไปตามสมการ 3.4

$$\begin{cases} m_x = \sum_{x=-r}^r xI(x, y) \\ m_y = \sum_{y=-r}^r yI(x, y) \end{cases} \quad (3.4)$$

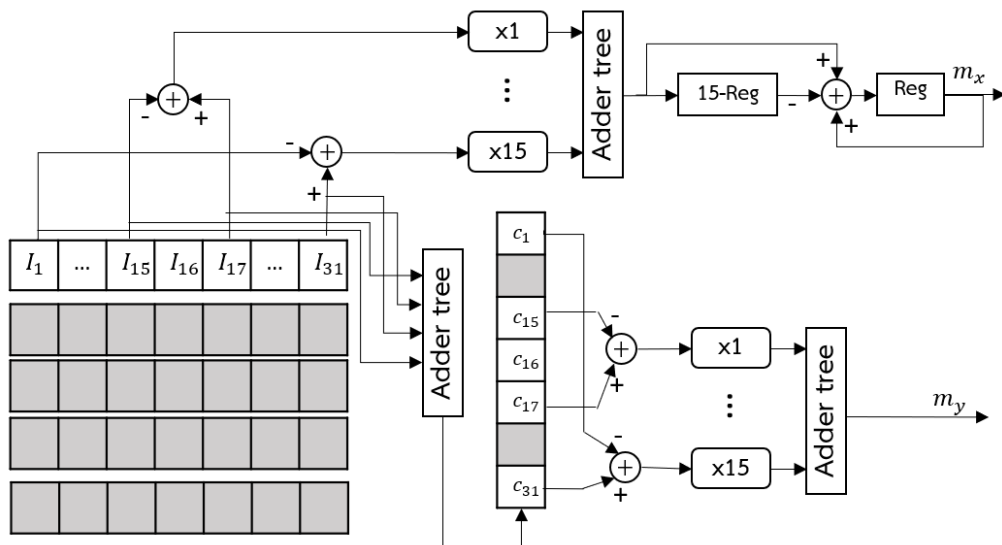
เมื่อ m_x, m_y คือ โมเมนต์ของของจุดภาพในแนวนอนและแนวตั้ง ตามลำดับ

$I(x, y)$ คือ ความเข้มของจุดภาพ

สถาปัตยกรรมสำหรับหาทิศทางของจุดเด่นจะดำเนินการต่อหลังจากทราบจุดเด่นทั้งหมดที่สอดคล้องเงื่อนไขจากจากมอดูลก่อนหน้า โดยจุดภาพที่นำมาคำนวณนั้นจะเป็นจุดภาพที่ได้จากตัวกรองแบบทวินาม ซึ่งจะค่อยๆไล่เข้ามาที่ละแถวของแผ่นภาพ ตัวคำนวณโมเมนต์ในแนวนอนจะนำค่าความเข้มของจุดภาพที่ได้มาคูณกับค่าประจำตำแหน่ง ซึ่งอยู่ในช่วง $-r$ ถึง r โดยวงจรสำหรับคูณค่าความเข้มแสงกับค่าประจำหลักเป็นไปดังรูปที่ 12 จากนั้นผลลัพธ์ที่ได้จะเข้าวงจรบวกแบบต้นไม้ (Adder tree) เพื่อรวมผลลัพธ์และไปรวมกับผลลัพธ์ของแถวก่อนหน้า และสำหรับตัวคำนวณโมเมนต์ในแนวตั้ง เนื่องจากแต่ละแถวของแผ่นภาพที่เข้ามาจะถูกคูณด้วยค่าประจำตำแหน่งค่าเดียวกันและข้อมูลที่เข้ามาที่ละแถว ระบบจึงรวมค่าความเข้มแสงของตัววงจรบวกแบบต้นไม้ก่อน จากนั้นนำไปเก็บในบัฟเฟอร์เพื่อเตรียมคูณค่าประจำหลักของแต่ละตำแหน่ง และรวมกันอีกครั้งด้วยวงจรบวกแบบต้นไม้ สถาปัตยกรรมสำหรับคำนวณโมเมนต์ในแนวนอนและแนวตั้งเป็นไปตามรูปที่ 13



รูปที่ 12 วงจรคูณค่าความเข้มของจุดภาพกับค่าประจำหลัก 1 ถึง 15



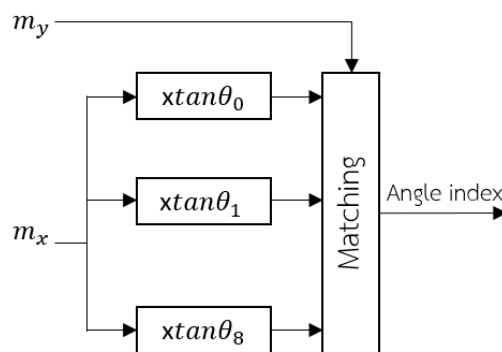
รูปที่ 13 วงจรคำนวณค่าโมเมนต์ของจุดเด่น

3.9 การประมาณทิศทางของจุดเด่น

โมเมนต์ในแนวแกนตั้งและแนวแกนนอนจะถูกนำมาใช้ในการประมาณช่วงของมุมแสดงทิศทางของจุดมุนั้น โดยพิจารณาจาก

$$\theta = \arctan\left(\frac{m_y}{m_x}\right) \quad (3.5)$$

เพื่อหลีกเลี่ยงความซับซ้อนในการคำนวณค่าฟังก์ชันตรีโกณมิติผกผัน ในวิทยานิพนธ์นี้เลือกใช้ตารางเชื่อมโยงข้อมูล (lookup table) และแบ่งมุมใน 1 จุดภาพออกเป็น 8 ส่วน เพื่อให้สอดคล้องการใช้หน่วยความจำของเอฟพีจีเอ สำหรับมอดูลประมาณทิศทางของจุดเด่นจะนำค่าสมบูรณ์ของ m_x มาคูณกับค่า $\tan\theta_1, \tan\theta_2, \dots, \tan\theta_n$ และพิจารณาค่าสมบูรณ์ของ m_y อยู่ในช่วงค่า $|m_x| \tan\theta_i$ ได้จะเป็นมุมแสดงทิศทางของจุดมุนั้น โดยให้ผลลัพธ์เป็นสัญญาณดัชนีมุมแสดงช่วงของมุมขนาด 4 บิต (แสดงค่า 0-8) สำหรับการพิจารณาจุดภาคของทิศทางจะดูจากเครื่องหมายของ m_x, m_y มอดูลประมาณทิศทางของจุดเด่นสามารถเขียนแผนภาพกล่องได้ดังรูปที่ 14 โดยสัญญาณข้อมูลทั้งดัชนีมุมและเครื่องหมายของโมเมนต์ในแนวตั้งและนอนจะถูกส่งไปยังมอดูลหมุนจุดภาพตามทิศทางของมุมต่อไป



รูปที่ 14 สถาปัตยกรรมการประมาณทิศทางจุดเด่น

3.7 การหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุน

คู่ของจุดภาพที่ใช้ในการเปรียบเทียบเพื่อสร้างตัวบรรยายซึ่งอยู่ในหน้าต่างขนาด 31x31 จุดภาพจะถูกหมุนด้วยมุมเท่ากับทิศทางของจุดมุนเป็นไปตามสมการ 3.6

$$\begin{bmatrix} rx \\ ry \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.6)$$

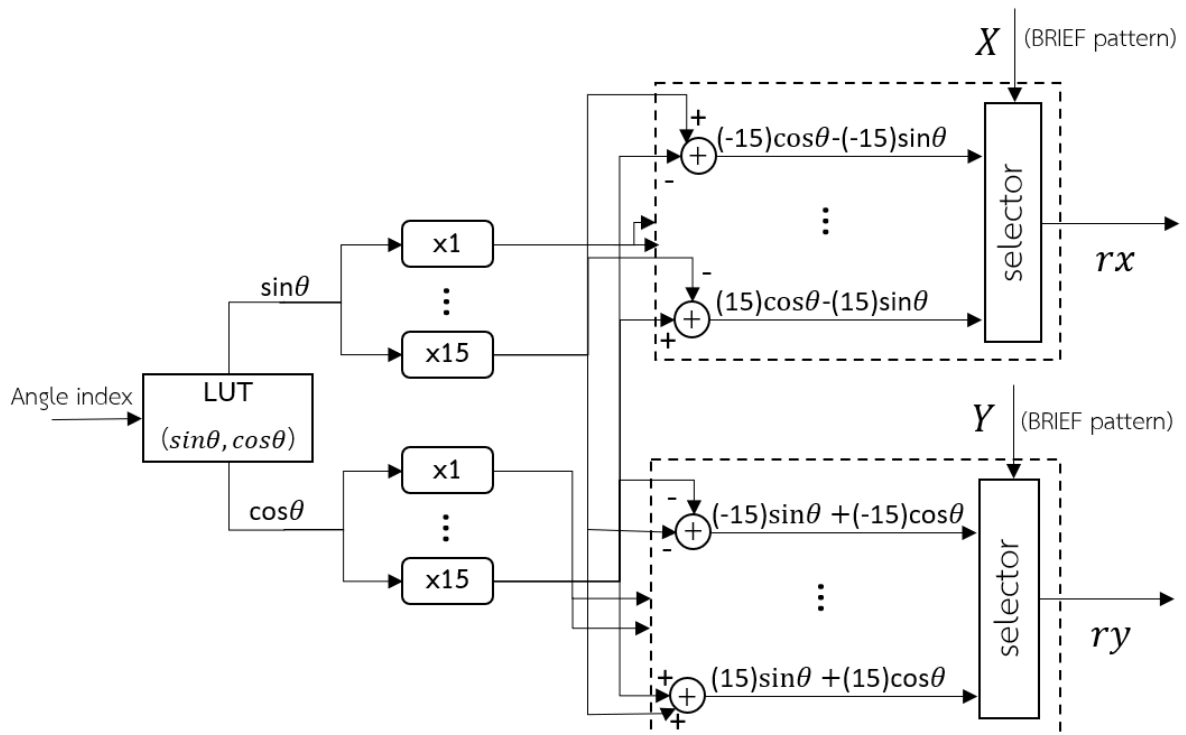
เมื่อ rx, ry คือ พิกัดของจุดภาพที่ผ่านการหมุนแล้วในแนวนอนและแนวตั้ง ตามลำดับ

x, y คือ พิกัดในแนวนอนและแนวตั้งของจุดภาพในแผนภาพที่ใช้สำหรับสร้างตัวบรรยายแบบบรีฟ

θ คือ มุมแสดงทิศทางของจุดมุนที่กึ่งกลางของแผนภาพ

เมื่อได้ดัชนีมุมแสดงทิศทางของจุดเด่นแล้ว มอดูลหมุนจุดภาพจะเลือกค่า $\sin\theta$ และ $\cos\theta$ ที่สอดคล้องกับดัชนีมุมตามค่าในตารางเชื่อมโยงข้อมูลที่ได้บันทึกไว้ก่อนแล้ว เพื่อใช้ในการหมุนตามสมการที่ 3.6 แต่ในการสร้างตัวบรรยายขนาด 256 บิต ต้องใช้จำนวนจุดภาพเพื่อเปรียบเทียบจำนวน 512 จุดภาพ หากดำเนินการตามสมการจะต้องใช้ตัวดำเนินการคูณ 2048 ครั้ง ตัวดำเนินการบวก-ลบอีก 1024 ครั้ง เพื่อลดจำนวนครั้งในการคูณ วิทยานิพนธ์นี้เลือกใช้แนวคิดของ [9] เนื่องจากจุดสนใจที่อยู่ในหน้าต่างภาพจะมีค่า x หรือ y อยู่ในช่วง -15 ถึง 15 อยู่แล้ว ดังนั้นเราสามารถคำนวณค่า $x \cdot \cos\theta$, $x \cdot \sin\theta$, $y \cdot \sin\theta$ และ $y \cdot \cos\theta$ โดยที่ x, y เป็นค่าจำนวนเต็มตั้งแต่ 1 ถึง 15 ได้ก่อนโดยไม่ต้องคำนึงถึงตำแหน่งของจุดภาพที่เป็นคู่ในการเปรียบเทียบ

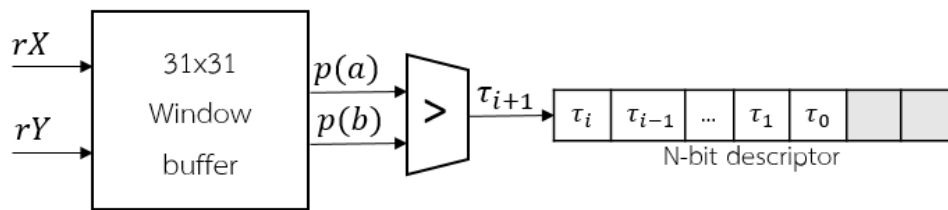
เมื่อทราบค่าตำแหน่งของคู่จุดสนใจที่จะใช้ในการเปรียบเทียบ (BRIEF pattern) ตัวมอดูลจะให้ผลลัพธ์เป็นค่าตำแหน่งที่ผ่านการหมุนซึ่งได้คำนวณไว้ล่วงหน้าแล้ว ส่งไปยังตัวสร้างตัวบรรยายต่อไป วิทยานิพนธ์นี้ได้ออกแบบสถาปัตยกรรมสำหรับหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุมเป็นไปดังรูปที่ 15



รูปที่ 15 สถาปัตยกรรมสำหรับหมุนจุดภาพในหน้าต่างภาพตามทิศทางของจุดมุม

3.8 การสร้างตัวบรรยายแบบบริฟ

มอดูลสร้างตัวบรรยายแบบบริฟได้รับข้อมูลความเข้มของจุดภาพในแถวแต่ละแถวแผ่นภาพขนาด 31×31 จากตัวควบคุมหน่วยความจำที่ผ่านตัวกรองทวินามจะนำข้อมูลนั้นเก็บในวินโดว์บัพเฟอร์สำหรับใช้สร้างตัวบรรยาย เมื่อได้ตำแหน่งของจุดภาพที่ผ่านการหมุนแล้ว ค่าความเข้มแสงของของจุดภาพในวินโดว์บัพเฟอร์ที่ตำแหน่งตรงกับตำแหน่งที่ได้รับจากการหมุนจุดมุมจะถูกหยิบออกมาเปรียบเทียบกับตัวเปรียบเทียบและสร้างเป็นตัวบรรยาย ดังรูปที่ 16



รูปที่ 16 สถาปัตยกรรมสำหรับสร้างตัวบรรยายแบบบริฟ



4. ผลการทดลอง

ในการออกแบบสถาปัตยกรรมเพื่อตรวจจับและบรรยายจุดเด่นบนพื้นฐานขั้นตอนวิธีโออาร์บี ขั้นตอนวิธีที่นำเสนอถูกออกแบบและปรับปรุงด้วยภาษาไพทอนก่อน เพื่อให้สะดวกในการปรับเปลี่ยนรูปแบบการประมวลผล โดยคำนึงถึงความเหมาะสมในการประมวลผลบนฮาร์ดแวร์เป็นหลัก จากนั้นจึงออกแบบและพัฒนาสถาปัตยกรรมตามขั้นตอนวิธีที่ได้ออกแบบไว้ด้วยภาษาวีเอชดีแอล และทดสอบการทำงานด้วยการจำลองผ่านโปรแกรม ModelSim โดยมีรายละเอียดผลการทดลองดังต่อไปนี้

4.1 การปรับปรุงขั้นตอนวิธีการด้วยภาษาไพทอน

เพื่อทดสอบผลการทำงานของขั้นตอนวิธีตรวจจับจุดเด่นแบบมุมและสร้างตัวบรรยายบนพื้นฐานของขั้นตอนวิธีโออาร์บีที่นำเสนอ วิทยานิพนธ์นี้ได้ทดสอบความต้องกันและความแม่นยำของตัวตรวจจับมุมของขั้นตอนวิธีการที่พัฒนาขึ้นกับรูปภาพต่างๆ จากฐานข้อมูล [17] ได้แก่ ชุดรูป UBC และ Boat โดยแต่ละชุดรูปจะประกอบไปด้วยภาพก่อนการแปลงและภาพที่ผ่านการแปลงอีกจำนวน 2 ภาพ และภาพทดสอบ Block และ Lab ดังแสดงในรูปที่ 17



(ก)



(ข)



(ค)



(ง)

รูปที่ 17 ตัวอย่างภาพทดสอบ

(ก) ภาพต้นฉบับ UBC (ข) ภาพต้นฉบับ Boat (ค) ภาพต้นฉบับ Block (ง) ภาพต้นฉบับ Lab

สำหรับรูปภาพจากฐานข้อมูล [17] ที่นำมาทดสอบมีรายละเอียดดังต่อไปนี้

1. ชุดภาพ UBC เป็นชุดภาพที่ประกอบไปด้วยภาพที่บีบอัดด้วยมาตรฐาน JPEG ที่คุณภาพรูป (image quality parameter) เป็น 100 เปอร์เซ็นต์, 40 เปอร์เซ็นต์ และ 16 เปอร์เซ็นต์ ตามลำดับ



(ก)



(ข)



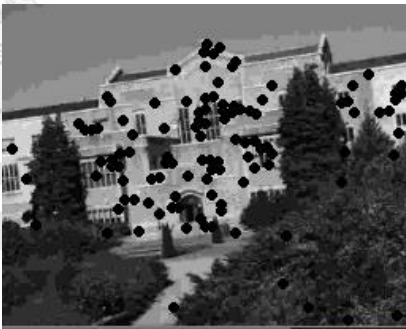
(ค)



(ง)



(จ)



(ฉ)

รูปที่ 18 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ

(ก) ภาพต้นฉบับมีคุณภาพ 100 เปอร์เซ็นต์ (ข) ผลลัพธ์จากภาพ ก

(ค) ภาพต้นฉบับมีคุณภาพ 40 เปอร์เซ็นต์ (ง) ผลลัพธ์จากภาพ ค

(จ) ภาพต้นฉบับมีคุณภาพ 16 เปอร์เซ็นต์ (ฉ) ผลลัพธ์จากภาพ จ

จากภาพที่ 18 เมื่อนำจำนวนจุดเด่นทั้งหมดที่ตรวจจับได้โดยตัวตรวจจับมุมในขั้นตอนวิธีการที่พัฒนาขึ้นมาคำนวณค่าความต้องกันของจำนวนจุดมุมตามสมการ 3.5 พบว่าได้ความต้องกันของจำนวนจุดมุมเป็น 51.31 เปอร์เซ็นต์ และ 46.65 เปอร์เซ็นต์ สำหรับภาพที่คุณภาพ 40 เปอร์เซ็นต์ และ 16 เปอร์เซ็นต์ ตามลำดับ จะพบว่า

เมื่อภาพมีคุณภาพน้อยลง ความตึงกันของจำนวนจุดมุมของตัวตรวจจับมุมก็ลดลงไปด้วย จุดที่เคยเป็นมุมอาจเบลอจนตัวตรวจจับไม่สามารถตรวจจับได้ ในขณะที่เดียวกันบางจุดที่เป็นจุดที่มีความสว่างใกล้เคียงกับจุดภาพโดยรอบ เมื่อคุณภาพรูปต่ำลง ความเข้มของบางจุดก็เปลี่ยนแปลงจนอาจทำให้เกิดมุมไม่จริงให้ตัวตรวจจับตรวจเจอได้

2. ชุดภาพเรือ (boat) เป็นชุดภาพที่ประกอบไปด้วยภาพต้นแบบที่ไม่ผ่านการแปลง และภาพที่ผ่านการซูมออกและหมุนด้วยมุม 30 องศาและ 45 องศาตามลำดับ



รูปที่ 19 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ

จากภาพที่ 19 (ก) ภาพต้นฉบับที่ไม่ผ่านการแปลง (ข) ผลลัพธ์จากภาพ ก (ค) ภาพต้นฉบับที่หมุน 30 องศา (ข) ผลลัพธ์จากภาพ ค (จ) ภาพต้นฉบับที่หมุน 45 องศา (ข) ผลลัพธ์จากภาพ จ เมื่อนำจำนวนจุดเด่นทั้งหมดที่ตรวจจับได้โดยตัวตรวจจับมุมในขั้นตอนวิธีการที่พัฒนาขึ้นมาคำนวณค่าความต้องกันของจำนวนจุดมุมตามสมการ 3.5 พบว่าได้ความต้องกันของจำนวนจุดมุมเป็น 35.04 เปอร์เซนต์ และ 31.86 เปอร์เซนต์ สำหรับภาพที่ได้จากต้นฉบับที่หมุน 30 องศา และ 45 องศา ตามลำดับ จะพบว่าค่าแม่นยำของตัวตรวจจับต่ำและได้น้อยลงเมื่อหมุนมากขึ้น อันเนื่องมาจากเมื่อหมุนด้วยมุมมากขึ้น ภาพที่นำมาตัวจับนั้นปรากฏวัตถุอื่นที่ไม่ได้ปรากฏในภาพต้นฉบับที่ไม่ได้ผ่านการแปลงใดๆ ตัวตรวจจับจึงตรวจจับมุมที่เกิดจากวัตถุใหม่นั้นด้วย จำนวนจุดเด่นที่ตรวจจับได้จึงไม่คงที่ในแต่ละภาพต้นฉบับ ความต้องกันของจำนวนจุดมุมจึงมีค่าต่ำ

สำหรับภาพทดสอบ Block และภาพทดสอบ Lab ก่อนทดสอบจะถูกแปลงด้วยการหมุนภาพต้นฉบับทวนเข็มนาฬิกาเป็นมุม 30 องศาและ 60 องศา เพื่อทดสอบความทนต่อการหมุน และปรับขนาดเป็น 0.8 เท่าและ 0.5 เท่า เพื่อทดสอบความทนต่อการขยายภาพ ได้ค่าความต้องกันและความแม่นยำเปรียบเทียบระหว่างผลที่ได้ขั้นตอนวิธีการที่นำเสนอกับผลที่ได้จากฟังก์ชัน cv.ORB() ใน OpenCV library เป็นดังตารางที่ 2

ตารางที่ 2 ความต้องกันของจุดมุมและความแม่นยำที่ได้จากขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับผลที่ได้จากฟังก์ชัน cv.ORB() ใน OpenCV library

ภาพทดสอบ	การแปลง	ความต้องกัน (%)		ความแม่นยำ (%)	
		งานวิจัยนี้	cv.ORB()	งานวิจัยนี้	cv.ORB()
Block	หมุนด้วยมุม 30 องศา	68.30	42.41	74.62	68.63
	หมุนด้วยมุม 60 องศา	56.45	38.55	50.63	61.64
	ปรับขนาดเป็น 0.8	68.30	51.32	60.88	64.07
	ปรับขนาดเป็น 0.5	42.41	42.41	51.99	51.60
Lab	หมุนด้วยมุม 30 องศา	62.09	62.09	73.50	72.58
	หมุนด้วยมุม 60 องศา	51.32	38.55	53.80	60.25
	ปรับขนาดเป็น 0.8	75.13	51.32	65.88	61.54
	ปรับขนาดเป็น 0.5	62.09	31.86	47.01	41.35

จากตารางที่ 2 ค่าความต้องกันของจุดมุมและความแม่นยำที่ได้จากทั้งขั้นตอนวิธีที่นำเสนอและจากฟังก์ชัน cv.ORB() ใน OpenCV library มีค่าน้อยลงเมื่อมีการแปลงทั้งการแปลงแบบหมุนและการปรับขนาด เมื่อเปรียบเทียบค่าระหว่างผลที่ได้จากขั้นตอนวิธีที่นำเสนอและจากฟังก์ชัน cv.ORB() พบว่า ค่าที่ได้จากขั้นตอนวิธีการที่นำเสนอมีค่ามากกว่าหรือใกล้เคียงกับค่าที่ได้จากฟังก์ชัน cv.ORB() จึงสามารถใช้ทดแทนกันได้

เมื่อเปรียบเทียบค่าความต้องกันของจุดมุมที่ได้จากการทดสอบขั้นตอนวิธีการที่นำเสนอกับภาพทดสอบ UBC, Boat, Block และ Lab พบว่าค่าความต้องกันของจุดมุมภาพ UBC และ Boat มีค่าน้อยกว่า Block และ Lab

เป็นอย่างมาก เนื่องจากภาพ UBC และ Boat เป็นภาพที่มีความละเอียดมาก ลักษณะของจุดมุ่มเป็นลักษณะของหลายมุ่มอยู่ในบริเวณใกล้เคียงกัน ทำให้เกิดข้อผิดพลาดในการตรวจจับมุ่ม ในขณะที่เดียวกันจุดมุ่มที่อยู่ในภาพ Block และ Lab มีลักษณะเป็นมุ่มชัดเจน ไม่กระจุกตัวหนาแน่นที่บริเวณใดบริเวณหนึ่ง ทำให้ตรวจจับจุดเด่นสามารถตรวจจับได้

4.2 การจำลองการทำงานด้วยโปรแกรม ModelSim

วิทยานิพนธ์นี้ทดสอบการทำงานของสถาปัตยกรรมที่นำเสนอด้วยโปรแกรม ModelSim โดยเขียนโปรแกรม testbench ด้วยภาษาวีเอชดีแอลจำลองการป้อนข้อมูลสู่วงจรที่ได้ออกแบบไว้

ข้อมูลนำเข้าที่เป็นรูปภาพสำหรับประมวลผลใน testbench นี้ ใช้ไฟล์รูปภาพที่อยู่ในรูปแบบไฟล์ภาพเชิงจุด (bitmap format) ขนาด 1920x1080 จุดภาพ รูปภาพในรูปแบบภาพเชิงจุดเป็นรูปภาพไม่ได้บีบอัด ทำให้สามารถใช้ STD.TEXTIO library อ่านข้อมูลจากไฟล์รูปภาพได้เลยโดยไม่ต้องผ่านการเข้ารหัสเพื่อคลายการบีบอัดข้อมูล เนื่องจากไฟล์รูปภาพเป็นชนิดภาพเชิงจุดจะมีส่วนหัว (header) ของไฟล์ที่ใช้บันทึกข้อมูลเกี่ยวกับรูปภาพนั้นๆ เช่น ขนาดของรูปภาพ ความละเอียดของภาพ ข้อมูลสี เป็นต้น ซึ่งไม่จำเป็นในการประมวลผล ต้องตัดข้อมูลส่วนหัวขนาด 54 ไบต์ก่อนจึงจะเป็นข้อมูลแต่ละจุดภาพ และเพื่อให้สอดคล้องกับลักษณะข้อมูลขาเข้าของสถาปัตยกรรมที่ออกแบบไว้เป็นภาพระดับสีเทาขนาด 8 บิต ข้อมูลจากการอ่านไฟล์ภาพเชิงจุดซึ่งจะได้ข้อมูลแต่ละสีมาในขนาด 8 บิตจะถูกแปลงเป็นข้อมูลระดับสีเทาด้วย testbench โดยแปลงค่าความเข้มแต่ละจุดภาพดังสมการ

$$\text{Grayscale} = 0.299R + 0.587G + 0.114B \quad (4.1)$$

เมื่อ	Grayscale	แทน ค่าความเข้มของจุดภาพ (ระดับสีเทา)
	R, G, B	แทน ค่าความเข้มสีแดง สีเขียวและสีน้ำเงินของจุดภาพที่อ่านได้ ตามลำดับ

เมื่อได้ข้อมูลความเข้มแต่ละจุดภาพ Testbench จะสร้างสัญญาณควบคุมขาเข้าให้แก่วงจรและจำลองการทำงานของสถาปัตยกรรมที่นำเสนอ แบ่งการทำงานออกเป็นสองส่วนคือส่วนตรวจจับจุดเด่นและส่วนสร้างตัวบรรยาย จำลองการทำงานแยกกัน โดยข้อมูลที่ต้องเก็บใน DDR3 ก่อนจะถูกเปลี่ยนเป็นเก็บในไฟล์ .txt แทน Testbench จะใช้ STD.TEXTIO library เรียกอ่านข้อมูลทดแทนการอ่านข้อมูลจาก DDR โดยมีผลการจำลองการทำงานของสถาปัตยกรรมดังรูปที่ 20 และ รูปที่ 21

ModelSim - INTEL FPGA STARTER EDITION 10.5b

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

Layout Simulate ColumnLayout Default

Objects

Name	Value	Kind	Mode	TC	Row	Col
Ck	1	Signal	Internal			
EndBinomial	0	Signal	Internal			
FRdData	0110...	Signal	Internal			
FRdEmpty	1	Signal	Internal			
FRdEn	0	Signal	Internal			
PyramidReady	0	Signal	Internal			
RotB	1	Signal	Internal			
TM	1	Signal	Internal			
TT	0	Signal	Internal			
b_in	1111...	Signal	Internal			
bmp_file	labTe...	WhD...	Internal			
g_in	1111...	Signal	Internal			
heapRdy	1	Signal	Internal			
out_b...	WhD...	Internal				
q	0	Signal	Internal			
r_in	1111...	Signal	Internal			
r_daddress	0000...	Signal	Internal			
starBinomial	0	Signal	Internal			
tCk	1000...	Con...	Internal			

Wave - Default

Transcript

```

*** Note: image_width: 1980, image_height: 1080
# Time: 1 us Iteration: 0 Instance: /tborb
*** Note: ##### Start Binomial Filter #####
# Time: 26685 ns Iteration: 0 Instance: /tborb
*** Note: ##### End Binomial Filter #####
# Time: 20739720 ns Iteration: 0 Instance: /tborb
*** Note: ----- Coordinate of feature Points is Ready-----
# Time: 20945100 ns Iteration: 3 Instance: /tborb
*** Failure: ##### End Detection #####
# Time: 20945100 ns Iteration: 3 Process: /tborb/u_Test File: ../testbench/TbORB.vhd
# Break in Process in Test at /testbench/TbORB.vhd line 295

```

(ก)

Transcript

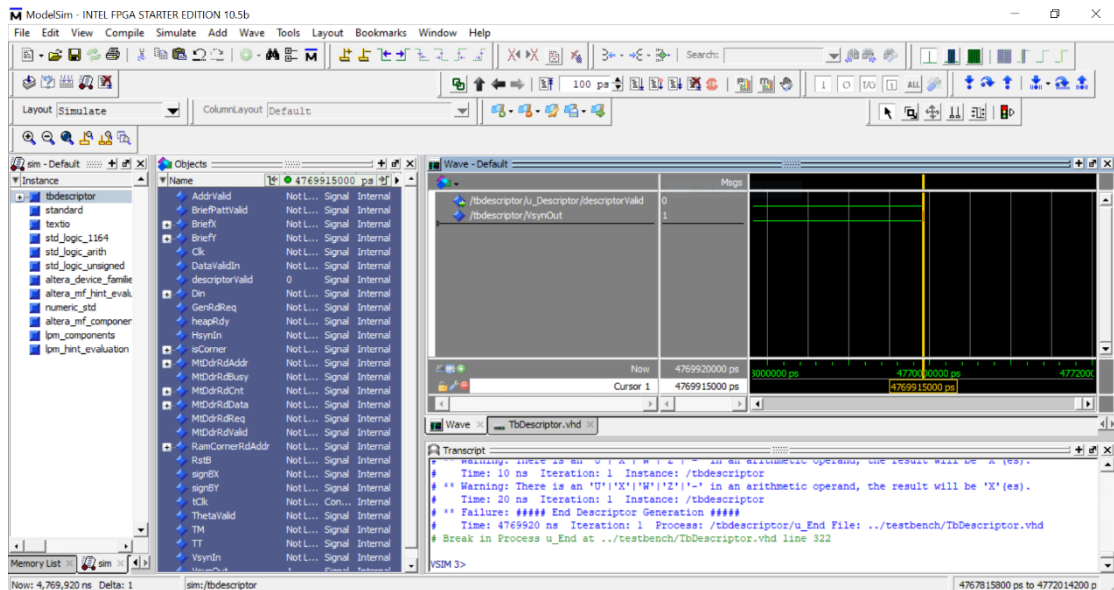
```

# *** Note: image_width: 1980, image_height: 1080
# Time: 1 us Iteration: 0 Instance: /tborb
# *** Note: ##### Start Binomial Filter #####
# Time: 26685 ns Iteration: 0 Instance: /tborb
# *** Note: ##### End Binomial Filter #####
# Time: 20739720 ns Iteration: 0 Instance: /tborb
# *** Note: ----- Coordinate of feature Points is Ready-----
# Time: 20945100 ns Iteration: 3 Instance: /tborb
# *** Failure: ##### End Detection #####
# Time: 20945100 ns Iteration: 3 Process: /tborb/u_Test File: ../testbench/TbORB.vhd
# Break in Process in Test at /testbench/TbORB.vhd line 295

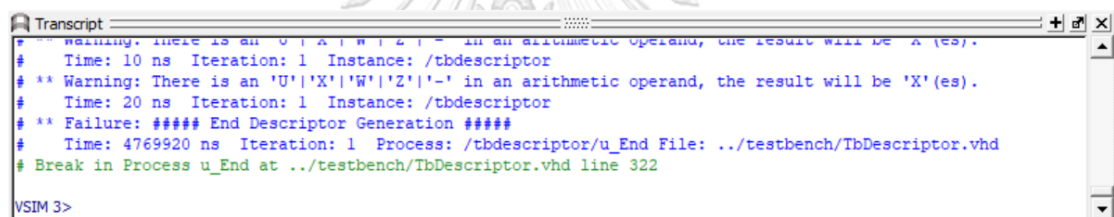
```

(ข)

รูปที่ 20 ผลการจำลองการทำงานของสถาปัตยกรรมส่วนตรวจจับจุดเด่นด้วยโปรแกรม ModelSim



(ก)



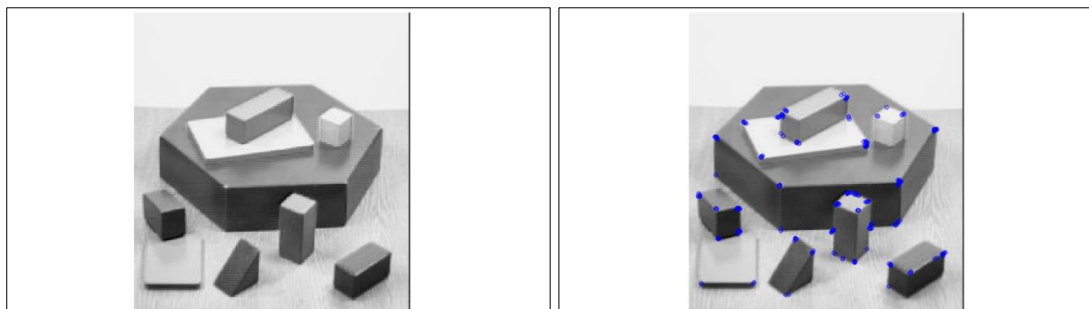
(ข)

รูปที่ 21 ผลการจำลองการทำงานของสถาปัตยกรรมส่วนสร้างตัวบรรยายด้วยโปรแกรม ModelSim

จากรูปที่ 20 พบว่าวงจรส่วนตรวจจับจุดเด่น ตรวจจับจุดเด่นพร้อมจัดเรียง 256 จุดเด่นที่จะนำมาพิจารณา ใช้เวลาในการประมวลผล 20,945,100 ns ในขั้นตอนนี้ตัวกรองแบบทวินามเริ่มทำงานขนานกันและทำงานเสร็จที่ 20,739,720 ns เก็บข้อมูล เตรียมรอไว้สำหรับให้วงจรส่วนสร้างตัวบรรยายเรียกข้อมูลไป

จากรูปที่ 20 พบว่าวงจรส่วนสร้างตัวบรรยายทำงานเสร็จใน 4,769,920 ns ต่อจากการทำงานส่วนตรวจจับจุดเด่นทำงานเสร็จ รวมแล้วใช้เวลาในการประมวลผลรวมทั้งสิ้น 25,715,020 ns หรือ 25.715020 ms ทำให้สามารถรองรับสัญญาณภาพขนาด 1920x1080 จุดภาพที่อัตราเฟรม 30 เฟรมต่อวินาทีได้ (ใช้เวลาต่อเฟรมไม่เกิน 33.33 ms)

ผลลัพธ์ตำแหน่งของจุดมุมและตัวบรรยายจะถูกเขียนลงบนไฟล์ .txt เพื่อนำไปประมวลผลต่อด้วยโปรแกรมภาษาไพทอนเพื่อวาดจุดที่เป็นจุดสนใจที่ตรวจจับได้และหาค่าความตึงกันของจุดมุมและความแม่นยำได้ผลรูปที่ 22



รูปที่ 22 จุดเด่นที่ตัวตรวจจับมุมในขั้นตอนวิธีที่นำเสนอตรวจจับได้เทียบกับภาพต้นฉบับ

เนื่องจากภาพทดสอบ Block มีอัตราส่วนด้านกว้างต่อด้านยาวเป็น 1:1 จึงใช้วิธีเดิมจุดภาพสีขาวที่ด้านข้างของภาพทดสอบ เพื่อให้มีขนาด 1920x1080 จุดภาพ และทดสอบการทำงานของสถาปัตยกรรม ได้ความแม่นยำเป็น 73.85%

4.3 ผลการสังเคราะห์วงจรและการใช้ทรัพยากร

สถาปัตยกรรมเพื่อตรวจจับและบรรยายจุดเด่นบนพื้นฐานขั้นตอนวิธีโออาร์บีถูกออกแบบสำหรับ DECA Altera MAX 10 FPGA evaluation board โดยใช้โปรแกรม Quartus Prime 17.1 Lite Edition ในการสังเคราะห์วงจร ได้ผลลัพธ์การใช้ทรัพยากรและผลการทำงานเปรียบเทียบกับงานวิจัยที่เกี่ยวข้องดังตารางที่ 3

ตารางที่ 3 การเปรียบเทียบการใช้ทรัพยากรและผลการทำงานระหว่างวิทยานิพนธ์และงานวิจัยที่เกี่ยวข้อง

การใช้ทรัพยากร	TH.PHAM [9]	J.Huang [15]	J.Weberruss [12]	W.Fang [11]	R.Sun [13]	R.Liu [14]	วิทยานิพนธ์
Logic element	12,503	80,472	206,000	25,648	28,168	56,954	27,768
Registers	10,574	112,166	231,973 +449DSPs	21,791 +8DSPs	9,528 +33DSPs	67,809FF+ 111DSPs	16,649
Frame rate	84.4	310	72.3	67	108	55.87	30
Throughput (MPixel/s)	175	81	150	20.5	223	17	62
Resolution	1980x1080	512x512	1920x1080	640x480	1920x1080	640x480	1920x1080
ความถี่ทำงาน	175 MHz	100 MHz	150 MHz	203 MHz	200 MHz	100 MHz	100 MHz

จากตารางที่ 3 จะเห็นว่าสถาปัตยกรรมที่นำเสนอสามารถประมวลผลสัญญาณภาพความละเอียดไม่เกิน 1920x1080 จุดภาพ ที่อัตรา 30 เฟรมต่อวินาที แม้ความถี่ที่ใช้ในการทำงานและ Throughput จะมีค่าน้อยกว่างานวิจัยที่เกี่ยวข้องบางชิ้น แต่ทดแทนด้วยการใช้ทรัพยากรน้อยกว่า ไม่มีการใช้งานวงจรจำนวนมากทางคณิตศาสตร์ใดๆ โดยที่ยังสามารถทำงานบนเวลาจริง ภายใต้เงื่อนไขขนาดของสัญญาณภาพและอัตราเฟรม 30 เฟรมต่อวินาทีได้

5. สรุปผล

5.1 สรุปผล

วิทยานิพนธ์นี้นำเสนอสถาปัตยกรรมเพื่อตรวจจับและบรรยายจุดเด่นบนพื้นฐานของขั้นตอนวิธีโออาร์บี โดยพัฒนาขั้นตอนวิธีที่นำเสนอด้วยภาษาไพทอน เพื่อปรับปรุงขั้นตอนวิธีประมวลผลให้สอดคล้องกับการประมวลผลในฮาร์ดแวร์ และพัฒนาสถาปัตยกรรมด้วยภาษาวีเอชดีแอลให้สามารถรองรับสัญญาณภาพความละเอียดไม่เกิน 1920x1080 จุดภาพ ที่อัตรา 30 เฟรมต่อวินาที

ขั้นตอนวิธีที่นำเสนอใช้หลักการจัดขั้นตอนบางขั้นตอนที่สามารถทำงานขนานกันได้ให้ทำงานพร้อมกัน เช่น การตรวจจับลักษณะภาพด้วยขั้นตอนวิธีแบบพาสต์กับการคำนวณค่าคะแนนความเป็นมุม และการหาจุดเด่นที่จะพิจารณาเกี่ยวกับการทำภาพให้เรียบเพื่อเตรียมสร้างตัวบรรยายแบบบริฟ เป็นต้น การปรับเปลี่ยนการลดการสุ่มตัวอย่างและการทำภาพให้เรียบจากการใช้ตัวกรองแบบเกาส์เซียนเป็นการใช้ตัวกรองแบบทวินามให้สามารถประมวลผลได้โดยใช้เฉพาะการดำเนินการบวกและการเลื่อนบิตได้ การออกแบบการจัดเรียงจุดมุมตามค่าคะแนนความเป็นมุมด้วยการเรียงแบบฮีป การปรับปรุงขั้นตอนในการคำนวณโมเมนต์ของจุดมุมให้สามารถเริ่มคำนวณทั้งแนวตั้งและแนวนอนได้พร้อมกันเมื่อได้รับข้อมูลจุดภาพแต่ละแถวในแผ่นภาพ 31x31 การปรับปรุงการหาช่วงทิศทางของจุดมุมโดยใช้ตารางเชื่อมโยงข้อมูลและการพิจารณาผลคูณแทนการใช้ฟังก์ชันผกผันตรีโกณมิติ เป็นการปรับปรุงขั้นตอนวิธีตรวจจับและบรรยายจุดเด่นทำให้เหมาะในการออกแบบสถาปัตยกรรมสำหรับเอฟพีจีเอ โดยยังคงค่าความต้องกันของจุดมุมและความแม่นยำ เมื่อเทียบกับผลที่ได้จากฟังก์ชัน cv.ORB() จาก OpenCV library

สถาปัตยกรรมที่นำเสนอถูกออกแบบบนบอร์ด DECA ALTERA MAX 10 FPGA ทดสอบการทำงานด้วยการจำลองการทำงานบน ModelSim สามารถทำงานตามเวลาจริง (real-time) ได้กับสัญญาณภาพความละเอียดไม่เกิน 1920x1080 จุดภาพ ที่อัตรา 30 เฟรมต่อวินาที

หากพิจารณาค่าความต้องกันของจุดมุมและความแม่นยำจะพบว่าภาพข้อมูลขาเข้าที่เหมาะสมในการตรวจจับและบรรยายจุดสนใจโดยใช้ขั้นตอนวิธีและสถาปัตยกรรมที่นำเสนอจะเป็นภาพที่มีจุดเด่นเป็นมุมที่มีชัดเจน ไม่กระจุกตัวรวมกัน และภาพที่ยังมีข้อจำกัดในการใช้ขั้นตอนวิธีและสถาปัตยกรรมที่นำเสนอคือ ภาพที่มีมุมไม่ชัดเจน มีมุมหลายมุมอยู่ในบริเวณใกล้เคียงหรือมุมลักษณะเป็นมุมมากกว่า 1 มุมชนกัน จะไม่สามารถตรวจจับมุมบริเวณนั้นได้

5.2 ข้อเสนอแนะ

สถาปัตยกรรมเพื่อตรวจจับและบรรยายจุดสนใจบนพื้นฐานขั้นตอนวิธีโออาร์บีที่นำเสนอถูกออกแบบจำเพาะต่อขนาดของภาพขาเข้า (1920x1080 จุดภาพ) แต่ในการใช้งานจริง สัญญาณภาพขาเข้าอาจมีขนาดอื่นได้ขึ้นกับตัวรับภาพ งานวิจัยในอนาคตควรออกแบบสถาปัตยกรรมให้ไม่จำเพาะต่อขนาดของสัญญาณภาพขาเข้า โดยผู้ใช้งานสามารถปรับเปลี่ยนได้ตามความต้องการหรือสามารถรองรับสัญญาณภาพหลายขนาดโดยไม่ต้องปรับแต่งเพิ่มจากผู้ใช้

สำหรับขั้นตอนวิธีตรวจจับมุมแบบพาสต์ที่มีการใช้ค่าขีดแบ่งเป็นเกณฑ์ในการพิจารณา หากสภาพแสงของรูปขาเข้ามีค่าเปลี่ยนไป ระบบจำเป็นต้องปรับค่าขีดแบ่งโดยผู้ใช้ งานวิจัยในอนาคตควรออกแบบตัวมอดูลที่สามารถ

ปรับค่าซีดแบ่งที่ใช้ในขั้นตอนตรวจจับจุดเด่นแบบฟาสต์ได้อัตโนมติ โดยอาจพิจารณาจากค่าความเข้มแสงรวมของภาพข้อมูลขาเข้าในช่วงแรก

สำหรับการนำไปใช้ เนื่องจากขั้นตอนวิธีที่นำเสนอเป็นขั้นตอนวิธีที่เหมาะสมในการตรวจจับจุดเด่นที่มีลักษณะเด่นเป็นมุมที่ชัดเจน ไม่รวมจุดยอดมุมของหลายมุมไว้ในจุดเดียวกัน ตัวอย่างการนำไปใช้ที่เหมาะสม เช่น การตรวจจับและบรรยายจุดเด่นของภาพถ่ายดาวเทียมในเมืองสำหรับต่อภาพถ่ายจากหลายมุมมองเข้าเป็นภาพใหญ่ หรือการตรวจจับและบรรยายจุดเด่นของภาพที่ได้จากโดรนที่ใช้สำรวจเมืองหรือสิ่งก่อสร้าง แต่หากเป็นภาพที่มีจุดเด่นเป็นมุมที่มีจุดยอดมุมอยู่ที่จุดเดียวกัน เช่น ตาราง ตัวตรวจจับของขั้นตอนวิธีที่นำเสนอจะไม่สามารถตรวจจับได้ ดังนั้น ก่อนนำไปใช้งาน ต้องพิจารณาลักษณะเด่นและขนาดของสัญญาณภาพด้วย



ภาคผนวก
บทความที่ได้ตีพิมพ์ใน iEECON 2021



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Feature Detection and Description based on ORB algorithm for FPGA-based Image Processing

Taksaporn Imsaengsak
 Dept. of Electrical Engineering
 Chulalongkorn University
 Bangkok, Thailand
 e-mail: taksaporn.im@gmail.com

Suree Pumrin
 Dept. of Electrical Engineering
 Chulalongkorn University
 Bangkok, Thailand
 e-mail: suree.p@chula.ac.th

Abstract— Feature detection and description is a key concept of image processing and computer vision. Oriented FAST and Rotated BRIEF (ORB) algorithm is one of feature detector and descriptor that is widely used in image stitching and registration systems such as simultaneous localization and mapping (SLAM) systems. ORB algorithm has improved the speed of processing and robustness. It has scale-invariance and rotational invariance properties. However, there are some obstacles to implement on low-power devices such as its high computational intensity. This paper presents the feature detection and description algorithm based on ORB algorithm which adopted parallel and pipeline techniques to accelerate the system and be beneficial for designing FPGA architecture. Compared with the results obtained by cv.ORB() function, the proposed algorithm improves the parallelism and reduces the operation of the process but could maintain the consistency of the corner numbers and the accuracy.

Keywords— Feature detection and description, ORB algorithm, FPGA, Image processing

I. INTRODUCTION

Image processing is applied in many fields for data acquisition. Sometimes it cannot capture the whole image by using a single sensor or camera. Image stitching and registration play an important role in this case. The feature points are localized in each input image and then the descriptor of each feature point is generated to describe the local appearance around it. The two feature point sets of each image are matched and merge those two images. Scale Invariant Feature Transform (SIFT) algorithm is a classic feature detector and descriptor which was proposed by [1] in 1999. The keypoints are detected by using Harris Corner Detector. Speeded-Up Robust Feature (SURF) algorithm proposed by [2] in 2006 accelerates the detector by using Hessian matrix-based measure. SIFT and SURF cannot meet real-time requirements. Rublee et al. [3] proposed Oriented FAST and Rotated BRIEF (ORB) algorithm. The algorithm uses Feature from Accelerated Segment Test (FAST) algorithm and Binary Robust Independent Elementary Features (BRIEF) algorithm as the feature detector and descriptor, respectively. Their technique accelerates the process to meet real-time requirements. ORB algorithm also has scale-invariance and rotational invariance properties. ORB algorithm is widely applied in many applications such as satellite images and Simultaneous Localization and Mapping (SLAM).

However, the implementation of ORB algorithm on a low-power platform is challenging for high-resolution image processing because of its high computational intensity. There are many previous efforts to design and accelerate the ORB-based system on various platforms. Field Programmable Gate Arrays (FPGA) is an interesting hardware accelerator that can process data in parallel. In 2017, Fang et al. [4] proposed FPGA architecture for ORB-SLAM system. To reduce hardware consumption, the word length in orientation estimation is optimized. Huang et al. [5] investigated the FPGA architecture of FAST and BRIEF algorithm. The pipeline design technique is used for making the system meet the real-time requirements. Minimizing memory accesses also reduces latency. Liu et al. [6] and Sun et al. [7] also applied pipeline structure and parallel workflow in their system. Not only the time constraint but also the robustness and the accuracy of the system have to be considered. Weberrus et al. [8] suggested that using the image pyramids with $\frac{1}{2}$ bilinear filter and two sublevels of each octave could avoid quality loss. The sublevel is calculated by applying a $\frac{1}{4}$ filter to the previous level. Weberrus et al. [8] also used bilinear interpolation to generate each image pyramid level by a scaling factor of $\frac{1}{4}$.

These recent developments in ORB implementation suggest that exploiting parallelism and pipelining in the algorithm is an interesting technique to reduce execution time. Robustness and accuracy are also important. This article aims to design the feature detector and descriptor based on ORB algorithm, which adopted parallel and pipeline techniques to accelerate the system and be beneficial for designing FPGA architecture. This paper is organized as follows. Section II reviews the overview of ORB algorithm. Section III presents the proposed algorithm for feature detection and description based on ORB algorithm. Section IV shows the experimental results. Finally, the paper concludes in section V.

II. ORB ALGORITHM

ORB algorithm is built based on FAST corner detection and BRIEF description algorithm. The multi-scale feature and the orientation of the feature point are taken into account to improve scale- and rotational invariance. ORB algorithm consists of two main steps: oriented FAST corner detection and rotated BRIEF feature description.

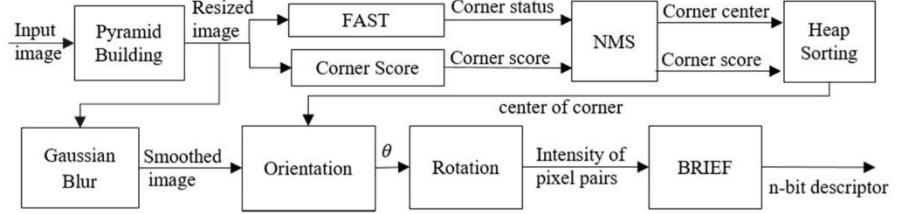


Fig. 1. Overview of the proposed algorithm.

A. Oriented FAST corner detection

FAST detection is a corner detection proposed by [9] in 2006. The detector uses 16 pixels on the Bresenham circle of radius 3 around a candidate pixel to classify whether the candidate pixel is a corner. The bright point is a pixel in Bresenham that is brighter than the intensity of the candidate pixel plus a threshold. The dark point is darker than the intensity of the candidate pixel minus the threshold. The candidate pixel is classified as a corner if the number of contiguous bright or dark pixel is larger than nine. Since the FAST detector could detect multiple corner points in adjacent locations. The corner score is computed by using Harris's corner score or the sum of absolute difference between the candidate point and the pixels on the Bresenham circle. Non-maximal suppression is applied to discard the candidate point with the lower corner score.

Fast corner detector is applied to each level of the image pyramid to improve scale-invariance. The image pyramid is built by smoothing the image with a low-pass filter and then subsampling the smoothed image. Gaussian filter is widely used for smoothing the image. Each level of image pyramid is computed by scaling-down by a scaling factor of $\frac{1}{2}$.

Before generating the BRIEF descriptor, the orientation of the feature point is calculated. The sampling pairs that are used to create the descriptor are rotated according to the orientation of the patch in order to improve the rotational invariance. The orientation of the corner is estimated from the moment of the area around the feature point. There are two components of the moment, horizontal component m_x and vertical component m_y . The moment of a feature point is calculated by the summation of weighted intensity,

$$\begin{cases} m_x = \sum_{x=-r}^r xI(x, y), \\ m_y = \sum_{y=-r}^r yI(x, y), \end{cases} \quad (1)$$

where $I(x, y)$ is the intensity of the pixel at the coordinate (x, y) . The orientation of the corner, θ , is determined by,

$$\theta = \arctan\left(\frac{m_y}{m_x}\right). \quad (2)$$

B. rBRIEF description

The rBRIEF descriptor is a binary descriptor. Each bit of the descriptor is computed by comparing the intensity of two pixels $p(a)$ and $p(b)$ sampled from the image patch centered at the feature point,

$$\tau(p; a, b) = \begin{cases} 1, & p(a) < p(b), \\ 0, & p(a) \geq p(b), \end{cases} \quad (3)$$

where $\tau(p; a, b)$ is the comparison result between $p(a)$ and $p(b)$. The coordinates of the sampled pairs are rotated according to the orientation of the feature point. Because rBRIEF descriptor is sensitive to noise, the source image has to be smoothed by Gaussian filters.

III. PROPOSED ALGORITHM

For maximizing FPGA's performance, the algorithm implemented on FPGA should be developed to be suitable for parallel processing and usage of resources. The proposed algorithm is adjusted to be more compatible with the FPGA architecture. It adds extra steps that make the algorithm results properly. The overview of the proposed algorithm for feature detection and description based on ORB algorithm is shown in Fig 1.

There are two main parts of the proposed algorithm, which are the feature detection module and the descriptor generator module. First of all, the image pyramid is constructed from the input image by downsampling with a scale factor of $\frac{1}{2}$ and using bilinear interpolation. Each level of the image pyramid becomes the source image of both the feature detection module and descriptor generation module.

In the feature detection module, the resized image is stored in the buffer. The 7×7 pixels sliding window moves across the image. To reduce memory access, the intensity of pixels in sliding window are sent to FAST module and corner score module at the same time. The sum of absolute differences between the intensity of the candidate pixel and each pixel on the circle is considered as the corner score. The candidate pixels with non-local maximum score value in 3×3 pixels window are removed by Non-maximum Suppression (NMS).

After filtering with the NMS Module, some edges are detected by FAST detector. Only the first N feature points with the highest corner score are considered as actual feature points. Heap sorting algorithm is used to sort the candidate feature point from FAST detector in non-decreasing order of the corner score of each point. The corner score of each candidate feature point is stored in each node of the complete binary tree. Between a parent node and its children nodes, there is

a sort node that swaps the value of the parent node, left-child node and right-child node to maintain sorting conditions. The parent node must not be greater than the children node. The left-child node is not greater than the right-child node. In each iteration, the smallest one is set at the root node and compared with the corner score of the new coming feature point. If the new corner score is not greater than the root node, the new point is disregarded. Otherwise, the root node is removed and a new iteration is initiated to arrange the remaining data and the new coming point. The number of the feature points that are considered in the next step could be set by the level of the binary tree. This step could limit the number of considered feature points and reduce the computing time for generating the descriptor of insignificant feature points.

In the descriptor module, the binomial filter is applied to the resized image instead of Gaussian filter. Because no multiplier is required, a binomial filter is implemented effectively in FPGA [9]. The 31×31 -pixel patch centered at the detected feature points of the smoothing image is used to estimate the orientation of feature points. For computing the horizontal component of the moment, the sum of pixel's intensity of each row on the patch is multiplied with the number of rows according to (1). The horizontal component is the summation of the multiplication results. In the same way, the vertical component of the moment is the summation of the multiplication between the sum of the pixel's intensity of each column on the patch and the number of columns.

In the orientation estimation module, the angle in the quadrant is divided into N sections. The angle that makes the multiplication between the absolute value of the horizontal component and the tangent of that angle close to the absolute value of the vertical component is considered as the orientation of the feature point.

The look-up table (LUT) is used in the rotator module. The LUT consists of the value of sine and cosine corresponding to the N angles. The sign of sine and cosine value is considered from the sign of m_x and m_y . Generating a 256-bit descriptor requires comparing 512 rotated pixels that means 2048 multiplications and 1024 additions are required. To reduce the number of operations, our proposed algorithm performs the rotation of the image patch instead. The pixels in the rotated image at the coordinate according to the sampling pattern are transmitted to BRIEF module to create a descriptor. The results of the algorithm are the coordinates of feature points and their binary descriptor.

Our proposed algorithm for feature detection and description based on ORB algorithm is implemented in Python for evaluation. The experimental results and discussion are shown in the next section.

IV. EXPERIMENTAL RESULT

In this section, the results of the proposed algorithm are compared with the results of ORB function, `cv.ORB()`, in OpenCV library. This paper

uses the consistency of corner numbers (CCN) and the accuracy (ACU) proposed by [10] as the criteria for evaluating the performance.

The consistency of corner numbers is measured,

$$CCN = 100 \times 1.1^{-|N_t - N_0|}, \quad (4)$$

where N_0 is the number of corners detected in the original image and N_t is the number of corners detected in the transformed image.

The accuracy is computed from

$$ACU = 100 \times \frac{N_m + N_g}{N_d + N_g}, \quad (5)$$

where ACU is an accuracy, N_d is the number of detected feature points, N_g is the number of feature points in the ground truth, and N_m is the number of matched corners compared between the feature points of the detector and the ground truth.



Fig. 2. Example of Image data set, (a) The original of UBC, (b) The original of Boat, (c) and (d) Transformed Boat image.

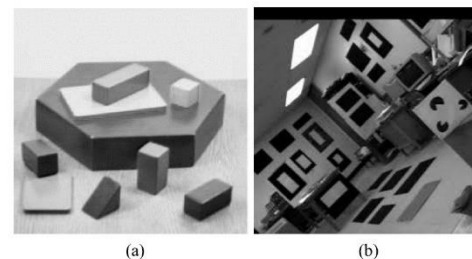


Fig. 3. Example of test images, (a) Block, (b) Lab.

TABLE I. THE CONSISTENCY OF CORNER NUMBERS OF THE PROPOSED ALGORITHM

Image Set	Transformation	CCN (%)
UBC	image quality parameter of 40 %	51.31
	image quality parameter of 16 %	46.65
Boat	zoom + rotation (Fig. 2(c))	35.04
	Zoom + rotation (Fig. 2(d))	31.86

TABLE II. CONSISTENCY OF CORNER NUMBERS COMPARISON BETWEEN THE PROPOSED ALGORITHM AND cv.ORB() FUNCTION

Test image	Transformation	CCN (%)	
		Proposed algorithm	cv.ORB() in OpenCV
Block	Rotating with 30°	46.65	42.41
	Rotating with 60°	12.28	12.28
Lab	Scaling by 0.8	46.65	51.32
	Scaling by 0.5	31.86	31.86

TABLE III. ACCURACY COMPARISON BETWEEN PROPOSED ALGORITHM AND cv.ORB() FUNCTION

Test image	Transformation	ACU (%)	
		Proposed algorithm	cv.ORB() in OpenCV
Block	Rotating with 30°	73.14	66.93
	Rotating with 60°	49.62	60.41
Lab	Scaling by 0.8	65.88	61.06
	Scaling by 0.5	47.00	41.22

For evaluation, the affine covariant regions datasets [11], which are UBC and boat image sets, are applied. Each image set consists of the original image and transformed images. The original images are shown in Fig. 2(a) and 2(b). The transformed UBC image sets the image quality parameter of JPEG standard of 40% and 16%. The transformed boat image set is created by rotating and scaling the original image as shown in Fig.2(c) and 2(d). Table I displays the CCN of our proposed algorithm when detecting feature points of UBC and boat image sets. It shows a dramatic decrease in the CCN and ACU of both UBC and boat test images. The CCN of the UBC image set is decreased from 51.31 % to 46.65 % because of the lower quality of images. Some pixels are classified as corners when the image quality is low. For the boat image set, some new objects appear in the camera frame of the rotated image. Hence, the CCN of the boat image set is quite low.

The proposed algorithm also evaluates block and lab test images as shown in Fig. 3. The test images consist of the original image and the rotated images created by rotating the original image by 30 degrees and 60 degrees and scaling factor of 0.8 and 0.5. The CCN and ACU of the block and lab are displayed in Table II and Table III, respectively. Our proposed algorithm is able to detect the feature as well as cv.ORB() algorithm. The CCN decreases dramatically during the rotational and scaling transformation but the ACU drops gradually. The reason could be that the key feature point still is detected by the detector of both the proposed algorithm and cv.ORB().

V. CONCLUSIONS

This paper proposes a feature detector and descriptor based on ORB algorithm. The detection and description process could be accelerated by exploiting parallelism and pipelining in the algorithm. By applying binomial filter instead of Gaussian filter and designing the data processing pipeline and orientation estimation method make the algorithm process in

parallel. Thus, they would be beneficial to design hardware architecture. To balance the number of feature points, Heap sorting algorithm is applied before generating the descriptor, which is suitable for FPGA implementation. The consistency of corner numbers and accuracy of the proposed algorithm are still maintained compared to that of cv.ORB() function with the same parameter. However, the feature detector in the proposed algorithm could be suitable for detecting single corner points. It appears that the adjacent corner points which have a common vertex could not be detected by the proposed algorithm. In our future work, a compatible hardware architecture based on the proposed algorithm will be designed and implemented on an FPGA board.

ACKNOWLEDGMENT

This research has been supported by Embedded System and IC Design (ESID) Research Laboratory and Electrical Engineering Chulalongkorn university, Master Honours Program Scholarship.

REFERENCES

- [1] D.G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision. 1999.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Speeded-Up Robust Features (SURF)," Computer Vision and Image Understanding, 2008, :(3)110 p. .359-346
- [3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in 2011 International Conference on Computer Vision, .2011
- [4] W. Fang, Y. Zhang, B. Yu, and S. Liu, "FPGA-based ORB feature extraction for real-time visual SLAM," in 2017 International Conference on Field Programmable Technology (ICFPT). 2017
- [5] J. Huang, G. Zhou, X. Zhou, and R. Zhang, "A new FPGA architecture of FAST and BRIEF algorithm for on-board corner detection and matching," Sensors, :18 .2018p. .1014
- [6] R. Liu, J. Yang, Y. Chen, and W. Zhao, "eSLAM: An Energy-Efficient Accelerator for Real-Time ORB-SLAM on FPGA Platform," in 56 2019th ACM/IEEE Design Automation Conference (DAC), .2019
- [7] R. Sun, J. Qian, R. H. Jose, Z. Gong, R. Miao, W. Xue, and P. Liu, "A Flexible and Efficient Real-Time ORB-Based Full-HD Image Feature Extraction Accelerator," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, :(2)28 .2020p. .575-565
- [8] J. Weberruss, L. Kleeman, D. Boland, and T. Drummond, "FPGA acceleration of multilevel ORB feature extraction for computer vision," in 27 2017th International Conference on Field Programmable Logic and Applications (FPL). .2017
- [9] E. Rosten, and T. Drummond, "Fusing points and lines for high performance tracking," in Tenth IEEE International Conference on Computer Vision (ICCV' 05Volume .2005 .1
- [10] F. Mokhtarian, and F. Mohanna, "Performance evaluation of corner detectors using consistency and accuracy measures," Computer Vision and Image Understanding, 2006. 102(1): p. 81-94.
- [11] Group, O.V.G. Affine Covariant Regions Datasets. 2007; Available from: <http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html>



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บรรณานุกรม

1. Lowe, D.G. *Object recognition from local scale-invariant features*. in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999.
2. Bay, H., et al., *Speeded-Up Robust Features (SURF)*. *Computer Vision and Image Understanding*, 2008. **110**(3): p. 346-359.
3. Rosten, E. and T. Drummond. *Fusing points and lines for high performance tracking*. in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. 2005.
4. Calonder, M., et al., *BRIEF: Binary Robust Independent Elementary Features*. Vol. 6314. 2010. 778-792.
5. Rublee, E., et al. *ORB: An efficient alternative to SIFT or SURF*. in *2011 International Conference on Computer Vision*. 2011.
6. Alif, A.F.M.F., S.M.R. Islam, and P. Deb. *Design and implementation of sorting algorithms based on FPGA*. in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*. 2019.
7. ben jmaa, Y., et al., *A Comparative Study of Sorting Algorithms with FPGA Acceleration by High Level Synthesis*. *Computación y Sistemas*, 2019. **23**.
8. Aubury, M. and W. Luk, *Binomial filters*. *Journal of VLSI signal processing systems for signal, image and video technology*, 1996. **12**(1): p. 35-50.
9. Pham, T.H., P. Tran, and S. Lam, *High-Throughput and Area-Optimized Architecture for rBRIEF Feature Extraction*. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019. **27**(4): p. 747-756.
10. Mokhtarian, F. and F. Mohanna, *Performance evaluation of corner detectors using consistency and accuracy measures*. *Computer Vision and Image Understanding*, 2006. **102**(1): p. 81-94.
11. Fang, W., et al. *FPGA-based ORB feature extraction for real-time visual SLAM*. in *2017 International Conference on Field Programmable Technology (ICFPT)*. 2017.

12. Weberuss, J., et al. *FPGA acceleration of multilevel ORB feature extraction for computer vision*. in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. 2017.
13. Sun, R., et al., *A Flexible and Efficient Real-Time ORB-Based Full-HD Image Feature Extraction Accelerator*. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020. **28**(2): p. 565-575.
14. Liu, R., et al. *eSLAM: An Energy-Efficient Accelerator for Real-Time ORB-SLAM on FPGA Platform**. in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. 2019.
15. Huang, J., et al., *A new FPGA architecture of FAST and BRIEF algorithm for on-board corner detection and matching*. *Sensors*, 2018. **18**: p. 1014.
16. Wang, G., et al. *A parallel method for aerial image stitching using ORB feature points*. in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. 2017.
17. Group, O.V.G. *Affine Covariant Regions Datasets*. 2007; Available from: <http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html>.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	ทักษพร อิ่มแสงสุข
วัน เดือน ปี เกิด	17 กุมภาพันธ์ 2540
สถานที่เกิด	สุพรรณบุรี
วุฒิการศึกษา	สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2561



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY