การประกอบกลับสามมิติหลายมุมมองจากภาพในร่มและกลางแจ้งด้วยวิธีโมเดลร่วมเอนเซมเบิลจากหลากหลายโครงข่ายประสาทแบบคอนโวลูชัน

MULTI-VIEW STEREO RECONSTRUCTION FROM INDOOR AND OUTDOOR IMAGES USING A CNN-BASED MULTI-MODEL ENSEMBLE METHOD

ภัทรภร วัฒนาชีพ

BHATTARABHORN WATTANACHEEP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2565

KMITL-2022-EN-D-018-072

MULTI-VIEW STEREO RECONSTRUCTION FROM INDOOR AND OUTDOOR IMAGES USING A CNN-BASED MULTI-MODEL ENSEMBLE METHOD

BHATTARABHORN WATTANACHEEP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2022

KMITL-2022-EN-D-018-072

หัวข้อวิทยานิพนธ์	การประกอบกลับสามมิติหลายมุมมองจากภาพในร่มและกลางแจ้งด้วยวิธีโมเดลร่วมเอนเซมเบิลจากหลากหลายโครงข่ายประสาทแบบคอนโวลูชัน
นักศึกษา	นางสาวภัทรภร วัฒนาชีพ
รหัสนักศึกษา	59601021
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2565
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.อรฉัตร จิตต์โสภักตร์

บทคัดย่อ

การประมาณค่าท่าทางของกล้องเป็นกระบวนการสำคัญที่รับรองความสำเร็จของการสร้างแบบจำลองสามมิติ เรานำเสนอการประกอบกลับสามมิติหลายมุมมองจากภาพในร่มและกลางแจ้งด้วยวิธีโมเดลร่วมเอนเซมเบิลจากหลากหลายโครงข่ายประสาทแบบคอนโวลูชันที่สามารถเรียนรู้ได้ในหลายโดเมน รวมถึงภาพจากสภาพแวดล้อมทั้งในร่มและกลางแจ้ง รูปภาพของแต่ละโดเมนมีคุณสมบัติและมุมมองการถ่ายภาพที่แตกต่างกัน จึงนำไปสู่ความยากลำบากในการเรียนรู้อย่างมีประสิทธิภาพในการประมาณค่าท่าทางที่มีความแตกต่างอย่างมาก และต้องใช้ทรัพยากรในการคำนวณจำนวนมาก เพื่อลดความซับซ้อนของโมเดลเดี่ยวแบบเบ็ดเสร็จ ที่พยายามเรียนรู้ความหลากหลายที่ซับซ้อนสูง รูปแบบที่นำเสนอจะถูกแบ่งออกเป็นตัวแทนการเรียนรู้หลายตัวซึ่งประกอบด้วยเอเจนท์เฉพาะโดเมนและเอเจนท์ความสัมพันธ์ระหว่างโดเมน โดยเอเจนต์เฉพาะโดเมนได้รับการฝึกอบรมอย่างเป็นอิสระจากชุดของคุณลักษณะเฉพาะของภาพ ตัวอย่างเช่น ชุดหนึ่งสำหรับชุดข้อมูลในร่มและอีกชุดสำหรับชุดข้อมูลกลางแจ้ง จากนั้นเอเจนต์ความสัมพันธ์ระหว่างโดเมนจะรวบรวมและวิเคราะห์คุณสมบัติของโดเมนหลายรายการและสรุปการประมาณค่าด้วยค่าความผิดพลาดเฉลี่ยกำลังสอง เราเปรียบเทียบประสิทธิภาพของโมเดลเดี่ยวแบบฝึกฝนแยกโดเมน โมเดลการฝึกฝนร่วมหลายโดเมน และโมเดลการเรียนรู้แบบโครงข่ายเอเจนท์เอนเซมเบิลจากหลากหลายโครงข่ายประสาทแบบคอนโวลูชันที่นำเสนอ ผลการทดลองระบุว่าแบบจำลองที่เสนอนั้นมีประสิทธิภาพดีกว่าวิธีอื่นๆ โดยมีข้อผิดพลาดในการทำนายการหมุนและการเลื่อนอยู่ที่ 0.112012266

Thesis Title	MULTI-VIEW STEREO RECONSTRUCTION FROM INDOOR AND OUTDOOR IMAGES USING A CNN-BASED MULTI-MODEL ENSEMBLE METHOD
Student	Ms. Bhattarabhorn Wattanacheep
Student ID	59601021
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2022
Thesis Advisor	Assoc.Prof.Dr. Orachat Chitsobhuk

ABSTRACT

Camera poses estimation is a critical process that ensures the success of Three-Dimensional (3D) modelling. We present a Multi-view Stereo Reconstruction from Indoor and Outdoor Images using a CNN-based Multi-model Ensemble Method capable of learning across multiple domains, including images from both indoor and outdoor environments. Each domain's images have distinct properties and shooting viewpoints, which leads to difficulty in efficient learning such a large difference and requires large amount of computational resources. In order to reduce complexity of the end-to-end single model, the proposed model is divided into multiple learning agents consisting of domain-specific agents and domain relationship agent. The domain-specific agent is trained independently on its own set of unique image characteristics, for example, one for indoor datasets and another for outdoor datasets. The domain relationship agent then ensembles and analyzes the multiple domain features and finalizes the estimation. In terms of average root mean square error, we compare the performance of the single domain separate training, the combined domain single model with the suggested ensemble CNN model. The experimental results indicate that the proposed model outperforms the others, with rotation and translation prediction errors of 0.112012266.

กิตติกรรมประกาศ

คุณงามความดีและประโยชน์ที่ได้จากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่บิดามารดาและครูอาจารย์ที่เคารพรักทุกท่านที่ได้ประสิทธิ์ประศาสน์วิชาความรู้และประสบการณ์ที่ดีอันเป็นประโยชน์แก่ข้าพเจ้า

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงตามวัตถุประสงค์ได้ด้วยความกรุณาของรองศาสตราจารย์ ดร.อรฉัตร จิตต์โสภักตร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งให้คำปรึกษา ข้อชี้แนะและแนวทางในการแก้ปัญหาต่างๆ ทั้งด้านวิชาการ ด้านการดำเนินชีวิต ตลอดจนให้ความรู้ และความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ. ที่นี้

ขอขอบพระคุณบิดามารดาและครอบครัว สำหรับการสนับสนุน การให้โอกาสทางการศึกษาและกำลังใจที่มอบให้ข้าพเจ้ามาโดยตลอด ไม่ว่าช่วงเวลาที่มีความสุขหรือแม้กำลังมีความทุกข์ แม้กำลังท้อแท้สิ้นหวัง หรือแม้ต้องเจอกับปัญหามากมายสักเพียงใด ขอขอบคุณที่ยืนข้างกันเสมอมา

ข้าพเจ้าขอกราบขอบพระคุณครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิประศาสน์วิชาความรู้อีกทั้งถ่ายทอดประสบการณ์อันเป็นประโยชน์ต่างๆ ทั้งโดยตรง และโดยอ้อม ตลอดจนนักวิจัยทุกท่านที่เอื้อเฟื้องานวิจัย จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณคณะกรรมการการสอบที่ให้ความกรุณาในการชี้แนะและแก้ไขข้อบกพร่องต่างๆ ของงานวิจัย

ขอบคุณ พี่ๆ เพื่อน และน้องภาควิชาวิศวคอมพิวเตอร์ทุกคน ที่คอยสอบถามด้วยความห่วงใย ให้ความช่วยเหลือ คำแนะนำ กำลังใจที่ดีตลอดมาและขอขอบคุณพิเศษสำหรับ “พี่แจ๋ว” คุณวงศ์สวรรค์ ศรีมนตรีสง่า “พี่จุ๋ม” ดร.รัตติกร สมบัติแก้ว “พี่กุล” ดร.กุลวลัญช์ วรุณสิน “พี่นพ” คุณนพพล น้อยแก้ว และ “พี่เบน” คุณจตุรนต์ เงินพลับพลา รวมถึงผู้มีพระคุณทุกท่านที่มิได้เอ่ยนามไว้ ณ ที่นี้

ขอบคุณ พี่ๆ น้องๆ ชาวสำนักทะเบียนและประมวลผล สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดหระบัง ที่คอยสอบถามด้วยความห่วงใย ให้ความช่วยเหลือ คำแนะนำ กำลังใจที่ดีตลอดมา

สุดท้ายนี้ ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งเป็นสถานที่ข้าพเจ้าศึกษามาเป็นเวลา 14 ปี (ป.ตรี ถึง เอก)

ภัทรภร วัฒนาชีพ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รายการคำย่อ

คำย่อ	*คำเรียกในภาษาไทย*
CNN	โครงข่ายประสาทเทียมแบบสังวัฒนาการ (Convolutional Neural Network)
2D	สองมิติ (2 Dimension)
3D	สามมิติ (3 Dimension)
LSA	การวิเคราะห์ความหมายแอบแฝง (Latent Semantic Analysis)
SIFT	อัลกอริทึมที่ใช้สำหรับการทำ Feature Detection (Scale Invariant Feature Transform)
SVD	การแยกค่าเอกฐาน (Singular Value Decomposition)
SVM	ซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machine) เป็น Algorithm ที่ใช้สำหรับแก้ปัญหาการจัดกลุ่มข้อมูล (Classification)
SVR	ซัพพอร์ตเวคเตอร์รีเกรชชั่น (Support Vector Regression) เป็น Algorithm ที่ใช้สำหรับแก้ปัญหาการวิเคราะห์การถดถอย (Regression)
RMSE	ค่าเฉลี่ยความผิดพลาดกำลังสอง (Root Mean Square Error)
GPS	ระบบระบุตำแหน่งบนพื้นโลก (Global Positioning System)
LiDAR	อุปกรณ์ที่ใช้แสงเพื่อตรวจจับและคาดคะเนระยะทางของวัตถุ (Light Detection And Raging)
ms	วินาที (millisecond)
m	เมตร (metre)
deg	องศา (degree)
ROI	บริเวณที่น่าสนใจ (ROI)
conv	ชั้นหรือเลเยอร์ Convolution
ReLU	ชั้นหรือฟังก์ชัน Rectified Linear Unit
FC	ชั้นหรือเลเยอร์ Fully Connected
SfM	การสร้างโครงสร้างในสามมิติโดยอาศัยข้อมูลภาพในหลายๆมุมมองจำนวนมาก (Structure from motion)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในงานการสร้างแบบจำลองโมเดลสามมิติที่มีขนาดใหญ่ (generation of largescale) ทั้งในรูปแบบจำลองโมเดลสามมิติในร่ม (indoor) และกลางแจ้ง (outdoor) เป็นเรื่องที่ได้รับความสนใจเป็นอย่างมาก เนื่องจากโมเดลสามมิติให้ข้อมูลและการรับรู้ที่เสมือนจริง ซึ่งเป็นประโยชน์ต่อหลากหลายงานในปัจจุบัน และมีแนวโน้มความต้องการที่เพิ่มขึ้น เพื่อการเข้าถึงข้อมูลในรูปแบบเชิงพื้นที่ที่ทันสมัย โดยแอพพลิเคชันที่สามารถนำไปประยุกต์ใช้ในงาน เช่น การนำเสนอโมเดลมรดกทางวัฒนธรรม [1, 2, 3] การระบุตำแหน่งด้วยรูปภาพ (image-based localization) [4, 5] จากกล้องแบบเรียลไทม์เพื่อใช้ในงานการติดตามบนอุปกรณ์เคลื่อนที่สำหรับ Augmented Reality [6] การให้คำแนะนำการนำทาง [7] การจัดการเหตุฉุกเฉิน [8] การวางแผนการบำรุงรักษา การปรับปรุงอาคาร [9] และบริการต่างๆ เช่น บริการแนะนำการตกแต่งภายในบ้านหรืออาคาร โดยผู้ที่ไม่มีประสบการณ์ด้านการตกแต่งภายในสามารถเรียกดูแผนกเฟอร์นิเจอร์ออนไลน์ซึ่งนำเสนอผลิตภัณฑ์แบบดิจิทัล 3 มิติและบริการความช่วยเหลือจากผู้เชี่ยวชาญหรือระบบผู้เชี่ยวชาญด้านดิจิทัล [10] บริการนำทางในอาคาร โดยสร้างโมเดลภายในอาคารด้วยเทคนิคเชิงเรขาคณิต ของทิศทางการเปิดประตูตำแหน่งของช่องว่างภายในอาคารและความสัมพันธ์แบบโทโพโลยีและคุณลักษณะเชิงความหมายของช่องว่างภายในอาคาร และสภาพแวดล้อม [11] แอพพลิเคชันแสดงภาพโมเดลสามมิติของอพาร์ทเมนต์เพื่อนำเสนอการขายอสังหาริมทรัพย์ [12] การท่องเที่ยวเสมือนจริง [13,14] การนำทางอัตโนมัติ [15] และแอพพลิเคชันช่วยการผลิตภาพยนตร์เพื่อสร้างฉากที่ไม่สามารถถ่ายทำได้ในสตูดิโอด้วยการจำลองสภาพแวดล้อมภายนอก เป็นต้น ตามหลักการแล้วการสร้างแบบจำลองที่สามารถสร้างได้ทั้งแบบจำลองสามมิติในร่มและกลางแจ้งด้วยอัลกอลิทึมเดียวนั้นเป็นสิ่งที่พึงปรารถนา เนื่องจากทำให้สามารถวิเคราะห์ความต่อเนื่องทั้งจากสภาพแวดล้อมภายนอกและภายในได้อย่างมีประสิทธิภาพ สามารถนำไปใช้ในงานที่หลากหลายและสะดวกยิ่งขึ้น เช่น ช่วยให้ผู้ใช้สามารถเข้าสู่อาคารในแบบจำลองเมืองเสมือนจริงได้อย่างราบรื่นแทนการสำรวจเฉพาะภายนอก เช่นเดียวกันกับการเพิ่มความสามารถให้หุ่นยนต์เคลื่อนที่ระหว่างโลกในร่มและกลางแจ้งได้อย่างอิสระและสะดวกยิ่งขึ้น

อย่างไรก็ดีอัลกอลิธึมที่สามารถทำนายมุมหมุนและการเลื่อนของฉากทั้งในร่มและกลางแจ้งเพื่อสร้างแบบจำลองสามมิติด้วยอัลกอลิธึมเดียวเป็นเรื่องที่ท้าทายและมีความยากด้วยเหตุผลหลายประการ ตัวอย่างเช่น ในงานวิจัยที่อาศัยการสร้างจุดเชื่อมต่อระหว่างภาพ มีข้อจำกัดของรูปแบบการถ่าย หากรูปแบบการถ่ายภาพของฉากมีส่วนเชื่อมต่อ (ซ้อนทับ) กันของภาพที่เล็กน้อยหรือจำนวนภาพน้อยต่อการหาจุดเชื่อมต่อระหว่างภาพ อาจส่งผลการสร้างโมเดลสามมิติผิดพลาดได้ ด้วยเหตุนี้จึงต้องใช้ความระมัดระวังเป็นอย่างยิ่งในการบันทึกข้อมูลเพื่อให้แน่ใจว่ามีการเหลื่อมกันของภาพเพียงพอสำหรับการจับคู่คุณสมบัติและเพื่อป้องกันไม่ให้โมเดลถูกตัดการเชื่อมต่อ [16] ปัญหานี้มักจะรุนแรงขึ้นเนื่องจากความจริงที่ว่าสภาพวะแสงที่อาจมีการเปลี่ยนแปลงอย่างมาก ในทางกลับกันแม้ว่าเราจะถ่ายภาพได้เพียงพอที่จะเชื่อมต่อด้วยสายตาในร่มและกลางแจ้ง แต่ก็ยากที่จะควบคุมและป้องกันไม่ให้เกิดปริมาณจุดทับซ้อนกันที่ไม่เพียงพอต่อการประกอบ หรือแม้กระทั่งปัญหาการถ่ายทำไม่เป็นไปตามลำดับความต่อเนื่องของภาพในฉาก

ดังนั้นในบทความนี้จึงนำเสนอการสร้างแบบจำลองโมเดลสามมิติ ที่ลดผลกระทบของจำนวนจุดเชื่อมต่อระหว่างภาพและรองรับการวิเคราะห์ฉากภายนอกและภายในอาคาร ด้วยการใช้เทคนิคการเรียนรู้เชิงลึกแบบ Ensemble Convolutional Neural Network (CNN) เพื่อทำนายมุมหมุนและการเลื่อนในสามมิติจากภาพสองมิติแบบหลายกล้อง โดยอาศัยการ transfer learning ของคุณลักษณะที่ได้จากโมเดลการเรียนรู้เชิงลึกต้นแบบ มาวิเคราะห์เพิ่มเติมด้วยการเรียนรู้เชิงลึกที่นำเสนอ โดยเริ่มจากการนำภาพตัวอย่างที่ถูกแบ่งออกเป็นชุดข้อมูลฝึกฝนและชุดข้อมูลทดสอบมาสกัดคุณลักษณะด้วยโมเดลการเรียนรู้เชิงลึกต้นแบบ VGG19 จากนั้นนำคุณลักษณะที่ได้ มาประยุกต์ใช้ทำนายมุมหมุนและการเลื่อนสามมิติของภาพ โดยจะทำการลดมิติข้อมูลด้วยเทคนิค Latent Semantic Analysis (LSA) เพื่อลดให้เหลือเพียงมิติของคุณลักษณะที่สำคัญ ก่อนนำเข้าโมเดลการเรียนรู้เชิงลึกที่นำเสนอในรูปแบบของ Ensemble CNN ซึ่งประกอบด้วยเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) และเอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมน (domain relationship agents) จากที่ได้กล่าวมาภาพทั้งหมดจะถูกนำมาพิจารณาพร้อมกันอย่างเท่าเทียมกัน ดังนั้นผลลัพธ์ที่ได้จะไม่ขึ้นอยู่กับลำดับภาพถ่ายที่ได้รับการพิจารณา จึงทำให้ประสิทธิภาพในการวิเคราะห์มุมหมุนและการเลื่อนมีความถูกต้องสูงขึ้น นอกจากนี้การฝึกฝนแบบโมเดลเดี่ยวและโดเมนร่วมช่วยลดความแปรปรวนและการโน้มเอียงของชุดข้อมูลที่ฝึกฝน ทำให้ช่วยเพิ่มประสิทธิภาพการประมาณค่ามุมหมุนและการเลื่อนให้ดียิ่งขึ้นอีกด้วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 ศึกษาเทคนิคและการประมาณค่าพารามิเตอร์สำหรับการประกอบกลับสามมิติจากภาพสองมิติ

1.2.2 ปรับปรุงวิธีการโดยอาศัยสมมติฐาน ความเข้าใจ และทฤษฎีอื่นประกอบการทดลอง

1.2.3 ทดลองการประมาณค่าพารามิเตอร์สำหรับการประกอบกลับสามมิติจากภาพสองมิติที่นิยมใช้ในการประกอบกลับจริง เพื่อทดสอบความเป็นไปได้และวัดประสิทธิภาพในการนำไปประยุกต์ใช้งานจริง

1.2.4 เพื่อเป็นแหล่งข้อมูลอ้างอิงสำหรับผู้สนใจงานวิจัยด้านนี้ต่อไป

1.3 สมมติฐานของการศึกษา

ในขั้นตอน Data preprocessing เป็นขั้นตอนการหาคุณลักษณะที่สำคัญของชุดภาพ อย่างไรก็ดีการหาคุณลักษณะร่วมของภาพเพื่อนำมาหามุมหมุนและการเลื่อนด้วยวิธี SIFT มีข้อจำกัดในส่วนของภาพที่นำมาประกอบ หากไม่มีความต่อเนื่อง มีความเบลอ หรือความสว่างไม่เพียงพอ จะส่งผลให้การหาคุณลักษณะร่วมของคู่ภาพมีความผิดพลาดสูง นอกจากนี้การประมาณค่ามุมหมุนและการเลื่อนของวิธี SIFT เป็นการนำภาพแต่ละคู่มาพิจารณาหามุมหมุนและการเลื่อนร่วมกันทีละคู่ภาพต่อๆกัน ส่งผลให้ลำดับการสุ่มภาพที่นำมาพิจารณามีผลต่อการทำนายและหากมีการประมาณค่ามุมหมุนคู่ใดคู่นึงผิดจะส่งผลเป็นค่าความผิดพลาดสะสมไปยังคู่ภาพที่ถูกนำมาพิจารณาลำดับถัดๆไป นอกจากวิธีที่กล่าวมาข้างต้นการประมาณท่าทางของกล้องด้วยการเรียนรู้ CNN เป็นวิธีที่นิยม เนื่องจากทำให้ค่าความผิดพลาดของการประมาณค่ามุมหมุนและการเลื่อนน้อยลง แต่การฝึกฝนให้โมเดลมีความแม่นยำนั้น ข้อมูลอินพุทที่นำมาฝึกฝนต้องมีปริมาณมากและระยะเวลาในการฝึกฝนมีความยาวนาน

จากที่กล่าวมาข้างต้นเพื่อลดข้อจำกัดภาพที่ไม่มีความต่อเนื่อง มีความเบลอ มีความสว่างไม่คงที่ ลำดับรูปภาพที่มีผลต่อประสิทธิภาพการทำนายและส่งผลให้เกิดค่าความผิดพลาดสะสมตามลำดับของภาพที่ไม่ต่อเนื่อง งานวิจัยที่นำเสนอจึงนำเสนอการเรียนรู้เชิงลึกแบบ Ensemble CNN ซึ่งทำการ transfer learning จากโมเดลการเรียนรู้เชิงลึกเพื่อดึงคุณลักษณะของภาพ เพื่อให้ได้คุณลักษณะที่ครอบคลุมหลากหลายลักษณะรูปภาพและวัตถุในภาพ และยังประหยัดเวลาในการฝึกฝนโมเดลสำหรับปริมาณข้อมูลจำกัด คุณลักษณะที่ได้นี้ถูกนำมาลดมิติข้อมูลด้วยเทคนิค LSA ก่อน

นำเข้าทำนายมุมหมุนและการเลื่อน โดย Ensemble CNN ที่นำเสนอ จะประกอบด้วยเอเจนท์เฉพาะโดเมนที่ถูกฝึกฝนด้วยข้อมูลแต่ละโดเมน ซึ่งมีความแตกต่างทั้งรูปแบบการถ่ายภาพในร่มและกลางแจ้ง มุมมองการถ่ายภาพ และสภาพแวดล้อม จากนั้นผลการทำนายของเอเจนท์เฉพาะโดเมนมาเชื่อมโยงความสัมพันธ์ด้วยเอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมน ด้วยโครงสร้างของโมเดลเชิงลึกแบบ Ensemble CNN ที่นำเสนอช่วยให้สามารถทำนายค่ามุมหมุนและการเลื่อนของชุดข้อมูลทั้งในร่มและกลางแจ้งได้พร้อมกัน และโมเดลที่นำเสนอแสดงให้เห็นประสิทธิภาพการทำนายชุดข้อมูลที่ไม่เคยถูกฝึกฝนด้วยค่าความผิดพลาดน้อยลงอีกด้วย จะเห็นได้ว่าการฝึกฝนโมเดล CNN ที่เรียนรู้เฉพาะทาง และโมเดลที่เรียนรู้ข้ามโดเมน ส่งผลให้ผลลัพธ์ที่ทำนายได้ยืดหยุ่นและไม่เอนเอียงต่อการเรียนรู้เฉพาะทางในโดเมนใดโดเมนหนึ่ง

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย

วิธีการประมาณค่ามุมหมุนและการเลื่อนของชุดภาพถ่ายในร่มและกลางแจ้งในงานวิจัยนี้อาศัยทฤษฎีและหลักการดังต่อไปนี้

1.4.1 การแปลงภาพจากโลกสามมิติเป็นสองมิติด้วยกล้อง (From Camera Coordinates to World Coordinates)

1.4.2 เทคนิค SIFT (Scale Invariant Feature Transform)

1.4.3 เรขาคณิต epipolar (epipolar geometry)

1.4.4 การเรียนรู้เชิงลึก Deep learning (VGG19) ถูกนำมาประยุกต์ใช้กับการดึงคุณลักษณะในขั้นตอน Preprocessing data

1.4.5 การแตกค่าแบบเอกฐาน (Singular Value Decomposition : SVD) ถูกนำมาประยุกต์ใช้กับการลดคุณลักษณะที่สกัดได้ให้เหลือแต่คุณลักษณะที่สำคัญ

1.4.6 การถ่ายโอนความรู้ Transfer learning CNN ถูกนำมาใช้สกัดคุณลักษณะของภาพถ่ายสองมิติโดยใช้ความรู้และค่าถ่วงน้ำหนักของโมเดลที่เคยถูกฝึกฝนด้วยชุดภาพอื่นๆ

1.4.7 ซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machine: SVM) ถูกนำมาประยุกต์ใช้กับการประมาณค่าพารามิเตอร์การวางท่าของกล้อง

1.4.8 CNN ถูกนำมาประยุกต์ใช้ประมาณค่าพารามิเตอร์การวางท่าของกล้อง

1.4.9 Ensemble CNN ถูกนำมาประยุกต์ใช้ประมาณค่าพารามิเตอร์การวางท่าของกล้อง

1.5 ขอบเขตงานวิจัย

วิทยานิพนธ์นี้ได้ทำการศึกษา และพัฒนาวิธีการสร้างโมเดลสำหรับการประมาณค่ามุมหมุนและการเลื่อนที่ใช้ในการประมาณค่าพิกัดสามมิติจากภาพในร่มและกลางแจ้งสองมิติ ซึ่งมีขอบเขตการวิจัยดังนี้

1.5.1 ลักษณะการถ่ายรูปในร่มและกลางแจ้งแบบไม่เป็นแพทเทิร์น โดยการถ่ายด้วยการวางท่าของกล้องที่หลากหลาย ไม่ขึ้นกับสภาพแสง ขนาดการซ้อนทับ ความต่อเนื่องของภาพ และชนิดของกล้อง

1.5.2 ชุดข้อมูลที่ใช้ในวิทยานิพนธ์นี้เป็นชุดข้อมูลภาพที่มีงานวิจัยในอดีตนิยมนำมาใช้วัดประสิทธิภาพ เนื่องจากเป็นชุดข้อมูลที่ให้คำตอบ (groundtruth) คือ พารามิเตอร์มุมหมุนและการเลื่อนของแต่ละภาพที่หลากหลายและครบถ้วน ประกอบด้วยชุดข้อมูลในร่ม (Dome) 1 แหล่งที่มา และชุดข้อมูลกลางแจ้ง (Map และ Cambridge) จาก 2 แหล่งที่มา เพื่อทำการทดสอบและวัดประสิทธิภาพเชิงเปรียบเทียบกับงานวิจัยอ้างอิง

1.5.3 การทดลองเพื่อเปรียบเทียบประสิทธิภาพของการประมาณค่ามุมหมุนและการเลื่อนกับงานวิจัยที่มีผู้ทดลองไว้ในลักษณะเดียวกัน โดยชุดข้อมูลกลางแจ้งจะนำมาเปรียบเทียบประสิทธิภาพผลลัพธ์กับวิธีโมเดล 4.3,4.4 และงานวิจัยอ้างอิง [23] ชุดข้อมูลในร่มชุดที่ 1 (Map) ถูกนำมาเปรียบเทียบกับวิธีโมเดล 4.3,4.4 และชุดข้อมูลในร่มชุดสุดท้าย (Cambridge) ถูกนำมาวัดประสิทธิภาพกับโมเดล 4.3, 4.4 และงานวิจัยอ้างอิง [53 - 61]

1.5.4 โดยโมเดลที่งานวิจัยนี้นำเสนอ มีความยืดหยุ่นของโมเดลที่สามารถประมาณค่ามุมหมุนและการเลื่อนของชุดภาพสองมิติในร่มและกลางแจ้งพร้อมกันในโมเดลเดียว

1.6 ขั้นตอนการศึกษา

1.6.1 กำหนดวัตถุประสงค์และขอบเขตของงานวิจัย ว่าต้องการศึกษางานวิจัยในหัวข้อนี้อย่างไร

1.6.2 ศึกษาวิธีการทำนายพารามิเตอร์มุมหมุนและการเลื่อนที่เคยมีผู้ทำไว้ก่อนหน้านี้ เปรียบเทียบข้อดีและข้อจำกัดของแต่ละวิธีการ เพื่อนำมาปรับปรุงและประยุกต์ใช้ให้เกิดประโยชน์กับงานวิจัยนี้

1.6.3 ตั้งสมมติฐานตามทฤษฎีหรือวิธีการที่ได้ศึกษาและกำหนดแนวทางในการวิจัย

1.6.4 เตรียมฐานข้อมูลงานวิจัยเพื่อนำมาใช้ทดลองในงานวิจัย โดยนำชุดข้อมูลภาพที่มีการเก็บพารามิเตอร์มุมหมุนและการเลื่อนของกล้องและเป็นที่นิยมมาใช้ในการวัดค่าความผิดพลาดที่เกิดจากการประมาณค่าของงานวิจัยอ้างอิง

1.6.5 นำชุดข้อมูลที่เตรียมไว้มาทำการทดลอง โดยใช้โปรแกรมที่ได้พัฒนาขึ้น ทำการเก็บข้อมูลและผลลัพธ์ที่ได้จากการทดลอง เพื่อวิเคราะห์และวัดประสิทธิภาพ เพื่อแนะนำการปรับปรุงเป็นแนวทางการพัฒนางานวิจัยต่อไป

1.6.6 ทำการสรุปผลการทดลองโดยเปรียบเทียบผลการทดลองในด้านความถูกต้องกับการทดลองในงานวิจัยอ้างอิง

1.7 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัย มีดังนี้

1.7.1 เครื่องคอมพิวเตอร์ส่วนบุคคลใช้หน่วยความจำ Intel CORE i7 + GPU จำนวน 3 เครื่อง

1.7.2 ระบบปฏิบัติการ Windows 10

1.7.3 โปรแกรม Matlab2019

1.7.4 โปรแกรม Jupyter

1.8 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท แต่ละบทประกอบด้วยเนื้อหาดังนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา สมมติฐานของการศึกษา ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย ขอบเขตงานวิจัย ขั้นตอนการศึกษา และเครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

บทที่ 2 กล่าวถึงงานวิจัยที่เกี่ยวข้องกับการประกอบกลับสามมิติ ในร่มและกลางแจ้ง รวมทั้งข้อดีและข้อจำกัดของแต่ละวิธี

บทที่ 3 กล่าวถึงความรู้พื้นฐานในเรื่องที่เกี่ยวข้องกับงานวิจัยได้แก่ การแปลงภาพจากโลกสามมิติเป็นสองมิติด้วยกล้อง (From Camera Coordinates to World

Coordinates), การสกัดคุณลักษณะด้วยวิธี SIFT (Scale Invariant Feature Transform), เรขาคณิต epipolar (epipolar geometry), การแตกค่าแบบเอกฐาน (Singular Value Decomposition : SVD), การทำซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machine: SVM), การเรียนรู้เชิงลึก (Deep learning), การถ่ายโอนความรู้ (Transfer learning), Ensemble CNN, การหาค่าความผิดพลาดเฉลี่ยกำลังสอง (Avereage RMSE)

บทที่ 4 กล่าวถึงภาพรวมและขั้นตอนการดำเนินการในงานวิจัยนี้ โดยมีขั้นตอนการทำ Data Preprocessing (Transfer Feature (VGG19) + LSA) ของงานวิจัยนี้ เริ่มจากการนำภาพมาสกัดคุณลักษณะและทำการลดมิติของคุณลักษณะให้หลือแต่คุณลักษณะที่สำคัญ จากนั้นเข้าสู่ขั้นตอนการประมาณค่าพารามิเตอร์ที่ใช้ในการประกอบภาพสามมิติด้วยโมเดลต่างๆ (3D reconstruction parameter estimation)

บทที่ 5 ผลการทดลองและการวิเคราะห์การวัดประสิทธิภาพ โดยการหาค่าความผิดพลาดเฉลี่ยกำลังสอง(ด้วยชุดข้อมูล Dome และ Map) และการหาค่าความผิดพลาดมัธยฐาน (ของชุดข้อมูล Cambridge)

บทที่ 6 สรุปผลการทดลอง และแนวทางในการพัฒนา

บทที่ 2

งานวิจัยที่เกี่ยวข้อง

การแปลงภาพจากโลกสามมิติเป็นสองมิติของการสร้างโมเดลกล้องด้วยเมทริกซ์ที่มีคุณสมบัติเฉพาะ โดยการแปลงตำแหน่งสามมิติในโลกจริงลงบนตำแหน่งภาพสองมิตินั้น เริ่มจากกล้องแปลงตำแหน่งสามมิติ (Scene) ลงบนจอแสดงภาพเป็นภาพถ่ายสองมิติ (Image) ด้วยการประมวลผลสีและแสงผ่านเซนเซอร์ระดับพิกเซลล์ (pixel) ภายในกล้อง ซึ่งการแปลงสามมิติเป็นสองมิติมี 2 ขั้นตอนหลัก คือการหาพารามิเตอร์ภายในกล้อง (Intrinsic camera parameters) ซึ่งเป็นการแมพตำแหน่งในโลกสามมิติลงบนกล้องไปยังพิกเซลภายในกล้อง (map camera coordinate to pixel coordinate) ที่มีความเฉพาะของรูปแบบการเกิดภาพของกล้อง (general projective camera) และพารามิเตอร์ภายนอกกล้อง (Extrinsic camera parameters) เป็นพารามิเตอร์แสดงการหมุนและการเลื่อนของกล้อง โดยพารามิเตอร์เหล่านี้มีความสัมพันธ์กันในเชิงเรขาคณิต (projective geometry) ภายในของกล้อง เนื่องจากพารามิเตอร์ภายในกล้องสามารถหาได้จากข้อมูลที่มากับรูปภาพ ดังนั้นพารามิเตอร์ที่สำคัญและเป็นที่นิยมนำมาสร้างเทคนิคสำหรับการประมาณค่าเพื่อใช้ในการสร้างแบบจำลองสามสามมิติให้มีความถูกต้องคือ การหาพารามิเตอร์ภายนอกกล้องประกอบด้วยมุมหมุนและการเลื่อนที่ในแนวแกนเอ็กซ์ แกนวาย และแกนแซด สามารถถูกประมาณค่าได้หลากหลายวิธี เช่นการประมาณค่ามุมหมุนและการเลื่อนแบบกล้องเดี่ยว (single camera) การประมาณค่ามุมหมุนและการเลื่อนแบบกล้องคู่ (stereo camera) และการประมาณค่ามุมหมุนและการเลื่อนแบบหลายกล้อง (multiple camera)

วิธีการสมัยใหม่ (state-of-the-art) ส่วนใหญ่อาศัยคุณลักษณะในพื้นที่ (local feature) เช่น SIFT [17] เพื่อแก้ปัญหาของ image-based localization จากโมเดล SfM ของฉากที่แต่ละจุดสามมิติมีความเกี่ยวเนื่องกันด้วยคุณลักษณะของภาพ เช่น การหามุมของวัตถุภายในภาพ การใช้เรขาคณิตที่เกี่ยวกับงานทางด้านคอมพิวเตอร์วิชัน (geometric computer vision tasks) และการประมาณการวางท่าของกล้อง (Camera pose) มาช่วยในการประมาณพารามิเตอร์โครงสร้างแบบจำลองสามมิติ โดยการหาจุดคุณลักษณะ (feature points) เป็นวิธีสำหรับการหาความสอดคล้องกันของภาพซึ่งขึ้นอยู่กับการตรวจจับคุณสมบัติ การสร้างการจับคู่สองมิติกับสามมิติระหว่างคุณลักษณะที่ดึงมาจากการประมวลผลภาพและจุดสามมิติในโมเดล SfM นิยมใช้ RANSAC มาช่วย การประมาณการวางท่า

การประมาณค่าจุดในสามมิติจะประสบความสำเร็จถ้าการจับคู่ที่พบในขั้นตอนแรกมีความถูกต้อง ดังนั้นข้อจำกัดของการจับคู่ด้วยการตรวจจับคุณลักษณะ เช่นภาพที่เบลอ มีการเคลื่อนไหว หรือแสงเปลี่ยนจะส่งผลให้ขั้นตอนข้างต้นล้มเหลว เมื่อการประมาณจุดคุณลักษณะไม่เพียงพอต่อการจับคู่หรือมีตำแหน่งผิดพลาดไป ทำให้ขาดความสามารถในการหาจุดร่วมคุณลักษณะระหว่างภาพให้เพียงพอต่อการทำนายมุมหมุนและการเลื่อน ส่งผลให้ความแม่นยำต่ำ ซึ่งปัญหานี้สามารถจัดการได้ด้วยวิธี end-to-end learning.

2.1 การประมาณท่าทางกล้องจากงานวิจัยอ้างอิงที่ใช้ SIFT และ RANSAC

การหาโครงสร้างจากข้อมูลการเคลื่อนที่หรือเทคนิคเอสเอฟเอ็ม (Structure from motion; SfM) จากชุดภาพถ่ายทางเว็บไซต์สื่อสังคมออนไลน์ เช่น ใน Flickr [18-21] ส่วนใหญ่จะทำการหาคุณลักษณะที่เหมือนกันจากภาพหลายๆภาพด้วยการทำ feature matching เช่น งานวิจัยของเซียวเหว่ย ลิและคณะ (Xiaowei Li et al.) [20] ขั้นแรกมีการจัดกลุ่มของรูปภาพโดยการหา ‘ภาพสัญลักษณ์’ (iconic images) เพื่อทำการคำนวณโครงสร้างคุณลักษณะพื้นฐานของการหาเส้นทาง (spanning tree) ความสัมพันธ์ของภาพ จากนั้นหามุมหมุนและการเลื่อนด้วยเทคนิคเอสเอฟเอ็ม เพื่อนำมุมหมุนและการเลื่อนที่ได้จากขั้นตอนนี้มาประกอบเป็นสามมิติ ซึ่งวิธีจากที่ได้กล่าวมาถูกนำเสนอในงานวิจัยของสินหาและคณะ (Sinha et al.) [22] ด้วยวิธีเอสเอฟเอ็มเชิงเส้นเช่นเดียวกัน โดยเริ่มต้นการประกอบจากจุดกำเนิดขนาดเล็กที่มี จากนั้นทำการปรับปรุงด้วยการวนซ้ำเพื่อเพิ่มจำนวนภาพ ในขั้นตอนนี้ผลการหามุมหมุนและการเลื่อนระหว่างภาพจะถูกปรับปรุงให้ดีขึ้น ในขณะที่เทคนิคการดำเนินการแบบค่อยเป็นค่อยไปนี้ประสบความสำเร็จอย่างมาก แต่ยังมีข้อบกพร่องอยู่ 2 ประการที่สำคัญ ประการแรกคือวิธีการเหล่านี้มีแนวโน้มที่จะต้องการการคำนวณอย่างหนัก เนื่องจากจำเป็นต้องทำการวนซ้ำสำหรับการหาค่าเหมาะสมแบบไม่เชิงเส้นเพื่อหาความสัมพันธ์ร่วมระหว่างพารามิเตอร์กล้องและโครงสร้างฉากให้ดีขึ้น ประการที่สองเนื่องจากวิธีการเหล่านี้ไม่พิจารณาภาพทั้งหมดอย่างเท่าเทียมกันจึงก่อให้เกิดผลที่แตกต่างกัน ซึ่งผลลัพธ์ที่ได้จะขึ้นอยู่กับลำดับภาพถ่ายที่ได้รับการพิจารณา สิ่งนี้สามารถนำไปสู่ความล้มเหลวในการประกอบภาพ เนื่องจากผลลัพธ์มีโอกาสเกิดความผิดพลาดจากลำดับการเรียงตัวของกล้องที่ไม่ถูกต้อง ซึ่งวิธีการนี้ยังทำให้การประมาณค่ามุมหมุนและการเลื่อนใช้ระยะเวลานานและมีค่าความผิดพลาดมาก

ต่อมาหยุ่ยฮี (yihui-he) [23] ใช้เทคนิคเอสเอฟเอ็ม เช่นเดียวกัน เริ่มจากการใช้ SIFT หา feature point ที่สอดคล้องกันระหว่างคู่ภาพ จากนั้นใช้เทคนิค MSAC เข้ามาช่วยกรอง feature

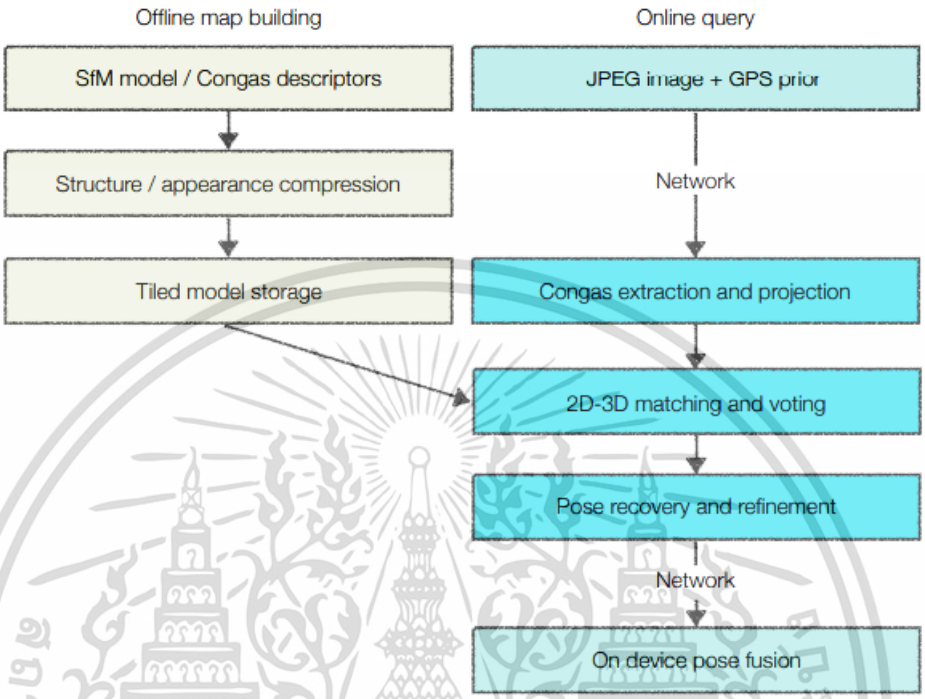
point ที่ไม่สอดคล้องกับภาพต้นฉบับออก แทนเทคนิค RANSAC ทำให้ได้การประมาณการวางตัวของกล้องถูกต้องและแม่นยำมากยิ่งขึ้น ถึงแม้ว่าวิธีการนี้จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพดีขึ้น แต่ยังคงโครงสร้างการคำนวณแบบวนซ้ำทำให้ไม่สามารถลดภาระการคำนวณสูงได้ และลำดับการนำเข้าภาพยังคงส่งผลต่อประสิทธิภาพการประมาณค่าอยู่ ทำให้ยังไม่สามารถแก้ปัญหาข้างต้นที่กล่าวมาได้

2.2 งานวิจัย Large-scale, real-time visual-inertial localization revisited [24]

งานวิจัยอ้างอิงของไซมอน ไลเนนและคณะ (Simon Lynen and et al.) แบ่งเป็นรูปแบบ online และ offline โดยระบบ offline จุดสามมิติจะถูกสร้างขึ้นจากชุดของรูปภาพในฐานข้อมูลด้วยวิธี SfM ในการสเกลพื้นที่ที่มีขนาดใหญ่พวกเขาแบ่งโมเดลจุดสามมิติออกเป็นส่วนย่อยๆ (subparts) ที่คลอบคลุมพื้นที่ประมาณ 2-3 บล๊อคถนน (street-blocks) หุ่นยนต์หรือโทรศัพท์มือถือต้องระบุตำแหน่งทางภูมิศาสตร์เพื่อใช้วิธี SLAM สำหรับติดตามการเคลื่อนไหวพิกัดพื้นที่ของกล้อง การติดตามตำแหน่งให้สัญญาณแบบเรียลไทม์สำหรับการควบคุมหรือการแสดงผล และทำงานโดยใช้ทรัพยากรบนอุปกรณ์เท่านั้นไม่ขึ้นกับเซิร์ฟเวอร์ ขั้นตอนของงานวิจัยอ้างอิงนี้แสดงดังรูปที่ 2.1

สำหรับเฟรม (frame) ย่อยๆที่ถ่ายโดยกล้องของอุปกรณ์ คุณลักษณะภาพจะถูกดึงออกมาและตัวบ่งชี้ (descriptors) จะถูกจับคู่กับตัวบ่งชี้ของระบบสร้างจุดในสามมิติที่สร้างไว้ก่อนหน้านี้ ขั้นตอนการจับคู่นี้เกิดขึ้นบนเซิร์ฟเวอร์ (server) โดยส่วนย่อยที่เกี่ยวข้องของโมเดลจะถูกเลือกตามตำแหน่งโดยประมาณจากสัญญาณ GPS/WiFi เมื่อโหลดโมเดลแล้ว อัลกอริธึมการจับคู่ที่มีประสิทธิภาพจะระบุจุดสามมิติเหล่านั้นในโมเดลที่เป็นวัตถุเดียวกันกับที่มองเห็นได้ในรูปภาพการสืบค้น ผลลัพธ์การหาความสัมพันธ์จากจุดในสองมิติไปยังสามมิติจะถูกกรองครั้งแรกจากเงื่อนไขทางเรขาคณิต (geometric constraints) และต่อมาใช้ RANSAC (Fischler และ Bolles 1981) เพื่อประเมินท่าทางกล้องส่วนกลาง (global) เนื่องจากการทำ localization มีเวลาแฝงสูงเกินไปสำหรับการประมวลผลแบบเรียลไทม์ จึงป้อนผลลัพธ์การ localization ในการติดตามท่าทางกล้อง (camera pose) ที่คำนวณสำหรับเฟรมแรกให้เป็นตำแหน่งและมุมหมุนของระบบมือถือเริ่มต้น ท่าทางกล้องนี้จึงถูกยึดเป็นเฟรมอ้างอิงภายใน (local) ของระบบ SLAM กับพิกัดส่วนกลาง (global) ของโมเดล (model) สำหรับเฟรมที่ถูกระบุตำแหน่งถัดมาทั้งหมด inliers สำหรับการประมาณท่าทางจะรวมเข้ากับตัวประมาณสถานะ (state estimator) ของระบบ SLAM เพื่อให้มีเงื่อนไขเพิ่มเติม (นอกเหนือจากคุณลักษณะที่ติดตามโดย SLAM ในภายใน (local)) เงื่อนไขเพิ่มเติมเหล่านี้ปรับปรุงการวางแนว

(alignment) ของแนววิถี (trajectory) ภายในที่ขัดกับโมเดลส่วนกลางและแก้ไขการเลื่อนของการติดตามภายใน [6-7,25-26]



รูปที่ 2.1 ขั้นตอนการประมาณท่าทางของงานวิจัยอ้างอิงของ Simon Lynen และ คณะ [24]

งานวิจัยอ้างอิงประเมิณโมเดลที่ถูกสร้างสำหรับเมือง ปารีส โตเกียว ซูริช และซานฟรานซิสโกโดยใช้กล้องที่ถ่ายด้วยโรลลิ่งชัตเตอร์ (rolling-shutter) 7 หรือ 15 ตัวติดตั้งบนรถยนต์และเป้สะพายหลัง ท่าทางกล้องและการคำนวณโครงสร้างยังต้องใช้อินพุต GPS, LiDAR, การวัดระยะทางและเซ็นเซอร์วัดความเร่ง (inertial sensors คือเซ็นเซอร์วัดความเร่งและความเร็วเชิงมุมของวัตถุหนึ่งตามแกนตั้งฉากกันสามแกน) [29] เพื่อสร้างสภาพแวดล้อมของโมเดลส่วนกลาง โมเดลถูกแบ่งย่อยเป็นชิ้นส่วนขนาดประมาณ 150x150 ม. ซึ่งสะดวกต่อการจัดเก็บและนำมาใช้ (load)

การได้ข้อมูลที่ถูกบันทึกด้วยโทรศัพท์โดยเลียนแบบลักษณะการถ่ายของผู้ใช้ทั่วไปและรูปแบบของหุ่นยนต์นำทาง ลำดับของภาพที่ถูกบันทึกจากการเดินข้างทางมุมมองของโทรศัพท์จะเป็นการถ่ายแบบมุมกดลง (facing downwards) ของเส้นทางการเดินหรือข้ามถนน มุมมองที่เป็นผลลัพธ์จึงแตกต่างอย่างมากจากมุมมองที่มีอยู่ในฐานข้อมูล เนื่องจากมุมมองในฐานข้อมูลเหล่านี้ถูกถ่ายโดยรถยนต์บนท้องถนน ซึ่ง ground truth ของท่าทางที่เอาไว้อ้างอิงสำหรับลำดับการประเมินของการ

สร้างการจับคู่สองมิติไปยังสามมิติของแต่ละเฟรมประเมินกับโมเดลสามมิติโดยอัตโนมัติ จะแสดงข้อผิดพลาดเป็นเมตรและองศา

พวกเขาประเมินคุณภาพการติดตามการวางท่าทาง (pose tracking) การจับคู่จากภาพสองมิติไปยังตำแหน่งในสามมิติ โดยสนใจมุมหมุนและการเลื่อนที่ได้จากระบบของพวกเขา เปรียบเทียบกับอัลกอริทึมที่เสนอโดย [6] ซึ่งใช้ SLAM ในภายใน (local) โดยใช้หน้าต่างบานเลื่อน BA (sliding window BA) ซึ่งปรับให้เหมาะสมเฉพาะพารามิเตอร์ภายในและให้โมเดลส่วนกลางคงที่ (the global model fixed)

วิธีการของ [6] และ [30] ขั้นแรกคำนวณการวางแนวเริ่มต้นโดยใช้ท่าทางกล้องที่ส่งกลับจากเซิร์ฟเวอร์หรือตำแหน่งจุดสังเกต (landmark) ส่วนกลาง สำหรับเฟรมต่อไปทั้งหมดที่ส่งไปยังเซิร์ฟเวอร์ เฟรมเหล่านั้นจะปรับการวางแนวให้เหมาะสม โดยรวมการจับคู่สองมิติกับพิกัดสามมิติที่ส่วนกลางไว้ใน bundle adjustment ของการจับคู่ภายใน เพื่อจำกัดความซับซ้อนในการคำนวณ พวกเขาดำเนินการขั้นตอน SLAM โดยการทำ sliding window เพื่อจำกัดจำนวนกล้อง กล้องที่อยู่ไกลจากการวางท่าปัจจุบันมากที่สุด [6] หรือเก่าที่สุด [30] ถูกทิ้งออกพร้อมกันในเงื่อนไขระดับภายในและส่วนกลาง อย่างไรก็ตามการไม่นำภาพเหล่านั้นมาร่วมประมวลผล พบว่านำไปสู่ประสิทธิภาพการประมาณค่าที่ต่ำกว่ามาตรฐาน [31-33] การนำออกของเงื่อนไขเพื่อให้มีประสิทธิภาพนี้ยังคงนำไปสู่ความไม่ต่อเนื่องในการประมาณการของผลลัพธ์การวางท่าทางของกล้อง เพื่อปรับปรุงประสิทธิภาพของการนำข้อมูลอ้างอิงไปใช้และช่วยให้มีการเปรียบเทียบที่ยุติธรรม พวกขาได้รวมเงื่อนไข IMU ไว้ใน bundle adjustment และแก้ปัญหาแบบไม่เชิงเส้น (non-linear) ตารางที่ 2.1 เปรียบเทียบวิธีการของงานวิจัยอ้างอิงกับแนวทางของ [6] สำหรับเฟรมของกล้องทุกเฟรมจะถูกประเมินข้อผิดพลาดระหว่างการวางท่าทางกับ ground truth เป็นระยะห่างแบบยุคลิด (Euclidean distance) ระหว่างการเลื่อนและมุมหมุนที่ต่างทิศทาง จากตาราง วิธีการที่ใช้ EKF ของงานวิจัยอ้างอิงมีความแม่นยำของตำแหน่งที่คล้ายกับแนวทาง bundle adjustment ที่มีประสิทธิภาพดีที่สุด ในขณะที่ให้การประมาณการมุมหมุนกล้องที่แม่นยำยิ่งขึ้น

ตาราง 2.1 การเปรียบเทียบ ค่าเฉลี่ยเวลาที่ต้องใช้ในการอัปเดตตัวประมาณตามค่าการจับคู่ 2D-3D ของระดับส่วนกลาง (t-up [ms]), ความผิดพลาดของการเลื่อนเฉลี่ย ($\|\bar{p}_{err}\|$ [m]) และข้อผิดพลาดมุมหมุน ($\|\bar{\theta}_{err}\|$ [deg]) ของวิธี EKF ที่งานวิจัยอ้างอิงนำเสนอเปรียบเทียบกับวิธีอื่นๆ

	t-up [ms]	$\|\bar{p}_{err}\|$ [m]	$\|\bar{\theta}_{err}\|$ [deg]
EKF	2.9 ± 1.5	0.17 ± 0.12	0.32 ± 0.16
BA-10-10	163.0 ± 43.0	0.13 ± 0.15	0.41 ± 0.17
BA-10-5	138.8 ± 36.	0.12 ± 0.14	0.58 ± 0.15
BA-5-10	100.3 ± 31.9	0.11 ± 0.13	0.53 ± 0.15
BA-5-5	77.6 ± 31.6	0.12 ± 0.14	0.57 ± 0.10
BA-5-2	38.6 ± 14.4	0.14 ± 0.16	0.54 ± 0.16

แม้ว่าทั้งสองวิธีจะให้ความแม่นยำของท่าทางเฉลี่ยที่ใกล้เคียงกัน แต่ตัวประมาณของงานวิจัยอ้างอิงมีความสอดคล้องทางเวลาที่ดีขึ้นมาก โดยเปรียบเทียบการเปลี่ยนแปลงข้อผิดพลาดระหว่าง ground truth กับค่าประมาณตำแหน่งและทิศทาง

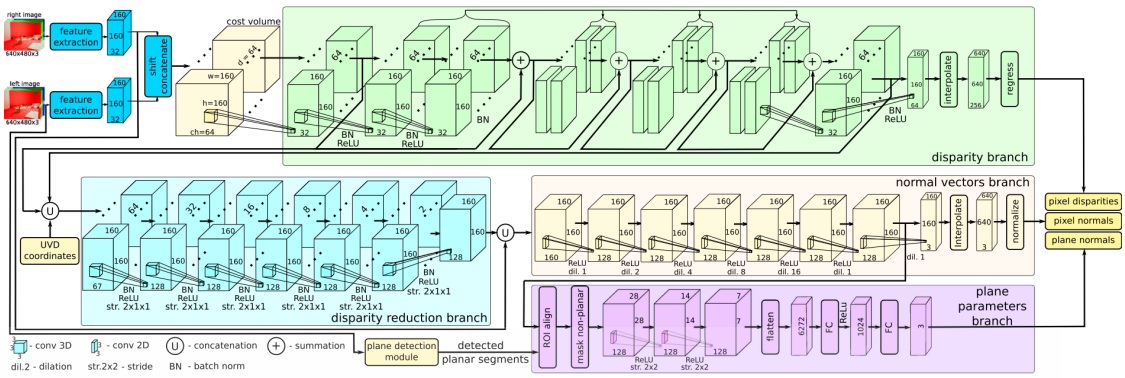
2.3 งานวิจัยอ้างอิง Stereo Plane R-CNN: Accurate Scene Geometry Reconstruction Using Planar Segments and Camera-Agnostic Representation [27]

งานวิจัยอ้างอิงของแจน (Jan Wietrzykowski) และโดมินิกส์ (Dominik Belter) เครือข่ายของงานวิจัยอ้างอิงนำเสนอสององค์ประกอบหลัก คือโมดูลตรวจจับระนาบ (plane detection module) และโมดูลเรขาคณิต (geometry module) (รูปที่ 1) ในส่วนของโมดูลการตรวจจับระนาบได้รับแรงบันดาลใจจากสถาปัตยกรรม Plane R-CNN [28] ที่ตรวจจับส่วนระนาบบนภาพ (ซ้ายในระบบ) งานวิจัยอ้างอิงนี้ได้ทำการปรับปรุงคุณภาพการตรวจจับของส่วนระนาบโดยใช้การแบ่งส่วน ROI ที่สมเหตุสมผล (ROI-aware segmentation) ระหว่างการฝึกฝนและโดยการเรียนรู้บนชุดข้อมูลที่มีป้ายกำกับ (labeled) อย่างเหมาะสม โมดูลเรขาคณิตได้รับแรงบันดาลใจจากผลงานของกุสุพาติและคณะ (Kusupati et al) [34] จากการตั้งค่าสเตอริโอเพื่อคาดการณ์เกี่ยวกับเรขาคณิตของฉาก โมดูลนี้สร้างปริมาณต้นทุนสำหรับสเปซ (space) สามมิติจากเซ็นเซอร์ตรวจจับและประมวลผลการฝังเครือข่ายประสาทเทียม (neural network embeddings) เพื่อประเมินความแตกต่างกันของพิกเซล (pixel-wise disparities), เวกเตอร์ปกติ (normal vectors), และพารามิเตอร์ของระนาบ (plane

parameters) สำหรับการตรวจพบการแบ่งส่วน เพื่อลดความสูญเสียที่เกี่ยวข้องกับค่าโดยประมาณทั้งหมด และปรับปรุงประสิทธิภาพเรขาคณิตการประกอบกลับใหม่

2.3.1 สถาปัตยกรรมโมดูลเรขาคณิต (Geometry Module Architecture)

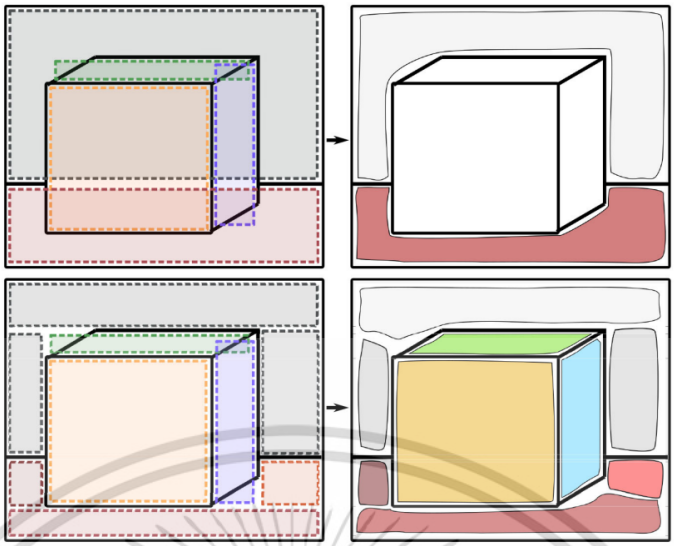
แม้ว่าพิกัดสามมิติของจุดที่คำนวณจากค่าความแตกต่าง (disparity) ที่ถูกประมาณนั้นไม่รับประกันว่าแบบจำลองระนาบจะแม่นยำ แต่โมดูลเรขาคณิต (รูปที่ 2.2) ใช้ประโยชน์จากคุณลักษณะที่ถูกสร้างขึ้นระหว่างการประมาณค่าความแตกต่างเพื่อประมาณค่าพารามิเตอร์ทั่วไปและพารามิเตอร์ของระนาบ โมดูลนี้สร้างขึ้นในสเปซ $\widetilde{\mathrm{UVD}}$ (อธิบายไว้ในส่วน 2.3.3) โดยที่คุณลักษณะจากภาพด้านซ้ายและขวาจะเชื่อมเข้าด้วยกันทุกจุดในสเปซนั้น ความแตกต่างของแต่ละสาขา (branch) (บล็อกสีเขียวในรูปที่ 2.2) อ้างอิงตามเครือข่ายการจับคู่สเตอริโอพีระมิด [35] ที่ใช้ 3D convolutions เพื่อประมวลผลคุณสมบัติที่เชื่อมต่อกันและสร้างการแจกแจงความน่าจะเป็น (probability distributions) ของความแตกต่างสำหรับแต่ละพิกเซล ค่าที่คาดหวังจะคำนวณจากการแจกแจงเหล่านั้นเพื่อค่าความแตกต่างแบบถดถอยขั้นสุดท้าย คุณลักษณะตั้งแต่เริ่มต้นและสิ้นสุดของการประมาณค่าความแตกต่างของสาขาจะถูกรวมเข้ากับพิกัด $\widetilde{\mathrm{UVD}}$ และถูกใช้ในการลดความแตกต่างของสาขา (สีน้ำเงินอ่อน) การใช้ 3D convolutions กับ stride ขนาด 2 ในมิติที่แตกต่างกัน ซึ่งลดขนาดของมิตินี้ลงครึ่งหนึ่งหลังจากการดำเนินการแต่ละครั้ง ผู้วิจัยลดขนาด (dimension) ของผังคุณลักษณะ (feature maps) ลงเหลือ 1 โดยขั้นตอนนี้จะช่วยลดคุณลักษณะที่แตกต่างกันออกอย่างมีประสิทธิภาพ เหลือไว้เพียงคุณลักษณะสองมิติ (2D) ที่ปกติ (normal features) คุณลักษณะปกติของสองมิติจะถูกรวมเข้ากับคุณลักษณะที่มองเห็นได้จากภาพด้านซ้ายและประมวลผลโดยใช้ convolutions สองมิติพร้อมการเปลี่ยนขนาดต่างๆ เพื่อให้ได้ผลค่าพารามิเตอร์สามค่าสำหรับทุกพิกเซล นอกจากนี้คุณลักษณะสองมิติภายในภาพถูกใช้ในพารามิเตอร์ของระนาบในสาขาย่อย (สีม่วงในรูปที่ 2.2) ที่สุ่มตัวอย่างโดยใช้ ROI ที่จัดเรียงตาม ROI ที่ตรวจพบ คุณลักษณะตัวอย่างที่ไม่ได้อยู่ในการแบ่งส่วนแต่อยู่ภายใน ROI จะถูกทำให้ค่าเป็นศูนย์ และผังคุณลักษณะดังกล่าวจะถูกประมวลผลโดยใช้ convolutional และ fully connected 2 ชั้นเพื่อประมาณค่าระนาบปกติ (plane normal)



รูปที่ 2.2 สถาปัตยกรรมของโมดูลเรขาคณิตใน Stereo Plane R-CNN 3D convolution โดย convolution สามมิติมีขนาด 3 × 3 × 3 และ convolution สองมิติมีขนาด 3 × 3 [27]

2.3.2 การตรวจจับและแบ่งส่วน ROI ที่สมเหตุสมผล (ROI-Aware Detection and Segmentation)

ผู้วิจัยเห็นว่า R-CNN มีปัญหากับส่วนที่เป็นระนาบซึ่งคลอบคลุมพื้นที่ขนาดใหญ่ของภาพ โดยเฉพาะอย่างยิ่งส่วนที่ซ้อนทับกับส่วนอื่นๆด้วย (ดังแสดงในรูปที่ 2.3) เกิดจากกล่อง ROI ที่มีหลายส่วน กล่องสำหรับส่วนต่างๆ ซ้อนทับกันและถูกระงับในขั้นตอน the Non-Maximum Suppression (NMS) ปัญหาเดียวกันนี้เกิดขึ้นกับทุกส่วนที่มีรูปร่างไม่ใช่สี่เหลี่ยมจัตุรัส ดังนั้นผู้วิจัยจึงเสนอให้แบ่งกลุ่มเป้าหมายระหว่างการฝึกออกเป็นกลุ่มที่เล็กลงและมีรูปร่างเหมือนสี่เหลี่ยมจัตุรัสมากขึ้น ตัวอย่างเช่น ด้านหน้าของตู้สามารถแบ่งเป็นส่วนเดียวหรือแยกส่วนสำหรับแต่ละประตูได้ การแบ่งส่วนทั้งสองแบบนั้นทำให้การโลคัลไลเซชัน (localization) หรือการนำทาง (navigation) ที่ใช้ระนาบเป็นพื้นฐานมีความถูกต้อง [36] อัลกอริธึมเริ่มต้นที่พิกเซลสุ่มและนำเข้าแบ่งส่วนตราบใดที่อัตราส่วนของ grown region ต่อพื้นที่ของขอบเขตกล่อง (bounding box) ที่เล็กที่สุดนั้นสูงกว่า 0.5 หากอัตราส่วนต่ำกว่าเกณฑ์นี้ ขอบเขตกล่องจะใหญ่กว่าการแบ่งส่วนนั้นมาก หมายความว่ารูปร่างของการแบ่งส่วนเบี่ยงเบนจากการเป็นสี่เหลี่ยมจัตุรัสและ ROI ของส่วนนี้สามารถซ้อนทับกับ ROI ของส่วนที่อยู่ใกล้เคียง แม้ว่าอัลกอริธึม greedy ที่ใช้นี้ไม่ได้รับประกันว่าเป็นการแบ่งส่วนที่ดีที่สุด แต่พบว่าสามารถใช้งานได้ดีในทางปฏิบัติกับการแบ่งส่วนเกินในระดับที่ยอมรับได้ ดังนั้น แทนที่จะใช้โมดูลที่ถูกปรับปรุงให้ดีขึ้นที่เสนอใน [28] ผู้วิจัยใช้มาสก์เป้าหมาย (target masks) ที่แบ่งกลุ่มอย่างระมัดระวังระหว่างการเรียนรู้เพื่อให้ได้การตรวจจับที่มีคุณภาพดีเมื่อทำการทดสอบ



รูปที่ 2.3 การแบ่งส่วน ROI ที่สมเหตุสมผล : NMS ลบการตรวจหาบางฉากสำหรับฉากที่ซับซ้อนซึ่งกล่องมีขอบเขตทับซ้อนกัน (บนสุด) การแบ่งส่วนรูปร่างที่ซับซ้อนมากเกินไป (ด้านล่าง) ทำให้เกิดการตรวจจับขนาดเล็กจำนวนมากขึ้น แต่ยังคงรักษาระนาบที่มีความสำคัญต่อการสร้างฉากขึ้นใหม่

2.3.3. เรขาคณิตของฉากจากกล้องสเตอริโอ (Scene geometry from stereo camera)

การจับคู่ระหว่างพิกัดสามมิติของจุดหรือเวกเตอร์ปกติและพิกัดพิกเซลนั้นขึ้นอยู่กับพารามิเตอร์ภายในกล้องเป็นอย่างมาก หากโมเดลกล่องดำ (เช่น โครงข่ายประสาทเทียม) ถูกนำไปใช้กับการประมาณพิกัดสามมิติจากรูปภาพจำเป็นต้องให้ความสนใจกับความสัมพันธ์นี้ ดังนั้นผู้วิจัยจึงทำการแทนค่าปกติของกล้องเพื่อลดความซับซ้อนของปัญหานี้ และเพื่อหลีกเลี่ยงการเปลี่ยนแปลงที่ไม่จำเป็นซึ่ง Deep Neural Networks (DNN) จะต้องเรียนรู้ หากอินพุตไปยัง DNN เป็นคู่ของภาพสเตอริโอ โครงสร้างข้อมูลที่มีภาพเหล่านั้นจะถูกจัดระเบียบตามคู่อันดับ (u, v) และความแตกต่าง (d) ของภาพ ดังนั้น u, v และ d จึงเป็นที่รู้จักในเครือข่ายสำหรับทุกจุดที่ประมวลผล สิ่งที่เครือข่ายไม่ทราบคือพิกัด XYZ ของจุด เนื่องจากพารามิเตอร์กล้องมีความจำเป็นในการคำนวณ ดังนั้นการประมาณค่าในสเปซ (space) XYZ ที่สัมพันธ์กับเฟรม (frame) ของกล้อง ผู้วิจัยใช้ประโยชน์จาก disparity-normalized UVD ($\widehat{\mathrm{UVD}}$) ที่เชื่อมโยงกับพิกัดพิกเซลและความไม่สอดคล้องกัน (disparity) มาช่วยลดปัญหาจากความแตกต่างระหว่างชุดข้อมูลการฝึกฝนที่มีอยู่และอุปกรณ์ที่ใช้ (target hardware) การแปลง (transformation) ระหว่างสเปซ XYZ และสเปซ UVD เป็นแบบเส้นตรง (linear) เพื่อให้ได้มาซึ่งการแปลงนี้ พิจารณาจากสมการของการฉายภาพบนจุดโลกสามมิติ (x, y, z) สำหรับกล้องสเตอริโอที่ปรับเทียบแล้วด้วยเส้นฐาน (baseline) b :

$$\begin{cases} u = \frac{f_x x}{z} + c_x - o_x \\ v = \frac{f_y y}{z} + c_y - o_y \\ d = \frac{f_x b}{z}, \end{cases} \tag{2.1}$$

โดยที่ $(x, y, z)^T$ คือตำแหน่งสามมิติของจุดในเฟรมของกล้อง (f_x, f_y), (c_x, c_y) เป็นพารามิเตอร์ภายในกล้อง และ o_x, o_y เป็นจุดออริจิน (origin) ของเฟรมพิกัด UVD ซึ่งสามารถเลือกได้โดยย้ายจากมุมบนซ้ายของภาพ การแปลงจุด $p = (x, y, z, 1)^T$ จาก XYZ เป็นจุด p_D ใน UVD โดยใช้ homogeneous coordinates สามารถเขียนได้ดังนี้:

$$p_D = G_{D,C} p \,, \tag{2.2}$$

โดยที่ $G_{D,C}$ เป็นเมทริกซ์จากสมการ (2.1) ดังนั้นพารามิเตอร์ของระนาบใน UVD $\pi_D = (n_u, n_v, n_d, -r_D)^T$ โดยที่ (n_u, n_v, n_d) เป็นเวกเตอร์ปกติ (normal vector), $\pi_D \cdot p_D = 0$ สามารถแปลงเป็น XYZ ได้โดย :

$$\pi = G_{D,C}^T \pi_D = G_{C,D}^{-T} \pi_D \,, \tag{2.3}$$

จากสมการ (2.2) โดยที่ $\pi = (n_x, n_y, n_z, -r)^T$ การแปลงนี้ยังเป็นเส้นตรง จะมีคุณสมบัติที่ไม่พึงประสงค์สำหรับวัตถุที่อยู่ไกล ความคลาดเคลื่อนเชิงมุมที่ค่อนข้างเล็กในการประมาณค่าปกติใน UVD จะส่งผลเป็นข้อผิดพลาดขนาดใหญ่ใน XYZ เพื่อแก้ปัญหานี้จึงมีการปรับพิกัดในพื้นที่ UVD ให้เป็นปกติด้วย :

$$\begin{cases} \tilde{u} = \frac{u}{d} = \frac{f_x}{f_x b} x + \frac{c_x - o_x}{f_x b} z \\ \tilde{v} = \frac{v}{d} = \frac{f_y}{f_x b} y + \frac{c_y - o_y}{f_x b} z \\ \tilde{d} = \frac{a}{d} = \frac{a}{f_x b} z \quad , \end{cases} \tag{2.4}$$

โดยที่ a เป็นค่าคงที่ที่ปรับขนาดสเปซให้สม่ำเสมอและทำให้ค่าใน $\widetilde{\mathrm{UVD}}$ มีขนาด (magnitude) เท่ากัน และการใช้ค่าของ o_x, o_y ใกล้เคียงกับ c_x, c_y (โดยปกติศูนย์กลางออปติคัลของกล้องจะอยู่ใกล้กับศูนย์กลางภาพและไม่แตกต่างกันมาก) ความสัมพันธ์ของมุมการสังเกตใน XYZ

และ $\widetilde{\mathrm{UVD}}$ จะเป็นแบบเส้นตรงโดยประมาณและไม่ขึ้นอยู่กับ d การใช้ homogeneous coordinates สามารถหาได้จาก :

$$p_{\bar{D}} = \begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{d} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f_x}{f_x b} & 0 & \frac{c_x - o_x}{f_x b} & 0 \\ 0 & \frac{f_y}{f_x b} & \frac{c_y - o_y}{f_x b} & 0 \\ 0 & 0 & \frac{a}{f_x b} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = G_{\bar{D},C}, p \tag{2.5}$$

ในการคำนวณสมการระนาบในสเปซนี้เพียงพอแล้วที่จะรู้พิกัดของภาพและความแตกต่างของจุดที่สร้างระนาบนี้ โดยไม่ต้องมีความรู้เกี่ยวกับความยาวโฟกัส f_x, f_y, optical center c_x, c_y และเส้นฐาน b นอกจากนี้ สเปซจะถูกปรับขนาดให้ใกล้เคียงกับสเปซ XYZ ดังนั้นข้อผิดพลาดเชิงมุมจึงไม่ถูกขยายอย่างมีนัยสำคัญ ในการแปลงระนาบ $\pi_{\bar{D}} = (n_{\bar{u}}, n_{\bar{v}}, n_{\bar{d}}, -r_{\bar{D}})^T$ จากสเปซ $\widetilde{\mathrm{UVD}}$ เป็นสเปซ XYZ ผู้วิจัยใช้สมการคล้ายกับสมการ (2.3) :

$$\pi = G_{C,\bar{D}}^{-T} \pi_{\bar{D}} \tag{2.6}$$

ในสาขาพารามิเตอร์ระนาบของ DNN ผู้วิจัยประเมินเฉพาะเวกเตอร์ปกติของการแบ่งส่วน ในการประมาณระยะทางไปยังจุดกำเนิด r ผู้วิจัยใช้ RANSAC และการประมาณค่าความแตกต่างจากสาขาความแตกต่าง (the disparity branch) ผู้วิจัยค้นหาชุด inlier ที่ดีที่สุดโดยใช้ RANSAC และใช้ค่า threshold ในระยะทางสัมพัทธ์ $\frac{n_h \cdot proj(p)}{r_h} < 0.05$ โดยที่ $proj(p)$ เป็นจุดสามมิติ XYZ ที่แสดงในพิกัดที่ไม่เท่ากัน n_h เป็นเวกเตอร์ปกติ (normal vector) ของสมมติฐาน RANSAC และ r_h คือระยะห่างจากจุดกำเนิดของสมมติฐาน RANSAC สุดท้าย r ถูกประมาณค่าโดยใช้ inliers ทั้งหมดจากสมมติฐานที่ดีที่สุด โดยปล่อยให้ DNN ประมาณค่าปกติไม่เปลี่ยนแปลง (RANSAC ที่ประมาณค่าปกติจะถูกละเว้น) ระหว่างการตรวจจับ ผู้วิจัยใช้เพียงสองคลาส (ระนาบและไม่ใช่ระนาบ) พบว่าการใช้จุดยึดสำหรับทิศทางปกติและการแบ่งระนาบออกเป็นคลาสที่เกี่ยวข้องกับทิศทางเหล่านั้น ดังเช่นใน [28] ไม่ได้ปรับปรุงความแม่นยำในการประมาณค่าปกติเมื่อเปรียบเทียบกับการประมาณค่าโดยตรงของพารามิเตอร์ปกติ 3 ตัว

ในงานวิจัยอ้างอิงนำเสนอวิธี Stereo Plane R-CNN ที่ตรวจจับและคำนวณพารามิเตอร์ของการแบ่งส่วนระนาบจากภาพคู่สเตอริโอ ระบบได้รับการฝึกฝนเกี่ยวกับชุดข้อมูลสังเคราะห์ที่ให้ข้อมูลที่ถูกต้องเกี่ยวกับความลึกของฉาก, การแบ่งส่วน และพารามิเตอร์ของระนาบ ชุดข้อมูลในงานวิจัยอ้างอิงมีสภาพแวดล้อมที่ไม่มีโครงสร้างและมีท้าทายในอาคาร ผู้วิจัยยังเสนอสถาปัตยกรรมโครงข่าย

ประสาทเทียมแบบใหม่ที่ใช้ประโยชน์จากข้อมูลความไม่เท่าเทียมกันจากกล้องสเตอริโอ เพื่อสร้างรูปทรงเรขาคณิตของฉากขึ้นใหม่ได้อย่างแม่นยำ

โครงข่ายประสาทเทียมใช้ UVD ซึ่งปรับปรุงความทนทานต่อการเปลี่ยนแปลงพารามิเตอร์ของกล้อง สุดท้าย ผู้วิจัยเสนอขั้นตอนการฝึกฝนที่ใช้พารามิเตอร์ของระนาบ ภาพเวกเตอร์ปกติแบบพิกเซล และการทำนายความแตกต่างไปพร้อม ๆ กัน เพื่อปรับปรุงความแม่นยำของการสร้างใหม่

บทที่ 3

วิธีดำเนินการวิจัย

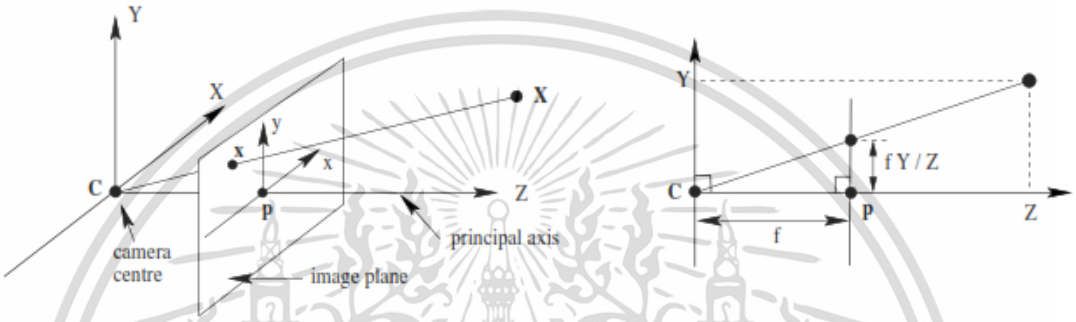
บทนี้จะกล่าวถึงข้อมูลทฤษฎีและอัลกอริทึมที่เกี่ยวข้องกับการค้นหาพารามิเตอร์ที่ใช้ประกอบภาพสามมิติจากภาพสองมิติ โดยแบ่งเป็น 9 ส่วน ประกอบด้วย การแปลงภาพจากโลกสามมิติเป็นสองมิติด้วยกล้อง, SIFT, เรขาคณิต epipolar, การแตกค่าแบบเอกฐาน (SVD), ซัพพอร์ตเวคเตอร์แมชชีน, Deep learning, Transfer learning, Ensemble CNN และ การหาค่าความผิดพลาดเฉลี่ยกำลังสอง โดยแต่ละส่วนมีรายละเอียดดังนี้

3.1 การแปลงภาพจากโลกสามมิติเป็นสองมิติด้วยกล้อง (From Camera Coordinates to World Coordinates) [37]

การแปลงภาพจากโลกสามมิติเป็นสองมิติด้วยกล้องที่เป็นความรู้พื้นฐานของงานวิจัยนี้นำเสนอการสร้างโมเดลกล้องด้วยเมทริกซ์ที่มีคุณสมบัติเฉพาะ โดยการแปลงตำแหน่งสามมิติในโลกจริงลงบนตำแหน่งภาพสองมิตินั้น เริ่มจากกล้องแปลงจากตำแหน่งสามมิติ (Scene) ลงบนจอแสดงภาพเป็นภาพถ่ายสองมิติ (Image) ด้วยการประมวลผลสีและแสงผ่านเซนเซอร์ระดับพิกเซลล์ (pixel) ภายในกล้อง ซึ่งการแปลงสามมิติเป็นสองมิติมี 2 ขั้นตอนหลัก คือการหาพารามิเตอร์ภายในกล้อง (Intrinsic camera parameters) ซึ่งในการแมพตำแหน่งในโลกสามมิติลงบนกล้องไปยังพิกเซลภายในกล้อง (map camera coordinate to pixel coordinate) ที่มีความเฉพาะของรูปแบบการเกิดภาพของกล้อง (general projective camera), ส่วนประกอบทั่วไปของโมเดลกล้อง (camera model) และพารามิเตอร์ภายนอกกล้อง (Extrinsic camera parameters) เป็นพารามิเตอร์แสดงการหมุนและการเลื่อนที่ของกล้อง โดยพารามิเตอร์เหล่านี้ถูกใช้สำหรับการโปรเจคภาพเชิงเรขาคณิต (projective geometry) ภายในของกล้อง เช่น การเกิดภาพกึ่งกลาง (projection centre) และระนาบภาพ (image plane) ถูกคำนวณในรูปของเมทริกซ์ที่ดังสมการที่ (3.1) ซึ่งการเกิดภาพของกล้องปกติสามารถหาได้จากคุณสมบัติของกล้อง เช่น การคำนวณทางเรขาคณิตจาก algebraic expressions เดียวกัน โดยจะเริ่มจากโมเดลกล้องที่พื้นฐานที่สุดคือ โมเดลรูเข็ม (pinhole model)

$$x_{2D} = PX_{3D} \tag{3.1}$$

โมเดลกล้องรูเข็มพื้นฐานพิจารณาจากจุดศูนย์กลางการเกิดภาพของจุดในพื้นที่บนระนาบ (plane) โดยให้จุดศูนย์กลางการเกิดภาพ (centre of projection) เป็นจุดศูนย์กลางของระบบคู่อันดับยูคลิเดียน (Euclidean coordinate system) และพิจารณาระนาบแกน Z ให้มีขนาดเท่าความยาวโฟกัส ในโมเดลกล้องรูเข็ม (pinhole camera model) เรียกระนาบนี้ว่าระนาบภาพหรือระนาบโฟกัส (image plane or focal plane) จุดในพื้นที่ที่คู่อันดับ $X_{3D} = (X, Y, Z)^T$ ถูกแมพไปยังจุดบนระนาบภาพที่เป็นเส้นร่วมจากจุด X ไปยังจุดกึ่งกลางที่เกิดภาพบนระนาบภาพ แสดงดังรูปที่ 3.1



รูปที่ 3.1 แสดงเรขาคณิตกล้องรูเข็ม (Pinhole camera geometry)

จากตำแหน่งจุดภาพในสามมิติ (X, Y, Z) มายังจุดภาพถ่ายสองมิติ การเกิดภาพจากกล้องรูเข็มสามารถหาได้ดังนี้

$$\begin{pmatrix} x_i \\ y_i \\ f \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} \tag{3.2}$$

$$x_i = f\frac{X_s}{Z_s}, \quad y_i = f\frac{Y_s}{Z_s} \tag{3.3}$$

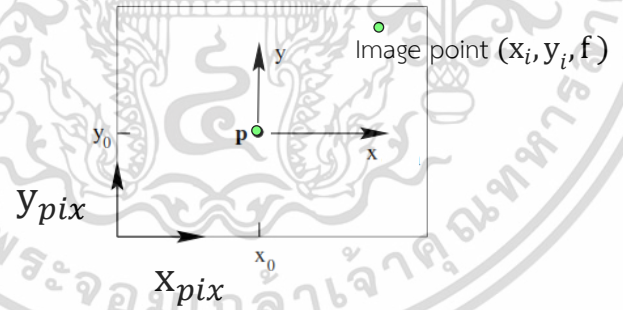
ซึ่ง C เป็นจุดศูนย์กลางกล้องที่จุดออริจิน (coordinate origin) และ p คือจุดสำคัญ โดยระนาบภาพเป็นตำแหน่งหน้าจุดศูนย์กลางกล้อง หาได้จากจุด $(X, Y, Z)^T$ ถูกแมพไปยังจุด $(fx/z, fy/z, f)^T$ บนระนาบภาพได้ดังนี้

$$(X_{3D}, Y_{3D}, Z_{3D})^T \mapsto (fx_{2D}/z_{2D}, fy_{2D}/z_{2D}, f)^T \tag{3.4}$$

จุดกึ่งกลางของการเกิดภาพเรียกจุดกึ่งกลางกล้องหรือเรียกว่า optical centre เส้นจากกึ่งกลางกล้องมาตั้งฉากกับระนาบภาพเรียกแกนพรินสิเพิล (principal axis) หรือรังสีพรินสิเพิลของกล้อง (principal ray) และจุดที่แกนพรินสิเพิลบรรจบกับระนาบภาพเรียกจุดสำคัญ (principal point) ระนาบที่ทะลุผ่านจุดกึ่งกลางกล้องซึ่งขนานกับระนาบภาพเรียกระนาบพรินสิเพิลของกล้อง โดยจุดกึ่งกลางการเกิดภาพหาได้จากการใช้โฮโมจีเนียสโคออดิเนท (Homogeneous coordinates) จุดบนโลกสามมิติและบนภาพถูกแสดงด้วยโฮโมจีเนียสเวคเตอร์ (Homogeneous vectors) จากนั้นจุดกึ่งกลางการเกิดภาพจะแมพแบบลิเนียร์ระหว่างคู่อันดับโฮโมจีเนียสเหล่านั้น ดังสมการที่ (3.4) สามารถเขียนให้อยู่ในรูปของมัลติพลิเคชันเมทริกซ์

$$\begin{pmatrix} X_{3D} \\ Y_{3D} \\ Z_{3D} \\ 1 \end{pmatrix} \longmapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.5}$$

เมื่อทำการแมพจากคู่อันดับสามมิติมาเป็นภาพสองมิติ จากนั้นเป็นการแปลงภาพสองมิติมาสู่พิกเซล Intrinsic camera parameters : map camera coordinate to pixel coordinate ดังรูปที่ 3.2



รูปที่ 3.2 คู่อันดับระดับพิกเซล $(x_{pix},\ y_{pix})$ และคู่อันดับของภาพ $(x_i,\ y_i)$

จากรูปที่ 3.2 จุด p (principal point) อยู่ที่ (p_x, p_y) โดยมีจำนวนพิกเซลต่อระยะทางในพิกัดของภาพ (สเกลลิ่ง) ในแกน x และ y เป็น k_x และ k_y ตามลำดับ ในกรณีกล้องมีขนาดของพิกเซลไม่ใช่จตุรัส หากภาพถูกวัดเป็นพิกเซลจตุรัส อาจส่งผลให้การทำสเกลลิ่งไม่เท่ากันในแต่ละ

ทิศทาง การแปลงจากพิกัดของสามมิติมาเป็นระดับพิกเซลในกล้องทำได้โดยการคูณ เข้าสมการ (3.5) ด้านซ้ายโดย $diag(k_x, k_y, 1)$ ดังนั้นรูปแบบทั่วไปของ calibration matrix ของกล้องคือ

$$K = \begin{bmatrix} \alpha_x & & x_0 & 0 \\ & \alpha_y & y_0 & 0 \\ & & 1 & 0 \end{bmatrix} \tag{3.6}$$

โดยที่ $\alpha_x = fk_x$ และ $\alpha_y = fk_y$ คือความยาวโฟกัสของกล้องในระดับพิกเซลในทิศทาง x และ y ตามลำดับ ดังนั้น $\tilde{\mathrm{x}}_0 = (x_0, y_0)$ เป็นจุดสำคัญในข้อมูลระดับพิกเซล ซึ่งมีพิกัด $x_0 = p_x k_x$ และ $y_0 = p_y k_y$ เขียนให้อยู่ในรูปของสมการดังนี้

$$x_{\mathrm{pix}} = k_{\mathrm{x}} x_{\mathrm{i}} + x_0 = f k_{\mathrm{x}} \frac{\mathrm{X_s} + \mathrm{Z_s} p_x}{\mathrm{Z_s}},$$

$$y_{\mathrm{pix}} = k_{\mathrm{y}} y_{\mathrm{i}} + y_0 = f k_{\mathrm{y}} \frac{\mathrm{Y_s} + \mathrm{Z_s} p_y}{\mathrm{Z_s}} \tag{3.7}$$

ให้จุดออริจินในระนาบภาพอยู่ที่จุดสำคัญ เมื่อ $(p_x . p_y)^{\mathrm{T}}$ เป็นคู่อันดับของจุดสำคัญ ดังรูปที่ 3.2 แสดงในคู่อันดับโฮโมจีเนียส จะได้สมการดังนี้

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ & \alpha_y & y_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X_S \\ Y_S \\ Z_S \\ 1 \end{bmatrix} = \mathrm{K_{3x3}}[\mathrm{I}; 0]_{3x4} \tag{3.8}$$

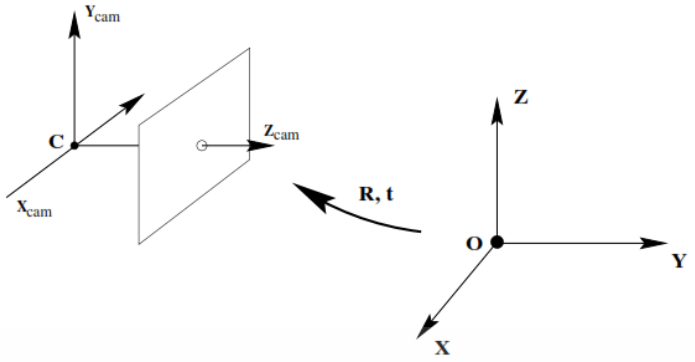
เมื่อ $x_{\mathrm{pix}} = \frac{u'}{w'}$, $y_{\mathrm{pix}} = \frac{v'}{w'}$,

α_x, α_y เป็น scale factor ในพิกเซลทิศทางของคู่อันดับ x และ y ตามลำดับ

x_0, y_0 เป็น คู่อันดับของกึ่งกลางภาพในพิกเซล

s เป็น พารามิเตอร์การบิด

K คือ แคลิเบชั่นเมทริกซ์ของกล้อง (camera calibration matrix)



รูปที่ 3.3 การแปลงยูคริเดียนระหว่างคู่อันดับของโครงสร้างสามมิติและกล้อง

การหมุนและการเลื่อนย้ายกล้องถูกแสดงให้อยู่ในรูปของโครงสร้างคู่อันดับยูคลิเดียนที่แตกต่างกันคือโครงสร้างคู่อันดับสามมิติ (world coordinate frame) คู่โครงสร้างที่เป็นคู่อันดับสองมิติมีความสัมพันธ์กับมุมหมุนและการเลื่อนย้าย ดังรูปที่ 3.3 เมื่อ $\tilde{X}$ เป็น inhomogeneous 3-vector แสดงคู่อันดับของจุดในโครงสร้างสามมิติ และ $\tilde{X}_{cam}$ คือจุดเดียวกันในโครงสร้างคู่อันดับกล้อง เขียนแทนด้วย $\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$ เมื่อ $\tilde{C}$ เป็นคู่อันดับของจุดกึ่งกลางกล้องในโครงสร้างคู่อันดับสามมิติ และ R เป็นเมทริกซ์การหมุนขนาด 3x3 แสดงมุมหมุนของโครงสร้างคู่อันดับกล้อง ดังสมการต่อไปนี้

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X \qquad (3.9)$$

เขียนอยู่ในรูปสูตรได้

$$X_{cam} = KR[I| - \tilde{C}]X$$

เมื่อ X_{cam} คือ ตำแหน่งกล้องที่สมมติให้ถูกย้ายมาที่จุดออริจินของระบบคู่อันดับยูคลิเดียน (Euclidean coordinate system) กับแกนพรินสิเพิลของจุดตั้งกล้อง

X คือ โครงสร้างคู่อันดับสามมิติซึ่งถูกแมพด้วยกล้องรูเข็ม

การเลื่อนจากโลกสามมิติเป็นรูปภาพ $\widetilde{\mathrm{X}}_{\mathrm{cam}} = \mathrm{R}\widetilde{\mathrm{X}} + \mathrm{t}$ สามารถเขียนให้อยู่ในรูปแบบเมทริกซ์ของกล้องเป็น

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \qquad (3.10)$$

เมื่อ $t = -\mathrm{R}\tilde{\mathrm{C}}$

จากนั้นแทนค่าให้ 3x4 projection matrix $P = K[R|t]$ ในสมการที่ (3.10)

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} (R \;\; t) \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix} = P \begin{pmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{pmatrix}$$

$$\mathrm{x}_{2D} = K[R, t]X_{3D}$$

$$\mathrm{P} = \begin{bmatrix} \mathrm{K}_{2\times 2} & \hat{0} \\ \hat{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathrm{R}} & \hat{\mathrm{t}} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}$$

$$\mathrm{x}_{2D} = \mathrm{PX}_{3\mathrm{D}}$$

3.2 การสกัดคุณลักษณะด้วยวิธี SIFT (Scale Invariant Feature Transform) [38]

การจับคู่คุณลักษณะต่างๆในภาพที่แตกต่างกันเป็นปัญหาที่พบโดยทั่วไปของคอมพิวเตอร์วิชัน (computer vision) เมื่อภาพทั้งหมดมีลักษณะคล้ายกัน (มีมาตราส่วน (scale), การวางแนว (orientation) ฯลฯ เท่ากัน) การตรวจจับมุมสามารถทำการจับคู่ภาพที่มีคุณลักษณะเหมือนกันได้ แต่เมื่อภาพมีมาตราส่วนและการวางแนวที่แตกต่างกัน จำเป็นต้องนำเทคนิค SIFT (Scale Invariant Feature Transform) มาช่วยในการแก้ปัญหา เนื่องจากคุณสมบัติของ SIFT มีความยืดหยุ่นต่อผลกระทบของจุดรบกวน (noise) ในภาพ เมื่อทำการดึงข้อมูลคุณสมบัติและวิธีบันทึกข้อมูลเหล่านี้ คุณลักษณะของภาพแบบ SIFT จะไม่ได้รับผลกระทบจากภาวะแทรกซ้อนที่พบแบบในวิธีการอื่น ๆ เช่นการปรับขนาด, การหมุนวัตถุ, ความเบลอ, ความเข้มแสง (Illumination), มุมมอง (Viewpoint) หากมีการเปลี่ยนแปลงข้อมูลเหล่านี้ เทคนิค SIFT ยังคงได้ผลลัพธ์การจับคู่ภาพที่มีประสิทธิภาพ

วิธีการ SIFT สำหรับการสร้างคุณลักษณะภาพด้วยวิธีแปลงรูปภาพเป็นกลุ่มของเวกเตอร์คุณสมบัติท้องถิ่น [39] เวกเตอร์คุณลักษณะแต่ละตัวนี้มีค่าไม่เปลี่ยนแปลงกับการปรับขนาดการหมุนหรือการแปลภาพ โดยการเตรียมการเบื้องต้นของขั้นตอน SIFT ต้องมีการสร้างพื้นที่มาตราส่วน (scale space) ภายในภาพต้นฉบับเพื่อให้มีการปรับค่าสเกลด้วยการประมาณของ LOG (Laplace Of Gaussian) สำหรับการค้นหาจุดที่น่าสนใจ (หรือจุดสำคัญ) ในภาพ ซึ่งค่อนข้างใช้ระยะเวลานาน ดังนั้นจึงมีการหาจุดสำคัญด้วยการประมาณค่าที่ใช้ระยะเวลาน้อยที่สุดในปัจจุบันคือการหา maxima และ minima ในความแตกต่างของภาพ Gaussian จากนั้นทำการกำจัดจุดสำคัญที่ไม่ดี ได้แก่ ขอบและพื้นที่ที่มีความคมชัดต่ำเนื่องจากเป็นจุดสำคัญที่ไม่ถูกต้อง การขจัดสิ่งเหล่านี้ทำให้อัลกอริทึมมีประสิทธิภาพซึ่งใช้เทคนิคที่คล้ายกับ Harris Corner Detector

ขั้นตอนของ SIFT ประกอบด้วย 4 ขั้นตอนสำคัญ คือ 1) การสร้างปริภูมิค่าในมิติขนาดระยะทาง 2) การกำหนดตำแหน่งจุดสำคัญ 3) การกำหนดทิศทางของจุดสำคัญ 4) การสร้างคำอธิบายลักษณะเด่นของภาพ ดังนี้ :

ขั้นตอนที่ 1 ในขั้นตอนก่อนหน้านี้เราได้สร้างพื้นที่มาตราส่วนของรูปภาพ หลักการของขั้นตอนนี้คือการทำให้ภาพเบลอขึ้นเรื่อยๆ จากนั้นใช้ภาพเบลอเหล่านี้ในการสร้างชุดภาพอื่นๆ เรียกการทำภาพ Difference of Gaussians (DoG) ดังสมการที่ (3.11) ซึ่งภาพ DoG เหล่านี้เหมาะสำหรับการค้นหาจุดสำคัญในภาพ ในขั้นตอนแรกการตรวจหาจุดสำคัญคือการระบุตำแหน่งและมาตราส่วนภายใต้มุมมองที่แตกต่างกันของวัตถุเดียวกัน โดยการระบุตำแหน่งจุดสำคัญให้คงที่แต่มีการปรับเปลี่ยนมาตราส่วนของภาพเพื่อค้นหาคุณสมบัติที่มีคุณลักษณะคงที่ที่เป็นไปได้ทั้งหมดด้วยฟังก์ชั่นมาตราส่วนต่อเนื่องของพื้นที่มาตราส่วน [40] ซึ่งถูกกำหนดไว้เป็นฟังก์ชัน $L(x,y,\sigma)$ ดังสมการ (3.12) ที่เกิดจากการทำ Gaussian $G(x,y,\sigma)$ ของภาพอินพุท (input) $I(x,y)$ ดังรูปที่ 3.4

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{3.11}$$

เมื่อ

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{3.12}$$

และ

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.13}$$

โดย $L(x,y,\sigma)$ คือ พื้นที่ขนาดของภาพถูกกำหนดไว้เป็นฟังก์ชัน

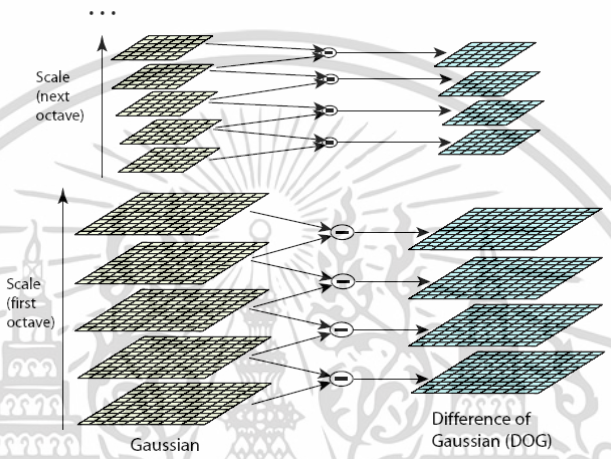
$I(x,y)$ คือ ภาพอินพุท

G คือ ตัวดำเนินการ Gaussian Blur

x, y คือ ตำแหน่งพิกัด

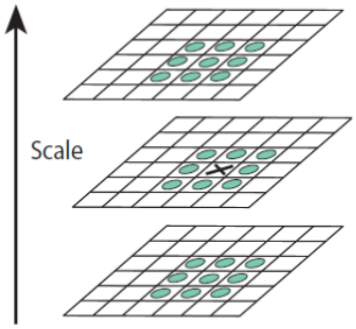
σ คือ พารามิเตอร์ "scale" เป็นจำนวนภาพเบลอ ค่ายิ่งมากภาพเบลอมากขึ้นขึ้น

$*$ คือ การดำเนินการ convolution ใน x และ y ใช้ "gaussian blur" G ลงบนภาพอินพุท

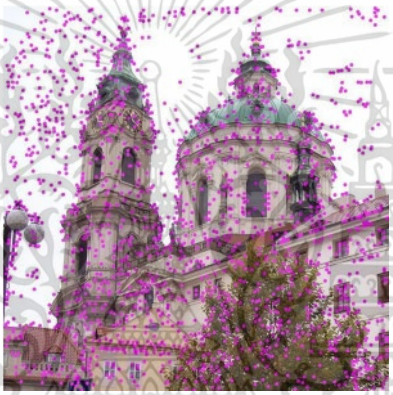


รูปที่ 3.4 การประมาณค่าจุดสำคัญในแต่ละพื้นที่

ภาพเริ่มต้นถูกแยกซ้ำๆด้วย Gaussians เพื่อสร้างชุดของพื้นที่มาตราส่วนในแต่ละภาพซึ่งถูกแยกเป็นแต่ละ Octave คือการทำเป็นสองเท่าของค่าเกาส์เซียน σ_0 หาได้จากค่าความแตกต่างของมาตราส่วนที่อยู่ติดกันในแต่ละ octave ด้วยค่าคงที่ k ถ้า octave มีจำนวน $s+1$ ภาพ แล้ว $k = 2^{(1/s)}$ ภาพแรกมีมาตราส่วนเป็น σ_0, ภาพที่สองมีมาตราส่วนเป็น $k\sigma_0$, ภาพที่สามมีมาตราส่วนเป็น $k^2\sigma_0$ และภาพสุดท้ายมีมาตราส่วนเป็น $k^s\sigma_0$ ลำดับของภาพแยกตำแหน่งด้วย Gaussians ของการเพิ่มขึ้นของ σ เรียกพื้นที่มาตราส่วน แต่ละ octave ภาพ Gaussian จะถูกเก็บตัวอย่างเพื่อสร้างภาพระดับถัดไป โดยการทำ extrema คือตรวจหา maxima และ minima ของความแตกต่างของ Gaussian ในพื้นที่มาตราส่วนหาได้จากการนำแต่ละจุดเปรียบเทียบกับ 8 เพื่อนบ้านในภาพปัจจุบันและ 9 เพื่อนบ้านแต่ละตัวในมาตราส่วนด้านบนและด้านล่างดังรูปที่ 3.5 และจะได้ภาพที่มีจุดสำคัญดังรูปที่ 3.6



รูปที่ 3.5 การค้นหา maxima และ minima ของภาพ DOG ซึ่งถูกตรวจจับโดยการเปรียบเทียบพิกเซล (เครื่องหมาย x) กับ 26 เพื่อนบ้านในพื้นที่ 3 * 3 ที่มาตราส่วนติดกัน



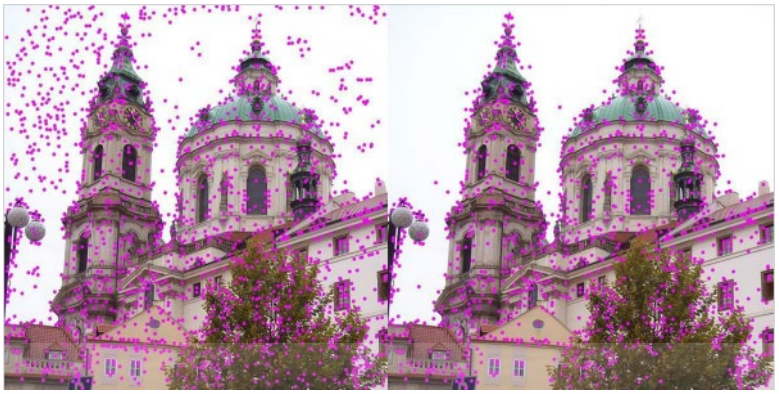
รูปที่ 3.6 จุดสำคัญที่ได้จากการทำขั้นตอนที่ 1 ของ SIFT (Initial detection of keypoints)

ขั้นตอนที่ 2: การกำหนดตำแหน่งจุดสำคัญ

ตำแหน่งของจุดสำคัญและมาตราส่วนที่ได้จากขั้นตอนที่ 1 มีความไม่ต่อเนื่อง สามารถแก้ไขทำให้ความถูกต้องของจุดสำคัญมากขึ้นในรูปของฟังก์ชัน DoG ของเพื่อนบ้านที่ล้อมรอบจุดสำคัญ (x_i, y_i, σ_i) โดยการหาอนุพันธ์ลำดับที่สองของ Taylor-series ดังสมการ (3.14) และการหา extremum ของค่า DoG ด้วยการให้อนุพันธ์ลำดับที่สองของ D(.) เป็น 0

$$D(x,y,\sigma) = D(x_i,y_i,\sigma_i) + \left(\frac{\partial D(x,y,\sigma)}{\partial (x,y,\sigma)}\right)^T_{\substack{x=x_i,\\ y=y_i,\\ \sigma=\sigma_i}} \Delta + \frac{1}{2}\Delta^T \left(\frac{\partial^2 D(x,y,\sigma)}{\partial (x,y,\sigma)^2}\right)_{\substack{x=x_i,\\ y=y_i,\\ \sigma=\sigma_i}} \Delta \quad (3.14)$$

เมื่อ $\Delta = \begin{pmatrix} x - x_i \\ y - y_i \\ \sigma - \sigma_i \end{pmatrix}$ จากนั้นตำแหน่งจุดสำคัญจะถูกอัพเดต ทุกๆจุดสำคัญที่มีค่า extrema น้อยหรือจุด low contrast จะถูกคัดออกด้วย $|D_{extreamal}| < 0.03$ ดังรูปที่ 3.7



รูปที่ 3.7 การนำจุดสำคัญที่ low-contrast ออก

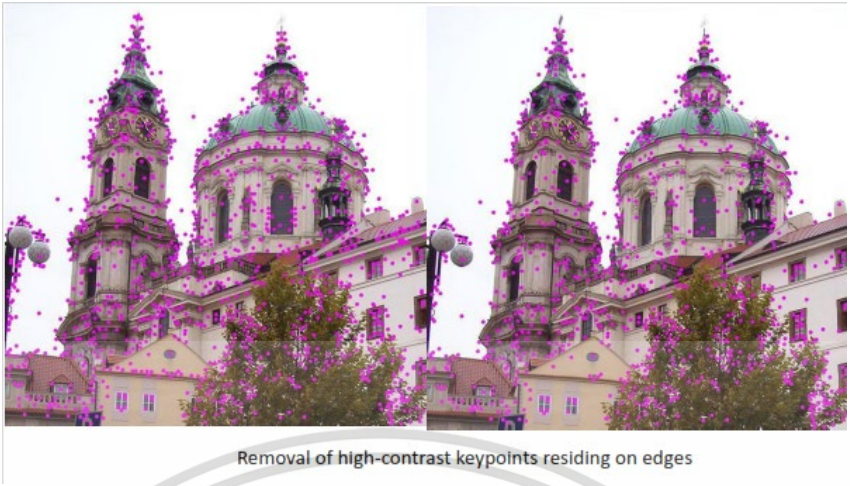
หลังจากการทำ extrema สำหรับบางจุดสำคัญที่อยู่บนตลอดแนวเส้นขอบที่มีค่าการตอบสนองสูง (high response) ต่อการกรอง DoG ซึ่งจำเป็นต้องนำจุดเหล่านั้นออกโดยการหาค่าไอเกนแวลูส์ (eigenvalues) ของเมทริกซ์ H (Hessian matrix) ที่เก็บข้อมูลเกี่ยวกับองค์ประกอบพื้นฐานรอบจุดสำคัญ ในการพิจารณาเส้นขอบหรือมุม สามารถพิจารณาได้จากค่าไอเกนแวลูส์คือค่าสูงสุดและต่ำสุดที่สำคัญของส่วนโค้งบนพื้นผิว (maximal and minimal principal curvatures) ของพื้นผิว $D(x,y)$ หากเป็นเส้นขอบ (edge) จะมี maximal curvature สูง แต่มี minimal curvature ต่ำ ในขณะที่มุม (corner) จะมี maximal curvature และ minimal curvature สูง ดังนั้น ค่าอัตราส่วน (γ) ของค่าไอเกนแวลูส์สูงสุด (α) ต่อค่าไอเกนแวลูส์ต่ำสุด (β) ใช้กำหนดว่าเป็นขอบหรือไม่ โดยใช้ความสัมพันธ์ระหว่าง trace (Tr) และ determinant (Det) ดังสมการที่ (3.15)

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma\beta+\beta)^2}{\gamma\beta^2} = \frac{(\gamma+1)^2}{\gamma} \tag{3.15}$$

เมื่อ $$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(H) = D_{xx}D_{yy} - \left(D_{xy}\right)^2 = \alpha\beta$$

$$\alpha = \gamma\beta$$



รูปที่ 3.8 การลดจุดสำคัญที่อยู่บนตลอดแนวเส้นขอบ

ขั้นตอนที่ 3 การกำหนดทิศทางของจุดสำคัญ

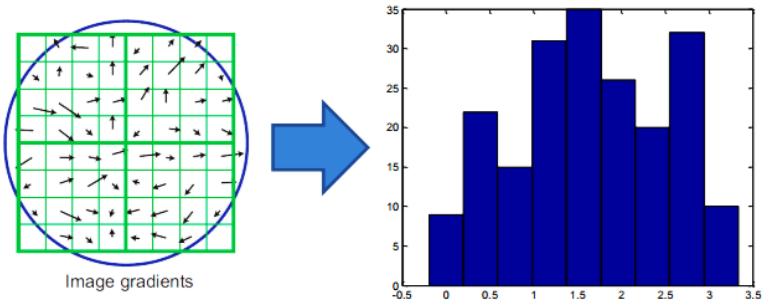
การคำนวณหาค่าของขนาดการเปลี่ยนแปลง (gradient magnitudes) และทิศทางการเปลี่ยนแปลง (gradient orientations) จากเพื่อนบ้านสี่ทิศรอบจุดสำคัญ ดังสมการที่ (3.16)

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3.16}$$

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y)))$$

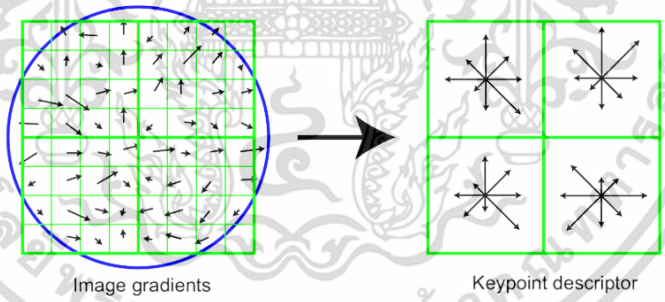
การกำหนดทิศทางหลักของทิศทางจุดสำคัญจากค่าฮิสโทแกรมทิศทางการเปลี่ยนแปลงของค่าเกรเดียนท์สามารถพิจารณาจากผลรวมของจำนวนบินของทิศทางเกรเดียนต์ตามขนาดของเกรเดียนต์ในแต่ละพื้นที่ย่อยนั้นที่สูงที่สุดหรือมากกว่า 0.8 x peak ซึ่งสร้างแยก descriptor สำหรับแต่ละทิศทาง โดยแต่ละบินจะสัมพันธ์กับทิศทางของการเปลี่ยนแปลง $\theta(x, y)$ ดังนั้นค่าทิศทางของการเปลี่ยนแปลงของแต่ละพิกเซลจะถูกเก็บสะสมลงในบินตรงกับทิศทางที่สัมพันธ์กัน และถ่วงน้ำหนักด้วยขนาดการเปลี่ยนแปลง $m(x, y)$ ของพิกเซลนั้นๆ (ในแต่ละบินมีมาตราส่วนและเป็นการนำทิศทางเดียวกันมาอยู่ในบินเดียวกัน) ดังรูปที่ 3.9



รูปที่ 3.9 กราฟฮิสโทแกรมของแนวทิศทางหลักของทิศทางจุดสำคัญ

ขั้นตอนที่ 4 การสร้างคำอธิบายลักษณะเด่นของจุดสำคัญ

การพิจารณาพื้นที่เล็ก ๆ รอบจุดสำคัญถูกแบ่งเป็น n x n เซลล์ (n = 2) แต่ละเซลล์มีขนาด 4 x 4 เพื่อสร้างทิศทางของเกรเดียนฮิสโทแกรมในแต่ละเซลล์ การบันทึกข้อมูลในแต่ละฮิสโทแกรมมีการถ่วงน้ำหนักโดยค่าขนาดการเปลี่ยนแปลงและฟังก์ชันการชั่งน้ำหนัก Gaussian ด้วยความกว้างของหน้าต่าง $\sigma = 0.5$ เท่า (การจัดเรียงฮิสโทแกรมของทิศทางการเปลี่ยนแปลง ในแต่ละทิศทางการเปลี่ยนแปลงหลักของจุดสำคัญ ที่ถูกกำหนดไว้ใน ขั้นตอนที่ 3) ดังรูปที่ 3.10



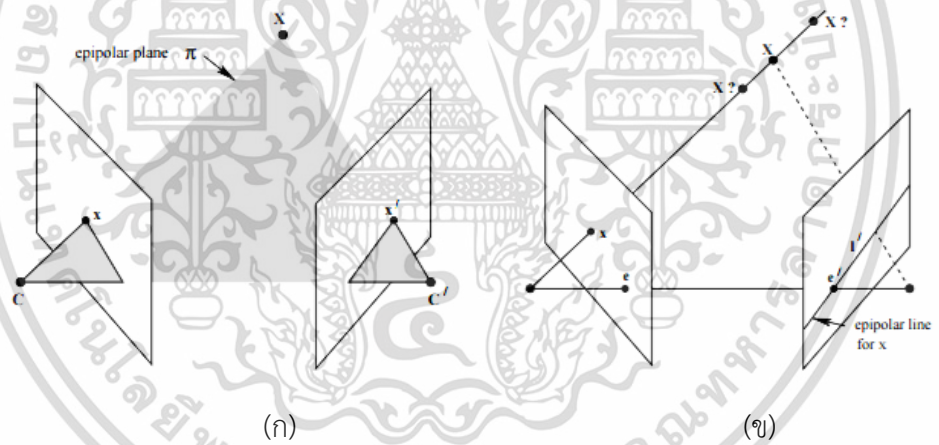
รูปที่ 3.10 รูปทิศทางการเปลี่ยนแปลงหลักของแต่ละจุดสำคัญ [38]

3.3 เรขาคณิต epipolar (epipolar geometry) [37]

เรขาคณิต epipolar (epipolar geometry) เป็นค่าเรขาคณิตที่อยู่ภายในภาพฉาย (intrinsic projective geometry) ระหว่างสองมุมมอง (view) ซึ่งเป็นอิสระต่อโครงสร้างฉาก (scene structure) แต่ขึ้นอยู่กับเฉพาะพารามิเตอร์ภายในกล้อง (cameras' internal parameters) และ

ตำแหน่งที่สัมพันธ์กัน (relative pose) เมทริกซ์ F (Fundamental matrix) เก็บพารามิเตอร์ภายในกล้องให้อยู่ในรูปเรขาคณิต 3x3 ในแถวที่ 2 โดย X อยู่บนพื้นที่ที่ถูกถ่าย x เป็นมุมมองจากภาพแรก และ x' เป็นมุมมองจากภาพที่สอง จากนั้นจะได้ความสัมพันธ์ของจุดจากภาพทั้งสองคือ $x'^T Fx = 0$

เรขาคณิต epipolar ระหว่างสองมุมมองเป็นพื้นฐานการคำนวณทางเรขาคณิตของจุดตัด (intersection) ของระนาบภาพกับพู่กันระนาบ (pencil of planes) ที่มีเส้นฐานเป็นแกน (เส้นฐานคือเส้นที่เป็นเส้นร่วมระหว่างศูนย์กลางกล้อง กล่าวคือเป็นเส้นที่เกิดจากลากจุดศูนย์กลางกล้องหนึ่งไปอีกกล้องหนึ่ง) เรขาคณิตนี้เริ่มต้นจากการพิจารณาโดยการค้นหาจุดที่สอดคล้องกันในการจับคู่สเตอริโอ สมมติว่าจุด X ใน สามมิติ (3-space) ถูกถ่ายภาพในสองมุมมองที่ x ในลำดับแรกและ x' ในลำดับที่สอง ความสัมพันธ์ระหว่างจุดภาพที่เหมือนกันของจุด x และ x' ดังแสดงในรูปที่ 3.11 จุด x และ x' บนภาพ, ระยะห่างจากจุด X และศูนย์กลางของกล้องถ่ายรูปจะมีแนวแกนเดียวกัน ให้ระนาบนี้เป็น π เมื่อรังสีย้อนกลับมาจาก x และ x' ตัดกันที่ X และรังสีมีลักษณะเป็น coplanar อยู่ในระนาบ π ซึ่งเป็นพื้นที่ที่สำคัญสำหรับการค้นหาจุดในสามมิติที่เป็นจุดเดียวกัน

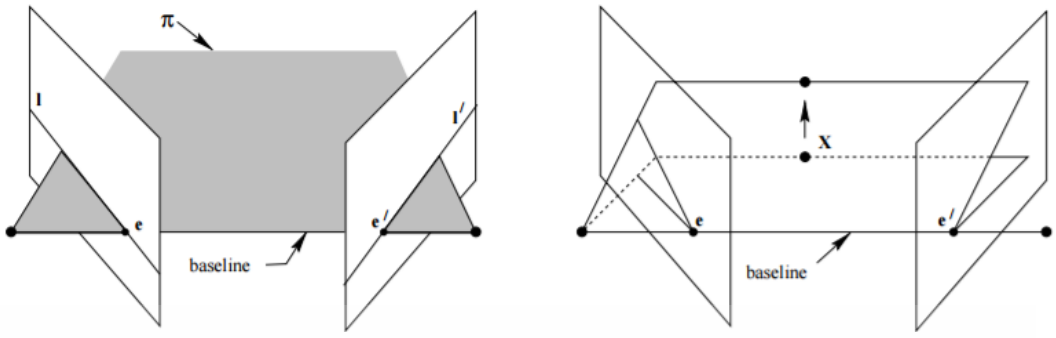


(ก) (ข)

รูปที่ 3.11 เรขาคณิตของจุดที่สัมพันธ์กัน

(ก) จุดศูนย์กลางกล้องสองตัวแทนค่าโดย C และ C' ของระนาบภาพสองมิติ จากกึ่งกลางแต่ละกล้องไปยังจุด X ในพื้นที่สามมิติและผ่านจุดบนภาพ x และ x' อยู่บนระนาบ π เดียวกัน

(ข) จากภาพมิติที่จุด x ผ่านรังสีฉายกลับไปยังพิกัดสามมิติซึ่งเริ่มจากจุดศูนย์กลางกล้อง C ของกล้องตัวแรกกับจุด x แทนรังสีนั้นด้วยเส้น l' ในรูป (ข) ดังนั้นจุด X ในพิกัดสามมิติที่ฉายไปยังจุด x จึงอยู่บนเส้นรังสี l เช่นกัน



ก ข

รูปที่ 3.12 เรขาคณิต epipolar (epipolar geometry)

(ก) เส้นฐานกล้อง (base line camera) ตัดกับแต่ละระนาบภาพที่จุด epipolar e และ e' เส้นฐานบนระนาบ π เป็นระนาบ epipolar (epipolar plane) และตัดกับระนาบภาพบนเส้น epipolar l และ l'

(ข) epipolar plane ของจุด X ที่ตำแหน่งสามมิติหมุนตามเส้นฐาน ซึ่งกลุ่มของระนาบเรียก epipolar pencil ทุกๆ epipolar line ตัดที่ epipole

สมมติเรารู้เพียงค่า x และต้องการหาความสัมพันธ์ของจุด x' บนระนาบ π หาได้จากเส้นฐานและรังสีที่กำหนดโดยจุด x หากรู้รังสีที่มีความเกี่ยวข้องกันกับจุด x' (ที่ไม่ทราบ) ที่อยู่บนระนาบ π ดังนั้นจุด x' จะอยู่บนเส้นที่ตัดผ่านเส้น l' บนระนาบ π ของภาพในมุมมองที่สอง ซึ่งเส้น l' เป็นเส้นรังสีที่ฉายกลับมาจากจุด x อยู่บนภาพในมุมมองที่สอง เป็นเส้นอิพิโพล่าที่สัมพันธ์กับ x ในส่วนของอัลกอริธึมความสัมพันธ์แบบสเตอริโอ (stereo correspondence algorithm) จุดที่มีความสัมพันธ์กับ x ไม่จำเป็นต้องครอบคลุมทั้งระนาบภาพ แต่สามารถอยู่บนเส้น l'

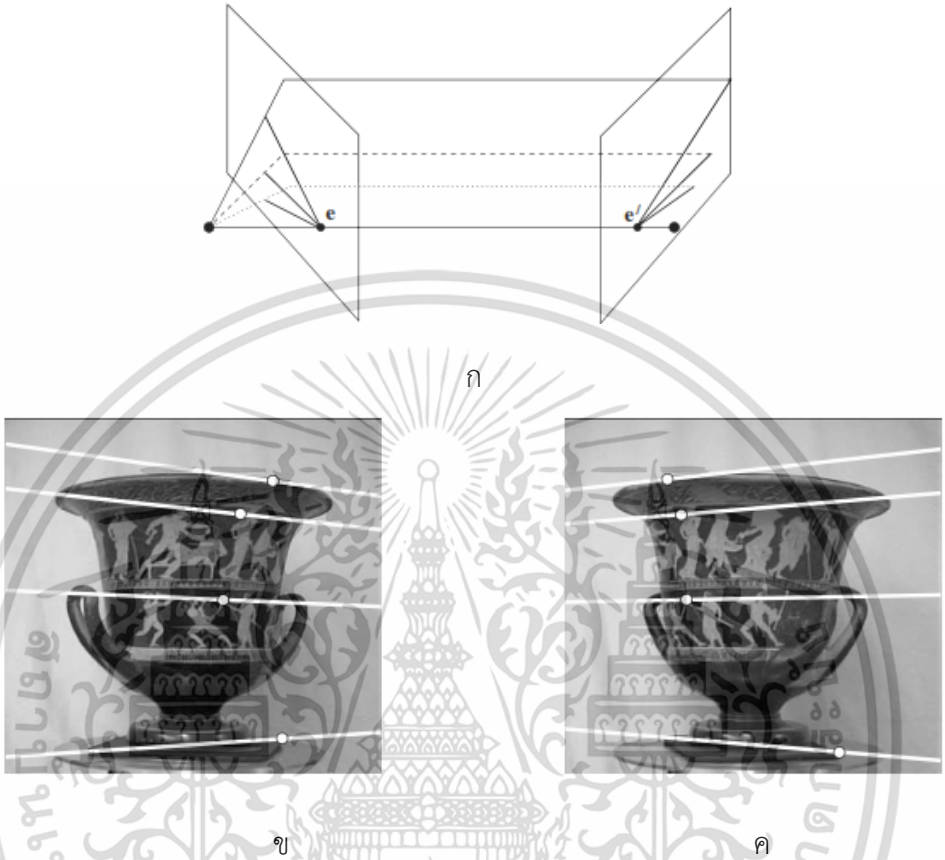
เรขาคณิตที่ใช้ในเรขาคณิตอิพิโพลาร์ในรูปที่ 3.12 มีดังนี้

epipole คือจุดตัดกันของเส้นที่เข้าร่วมสู่กลางกล้อง (เส้นฐาน) กับระนาบภาพ โดย epipole เกิดจากภาพในมุมมองเดียวของจุดศูนย์กลางกล้อง จากศูนย์กลางของกล้องในมุมมองอื่น นอกจากนี้ยังเป็นจุดที่หายไป (Vanishing point) ของทิศทางเส้นฐาน (การเลื่อน)

ระนาบ epipolar เป็นระนาบที่มีเส้นฐาน ซึ่งเป็นกลุ่มพารามิเตอร์กลุ่มหนึ่ง (pencil) ของระนาบ epipolar

เส้น epipolar เป็นจุดตัดของระนาบ epipolar กับระนาบภาพ เส้น epipolar ทุกเส้นตัดกันที่จุด epipole โดยระนาบ epipolar ตัดกบระนาบภาพด้านซ้ายและขวาในเส้น epipolar และ

กำหนดความสัมพันธ์ระหว่างเส้น ตัวอย่างของเรขาคณิต epipolar แสดงในรูปที่ 3.13 และ 3.14 เป็นเรขาคณิตของคู่ภาพ



ก

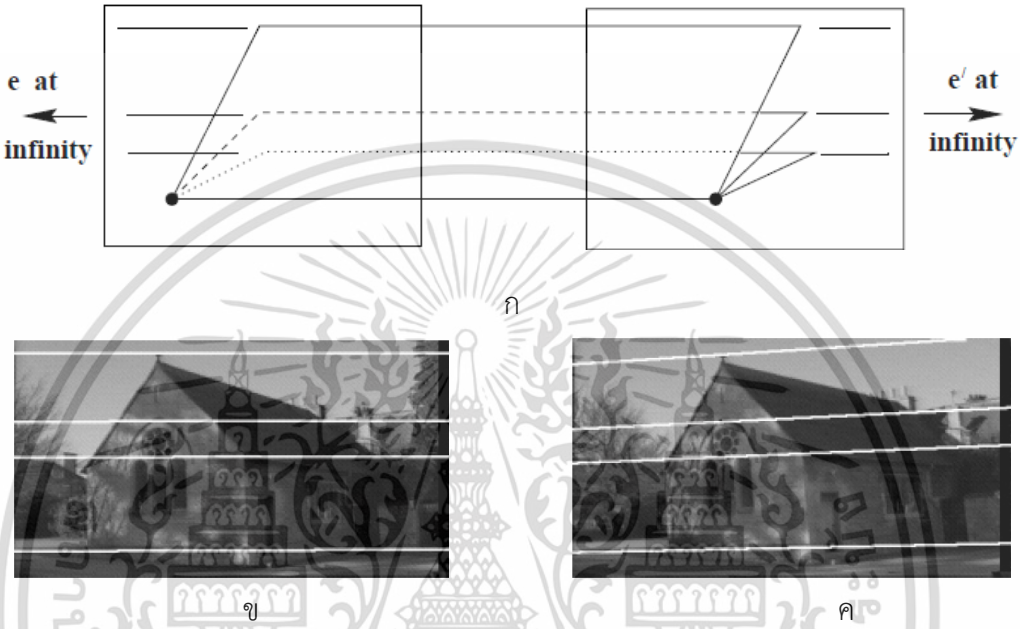
ข ค

รูปที่ 3.13 ภาพที่เกิดจากการกล้องถ่ายให้เบนเข้าหากัน (Converging cameras)
(ก) เรขาคณิต Epipolar สำหรับ converging cameras,
(ข) และ (ค) แสดงคู่ภาพที่มีจุดซ้อนทับที่สอดคล้องกันและเส้น epipolar (สีขาว) การเปลี่ยนตำแหน่งระหว่างมุมมองด้วยการเลื่อนตำแหน่งและการหมุน ในแต่ละภาพทิศทางของกล้องตัวอื่นอาจถูกอนุมานจากจุดตัดกันของ pencil ของเส้น epipolar ในกรณีนี้ epipoles ของทั้งสองภาพไปบรรจบกันอยู่นอกภาพ

3.3.1 เมทริกซ์พื้นฐาน F (The fundamental matrix F)

เมทริกซ์พื้นฐานเป็นตัวแทนเกี่ยวกับพีชคณิตของเรขาคณิต epipolar จากหัวข้อ 3.1 เมทริกซ์พื้นฐานจากการส่ง (mapping) ระหว่างจุดและเส้น epipolar จากนั้นทำการระบุคุณสมบัติของเมทริกซ์ จากคู่ภาพในรูปที่ 3.11 แต่ละจุด $\boldsymbol{x}$ ในหนึ่งภาพมีเส้น epipolar $\boldsymbol{l'}$ ที่สอดคล้องกันในภาพ

อื่น ๆ ที่จุด x' ใดๆในภาพที่สองที่ตรงกับจุด x ต้องอยู่บนเส้น epipolar l' โดยเส้น epipolar คือการฉายในภาพที่สองของ เส้นรังสีจากจุด x ผ่านศูนย์กลางกล้อง เขียนแทนด้วย $x \longmapsto l'$ คือจากจุดในภาพหนึ่งมีเส้น epipolar สัมพันธ์กับอีกภาพหนึ่ง ซึ่งเป็นการส่งความสัมพันธ์แบบเอกพจน์ (singular) ของการแมพจากจุดไปยังเส้นที่อยู่ในเมทริกซ์ F



ก

ข ค

รูป 3.14 การเคลื่อนที่แบบขนาน (Motion parallel) กับระนาบภาพ ในกรณีที่มีการเคลื่อนที่เฉพาะ (special motion) คือการเลื่อนที่มีการขนานกับระนาบภาพและแกนหมุนตั้งฉากกับระนาบภาพ จุดตัดของเส้นฐานกับระนาบภาพอยู่ที่ระยะอนันต์ ดังนั้น epipoles อยู่ที่อนันต์และเส้น epipolar เป็นเส้นขนาน

(ก) แสดงเรขาคณิต Epipolar สำหรับการเคลื่อนที่แบบขนานกับระนาบภาพ

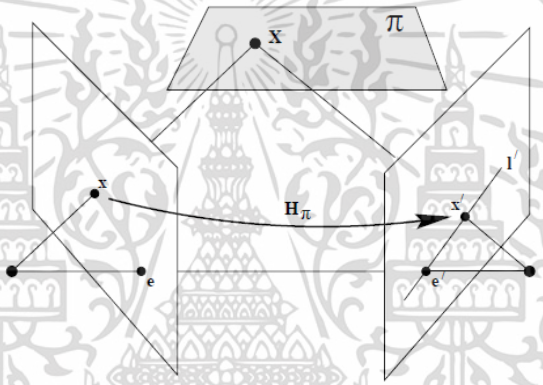
(ข) และ (ค) คือคู่ภาพที่มีการเคลื่อนที่ระหว่างมุมมองด้วยการเลื่อนที่ในแนวขนานกับแกน x โดยไม่มีการหมุน สี่เส้น epipolar ที่สอดคล้องกันแทนด้วยเส้นสีขาวและจุดที่สัมพันธ์กันอยู่บนเส้น epipolar ที่สอดคล้องกัน

3.3.1 การแปลงทางเรขาคณิต

เริ่มต้นจากรูปแบบทางเรขาคณิตของเมทริกซ์พื้นฐาน ซึ่งการแมพจากจุดในภาพหนึ่งไปยังเส้น epipolar ที่สอดคล้องกันกับในภาพอื่น ๆ ถูกแบ่งเป็นสองขั้นตอน ในขั้นตอนแรกจุด x จะถูก

แมพกับจุด x' ในภาพอื่นๆที่อยู่บนเส้น epipolar l' จากนั้นขั้นตอนที่สอง เส้น epipolar l' ถูกรวมเข้ากับ x' ไปที่จุด epipole e'

ขั้นตอนที่ 1: การเปลี่ยนตำแหน่งจุดผ่านระนาบ จากรูปที่ 3.15 พิจารณาระนาบ π ในพื้นที่ไม่ผ่านศูนย์การกล้องทั้งสองมุมมอง รังสีที่ผ่านกล้องตัวแรก ศูนย์กลางตรงกับจุด x บนระนาบ π ที่จุด X ซึ่งคือจุดที่ถูกฉายไปที่จุด x' ในภาพที่สอง ขั้นตอนนี้เรียกว่าการถ่ายโอนผ่านระนาบ π ตั้งแต่ X อยู่บนรังสีที่สอดคล้องกับ x จุดที่ฉาย x' ต้องอยู่บนเส้น epipolar l' ตรงกับภาพของรังสี ดังที่แสดงในรูปที่ 3.11ข จุด x และ x' ทั้งสองภาพของจุดสามมิติ X นอนอยู่บนระนาบ ชุดของทุกจุด x_i ในภาพแรกและจุดที่สอดคล้องกัน x_i' ในภาพที่สองฉายสมมูลกัน เนื่องจากแต่ละการฉายสมมูลกันไปยังจุดระนาบ X_i ดังนั้นจึงมีการทำการแมพแบบสองมิติ H_π ในแต่ละจุด x_i ไปเป็นจุด x_i'



รูปที่ 3.15 แสดงจุด x ในภาพหนึ่งถูกส่งผ่านระนาบ π ไปยังจุด x' ที่ตรงกันในภาพที่สอง เส้น epipolar ผ่าน x' ได้โดยการเข้าร่วม x' กับ epipole e' ในสัญลักษณ์หนึ่งอาจเขียนแทนด้วย $x' = H_\pi x$ และ $l' = [e'] \times x' = [e'] \times H_\pi x = Fx$ เมื่อ $Fx = [e'] \times H_\pi$ ซึ่งเป็นเมทริกซ์พื้นฐาน

ขั้นตอนที่ 2: การสร้างเส้น epipolar โดยให้จุด x' เส้น epipolar l' ที่ผ่านไปยังจุด x' และ epipole e' สามารถเขียนแทนด้วย $l' = e' \times x' = [e'] \times x'$ ดังนั้น x' สามารถเขียนเป็น $x' = H_\pi x$ ดังสมการที่ (3.17)

$$l' = [e'] \times H_\pi x = Fx \tag{3.17}$$

เมื่อกำหนดให้ $F = [e'] \times H_\pi$ ซึ่งเป็นเมทริกซ์พื้นฐาน

$$x'^{T}Fx = 0 \tag{3.18}$$

จากหัวข้อ 3.1 สามารถหาค่า F ได้ดังนี้

$$(x'_{2D}K'[R', t'])^{-T}[B]((K[R, t])^{-1}x_{2D}) = 0 \tag{3.19}$$

เมื่อทำการยึดตำแหน่งกล้องของกล้องแรกจึงส่งผลให้ $[R', t']$ ของกล้องแรกหายไป

$$(\mathrm{x}'_{2D}K')^{-T}[B]((K[R, t])^{-1}\mathrm{x}_{2D}) = 0 \tag{3.20}$$

$$\mathrm{x}'_{2D}E\mathrm{x}_{2D} = 0 \tag{3.21}$$

เมื่อ

$$(K')^{-T}[B](K[R, t])^{-1} = F \tag{3.22}$$

$$(K')^{-T}EK^{-1} = F \tag{3.23}$$

$$(K')^{T}FK = E \tag{3.24}$$

3.4 การแตกค่าแบบเอกฐาน (Singular Value Decomposition : SVD) [37]

SVD เป็นหนึ่งในการแยกเมทริกซ์ที่มีประโยชน์มากที่สุดสำหรับการคำนวณเชิงตัวเลข ซึ่งแอพพลิเคชันที่พบมากที่สุดคือการแก้ปัญหาของระบบที่มีการกำหนดจำนวนสมการมากเกิน โดยกำหนดให้เมทริก A เป็นเมทริกจตุรัส จากนั้นทำการแยกตัวประกอบของเมทริกซ์ A ได้ $A = UDV^{T}$ เมื่อ U และ V เป็นเมทริกซ์มุมฉาก (orthogonal matrices) และ D เป็นเมทริกซ์ทแยงมุม (diagonal matrices) ที่ภายในไม่มีจำนวนติดลบ ซึ่งการแยกเมทริกซ์นิยมเขียน V^{T} แทนด้วย V การแยกเมทริกซ์ (decomposition) สำหรับเมทริกซ์ D ลำดับในเมทริกซ์จะเป็นการเรียงรายการแบบเส้นทแยงมุมลงมา และคอลัมน์ของเมทริกซ์ V ที่มีค่า SVD น้อยที่สุดจะอยู่ที่คอลัมน์สุดท้ายของเมทริกซ์ V

นอกจากนี้ SVD สามารถแยกเมทริกซ์ที่ไม่เป็นจตุรัสได้ ให้ A มีจำนวนแถวมากกว่าจำนวนคอลัมน์ คือเมทริกซ์ขนาด $m \times n$ โดย $m \geq n$ จาก $A = UDV^{T}$ จึงส่งผลให้เมทริกซ์ U เป็นออโธ

กอนอลเมทริกซ์มีขนาด $m \times n$, D เป็นไดอกอนอลเมทริกซ์ขนาด $n \times n$ และ V เป็นออโธกอนอลเมทริกซ์ขนาด $n \times n$ ซึ่ง U มีคอลัมน์เป็นออโธกอนอล แสดงว่า $U^T U = I_{n \times n}$ ซึ่งมีคุณสมบัติ $\|Ux\| = \|x\|$ สำหรับเวกเตอร์ x ใดๆ และ UU^T ไม่เป็นเมทริกซ์เอกลักษณ์ยกเว้น $m = n$ เมื่อเมทริกซ์ A มีจำนวนคอลัมน์มากกว่าจำนวนแถว ($m < n$) จำเป็นต้องขยายเมทริกซ์ A โดยการเพิ่มแถวของศูนย์เพื่อให้เป็นเมทริกซ์จัตุรัส จากนั้นหาผลลัพธ์ของเมทริกซ์ด้วย SVD

การทำให้เกิดผลโดยทั่วไปของ SVD เช่นเดียวกับ [41] สมมติให้ $m \geq n$ เนื่องจากในกรณีนี้เมทริกซ์ U มีขนาด $m \times n$ เดียวกันเป็น input, เมตริกซ์ A อาจถูกแทนที่ด้วยเมทริกซ์เอาต์พุท U

3.4.1 ค่าเอกฐาน (SVD) และ ค่าไอเกนแวลูวส์ (eigenvalues)

จากที่ได้กล่าวก่อนหน้านี้ว่าค่าของเส้นทแยงมุมแต่ละตัวของเมทริกซ์ D ใน SVD ไม่เป็นจำนวนเชิงลบ และเรียกรายการเหล่านี้ว่า SVD ของเมทริกซ์ A แต่ไม่ใช่สิ่งเดียวกับค่าไอเกนแวลูวส์ ดังนั้นความสัมพันธ์ของ SVD ของ A กับค่าไอเกนแวลูวส์ เริ่มจาก $A = UDV^T$ จากนั้น $A^T A = VDU^T UDV^T$ เนื่องจาก V เป็นออโธกอนอล $V^T = V^{-1}$ ดังนั้น $A^T A = VD^2 V^{-1}$ นี่คือสมการการหาค่าไอเกนแวลูวส์ จากสมการข้างต้นจึงได้ D^2 เป็นค่าไอเกนแวลูวส์ของ $A^T A$ และคอลัมน์ของ V เป็นค่าไอเกนเวกเตอร์ของ $A^T A$ ค่า SVD ของ A คือ รากที่สองของค่าไอเกนแวลูวส์ของ $A^T A$ ดังนั้น SVD เป็นจำนวนจริงและที่ไม่ใช่เชิงลบ ซึ่งในหลักการสามารถแก้ไขได้โดยการอินเวอร์ตติ้ง (inverting) เมตริกสมมาตร ($A^T A$) ซึ่งเมทริกซ์ A โดยทั่วไปไม่ใช่เมทริกซ์สมมาตรและไม่มีการผกผัน

อัลกอริทึมเชิงตัวเลขสำหรับการแก้สมการเชิงเส้นภายใต้ข้อจำกัดต่างๆ จะเห็นได้ว่าปัญหาดังกล่าวได้รับการแก้ไขได้อย่างสะดวกโดยใช้ SVD

การแก้สมการเชิงเส้น

พิจารณาระบบสมการของ $Ax = b$ ให้ A เป็นเมทริกซ์ขนาด $m \times n$ ซึ่งมีความเป็นไปได้สามกรณี :

(1) ถ้า $m < n$ คือมีจำนวนตัวแปรไม่ทราบค่า (unknowns) มากกว่าจำนวนสมการ ในกรณีนี้จะไม่มีคำตอบที่ไม่ซ้ำกัน แต่เป็นเวกเตอร์ของโซลูชั่น

(2) ถ้า $m = n$ คือมีทางออกที่ไม่ซ้ำตราบเท่าที่มีการหาตัวผกผันได้

(3) ถ้า $m > n$ คือมีสมการมากกว่าจำนวนตัวแปรไม่ทราบค่า โดยทั่วไประบบจะไม่มีวิธีแก้ปัญหา ยกเว้นเปลี่ยน b ให้อยู่ในสแปน (span) ของคอลัมน์ของเมทริกซ์ A

วิธีหาคำตอบแบบ Least-squares : กรณีเต็มรูปแบบ (full-rank) เราพิจารณากรณี $m \geq n$ และสมมติว่า A เป็นอันดับที่ n ในหลาย ๆ กรณี การหาเวกเตอร์ x ที่ใกล้เคียงที่สุดในการแก้ปัญหาให้กับระบบ $Ax = b$ กล่าวอีกนัยหนึ่งคือ หา x ที่ $\|Ax - b\|$ มีค่าน้อยที่สุด เมื่อ $\|\cdot\|$ คือเวกเตอร์นอร์ม (vector norm) เรียก x ดังกล่าวว่า วิธีหาคำตอบแบบ Least-squares สามารถทำได้โดยใช้ SVD ดังต่อไปนี้ $\|Ax - b\| = \|UDV^T x - b\|$ เนื่องจากคุณสมบัติออโธกอนอล $\|UDV^T x - b\| = \|DV^T x - U^T b\|$ และนี่คือปริมาณที่เราต้องการลดขนาดลงให้มากที่สุด (minimize) ซึ่งสามารถเขียนให้อยู่ในรูป $y = V^T x$ และ $b' = U^T b$ เป็นปัญหาลดให้เหลือน้อยที่สุด $\|Dy - b'\|$ เมื่อ D เป็นเมทริกซ์ขนาด $m \times n$ ที่มีการหายไปของรายการนอกเส้นทแยงมุม รูปแบบชุดของสมการนี้คือ

$$m \times n \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \\ \hline & & 0 & \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \\ \hline b'_{n+1} \\ \vdots \\ b'_m \end{pmatrix} \tag{3.25}$$

ค่าที่เข้าใกล้ Dy ที่สุดสามารถมีค่าคล้ายกับ b' ซึ่งเป็นเวกเตอร์ $(b'_1, b'_2, \dots, b'_n, 0, \dots, 0)^T$ โดยให้ค่า $y_i = b'_i / d_i$ เมื่อ $i = 1, \dots, n$ เมื่อ จำนวนแถวของเมทริกซ์ $A = n$ และต้องแน่ใจว่า $d_i \neq 0$ ค่า x นำกลับคืนได้จาก $x = Vy$ เขียนแทนด้วยอัลกอลิธึม ดังนี้

วัตถุประสงค์

ค้นหาวิธีหาคำตอบแบบ Least-squares ของสมการ $Ax = b$ ขนาด $m \times n$ เมื่อ $m > n$ และ rank ของเมทริกซ์ $A = n$

ขั้นตอนวิธี

(i) ค้นหา SVD ด้วยสมการ $A = UDV^T$

(ii) ตั้งค่า $b' = U^T b$

(iii) ค้นหาเวกเตอร์ y โดย $y_i = b'_i / d_i$ เมื่อ d_i คือเส้นทแยงมุมตำแหน่งที่ i ของ D

(iv) ให้ค่า $x = Vy$

รูปที่ 3.16 วิธีการหา Linear Least-squares เพื่อกำหนดรูปแบบแบบเต็มของสมการแบบ Linear

จำนวนแถวไม่เพียงพอ บางครั้งเรียกการแก้สมการซึ่งคาดว่าจะไม่ได้เป็นรูปแบบแบบเต็มจำนวนคอลัมน์ ดังนั้นให้ $r = rankA < n$ เมื่อ n คือจำนวนคอลัมน์ของ A เป็นไปได้ว่าเนื่องจากสิ่งรบกวน (noise corruption) เมทริกซ์ A จึงมีจำนวนแถวมากกว่าจำนวนเงื่อนไข r เพราะการพิจารณาทฤษฎีมาจากปัญหาเฉพาะ (particular problem) ที่ได้รับการพิจารณา ในกรณีนี้จะมีกลุ่มพารามิเตอร์ $(n - r)$ ของการแก้ปัญหาเซตของสมการ โดยที่ $r = rank\ A < n$ ครอบครัวของโซลูชั่นนี้ได้รับการแก้ไขอย่างเหมาะสมด้วยวิธี *SVD* ดังต่อไปนี้:

อัลกอริธึมนี้จะให้กลุ่มพารามิเตอร์ $n - r$ (การหาค่าพารามิเตอร์โดยไม่ทราบค่า λ_i แน่ชัด ;parametrized by the indeterminate values λ_i) ของการแก้ปัญหาแบบ Least-squares ซึ่งมีจำนวนแถวไม่เพียงพอ

ระบบของจำนวนแถวที่ไม่ทราบค่า ในกรณีส่วนใหญ่จำนวนแถวของระบบสมการเชิงเส้น (System of linear equations) เป็นที่รู้จักในทางทฤษฎีของการแก้ปัญหาชั้นสูง หากไม่รู้จำนวนแถวของระบบสมการแล้วนั้นจำเป็นต้องคาดการจำนวนแถว ในกรณีที่เหมาะสมเพื่อกำหนดค่า SVD ที่มีขนาดเล็กเมื่อ*เทียบกับค่า* SVD ที่ใหญ่ที่สุดเป็นศูนย์ ดังนั้นถ้า $d_i/d_0 < \delta$ โดยที่ δ เป็นค่าคงที่ขนาดเล็กของลำดับของความแม่นยำเครื่อง จากนั้นให้ $y_i = 0$ การแก้ปัญหาแบบ Least-squares โดย $x = Vy$ เช่นเดียวกับข้างต้น

วัตถุประสงค์

การแก้ปัญหาทั่วไปของชุดสมการ $Ax = b$ เมื่อ A เป็นเมทริกซ์ขนาด $m \times n$ และ rank $r < n$

ขั้นตอนวิธี

(i) ค้นหา SVD $A = UDV^T$ โดยที่เส้นทแยงมุม d_i ของ D อยู่ในรูปตัวเลขที่ลดลงตามลำดับ

(ii) ตั้งค่า $b' = U^T b$

(iii) ค้นหาเวกเตอร์ y จาก $y_i = b'_i/d_i$ สำหรับ $i = 1, \dots, r$ และ $y_i = 0$ สำหรับกรณีอื่นๆ

(iv) หาค่า x ของค่านอร์มที่น้อยที่สุดของ x จาก Vy

(v) การแก้ปัญหาทั่วไปคือ $x = Vy + \lambda_{r+1}v_{r+1} + \cdots + \lambda_n v_n$ โดยที่ $v_{r+1} \dots v_n$ เป็นคอลัมน์สุดท้าย $n - r$ *ของ* V

รูปที่ 3.17 การแก้ปัญหาทั่วไปสำหรับระบบที่จำนวนแถวไม่เพียงพอ

สามารถนำมาประยุกต์ใช้ ดังนี้

จากหัวข้อ 3.3 สมการที่ (3.21)

$$x'_{2D} E x_{2D} = 0 \tag{3.26}$$

การหาค่า E ที่เป็นค่าความผิดพลาดจากการหาตำแหน่งร่วมในสามมิติจากภาพสองภาพ เพื่อมีค่าความผิดพลาดน้อยที่สุด สามารถหาได้จากการทำ Least-squares ดังนี้

$$A\vec{e} \approx w \tag{3.27}$$

$$\min_{\vec{e}} w^T w = \min_{\vec{e}} (A\vec{e})^T (A\vec{e}) \tag{3.28}$$

$$= \min_{\vec{e}} \left(\left(A\vec{e}\right)^T \left(A\vec{e}\right) \right) \tag{3.29}$$

$$= \min_{\vec{e}} (\vec{e}^T (A^T A)\vec{e}) \tag{3.30}$$

เมื่อ $A = UDV^T$ แทนค่าลงในสมการ 3.30 ดังนี้

$$= \min_{\vec{e}} (\vec{e}^T ((\mathrm{UDV^T})^T \mathrm{UDV^T})\vec{e}) \tag{3.31}$$

$$= \min_{\vec{e}} (\vec{e}^T (\mathrm{VD}^T \mathrm{U}^T \mathrm{UDV}^T)\vec{e})$$

$$= \min_{\vec{e}} (\vec{e}^T (\mathrm{VD}^T \mathrm{DV}^T)\vec{e})$$

3.5 การทำซัพพอร์ตเวคเตอร์แมชชีน (Support Vector Machine: SVM) [42]

ซัพพอร์ตเวคเตอร์แมชชีนเป็นการจำแนกประเภทแบบมีผู้สอน (Supervised Learning) ทฤษฎีนี้ได้มาจากแนวความคิดของแวพนิคและคณะ (Vapnik and et al.) [43] ดังรูปที่ 3.18 ซึ่งซัพพอร์ตเวคเตอร์แมชชีนมีพื้นฐานมาจากการเรียนรู้ทฤษฎีทางสถิติที่จะจำแนกข้อมูลด้วยสมการเส้นตรงให้ได้ผลลัพธ์ออกเป็น 2 กลุ่มออกจากกัน เทคนิคนี้จะพยายามสร้างเส้นแบ่งตรงกึ่งกลางระหว่างกลุ่ม

ให้มีระยะห่างระหว่างกลุ่มสองกลุ่มมากที่สุด ซึ่งเทคนิคนี้เป็นหนึ่งในการเรียนรู้ด้วยตนเอง (Machine Learning) โดยซัพพอร์ตเวคเตอร์แมชชีนจะมีเคอร์แนลฟังก์ชัน (Kernel Function) สำหรับเปลี่ยนแปลงตำแหน่งของข้อมูลไปยังโดเมนอื่นจากปริภูมิคุณลักษณะ (Feature Space) ไปยังปริภูมิผลลัพธ์ (Output Space) เพื่อใช้ในการจำแนกข้อมูลออกเป็น 2 กลุ่ม สำหรับเคอร์แนลฟังก์ชันที่พบบ่อย 4 แบบด้วยกัน ประกอบด้วย Linear Kernel เป็นเคอร์แนลพื้นฐานที่ใช้สมการเชิงเส้นที่มีลักษณะเป็นเส้นตรงในการคำนวณหาเส้นแบ่งกลุ่ม Polynomial Kernel เป็นเคอร์แนลที่ใช้ในสมการเชิงเส้นมีลักษณะไม่เป็นเส้นตรงในการคำนวณหาเส้นแบ่งกลุ่ม Gaussian (Radial Basis Function: RBF) Kernel เป็นเคอร์แนลที่ใช้การคำนวณหาขอบเขตของข้อมูลโดยอาศัยวิธีการแบบ Radial เข้ามาช่วยในการคำนวณ และ Sigmoid Kernel เป็นเคอร์แนลที่ใช้การคำนวณหาขอบเขตของข้อมูลเหมาะใช้สำหรับข้อมูลที่มีมิติของข้อมูลสูง ดังสมการที่ 3.32-3.35

Linear Kernel:

$$K(x_i, x_j) = x_i^T x_j \tag{3.32}$$

Polynomial Kernel:

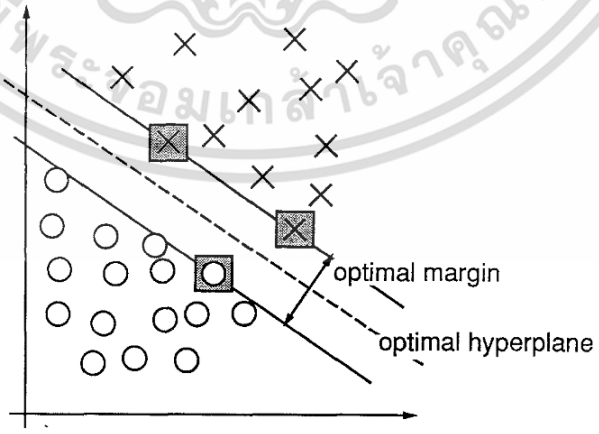
$$K(x_i, x_j) = (1 + x_i^T x_j)^P \tag{3.33}$$

Gaussian (Radial Basis Function: RBF) Kernel:

$$K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{3.34}$$

Sigmoid Kernel:

$$K(x_i, x_j) = tanh(\beta_0 x_i^T . x_j + \beta_1) \tag{3.35}$$



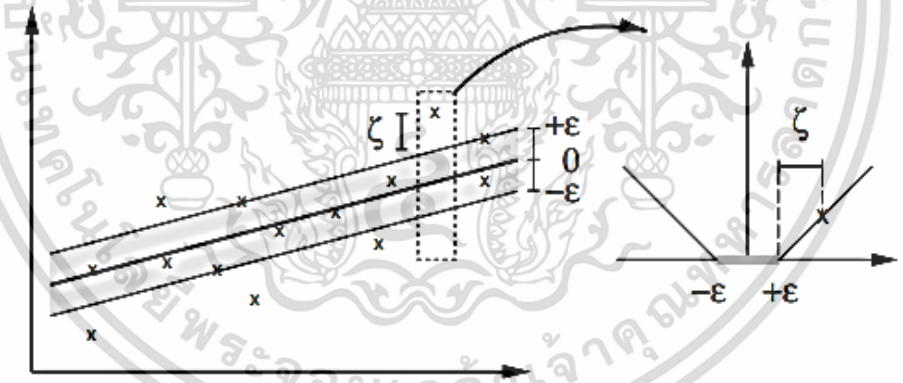
รูปที่ 3.18 แบบจำลองซัพพอร์ตเวคเตอร์แมชชีน [43]

input ของ SVR model สมมติได้รับข้อมูลฝึกฝน $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset \mathcal{X} \times \mathbb{R}$ เมื่อ $\mathcal{X}$ หมายถึงขอบเขต (space) ของรูปแบบอินพุต เป้าหมายคือการหาฟังก์ชั่น $f(x)$ ที่มีการเบี่ยงเบน $\mathcal{E}$ จากเป้าหมายจริง y_i ที่ได้รับสำหรับข้อมูลการฝึกฝนทั้งหมด ในทำนองเดียวกันค่าความผิดพลาดจะไม่ถูกสนใจตราบใดที่ยังน้อยกว่าค่าการเบี่ยงเบน โดยเริ่มต้นด้วยการอธิบายกรณีของสมการเส้นตรงดังสมการ (3.36)

$$f(x) = \langle w, x \rangle + b \quad \text{โดย } w \in \mathcal{X}, b \in \mathbb{R} \tag{3.36}$$

เมื่อ $\langle \cdot, \cdot \rangle$ คือ ดอทโปรดัคของ $\mathcal{X}$, w คือ เวกเตอร์น้ำหนัก

ฟังก์ชัน f เป็นฟังก์ชันประมาณค่าความแม่นยำคู่ (x_i, y_i) ด้วย $\mathcal{E}$ ซึ่งเป็นการแก้ปัญหา convex optimization ที่เป็นไปได้ แต่อย่างไรก็ตามบางค่าความผิดพลาดอาจถูกยกเว้น ในทางปฏิบัติข้อมูลอาจไม่สามารถแบ่งออกได้โดยวิธีเชิงเส้น ในกรณีเช่นนี้อาจใช้วิธี Soft-Margin แทน ซึ่งตัวแปรหย่อน (Slack) $\xi_i, \xi_i^* \geq 0$ จะถูกนำมาใช้สำหรับตัวอย่างข้อมูลฝึกฝนซึ่งอนุญาตให้มีการละเมิดเงื่อนไขของเวกเตอร์สนับสนุนได้โดยสามารถละเมิดได้ตามเงื่อนไขของค่า Slack ที่กำหนด ดังตัวอย่างแสดงในรูปที่ 3.19



รูปที่ 3.19 ตัวอย่างการใช้ SVM แบ่งกลุ่มข้อมูลเชิงเส้น [42]

ดังนั้นขอบเขตมีได้ทั้งหมดดังสมการ 3.37

โดยมีฟังก์ชั่นเป้าหมายเป็น $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$

$$y_i - \langle w, x \rangle - b \leq \mathcal{E} + \xi_i \tag{3.37}$$

$$\langle w, x \rangle + b - y_i \leq \mathcal{E} + \xi_i^*$$

จากสมการกำหนดให้ C เป็นค่าตัวแปรปรับความสมดุลที่ผู้ใช้สามารถกำหนดค่าได้เอง ระหว่างการให้ความสำคัญของระยะจำแนกสูงสุด หรือให้ความสำคัญกับค่าความผิดพลาดที่ต้องการให้ต่ำที่สุด กล่าวอีกนัยหนึ่งคือค่า C ที่น้อยทำให้วิธีการมุ่งเน้นไปที่ Soft-margin SVM ในขณะที่ค่า C ที่มากทำให้วิธีการมุ่งเน้นไปที่ Hard-margin SVM

นอกจากนี้ยังสามารถใช้การแปลงค่าตัวแปรคุณลักษณะเพื่อออกแบบวิธี SVM ที่ไม่ใช่เชิงเส้นได้ ในทางปฏิบัติวิธี SVM ที่ไม่ใช่เชิงเส้นจะเรียนรู้โดยใช้วิธีการของ Kernel แนวคิดหลักคือสมการ SVM สามารถแก้ปัญหาโจทย์ได้โดยใช้ผลคูณคู่จุด (Pairwise Dot Product) หรือค่าความคล้ายคลึงกันระหว่างวัตถุ แสดงได้ด้วยตัวแปรเหล่านี้

1. ผลคูณจุดคู่ของตัวอย่างข้อมูลฝึกฝนที่ต่างกัน
2. ผลคูณจุดคู่ของตัวอย่างข้อมูลทดสอบกับข้อมูลฝึกฝนในชุดที่ต่างกัน

ผลคูณระหว่างคู่ตัวอย่างสามารถมองเห็นเป็นค่าความคล้ายคลึงกันในหมู่ตัวอย่างได้ ดังนั้นข้อสังเกตดังกล่าวจึงมีความเป็นไปได้ที่จะสามารถใช้ SVM แบ่งกลุ่มได้ด้วยข้อมูลความคล้ายกันระหว่างคู่ข้อมูลฝึกฝนและคู่ข้อมูลฝึกฝนกับข้อมูลทดสอบได้โดยไม่จำเป็นต้องใช้ค่าคุณลักษณะที่แท้จริง ซึ่งแทนที่ด้วยค่าความคล้ายคลึงของข้อมูล ความคล้ายคลึงกันเหล่านี้สามารถมองเป็น Kernel ฟังก์ชั่น $K(\bar{X},\bar{Y})$ ซึ่งจะวัดความคล้ายคลึงกันระหว่างจุด $\bar{X}$ และ $\bar{Y}$ แนวคิดอาจกล่าวได้ว่า Kernel ฟังก์ชั่นอาจเป็นผลคูณระหว่างคู่ของจุดในพื้นที่ที่แปลงใหม่ (แสดงโดยการแม็ปฟังก์ชั่น $\Phi(\cdot)$)

$$K(\bar{X},\bar{Y}) = \Phi(\bar{X}) \cdot \Phi(\bar{Y}) \tag{3.38}$$

ดังนั้นการคำนวณทั้งหมดสามารถทำได้ในพื้นที่เดิมโดยใช้ผลการด็อตตามนัยของ Kernel ฟังก์ชัน ซึ่งในงานวิจัยนี้ใช้ Gaussian Radial Basis Kernel ดังนี้

$$K(\bar{X}_i,\bar{X}_j) = e^{-\left|\left|\bar{X}_i-\bar{X}_j\right|\right|^2/2\sigma^2} \tag{3.39}$$

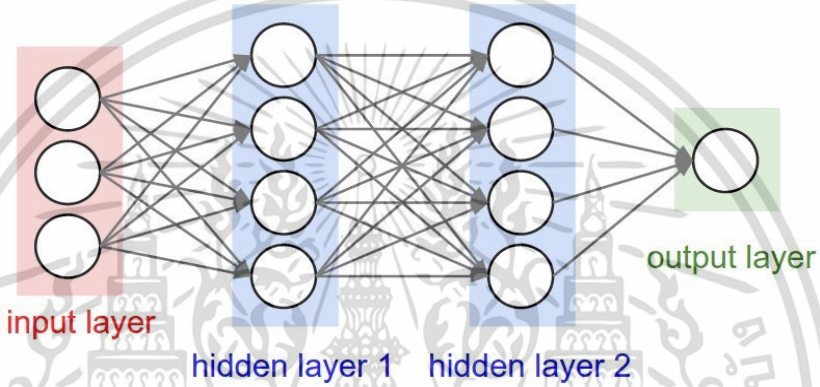
โดยที่ $\bar{X}_i$ คืออินพุทเวกเตอร์

$\bar{X}_j$ คืออินพุทเวกเตอร์

σ คือค่าส่วนเบี่ยงเบนมาตรฐาน

3.6 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึก มีความคล้ายกับโครงข่ายประสาทเทียม (Network of Neuron) ซึ่ง Deep learning เป็นส่วนหนึ่งของการเรียนรู้ด้วยเครื่อง (Machine Learning) โดยถูกสร้างขึ้นจากการนำเอา Neural Network หลายๆ layer มาต่อกัน เริ่มจาก layer แรกสุดทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้าย (Output layer) ทำหน้าที่ส่งผลลัพธ์ที่ประมวลผลออกมา ส่วน layer ระว่างเลเยอร์แรกสุด และ เลเยอร์สุดท้าย จะถูกเรียกว่า Hidden layer ดังรูปที่ 3.20



รูปที่ 3.20 โครงข่ายประสาทเทียม

VGG19 [44] เป็นเครือข่ายประสาทเทียมที่ได้รับการฝึกฝนบนภาพมากกว่าล้านภาพจากฐานข้อมูล ImageNet ซึ่งสามารถจำแนกภาพภาพวัตถุได้ 1,000 ประเภท เช่นแป้นพิมพ์ เมาส์ ดินสอ และสัตว์หลายชนิด เป็นต้น ส่งผลให้เครือข่ายได้เรียนรู้การแสดงคุณสมบัติที่หลากหลายสำหรับภาพที่หลากหลาย เริ่มจากเครือขายมีภาพอินพุตเป็นภาพสี (RGB) ขนาด 224*224 ในการฝึกฝนภาพภาพอินพุตถูกส่งผ่านสแต็คของชั้น convolutional (conv.) ซึ่งเป็นตัวกรองที่มีขนาด 3 × 3 การเลื่อนของตัวกรองถูกกำหนดให้เลื่อนทีละ 1 พิกเซล และมีเลเยอร์การทำ pooling เพื่อลดขนาด spatial dimensions (ความกว้าง x ความยาว) ของอินพุตให้มีขนาดเล็กลง (down-sampling) ด้วยตัวกรองขนาด 2 × 2 เพื่อนำไปคำนวณในเลเยอร์ถัดไป จากนั้นเลเยอร์ Full-Connected (FC) ประกอบด้วยสามเลเยอร์ ซึ่งเลเยอร์ FC ที่ 1 (fc6 ดังตารางที่ 3.1) ,เลเยอร์ FC ที่ 2 (fc7 ดังตารางที่ 3.1) จะมีเลเยอร์ละ 4096 แชนเนล (4096 คุณลักษณะ) และเลเยอร์ FC ที่ 3 (fc8 ดังตารางที่3.1) ทำการจำแนก ILSVRC ออกเป็น 1000 กลุ่มสำหรับแต่ละคลาส เลเยอร์ที่มีทั้งหมดจะถูกนำมาคำนวณแบบไม่เชิงเส้นในขั้นตอน ReLU (nonlinear layer) เพื่อเพิ่มประสิทธิภาพในการคำนวณ (ReLU [45])

ชั้นสุดท้ายคือชั้น soft-max เป็นเลเยอร์สุดท้ายเพื่อให้เอาท์พุท (output) ออกมาเป็นความน่าจะเป็น (probability) ไปคำนวณ Negative Log Likelihood เป็น Cross Entropy Loss โดยผลลัพธ์จากขั้นตอนนี้ได้ออกมาว่าภาพอินพุตอยู่ในกลุ่มใด (classification)

ตารางที่ 3.1 แสดงเลเยอร์ของการเรียนรู้เชิงลึก VGG19

1 'input'	Image Input	224x224x3 images with 'zero center' normalization
2 'conv1_1'	Convolution	64 channels ของ convolutions ขนาด 3x3x3 with stride [1 1] and padding [1 1 1 1]
3 'relu1_1'	ReLU	ReLU
4 'conv1_2'	Convolution	64 channels ของ convolutions ขนาด 3x3x3 with stride [1 1] and padding [1 1 1 1]
5 'relu1_2'	ReLU	ReLU
6 'pool1'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
7 'conv2_1'	Convolution	128 channels ของ convolutions ขนาด 3x3x64 with stride [1 1] and padding [1 1 1 1]
8 'relu2_1'	ReLU	ReLU
9 'conv2_2'	Convolution	128 channels ของ convolutions ขนาด 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
10 'relu2_2'	ReLU	ReLU
11 'pool2'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
12 'conv3_1'	Convolution	256 channels ของ convolutions ขนาด 3x3x128 with stride [1 1] and padding [1 1 1 1]
13 'relu3_1'	ReLU	ReLU
14 'conv3_2'	Convolution	256 channels ของ convolutions ขนาด 3x3x256 with stride [1 1] and padding [1 1 1 1]
15 'relu3_2'	ReLU	ReLU
16 'conv3_3'	Convolution	256 channels ของ convolutions ขนาด 3x3x256 with stride [1 1] and padding [1 1 1 1]
17 'relu3_3'	ReLU	ReLU
18 'conv3_4'	Convolution	256 channels ของ convolutions ขนาด 3x3x256 with stride [1 1] and padding [1 1 1 1]
19 'relu3_4'	ReLU	ReLU
20 'pool3'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]

ตารางที่ 3.1 แสดงเลเยอร์ของการเรียนรู้เชิงลึก VGG19

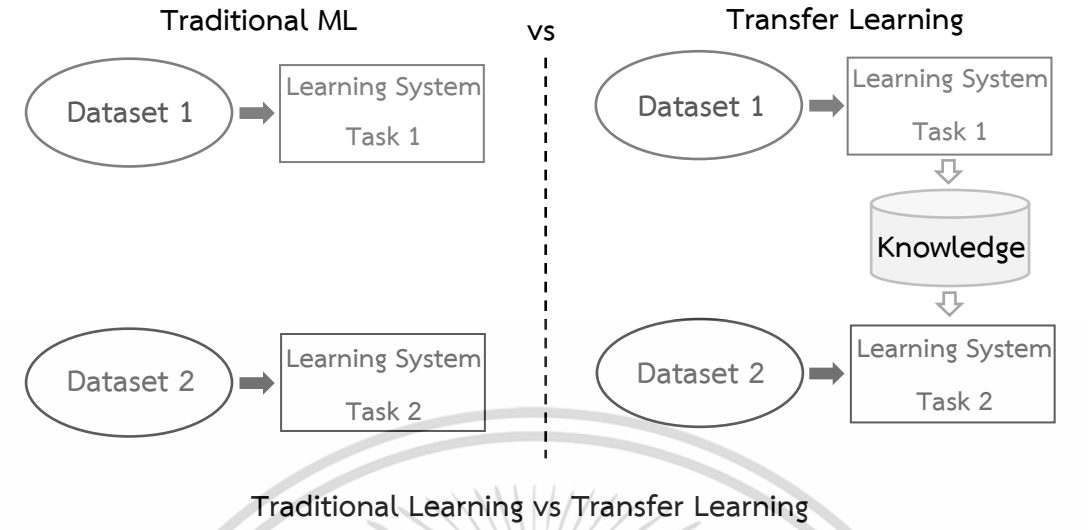
21 'conv4_1'	Convolution	512 channels ของ convolutions ขนาด 3x3x256 with stride [1 1] and padding [1 1 1 1]
22 'relu4_1'	ReLU	ReLU
23 'conv4_2'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
24 'relu4_2'	ReLU	ReLU
25 'conv4_3'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
26 'relu4_3'	ReLU	ReLU
27 'conv4_4'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
28 'relu4_4'	ReLU	ReLU
29 'pool4'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
30 'conv5_1'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
31 'relu5_1'	ReLU	ReLU
32 'conv5_2'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
33 'relu5_2'	ReLU	ReLU
34 'conv5_3'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
35 'relu5_3'	ReLU	ReLU
36 'conv5_4'	Convolution	512 channels ของ convolutions ขนาด 3x3x512 with stride [1 1] and padding [1 1 1 1]
37 'relu5_4'	ReLU	ReLU
38 'pool5'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
39 'fc6'	Fully Connected	4096 nodes of fully connected layer
40 'relu6'	ReLU	ReLU
41 'drop6'	Dropout	50% dropout
42 'fc7'	Fully Connected	4096 nodes of fully connected layer
43 'relu7'	ReLU	ReLU

ตารางที่ 3.1 แสดงเลเยอร์ของการเรียนรู้เชิงลึก VGG19

44 'drop7'	Dropout	50% dropout
45 'fc8'	Fully Connected	1000 nodes of fully connected layer
46 'prob'	Softmax	softmax
47 'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

3.7 การถ่ายโอนความรู้ (Transfer learning)

การเรียนรู้แบบดั้งเดิม (Traditional Learning) นั้น เป็นการถูกสอนสำหรับงานชุดข้อมูลและการฝึกฝนแบบแยกตัวเพื่อให้ได้โมเดลที่เฉพาะของงาน ไม่มีความรู้ถูกเก็บไว้สำหรับความสามารถถ่ายโอนจากโมเดลหนึ่งไปยังอีกแบบหนึ่ง ในการเรียนรู้การถ่ายโอนทำให้สามารถยกระดับความรู้ (คุณลักษณะ (features) น้ำหนัก (weights) และอื่น ๆ) จากโมเดลที่ผ่านการฝึกฝนมาก่อนให้สามารถแก้ไขปัญหาสำหรับงานใหม่โดยใช้ข้อมูลในการฝึกฝนที่น้อยลง [46] ดังรูปที่ 3.21 แสดงความแตกต่างระหว่างการเรียนรู้แบบดั้งเดิมกับการถ่ายโอนการเรียนรู้ (Transfer Learning) เช่น ฝั่งทางซ้ายมือเป็นการเรียนรู้ดั้งเดิม สมมติดังตัวอย่าง เช่น มีชุดข้อมูลสำหรับฝึกฝน โมเดลสำหรับระบุวัตถุภายในภาพในโดเมนร้านอาหารที่จำกัด และได้รับการปรับให้ทำงานได้ดี แต่เมื่อนำมาใช้กับข้อมูลที่ไม่เคยเจอ แม้มาจากโดเมนเดียวกัน (ร้านอาหาร) การดำเนินการเรียนรู้แบบดั้งเดิมกลับล้มเหลว เนื่องจากไม่ได้รับตัวอย่างสำหรับการฝึกฝนสำหรับงานนั้นที่เพียงพอ ความสำเร็จจากการรับโดเมนมาคือ การตรวจจับวัตถุจากภาพในสวนสาธารณะหรือร้านกาแฟจะสามารถใช้โมเดลที่ถูกเทรนจากร้านอาหารได้ การถ่ายโอนการเรียนรู้จะทำให้สามารถใช้ความรู้จากงานที่เรียนรู้ก่อนหน้านี้และนำไปใช้กับงานใหม่ที่เกี่ยวข้อง หากเรามีข้อมูลมากขึ้นสำหรับงานหนึ่ง เราอาจใช้การเรียนรู้และสรุปความรู้นี้ (คุณสมบัติ, น้ำหนัก) สำหรับงานอื่นได้ (โดยใช้จำนวนข้อมูลสำหรับฝึกฝนน้อยลงอย่างมีนัยสำคัญ) ในกรณีที่เกิดปัญหาในโดเมนของคอมพิวเตอร์วิชัน คุณสมบัติบางอย่างในระดับต่ำเช่น ขอบ รูปร่าง มุม และความเข้มแสง สามารถประยุกต์ใช้ร่วมกันข้ามงาน และทำให้สามารถถ่ายโอนความรู้ระหว่างงานได้ ซึ่งความรู้จากงานที่มีอยู่ทำหน้าที่เป็นข้อมูลเพิ่มเติมเมื่อเรียนรู้งานของเป้าหมายใหม่



รูปที่ 3.21 การเปรียบเทียบระหว่างการเรียนรู้แบบดั้งเดิมและการเรียนรู้แบบถ่ายโอนความรู้

3.8 Ensemble CNN [47,50]

การฝึกฝนเครือข่ายประสาทเทียมเชิงลึก (Training deep neural networks) อาจมีราคาแพงมากในการคำนวณ เครือข่ายที่ลึกมากที่ได้รับการฝึกฝนจากตัวอย่างนับล้านอาจต้องใช้เวลาเป็นวัน สัปดาห์ และบางครั้งเป็นเดือน เช่น โมเดลพื้นฐานของ Google [48] เป็นโครงข่ายประสาทเทียมเชิงลึก ซึ่งได้รับการฝึกฝนมาประมาณหกเดือนโดยใช้การไล่ระดับสีสุ่มแบบอะซิงโครนัสบนแกนจำนวนมาก (asynchronous stochastic gradient descent) หลังจากทุ่มเทเวลาและทรัพยากรไปมากขนาดนี้ ไม่สามารถรับประกันว่าโมเดลสุดท้ายจะมีข้อผิดพลาดระดับต่ำ (low generalization error) และทำงานได้ดีกับตัวอย่างที่ไม่ได้เห็นในระหว่างการฝึกฝนต่างๆมากมาย หลากหลายพารามิเตอร์ถูกปรับ จากนั้นเลือกเครือข่ายที่ดีที่สุดและละทิ้งส่วนที่เหลือ อย่างไรก็ดี วิธีดังกล่าวมีข้อเสียสองประการคือ ประการแรก ความพยายามทั้งหมดที่เกี่ยวข้องกับการฝึกฝนเครือข่ายที่เหลือนั้นสูญเปล่า ประการที่สอง เครือข่ายที่มีประสิทธิภาพดีที่สุดในชุดการตรวจสอบอาจไม่ใช่เครือข่ายที่มีประสิทธิภาพดีที่สุดในข้อมูลการทดสอบใหม่ [49] โมเดลโครงข่ายประสาทเทียมวิธีที่ไม่เชิงเส้นนี้สามารถเรียนรู้ความสัมพันธ์ที่ไม่เชิงเส้นที่ซับซ้อนในข้อมูลได้ แต่ยังมีข้อจำกัดของความยืดหยุ่นเนื่องจากมีความไวต่อสภาวะเริ่มต้น ทั้งในแง่ของน้ำหนักสุ่มเริ่มต้นและในแง่ของสัญญาณรบกวนทางสถิติ (statistical noise) ในชุดข้อมูลการฝึก ลักษณะการสุ่มของอัลกอริธึมการเรียนรู้นี้ ทุกครั้งที่มีการฝึกโมเดลโครงข่ายประสาทเทียม อาจเรียนรู้เวอร์ชันที่แตกต่างกันเล็กน้อย (หรืออย่างมาก) ของ

ฟังก์ชันการแมพจากอินพุตไปยังเอาต์พุต ซึ่งจะมีผลต่อประสิทธิภาพที่แตกต่างกันในการฝึกและชุดข้อมูลที่นำมาจัดการ (holdout datasets)

ด้วยเหตุนี้ เราจึงใช้โครงข่ายประสาทเทียมด้วยวิธี an Ensemble of Models คือการฝึกโมเดลที่เรียนรู้หลายตัวและรวมการคาดการณ์เข้าด้วยกัน แนวคิดการรวมการทำนายจากโมเดลที่ดีหลายแบบ แต่ให้ค่าความผิดพลาดของการทำนายแตกต่างกัน ทำให้ไม่สร้างข้อผิดพลาดแบบเดียวกันทั้งหมดในชุดทดสอบ

ซึ่งข้อผิดพลาดในการทำนายของโมเดลการเรียนรู้ของเครื่องสามารถอธิบายได้จากอคติ (bias) และความแปรปรวน (variance)โดยข้อผิดพลาดอันเนื่องมาจากอคติหมายถึงค่าของความแตกต่างระหว่างผลการทำนายและผลลัพธ์ที่แท้จริง แต่ละอัลกอริธึมเริ่มต้นด้วยอคติจำนวนหนึ่งที่เกิดขึ้นจากเงื่อนไขของการสันนิษฐานภายในโมเดล ซึ่งทำให้ฟังก์ชันเป้าหมายให้ผลการเรียนรู้แตกต่างจากผลลัพธ์ที่แท้จริง หากอคติ (ข้อผิดพลาด) สูงจะส่งผลให้ประสิทธิภาพการทำนายต่ำ ไม่สามารถทำงานได้ดีกับข้อมูลใหม่ ความแปรปรวนบ่งบอกถึงความแตกต่างของผลการทำนายกับค่าเฉลี่ย ซึ่งสามารถใช้อธิบายความสามารถของโมเดลในการปรับให้เข้ากับการเปลี่ยนแปลงข้ามชุดฝึกฝน ความแปรปรวนมีทั้งความแปรปรวนต่ำหรือความแปรปรวนสูง ความแปรปรวนต่ำหมายความว่ามีความแตกต่างเล็กน้อยในการคาดการณ์ของฟังก์ชันเป้าหมายเมื่อมีการเปลี่ยนแปลงชุดข้อมูลการฝึกฝน ในทางกลับกันความแปรปรวนสูงแสดงความผันแปรอย่างมากในการคาดคะเนของฟังก์ชันเป้าหมายเมื่อมีการเปลี่ยนแปลงชุดข้อมูลการฝึก ดังนั้นอาจกล่าวได้ว่าหากความแปรปรวนสูง จะส่งผลให้เกิดการ over-fitting เนื่องจากโมเดลมีความสามารถในการเรียนรู้ชุดข้อมูลฝึกฝนชุดหนึ่ง (มีอคติต่ำ) แต่ไม่อาจเรียนรู้ได้ดีเมื่อเปลี่ยนชุดข้อมูลไป (เกิดอคติสูง) ดังนั้นจุดประสงค์หลักของการทำ Ensemble คือ การลดความแปรปรวนและโมเดลยังคงมีอคติน้อยลง

นอกเหนือจากการลดความแปรปรวนในการทำนายแล้ว โมเดล ensemble ยังสามารถให้ผลลัพธ์ในการทำนายได้ดีกว่าโมเดลที่ดีที่สุดตัวใดตัวหนึ่ง โดยจะเริ่มจากโมเดลต่างๆที่ฝึกฝนเฉพาะทางทำการคาดการณ์ก่อน จากนั้นรวมการทำนายเข้าสู่โมเดล ensemble เพื่อสร้างผลลัพธ์การคาดการณ์ขั้นสุดท้าย

CNN ของงานวิจัยนี้ประกอบไปด้วย ตัวกรอง (filter) หรือ เคอร์เนล (kernel) ที่ช่วยดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออกมา โดยปกติตัวกรองอันหนึ่งจะดึงคุณลักษณะที่สนใจออกมาได้หนึ่งอย่าง เราจึงจำเป็นต้องใช้ตัวกรองหลายตัวกรองเพื่อหาคุณลักษณะหลายอย่างประกอบกัน ซึ่งจะมี Stride เป็นตัวกำหนดว่าตัวกรอง (filter) จะถูกเลื่อนไปด้วย Step เท่าไร (เช่นการกำหนดค่าของ Stride ให้มากขึ้น คือต้องการให้การคำนวนหาคุณลักษณะมีพื้นที่ทับซ้อนกันน้อยลง แต่อย่างไรก็ตาม

การกำหนดค่าของ Stride ที่มากขึ้นจะทำให้ได้ผังคุณลักษณะ (feature map) ที่มีขนาดเล็กลง และ Padding คือการเติมพื้นที่เข้าไปรอบ dimension ของ input โดยอาจจะเป็น 0 หรือค่าต่างๆเข้าไป เนื่องจากในบางปัญหา Input ที่อยู่ตามขอบภาพอาจมีความสำคัญที่ส่งผลต่อการตัดสินใจบางอย่าง เราจึงจำเป็นต้องเก็บคุณลักษณะตามขอบของ input ไว้ด้วย ตัวกรองจะถูกทาบลงในพิกเซลแรกของภาพข้อมูลเข้า จากนั้นจะถูกเลื่อนไปทาบบนพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซลในภาพ (เราอาจจะไม่ทาบตัวกรองบนพิกเซลที่อยู่ใกล้กรอบภาพ เพราะตัวกรองจะล้นออกไปนอกภาพ) เมื่อตัวกรองถูกเลื่อนไปเรื่อยๆจนครบทุกพิกเซลที่สามารถเลื่อนได้ สิ่งที่เราได้นั้นจะเป็นสิ่งที่เรียกว่า ผังคุณลักษณะ (feature map) ในการเรียนรู้จำเป็นต้องมีข้อมูลทั้งหยาบและละเอียดควบคู่กันไป จึงจำเป็นต้องคำนวณภาพในหลายสเกล แต่ปัญหาที่สำคัญคือเราจะทำให้การคำนวณอยู่ในรูปหลายสเกลได้อย่างไร หากเราใช้ตัวกรองขนาด m x n จัดการกับรายละเอียดเล็กๆ (ภาพใหญ่มีรายละเอียดมาก จึงถือว่าเป็นสเกลละเอียด) แต่ด้วยตัวกรองขนาดเท่าเดิม หากทำกับภาพที่ขนาดเล็กลงแล้ว มันจะครอบคลุมพื้นที่วัตถุเดิมมากขึ้น ดังนั้นโครงข่ายของเราควรจะต้องมีการย่อรูปประกอบด้วย เราก็จะสามารถเข้าถึงความสามารถด้านการวิเคราะห์หลายความละเอียดได้ Pooling คือความสามารถในการย่อรูปแบบหนึ่ง ซึ่งเราใช้ max Pooling เป็นตัวกรองแบบหนึ่งที่หาค่าสูงสุดในบริเวณที่ตัวกรองทาบอยู่มาเป็นผลลัพธ์ โดยเราจะเตรียมตัวกรองในลักษณะเดียวกับการทำ Feature Extraction ของ CNN มาทาบบนข้อมูลแล้วเลือกค่าที่สูงที่สุดบนตัวกรองนั้นมาเป็นผลลัพธ์ใหม่ และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนดไว้ โดยขนาดตัวกรองของการทำ max pooling นิยมเรียกว่า pool size

นอกจากนี้เราเพิ่มเลเยอร์การ dropout คือการปิด Neuron บางตัวในเครือข่ายแต่ละชั้น เป็นการเลือกปิดแบบสุ่ม ซึ่งเทคนิคนี้ใช้งานได้ดี เพราะเราไม่ได้ไปปิด Input แต่ไปปิดหน่วยประมวลผลบางตัว การเลือกปิดแบบสุ่มนี้จะป้องกันการเกิด Co-adaptation และเป็นการลด Overfitting เพราะทำให้โมเดลนั้นง่ายลง แต่ไม่ได้จงใจกดข้อมูลลักษณะใดลักษณะหนึ่งลง เป็นการสุ่มกดเพื่อลดความซับซ้อนของทั้งระบบ ในทางเทคนิค วิธีการทำ Dropout คือการสร้าง Boolean matrix มิติเท่ากับ Matrix ของ Activation ที่ต้องการใช้ Dropout โดยสุ่มค่า Yes/No ตามสัดส่วนที่เรากำหนด เช่น ถ้ากำหนดให้ปิด k% ก็คือให้ทั้ง Matrix มีค่า True 100-k% ค่า False k% จากนั้นนำ Dropout matrix นี้ไปคูณแบบ Element-wise เข้ากับ Activation matrix จะได้ Activation matrix ใหม่ที่บาง Element มีค่าเท่ากับศูนย์ในตำแหน่งเดียวกับที่ใน Dropout matrix มีค่าเป็น False จากนั้นนำสัดส่วนการปิดไปหารแบบ Element-wise ออกจาก Activation matrix ใหม่ เพื่อทำการ Normalize โดยกระบวนเรียนรู้จะถูกทำซ้ำเพื่อปรับค่าพารามิเตอร์หลายๆรอบ (Epoch) เพื่อให้ค่า Error จากการทำนายลดลงในแต่ละรอบ

3.9 การหาค่าความผิดพลาดเฉลี่ยกำลังสอง (Avereage RMSE) [49]

การหาค่าความผิดพลาดเฉลี่ยกำลังสองเป็นการวัดประสิทธิภาพผลลัพธ์กับคำตอบ

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{3.40}$$

เมื่อ N คือจำนวนค่าความผิดพลาด

x_i = observed values

$\hat{x}_i$ = forecasts

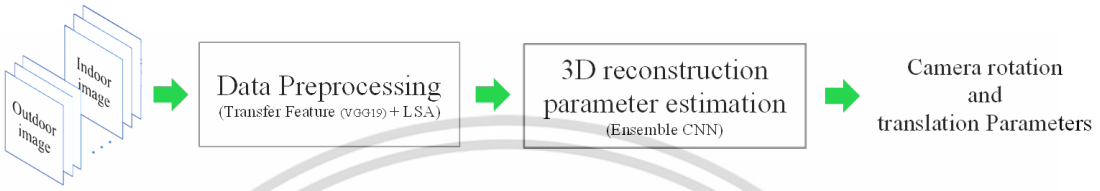
บทที่ 4

งานวิจัยที่นำเสนอ

ในงานวิจัยนี้เรานำเสนอการประมาณค่าพารามิเตอร์มุมหมุนและการเลื่อนของภาพถ่ายสองมิติ เพื่อนำมาใช้ในการประกอบกลับภาพ 3 มิติ โดยอาศัยการทำการถ่ายโอนการเรียนรู้ (Transfer learning) จากโมเดลการเรียนรู้เชิงลึกแบบ CNN อย่างไรก็ดีด้วยข้อจำกัดของการฝึกฝน CNN โมเดลเดี่ยวซึ่งเรียนรู้คุณลักษณะเฉพาะเชิงพื้นที่ การจะฝึกฝนโมเดล CNN ให้สามารถเรียนรู้ข้อมูลหลากหลายโดเมน และสามารถวิเคราะห์ผลลัพธ์ให้มีประสิทธิภาพ โดยไม่ขึ้นกับความแตกต่างของคุณลักษณะเฉพาะแต่ละโดเมนนั้นทำได้ยาก เนื่องจากการเรียนรู้คุณลักษณะของภาพ มุมมองการถ่ายภาพ และพารามิเตอร์ที่เกี่ยวข้องกับการรับภาพ (Acquisition) ที่แตกต่างกันอย่างมาก มีผลให้คุณลักษณะเชิงพื้นที่ที่วิเคราะห์ได้จากโมเดลแตกต่างกันไป ทำให้การฝึกฝนต้องการข้อมูลปริมาณมาก และเมื่อมีข้อมูล domain ใหม่เข้ามา จะต้อง retrain ข้อมูลทั้งหมดอีกครั้ง ส่งผลให้ใช้เวลาในการพัฒนาสูง เราจึงได้นำเสนอโมเดลโครงข่ายเอเจนท์การเรียนรู้ ที่เรียกว่า Ensemble CNN โดยจะแบ่งเป็นเอเจนท์ที่เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) และเอเจนท์ที่เรียนรู้ความสัมพันธ์ระหว่างโดเมน (domain relationship agents) เพื่อให้เกิดความยืดหยุ่นต่อการเรียนรู้หลากหลายโดเมนข้อมูลภาพ ในการวิเคราะห์และประมาณพารามิเตอร์ที่ต้องการ โดยมีการศึกษาและเปรียบเทียบประสิทธิภาพของ โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR model, โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN, โมเดลเดี่ยวแบบฝึกฝนร่วมหลายโดเมนด้วย CNN และแบบโครงข่ายเอเจนท์ด้วย CNN model โดยโมเดลที่กล่าวมา จะนำไปใช้เปรียบเทียบผลการประมาณค่าพารามิเตอร์มุมหมุนและการเลื่อนของกล้อง ในรูปแบบของการวิเคราะห์เชิงโมเดลเดี่ยว (based-line end-to-end single model) และ รูปแบบการวิเคราะห์เชิงโครงข่ายเอเจนท์ (Ensemble model) ที่นำเสนอ สำหรับโดเมนภาพในร่มและกลางแจ้ง ที่มีรูปแบบการรับภาพและคุณลักษณะของภาพที่แตกต่างกันอย่างมาก

4.1 ภาพรวมกระบวนการทำงานของระบบ

ระบบที่นำเสนอประกอบด้วย 2 ส่วนหลัก คือ ส่วนของ preprocessing data เพื่อดึงคุณลักษณะที่สำคัญจากภาพ และส่งมายังส่วนที่ 2 เข้าสู่โมเดลแต่ละแบบ เพื่อประมาณค่าพารามิเตอร์การสร้างกลับภาพ 3 มิติ ดังแสดงในรูปที่ 4.1



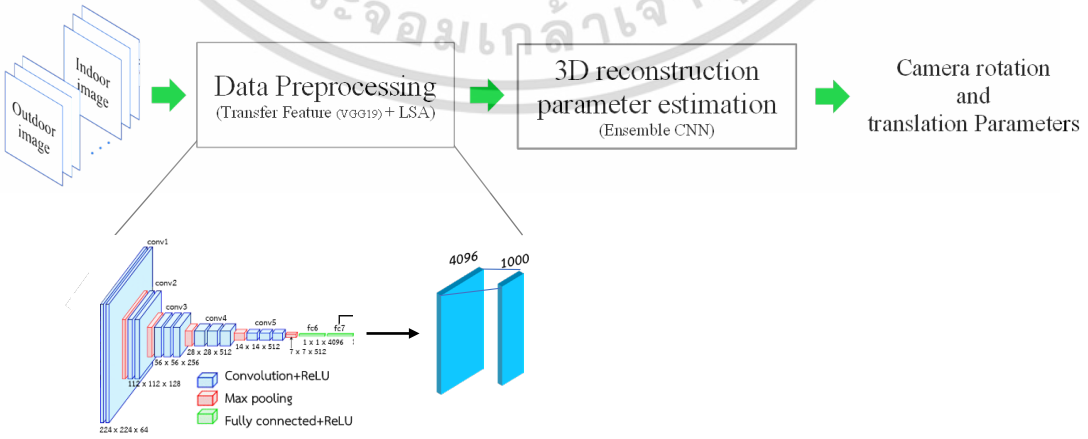
รูปที่ 4.1 ภาพรวมระบบการ 3D reconstruction parameter estimation

ในส่วนของ Preprocessing data เราได้ทำการถ่ายโอนคุณลักษณะ (transfer feature) ออกมาจาก Pre-trained CNN Model โดยใช้ต้นแบบเป็น VGG19 ที่ถูกฝึกฝนด้วยข้อมูล ImageNet [45] โดยขั้นแรกภาพอินพุทจะถูกนำมาสุ่มแยกเพื่อแบ่งเป็นชุดข้อมูลฝึกฝนและชุดข้อมูลทดสอบ จากนั้นนำแต่ละชุดมาสกัดคุณลักษณะที่มีความสำคัญของแต่ละชุดข้อมูลออกมา การสกัดคุณลักษณะ (Feature Extraction) คือกระบวนการสกัดคุณลักษณะเด่น หรือคุณลักษณะที่สำคัญของข้อมูลออกมาเพื่อช่วยอำนวยความสะดวกในการนำไปใช้ทำนายมุมหมุนและการเลื่อนของภาพนั้นในภายหลัง ในเอกสารฉบับนี้ใช้การสกัดคุณลักษณะสำคัญของมุมหมุนและการเลื่อนจากภาพด้วยโมเดลการเรียนรู้เชิงลึก VGG19 ซึ่งเดิมถูกฝึกฝนและใช้เพื่อสกัดคุณลักษณะจากภาพสองมิติสำหรับงานการรู้จำวัตถุในภาพ แต่ในงานวิจัยนี้จะนำคุณลักษณะที่ได้นั้นมาประยุกต์ใช้เพื่อการเรียนรู้และทำนายมุมหมุนและการเลื่อนสามมิติของภาพ จากการถ่ายโอนการเรียนรู้ จะทำให้สามารถใช้ความรู้จากงานที่เรียนรู้ก่อนหน้านี้ไปใช้กับงานใหม่ที่เกี่ยวข้อง เช่น การเรียนรู้ขอบ รูปร่าง มุม และความเข้ม เป็นต้น จึงนำมาประยุกต์ใช้ร่วมกันข้ามงานดังรูปที่ 4.3 และทำให้สามารถถ่ายโอนความรู้ระหว่างงานได้ ซึ่งงานวิจัยนี้จะถ่ายโอนคุณลักษณะออกมาจาก fully connected layer ที่ 7 โดยมีขนาดมิติของข้อมูล (feature dimension) เท่ากับ 4096

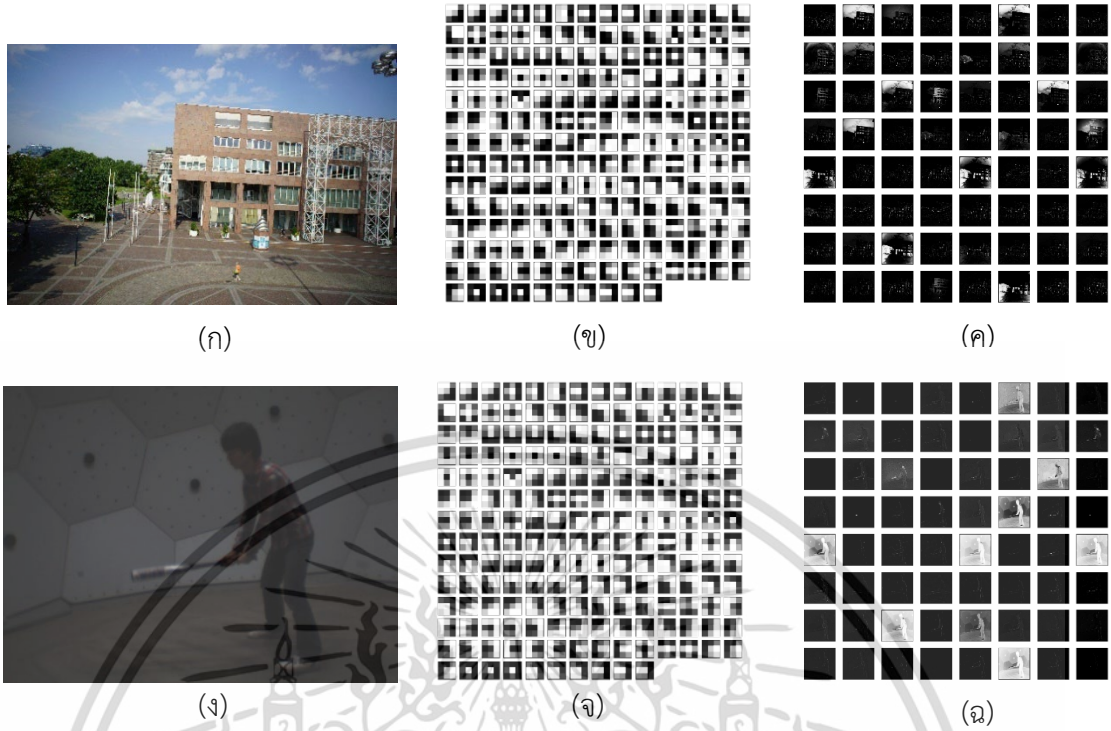
อย่างไรก็ดีข้อมูลคุณลักษณะที่ได้ มีมิติและความซับซ้อนของข้อมูลสูง ทำให้ใช้เวลาในการประมวลผลสูงตามไปด้วย โดยเฉพาะเมื่อภาพมีจำนวนมาก ส่งผลให้การทำนายมุมหมุนและการเลื่อนใช้ระยะเวลานาน ดังนั้นการลดมิติของข้อมูลจะมีส่วนช่วยลดเวลาในการประมวลผลข้อมูลลงได้ อย่างไรก็ดี ถึงแม้ว่าการลดมิติของข้อมูลสามารถทำได้ภายใน CNN แต่ด้วยมิติข้อมูลขนาดใหญ่ทำให้

โครงข่าย CNN จำนวน 4 เลเยอร์ที่ผู้วิจัยออกแบบไว้ ให้ประสิทธิภาพไม่ดีนัก อาจเป็นเพราะคุณลักษณะที่ได้จากการถ่ายโอนการเรียนรู้มีความซับซ้อนสูงอยู่ ทั้งยังเป็นคุณลักษณะที่ถูกเรียนรู้สำหรับงานการจัดกลุ่มข้อมูล หากต้องการนำมาใช้สำหรับงานทำนายแบบถดถอย อาจจำเป็นต้องขยายขนาดโครงข่าย CNN เพื่อให้สามารถเรียนรู้คุณลักษณะที่มีมิติสูงและแมพคุณลักษณะให้เหมาะสมกับงานทำนายแบบถดถอยและมีประสิทธิภาพการทำนายที่แม่นยำเพิ่มขึ้น อย่างไรก็ดี โครงข่าย CNN ที่มีขนาดใหญ่ขึ้น จะส่งผลให้องค์ประกอบของพารามิเตอร์สำหรับชั้นคอนเวอร์ลูชันมีมากขึ้น โมเดลของขั้นตอนการประมาณค่ามุมหมุนและการเลื่อนจึงมีขนาดใหญ่ขึ้นและใช้ระยะเวลาในการฝึกฝนมากยิ่งขึ้น และด้วยข้อจำกัดของอุปกรณ์ที่ผู้วิจัยใช้ในการสอนโครงข่าย CNN ทำให้มีข้อจำกัดในการขยายขนาดมากกว่าที่ออกแบบไว้ จึงตัดสินใจเลือกใช้ เทคนิคการลดมิติของข้อมูล เป็นส่วนช่วยประมวลผลคุณลักษณะก่อนเข้า CNN และผู้วิจัยพบว่าการลดมิติช่วยลดความซับซ้อนของข้อมูลและสามารถทำให้ CNN ที่มีขนาดโครงข่ายตามที่ออกแบบ สามารถเรียนรู้งานทำนายแบบถดถอยได้มีประสิทธิภาพดี และสามารถทำนายมุมหมุนและการเลื่อนได้ถูกต้องมากขึ้นกว่าการไม่ใช้เทคนิคการลดมิติข้อมูล

จากปัจจัยที่กล่าวมา ผู้วิจัยได้เลือกใช้เทคนิคการลดความซับซ้อนของข้อมูลที่ด้วยเทคนิค Latent Semantic Analysis (LSA) ถูกสร้างขึ้นมาจากเทคนิคทางคณิตศาสตร์ชื่อ การแยกค่าแบบเอกฐาน (Singular Value Decomposition : SVD) [37] ด้วยวิธีนี้จะลดจำนวนแถวลงแต่ยังคงรักษาโครงสร้างคุณลักษณะของภาพที่คล้ายกันไว้ ดังรูปที่ 4.4 โดยการแยกค่าแบบเอกฐานเป็นหนึ่งในการแยกเมทริกซ์ที่มีประโยชน์มากที่สุดสำหรับการคำนวณเชิงตัวเลข ซึ่งหลังจากทำการลดมิติของข้อมูลให้เหลือแต่มิติของคุณลักษณะที่สำคัญ แล้วจะส่งไปยังขั้นตอนต่อไป เป็นข้อมูลขาเข้าจำนวน 1000 คุณลักษณะของแต่ละภาพ ดังรูปที่ 4.2 เพื่อประมวลผลการประมาณค่าพารามิเตอร์คำตอบ



รูปที่ 4.2 ขั้นตอน Preprocessing data ของงานวิจัยที่นำเสนอ

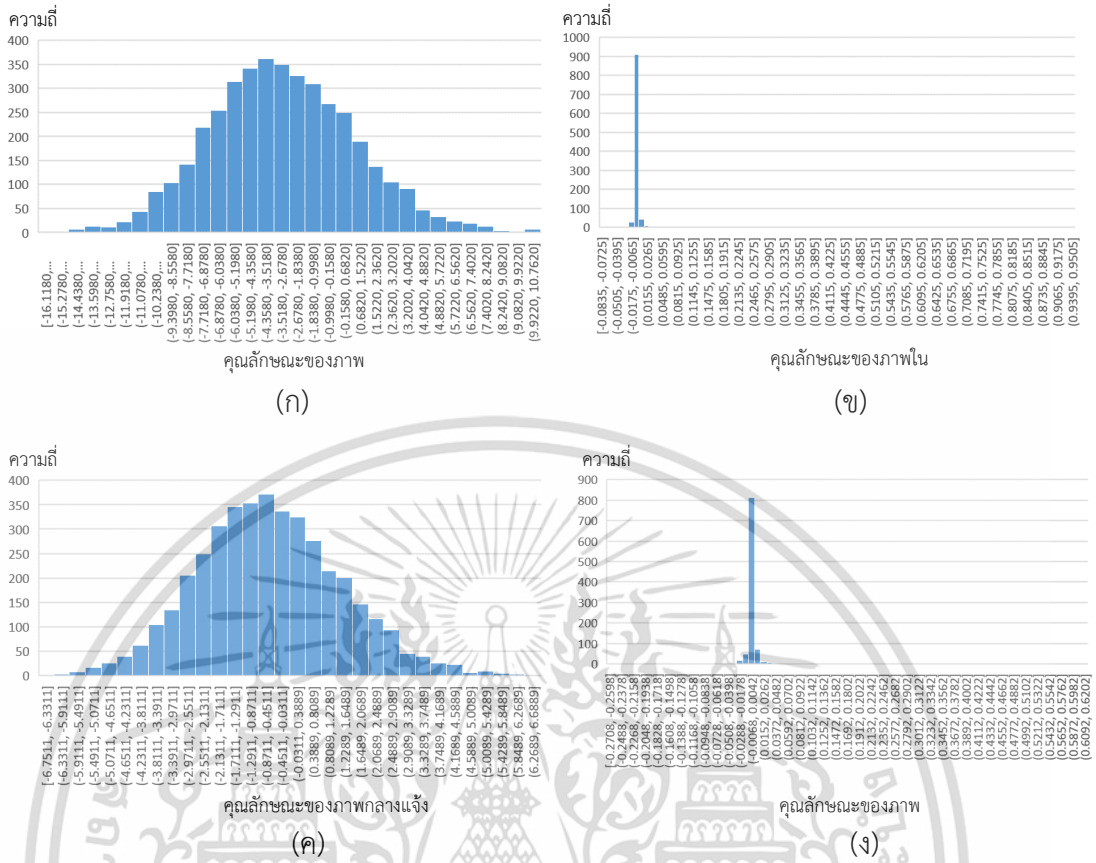


(ก) (ข) (ค)

(ง) (จ) (ฉ)

รูปที่ 4.3 แสดง feature map ของ layer 1 จาก VGG 19 สำหรับชุดภาพกลางแจ้งถ่ายทางอากาศ (Map) [51] และชุดภาพในร่ม [52].

(ก) ชุดภาพถ่ายกลางแจ้งในชุดภาพ rathaus. (ข),(จ) คือตัวกรองแต่ละตัวในเลเยอร์ convolutional layer ที่ 1 มี 64 บล็อกและพล็อต (plot) แต่ละสามแชนแนล (channels) จึงได้ feature map ทั้งหมด 192 รายการ (ตัวกรอง 64 ตัว * 3 แชนแนล) โดยปรับค่าให้อยู่ในช่วง 0-1 สี่เหลี่ยมสีเข้มแสดงถึงค่าน้ำหนักขนาดเล็กหรือแบบยับยั้ง (small or inhibitory weights) และสี่เหลี่ยมสีอ่อนแสดงถึงค่าน้ำหนักขนาดใหญ่หรือแบบกระตุ้น (large or excitatory weights). (ง) ชุดภาพถ่ายในร่มของชุดภาพ Batswing. (ค), (ฉ) การบันทึกผลลัพธ์ของการใช้ตัวกรองว่าคุณลักษณะใดของอินพุตที่ตรวจพบหรือเก็บรักษาไว้ใน feature map ซึ่งผลลัพธ์ของการใช้ฟิลเตอร์ในเลเยอร์ convolutional แรกได้รูปภาพอินพุตหลายเวอร์ชันที่มีการเน้นคุณลักษณะที่แตกต่างกัน เช่น เส้นไฮไลท์บางเส้น โฟกัสอื่นๆ ที่พื้นหลังหรือพื้นหน้า เป็นต้น ของภาพกลางแจ้งและในร่ม ตามลำดับ

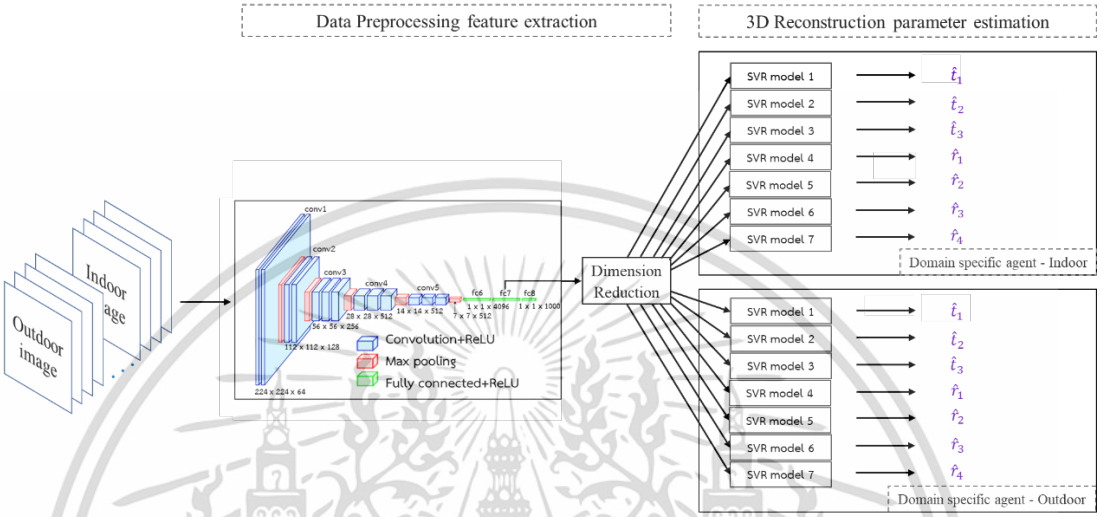


รูปที่ 4.4 แสดงกราฟฮิสโตแกรม (Histogram) ของการลดคุณลักษณะ 4096 คุณลักษณะเหลือแต่มิติของคุณลักษณะที่สำคัญจำนวน 1000 คุณลักษณะ โดย (ก),(ค) กราฟฮิสโตแกรมของคุณลักษณะภาพในร่มและกลางแจ้งที่ได้จาก fc7 ของ VGG19 จำนวน 4096 คุณลักษณะ (ข),(ง) กราฟฮิสโตแกรมของคุณลักษณะ (ก),(ค) ที่ผ่านขั้นตอนการลดจำนวนมิติของคุณลักษณะด้วยเทคนิค LSA ให้เหลือ 1000 คุณลักษณะตามลำดับ

ในงานวิจัยนี้เราทำการศึกษาเปรียบเทียบประสิทธิภาพโมเดลเรียนรู้ในส่วนที่ 2 แบบ Ensemble CNN ที่นำเสนอ (Fig. 4.10) กับ โมเดลเดี่ยวแบบฝึกฝนร่วมหลายโดเมน (Fig. 4.9), โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN (Fig. 4.7) และโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR (Fig. 4.5) โดยทั้ง 4 โมเดล จะมีส่วนของ Data Preprocessing เหมือนกัน ต่างกันเพียงสถาปัตยกรรมของการประมาณค่ามุมหมุนและการเลื่อนในขั้นตอน 3D reconstruction parameter estimation

4.2 โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR

ในขั้นตอน 3D reconstruction parameter estimation นี้ใช้ SVR ในการประมาณค่ามุมหมุนและการเลื่อน โดยการฝึกฝนแยกโดเมน Indoor และ outdoor ดังรูปที่ 4.5



รูปที่ 4.5 โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR

คุณลักษณะของแต่ละโดเมนจากขั้นตอน data preprocessing ถูกส่งเข้ามาใน model SVR ที่เรียนรู้และทำนายคำตอบของมุมหมุนและการเลื่อนแต่ละตัว การทำงานของ Grid-search นั้นคือการปรับพารามิเตอร์ค่าต่าง ๆ เข้าไป Fit ในโมเดลหลาย ๆ รอบแล้วเก็บค่าแต่ละครั้งไว้ เพื่อหาค่าที่ทำให้ความแม่นยำของโมเดลสูงที่สุดในการประมาณค่าคำตอบของโมเดลต่อไป ซึ่งในแต่ละโมเดลมีค่าพารามิเตอร์ที่ต้องการแตกต่างกัน พารามิเตอร์ที่มีความสำคัญต่อการ Fit โมเดลได้แก่ kernel , epsilon, C และ gamma ซึ่งอาจไม่จำเป็นต้องปรับพารามิเตอร์ทุกตัว ดังนั้นงานวิจัยของเราจึงได้ทำการปรับพารามิเตอร์ kernel = ‘rbf’, C = [0.001, 0.01, 0.1, 1, 10, 100, 1000], gamma = [0.0001, 0.001, 0.01, 0.1, 1.0, 10, 100]

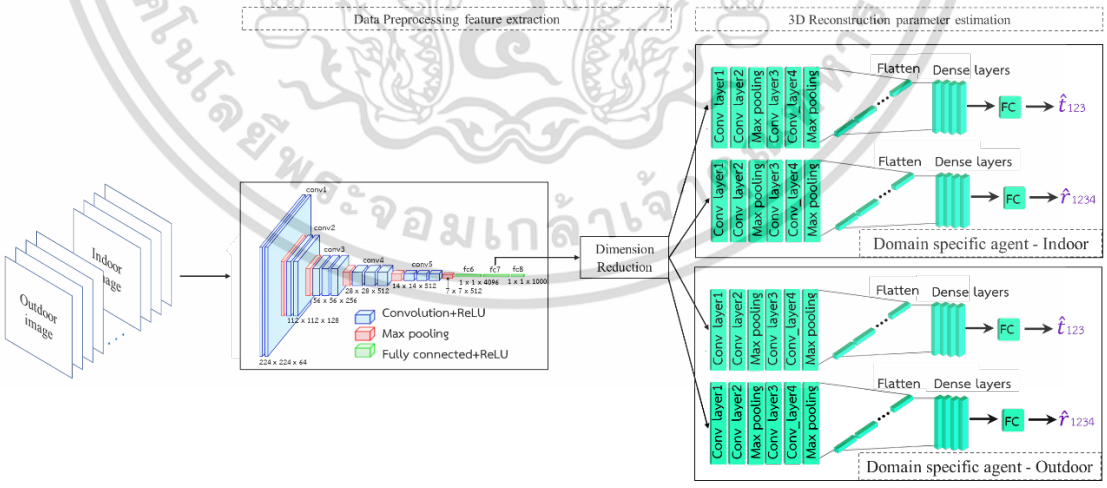
พารามิเตอร์มุมหมุนและการเลื่อนที่ถูกทำนายของภาพประกอบด้วย 7 คำตอบ ดังนั้นในการทำนายพารามิเตอร์ของภาพ 1 ภาพ จะมี 7 model เมื่อทำนายมุมหมุนและการเลื่อนของชุดทดสอบแล้ว นำผลลัพธ์มาตรวจสอบด้วยคำตอบ จากนั้นหาค่าความผิดพลาดด้วยค่าความผิดพลาดเฉลี่ยกำลังสอง โดยการนำข้อมูลเข้าเพื่อทำนายและวัดประสิทธิภาพผลลัพธ์ของแต่ละคำตอบแสดงดังรูปที่ 4.6

```
from sklearn.svm import SVR
for each parameter of rotation and translation in [r1, r2, r3, r4, t1, t2, t3]
    for C in [0.001,0.01,0.1,1,10,100,1000]
        for gamma in [0.0001,0.001,0.01,0.1,1.0,10,100]
            svr_rbf    = SVR (kernel = ' rbf ', C , gamma )
            answer of each image = svr_rbf.fit ( X, Y)
            calculate answer of each image with groundtruth by using RMSE
```

รูปที่ 4.6 การนำข้อมูลเข้าเพื่อทำนายและวัดประสิทธิภาพผลลัพธ์ของแต่ละคำตอบ

4.3 โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN

การประมาณค่าพารามิเตอร์สำหรับการประกอบกลับสามมิตินี้ใช้ CNN ในการประมาณค่ามุมหมุนและการเลื่อน แบบการฝึกฝนแยกโดเมน Indoor และ Outdoor การประมาณค่าพารามิเตอร์ด้วยวิธีนี้ประกอบด้วย CNN 4 model คือ CNN model ของชุดข้อมูลในร่ม 2 โมเดล และ CNN model ของชุดข้อมูลกลางแจ้ง 2 โมเดล โดยจะแยกเป็นโมเดลทำนายพารามิเตอร์มุมหมุนทั้ง 4 ค่า ($\hat{r}_{1234}$) และโมเดลทำนานพารามิเตอร์การเลื่อน 3 ค่า ($\hat{t}_{123}$) ดังรูปที่ 4.7



รูปที่ 4.7 โมเดลการฝึกฝนแยกโดเมนด้วย CNN

โครงสร้าง CNN ที่นำเสนอ ประกอบด้วย ข้อมูลขาเข้าจำนวน 1000 คุณลักษณะของแต่ละภาพ ที่ได้จากขั้นตอน data preprocessing ถูกประมวลผลผ่าน convolutional layer 1 มิติ จำนวน 4 layers และมีขั้นตอนการทำ max pooling ทุก convolutional layer 2 layer เพื่อลดความซับซ้อนของ network และเลเยอร์ Drop out คือการนำข้อมูลทิ้งบางส่วน เพื่อป้องกันไม่ให้เกิด over-fitting จากนั้นเข้าสู่การประมวลผลโดย neurons ใน dense layer ช่วยสนับสนุนงานด้านการถดถอย (Regression) หลังจากถูกฝึกฝนอย่างเต็มที่แล้วค่าของค่าถ่วงน้ำหนักจะให้ผลลัพธ์เข้าใกล้กับค่าของคำตอบที่ถูกต้อง (ground truth) โดยผลลัพธ์ (Output) ที่เราต้องการประมาณค่าได้แก่ ค่ามุมหมุนและการเลื่อนของกล้อง จำนวน 7 คำตอบ สามารถแสดงโมเดลสรุป (model summary) ได้ดังรูปที่ 4.8 ประกอบด้วย เลเยอร์และลำดับในแบบจำลอง, ขนาดผลลัพธ์ (output shape) ของแต่ละชั้น, จำนวนพารามิเตอร์ (weights) ในแต่ละชั้น, จำนวนพารามิเตอร์ทั้งหมดในแบบจำลอง ซึ่งขนาดอินพุท คือ 1000 x 1, ตัวกรอง (filter) ของ convolutional layer ที่ 1 ถึง 4 คือ 128, 64, 64, 64 ตามลำดับ

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 1000, 128)	256
activation_1 (Activation)	(None, 1000, 128)	0
conv1d_2 (Conv1D)	(None, 1000, 64)	8256
activation_2 (Activation)	(None, 1000, 64)	0
max_pooling1d_1 (MaxPooling1	(None, 500, 64)	0
dropout_1 (Dropout)	(None, 500, 64)	0
conv1d_3 (Conv1D)	(None, 500, 64)	4160
activation_3 (Activation)	(None, 500, 64)	0
dropout_2 (Dropout)	(None, 500, 64)	0
conv1d_4 (Conv1D)	(None, 500, 64)	4160
activation_4 (Activation)	(None, 500, 64)	0
max_pooling1d_2 (MaxPooling1	(None, 250, 64)	0
flatten_1 (Flatten)	(None, 16000)	0
dense_1 (Dense)	(None, 128)	2048128
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 4)	260

Total params: 2,073,476
Trainable params: 2,073,476
Non-trainable params: 0

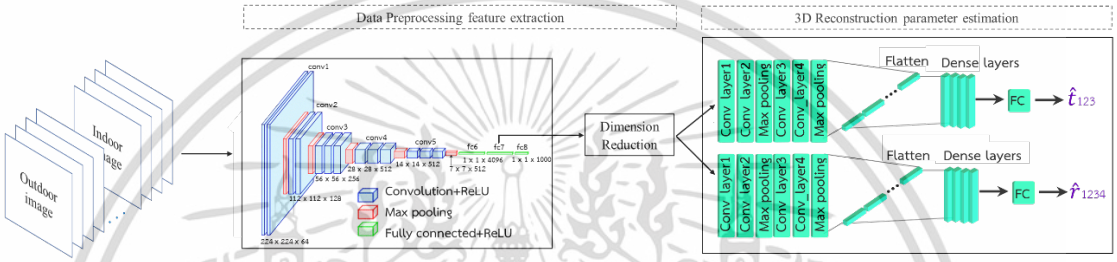
รูปที่ 4.8 โมเดลสรุปการฝึกฝนโมเดลการเลื่อน

4.4 โมเดลการฝึกฝนร่วมหลายโดเมน (Combined domain single model)

การสร้างแบบจำลองที่สามารถสร้างพิกัดสามมิติได้ทั้งภายในและภายนอกเป็นสิ่งที่พึงปรารถนาเนื่องจากสามารถช่วยให้สามารถสร้างโมเดลสามมิติเข้าสู่อาคารในแบบจำลองเมืองเสมือนจริงได้อย่างราบรื่นแทนที่จะสำรวจเพียงเฉพาะภายนอก ในทำนองเดียวกันโมเดลรวมกันจะช่วยให้สามารถเปลี่ยนระหว่างโลกในร่มและกลางแจ้งได้อัตโนมัติ ไม่จำเป็นต้องใช้มนุษย์ควบคุมการสร้างโมเดลว่าต้องสร้างโมเดลแบบในร่มหรือกลางแจ้งให้แก่ระบบ อย่างไรก็ตามวิธีการสร้างแบบจำลองที่สามารถทำนายมุมหมุนและการเลื่อนของฉากในร่มและกลางแจ้งด้วยโมเดลเดียวเป็นเรื่องยากและ

เป็นปัญหาที่ท้าทายเนื่องจากโครงสร้างเชิงพื้นที่ที่ซับซ้อน เช่น รูปแบบการถ่าย, สภาพวะแสงที่อาจมีการเปลี่ยนแปลงอย่างมาก และพื้นผิวที่แตกต่างกัน เป็นต้น ด้วยเหตุนี้เราจึงนำเสนอแบบจำลอง CNN ของการประกอบกลับภาพสามมิติจากภาพสองมิติแบบโมเดลการฝึกฝนร่วมหลายโดเมน จากโมเดลการถ่ายโอนการเรียนรู้ก่อนหน้าเพื่อให้ได้ประสิทธิภาพที่ดีขึ้นบนชุดข้อมูลมาตรฐานในโลกแห่งความเป็นจริง(on benchmark real-world datasets)

3D reconstruction parameter estimation นี้ใช้ CNN ในการประมาณค่ามุมหมุนและการเลื่อน แบบการฝึกฝนโดเมน Indoor และ Outdoor ร่วมกัน ดังรูปที่ 4.9



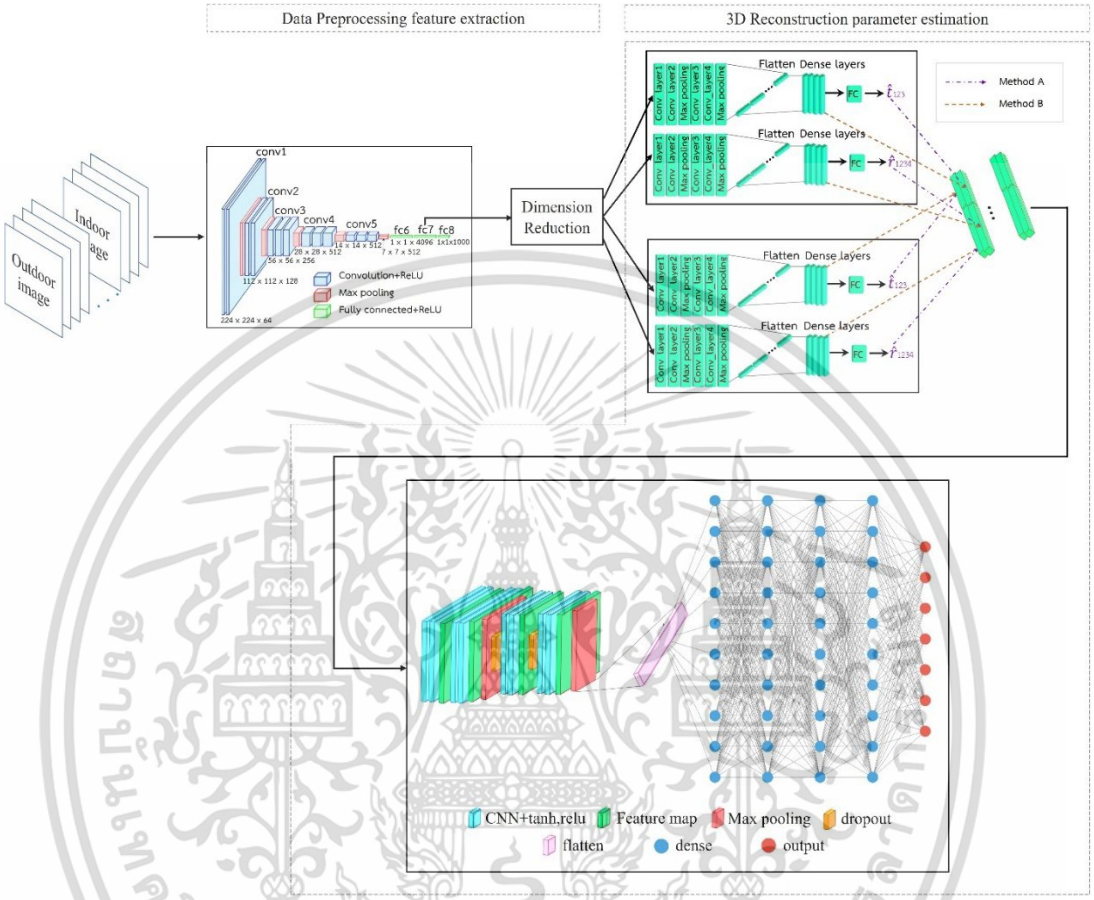
รูปที่ 4.9 โมเดลการฝึกฝนร่วมหลายโดเมน

จากรูปที่ 4.7 เป็นโมเดลเดี่ยวที่เรานำ Feature ที่ได้จากขั้นตอน Data Preprocessing ของทั้งข้อมูลในร่มและกลางแจ้งมาเรียนรู้ร่วมกันด้วยโมเดลเดี่ยว CNN 1 มิติจำนวน 2 ชุด ได้แก่ ชุดประมาณค่าพารามิเตอร์การหมุน ($\hat{\mathbf{r}}_{1234}$) และ ชุดประมาณค่าพารามิเตอร์การเลื่อน ($\hat{\mathbf{t}}_{123}$) ซึ่ง CNN แต่ละชุดประกอบด้วย Convolutional Layer จำนวน 4-layer เพื่อทำการดึงคุณลักษณะเชิงพื้นที่ โดยสลับทำ Max pooling หลัง CNN layer ที่ 2 และ 4 จากนั้นทำการ Flatten เพื่อเข้าสู่ส่วน Dense Layer จำนวน 4-layer เพื่อทำการประมาณค่าพารามิเตอร์ที่ต้องการ

4.5 โมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN

จาก model ที่ผ่านมาข้างต้น แม้ว่าเทคนิคข้างต้นจะให้ผลลัพธ์ในการประมาณค่าผลลัพธ์ที่แม่นยำ แต่การเป็นโมเดลเดียว (Single Model) นั้นทำให้มีการกำหนดกลุ่มของข้อมูลที่ใช้ในการเรียนรู้ รวมทั้งมีการกำหนดค่าพารามิเตอร์ที่ตายตัว บางครั้งเกิดปัญหาความโน้มเอียง (Bias) หนทางหนึ่งที่จะสามารถลดค่าความโน้มเอียงได้คือการใช้วิธี Ensemble คือเทคนิคการรวมของโมเดลการ

เรียนรู้ที่หลากหลาย ที่มีความแตกต่างและมีอิสระต่อกัน เข้าด้วยกัน เพื่อ low bias and high variance นอกจากนี้ยังเพิ่มประสิทธิภาพของโมเดล โดยขั้นตอนของวิธีนี้แสดงดังรูปที่ 4.10



รูปที่ 4.10 โมเดลเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN

ส่วนของโมเดลการเรียนรู้แบบ Ensemble CNN จะแบ่งเป็น เอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) และเอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมน (domain relationship agents) เพื่อให้เกิดความยืดหยุ่นต่อการเรียนรู้หลากหลายโดเมนข้อมูลภาพในการวิเคราะห์และประมาณพารามิเตอร์ที่ต้องการ

เอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) จะใช้โครงสร้างเดียวกับโครงสร้างของโมเดลเดี่ยว แต่จะทำการฝึกฝนโมเดลด้วยข้อมูลเฉพาะโดเมนนั้นๆ ในงานวิจัยนี้จะทดสอบต้นแบบกับเอเจนท์การเรียนรู้เฉพาะโดเมนภาพในร่ม และเอเจนท์การเรียนรู้เฉพาะโดเมนภาพกลางแจ้ง ผลลัพธ์การวิเคราะห์และประมาณค่าพารามิเตอร์ที่ได้จากเอเจนท์การเรียนรู้เฉพาะ

โดเมนทั้งหมด จะถูกส่งไปประมวลผลยังเอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมน เพื่อให้ได้คำตอบที่ดีที่สุดและสามารถทำนายมุมหมุนและการเลื่อนของประเภทการถ่ายภาพที่แตกต่างกันของทั้งโดเมนในร่มและกลางแจ้ง โดยในงานวิจัยนี้จะทำการทดสอบรูปแบบการดึงข้อมูล 2 รูปแบบ ได้แก่ แบบ A เป็นการดึงคุณลักษณะได้จากเลเยอร์ output ของเอเจนท์เรียนรู้คุณลักษณะเฉพาะแต่ละโดเมน และมาจัดเรียงเป็น Feature ที่มีมิติข้อมูล 7x2 ซึ่งประกอบจาก 7 ค่าพารามิเตอร์การหมุน ($\hat{r}_{1234}$) และ ค่าพารามิเตอร์การเลื่อนตำแหน่ง ($\hat{t}_{123}$) จากเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพในร่มและเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพกลางแจ้ง ดังแสดงเป็นเส้นประสีม่วงในรูปที่ 4.10 แบบ B เป็นการดึงคุณลักษณะจากเลเยอร์ก่อน output layer เพื่อประกอบกันเป็น Feature ที่มีมิติขนาด 64x2 โดยมีคุณลักษณะของมุมหมุนและการเลื่อนอย่างละ 64 คุณลักษณะ จากเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพในร่มและเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพกลางแจ้ง ดังแสดงเป็นเส้นประสีแดงในรูปที่ 4.10

เอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมนประกอบด้วย CNN 2 มิติ ที่มีชุด Convolutional layer 2 layers และ Max Polling จำนวน 2 ชุด เพื่อวิเคราะห์ความสัมพันธ์ระหว่างโดเมน ตามด้วย Neural Network layer (Dense layer) เพื่อสรุปผลการประมาณค่าพารามิเตอร์ ผลลัพธ์จาก Dense Layer ที่ส่งออกจะให้การทำนายเป็นแบบ regression คำตอบจึงเป็น continuous values เมื่อได้ผลลัพธ์ (Output) ที่เราต้องการประมาณค่า ได้แก่ ค่ามุมหมุนจำนวน 4 คำตอบ และค่าการเลื่อนของกล้อง จำนวน 3 คำตอบ รวม 7 คำตอบ จากนั้นนำไปทำการประเมินผลหาค่าความผิดพลาดของการทำนายค่ามุมหมุนและการเลื่อนที่ได้ออกมาด้วยวิธีค่าความผิดพลาดเฉลี่ยกำลังสอง (Root mean square error, RMSE)

ในงานวิจัยนี้เรานำเสนอการประมาณค่าพารามิเตอร์มุมหมุนและการเลื่อนของภาพถ่ายสองมิติ เพื่อนำมาใช้ในการประกอบกลับภาพ 3 มิติ โดยอาศัยการทำการถ่ายโอนการเรียนรู้ (Transfer learning) จากโมเดลการเรียนรู้เชิงลึกแบบ CNN อย่างไรก็ดีด้วยข้อจำกัดของการฝึกฝน CNN โมเดลเดี่ยวซึ่งเรียนรู้คุณลักษณะเฉพาะเชิงพื้นที่ การจะฝึกฝนโมเดล CNN ให้สามารถเรียนรู้ข้อมูลหลากหลายโดเมน และสามารถวิเคราะห์ผลลัพธ์ให้มีประสิทธิภาพ โดยไม่ขึ้นกับความแตกต่างของคุณลักษณะเฉพาะแต่ละโดเมนนั้นทำได้ยาก เนื่องจากการเรียนรู้คุณลักษณะของภาพ มุมมองการถ่ายภาพ และพารามิเตอร์ที่เกี่ยวข้องกับการรับภาพ (Acquisition) ที่แตกต่างกันอย่างมาก มีผลให้คุณลักษณะเชิงพื้นที่ที่วิเคราะห์ได้จากโมเดลแตกต่างกันไป ทำให้การฝึกฝนต้องการข้อมูลปริมาณมาก และเมื่อมีข้อมูล domain ใหม่เข้ามา จะต้อง retrain ข้อมูลทั้งหมดอีกครั้ง ส่งผลให้ใช้เวลาในการพัฒนาสูง เราจึงได้นำเสนอโมเดลโครงข่ายเอเจนท์การเรียนรู้ ที่เรียกว่า Ensemble CNN โดยจะ

แบ่งเป็นเอเจนท์ที่เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) และเอเจนท์ที่เรียนรู้ความสัมพันธ์ระหว่างโดเมน (domain relationship agents) เพื่อให้เกิดความยืดหยุ่นต่อการเรียนรู้หลากหลายโดเมนข้อมูลภาพ ในการวิเคราะห์และประมาณพารามิเตอร์ที่ต้องการ โดยมีการศึกษาและเปรียบเทียบประสิทธิภาพของ โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR model, โมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN, โมเดลเดี่ยวแบบฝึกฝนร่วมหลายโดเมนด้วย CNN และแบบโครงข่ายเอเจนท์ด้วย CNN model โดยโมเดลที่กล่าวมา จะนำไปใช้เปรียบเทียบผลการประมาณค่าพารามิเตอร์มุมหมุนและการเลื่อนของกล้อง ในรูปแบบของการวิเคราะห์เชิงโมเดลเดี่ยว (based-line end-to-end single model) และ รูปแบบการวิเคราะห์เชิงโครงข่ายเอเจนท์ (Ensemble model) ที่นำเสนอ สำหรับโดเมนภาพในร่มและกลางแจ้ง ที่มีรูปแบบการรับภาพและคุณลักษณะของภาพที่แตกต่างกันอย่างมาก

บทที่ 5

ผลการทดลองและการวิเคราะห์

ในบทนี้จะกล่าวถึงการทดลอง ผลการทดลอง และการวิเคราะห์ผลที่ได้จากการทดสอบการประมาณค่ามุมหมุนและการเลื่อนของรูปภาพสองมิติ โดยนำเสนอโครงสร้างโมเดล Ensemble CNN ที่ประยุกต์คุณลักษณะที่ทำ transfer learning ออกมาจากการเรียนรู้เชิงลึก เพื่อประมาณค่ามุมหมุนและการเลื่อนในสามมิติจากการประมาณค่าหลายกล้องพร้อมกัน โดยเริ่มจากการนำข้อมูลภาพถูกแบ่งออกเป็นชุดข้อมูลฝึกฝนและชุดข้อมูลทดสอบมาสกัดคุณลักษณะด้วยโมเดลการเรียนรู้เชิงลึก VGG19 ซึ่งคุณลักษณะจะถูกนำออกมาจาก fully connected ที่ 7 ด้วยจำนวนคุณลักษณะ 4096 ของแต่ละภาพ เพื่อให้ระยะเวลาในการประมาณค่าถดถอยลดลง จึงจำเป็นต้องนำคุณลักษณะมาลดมิติข้อมูลให้เหลือเพียงมิติของคุณลักษณะที่สำคัญด้วยเทคนิค Latent Semantic Analysis (LSA) จากนั้นนำมาเข้าสู่ขั้นตอนการประมาณค่ามุมหมุนและการเลื่อนของการประกอบกลับสามมิติ และสุดท้ายประมาณค่าพารามิเตอร์ความสัมพันธ์ระหว่างกล้อง โดยชุดข้อมูลที่งานวิจัยฉบับนี้นำมาใช้เป็นข้อมูลรูปภาพสองมิติ ทั้งหมดถูกทำการเปรียบเทียบประสิทธิภาพผลลัพธ์กับชุดทดสอบจาก 3 แหล่งที่มาด้วยกัน คำตอบที่ทำนายได้ถูกนำมาหาค่าความถูกต้องด้วยค่าความผิดพลาดเฉลี่ยกำลังสอง (Root mean square error, RMSE) สำหรับชุดทดสอบในร่มชุดที่หนึ่งและชุดทดสอบกลางแจ้งชุดที่สอง ส่วนชุดทดสอบชุดที่สามจะถูกวัดประสิทธิภาพด้วยค่าความผิดพลาดมัธยฐาน

สำหรับหัวข้อนี้แบ่งออกเป็น 5 หัวข้อหลักๆ ได้แก่ รายละเอียดฐานข้อมูลภาพ, การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR, การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN, การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลการฝึกฝนร่วมหลายโดเมน (Combined domain single model) และส่วนสุดท้ายคือ การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN ดังรายละเอียดในแต่ละหัวข้อได้ดังนี้

5.1 รายละเอียดฐานข้อมูลภาพ

แหล่งที่ 1 คือ เป็นภาพถ่ายแบบ Indoor รูปทรงโดมจาก CMU Panoptic Studio [52] ภาพถูกถ่ายด้วยมุมหมุนและการเลื่อนในแนวแกน x แกน y และแกน z รอบวัตถุ มีชุดข้อมูลภาพ

ทั้งหมด 209,934 ภาพ แบ่งเป็นชุดภาพสำหรับฝึกฝนจำนวน 146,957 ภาพ และชุดภาพสำหรับทดสอบจำนวน 62,977 ภาพ แต่ละภาพจะมีคำตอบประกอบด้วย ค่าของมุมหมุน (Rotation; R) และค่าการเลื่อน (Translation; T) ซึ่งเป็นพารามิเตอร์ภายนอกของกล้อง ชุดภาพทั้งหมดมีคำตอบ 480 คำตอบ ชุดการทดลองนี้จะประกอบด้วยการทำท่าอิริยาบถต่างๆของคนหรือกลุ่มคน เช่น การเดินแบบมีทิศทางและไร้ทิศทาง การสวิงไม้เบสบอล เป็นต้น ดังรูปที่ 5.1

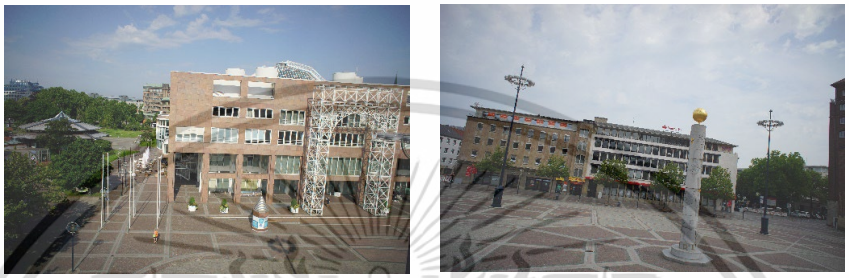
รูปที่ 5.1. ชุดภาพในร่มของ CMU Panoptic Studio.[52]

(ก) ตัวอย่างชุดการสวิงไม้เบสบอล

(ข) ตัวอย่างชุดการเดินแบบมีทิศทางและไร้ทิศทาง

แหล่งที่ 2 เป็นภาพถ่ายทางอากาศกลางแจ้ง [51] (Map) ที่ถูกถ่ายด้วยกล้องที่ติดกับโดรนซึ่งบินสูงเหนือสถานที่ต่างๆ ประกอบด้วย 4 สถานที่คือ square nadir, stadthaus, rathaus, square obelisk oblique การถ่ายในลักษณะนี้ การเลื่อนขนาดใหญ่จะเกิดขึ้นในแนวแกน x, y แต่การเลื่อนในแกน z ของแต่ละภาพมีความแตกต่างกันเล็กน้อย ประกอบด้วยภาพการฝึกฝน 1,500 ภาพและภาพการทดสอบ 578 ภาพ ดังรูปที่ 5.2

(ก) (ข)

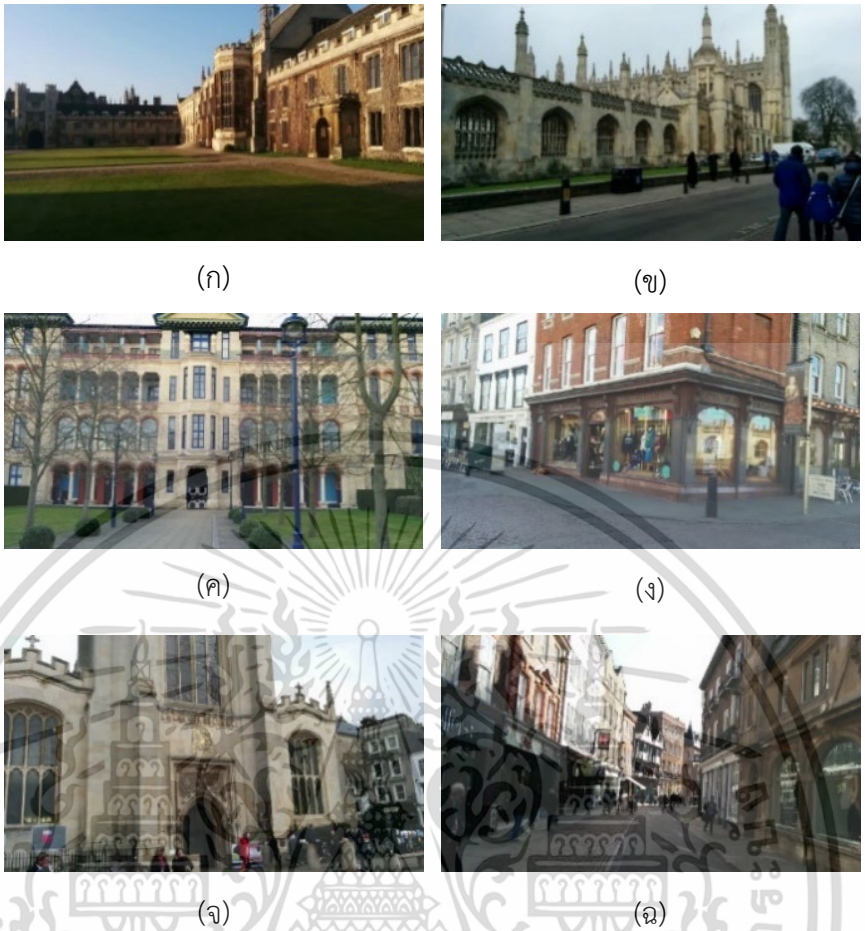


(ค) (ง)

รูปที่ 5.2 ชุดภาพกลางแจ้งถ่ายทางอากาศ (Map) [51].

(ก) square nadir. (ข) stadthaus. (ค) rathaus. (ง) square obelisk oblique.

แหล่งที่มาสุดท้าย (Outdoor: Cambridge dataset) [53] ประกอบด้วยชุดข้อมูลย่อย 6 ชุด เป็นชุดข้อมูลที่เป็นที่นิยมสำหรับนำมาใช้วัดประสิทธิภาพอัลกอลิทึมสำหรับการประมาณค่ามุมหมุนและการเลื่อน ประกอบด้วย ชุดภาพ GreatCourt, Kings College, Old Hospital, Shop Facade, St Marys Church และ Street โดย GreatCourt มีภาพการฝึก 1532 ภาพและภาพทดสอบ 760 ภาพ Kings College มีขอบเขตเชิงพื้นที่ที่ใหญ่ที่สุดที่ 5000 ตร.ม.(ตารางเมตร) ในบรรดาชุดข้อมูลทั้งหมด ประกอบด้วยภาพการฝึก 1220 ภาพและภาพทดสอบ 346 ภาพ Old Hospital มีพื้นที่ 2,000 ตร.ม. มีการฝึกฝน 895 ภาพและการทดสอบ 182 ภาพ Shop Facade มีขอบเขตเชิงพื้นที่ 875 ตร.ม. มีรูปภาพการฝึกฝน 230 รูปและรูปภาพการทดสอบ 103 ภาพ St Marys Church มีขอบเขตเชิงพื้นที่ 4800 ตร.ม. ประกอบด้วยรูปภาพการฝึก 1487 ภาพและภาพการทดสอบ 530 ภาพ และ Street ประกอบด้วยภาพการฝึก 3015 ภาพ และภาพการทดสอบ 2923 ภาพ สำหรับชุดข้อมูล Cambridge กลางแจ้ง ดังรูปที่ 5.3



(ก) (ข)

(ค) (ง)

(จ) (ฉ)

รูปที่ 5.3 ชุดภาพถ่ายกลางแจ้ง Cambridge [53].

(ก) GreatCourt. (ข) Kings College. (ค) Old Hospital.

(ง) Shop Facade. (จ) St Marys Church. (ฉ) Street.

การทดลองจะแบ่งเป็นการหาค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนโดยเปรียบเทียบกับอัลกอลิธึมของงานวิจัย [23] ในชุดข้อมูลที่ 1 (Dome) และ ชุดที่ 2 (Map) เริ่มจากการนำชุดข้อมูลที่ 1 แบ่งออกเป็นชุดฝึกฝนและทดสอบจำนวน 2 ใน 3 และ 1 ใน 3 ของจำนวนภาพในชุดข้อมูลตามลำดับ ในการวัดประสิทธิภาพผลลัพธ์จะนำเสนอ การวิเคราะห์ผลลัพธ์จากโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR ในหัวข้อ 5.2 วิเคราะห์ผลลัพธ์จากโมเดลฝึกฝนแยกโดเมนด้วย CNN ในหัวข้อ 5.3 วิเคราะห์ผลลัพธ์จากโมเดลการฝึกฝนร่วมหลายโดเมน ในหัวข้อ 5.4 และวิเคราะห์ผลลัพธ์จากโมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN

5.2 การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย SVR

การประเมินค่าความถูกต้องการทำนายมุมหมุนและการเลื่อนของชุดข้อมูลชุดที่ 1 (Dome ในร่ม) ของเทคนิคการประมาณค่าการถดถอย (Support Vector Regression: SVR) ซึ่งเป็นเทคนิคที่อาศัยพื้นฐานหลักการของ Support Vector Machine (SVM) มาทำการประมาณค่า โดยการทดลองจะทำการปรับ 2 พารามิเตอร์ คือพารามิเตอร์ C และ gamma ซึ่งพารามิเตอร์ C ที่นำมาทดสอบมี 7 ค่า คือ 0.001, 0.01, 0.1, 1, 10, 100 และ 1000 และพารามิเตอร์ gamma มี 7 ค่า คือ 0.0001, 0.001, 0.01, 0.1, 1.0, 10 และ 100 ตามลำดับ จากนั้นนำค่าที่ทำนายได้มาทำการหาค่าความคลาดเคลื่อนจากคำตอบ (ground truth) ตารางที่ 5.1 แสดงผลการทดลองค่าความผิดพลาดของการทำนายชุดทดสอบ จากการเรียนรู้ด้วยชุดฝึกฝนแต่ละชุด พบว่าค่าความผิดพลาดเฉลี่ยของการทำนายจากชุดทดสอบทั้งหมด พารามิเตอร์ C และ gamma ที่ทำให้เกิดค่าความผิดพลาดของการทำนายมุมหมุนที่น้อยที่สุดคือ C=10, gamma=1 คือ 0.242 องศา และค่าความผิดพลาดของการทำนายการเลื่อนที่น้อยที่สุดคือ C=1000 และ gamma=1 คือ 1.364 เมตร

ตารางที่ 5.1 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน(R) และการเลื่อน (T)

C	gamma	ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน	ค่าความผิดพลาดเฉลี่ยกำลังสองของการเลื่อน
0.001	0.0001	0.345	1.781
0.001	0.001	0.345	1.781
0.001	0.01	0.345	1.781
0.001	0.1	0.342	1.781
0.001	1	0.330	1.781
0.001	10	0.333	1.781
0.001	100	0.345	1.781
0.01	0.0001	0.345	1.781
0.01	0.001	0.345	1.781
0.01	0.01	0.342	1.781
0.01	0.1	0.325	1.781
0.01	1	0.289	1.779
0.01	10	0.295	1.779
0.01	100	0.342	1.781

ตารางที่ 5.1 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน(R) และการเลื่อน (T)

C	gamma	ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน	ค่าความผิดพลาดเฉลี่ยกำลังสองของการเลื่อน
0.1	0.0001	0.345	1.781
0.1	0.001	0.342	1.781
0.1	0.01	0.324	1.781
0.1	0.1	0.288	1.778
0.1	1	0.266	1.761
0.1	10	0.267	1.765
0.1	100	0.332	1.781
1	0.0001	0.342	1.781
1	0.001	0.324	1.781
1	0.01	0.288	1.777
1	0.1	0.272	1.753
1	1	0.249	1.663
1	10	0.262	1.667
1	100	0.326	1.777
10	0.0001	0.324	1.781
10	0.001	0.288	1.777
10	0.01	0.275	1.752
10	0.1	0.255	1.653
10	1	0.242	1.553
10	10	0.262	1.474
10	100	0.326	1.747
100	0.0001	0.288	1.777
100	0.001	0.276	1.752
100	0.01	0.259	1.653
100	0.1	0.250	1.600
100	1	0.243	1.469
100	10	0.262	1.379
100	100	0.326	1.666
1000	0.0001	0.276	1.752
1000	0.001	0.260	1.653
1000	0.01	0.256	1.610

ตารางที่ 5.1 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน(R) และการเลื่อน (T)

C	gamma	ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน	ค่าความผิดพลาดเฉลี่ยกำลังสองของการเลื่อน
1000	0.1	0.248	1.569
1000	1	0.243	1.355
1000	10	0.262	1.364
1000	100	0.326	1.648

จากโมเดล SVR ที่ดีใช้พารามิเตอร์ที่สุดที่สรุปจากหัวข้อก่อน จะถูกนำมาเปรียบเทียบประสิทธิภาพในรูปแบบการหาค่าความผิดพลาดเฉลี่ยกำลังสอง (Average RMSE) กับ เทคนิค SFM [23] ด้วยข้อมูลชุดที่ 1 ดังแสดงในตารางที่ 5.2

ตารางที่ 5.2 ตารางเปรียบเทียบประสิทธิภาพการทำนายมุมหมุนและการเลื่อนของงานวิจัยที่นำเสนอกับงานอ้างอิง [23]

โมเดล	ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุน (องศา)	ค่าความผิดพลาดเฉลี่ยกำลังสองของการเลื่อน (เมตร)
SfM [23]	2.930	4.470
SVR	0.243	1.355
ผลต่างค่าความผิดพลาด (SfM, SVR)	2.687	3.115

จากผลการเปรียบเทียบ การหาค่าความผิดพลาดเฉลี่ยกำลังสองจะเห็นว่า วิธี SVR ที่นำเสนอให้ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนเป็น 0.2419 องศา และการเลื่อนเป็น 1.35 เมตร ซึ่งมีความผิดพลาดน้อยกว่างานวิจัย [23] อยู่ 2.69 องศา และ 3.11 เมตร ตามลำดับ เนื่องจากภาพทั้งหมดจะถูกนำมาพิจารณาพร้อมกันอย่างเท่าเทียมกันโดยที่ไม่ต้องใช้พารามิเตอร์ภายในกล้องมาร่วมประมาณค่า ผลลัพธ์ที่ได้จะไม่ขึ้นอยู่กับลำดับภาพถ่ายที่ได้รับการพิจารณา จึงทำให้ค่าความถูกต้องสูงขึ้นจากวิธี [23] ที่ประสิทธิภาพขึ้นกับลำดับภาพที่นำเข้าประมวลผล

จากที่กล่าวมาข้างต้น แม้วิธี SVR ที่นำเสนอ จะให้ค่าความผิดพลาดน้อยลง แต่ระยะเวลาในการคำนวณคำตอบจะใช้ระยะเวลานานเช่นกัน ต่อมางานวิจัยนี้ จึงนำเสนอโมเดลการประมาณค่าผลลัพธ์ โดยเราได้ทำการสร้างโมเดลเดี่ยวแบบแยกฝึกฝนโดเมนด้วยโมเดล CNN 1 มิติ เพื่อทำการสร้างคุณลักษณะที่เหมาะสมและทำการประมาณค่าพารามิเตอร์มุมหมุนและการเลื่อน ดังแสดงในหัวข้อที่ 5.3

5.3 การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN

การทำนายพารามิเตอร์มุมหมุนและการเลื่อนด้วยโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN ในหัวข้อนี้จะอธิบายการฝึกฝนโมเดลทั้งหมด 4 โมเดล ได้แก่ โมเดลแรกเป็นโมเดลสำหรับการทำนายมุมหมุนของภาพในร่ม โมเดลที่สองเป็นโมเดลสำหรับการทำนายการเลื่อนของภาพในร่ม โมเดลที่สามเป็นโมเดลสำหรับการทำนายมุมหมุนของภาพกลางแจ้ง สุดท้ายโมเดลที่สี่เป็นโมเดลสำหรับการทำนายการเลื่อนของภาพกลางแจ้ง โดยจะทำการทดสอบกับข้อมูลภาพทั้งที่ใช้ในการฝึกฝนและไม่ได้ใช้ในการฝึกฝนโมเดล และทำการวัดประสิทธิภาพด้วยค่าความผิดพลาดเฉลี่ยกำลังสอง นอกจากเปรียบเทียบประสิทธิภาพระหว่างโมเดลทั้ง 4 แล้วยังทำการเปรียบเทียบกับงานวิจัยอ้างอิงที่มีการวัดประสิทธิภาพผลลัพธ์ด้วยค่ามัธยฐานกับชุดภาพชุดที่สาม (Cambridge dataset) โดยทำการฝึกฝนโมเดลทั้งหมด 2 โมเดลสำหรับข้อมูลชุดนี้ ได้แก่ โมเดลที่เป็นการนำชุดภาพกลางแจ้งชุดที่ 3 มาฝึกฝนเพื่อเป็นโมเดลสำหรับการทำนายมุมหมุนและโมเดลสำหรับการทำนายการเลื่อนของชุดภาพกลางแจ้ง

โครงสร้าง CNN ที่ใช้เป็น CNN 1 มิติ พารามิเตอร์โครงสร้างของโมเดลและ optimizer ที่ได้ทำการทดสอบเพื่อเลือกโครงสร้างโมเดลที่ดีที่สุดได้แก่ การวิเคราะห์จำนวน filter และจำนวน Node ของ CNN layer, รูปแบบของ Activation function ที่ใช้, ชนิดของ optimizer ขนาด Batch_size และจำนวนรอบ epoch ที่เหมาะสม โดยกำหนดให้ใช้ค่าความผิดพลาดเฉลี่ยกำลังสองเป็นค่าประเมินผลลัพธ์ โดยกำหนดการปรับค่าจำนวน filter [64,128] จำนวนโหนดในเลเยอร์ [64,128] Output Activation ที่ใช้ทดสอบ [tanh, linear] โดยทดสอบ optimizer แบบ Adam และ Adadelta batch_size = [32, 50, 100] และ จำนวน epoch = 400, 500 และ 600 ตามลำดับ

พารามิเตอร์ที่ให้ค่าความผิดพลาดของมุมหมุนและการเลื่อนน้อยที่สุดจะพิจารณาเป็นพารามิเตอร์สำหรับชุดทดสอบที่เป็นชนิดในร่มจำนวน 1 ชุด (Dome) และพารามิเตอร์ที่ดีที่สุดสำหรับชุดกลางแจ้งจำนวน 2 ชุด (Map and Cambridge dataset) พารามิเตอร์ที่ดีที่สุดคือจำนวน epoch =500, batch size = 32 ด้วย optimizer แบบ Adam ซึ่ง จำนวน filter ของชุดภาพในร่มที่เรานำเสนอสำหรับแต่ละ CNN layer คือ {128,128,128,64} สำหรับมุมหมุน, และ {64,64,64,128} สำหรับการเลื่อน และสำหรับชุดกลางแจ้งคือ {128,128,64,64} สำหรับมุมหมุน, {128,128,128,64} สำหรับการเลื่อน โดยทำการทดลองเปรียบเทียบค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนกับอัลกอลิธึมของโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN ด้วยชุดข้อมูลที่ 1 (Dome) และ

2 (Map) ดังแสดงในตารางที่ 5.3 และเปรียบเทียบค่าความผิดพลาดมัธยฐานของการหมุนและการเลื่อนกับงานวิจัย [61] ด้วยข้อมูลชุดที่ 3 ดังตารางที่ 5.4 ตามลำดับ

ตารางที่ 5.3 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของการฝึกฝนและทดสอบ วิธีโมเดลเดี่ยวแบบฝึกฝนแยกโดเมนด้วย CNN (โมเดล 4.4) ที่นำเสนอเปรียบเทียบกับ SVR

		SVR (โมเดล 4.2)		CNN (โมเดล 4.4)	
ชุดฝึกฝน	ชุดทดสอบ	มุมหมุน(องศา)	การเลื่อน (เมตร)	มุมหมุน (องศา)	การเลื่อน (เมตร)
Dome	Dome	0.45	2.45	0.07	1.15
	Map	0.35	1.29	0.27	1.25
Map	Map	0.14	0.0035	0.03	0.0023
	Dome	0.74	4.49	0.37	2.59
Average RMSE		0.44	2.25	0.19	1.25

จากตารางที่ 5.3 แสดงผลการทดสอบโมเดลที่ได้รับการฝึกฝนด้วยข้อมูลชุดรูปภาพในร่มข้อมูลชุดที่ 1 (Dome) และข้อมูลชุดรูปภาพกลางแจ้งชุดที่ 2 (Map) และ ทดสอบแต่ละโมเดลด้วยชุดข้อมูลที่ถูกฝึกและไม่ได้ถูกฝึก การวัดประสิทธิภาพของอัลกอริทึมในการทดลองนี้ เพื่อต้องการทราบว่าการทดสอบด้วยชุดข้อมูลที่มีมุมมองหรือรูปแบบการถ่ายเดียวกับชุดฝึกฝนและชุดข้อมูลที่มีมุมมองหรือรูปแบบการถ่ายแตกต่างจากชุดฝึกฝนนั้น โมเดลจะให้ค่าความผิดพลาดเฉลี่ยกำลังสองแตกต่างกันอย่างไร จากสิ่งแวดล้อมของชุดข้อมูลที่ 1 เป็นชุดข้อมูลแบบในร่ม ที่มีมุมมองการถ่ายภาพที่หลากหลายทิศทาง ในขณะที่ชุดข้อมูลที่ 2 เป็นแบบกลางแจ้งที่มีมุมมองการถ่ายภาพ ที่จำกัดในทิศทางแกน z มีการปรับเปลี่ยนมุมองเฉพาะทิศทางแกน x, y จะเห็นว่าผลรวมเฉลี่ยของค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนของวิธี CNN สำหรับชุดฝึกฝนของข้อมูลชุดที่ 1 เมื่อนำมาทดสอบด้วยชุดทดสอบของชุดที่ 1 และชุดที่ 2 ให้ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนเป็น 0.07, 0.27 องศา และ 1.15, 1.25 เมตร ตามลำดับ นอกจากนี้สำหรับชุดฝึกฝนของข้อมูลชุด 2 เมื่อนำมาทดสอบด้วยชุดทดสอบของชุดที่ 1 และชุดที่ 2 ให้ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนเป็น 0.03, 0.37 องศาและ 0.0023, 2.59 เมตร ตามลำดับ ซึ่งมีผลรวมเฉลี่ยของค่าความผิดพลาดเฉลี่ยกำลังสองของค่ามุมหมุนและการเลื่อนน้อยกว่าวิธีของ SVR อยู่ 0.26 องศา และ 1 เมตร ตามลำดับ จากการทดลอง จะเห็นว่า การฝึกฝนด้วยข้อมูลรูปภาพชุดที่ 1 ค่าความผิดพลาดเฉลี่ยกำลังสองระหว่างชุดทดสอบ 1 และ 2 นั้นค่าของมุมหมุนและการเลื่อนต่างกัน 0.2 องศา และ 0.1 เมตร ตามลำดับ ซึ่งมีความต่างกันเพียงเล็กน้อย เมื่อเทียบกับการฝึกฝน

ด้วยชุด 2 ซึ่งถูกทดสอบด้วยชุดทดสอบที่ 1 และ 2 นั้นให้ค่าความแตกต่างของค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนเป็น 0.34 องศา และ 2.5877 เมตร ตามลำดับ เนื่องจาก จำนวนชุดฝึกฝนของชุดที่ 2 มีจำนวนน้อยกว่าชุดฝึกฝนของชุดที่ 1 ถึง 100 เท่า และรูปแบบการถ่ายของชุดที่ 2 มีรูปแบบการถ่ายหลากหลายน้อยกว่ารูปแบบการถ่ายของชุดที่ 1 จึงส่งผลให้การเรียนรู้ข้ามวิธีการถ่าย อาจยังไม่เพียงพอ

ตารางที่ 5.4 แสดงค่าความผิดพลาดมัธยฐานของการเลื่อน (เมตร) และการหมุน (องศา) สำหรับการประมาณท่าทางของกล้องบนข้อมูลชุดที่ 3 the Cambridge dataset.

Method	GreatCourt	Kings College	Old Hospital	Shop Facade	St Marys Church	Street
PoseNet [53]	NA	1.97 เมตร, 5.40 องศา	2.31 เมตร, 5.38 องศา	1.46 เมตร, 8.08 องศา	2.65 เมตร, 8.48 องศา	3.67 เมตร, 6.50 องศา
Dense PoseNet [53]	NA	1.66 เมตร, 4.86 องศา	2.57 เมตร, 5.14 องศา	1.41 เมตร, 7.18 องศา	2.45 เมตร, 7.96 องศา	2.96 เมตร, 6.00 องศา
Bayesian PoseNet [54]	NA	1.74 เมตร, 4.06 องศา	2.57 เมตร, 5.14 องศา	1.25 เมตร, 7.54 องศา	2.11 เมตร, 8.38 องศา	2.14 เมตร, 4.96 องศา
LST เมตร-Pose [55]	NA	0.99 เมตร, 3.65 องศา	1.51 เมตร, 4.29 องศา	1.18 เมตร, 7.44 องศา	1.52 เมตร, 6.68 องศา	NA
SVS-Pose [56]	NA	1.06 เมตร, 2.81 องศา	1.50 เมตร, 4.03 องศา	0.63 เมตร, 5.73 องศา	2.11 เมตร, 8.11 องศา	NA
PoseNet + Reprojection error pose loss [57]	7.00 เมตร, 3.7 องศา	0.99 เมตร, 1.1 องศา	2.17 เมตร, 2.9 องศา	1.05 เมตร, 4.0 องศา	1.49 เมตร, 3.40 องศา	20.7 เมตร, 25.7 องศา
VLocNet [58]	NA	0.83 เมตร, 1.42 องศา	1.07 เมตร, 2.41 องศา	0.59 เมตร, 3.53 องศา	0.63 เมตร, 3.91 องศา	NA
DSAC [59]	2.80 เมตร, 1.5 องศา	0.30 เมตร, 0.5 องศา	0.33 เมตร, 0.6 องศา	0.09 เมตร, 0.40 องศา	0.55 เมตร, 16 องศา	NA
LearnLess (DSAC++) [60]	0.4 เมตร, 0.2 องศา	0.18 เมตร, 0.3 องศา	0.20 เมตร, 0.3 องศา	0.06 เมตร, 0.30 องศา	0.13 เมตร, 0.4 องศา	NA
Active Search [61]	NA	0.42 เมตร, 0.6 องศา	0.44 เมตร, 1.0 องศา	0.12 เมตร, 0.40 องศา	0.19 เมตร, 0.5 องศา	0.85 เมตร, 0.8 องศา
CNN (โมเดล 4.3)	0.16 เมตร, 0.18 องศา	0.20 เมตร, 0.21 องศา	0.22 เมตร, 0.48 องศา	0.11 เมตร, 0.20 องศา	0.10 เมตร, 0.39 องศา	0.77 เมตร, 0.76 องศา

จากตารางที่ 5.4 เป็นการนำชุดข้อมูลชุดที่ 3 (the Cambridge dataset) ที่มีชุดภาพย่อยทั้งหมด 6 ชุดย่อย โดยชุดย่อยที่ 1 ถึง 5 เป็นการถ่ายภาพสถานที่ และชุดสุดท้ายเป็นการถ่ายภาพถนน ซึ่งเป็นชุดที่มีความยากของการประกอบภาพมากที่สุดเนื่องจาก สีและพื้นผิว หรือลักษณะของถนนนั้นมีความใกล้เคียงกัน จากงานวิจัยอ้างอิงจะเห็นได้ว่าค่าความผิดพลาดมัธยฐานของชุดย่อยที่ 6

จะให้ค่าความผิดพลาดกว่าชุดย่อยชุดอื่นๆ (ชุดข้อมูลย่อยที่ 1 ถึง 5) โดยลักษณะการถ่ายภาพของข้อมูลชุดที่ 3 นี้ ถูกถ่ายด้วยโดรน ที่ตัวกล้องมีการเคลื่อนที่และมีมุมมองการถ่ายภาพเป็นทั้งในลักษณะมุมเงยและมุมที่เป็นระนาบเดียวกันกับสถานที่ ซึ่งมีลักษณะการถ่ายภาพแตกต่างจากการถ่ายภาพในชุดข้อมูลที่ 1 กล้องถูกติดตั้งอยู่กับที่ในตำแหน่งการถ่ายภาพด้วยมุมก้ม มุมขนานกับวัตถุและมุมเงย และมีความแตกต่างการถ่ายภาพกับชุดข้อมูลที่ 2 ที่ถ่ายด้วยโดรนที่ตัวกล้องเคลื่อนที่และทำการถ่ายภาพด้วยมุมก้มจากโดรนที่บินขนานกับสถานที่)

จากค่าความผิดพลาดมัธยฐานของโมเดลที่ถูกนำเสนอใน [61] จำนวน 10 โมเดล ที่ถูกทดสอบด้วยข้อมูลชุดที่ 3 โมเดลที่ให้ค่าความผิดพลาดมัธยฐานน้อยที่สุดคือโมเดล LearnLess (DSAC++) เมื่อนำค่าความผิดพลาดมัธยฐานของมุมหมุนและการเลื่อนของวิธี CNN ที่นำเสนอ เทียบกับทุกโมเดลที่กล่าวมาพบว่าให้ค่าความผิดพลาดน้อยกว่าวิธีอื่นๆ โดยเฉพาะกับโมเดล Learnless (DSAC++) ซึ่งทดสอบกับชุดข้อมูลย่อย 5 ชุดแรก พบว่าวิธี CNN ที่นำเสนอ ให้ผลรวมของค่าความผิดพลาดมัธยฐานของมุมหมุนน้อยกว่า 0.04 องศา และให้ค่าผลรวมความผิดพลาดมัธยฐานของการเลื่อนน้อยกว่า 0.18 เมตร อย่างไรก็ดีโมเดล Learnless (DSAC++) ไม่ได้ถูกทดสอบด้วยชุดข้อมูลย่อยที่ 6 ดังนั้นสำหรับชุดข้อมูลย่อยที่ 6 ใน [61] โมเดล Active Search(SfM) ที่ให้ค่าความผิดพลาดมัธยฐานของมุมหมุนและการเลื่อนในชุดนั้นน้อยที่สุดคือ 0.8 องศา และ 0.85 เมตร และเมื่อนำมาเปรียบเทียบกับโมเดล CNN ที่นำเสนอ ปรากฏว่าโมเดล CNN ที่นำเสนอให้ค่าความผิดพลาดของมุมหมุนและการเลื่อนน้อยกว่าโมเดล Active Search(SfM) ดังกล่าว 0.04 องศาและ 0.08 เมตร ตามลำดับ

จากตาราง 5.3 และ 5.4 พบว่า โมเดล CNN (โมเดล 4.4) ที่นำเสนอนี้ เหมาะสำหรับการหามุมหมุนและการเลื่อนในรูปแบบการถ่ายภาพของชุดข้อมูลที่ 2 มากที่สุด คือการใช้โดรนถ่ายภาพในลักษณะการถ่ายบนระนาบโดยมีมุมกล้องแบบมุมก้ม เนื่องจากให้ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนของการฝึกฝนและฝึกฝนด้วยข้อมูลภายในชุดที่ 2 น้อยกว่าชุดข้อมูลที่ 1 ด้วยระยะเลื่อนผิดพลาดน้อยกว่า 1.15 เมตร และมุมหมุนผิดพลาดน้อยกว่า 0.04 องศา ตามลำดับ และน้อยกว่าชุดข้อมูลที่ 3 ด้วยระยะเลื่อนผิดพลาดน้อยกว่า 0.26 เมตร และมุมหมุนผิดพลาดน้อยกว่า 0.34 องศา ตามลำดับ

5.4 การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลการฝึกฝนร่วมหลายโดเมน (Combined domain single model)

เนื่องจากในโลกแห่งความเป็นจริง เป็นไปได้ยากที่เราจะหาจำนวนชุดฝึกฝนภาพถ่ายในร่มและกลางแจ้งในอัตราส่วนที่เท่ากันได้เสมอไป และเนื่องจากในกรณีของงานวิจัยนี้ ชุดข้อมูลในร่ม ชุดที่ 1 มีปริมาณมากกว่าชุดข้อมูลกลางแจ้ง ชุดที่ 2 หลายเท่า เราจึงได้นำจำนวนภาพฝึกฝนมาแบ่งเป็น 3 สัดส่วน โดยนำชุดข้อมูลในร่มชุดที่ 1 และชุดข้อมูลกลางแจ้งชุดที่ 2 มาแบ่งออกเป็นชุดฝึกฝนจำนวน 500, 1000 และ 1500 ภาพ จากนั้นสร้างชุดข้อมูลฝึกฝนร่วมกันของชุดข้อมูลในร่มชุดที่ 1 และชุดข้อมูลกลางแจ้งชุดที่ 2 ด้วยอัตราส่วน 1:1, 3:1 และ 1:3 ตามลำดับ เพื่อทดสอบผลกระทบของสัดส่วนข้อมูลที่แตกต่าง เมื่อนำมาฝึกฝนโมเดลแล้วจะมีผลกระทบต่อการประมาณค่ามุมหมุนและการเลื่อนของโมเดลอย่างไร

โครงสร้าง CNN ที่นำเสนอจะใช้เป็น CNN 1 มิติ โดยจะทำการวิเคราะห์พารามิเตอร์โครงสร้างของโมเดลและ optimizer เพื่อเลือกโครงสร้างโมเดลที่ดีที่สุด ได้แก่ การวิเคราะห์จำนวน filter และจำนวน Node ของ CNN layer, รูปแบบของ Activation function ที่ใช้, ชนิดของ optimizer, ขนาด Batch_size, จำนวน Dense layer และจำนวนรอบ epoch ที่เหมาะสม โดยกำหนดการปรับค่าจำนวน filter [16,32,64,128], จำนวนโหนดในเลเยอร์ [16,32,64,128], Output Activation ที่ใช้ทดสอบ [tanh, relu, linear], โดยทดสอบ optimizer แบบ Adam และ Adadelta, batch_size = [32], Dense layer มีการปรับจำนวน 2 และ 4 เลเยอร์ (โดยการปรับจำนวน Dense layer จำนวน 2 layer จะมีคำว่า Reduce Dense ต่อท้าย) และ จำนวน epoch = 200 และ 500 รอบ ตามลำดับ ในการวัดประสิทธิภาพผลลัพธ์จะพิจารณาโดยนำชุดทดสอบของชุดข้อมูลที่ 1 และ 2 มาทำการทดสอบ เพื่อให้ได้ค่าความผิดพลาดเฉลี่ยกำลังสองของชุดสอบทั้งสองประเภทน้อยที่สุด ดังแสดงในตารางที่ 5.5

ตารางที่ 5.5 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของการทำนายมุมหมุนและการเลื่อนของการทดลองของโมเดลการฝึกฝนร่วมหลายโดเมนข้อมูล (โมเดล 4.4)

การแบ่งจำนวนภาพฝึกฝน	**activation**	**พารามิเตอร์ที่ดีที่สุดสำหรับมุมหมุน**	**พารามิเตอร์ที่ดีที่สุดสำหรับการเลื่อน**	**RMSE ของมุมหมุนและการเลื่อนที่น้อยที่สุด**
การแบ่งจำนวนภาพฝึกฝนแบบที่ 1 (Set 1 = 1:1)	Tanh+linear	128,128,32,32 adadelta 500 epochs	128,32,16,16 adadelta 500 epochs	0.616
	Tanh+linear (Reduce Dense)	128,32,32,16 adam 200 epochs	128,16,16,16 adadelta 500 epochs	0.655
	Tanh+relu+linear	128,32,32,16 adadelta 500 epochs	128,16,64,16 adadelta 500 epochs	**0.474**
	Tanh+relu+linear (Reduce Dense)	128,128,128,16 adam 500 epochs	128,16,16,16 adadelta 500 epochs	0.602

ตารางที่ 5.5 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของการทำนายมุมหมุนและการเลื่อนของการทดลองของโมเดลการฝึกฝนร่วมหลายโดเมนข้อมูล (โมเดล 4.4)

การแบ่งจำนวนภาพฝึกฝนแบบที่ 2 (Set 2 = 3:1)	Tanh+linear	64,64,32,16 adadelta 500 epochs	64,64,32,16 adadeltadate 500 epochs	0.675
	Tanh+linear (Reduce Dense)	64,32,32,16 adadelta 500 epochs	128,16,32,16 adadelta 200 epochs	**0.548**
	Tanh+relu+linear	128,16,128,32 adadelta 500 epochs	128,16,32,16 adam 500 epochs	0.551
	Tanh+relu+linear (Reduce Dense)	128,64,128,16 adam 500 epochs	16,32,32,16 adadelta 200 epochs	0.697
การแบ่งจำนวนภาพฝึกฝนแบบที่ 3 (Set 3 = 1:3)	Tanh+linear	128,128,64,32 adadelta 500 epochs	128,128,64,16 adadelta 500 epochs	0.620
	Tanh+linear (Reduce Dense)	128,64,32,64 adam 500 epochs	32,64,64,16 adadelta 500 epochs	0.684
	Tanh+relu+linear	128,128,128,16 adam 500 epochs	128,32,32,16 adadelta 500 epochs	**0.495**
	Tanh+relu+linear (Reduce Dense)	128,128,64,16 adam 500 epochs	128,128,64,16 adadelta 500 epochs	0.548

ตารางที่ 5.5 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองของการทำนายมุมหมุนและการเลื่อนจากการฝึกฝนชุดข้อมูลในร่มและกลางแจ้ง 3 ชุด โดยใช้ชุดทดสอบในร่มและกลางแจ้งอย่างละ 578 ภาพ จากผลการทดลองพบว่าค่าความผิดพลาดเฉลี่ยกำลังสองของการทดสอบการฝึกฝนแบบที่ 1 ที่น้อยที่สุดคือ 0.474 ซึ่งใช้ activator tanh+relu+linear จำนวนโหนด [128, 32, 32, 16] ใน layer 1 ถึง

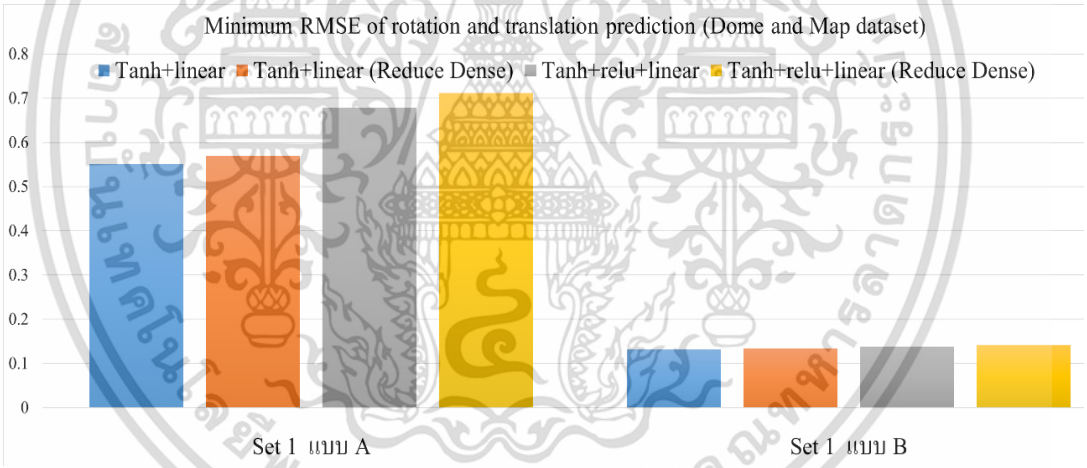
4 ตามลำดับ optimizer adadelta จำนวน 500 รอบ และ Dense 4 layers สำหรับการหาค่าประมาณมุมหมุน และจำนวนโหนด [128, 16, 64, 16] ใน layer 1 ถึง 4 ตามลำดับ optimizer adadelta จำนวน 500 รอบ และ Dense 4 layers สำหรับการหาค่าประมาณของการเลื่อน, ค่าความผิดพลาดเฉลี่ยกำลังสองของการทดสอบการฝึกฝนแบบที่ 2 ที่น้อยที่สุดคือ 0.548 ซึ่งใช้ activator tanh+linear (Reduce dense) จำนวนโหนด [64, 32, 32, 16] ใน layer 1 ถึง 4 ตามลำดับ optimizer adadelta จำนวน 500 รอบ และ Dense 2 layers สำหรับการหาค่าประมาณมุมหมุน และจำนวนโหนด [128, 16, 32, 16] ใน layer 1 ถึง 4 ตามลำดับ optimizer adadelta จำนวน 200 รอบ และ Dense 2 layers สำหรับการหาค่าประมาณของการเลื่อน และค่าความผิดพลาดเฉลี่ยกำลังสองของการทดสอบการฝึกฝนแบบที่ 3 ที่น้อยที่สุดคือ 0.495 ซึ่งใช้ activator tanh+relu+linear (Reduce Dense) จำนวนโหนด [128, 128, 128, 16] ใน layer 1 ถึง 4 ตามลำดับ optimizer adam จำนวน 500 รอบ และ Dense 4 layers สำหรับการหาค่าประมาณมุมหมุนและจำนวนโหนด 128, 32, 32, 16 ใน layer 1 ถึง 4 ตามลำดับ optimizer adadelta จำนวน 500 รอบ และ Dense 2 layers

5.5 การวิเคราะห์ผลลัพธ์จากการหามุมหมุนและการเลื่อนด้วยโมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN (โมเดล 4.5)

เริ่มจากทำการฝึกฝนเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน (domain-specific agents) สำหรับแต่ละโดเมน โดยใช้ชุดภาพในร่มและกลางแจ้ง สำหรับเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพในร่ม และสำหรับเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพกลางแจ้ง เพื่อแต่ละโดเมนจะสร้างโมเดลการทำนายมุมหมุนและการเลื่อนพร้อมกันทั้ง 2 โมเดล โครงสร้างของเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน โครงสร้าง CNN ที่ใช้เป็น CNN 1 มิติ พารามิเตอร์โครงสร้างของโมเดลและ optimizer ที่ได้ทำการทดสอบเพื่อเลือกโครงสร้างโมเดลที่ดีที่สุดได้แก่ การวิเคราะห์จำนวน filter และจำนวน Node ของ CNN layer, รูปแบบของ Activation function ที่ใช้, ชนิดของ optimizer, ขนาด Batch_size, จำนวน Dense layer และจำนวนรอบ epoch ที่เหมาะสม โดยกำหนดให้ใช้ค่าความผิดพลาดเฉลี่ยกำลังสองเป็นค่าประเมินผลลัพธ์ โดยกำหนดการปรับค่าจำนวน filter [16,32,64,128], จำนวนโหนดในเลเยอร์ [16,32,64,128], Output Activation ที่ใช้ทดสอบ [tanh, relu, linear], โดยทดสอบ optimizer แบบ Adam และ Adadelta, batch_size = [32], Dense layer มีการปรับจำนวน 2 (จะมี Reduce Dense ต่อท้าย) และ 4 เลเยอร์ และ จำนวน epoch =

200 และ 500 รอบ ตามลำดับ จากนั้นเข้าสู่โมเดลฝึกฝนร่วมในโครงสร้าง CNN ที่ใช้เป็น CNN 2 มิติ ที่พารามิเตอร์โครงสร้างของโมเดลและ optimizer ที่ได้ทำการทดสอบเพื่อเลือกโครงสร้างโมเดลที่ดีที่สุดทดสอบด้วยพารามิเตอร์เช่นเดียวกันกับ CNN 1 มิติ ซึ่งขนาดฟีเจอร์ที่นำเข้า CNN 2 มิตินี้มี 2 แบบ คือ แบบ A เป็นการดึงคุณลักษณะได้จากเลเยอร์ output ของเอเจนท์เรียนรู้คุณลักษณะเฉพาะแต่ละโดเมน และมาจัดเรียงเป็น Feature ที่มีมิติข้อมูล 7x2 จากเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพในร่มและเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพกลางแจ้ง และแบบ B เป็นการดึงคุณลักษณะจากเลเยอร์ก่อน output layer เพื่อประกอบกันเป็น Feature ที่มีมิติขนาด 64x2 โดยมีคุณลักษณะของมุมหมุนและการเลื่อนอย่างละ 64 คุณลักษณะ จากเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพในร่มและเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมนภาพกลางแจ้ง ดังแสดงในรูปที่ 4.8

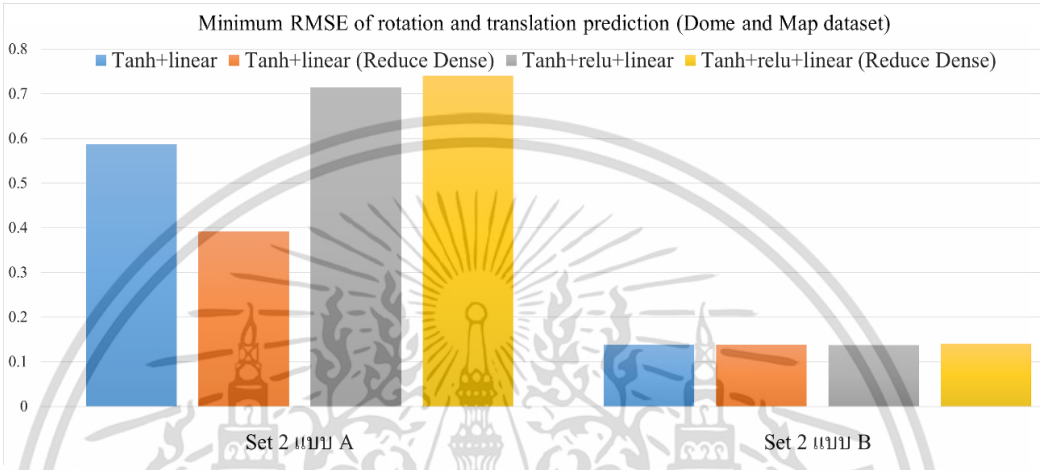
ผลการทดลองแสดงในรูปที่ 5.4 โดยจะเปรียบเทียบผลของรูปแบบการจัดเรียง Feature จากโมเดลเอเจนท์เรียนรู้คุณลักษณะเฉพาะโดเมน เข้าสู่เอเจนท์เรียนรู้ความสัมพันธ์ระหว่างโดเมน 2 แบบ คือแบบ A และแบบ B



รูปที่ 5.4 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดจากวิธี Ensemble CNN ของชุดข้อมูล Dome และ Map ที่มีการแบ่งจำนวนชุดข้อมูลแบบ Set 1 ด้วยการส่ง Feature แบบ A และ B

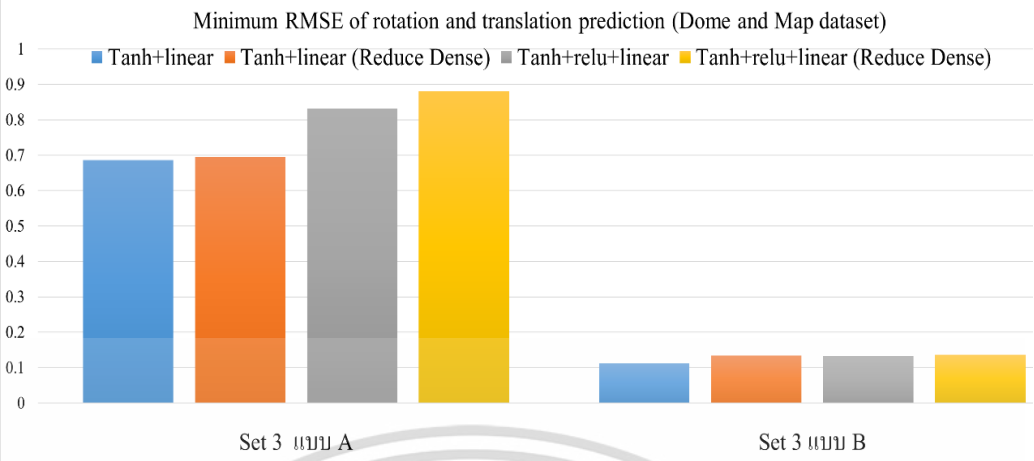
จากรูปที่ 5.4 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของผลลัพธ์ที่ถูกทดสอบด้วยจำนวนชุดข้อมูลในร่มและกลางแจ้ง อย่างละ 578 ภาพ ของการฝึกฝนชุดข้อมูลในร่มและชุดฝึกฝนกลางแจ้งชุดละ 1000 ภาพ จำนวนโหนดในเลเยอร์ที่ทดลองมี [16, 32, 64, 128], Output Activation ที่ใช้ทดสอบ [tanh, relu, linear] โดยทดสอบ optimizer แบบ [Adam, Adadelta], จำนวน epoch = [200, 500] และจำนวน Dense layer = [2 (Reduce Dense) (อ้างอิงในหัวข้อ

5.4)] ตามลำดับ จากการทดลองพบว่าค่าผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุด คือ 0.131 สำหรับการประกอบ Feature แบบ B โดยพารามิเตอร์ที่ดีที่สุดสำหรับภาพในร่ม ได้แก่ จำนวนโหนด [64, 128, 128, 128] ใน layer 1 ถึง 4 ตามลำดับ optimizer adam จำนวน 500 รอบ และ พารามิเตอร์ที่ดีที่สุดสำหรับภาพกลางแจ้ง ได้แก่ จำนวนโหนด [128, 128, 64, 16] ใน layer 1 ถึง 4 ตามลำดับ activation ที่ใช้คือ Tanh+linear optimizer adam จำนวน 500 รอบ และ Dense 4 layers



รูปที่ 5.5 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดจากวิธี Ensemble CNN ของชุดข้อมูล Dome และ Map ที่มีการแบ่งจำนวนชุดข้อมูลแบบ Set 2 ด้วยการส่ง Feature แบบ A และ B

จากรูปที่ 5.5 แสดงผลลัพธ์ที่ถูกทดสอบด้วยจำนวนชุดข้อมูลในร่มและกลางแจ้ง อย่างละ 578 ภาพ ของการฝึกฝนชุดข้อมูลในร่ม 1500 ภาพ และชุดฝึกฝนกลางแจ้ง 500 ภาพ ตามลำดับ จากการทดลองพบว่าค่า average rmse ที่น้อยที่สุด คือ 0.137 สำหรับการประกอบ Feature แบบ B โดยพารามิเตอร์ที่ดีที่สุดสำหรับภาพในร่ม ได้แก่ activation คือ Tanh+relu+linear จำนวนโหนด [128, 128, 128, 64] ใน layer 1 ถึง 4 ตามลำดับ optimizer adam จำนวน 500 รอบ และ พารามิเตอร์ที่ดีที่สุดสำหรับภาพกลางแจ้ง ได้แก่ จำนวนโหนด [128, 128, 128, 128] ใน layer 1 ถึง 4 ตามลำดับ optimizer adam จำนวน 200 รอบ และ Dense 4 layers



รูปที่ 5.6 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดจากวิธี Ensemble CNN ของชุดข้อมูล Dome และ Map ที่มีการแบ่งจำนวนชุดข้อมูลแบบ Set 3 ด้วยการส่ง Feature แบบ A และ B

จากรูปที่ 5.6 แสดงผลลัพธ์ที่ถูกทดสอบด้วยจำนวนชุดข้อมูลในร่มและกลางแจ้ง อย่างละ 578 ภาพ ของการฝึกฝนชุดข้อมูลในร่ม 500 ภาพ และชุดฝึกฝนกลางแจ้ง 1500 ภาพ จากการทดลองพบว่าค่าเฉลี่ยความผิดพลาดกำลังสองที่น้อยที่สุด คือ 0.112 สำหรับการประกอบ Feature แบบ B โดยพารามิเตอร์ที่ดีที่สุดสำหรับภาพในร่ม ได้แก่ activation คือ Tanh+linear, จำนวนโหนด [128, 128, 128, 32] ใน layer 1 ถึง 4 ตามลำดับ optimizer adadelta จำนวน 200 รอบ และพารามิเตอร์ที่ดีที่สุดสำหรับภาพกลางแจ้ง ได้แก่ จำนวนโหนด [128, 128, 16, 16] ใน layer 1 ถึง 4 ตามลำดับ optimizer adam จำนวน 500 รอบ

โดยภาพรวมกระบวนการทำงานของระบบการทำนายมุมหมุนและการเลื่อนในสามมิติจากภาพสองมิติของงานวิจัยนี้ได้นำเสนอ จากโมเดลการเรียนรู้แบบ Ensemble CNN ให้ค่าความผิดพลาดของการทำนายมุมหมุนและการเลื่อนน้อยที่สุด โดยจากการทดลองจะเห็นว่า การจัดรูปแบบ Feature แบบ B ให้ค่าความผิดพลาดของการทำนายมุมหมุนและการเลื่อนน้อยที่สุด ซึ่งการทดลองของเคสที่ดีที่สุดของแต่ละแบบนั้น พารามิเตอร์สำหรับชุดข้อมูลในร่มและกลางแจ้งไม่เหมือนกัน หากเราอยากเลือกพารามิเตอร์เดียวของชุดข้อมูลในร่มและกลางแจ้งสำหรับพารามิเตอร์ที่ใช้ทดสอบเดียวกัน แล้วให้ค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุด สามารถทำได้ โดยได้แสดงการเปรียบเทียบค่าความผิดพลาดทั้งสองกรณีดังกล่าว ในตารางที่ 5.6

ตารางที่ 5.6 แสดงค่าความผิดพลาดของการทำนายมุมหมุนและการเลื่อนน้อยที่สุดของการทดลองด้วยรูปแบบ Feature แบบ B โดยพารามิเตอร์ของเคสที่ดีที่สุดของชุดภาพถ่ายในร่ม (Dome) และกลางแจ้ง (Map) ต่างกัน และพารามิเตอร์ของเคสที่ดีที่สุดของชุดภาพถ่ายในร่มและกลางแจ้งเดียวกัน

	ค่าความผิดพลาดเฉลี่ยกำลังสอง (Average RMSE) ที่น้อยที่สุดของพารามิเตอร์แต่ละชุดทดสอบ (Dome, Map)			**ค่าความผิดพลาดเฉลี่ยกำลังสอง (Average RMSE) ที่น้อยที่สุดของพารามิเตอร์เดียวกันของชุดทดสอบทั้งสอง (Dome, Map)**	
การแบ่งจำนวนภาพฝึกฝน	**พารามิเตอร์ที่ดีที่สุดสำหรับชุด Dome**	**พารามิเตอร์ที่ดีที่สุดสำหรับชุด Map**	**Average RMSE ที่น้อยที่สุด**	**พารามิเตอร์ที่ดีที่สุดของทั้ง Dome และ Map**	**Average RMSE ที่น้อยที่สุด**
Set 1	Tanh+linear 64,128,128,128 adam 500 epochs	Tanh+linear 128,128,64,16 adam 500 epochs	0.131	Tanh+linear (Reduce Dense) 128,128,128,32 adam 500 epochs	0.134
Set 2	Tanh+relu+linear 128,128,128,64 adam 500 epochs	Tanh+relu+linear 128,128,128,128 adam 200 epochs	0.137	Tanh+linear (Reduce Dense) 128,128,128,32 adam 500 epochs	0.139
Set 3	Tanh+linear 128,128,128,32 adadelta 200 epochs	Tanh+linear 128,128,16,16 adam 500 epochs	0.112	Tanh+linear 128,128,128,32 adadelta 200 epochs	0.113

จากตารางที่ 5.6 แสดงการสรุปพารามิเตอร์ ที่ให้ค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของชุดทดสอบแต่ละโดเมน และแสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของ พารามิเตอร์เดียวกันของทั้งสองชุดทดสอบ จะเห็นว่าให้ค่าความผิดพลาดเฉลี่ยกำลังสองของมุมหมุนและการเลื่อนไม่แตกต่างกันมากคือ 0.006233647

ตารางที่ 5.7 ค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของโมเดลเดี่ยวฝึกฝนร่วมหลากโดเมนและโมเดล Ensemble CNN ที่นำเสนอ

การแบ่งจำนวนภาพฝึกฝน	**รูปแบบ activation**	**โมเดล 4.3**	**โมเดล 4.4**	**โมเดล 4.5 แบบ A**	**โมเดล 4.5 แบบ B**
Set 1	Tanh+linear	2.501	0.616	**0.552**	**0.131**
	Tanh+linear (Reduce Dense)		0.655	0.569	0.133
	Tanh+relu+linear		**0.474**	0.678	0.138
	Tanh+relu+linear (Reduce Dense)		0.602	0.711	0.141
Set 2	Tanh+linear		0.675	0.586	0.13760

ตารางที่ 5.7 ค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของโมเดลเดี่ยวฝึกฝนร่วมหลากโดเมนและโมเดล Ensemble CNN ที่นำเสนอ

การแบ่งจำนวนภาพฝึกฝน	รูปแบบ activation	โมเดล 4.3	โมเดล 4.4	โมเดล 4.5 แบบ A	โมเดล 4.5 แบบ B
	Tanh+linear (Reduce Dense)		0.548	**0.391**	0.13762
	Tanh+relu+linear		0.551	0.714	**0.13713**
	Tanh+relu+linear (Reduce Dense)		0.697	0.740	0.14031
Set 3	Tanh+linear		0.620	0.686	**0.112**
	Tanh+linear (Reduce Dense)		0.684	0.695	0.134
	Tanh+relu+linear		0.495	0.831	0.133
	Tanh+relu+linear (Reduce Dense)		0.548	0.880	0.136

ตารางที่ 5.7 แสดงค่าความผิดพลาดเฉลี่ยกำลังสองที่น้อยที่สุดของโมเดลเดี่ยวฝึกฝนร่วมหลากโดเมน (4.4) และโมเดล Ensemble CNN ที่นำเสนอ (4.5 แบบ B) จากการทดสอบของแต่ละการแบ่งสัดส่วนชุดฝึกฝนและแต่ละ activation เห็นว่าโมเดล Ensemble CNN ที่นำเสนอ (4.5 แบบ B) ให้ค่าความผิดพลาดของการทำนายมุมหมุนและการเลื่อนน้อยที่สุดคือ 0.112 โดยน้อยกว่าแบบ 4.4 และ (4.5 แบบ A) อยู่ 0.362 และ 0.279 ตามลำดับ นอกจากนี้ ค่าความผิดพลาดเฉลี่ยกำลังสองของโมเดล Ensemble CNN ที่น้อยที่สุดของแต่ละชุดการแบ่งจำนวนภาพฝึกฝน set 1 ถึง 3 ให้ค่าความผิดพลาดการทำนายมุมหมุนและการเลื่อนน้อยกว่าการใช้โมเดล 4.3 อยู่ 2.370, 2.363 และ 2.389 แบบตามลำดับ

ตารางที่ 5.8 แสดงการนำโมเดลมาทดสอบชุดข้อมูล Cambridge ซึ่งมักใช้ในการประเมินประสิทธิภาพของอัลกอริธึมการประเมินท่าทางกล้องจากโมเดลที่ถูกฝึกฝนด้วยชุดฝึกฝนของชุดภาพในร่ม (Dome) และชุดภาพกลางแจ้ง (Map) ซึ่งชุดข้อมูลชุดที่ 3 Cambridge dataset ใช้รูปแบบการถ่ายภาพที่แตกต่างกันเมื่อเทียบกับชุดข้อมูล Dome (ภาพถ่ายในร่มที่มีมุมมองการถ่ายวัตถุแตกต่างกัน) และชุดข้อมูล Map (ภาพถ่ายกลางแจ้งที่มีมุมมองถ่ายด้านล่างจากโดรนที่บินในแนวนอน) จากตารางที่ 5.8 ผลลัพธ์สำหรับ 10 โมเดลอ้างอิง [61] โมเดล LearnLess (DSAC++) [60] ให้ค่าความผิดพลาดมัธยฐานน้อยที่สุดสำหรับชุดย่อย 1 ถึง 5 อย่างไรก็ตามเทคนิคของเราแสดงค่าความผิดพลาดมัธยฐานของมุมหมุนและการเลื่อนต่ำกว่าวิธีอื่นๆ อย่างมีนัยสำคัญ โดยข้อมูลชุดย่อย 1 ถึง 5 โมเดล Ensemble CNN ที่นำเสนอ ให้ค่าความผิดพลาดค่ามัธยฐานของมุมหมุนและการ

เลื่อนน้อยกว่าวิธีของงานวิจัยอ้างอิง [60] 0.51 องศา และ 0.84 เมตร ตามลำดับ และน้อยกว่าวิธี Active Search (SfM) [61] 0.89 องศา และ 2.2 เมตร สำหรับข้อมูลชุดย่อย 2 ถึง 6 นอกจากนี้ เมื่อเทียบกับโมเดล CNN โมเดล Ensemble CNN มีค่าความผิดพลาดมัธยฐานน้อยกว่า 32.21 เมตร, 0.37 เมตร และ 39.58 องศา, 1 องศา ตามลำดับ

ตาราง 5.8 ค่าความผิดพลาดค่ามัธยฐาน สำหรับการประมาณท่าทางกล้องของการเลื่อน (เมตร) และมุมหมุน (องศา) โดยชุดทดสอบ Cambridge dataset

Algorithm	Great Court	Kings College	Old Hospital	Shop Facade	St Marys Church	Street
PoseNet [53]	NA	1.97 เมตร, 5.40 องศา	2.31 เมตร, 5.38 องศา	1.46 เมตร, 8.08 องศา	2.65 เมตร, 8.48 องศา	3.67 เมตร, 6.50 องศา
Dense PoseNet [53]	NA	1.66 เมตร, 4.86 องศา	2.57 เมตร, 5.14 องศา	1.41 เมตร, 7.18 องศา	2.45 เมตร, 7.96 องศา	2.96 เมตร, 6.00 องศา
Bayesian PoseNet [54]	NA	1.74 เมตร, 4.06 องศา	2.57 เมตร, 5.14 องศา	1.25 เมตร, 7.54 องศา	2.11 เมตร, 8.38 องศา	2.14 เมตร, 4.96 องศา
LSTเมตร-Pose [55]	NA	0.99 เมตร, 3.65 องศา	1.51 เมตร, 4.29 องศา	1.18 เมตร, 7.44 องศา	1.52 เมตร, 6.68 องศา	NA
SVS-Pose [56]	NA	1.06 เมตร, 2.81 องศา	1.50 เมตร, 4.03 องศา	0.63 เมตร, 5.73 องศา	2.11 เมตร, 8.11 องศา	NA
PoseNet + Reprojection error pose loss [57]	7.00 เมตร, 3.7 องศา	0.99 เมตร, 1.1 องศา	2.17 เมตร, 2.9 องศา	1.05 เมตร, 4.0 องศา	1.49 เมตร, 3.40 องศา	20.7 เมตร, 25.7 องศา
VLocNet [58]	NA	0.83 เมตร, 1.42 องศา	1.07 เมตร, 2.41 องศา	0.593 เมตร, 3.53 องศา	0.63 เมตร, 3.91 องศา	NA
DSAC [59]	2.80 เมตร, 1.5 องศา	0.30 เมตร, 0.5 องศา	0.33 เมตร, 0.6 องศา	0.09 เมตร, 0.40 องศา	0.55 เมตร, 16 องศา	NA
LearnLess(DSAC++) [60]	0.4 เมตร, 0.2 องศา	0.18 เมตร, 0.3 องศา	0.20 เมตร, 0.3 องศา	0.06 เมตร, 0.30 องศา	0.13 เมตร, 0.4 องศา	NA
Active Search [61]	NA	0.42 เมตร, 0.6 องศา	0.44 เมตร, 1.0 องศา	0.12 เมตร, 0.40 องศา	0.19 เมตร, 0.5 องศา	0.85 เมตร, 0.8 องศา
CNN (โมเดล 4.3)	0.16 เมตร, 0.18 องศา	0.20 เมตร, 0.21 องศา	0.22 เมตร, 0.48 องศา	0.11 เมตร, 0.20 องศา	0.10 เมตร, 0.39 องศา	0.77 เมตร, 0.76 องศา
โมเดล Ensemble CNN ที่นำเสนอ	**0.06 เมตร, 0.12 องศา**	**0.18 เมตร, 0.05 องศา**	**0.09 เมตร, 0.46 องศา**	**0.09 เมตร, 0.13 องศา**	**0.09 เมตร, 0.08 องศา**	**0.68 เมตร, 0.38 องศา**

บทที่ 6

สรุปผลและแนวทางในการพัฒนา

วิทยานิพนธ์เล่มนี้ได้นำเสนออัลกอริธึมการประมาณค่ามุมหมุนและการเลื่อนของการประกอบกลับสามมิติโดยใช้โมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN เพื่อเพิ่มประสิทธิภาพและความเร็วของการหาพารามิเตอร์ภายนอกของกล้อง ชุดข้อมูลที่ใช้ในการทดลองเป็นชุดภาพถ่ายที่กลางแจ้งและในร่มถูกนำมาจาก 3 แหล่งที่เป็นที่นิยม ซึ่งมีการตรวจสอบประสิทธิภาพโดยใช้การวัดค่าความผิดพลาดเฉลี่ยกำลังสองและค่าความผิดพลาดมัธยฐานของมุมหมุนและการเลื่อนของภาพ

6.1 การวิเคราะห์และสรุปผลการดำเนินงานวิจัย

สำหรับงานวิจัยนี้ได้นำเสนอการทำนายมุมหมุนและการเลื่อนที่เป็นพารามิเตอร์ภายนอกกล้องสำหรับการประกอบกลับสามมิติ เพื่อลดระยะเวลาและความผิดพลาดมากที่สุด ซึ่งเทคนิคที่ได้นำเสนอคือเทคนิคการประมาณค่าถดถอยของการวางท่าของกล้องด้วยโครงข่ายประสาทแบบคอนโวลูชัน โดยอาศัยการถ่ายโอนคุณลักษณะเบื้องต้นจาก VGG19 ผ่านการลดมิติด้วยเทคนิค LSA เพื่อเข้า CNN ที่นำเสนอ โดยในงานวิจัยนี้นำเสนอโมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN จากผลการวัดประสิทธิภาพของการฝึกฝนข้อมูลหลากหลายรูปแบบไม่ว่าจะเป็นความหลากหลายของมุมมองการถ่ายภาพ ความสว่าง ความเบลอ และการเลื่อน จากผลการทดลองแสดงให้เห็นว่าเทคนิคที่นำเสนอสามารถแก้ไขข้อจำกัดของงานวิจัย [7] เมื่อเทียบกับงานวิจัยอ้างอิงพบว่าโมเดลที่นำเสนอให้ค่าความผิดพลาดของการทำนายโดยรวมน้อยกว่าวิธีของงานวิจัยอ้างอิง

6.2 ข้อจำกัดและขอบเขตของงานวิจัย

ในงานวิจัยฉบับนี้เป็นการประมาณค่ามุมหมุนและการเลื่อนซึ่งเป็นพารามิเตอร์ภายนอกกล้องสำหรับการประกอบกลับสามมิติภาพถ่ายเป็นภาพถ่ายในร่มและกลางแจ้ง โดยใช้โมเดลการเรียนรู้แบบโครงข่ายเอเจนท์ Ensemble CNN ซึ่งสามารถทำนายมุมหมุนและการเลื่อนของภาพที่มีความสว่างมาก (กลางแจ้ง) และความสว่างน้อยมาก (ในร่ม) ได้อย่างมีประสิทธิภาพ โดยจะให้ประสิทธิภาพดีที่สุดกับรูปแบบการถ่ายด้วยโดรนขับเคลื่อนไปเหนือรอบวัตถุแล้วถ่ายภาพลงมา

อย่างไรก็ดีสำหรับลักษณะการถ่ายแบบชุดข้อมูล Street โมเดลที่นำเสนอถึงแม้จะให้ค่าความผิดพลาดมัธยฐานน้อยกว่าวิธีอื่น แต่ก็ให้ประสิทธิภาพต่ำที่สุดในชุดข้อมูลทดสอบทั้งหมด เนื่องจากองค์ประกอบของภาพส่วนใหญ่เป็นถนนที่มีสีและรูปร่างคล้ายกัน และมีเพียงรายละเอียดของร้านข้างถนนเพียงเล็กน้อยจึงทำให้ยากต่อการประมาณค่ามุมหมุนและการเลื่อนของชุดภาพ

6.3 แนวทางในการพัฒนา

จากที่กล่าวถึงข้อจำกัดของงานวิจัยข้างต้น ดังนั้นการพัฒนาอัลกอริธึมโดยการเพิ่มโมเดลการเรียนรู้เฉพาะทางเกี่ยวกับความลึก จากนั้นนำคุณลักษณะที่ได้จากโมเดลเรียนรู้เฉพาะทางต่างๆมาประยุกต์ใช้ร่วมในการทำนายขั้นตอนสุดท้ายจะสามารถเพิ่มความถูกต้องของการประมาณค่ามุมหมุนและการเลื่อนให้สูงขึ้น

เอกสารอ้างอิง

[1] Cosmas, J., Itegaki, T., Green, D., Joseph, N., Gool, L.V., Zalesny, A., Vanrintel, D., Leberl, F., Grabner, M., Schindler, K., Karner, K., Gervautz, M., Hynst, S., Waelkens, M., Vergauwen, M., Pollefeys, M., Cornelis, K., Vereenooghe, T., Sablatnig, R., Kampel, M., Axell, P., Meyns, E. 2003. "Providing multimedia tools for recording, reconstruction, visualisation and database storage/access of archaeological excavations." **4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage VAST (2003).** : 165-174.

[2] Russell, B.C., Martin-Brualla, R., Butler, D.J., Seitz, S.M., Zettlemoyer, L. 2013. "3D Wikipedia: using online text to automatically label and navigate reconstructed geometry." **SIGGRAPH Asia (2013).**

[3] Xiao, J., Furukawa, Y. 2012. "Reconstructing the world's museums." **ECCV (2012). :** 668–681. https://doi.org/10.1007/978-3-642-33718-5_48

[4] Li, Y., Snavely, N., Huttenlocher, D., Fua, P. 2012. "Worldwide pose estimation using 3D Point clouds. In: Fitzgibbon," A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) **ECCV 2012**. LNCS, vol. 7572 : 15–29. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33718-5_2.

[5] Zeisl, B., Sattler, T., Pollefeys, M. 2015. "Camera pose voting for large-scale image-based localization." **2015 IEEE International Conference on Computer Vision (ICCV).** : 2704-2712, doi: 10.1109/ICCV.2015.310.

[6] Middelberg, S., Sattler, T., Untzelmann, O., Kobbelt, L. 2014. "Scalable 6-DOF localization on mobile devices." Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) **ECCV 2014.** LNCS, vol. 8690, Springer, Heidelberg (2014). : 268–283. doi:10.1007/978-3-319-10605-2_18

[7] Lynen, S., Sattler, T., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R. 2015. "Get out of my lab: large-scale, real-time visual-inertial localization." **RSS (2015).**

[8] Ahmed, A.A. Al-Shaboti, M. Al-Zubairi, A. "An indoor emergency guidance algorithm based on wireless sensor networks." **In Proceedings of the 2015**

International Conference on Cloud Computing (ICCC), Riyadh, Saudi Arabia, 26–29 April 2015 : 1–5.

[9] Chen, C., Tang, L. 2019. "BIM-based integrated management workflow design for schedule and cost planning of building fabric maintenance." **Automation in Construction**, 107, 102944. https://doi.org/10.1016/j.autcon.2019.102944.

[10] Chen, K., Lai, YK. and Hu, SM. 2015. "3D indoor scene modeling from RGB-D data." **A survey.** Comput. Vis. Media 2015, 1, 267–278.

[11] Dasgupta, S., Fang, K., Chen, K. and Savarese, S. "Delay: Robust spatial layout estimation for cluttered indoor scenes." **IEEE Conference on Computer Vision and Pattern Recognition**, Las Vegas, NV, USA, 26 June–1 July 2016 : 616–624.

[12] Liu, C., Schwing, A.G., Kundu, K., Urtasun, R. and Fidler, S. 2015. "Rent3D: floor-plan priors for monocular layout estimation." **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).** : 3413-3421.

[13] Kushal, A., Self, B., Furukawa, Y., Gallup, D., Hernandez, C., Curless, B., Seitz, S. 2012. "Photo Tours." **2012 Second International Conference on 3D Imaging, Modeling**, Processing, Visualization & Transmission : 57-64. doi: 10.1109/3DIMPVT.2012.62.

[14] Snavely, N., Garg, R., Seitz, S.M. and Szeliski, R. 2008. "Finding paths through the world's photos." **ACM SIGGRAPH 2008 papers (SIGGRAPH '08).** Association for Computing Machinery, New York, NY, USA, Article 15 : 1–11. https://doi.org/10.1145/1399504.1360614.

[15] Isikdag, U., Zlatanova, S. and Underwood, J. 2013. "A BIM-Oriented Model for supporting indoor navigation requirements." **Computers Environment and Urban Systems.** 41 : 112–123.

[16] Strecha, C., Krull, M., Betschart, S. 2014. "The chillon project: aerial/terrestrial and indoor integration." **Technical report, Pix4D.** https://pix4d.com/chillon/

[17] Mortensen, E. N., Deng, H., and Shapiro, L. 2005. "A sift descriptor with global context." **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),** vol. 1 : 184–190.

[18] Agarwal, S., Snavely, N., Simon, I., Seitz, S. and Szeliski, R. 2009. “Building Rome in a Day.” **2009 IEEE 12th International Conference on Computer Vision**. : 72-79. doi: 10.1109/ICCV.2009.5459148.

[19] Jan-Michael, F., Pierre, F., David, G., Tim, J., Rahul, R., Changchang, W., Yi-Hung, J., Enrique, D., Brian, C., Svetlana, L., and Marc, P. 2010. “Building Rome on a cloudless day.” **the 11th European conference on Computer vision**: Part IV (ECCV'10). Springer-Verlag, Berlin, Heidelberg : 368–381.

[20] Xiaowei, L., Changchang, W., Christopher, Z., Svetlana, L. and Jan-Michael, F. 2008. “Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs.” **the 10th European Conference on Computer Vision**: Part I (ECCV '08) : 427–440. https://doi.org/10.1007/978-3-540-88682-2_33.

[21] Noah, S., Steven, M. S. and Richard, S. 2006. “Photo tourism: exploring photo collections in 3D.” **ACM Trans. Graph. 25**, 3 (July 2006), 835–846. https://doi.org/10.1145/1141911.1141964.

[22] Sinha, S. N., Drew, S. and Richard, S. 2010. “A Multi-stage Linear Approach to Structure from Motion.” **ECCV 2010.** https://doi.org/10.1007/978-3-642-35740-4_21

[23] Yihui-he. “3D-reconstruction : two view structure from motion” [Online]. Available : https://github.com/yihui-he/3D-reconstruction. 2016.

[24] Lynen, S., Zeisl, B., Aiger, D., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R. and Sattler, T. 2019. “Large-scale, Real-Time Visual-Inertial Localization Revisited.” **The International Journal of Robotics Research.** 39(9):027836492093115 : 1-21. https://doi.org/10.1177/0278364920931151.

[25] Anastasios, I. M., Nikolas, T., Stergios, I. R., Andrew, E. J., Adnan, A. and Larry, M. "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing." **in IEEE Transactions on Robotics**, vol. 25, no. 2April 2009 : 264-280. doi: 10.1109/TRO.2009.2012342.

[26] Guillaume, B., Romuald, A. and Roland, C. 2013. “Making visual slam consistent with geo-referenced landmarks.” **2013 IEEE Intelligent Vehicles Symposium (IV), 2013**. : 553–558.

[27] Wietrzykowski, J. and Belter, D. "Stereo Plane R-CNN: Accurate Scene Geometry Reconstruction Using Planar Segments and Camera-Agnostic Representation." **IEEE Robotics and Automation Letters**, vol. 7, no. 2, April 2022 : 4345-4352. doi: 10.1109/LRA.2022.3150841.

[28] Liu, C., Kim, K., Gu, J., Furukawa, Y. and Kautz, J. 2019. “PlaneRCNN: 3D plane detection and reconstruction from a single image.” **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).** : 4445-4454. doi: 10.1109/CVPR.2019.00458.

[29] Bryan, K., David, M. and James R. 2013. “Street view motion-from-structure-from-motion.” **IEEE International Conference on Computer Vision.** : 953-960,=. doi: 10.1109/ICCV.2013.122.

[30] Jonathan, V., Clemens, A., Gerhard, R. and Dieter, S. 2014. “Global Localization from Monocular SLAM on a Mobile Phone.” **IEEE Transactions on Visualization and Computer Graphics**, vol. 20, no. 4 : 531-539. doi: 10.1109/TVCG.2014.27.

[31] Tue-Cuong, D. and Anastasios, I. M. 2011. “Motion Tracking with Fixed-lag Smoothing: Algorithm and Consistency Analysis.” **2011 IEEE International Conference on Robotics and Automation (ICRA)**. : 5655-5662. doi: 10.1109/ICRA.2011.5980267.

[32] Gabe, S., Larry, M. and Gaurav, S. 2010. “Sliding window filter with application to planetary landing.” **Journal of Field Robotics.** No.5 : 587-608

[33] Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K. and Siegwart, R. 2015. “Keyframe-based visual–inertial odometry using nonlinear optimization.” **International Journal of Robotics Research**, vol 34, issue 3 (Mar 2015) : 314–334. https://doi.org/10.1177/0278364914554813.

[34] Kusupati, U., Cheng, S., Chen, R. and Su, H. 2020. “Normal assisted stereo depth estimation.” **2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).** : 2186–2196.

[35] Chang, J.-R. and Chen, Y.-S. 2018. “Pyramid stereo matching network.” **IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).** : 5410–5418.

[36] Wietrzykowski, J. and Skrzypczy, P. 2019. “PlaneLoc: Probabilistic global localization in 3-D using local planar features.” **Robot. Auton. Syst,** vol. 113 : 160–173. https://doi.org/10.1016/j.robot.2019.01.008.

[37] Hartley, R. and Zisserman, A. 2004. **Multiple View Geometry in Computer Vision (2 nd edition).** Cambridge: Cambridge University Press. , ISBN: 0521540518. doi:10.1017/CBO9780511811685.

[38] Lowe, D.G. 2004. “Distinctive Image Features from Scale-Invariant Keypoints.” **International Journal of Computer Vision.** 60(2) : 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[39] Lowe, D.G. 1999. "Object recognition from local scale-invariant features." **The Seventh IEEE International Conference on Computer Vision**, vol.2 : 1150-1157. doi: 10.1109/ICCV.1999.790410.

[40] Lindeberg, T. 1994 “Scale-space theory: A basic tool for analysing structures at different scales.” **Journal of Applied Statistics**, 21, 2 : 225–270.

[41] Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling. W. T. 1988. **Numerical recipes in C—the art of scientific computing.** Cambridge University Press. ISBN 0-521-35465-X. The Mathematical Gazette, 73(464), 167-170. doi:10.2307/3619708.

[42] Smola, A.J. and Schölkopf, B. 2004. “A tutorial on support vector regression.” **Statistics and Computing** : 199–222.

[43] Cortes, C. and Vapnik, V. 1995. “Support-vector networks.” **Machine Learning 20.** : 273–297. https://doi.org/10.1007/BF00994018.

[44] Karen S. and Andrew Z. 2015. “Very Deep Convolutional Networks For large-scale Image Recognition.” **ICLR**, https://arxiv.org/abs/1409.1556.

[45] Krizhevsky, A., Sutskever, I. and Hinton, G. E. 2012. “ImageNet classification with deep convolutional neural networks.” **NIPS.** : 1106–1114.

[46] Dipanjan, S. 2018. “A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning” [Online]. Available : https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

[47] Hinton, G., Dean, J. and Vinyals, O. 2014. “Distilling the Knowledge in a Neural Network.” : 1-9.

[48] Christopher, M. B. 1995. **Neural Networks for Pattern Recognition.** Oxford University Press ISBN:978-0-19-853864-6, Inc., USA.

[49] Barnston, A. G. 1992. “Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score.” **Weather and Forecasting,** vol.7, no.4 :.699-709.

[50] Jason, B. 2019. “Ensemble Learning Methods for Deep Learning Neural Networks” [Online]. Available : https://machinelearningmastery.com/ ensemble-methods-for-deep-learning-neural-networks/.

[51] Joo., H., Park, H. S., and Sheikh, Y. 2014. “MAP visibility estimation for large-scale dynamic 3D reconstruction” **2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),** Columbus, OH, USA : 1122-1129 doi:10.1109/CVPR.2014.147.

[52] Nex, F., Gerke, M., Remondino, F., Przybilla, H. J., Baumker, M. and Zurhorst, A. 2015. “ISPRS benchmark for multi-platform photogrammetry, Annals of the Photogrammetry.” **Remote Sensing and Spatial Information Science,** vol.II-3/W4, Munich, Germany : 135-142.

[53] Kendall, A., Grimes, M. and Cipolla, R. 2015. “PoseNet: A convolutional network for real-time 6-DoF camera relocalization.” **IEEE International Conference on Computer Vision (ICCV).** : 2938-2946, doi: 10.1109/ICCV.2015.336.

[54] Alex, K. and Cipolla, R. 2016. “Modelling uncertainty in deep learning for camera delocalization” **2016 IEEE International Conference on Robotics and Automation (ICRA).** : 4762-4769, arXiv Preprint, arXiv: 1509.05909.

[55] Bell, S., Zitnick, C. L., Bala, K. and Girshick, R. 2016. “Inside-outside Net: Detecting objects in context with skip pooling and recurrent neural networks.” **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),** Las Vegas, NV, USA, pp.2874-2883, doi: 10.1109/CVPR.2016.314.

[56] Naseer, T. and Burgard, W. 2017. “Deep regression for monocular camera-based 6-DoF global localization in outdoor environments.” **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).** : 1525-1530, doi: 10.1109/IROS.2017.8205957.

[57] Kendall, A. and Cipolla, R. 2017. “Geometric loss functions for camera pose regression with deep learning.“ **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),** Honolulu, HI, USA : 6555-6564.

[58] Abhinav, V., Radwan, N. and Burgard, W. 2018. “Deep auxiliary learning for visual localization and odometry.” **2018 IEEE International Conference on Robotics and Automation (ICRA).** : 6939-6946.

[59] Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S. and Rother, C. 2017. “DSAC – Differentiable RANSAC for camera localization.” **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),** Honolulu, HI, USA : 6684-6692, https://github.com/cvlabdresden/DSAC.

[60] Brachmann, E. and Rother, C. 2018. “Learning less is more – 6D camera localization via 3D surface regression.” **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),** Salt Lake City, UT, USA, : 4654-4662, https://github.com/vislearn/LessMore.

[61] Shavit, Y. and Ferens, R. 2019. “Introduction to camera pose estimation with deep learning.” **arXiv: Computer Vision and Pattern Recognition,** arXiv: 1907.05272.

ภาคผนวก

งานวิจัยที่ได้รับการตีพิมพ์

Prediction of 3D rotation and translation from 2D images

Bhattarabhorn Wattanacheep
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang Bangkok,
Thailand 10520
nroskool2@gmail.com

Orachat Chitsobhuk
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang Bangkok,
Thailand 10520
orachat.ch@kmitl.ac.th

ABSTRACT

The prediction of three-dimensional (3D) rotation and translation can be retrieved from two-dimensional (2D) images to build 3D models from large collections of images. In this paper, the process starts by extracting the features of images via transfer learning approach from Deep Neural Network model called VGG19. Even though the features extracted from VGG19 are usually adopted in image recognition application; in this research, we apply these features to the prediction model to obtain rotation and translation parameters. Due to the large size of the feature dimensions, it is necessary to perform dimensional reduction technique called latent semantic analysis (LSA) to decrease the feature dimensions and remain only the important ones. Then, the regression estimation technique based on the idea of Support Vector Machine (SVM) is used to predict the rotation and translation parameters. The accuracy is estimated by comparing the prediction results with the corresponding ground truth set. The average errors of rotation and translation of 3D prediction from 2D images are approximately 0.2419 degrees and 1.35 meters respectively.

CCS Concepts

• **Computing methodologies→Camera calibration.**

Keywords

3D Reconstruction; Image Processing; Robotics; Deep Learning.

1. INTRODUCTION

Image transformation is a process to convert a 3D world into a 2D image using a camera model with geometric relationship between a 3D position and its 2D corresponding projection onto the image plane. The model starts by processing the color and light through the pixel sensor within the camera. Two types of parameters needed to be estimated are intrinsic camera parameters- the parameters necessary to link the pixel coordinates of an image with the corresponding coordinates in the camera reference frame- and extrinsic camera parameters- the parameters that define the translation and orientation of the camera reference frame with respect to a known world reference frame. The intrinsic camera parameters include focal length, optical center, and skew coefficient can be obtained through the camera calibration process.

ICCCM 2019, July 27–29, 2019, Bangkok, Thailand

ACM ISBN 978-1-4503-7195-7/19/07…$15.00
https://doi.org/10.1145/3348445.3348485

However, in this paper, we focus on extrinsic camera parameter estimation consisting of rotation and translation in the X-axis, Y axis and Z-axis. Several techniques have been proposed to estimate these parameters from a single camera, stereo camera and multiple cameras.

Recent well-known work on estimating extrinsic camera parameters was a structure from motion, SfM, of a series of photos on social media sites such as Flickr [2-7]. SfM operation was mostly based on feature mapping as presented in Li et al. [4]. First, the images were grouped by finding the iconic images to construct the basic spanning tree structure of the image relation. Then, the rotation and translation were estimated using the SfM technique as presented by Singha's et al [6] with the linear SfM method. Most approaches to SfM from unstructured image collections were operated iteratively, starting with a small seed reconstruction, then growing through repeated integration of additional cameras and scene points. Even though such iterative approaches have been quite successful, they initiated two significant drawbacks. First, these methods tended to require heavy calculations on repeated non-linear optimization that attempted to refine camera parameters and scene structure as well as outlier rejection to remove inconsistent measurements. Second, these methods did not consider all images equally, thus led to different results depending on the order in which photos were considered. This sometimes can cause failure due to local minima or cascades of misestimated cameras. Such methods can also make estimation of rotation and translation parameters grow over time and introduce many errors.

Afterwards, Yihui-he [14] used the same SfM technique, starting with using SIFT to find the corresponding feature point between two images, then used the MSAC technique to filter the feature points that were not consistent with the original image instead of the RANSAC. This makes the projection of the camera more accurate. Although this method provides better performance, it still continues the incremental calculation structure. As a result, the calculation load cannot be reduced, and the order still affects the estimation efficiency. Therefore, the above-mentioned problems still cannot be solved.

Consequently, this paper proposes the prediction of 3D rotation and translation from 2D images of multiple cameras - every photo is used to calculate the rotation and translation simultaneously - using the machine learning of the features learned from deep learning model. The process starts by training the VGG19 to extract features from the sample images. These features are then used to predict the rotation and translation in 3D of image. In order to reduce large computation, the dimensional reduction technique called Latent Semantic Analysis (LSA) technique is adopted to remain just the dimension of important features. Finally, the features are regressed with the principle of Support Vector Machine, in which all images are considered equally. Therefore, the results will not depend on the sequence of photos

that have been processed. In addition, the proposed method introduces the noniterative process for parameter estimation, which help to improve computational efficiency.

The proposed algorithm is detailed in section 2 while section 3 presents the experimental results. Finally, summarize and conclusion are presented in section 4.

2. PROPOSED ALGORITHM

This research presents the prediction of 3D rotation and translation from multiple views of 2D images simultaneously. The overview of the proposed prediction system is shown in Figure 2.1. Image features are analyzed using Deep feature extraction; then the extrinsic camera parameters in terms of rotation and translation are further estimated by the principle of Support Vector Machine (SVM). The number of 4096 features of each image are extracted from the 7th fully connected layers of the VGG19. Since feature dimension is quite large, it would require excessive computational complexity. It is necessary to reduce the feature dimensions in order to preserve only the necessary ones. In this paper, we adopt the Latent Semantic Analysis (LSA) technique for the dimensional reduction task. Finally, rotation and translation parameters are estimated using support vector regression technique. We obtain the means of the root mean square error (RMSE) to determine prediction errors.

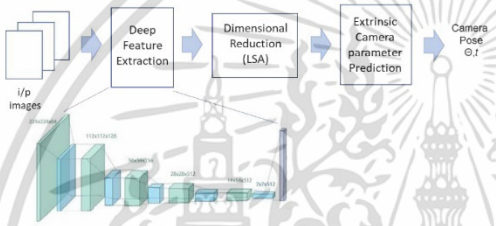


Figure 1. The overview of the proposed prediction of 3D rotation and translation from 2D images.

2.1 Deep Feature Extraction using VGG19 [8]

The input image will be randomly divided into training dataset and test dataset. Then, we apply deep feature extraction adopted from VGG19 deep learning model to estimate the important image features. Even though the model was originally used to extract features from the two-dimensional images for the object recognition applications; in this research, the features will be applied for learning and predicting rotation and translation in 3D from images.

Typically, the size of the input image submitted to the VGG19 is 224×224 pixels of the RGB color image obtained from ImageNet. The image is passed through a stack of convolutional (conv.) layers; where filters are employed with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, the model also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution. Spatial pooling is carried out by five max-pooling layers, which is performed over a 2×2 pixel window with a stride of 2. A stack of convolutional layers is followed by three fully connected (fc) layers: the first two layers consist of 4096 channels each while the third one performs 1000-way ImageNet Large Scale Visual Recognition Challenge (ILSVRC) classification. The final layer is the soft-max layer, which is classification output layer. Since the objective of our proposed system is to perform regression not classification, we choose to retrieve the image features from the second fc layer (fc7) and used them for estimating the designated parameters.

2.2 Dimensional Reduction [9]

Features extracted from previous process contain large dimensions and can introduce excessively high complexity in prediction of rotation and translation. As a result, feature dimension must be reduced in order to accelerate the prediction while preserve only the necessary meaningful ones. In this paper, the Latent Semantic Analysis (LSA) technique is adopted. It is constructed from a mathematical technique called Singular Value Decomposition (SVD). In this way, the number of rows will be reduced, but still retaining a similar image feature structure.

SVD is robust and reliable orthogonal matrix decomposition method. It is one of the most powerful computational tools in numerical linear algebra. In particular, SVD is commonly used to solve i) the unconstrained linear least squares problems, ii) matrix rank estimation and iii) canonical correlation analysis. Further, SVD tells that any matrix A with arbitrary dimensions $m \times n$ can be represented as orthogonal matrices U and V and a diagonal matrix D as followed.

$$A = UDV^T \tag{1}$$

where the columns of U and V are the left and right singular vectors, respectively, and D is a diagonal matrix whose diagonal entries are the singular values of A. Since matrix V is orthogonal, V^T can instead be V^{-1}. The diagonal of the matrix D consists of singular values ordered from the largest value in the first column to the smallest value in the last column of the matrix. Since the matrix A consists of larger number of rows than those of columns ($m \geq n$), it results in orthogonal matrix U of size $m \times n$, D is the diagonal matrix of size $n \times n$and V is the $n \times n$ orthogonal matrix. Therefore, the matrix product gives the relationship between the image and the feature, which can be written as: $A^TA = VDU^TUDV^T = VD^2V^{-1}$. D^2 is the eigen value of A^TA and the column of V is the eigen vector of A^TA. The singular value of A is the square root of the eigen value of A^TA, Therefore, the singular value is a real number and $AA^T = UD^2U^{-1}$. In order to understand the SVD, the rows of an $m \times n$ matrix A are represented as m images in a n-dimensional space and it is considered as the problem of finding the best k-dimensional subspace with respect to the set of images as shown in equation 2.

$$X_k = U_k \Sigma_k V_k^T \tag{2}$$

After finding best k-dimensional subspace of the features, they will be derived to obtain reduced dimension features and used for prediction of 3D rotation and translation. The prediction is performed using the Support Vector Regression method as discussed in the next section.

2.3 Support Vector Regression [10]

Support Vector Machine is one of the most popular machine learning algorithms. The concepts are relatively simple. Suppose $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset \mathcal{X} \times \mathbb{R}$ is a training data when $\mathcal{X}$ denotes the space of input format. The goal is to find a function $f(\boldsymbol{x})$ with $\mathcal{E}$ deviation from the actual target, y_i are received for all the

training data. Similarly, the error value will be ignored as long as it is less than the deviation. Equation 3 descries the case of linear relationship, where $\langle \cdot,\cdot \rangle$ is the dot product of $\mathcal{X}$, w is the small seek value.

$$f(x) = \langle w, x \rangle + b \qquad \text{where } w \in \mathcal{X}, b \in \mathbb{R} \quad (3)$$

The linear functions f estimates precision of all pairs (x_i, y_i) with $\mathcal{E}$; which is a possible convex optimization solution. However, some error values may be excluded. In practice, data may not be able to be linearly divided. In this case, the "soft margin" loss function may be used instead of slack variable $\xi_i, \xi_i^* \geq 0$ for training data samples that violate the support vector conditions. The conditions of the specified Slack value are shown in Figure 2

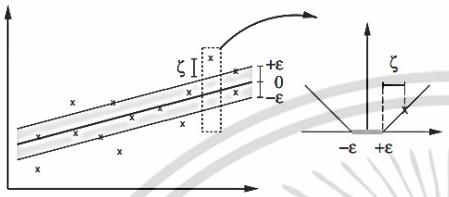


Figure 2. The soft margin loss setting for a linear SVM (from Scholkopf and Smola, 2002)

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|w\|^2 + C\textstyle\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to} && \begin{cases} y_i - \langle w, x \rangle - b \leq \mathcal{E} + \xi_i \\ \langle w, x \rangle + b - y_i \leq \mathcal{E} + \xi_i^* \end{cases} \end{aligned} \quad (4)$$

The constant C determines the importance of the scope and the need of the Slack variable. In other words, the low value C makes the method focus on the Soft-margin SVM, while the large value C makes the method focus on Hard-margin SVM.

The dot product of the pair of samples can be viewed as similarity within the samples. Therefore, it is possible to use SVM to divide the group of similar data without the need to use the actual feature values which are replaced by the similarity of data. These similarities can be formed as a Kernel. Function $K(\bar{X}, \bar{Y})$ measures the similarity between point $\bar{X}$ and $\bar{Y}$. The concept is that the Kernel function may be a dot product between pairs of points in the newly converted area (shown by mapping function $\Phi(\cdot)$)

The SVM algorithm only depends on dot products between patterns $\bar{X}$. Hence, it suffices to know $K(\bar{X}, \bar{Y}) = \Phi(\bar{X}) \cdot \Phi(\bar{Y})$ rather than Φ explicitly which allows us to restate the Support Vector optimization problem.

Several kernels can be chosen such as Linear, Radial Basis, or Polynomial kernels. In this research, the Gaussian Radial Basis Kernel is used as followed

$$K(\bar{X}_i, \bar{X}_j) = e^{-\left|\left|\bar{X}_i - \bar{X}_j\right|\right|^2 / 2\sigma^2} \quad (5)$$

Where $\left|\left|\bar{X}_i - \bar{X}_j\right|\right|^2$ may be recognized as the squared Euclidean distance between the two feature vectors, σ is the standard derivation

After predicting the rotation and translation of the test set, the model accuracy is computed using Root Mean Square Error (RMSE) measurement.

2.4 The Root Mean Square Error (RMSE)

The RMSE has been used as a standard statistical parameter to measure model performance in several natural sciences. The parameter indicates the standard deviation of the residuals or how far the points are from the regression or modelled line as presented in equation 6.

$$RMSE = \sqrt{\tfrac{1}{N}\textstyle\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \quad (6)$$

Where: N is number of samples, x_i is the observed values, $\hat{x}_i$ is the forecasts.

3. EXPERIMENTAL RESULTS

In this paper, we adopt the 2D image Dataset from the CMU Panoptic Studio [1]. The images are taken by different rotation and translation parameters in X, Y and Z axis around the objects. We conduct the model performance evaluation on a variety of challenging scenes such as in the presence of significant occlusion (Circular Movement: Three people rotate around a person at the center (Figure 3(a)) and large displacement (Bat Swing: A person swings a baseball bat. (Figure 3(b)). Sequence frames of Circular Movement were captured for 250 samples and Bat Swing was captured 200 samples for each of 480 cameras. The cameras are extrinsically and intrinsically calibrated and synchronized via an external clock. The dataset contains 209,934 images, which are randomly divided into train data 146,957 images and test data 62,977 images. The ground truth of each image consists of observed rotation (R) and translation (T) parameters defined as extrinsic camera parameters. The total number of R and T ground truth combinations were 480 values.

(a) The examples of postures from Bat Swing Dataset

(b) The examples of postures from Circular Movement Dataset

Figure 3. The examples of Dataset from different challenging scenes.

This research presents the prediction of 3D rotation and translation from 2D images from multiple cameras estimated simultaneously. The deep features of each image are extracted from the second fully connected (fc7); with the number of 4096 features, from the VGG19 and reduced to 1000 features using LSA. Finally, the rotation and translation parameters are estimated with support vector regression (SVR). The prediction performance of the SVR with several adjusted C and gramma parameters is

presented in Table 1, where C is in [0.001,0.01,0.1,1,10,100 and 1000] and the gamma is in [0.0001,0.001,0.01,0.1,1.0,10,100], respectively. From table 1, the best C and gamma parameters are C = 10 and gamma = 1 for rotation estimation with the RMSE of 0.24 degree and C = 1000 and gamma = 1 for translation prediction with the RMSE of 1.35 meters.

Table 1. The prediction accuracy of rotation (R) and translation (T) for fine tune model parameters

C	AVG RMSE of R							AVG RMSE of T						
	gamma							Gamma						
	0.0001	**0.001**	**0.01**	**0.1**	**1**	**10**	**100**	**0.0001**	**0.001**	**0.01**	**0.1**	**1**	**10**	**100**
0.001	0.34	0.34	0.34	0.34	0.34	0.34	0.34	1.78	1.78	1.78	1.78	1.78	1.78	1.78
0.01	0.34	0.34	0.34	0.32	0.28	0.29	0.34	1.78	1.78	1.78	1.78	1.77	1.77	1.78
0.1	0.34	0.34	0.32	0.28	0.26	0.26	0.33	1.78	1.78	1.78	1.77	1.76	1.76	1.78
1	0.34	0.32	0.28	0.27	0.24	0.26	0.32	1.78	1.78	1.77	1.75	1.66	1.67	1.77
10	0.32	0.28	0.27	0.25	0.24	0.26	0.32	1.78	1.77	1.75	1.65	1.55	1.47	1.74
100	0.28	0.27	0.25	0.24	0.24	0.26	0.32	1.77	1.75	1.65	1.59	1.46	1.37	1.66
1000	0.27	0.25	0.25	0.24	0.24	0.26	0.32	1.75	1.65	1.61	1.56	1.35	1.36	1.64

Once we obtain the optimized parameters of C and gamma, they are adopted as SVR model parameters. The model performance evaluation is performed on test data using the proposed model and that of [14] as shown in Table 2. From the experimental results, it can be seen that the performance of the proposed technique in term of the RMSE of rotation and translation is decreased approximately 2.69 degree and 3.11 meters, respectively compared to that of [14].

Table 2. The performance evaluation of the rotation and translation of the proposed with [14]

Method	AVG RMSE of R (Degree)	AVG RMSE of T (Meter)
SfM [14]	2.93	4.47
Propose	0.24	1.36
Performance comparison	2.69	3.11

4. CONCLUSION

This article presents the prediction of 3D rotation and translation from 2D multi-view images. The prediction model has been trained from large amount of 2D image dataset with a variety of challenging scenes. Deep feature extraction is proposed to construct corresponding features among images. The reduced features from LSA are then learned by Support Vector Regression to predict rotation and translation of the images. The model performance comparison is conducted. The experimental results show that the RMSE of rotation and translation prediction of the propose method is 0.24 degree and 1.35 meters, respectively. It can be seen that the average errors in term of RMSE is decreased by 2.69 degree and 3.11 meters respectively compared to the reference. This can demonstrate the significant performance improvement of the proposed algorithm.

5. REFERENCES

[1] Hanbyul J., Hyun S. P., Yaser Sh., "MAP Visibility Estimation for Large-Scale Dynamic 3D Reconstruction," In Proceedings of CVPR, pp. 4321-4328, 2014.

[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, "Building Rome in a Day," Proc. 12th IEEE Int'l Conf. Computer Vision, 2009.

[3] J. - M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik, "Building Rome on a Cloudless Day," Proc. 11th European Conf. Computer Vision, 2010.

[4] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm, "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs," Proc. 10th European Conf. Computer Vision, pp. 427-440, 2008.

[5] N. Snavely, S. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," ACM Trans. Graphics, vol. 25, no. 3, pp. 835-846, 2006.

[6] S. Sinha, D. Steedly, and R. Szeliski, "A Multi - Stage Linear Approach to Structure from Motion," Proc. 11th European Conf. Computer Vision, 2010.

[7] David J. C., Andrew O., Noah S. and Daniel P. H., "SfM with MRFs: Discrete-Continuous Optimization for Large-Scale Structure from Motion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 12, pp. 2841 - 2853, 2013.

[8] V.M. Govindu, "Combining Two - View Constraints for Motion Estimation," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 218-225, 2001.

[9] C. Rother, "Linear Multi-View Reconstruction of Points, Lines, Planes and Cameras Using a Reference Plane," Proc. Ninth IEEE Int'l Conf. Computer Vision, pp. 1210-1217, 2003.

[10] D. Martinec and T. Pajdla, "Robust Rotation and Translation Estimation in Multiview Reconstruction," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007.

[11] K. Sim and R. Hartley, "Recovering Camera Motion Using l1 Minimization," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1230-1237, 2006.

[12] F. Kahl and R. Hartley, "Multiple - View Geometry under the l - Infinity - Norm, " IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 9, pp. 1603-1617, Sept. 2008.*Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= http://doi.acm.org/10.1145/161468.16147.

[13] Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.

[14] Github (2016) at: https://github.com/yihui-he/3D-reconstruction

Camera Pose Estimation using CNN

BHATTARABHORN WATTANACHEEP*
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
nroskool2@gmail.com

ORACHAT CHITSOBHUK
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
orachat.ch@kmitl.ac.th

ABSTRACT

Estimating camera pose is a significant process, which assures the success of the 3D modeling performance. This research presents a camera pose estimation using convolutional neural network (CNN) to transfer learning from pre-trained deep learning VGG19 model in order to extract features from a single image using several datasets captured in indoor and outdoor environments with diverse perspectives and photographic styles. Due to the large dimensions of the extracted features, Latent Semantic Analysis (LSA) are introduced prior to the CNN input. Then, the CNN is trained to predict the camera views and translations. The prediction performance is measured in terms of average mean square errors and compared to the reference techniques. As a result, the regression estimation of the proposed CNN model outperforms the others with average 0.24 degrees rotation error and 0.26 m. translation errors.

CCS CONCEPTS

• **Artificial Intelligence**; • **Computer vision**; • **Image and video acquisition**; • **Camera calibration**;

KEYWORDS

3D Reconstruction, Image Processing, Robotics, Deep Learning

ACM Reference Format:
BHATTARABHORN WATTANACHEEP and ORACHAT CHITSOBHUK. 2020. Camera Pose Estimation using CNN. In *2020 the 3rd International Conference on Control and Computer Vision (ICCCV'20), August 23–25, 2020, Macau, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3425577.3425593

1 INTRODUCTION

Nowadays, 3D modeling techniques are widely adopted in a variety of fields, such as robotics, aircraft, Unmanned Aerial Vehicles (UAVs) [1–3], navigating autonomous vehicles [4], mobile robotics and augmented reality [5], virtual application simulation and various components of large-scale localization, etc. The basic techniques commonly used in finding structures for 3D modeling are the Structure from Motion (SfM) and SLAM [6]. These techniques require feature extraction from two-dimensional images such as finding the angle of the object within the image and approximating geometric structure and camera poses to help estimate 3D models structural parameters. The process of finding image consistency depends on the ability of the feature detection technique to discovery the corresponding feature points, such as the SURF [7], ORB [8] or SIFT [9] methods. For the detection of feature points and image matching, the method requires an iterative comparison of the initial images in order to find rotation and translation within each pair of images. The size of the image dataset therefore affects the prediction time. RANSAC is usually used to aid the camera pose estimation in choosing pairs of feature points. If the pairing found in the first step is correct, the camera pose will be estimated successfully. Consequently, the limitations of feature matching in case of motion, blurring or light variation and the errors resulting from the different order of image matching pairs may lead to the failure in the above step. Moreover, if the estimation of the feature points is insufficient or their positions are errors, this makes it incapable of finding sufficient corresponding features between pairs of images resulted to predict rotation and translation.

Recently, the Convolutional Neural Networks (CNN) have been wildly adopted in related tasks such as image classification [10, 11], pattern recognition, image enhancement, object detection [12, 13], and semantic segmentation [14, 15] due to its high precision. In addition, the use of information transfer principles from these previously trained neural networks to transfer knowledge to other works. Since conventional machine learning and deep learning algorithms, so far, have generally been designed to operate in isolation, these algorithms are being trained to solve different tasks. When the feature-space distribution changes, the models must be reconstructed from scratch. Especially, when considering the context of learning for solving complex problems, most models require a large amount of information to learn. This leads to the difficulty in labeling the answers for each image in such a large training dataset. For example, ImageNet's dataset training requires more than a million images divided into different categories. Transfer learning is the concept to resolve the isolated learning model and derive the knowledge obtained for one task to solve relevant one. Therefore, a research, which transfers learning from CNN, starts with learning the desired job and modifies CNNs to estimate the camera position from the input image and extract reasonable features that are robust against motion blur and illumination for localization problems. PoseNet research [16] reveals that it is inappropriate to estimate camera poses from the high dimensional output of FC layers, since they cannot provide the best results due to the overfit problem from the PoseNet training data. From the above-mentioned PoseNet problem, subsequent research [17] introduced a modification of the PoseNet architecture using the GoogleNet architecture [12], deep learning network in which the softmax layer (for classification) was replaced by the FC layer. By [17] removing features of the FC layer,

*Place the footnote text for the author (if applicable) here.

ICCCV'20, August 23–25, 2020, Macau, China

ACM ISBN 978-1-4503-8802-3/20/08…$15.00
https://doi.org/10.1145/3425577.3425593

the authors then reshapes the feature vector into a 32 x 64 matrix for LSTM gesture estimation to minimize image encoding dimensions. The high dimensions of image encoding compared to a relatively small number of training examples may lead to overfitting since the last FC regression layer must be capable of learning regression issues with a variety of independent degrees of problems including the localization error (translation and rotation) and generalization of unknown scenes, which has not been discussed in the original study of [18].

Recent work has shown the consideration of camera pose estimation without relying on video inputs but using single images [19]. The idea implemented using transferred features from deep learning model from both indoor [21] and outdoor datasets [22] and submitting these features to the support vector regression (SVR) for camera pose estimation provides less rotation and translation errors. However, for such a large dataset, it takes quite large amount of training time since the SVR is a regression method to preserve all the key features that define the functionality of the algorithm. In other words, the SVR attempts to ensure that the error in the estimation falls within a specified threshold. Hyperplane is utilized for predicting the results from multidimensional inputs. Searching for a hyperplane in a multidimensional space would increase the cost. Moreover, another difficulty is a variation of magnitudes, units, and range in real-world datasets. There are 7 camera pose parameters: 4 for rotation and 3 for translation are required; however, only one can be predicted at a time resulted in a large cost.

Building deep learning architecture from scratch requires a large amount of practice datasets and plenty of time to produce an efficient result. Transferring circumstances or things observed in one setting can help to improve the overall implications of another setting, resulting in less effort to prepare excessive amount of new practical datasets and processing time. Consequently, our research applies the transfer of single image features from in-depth learning of the VGG19 model to a proposed CNN in order to predict camera pose parameters in the form of rotation and translation. In the first phase, the local pattern in the input will be learnt through parametrizing each trained filter in the convolutional layers of the CNN. In other words, CNN is seeking to find the most appropriate way to predict the best result. In our study, for camera pose estimation, a performance evaluation of several architectures is conducted using 3 datasets where the first and second datasets are used to compare with [19], and the third one is used to compare with architectures in [18].
data.

2 PROPOSE ALGOLITHM

The next subsections provide instructions on how to insert figures, tables, and equations in your document.

2.1 Tables

This research presents camera pose estimation using CNN, which transfers knowledge from pretrained VGG19 model [23]. Deep transferred features from a single image are adopted to learn the relationship of each camera pose in terms of a 7-dimensional vector, where $\hat{t}$ is a 3D translation vector and $\hat{r}$ is a 4-dimensional rotation vector (quaternion).

Most of the traditional learning trained on small dataset and used in the specific task might not be as successful as expected especially for the unseen data since not enough retrained knowledge can be passed from one model to another. Transfer learning should enable us to utilize knowledge from previously learned tasks to newer, related ones. In the case of computer vision issues, other features of low quality such as edges, shapes, corners and intensity, can be transmitted across tasks, thereby enabling the transfer of knowledge across jobs. Therefore, as we have mentioned in the previous, information from an existing task serves as an additional input when learning a new goal.

To perform the transfer learning, many researchers have chosen VGG19 for their tasks. In [27, 28], Gatys et al. used the VGG19 through training approaches on object recognition for texture synthesis. Li et al. [29, 30] applied the VGG19 trained on ImageNet to extract hierarchical convolutional features for visual object tracking. Long et al. [31] have used VGG19 transfer learning to diagnose faults in the manufacturing industry. To achieve the characteristics of the desired image, in this research, we integrate transfer learning from the pre-trained VGG19 on the enormous ImageNet dataset used for 1000 category classification.

In this research, we demonstrate the feasibility of estimating the relationship between cameras via VGG19 transfer feature. The network begins with passing a 224 x 224 input image (RGB) through the stack of convolutional (conv.) and pooling layer to reduce the input's spatial dimensions (width x height) to a smaller size (downsampling). Nonlinear ReLU activation layers [38] are chosen to improve efficiency. Then, the three Full-Connected (FC) layers are trained to classify ImageNet Large Scale Visual Recognition Challenge (ILSVRC) into 1000 classes followed by the final softmax. Since our work does not require classification of data, the 4096 features dropped out from the fc7 layer of VGG19 are delivered to our camera pose estimation system. However, the resulting features have such a large dimension, which may cause data to be fragmented (sparse) and lead to the difficulty, time-consuming and inefficiency in the data analysis known as the curse of dimensionality [24]. The problem can be minimized by reducing the dimension of the data using the Latent Semantic Analysis (LSA) [25]. We obtain a 1000-dimensional feature vector as a result. Root mean square error (RMSE) [26] is chosen as our performance measurement matric to determine the rotation and translation prediction error. The overall workflow of our proposed camera pose estimation system is presented in Figure 1.

In this research, we have developed a CNN model to create the appropriate features and to estimate the camera's relationship parameters for 3D reconstruction. The LSA reduced features are processed through our 4 convolutional layers and max pooling to simplify the network. The drop out layer is adopted to prevent overfit. Then, the dense layers are trained to support camera pose regression. The total of seven parameters are predicted for the rotation and translation of the cameras using two deep learning models, the rotation estimation model (four answers), and the translation estimation model (three answers).

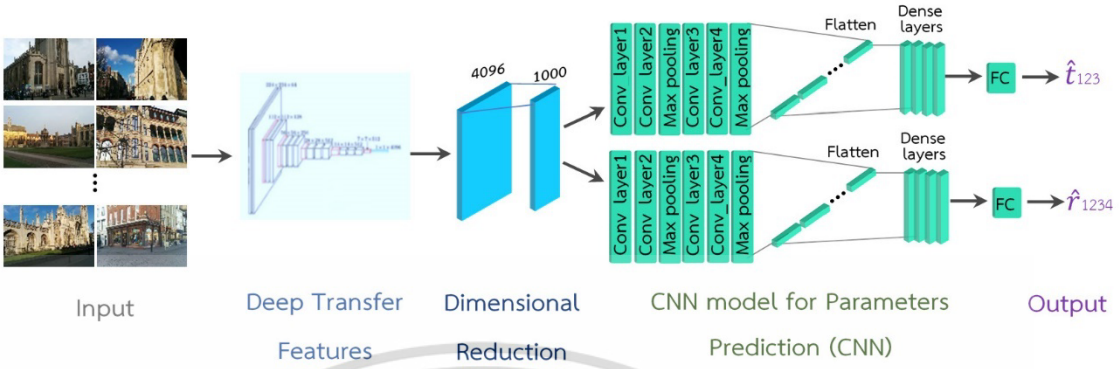


Figure 1: Overview of system operating process

Table 1: The average RMSE of our proposed algorithm compared with [19].

Train Data	Test Data	VGG+SVR [19]		Proposed (VGG+CNN)	
		Rotation (°)	Translation (m)	Rotation (°)	Translation (m)
Dome	Dome	0.45	2.45	0.07	1.15
Map		0.35	1.29	0.27	1.25
Map	Map	0.14	0.0035	0.03	0.0023
Dome		0.74	4.49	0.37	2.59
Average RMSE		0.44	2.25	0.19	1.25

3 EXPERIMENTS

In our experiments, the test dataset from 3 sources are used. The first one is indoor dome-shaped photography from CMU Panoptic Studio [21]. The image was taken by shooting around the object, with variety of rotation and translation. There are 209,934 images in dataset, divided into 146,957 training images and 62,977 test images. Each image comes with the extrinsic camera parameters; a set of 480 possible patterns of different individuals or groups movements such as moving, walking around and baseball swing etc. The second source is aerial photography [22] (Outdoor) taken from a camera attached to the drone which flew above 4 different locations - namely nadir square, stadthaus, rathaus, obelisk oblique square. Large translation will take place on X and Y axis, while Z axis is slightly different. The final source (Outdoor: Cambridge dataset) [32] consists of 6 sub-datasets, GreatCourt, KingsCollege, OldHospital, ShopFacade, StMarysChurch and Street. It is the popular dataset used to measure the efficiency of the algorithms.

The CNN model structure parameters and optimizer are evaluated in terms of the appropriate number of filter nodes {64, 128}, type of activation function [tanh, linear] and optimizer [adam and adadelta], batch size {32, 50, 100}, and the number of epochs {400, 500, 600} for achieving the optimum model structure with least RMSE value. The parameters that provide the lowest rotation (R) and translation (T) errors are considered to be the applied parameters for indoor (dome) and outdoor (Map and Cambridge) datasets. From experimental results, the best parameters obtained are 500 epochs, 32 batch size with adam optimizer and filter nodes of R {128,128,128,64}, T {64,64,64,128} for the indoor and and R {128,128,64,64}, T {128,128,128,64} for outdoor datasets, respectively. From the previously defined CNN structure, the experimental results of camera pose estimations of dataset 1 (Dome) and 2 (Map) are compared with the research results [19] as shown in Table 1

In order to assess the efficiency of the proposed model compared to [19], RMSEs are measured using the cases of dataset with the same view or shooting style as the training set while shooting different view or configuration as the test set. It is shown that the average RMSE of the rotation and translation of the proposed method training from dataset 1 (Indoor) and testing with the dataset 1 and 2 (outdoor), are 0.07, 0.27 degrees of rotation and 1.15, 1.25 meters of translation, respectively. Additionally, the average RMSE of rotation and translation is 0.03, 0.37 degrees and 0.0023, 2.59 meters respectively for the training with dataset 2 thus testing with the dataset of set 1 and set 2, which outperforms the methods[19] by 0.26 degrees and 1 meter, respectively. The experimental results show that the proposed model offers better average RMSE when training with dataset 1 than dataset 2 by 0.2 degrees of rotation and 0.1 meters of translation. This is due to the different amount of dataset 2, which is 100 times smaller than that of dataset 1 and the different in the shooting style of dataset 2 less variation than the dataset 1, resulting in insufficient cross-learning of the different

Table 2: Median translation (in meters) and rotation (in degrees) errors of different deep absolutes pose estimators, when tested on the Cambridge dataset.

Algorithm	GreatCourt	KingsCollege	OldHospital	ShopFacade	StMarysChurch	Street
PoseNet [16]	NA	1.97m, 5.40°	2.31m, 5.38°	1.46m, 8.08°	2.65m, 8.48°	3.67m, 6.50°
Dense PoseNet [16]	NA	1.66m, 4.86°	2.57m, 5.14°	1.41m, 7.18°	2.45m, 7.96°	2.96m, 6.00°
Bayesian PoseNet [32]	NA	1.74m, 4.06°	2.57m, 5.14°	1.25m, 7.54°	2.11m, 8.38°	2.14m, 4.96°
LSTM-Pose [20]	NA	0.99m, 3.65°	1.51m, 4.29°	1.18m, 7.44°	1.52m, 6.68°	NA
SVS-Pose [33]	NA	1.06m, 2.81°	1.50m, 4.03°	0.63m, 5.73°	2.11m, 8.11°	NA
PoseNet + Reprojection error pose loss [34]	7.00m, 3.7°	0.99m, 1.1°	2.17m, 2.9°	1.05m, 4.0°	1.49m, 3.40°	20.7m, 25.7°
VLocNet [35]	NA	0.836m, 1.42°	1.07m, 2.41°	0.593m, 3.53°	0.631m, 3.91°	NA
DSAC [36]	2.80m, 1.5°	0.30m, 0.5°	0.33m, 0.6°	0.09m, 0.40°	0.55m, 16°	NA
LearnLess(DSAC++) [37]	0.4m, 0.2°	0.18m, 0.3°	0.20m, 0.3°	0.06m, 0.30°	0.13m, 0.4°	NA
Active Search	NA	0.42m, 0.6°	0.44m, 1.0°	0.12m, 0.40°	0.19m, 0.5°	0.85m, 0.8°
Proposed	0.16m, 0.18°	0.20m, 0.21°	0.22m, 0.48°	0.11m, 0.20°	0.10m, 0.39°	0.77m, 0.76°

shooting. The third dataset is a comparison of the median rotation and translation errors with research [18]. Each study was trained and tested using its own dataset as shown in Table 2

Table 2 shows the implementation of the Cambridge dataset, which is popular for assessment the efficiency of the camera pose estimation algorithm. This dataset was taken with the angles of elevation that are the same plane as the location, which shooting style different from that of dataset 1 (indoor with the angles of press, parallel and elevation to the object) and dataset 2 (outdoor with the angles of press from drone flying parallel to the location). There are 6 subsets, subset 1 to 5 are locations and the last subset is the street. The last one is the most difficult combination of images since the whole image contains similar color and texture. In table 2, the results for the first 10 methods are based on [18], where LearnLess (DSAC++)[37] provides the least median error for subset 1 to 5. However, in [37], there is no result reported for the last subset. Nevertheless, it can be seen that the proposed method provides less median rotational and translation errors than those of the other methods. The sum of median error obtained from our proposed algorithm is less than [37] by 0.04 degrees rotation and 0.18 meters translation for subset 1 to 5 and less than the Active Search (SfM) by 0.04 degrees and 0.08 metres, respectively. From Tables 1 and 2, it is obvious that the our algorithm offers the best results in finding the rotation and translation in the shooting style similar to the dataset 2 with less average RMSE of 0.03 and 0.0023 degrees rotation compared with dataset 1 and 3, respectively.

4 CONCLUSION

This research presents the regression of camera pose estimation using the convolution neural network by transferring the basic features of VGG19 through LSA dimensional reduction technique prior to the input of the proposed CNN. In this research, the estimation of rotation and translation is derived from separated CNN models. According to the performance evaluation with a variety of dataset such as a different views, photography, brightness, blur and scrolling, the proposed model provided less overall predictive error than the other methods. The best performance is obtained from the dataset using the drone shooting down over the object. Nevertheless, even though the proposed model offers the least median error than the other methods for the Street subset, it provides the lowest performance compared to other subsets due to the ambiguity in color and texture of the scene.

REFERENCES

[1] Forster, C., Pizzoli, M., and Scaramuzza, D. 2014. SVO: Fast semi-direct monocular visual odometry. In: Intl. Conf. on Robotics and Automation (ICRA).

[2] Engel, J., Sturm, J., and Cremers, D. 2012. Camera-based navigation of a low-cost quadrocopter. In: Intl. Conf. on Intelligent Robot Systems (IROS).

[3] Achtelik, M., Weiss, S., and Siegwart, R. 2011. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In: Intl. Conf. on Robotics and Automation (ICRA).

[4] Lim H., Sinha S. N., Cohen M. F., and Uyttendaele, M. 2012. Realtime image-based 6-dof localization in large-scale environments. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Lynen S., Sattler T., Bosse M., Hesch J., Pollefeys M., and Siegwart R.. 2015. Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization. In Robotics: Science and Systems (RSS).

[6] Hartley R. I. and Zisserman A. 2004. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition.

[7] Bay H., Ess A., Tuytelaars T., and Van G. L. 2008. Speeded-up robust features (surf). Computer vision and image understanding, vol. 110, no. 3, pp. 346–359.

[8] Rublee E., Rabaud V., Konolige K., and Bradski, G. 2011, Orb: An efficient alternative to sift or surf. in Computer Vision, international conference on. IEEE, pp. 2564–2571.

[9] Mortensen, E. N., Deng, H., and Shapiro, L. 2005. A sift descriptor with global context. in Computer vision and pattern recognition, CVPR. IEEE computer society conference on, vol. 1. IEEE, pp. 184–190.

[10] Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A. and van der Maaten, L., 2018. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 181-196).

[11] Tan, M. and Le, Q.V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946.

[12] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[13] Westlake, N., Cai, H. and Hall, P., 2016, October. Detecting people in artwork with CNNs. In European Conference on Computer Vision (pp. 825-841). Springer, Cham.

[14] Zhu, Y., Sapra, K., Reda, F.A., Shih, K.J., Newsam, S., Tao, A. and Catanzaro, B., 2019. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8856-8865).

[15] Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), pp.2481-2495.

[16] Kendall, A., Grimes, M., and Cipolla, R. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In IEEE International Conference on Computer Vision (ICCV).

[17] Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S. and Cremers, D., 2017. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 627-637).

[18] Shavit, Y., and Ferens, R. 2019. Introduction to Camera Pose Estimation with Deep Learning., arXiv preprint arXiv:1907.05272.

[19] Wattanacheep, Bh. ,and Chitsobhuk, O. 2019. Prediction of 3D rotation and translation from 2D images ,In ICCCM 2019, July 27–29, 2019, Bangkok, Thailand. https://drive.google.com/file/d/1mE23Eg2x_4dHp7lKfQZNKhJoeXQXH1Be/view.

[20] Bell, S., Zitnick, C. L., Bala, K., and Girshick, R. 2016. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[21] Hanbyul J., Hyun S. P., Yaser Sh. 2014. MAP Visibility Estimation for Large-Scale Dynamic 3D Reconstruction. In Proceedings of CVPR, pp. 4321-4328.

[22] Nex, F., Gerke, M., Remondino, F., Przybilla H.-J., Bäumker, M., and Zurhorst, A., 2015. ISPRS Benchmark for Multi-Platform Photogrammetry. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-3/W4, pp.135-142.

[23] Karen, S. ,and Andrew, Z. 2015. Very Deep Convolutional Networks For large-scale Image Recognition. ICLR.

[24] Venkat, Naveen. 2018. The Curse of Dimensionality. Inside Out. 10.13140/RG.2.2.29631.36006.

[25] Landauer, T., *et al.* 2007. Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates.

[26] Alex, J. S, and Bernhard, S. 2004. A tutorial on support vector regression. Statistics and Computing 14, pp. 199–222.

[27] Dipanjan, S. 2018. A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning.

[28] Gatys, Leon, Alexander, S. E., and Matthias, B. 2015. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems, pp. 262-270.

[29] Bruna, Joan, Pablo, S., and Yann, L. 2015. Super-resolution with deep convolutional sufficient statistics. arXiv preprint arXiv:1511.05666.

[30] Li, Y., Yafei, Z., Yulong, X., Jiabao W., and Zhuang M. 2016. Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features. IEEE Signal Processing Letters 23, no. 8 : 1136-1140.

[31] Wen L., Li X., Li X., and Gao L. 2019. A New Transfer Learning Based on VGG-19 Network for Fault Diagnosis. 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD), Porto, Portugal, pp. 205-209.

[32] Kendall, A., and Cipolla, R. 2015. Modelling uncertainty in deep learning for camera relocalization. arXiv preprint arXiv:1509.05909.

[33] Naseer, T. and Burgard, W. 2017. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1525-1530). IEEE.

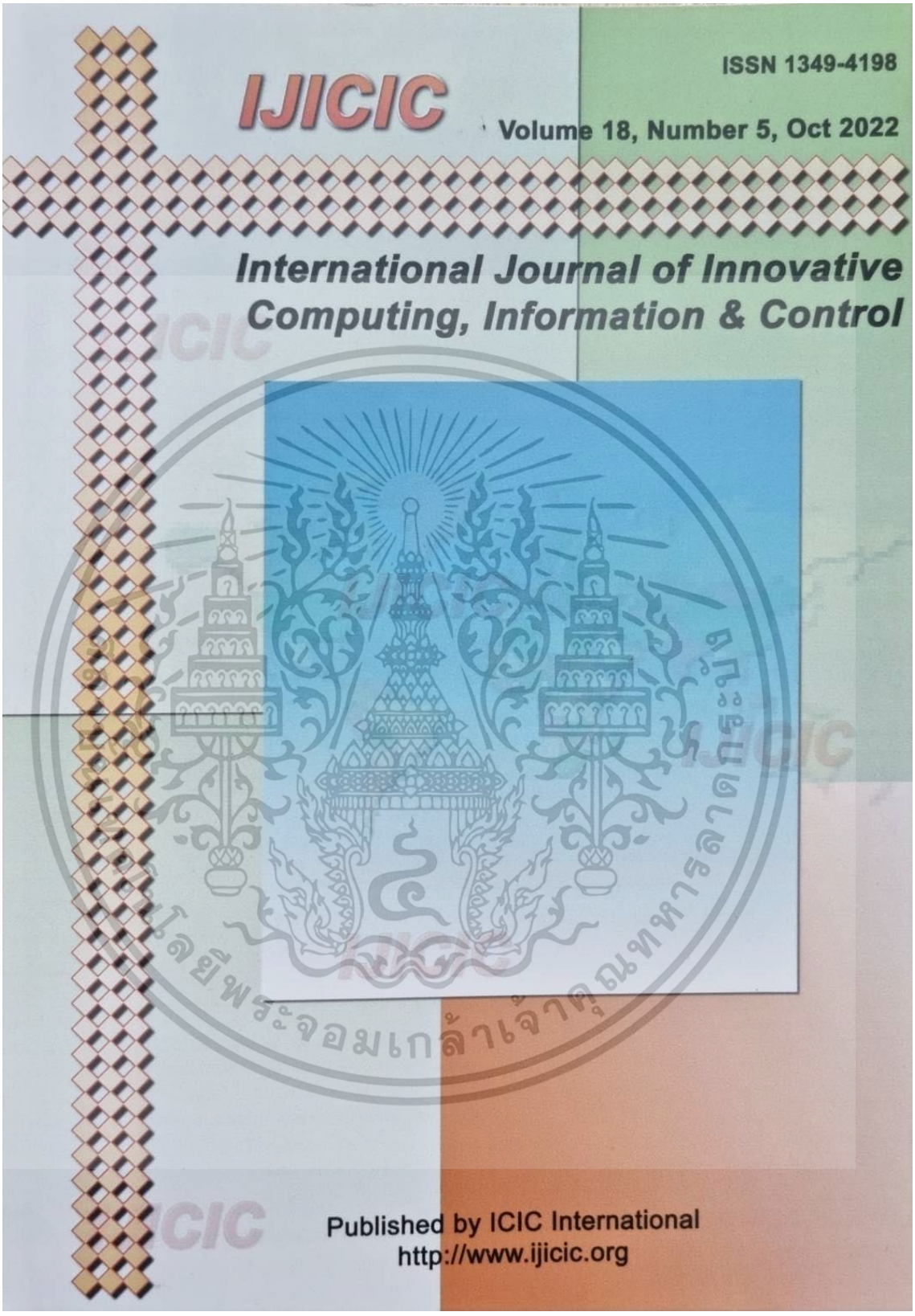
[34] Kendall, A. and Cipolla, R. 2017. Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5974-5983).

[35] Valada, A., Radwan, N., and Burgard, W., 2018. Deep auxiliary learning for visual localization and odometry. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6939-6946). IEEE.

[36] Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., and Rother, C., 2017. DSAC-differentiable RANSAC for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6684-6692). https://github.com/cvlab-dresden/DSAC.

[37] Brachmann, E., and Rother, C., 2018. Learning less is more-6d camera localization via 3d surface regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4654-4662). https://github.com/vislearn/LessMore.

[38] Krizhevsky, A., Sutskever, I., and Hinton G. E. 2012. Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1097–1105.



International Journal of Innovative Computing, Information and Control
Volume **18**, Number **5(tentative)**, October **2022**

ICIC International ©2022 ISSN 1349-4198
pp. 1–2204-006

A CNN-BASED MULTI-MODEL ENSEMBLE METHOD FOR INDOOR AND OUTDOOR MULTI-VIEW STEREO RECONSTRUCTION

Bhattarabhorn Wattanacheep and Orachat Chitsobhuk

School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand
{ bhattarabhorn.wa; orachat.ch }@kmitl.ac.th

Received April 2022; revised July 2022

Abstract. *Camera poses estimation is a critical process that ensures the success of Three-Dimensional (3D) modelling. We present a Convolutional Neural Network (CNN)-based multi-model ensemble method for indoor and outdoor multi-view stereo reconstruction capable of learning across multiple domains, including images from both indoor and outdoor environments. Each domain's images have distinct properties and shooting viewpoints, which leads to difficulty in efficient learning such a large difference and requires large amount of computational resources. In order to reduce complexity of the end-to-end single model, the proposed model is divided into multiple learning agents consisting of domain-specific agents and domain relationship agent. The domain-specific agent is trained independently on its own set of unique image characteristics, for example, one for indoor datasets and another for outdoor datasets. The domain relationship agent then ensembles and analyzes the multiple domain features and finalizes the estimation. In terms of average root mean square error, we compare the performance of the combined domain single model with the suggested ensemble CNN model. The experimental results indicate that the proposed model outperforms the others, with rotation and translation prediction errors of 0.112012266.*

Keywords: 3D reconstruction, Convolutional neural network, Deep learning, Transfer learning, Ensemble CNN

1. **Introduction.** Nowadays, 3D modeling techniques are widely adopted in a variety of fields, for example, robotics, aircraft, Unmanned Aerial Vehicles (UAVs) [1-3], navigating autonomous vehicles [4], mobile robotics and augmented reality [5], virtual application skeleton-based action recognition [6] simulation and various components of large-scale localization. The basic techniques commonly used in finding structures for 3D modeling are Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) [7]. These techniques require feature extraction from two-dimensional images, for example, determining the angle of the object within the image and approximating geometric structure and camera poses to help estimate 3D model structural parameters. The ability of a feature detector such as Speeded Up Robust Features (SURF) [8], Oriented FAST and Rotated BRIEF (ORB) [9] or SIFT [10] methods used to discover corresponding feature points is critical. To detect feature points and match images, initial images are iteratively compared to find rotations and translations, within each image pair. The prediction time is proportional to the image size. Typically, Random Sample Consensus (RANSAC) is employed to assist the camera pose estimation algorithm in selecting feature point pairs. If the pairing discovered in the first stage is accurate, the camera pose will be estimated correctly. As a result, when the predetermined number of matches is satisfied, prioritized

DOI: 10.24507/ijicic.18.05.xxx

search strategies [11-13] will complete the procedure. Situations such as motion blurring, light variation, and errors from different image order may all lead to feature matching failure. Moreover, if the estimation of the feature points is insufficient or their positions are errors, this makes it incapable of finding sufficient corresponding features between pairs of images, resulting in degraded rotation and translation prediction.

3D objects are widely used in computer vision applications ranging from human-machine interactions to autonomous vehicles and robotics [14]. Deep Learning (DL) has achieved impressive success in 2D fields [15-17,20,39] with various applications such as face recognition and image classification [18,19,41], pattern recognition, image enhancement, object detection [20,21,54], and semantic segmentation [22,23]. Since everything we perceive in the real world is in 3D space, 3D data can help improve the performance of computer vision-based applications [24].

In the recent years, several 3D databases have been made available to the public [25-27]. These advancements have enabled computer vision researchers to work with real-world objects, and DL-based 3D shape analysis research, including 3D classification, segmentation, retrieval, and reconstruction [55]. However, unlike the regular sampled 2D images, 3D shapes are irregular triangle meshes or point clouds; it is a challenging task for DL to extract distinguishing features [28] that can characterize the shapes and parts of a 3D object. In addition, knowledge from previously trained networks can be transferred to train on new problems. Conventional machine learning and deep learning algorithms, so far, have generally been designed to operate on solving specific tasks. When the feature space changes, the models must be rebuilt from scratch. Most models require a large amount of information to learn, especially, when solving complex problems. This leads to difficulty in labeling the ground truth for each image in these large training datasets. ImageNet's dataset, for example, consists of over a million images divided into several categories. Transfer learning attempts to derive the based learning model and apply the knowledge obtained for one task to solving a relevant one.

Transfer learning research begins with learning a target job and then modifies CNNs to predict camera poses from input images and extract valuable features that are robust to motion blur and illumination for localization problems. Kendall et al. [29] transferred learning from PoseNet and demonstrated that it was ineffective to predict camera poses using the high-dimensional output of fully connected layers, as they cannot produce the best results due to overfitting with the PoseNet training data.

[30] proposed a sorting algorithm that took advantage of scene semantics to create consistency between indoor and outdoor models. The research detected building windows and used as a key in reconstructing the three-dimension scenes since they were visible both inside and outside. The detected windows were then classified as indoor or outdoor using semantic classifiers and imported to Patch-based Multi-View Stereo (PM-VS) [31]. The results were compared to those of SfMs and they illustrated the efficiency of PMVS even in the case of noisy windows and misaligned indoor and outdoor position.

3D reassembly is an innovative and practical application which integrates indoor and outdoor 3D reconstruction algorithms into a single application. However, dense geometry of window detection is necessary since it affects the assembling of the 3D images. In [32], the authors adopted deep learning technology to automatically learn specific area patterns from a single input image. The dimension of the features was reduced after transferred learning from fully connected layer of VGG19. Finally, a regression estimation method was employed. Based on the Support Vector Regression (SVR) principles, this resulted in rotation and translation estimates with lower prediction error and independence from object geometry. The SVR's kernel functions were used to transform the original dataset

(linear/nonlinear) into higher dimensional space so that the data became linearly separable and make hyperplane decision boundary among classes. The larger the dimensional space, the longer the processing time. This resulted in a very long prediction time when dealing with a huge dataset. Furthermore, testing using pictures taken with various imaging characteristics (for example, learning with direct shooting to the objects but testing with parallel shooting to them) may lead to unsatisfactory results.

Later, there was a study on transfer learning using the neural network instead of SVR to learn on different image characteristics especially with shooting point of views [33]. Pooling was adopted to reduce the number of features and the neural network parameters were adjusted during training time on multiple epochs until reaching the minimum prediction errors. Parallel processes from neuron nodes of one layer to another layer allowed faster decision and offered higher accuracy.

Nevertheless, it was quite a complicated task to understand a wide range of image characteristics. If the model was previously trained on certain picture attributes, it will need to be retrained in order to learn about new aspects. Adjusting the model and dealing with all of the data volatility would be challenging. In this case, the model must be trained on substantial samples of all possible aspects from such as both indoor and outdoor datasets to recognize all desired cases. Another difficulty would be dealing with imbalanced datasets since we could not get sufficient looks of the underlying classes, resulting in poor accuracy for classes with less observation. Even if a deep learning model is fine-tuned to surpass the competition, the model may still have flaws in certain situations. Consequently, it is reasonable to assume that a strategy that makes use of a variety of deep learning techniques will deliver superior performance. Several researches have been published in the literature to this objective, with the goal of increasing the accuracy of prediction by integrating models into one another. As a result, there is a study that presents the concepts of ensembling [50,51,53], which is the process of integrating various learning algorithms in order to gain their collective performance, i.e., to improve the performance of existing models by combining several models into a single reliable model. Models are stacked together to improve their performance and obtain a single final prediction to ensure the most stable and accurate prediction possible. To reduce training complexity while increasing model accuracy, we proposed a multi-domain model in which each sub-model was trained independently on its own set of unique image characteristics, such as one for indoor and one for outdoor datasets, followed by an aggregate model to refine the final solution. The suggested model's hierarchical structure enabled it to estimate rotation and translation more accurately for indoor and outdoor camera pose estimation.

We demonstrated that estimation for rotation and translation can be calculated concurrently for all images without the need for an initial image for iteratively computing pairwise estimation, resulting in faster predictions and the ability to understand the image characteristics well regardless of the change in brightness, thereby alleviating the constraint associated with shooting methods. This led to a better understanding of more versatile and efficient shooting. Additionally, our study was shown to be applicable to images with less dense geometry and a wide variety of challenging datasets. We illustrated the effectiveness of our approach compared to the SIFT-based methodology especially for handling large texture region and repeating structure on the same datasets [32,33].

The remainder of this paper is organized as follows. Section 2 describes framework for 3D reconstruction parameter estimation for combined domain single model and the proposed ensemble of CNN models for indoor and outdoor multi-view stereo reconstruction. The simulation experiments are discussed in Section 3. Finally, Section 4 concludes this paper.

2. **Methodology.** In this research, we proposed an estimation of rotation and translation parameters of 2D pictures used to reconstruct 3D images based on transfer learning from a CNN model. However, training a CNN for multiple domains containing a variety of image attributes using a single model is a challenging task since the final model may have a deep and complicated architecture, which not only requires high computational cost but also can damage the accuracy and validation of the model [34]. Multiple domains can contain significantly different aspects of the shot, and the perspective and factors related to image acquisition resulted in a need for a large training dataset in order to cope with such a wide range of local characteristics. Moreover, introducing a new domain data requires the model to be retrained from scratch resulting in a lengthy development time. As a result, we introduced the CNN-based multi-model ensemble method for indoor and outdoor multi-view stereo reconstruction. We divided our agents into domain-specific agents and domain relationship agents to offer flexibility in learning on multiple domains of picture data to evaluate and estimate the required parameters. The efficacy of the based-line end-to-end single model and the recommended ensemble CNN model were explored and compared for indoor and outdoor image domains with highly diverse image acquisition formats and image properties.

2.1. **Proposed framework for 3D reconstruction parameter estimation.** The suggested system was divided into two parts: the first part is a preprocessing data section, which extracted significant properties from the image and sent them to the second part (ensemble CNN model), for learning and estimating by the learning agent network model.

For the preprocessing process, we transferred knowledge from the pre-trained CNN model of the prototype VGG19 trained with ImageNet [35] dataset and extracted the features from the fully connected layer 7 with feature dimension of 4096. Since our work did not require classification of data, 4096 features were dropped from the fully connected layer 7 of VGG19, which was delivered to our camera pose estimation system. However, the resulting features had a very high dimension, which might lead to data fragmentation (sparsity) and made data analysis more complex, slow, and inefficient. Therefore, it was necessary to reduce feature dimension that might not be highly relevant information for prediction. The problem was minimized by reducing the data dimension using Latent Semantic Analysis (LSA) [52], which was based on a mathematical technique called Singular Value Decomposition (SVD) [36], reducing feature dimension while maximizing the signal energy into as few coefficients as possible. The compact features were then sent to the ensemble model to estimate the camera pose parameters.

In this study, we compared the performance between the proposed ensemble CNN model (see in Figure 2) and a single model trained with multiple domains (see in Figure 1). The preprocessing of the data was handled in exactly the same way by both models; the only difference was in the design of the deep learning model.

2.1.1. *Combined domain single model.* Figure 1 showed a combined domain single model, which consisted of two single CNN models, one for estimating rotation parameters ($\hat{r}_{1234}$) and the other for estimating translation parameters ($\hat{t}_{123}$), which were used to learn transferred features from indoor and outdoor data. Each CNN contained four convolutional layers to extract spatial characteristics. There were max pooling layers after the second and fourth layers of each CNN followed by flatten layers to approximate the required parameters.

2.1.2. *Ensemble CNN model (A CNN-based multi-model ensemble method for indoor and outdoor multi-view stereo reconstruction).* The ensemble CNN model was separated into

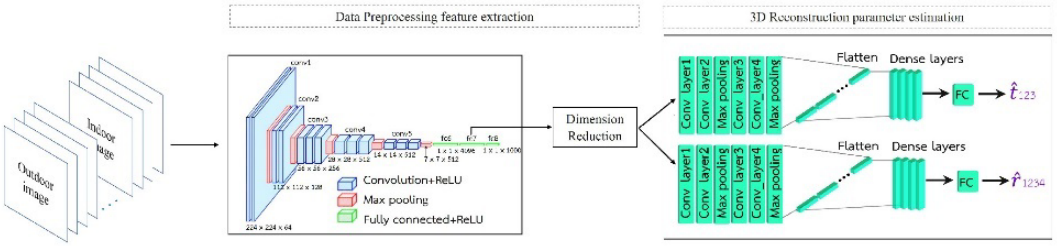


FIGURE 1. Combined domain single model

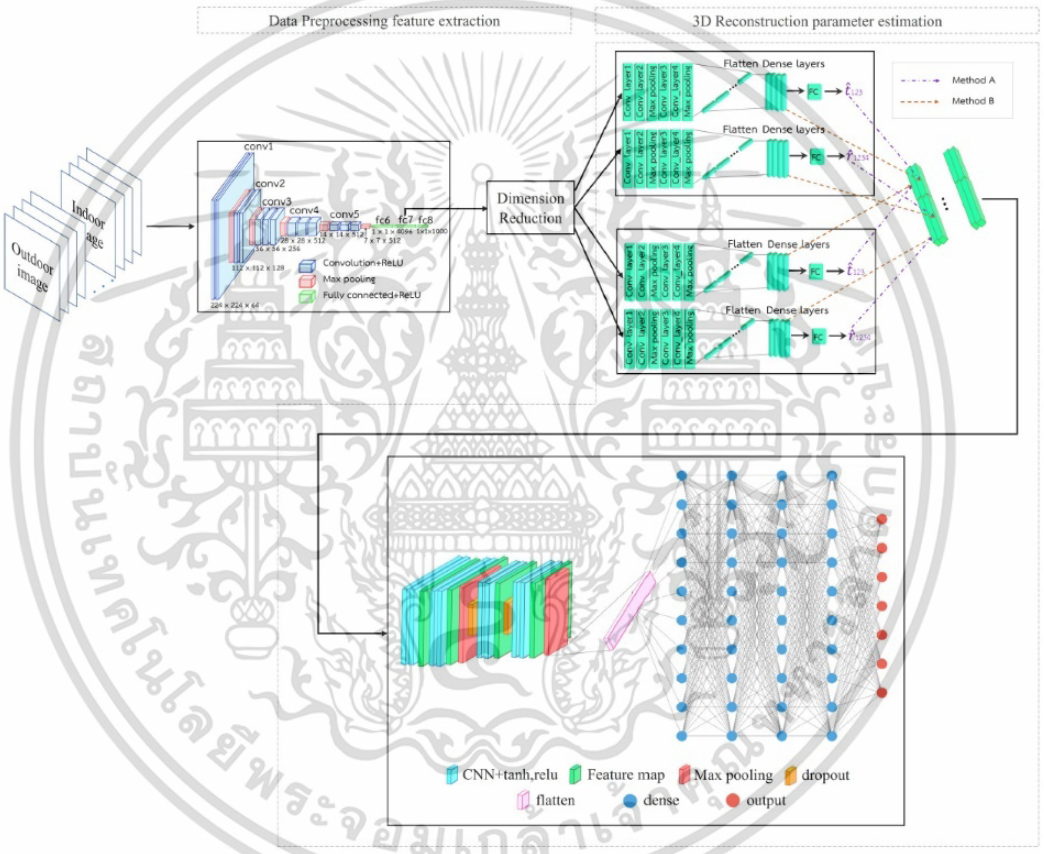


FIGURE 2. A CNN-based multi-model ensemble method for indoor and outdoor multi-view stereo reconstruction

domain-specific agents and domain relationship agents to enable the flexibility in learning a variety of multiple domain image dataset.

For a given dataset, a single algorithm may not be able to make the perfect prediction. Machine learning algorithms have limitations and achieving a high-accuracy model is challenging. The overall accuracy of the model could be improved if we build and combine multiple models. The combination can be implemented by combining the outputs from each model with two goals in mind: minimizing model error while maintaining generalization.

Domain-specific agents had the same structure as a single model's architecture but were trained the model with domain-specific data. Two learning agents were used to test the prototype: one was an indoor leaning agent, and the other was an outdoor learning agent. The parameter estimation results from all the domain-specific learning agents were sent to the domain relationship agent, which was used to assess the relationship between domains and predict the final rotation and translation parameters. We had investigated two methods of the feature arrangement from each domain-specific agent as illustrated in Figure 2 Method A (dash-dotted line) extracting features of the output layer of each domain-specific agent and arranged them into a structure of 7×2 features (4 rotation ($\hat{r}_{1234}$) and 3 translation parameters ($\hat{t}_{123}$)) while Method B (dashed line) combine features from the layer before the output layer of domain-specific agent and constructed 128×2 features (64 rotation and 64 translation features) from indoor and outdoor domain-specific agents.

The domain relationship agent consisted of a 2D CNN with a set of 2 convolutional layers and 2 max pooling layers followed by dense layers to derive inferences of regression prediction of the final camera pose parameters.

2.2. **Model performance measurement.** After predicting the rotation and translation of the test set, the model accuracy is computed using Root Mean Square Error (RMSE) [56] measurement.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{1}$$

where N = number of samples; x_i = observed values; $\hat{x}_i$ = predictions.

3. **Experimental Results.** In this research, datasets from three sources were used. The first one was Dome dataset, indoor photography from the Dome in the CMU Panoptic Studio [37]. The 209,934 images were taken by shooting around the object, with a variety of rotation and translations. The dataset was divided into 146,957 training and 62,977 test images. Each image came with the extrinsic camera parameters; a set of 480 possible patterns of different individual or group movements, for example, moving, walking around and baseball swings, as shown in Figure 3. The second source was Map dataset, outdoor aerial photography [38] taken from a drone camera, which flew above four different locations – nadir square, Stadthaus, Rathaus, and obelisk oblique square. There were large translations along the X and Y axes, while the Z-axis varied slightly. It comprised of 1500 training images and 578 testing images, as illustrated in Figure 4. The third source was the outdoor Cambridge dataset [29] which had six sub-datasets: GreatCourt, KingsCollege, OldHospital, ShopFacade, StMarysChurch and Street, as show in Figure 5. GreatCourt contains 1532 training images and 760 testing images. KingsCollege has the largest spatial extent of 5000 m^2 amongst all the datasets. It consists of 1220 training images and 346 testing images. OldHospital has a spatial extent of 2000 m^2, and contains 895 training and 182 testing images. ShopFacade contains 230 training images and 103 testing images. StMarysChurch contains 1487 training images and 530 testing images and Street contains 3015 training images and 2923 testing images. For the outdoor Cambridge dataset, we only used the test set to predict rotation and translation using in Section 2.1.2 Method B for benchmarks.

A variety of criteria were explored to determine the most effective model structure, including filter analysis, CNN layers, activation functions, optimizer type, batch size (epoch), and the appropriate number of dense layers. Filter size [2, 2] and [1, 2], number of filters [16, 32, 64, 128], output activation [TANH, RELU, LINEAR], Optimizer [Adam

FIGURE 3. Dome dataset, indoor photography from the Dome in the CMU Panoptic Studio [37]

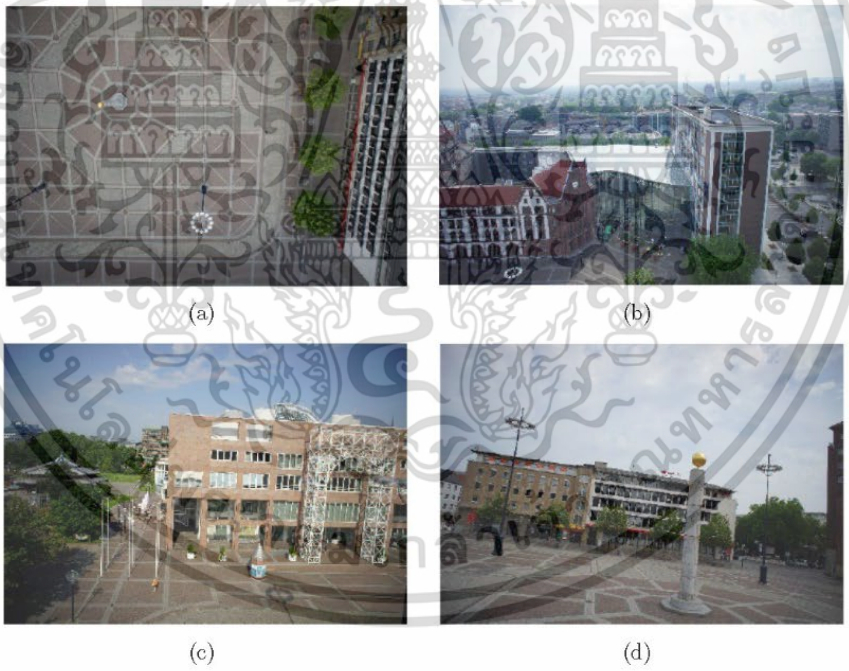


FIGURE 4. Outdoor dataset of aerial photographs [38]: (a) Nadir square; (b) Stadthaus; (c) Rathaus; (d) Obelisk oblique square

and Adadelta], batch size [32, 50, 100], number of epochs [200, 500], and number of density layers [2 (decrease density), 4] were among the parameters. The experiments were divided into two parts: the first part was a parameter analysis of the combined domain single model described in Section 3.1 and that of the ensemble CNN model detailed in Section

FIGURE 5. Outdoor dataset of Cambridge dataset [29]: (a) GreatCourt; (b) KingsCollege; (c) OldHospital; (d) ShopFacade; (e) StMarysChurch; (f) Street

3.2, and the second part was performance comparison among the research references [32], the combined domain single model and our proposed ensemble CNN model using the Root Mean Square Error (RMSE) as the performance metric illustrated in Section 3.3.

3.1. **Parameter analysis of a combined domain single model.** In the real world, we were occasionally confronted with unbalanced datasets, where the amount of data from several domains was unequally distributed. The indoor dataset was several times larger than the outdoor dataset approximately $1:100$ for this experiment, which could impact the system's overall performance. When the data from multiple domains are combined for the purpose of training a model, the proportions of features from larger domains tend to dominate the results. This means that predictions for domains with large volumes of data tend to be more accurate, as opposed to domain data with a limited volume, which results in poor predictive performance. According to the results of the experiment, we discovered that this effect starts to appear when the proportion of distinct data for each domain is different by 3 or more times. Consequently, we create a model trained on both the same $(1:1)$ and distinct quantities $(1:3, 3:1)$. The expectation is that, when evaluated with our approach, the approximations will provide comparable results even for domain datasets with unequal distribution. To evaluate the effects of an unbalanced dataset, we

conducted three trials with different ratios of training images from the two datasets: 1 : 1, 3 : 1, and 1 : 3 of the number of training images from the indoor dataset relative to the number of training images from the outdoor dataset, respectively. For performance evaluation, an equal number of test images were distributed across two domains. The average RMSE of rotation and translation across various optimizers, activations, epochs, and other configurations was shown in Table 1.

TABLE 1. The average RMSE values of rotation and translation predictions from the combined domain single model

Ratio of training images from two domain dataset	Activation	Minimum RMSE of rotation and translation
Set 1 – 1 : 1 (number of training indoor images = the number of training outdoor images = 1000)	Tanh+linear	0.616
	Tanh+linear (Reduce Dense)	0.655
	Tanh+relu+linear	**0.474**
	Tanh+relu+linear (Reduce Dense)	0.602
Set 2 – 3 : 1 (number of training indoor images = 1500, the number of training outdoor images = 500)	Tanh+linear	0.675
	Tanh+linear (Reduce Dense)	**0.548**
	Tanh+relu+linear	0.551
	Tanh+relu+linear (Reduce Dense)	0.697
Set 3 – 1 : 3 (number of training indoor images = 500, the number of training outdoor images = 1500)	Tanh+linear	0.620
	Tanh+linear (Reduce Dense)	0.684
	Tanh+relu+linear	**0.495**
	Tanh+relu+linear (Reduce Dense)	0.548

The average RMSE of rotation and translation predictions from training model with three sets of indoor and outdoor datasets and testing with 578 images each from indoor and outdoor test set was shown in Table 1. The minimum average RMSE of the training set 1 was 0.47409921 from the model parameters: 4 CNN layers with the number of filters (128, 32, 32, 16) for rotation model and (128, 16, 64, 16) for translation model, of each layer respectively, adadelta optimizer, 500 epochs, activation function (tanh+relu+linear), and 4 dense layers whereas the minimum average RMSE of the training set 2 was 0.548575566 from the model parameters: 4 CNN layers with the number of filters (64, 32, 32, 16) for rotation and (128, 16, 32, 16) for translation, of each layer respectively, adadelta optimizer, 500 epochs for rotation and 200 epochs for translation, activation function (tanh+linear (Reduce Dense)), and 2 dense layers. The minimum average RMSE of the third training set was 0.495257541 from the model parameters: 4 CNN layers with the number of filters (128, 128, 128, 16) for rotation and (128, 32, 32, 16) for translation, of each layer respectively, adadelta optimizer for rotation and adam optimizer for translation, 500 epochs, activation function (tanh+relu+linear).

3.2. **Parameter analysis of ensemble CNN method.** In this section, we trained two domain-specific agents, one from an indoor (Dome) dataset and the other from an outdoor (Map) dataset, with an integration of rotation and translation prediction into the same model. The experimental results with different feature arrangements from each domain-specific agent as mentioned in Section 2.1.2 named as Methods A and B from the training sets are shown in Figure 6.

It can be seen in Figure 6 that the minimum average RMSE of training set 1 was 0.131917386317026 with best model parameters as 4 CNN layers with the number of filters (64, 128, 128, 128), adam optimizer, 500 epochs, and 4 dense layers for indoor

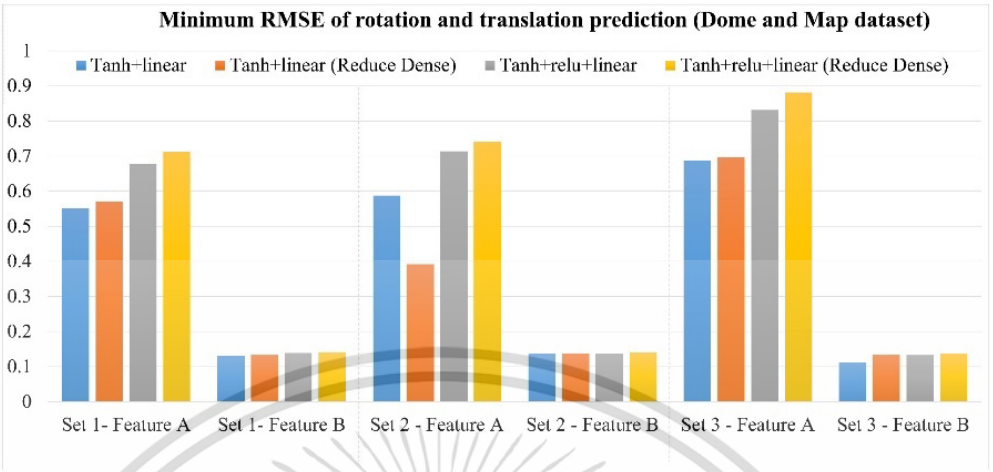


FIGURE 6. The minimum average RMSE of rotation and translation predictions for training sets 1, 2, and 3 with different ratios of training images from indoor (Dome) and outdoor (Map) datasets

Dome dataset and the number of filters (128, 128, 64, 16), adam optimizer, 500 epochs, and 4 dense layers for outdoor Map dataset, and activation functions (tanh+linear) for both datasets.

The minimum average RMSE of training set 2 is 0.137139154124717 with best model parameters as 4 CNN layers, the number of filters (128, 128, 128, 64) and 500 epochs for indoor Dome dataset and the number of filters (128, 128, 128, 128) and 200 epochs for outdoor Map dataset and activation functions (tanh+relu+linear) and adam optimizer for both datasets.

Accordingly, the minimum average RMSE of training set 3 was 0.11201226608203 with best model parameters as 4 CNN layers, the number of filters (128, 128, 128, 32) with adadelta optimizer and 200 epochs for indoor Dome dataset, and the number of filters (128, 128, 16, 16) with adam optimizer and 500 epochs for outdoor Map dataset, and activation functions (tanh+linear) for both datasets.

From the experiment, we noticed that the feature arrangement influenced the accuracy of the rotation and translation predictions. A feature arrangement Method B that combined features from the layer before the output layer of a domain-specific agents and constructed 128×2 features (64 rotation features and 64 translation features) from both indoor and outdoor domain-specific agents yielded the best results. The model trained on feature arrangement Method B was tested on the Cambridge dataset and the achieving median localization errors for both position and orientation were presented in Figure 7.

Comparing the localization accuracy across the datasets, as shown by a cumulative histogram, can reveal the differences in relative errors between them. It can be shown that the Street in the Cambridge Landmarks' outdoor dataset appears to be the most difficult to analyze. The same observation is pointed out by Walch et al. [40] concerning this unique behavior on this dataset. This dataset comprises videos recorded in opposite compass directions with similar spatial positions resulting in large angular deviations at similar global position. Although, OldHospital has a smaller spatial extent, it has relatively lower localization accuracy than KingsCollege because of large spatial camera movements. The cumulative distributions errors show that datasets with large angular deviations (ShopFacade) resulted in higher orientation errors than scenarios where the camera did not undergo severe rotations.

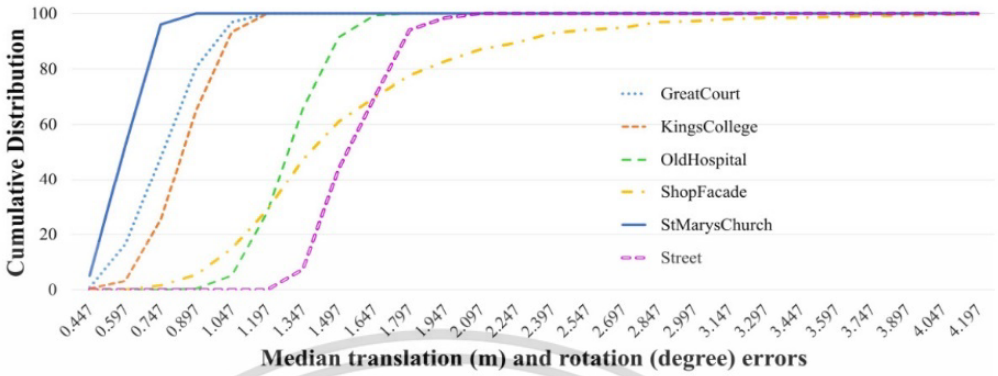


FIGURE 7. The median of rotation and translation predictions for Cambridge dataset

3.3. **The performance comparison of the reference model, the combined domain single model, and the proposed CNN-based multi-model ensemble method.** In this section, we illustrated the performance comparison among the reference technique [33], the implemented combined domain single model (2.1.1), and our proposed learning agent network model with different methods of feature arrangements (Methods A and B) on 3 training sets with various ratios of training images from Dome and Map datasets and several activation functions as shown in Table 2.

TABLE 2. The minimum average RMSE of the proposed ensemble CNN model and combined domain single model for multiple data domains

The number of training images divided	[33]	Combined domain single model	Ensemble CNN (Method A)	Ensemble CNN (Method B)
Set 1	2.501	0.474	0.495	0.131
Set 2		0.548	0.391	0.137
Set 3		0.495	0.686	0.112

Table 2 shows that the proposed ensemble CNN model with feature arrangement Method B provides the best performance for all training sets and activation function options with best minimum average RMSE of 0.112012266 which is 0.362086944 less prediction errors than the combined domain single model and 0.27951777 less prediction errors than the proposed ensemble CNN model with feature arrangement Method A, respectively. Moreover, when compared to the reference technique [33], the proposed ensemble CNN with feature organization Method B achieves lower prediction errors by 0.342181824, 0.411436412, and 0.460477478 for the three training sets and lower average RMSE by 2.389914218.

Predictions from a variety of reliable models can be combined to improve accuracy. A good model has skill, which means it can make better predictions than likelihood. Another important consideration is that the models must be good in a variety of ways, with a range of prediction errors. The reason that model averaging is effective is that different models do not always make the same mistakes on the test set when they are compared. The reason that model averaging is effective is that different models do not always make the same mistakes on the test set. When multiple neural networks' predictions are combined, a bias is introduced, which counteracts the variance of a single trained neural network

model. The result is predictions that are less sensitive to training data specifics, training scheme selection, and the serendipity of a single training run.

Table 3 illustrates the implementation of the Cambridge dataset, which is often used to evaluate the efficacy of camera pose estimation algorithms. A different shooting style was used in this dataset compared to Dome (indoor with differing view angles to the object) and Map (outdoor with the downward view angles from a drone flying horizontally). There are six subsets, the first five of which were locales, and the sixth of which was the street. The last one was the most difficult combination of images since the images were generally textureless. The results in Table 3 showed that the results for the first 10 methods were based on Shavit and Ferens [49], where LearnLess (DSAC++) [48] provided the least median error for subsets 1 to 5. However, no report for the last subset was found by Brachmann and Rother [48]. Despite this, our technique showed significantly lower median rotation and translation errors than the others. For subsets 1 to 5, the total of median errors from our approach was fewer than Brachmann and Rother [48] by 0.51° and 0.84 m, respectively, and less than Active Search (SfM) [49] by 0.89° and 2.2 m for subsets 2 to 6. Furthermore, for subsets 1 to 6, the methods in [33,34] reported the median errors, in which our method provides less median error by 32.21 m, 0.37 m and 39.58°, 1°, respectively.

TABLE 3. Median translation (m) and rotation errors for different pose estimators – using the Cambridge dataset

Algorithm	Great-Court	Kings-College	Old-Hospital	Shop-Facade	StMarys-Church	Street
PoseNet [29]	NA	1.97 m, 5.40°	2.31 m, 5.38°	1.46 m, 8.08°	2.65 m, 8.48°	3.67 m, 6.50°
Dense PoseNet [29]	NA	1.66 m, 4.86°	2.57 m, 5.14°	1.41 m, 7.18°	2.45 m, 7.96°	2.96 m, 6.00°
Bayesian PoseNet [42]	NA	1.74 m, 4.06°	2.57 m, 5.14°	1.25 m, 7.54°	2.11 m, 8.38°	2.14 m, 4.96°
LSTM-Pose [43]	NA	0.99 m, 3.65°	1.51 m, 4.29°	1.18 m, 7.44°	1.52 m, 6.68°	NA
SVS-Pose [44]	NA	1.06 m, 2.81°	1.50 m, 4.03°	0.63 m, 5.73°	2.11 m, 8.11°	NA
PoseNet+Reprojection error pose loss [45]	7.00 m, 3.7°	0.99 m, 1.1°	2.17 m, 2.9°	1.05 m, 4.0°	1.49 m, 3.40°	20.7 m, 25.7°
VLocNet [46]	NA	0.83 m, 1.42°	1.07 m, 2.41°	0.59 m, 3.53°	0.63 m, 3.91°	NA
DSAC [47]	2.80 m, 1.5°	0.30 m, 0.5°	0.33 m, 0.6°	0.09 m, 0.40°	0.55 m, 16°	NA
LearnLess (DSAC++) [48]	0.4 m, 0.2°	0.18 m, 0.3°	0.20 m, 0.3°	0.06 m, 0.30°	0.13 m, 0.4°	NA
Active Search [49]	NA	0.42 m, 0.6°	0.44 m, 1.0°	0.12 m, 0.40°	0.19 m, 0.5°	0.85 m, 0.8°
VGG19+CNN [33]	0.16 m, 0.18°	0.20 m, 0.21°	0.22 m, 0.48°	0.11 m, 0.20°	0.10 m, 0.39°	0.77 m, 0.76°
Our new system	**0.06 m, 0.12°**	**0.18 m, 0.05°**	**0.09 m, 0.46°**	**0.09 m, 0.13°**	**0.09 m, 0.08°**	**0.68 m, 0.38°**

4. **Conclusion.** In this research, we presented the estimation of the rotation and translation parameters of two-dimensional images used to reconstruct the 3D images using the proposed ensemble CNN model called a CNN-based multi-model ensemble method. To ensure flexibility in learning a variety of domains, the ensemble CNN model was composed of domain-specific agents and the domain relationship agent. The arrangement of features acquired from domain-specific agents and to be learned by the domain relationship agent was quite significant and had a large influence on the accuracy of the proposed ensemble CNN model. The feature arrangement that combined features from the layer before the output layer of both indoor and outdoor domain-specific agents and constructed 128×2 features (64 rotation features and 64 translation features) produced the best results, according to the experiments. Different ratios of training images and several test sets from indoor and outdoor domain datasets from various locations and shooting perspectives were evaluated in the experiments. The ensemble CNN model showed the highest predictive performance compared to the other algorithms. The results revealed that outdoor prediction has great potential for improvement. Since the Map dataset received by the drone's camera produces a large misalignment estimation error, we would like to train a specialized CNN that learns on additional depth parameters to assist with feature estimation prior to final aggregate estimation. We hope to expand our technique into these areas in the future.

REFERENCES

[1] C. Forster, M. Pizzoli and D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp.15-22, doi: 10.1109/ICRA.2014.6906584, 2014.

[2] J. Engel, J. Sturm and D. Cremers, Camera-based navigation of a low-cost quadrocopter, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2815-2821, doi: 10.1109/IROS.2012.6385458, 2012.

[3] M. Achtelik, M. Achtelik, S. Weiss and R. Siegwart, Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments, *2011 IEEE International Conference on Robotics and Automation*, pp.3056-3063, doi: 10.1109/ICRA.2011.5980343, 2011.

[4] H. Lim, S. N. Sinha, M. F. Cohen and M. Uyttendaele, Real-time image-based 6-DOF localization in large-scale environments, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp.1043-1050, doi: 10.1109/CVPR.2012.6247782, 2012.

[5] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys and R. Siegwart, Get out of my lab: Large-scale, real-time visual-inertial localization, *Computer Vision and Pattern Recognition*, doi: 10.15607/RSS.2015.XI.037, 2019.

[6] C. Ding, K. Liu, F. Cheng and E. Belyaev, Spatio-temporal attention on manifold space for 3D human action recognition, *Appl. Intell.*, pp.560-570, 2021.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Edition, Cambridge University Press, 2004.

[8] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, Speeded-up robust features (SURF), *Computer Vision and Image Understanding*, vol.110, no.3, pp.346-359, 2008.

[9] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, ORB: An efficient alternative to SIFT or SURF, *2011 International Conference on Computer Vision*, pp.2564-2571, doi: 10.1109/ICCV.2011.6126544, 2011.

[10] E. N. Mortensen, H. Deng and L. Shapiro, A SIFT descriptor with global context, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol.1, pp.184-190, doi: 10.1109/CVPR.2005.45, 2005.

[11] S. Choudhary and P. J. Narayanan, Visibility probability structure from SfM datasets and applications, *European Conference on Computer Vision (ECCV)*, 2012.

[12] Y. Li, N. Snavely and D. P. Huttenlocher, Location recognition using prioritized feature matching, *European Conference on Computer Vision (ECCV)*, vol.6312, pp.791-804, 2010.

[13] T. Sattler, B. Leibe and L. Kobbelt, Efficient & effective prioritized matching for large-scale image-based localization, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.39, no.9, pp.1744-1756, doi: 10.1109/TPAMI.2016.2611662, 2017.

[14] E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova and D. Aouada, Deep learning advances on different 3D data representations: A survey, *http://arxiv.org/abs/1808.01462v2*, 2019.

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger (eds.), Red Hook, NY, USA, Curran Associates, Inc., 2012.

[16] S. Tural, R. Samet, S. Aydin and M. Traore, Deep learning based classification of military cartridge cases and defect segmentation, *IEEE Access*, vol.10, pp.74961-74976, 2022.

[17] W. Wang, C. Tang, X. Wang, Y. Luo, Y. Hu and J. Li, Image object recognition via deep feature-based adaptive joint sparse representation, *Computational Intelligence and Neuroscience*, vol.2019, Article ID: 8258275, 2019.

[18] S. Lee and K. Jo, Person browser system based on named entity recognition for broadcast news interview videos, *Int. J. Control Autom. Syst.*, vol.19, pp.186-199, https://doi.org/10.1007/s12555-019-0391-z, 2021.

[19] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, *International Conference on Machine Learning*, pp.6105-6114, arXiv Preprint, arXiv: 1905.11946, 2019.

[20] C. Szegedy et al., Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp.1-9, doi: 10.1109/CVPR.2015.7298594, 2015.

[21] N. Westlake, H. Cai and P. Hall, Detecting people in artwork with CNNs, *European Conference on Computer Vision (ECCV)*, pp.825-841, 2016.

[22] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao and B. Catanzaro, Improving semantic segmentation via video propagation and label relaxation, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.8856-8865, 2019.

[23] V. Badrinarayanan, A. Kendall and R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.39, no.12, pp.2481-2495, doi: 10.1109/TPAMI.2016.2644615, 2017.

[24] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, Deep learning for computer vision: A brief review, *Computational Intelligence and Neuroscience*, vol.2018, Article ID: 7068349, 2018.

[25] A. X. Chang, T. Funkhouser, L. Guibas et al., ShapeNet: An information-rich 3D model repository, *http://arxiv.org/abs/1512.03012*, 2019.

[26] S. Choi, Q.-Y. Zhou, S. Miller and V. Koltun, A large dataset of object scans, *http://arxiv.org/abs/1602.02481*, 2019.

[27] S. Song, S. P. Lichtenberg and J. Xiao, SUN RGB-D: A RGB-D scene understanding benchmark suite, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp.567-576, 2015.

[28] I. K. Kazmi, L. You and J. J. Zhang, A survey of 2D and 3D shape descriptors, *Proc. of the 2013 10th International Conference Computer Graphics, Imaging and Visualization*, Los Alamitos, CA, USA, pp.1-10, 2013.

[29] A. Kendall, M. Grimes and R. Cipolla, PoseNet: A convolutional network for real-time 6-DoF camera relocalization, *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[30] A. Cohen, J. L. Schönberger, P. Speciale, T. Sattler, J.-M. Frahm and M. Pollefeys, Indoor-outdoor 3D reconstruction alignment, in *Computer Vision – ECCV 2016, Lecture Notes in Computer Science*, B. Leibe, J. Matas, N. Sebe and M. Welling (eds.), vol.9907, Springer, https://doi.org/10.1007/978-3-319-46487-9_18, 2016.

[31] Y. Furukawa and J. Ponce, Accurate, dense, and robust multiview stereopsis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, no.8, pp.1362-1376, doi: 10.1109/TPAMI.2009.161, 2010.

[32] B. Wattanacheep and O. Chitsobhuk, Prediction of 3D rotation and translation from 2D images, *The 7th International Conference on Computer and Communications Management (ICCCM2019)*, Association for Computing Machinery, New York, NY, USA, pp.49-52, https://doi.org/10.1145/3348445.3348485, 2019.

[33] B. Wattanacheep and O. Chitsobhuk, Camera pose estimation using CNN, *2020 the 3rd International Conference on Control and Computer Vision (ICCCV'20)*, Association for Computing Machinery, New York, NY, USA, pp.84-88, doi: https://doi.org/10.1145/3425577.3425593, 2020.

[34] O. Lantang, G. Terdik, A. Hajdú and A. Tiba, Comparison of single and ensemble-based convolutional neural networks for cancerous image classification, *Annales Mathematicae et Informaticae (54.)*, pp.45-56, http://dx.doi.org/10.33039/ami.2021.03.013, 2021.

[35] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *The 3rd International Conference on Learning Representations (ICLR2015)*, https://arxiv.org/abs/1409.1556, 2015.

[36] R. A. Sadek, SVD based image processing applications: State of the art, contributions and research challenges, *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol.3, no.7, pp.26-34, https://arxiv.org/ftp/arxiv/papers/1211/1211.7102.pdf, 2012.

[37] H. Joo, H. S. Park and Y. Sheikh, MAP visibility estimation for large-scale dynamic 3D reconstruction, *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp.1122-1129, doi: 10.1109/CVPR.2014.147, 2014.

[38] F. C. Nex, M. Gerke, F. Remondino, H. J. Przybilla, M. Baumker and A. Zurhorst, ISPRS benchmark for multi-platform photogrammetry, *Annals of the Photogrammetry, Remote Sensing and Spatial Information Science*, vol.II-3/W4, Munich, Germany, pp.135-142, 2015.

[39] S. M. Noe, T. T. Zin, P. Tin and I. Kobayashi, Automatic detection and tracking of mounting behavior in cattle using a deep learning-based instance segmentation model, *International Journal of Innovative Computing, Information and Control*, vol.18, no.1, pp.211-220, 2022.

[40] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck and D. Cremers, Image-based localization using LSTMs for structured feature correlation, *Proc. of the IEEE International Conference on Computer Vision*, pp.627-637, 2017.

[41] A. F. Siregar and T. Mauritsius, Ulos fabric classification using android-based convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.17, no.3, pp.753-766, 2021.

[42] K. Alex and R. Cipolla, Modelling uncertainty in deep learning for camera delocalization, *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp.4762-4769, arXiv Preprint, arXiv: 1509.05909, 2016.

[43] S. Bell, C. L. Zitnick, K. Bala and R. Girshick, Inside-outside Net: Detecting objects in context with skip pooling and recurrent neural networks, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp.2874-2883, doi: 10.1109/CVPR.2016.314, 2016.

[44] T. Naseer and W. Burgard, Deep regression for monocular camera-based 6-DoF global localization in outdoor environments, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.1525-1530, doi: 10.1109/IROS.2017.8205957, 2017.

[45] A. Kendall and R. Cipolla, Geometric loss functions for camera pose regression with deep learning, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp.6555-6564, 2017.

[46] V. Abhinav, N. Radwan and W. Burgard, Deep auxiliary learning for visual localization and odometry, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp.6939-6946, 2018.

[47] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold and C. Rother, DSAC – Differentiable RANSAC for camera localization, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp.6684-6692, https://github.com/cvlab-dresden/DSAC, 2017.

[48] E. Brachmann and C. Rother, Learning less is more – 6D camera localization via 3D surface regression, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp.4654-4662, https://github.com/vislearn/LessMore, 2018.

[49] Y. Shavit and R. Ferens, Introduction to camera pose estimation with deep learning, *arXiv Preprint*, arXiv: 1907.05272, 2019.

[50] J. Zhang, L. Chen, J. Tian et al., Breast cancer diagnosis using cluster-based undersampling and boosted C5.0 algorithm, *Int. J. Control Autom. Syst.*, vol.19, pp.1998-2008, https://doi.org/10.1007/s12555-019-1061-x, 2021.

[51] A. Manna, R. Kundu, D. Kaplun et al., A fuzzy rank-based ensemble of CNN models for classification of cervical cytology, *Sci. Rep.*, vol.11, 14538, https://doi.org/10.1038/s41598-021-93783-8, 2021.

[52] C.-H. Liao, S.-M. Chen, B.-C. Kuo and K.-C. Pai, A Chinese vocabulary learning system: Latent semantic analysis approach, *International Journal of Innovative Computing, Information and Control*, vol.10, no.6, pp.2179-2191, 2014.

[53] A. Y. Yousif, S. M. Younis, S. A. Hussein and N. M. G. Al-Saidi, An intelligent computing for diagnosing COVID-19 using available blood tests, *International Journal of Innovative Computing, Information and Control*, vol.18, no.1, pp.57-72, 2022.

[54] S. Lata and O. Surinta, An end-to-end Thai fingerspelling recognition framework with deep convolutional neural networks, *ICIC Express Letters*, vol.16, no.5, pp.529-536, doi: 10.24507/icicel.16.05.529, 2022.
[55] J. Wietrzykowski and D. Belter, Stereo plane R-CNN: Accurate scene geometry reconstruction using planar segments and camera-agnostic representation, *IEEE Robotics and Automation Letters*, vol.7, no.2, pp.4345-4352, doi: 10.1109/LRA.2022.3150841, 2022.
[56] A. G. Barnston, Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score, *Weather and Forecasting*, vol.7, no.4, pp.699-709, 1992.

Author Biography

Bhattarabhorn Wattanacheep received the B.E. degree in Computer Engineering from King Mongkut's Institute of Technology, Ladkrabang, Thailand, in 2012, the M.S. degree in Computer Engineering from King Mongkut's Institute of Technology, Ladkrabang, Thailand, in 2014. She is currently a student in Ph.D. program (Electrical Engineering at King Mongkut's Institute of Technology Ladkrabang, Thailand). Her research interests include robotics, image processing and optimization of technologies using machine learning.

Orachat Chitsobhuk received the B.E. degree in Electronics Engineering from King Mongkut's Institute of Technology, Ladkrabang, Thailand, in 1992, the M.S. degree in Computer Engineering from Arizona State University, AZ, in 1997, and the Ph.D. degree in Electrical Engineering from University of Texas, Arlington, US, in 2001. She is currently an associate professor and a lecturer at King Mongkut's Institute of Technology Ladkrabang, Thailand. Her research interests include image and scene analysis, machine learning and pattern recognition, and hardware design for image processing applications.

ประวัติผู้เขียน

ชื่อ-นามสกุล นางสาวภัทรภร วัฒนาชีพ

ประวัติการศึกษา

1) พ.ศ. 2552 – 2555 วิศวกรรมศาสตรบัณฑิต สาขาวิชาคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2) พ.ศ. 2556 – 2558 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงาน

3) พ.ศ. 2559 – 2562 มหาวิทยาลัยเกษมบัณฑิต

4) พ.ศ. 2562 – ปัจจุบัน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผลงานวิชาการ

1) Wattanacheep, B., & Chitsobhuk, O. 2015. “Plane alignment algorithm for torn document reconstruction.” **12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).** 1-5.

2) Wattanacheep, B., & Chitsobhuk, O. 2020. “Camera pose estimation using CNN.” **the 3 rd International Conference on Control and Computer Vision (ICCCV’20)**, Association for Computing Machinery, New York, NY, USA, 84-88, doi: https://doi.org/10.1145/3425577.3425593, 2020.

3) Wattanacheep, B., & Chitsobhuk, O. 2019. “Prediction of 3D rotation and translation from 2D images”, **The 7th International Conference on Computer and Communications Management (ICCCM2019)**, Association for Computing Machinery, New York, NY, USA,49-52, https://doi.org/10.1145/3348445.3348485,

4) Wattanacheep, B., & Chitsobhuk, O. 2022. “A CNN-based multi-model ensemble method for indoor and outdoor multi-view stereo reconstruction.” **International Journal of Innovative Computing, Information and Control**, vol.18, no.5, 1-15.