

# Floor Detection

การตรวจหาพื้น (Floor Detection) เป็นอีกแขนงหนึ่งของ Image Processing ที่มีการศึกษาอยู่ตลอด เพื่อใช้ในการตรวจหาพื้นให้กับหุ่นยนต์ในการเคลื่อนที่และหลบสิ่งกีดขวางเพื่อนำทางให้กับหุ่นยนต์ไปยังจุดหมายนั้นได้ แต่ก็มีอุปสรรคในการตรวจจับเหมือนกับการตรวจจับอื่นทั่วไปคือเรื่องของการสะท้อน แสงเงา หรือแม้แต่ลายของพื้น เป็นต้น (รูปที่ 1) ทำให้เรื่องนี้เป็นเรื่องท้าทายในวงการ Image Processing อยู่เสมอ โดยการใช้วิธีการดังนี้



รูปที่ 1 ตัวอย่างพื้น

## Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot

เป็นวิธีการตรวจหาพื้นแบบหนึ่งที่ใช้เพียง 1 รูปขาวดำ ไม่ต้องนำหลายๆรูปมาหา Optical flow (รูปที่ 2) หรือ Stereo (รูปที่ 3) ซึ่งวิธีนี้เกือบจะเป็นแบบเรียลไทม์แล้ว แล้วยังมีความแม่นยำสูงถึง 90% โดยวิธีนี้จะใช้สามวิธี (รูปที่ 4) มารวมกันเพื่อมาหาลำเส้นแบ่งพื้น ดังนี้



## ขั้นตอนที่ 1 : ตรวจหาเส้นตัด

เนื่องจากทั้ง 3 วิธี จำเป็นต้องใช้เส้นตัดแนวนอน หรือ Horizontal Line Segments ในการช่วยหาคำตอบ (รูปที่ 5) จึงต้องตรวจหาเส้นตัดเป็นลำดับแรก มี 3 ขั้นตอนดังนี้

### 1.1 ตรวจสอบและแยกประเภทของเส้นตัด

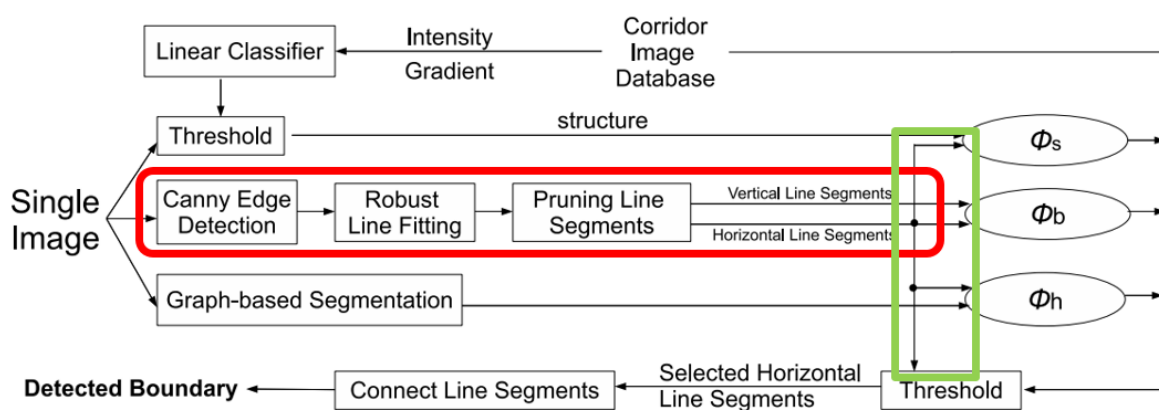
เริ่มด้วยการหาเส้นขอบของภาพด้วยวิธี Canny Edge Detection (รูปที่ 6) แล้วนำมาผ่าน Douglas-Peucker algorithm คือการเชื่อมจุดหรือเส้นให้มีองศาการเอียงน้อยลงและยังคงรูปร่างเดิมอยู่ (รูปที่ 7) และ Visual detection of lintel-occluded doors (รูปที่ 8) ในการเพิ่มความถูกต้องและแม่นยำในส่วนขอบล่างของประตูหลังจากนั้นนำเส้นที่ได้ทั้งหมดมาผ่านการกรองด้วยองศาการเอียงที่เก็บมาจากรูปร่างข้อมูลรูปพื้นทั้งหมด 300 รูป ดังนี้ เส้นแนวตั้ง ต้องเอียงได้ไม่เกิน 5 องศา และแนวนอนต้องเอียงได้ไม่เกิน 45 องศา ถ้าเกินกว่าค่าดังกล่าวให้ทำการตัดออกทั้งหมด ได้ผลลัพธ์ดังรูปที่ 9

## 1.2 เส้นตัด Pruning

นำเส้นที่เหลือจากข้อ 1.1 มาทำการผ่านการกรองด้วยความยาวของเส้นดังนี้ เส้นแนวตั้งต้องมีความยาวมากกว่า 60 พิกเซล และเส้นแนวนอนต้องมีความยาวมากกว่า 15 พิกเซล ถ้าน้อยกว่าค่าดังกล่าวให้ทำการตัดออกทั้งหมด แล้วนำเส้นตัดแนวนอนมาหาจุดตัดเป็นคู่ๆ เพื่อหาค่าเฉลี่ยแกน Y ที่จะใช้เป็น Vanishing Line จากสมการ Homogeneous Coordinates

$$\begin{bmatrix} wv_x \\ wv_y \\ w \end{bmatrix} = \sum_{i,j} \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} \times \begin{bmatrix} a_j \\ b_j \\ c_j \end{bmatrix}$$

ดังนั้นเส้นตัดแนวนอนที่อยู่เหนือกว่าค่าเฉลี่ย  $\bar{Y}$  นี้ให้นำออกไปทั้งหมด ได้ผลลัพธ์ดังรูปที่ 10

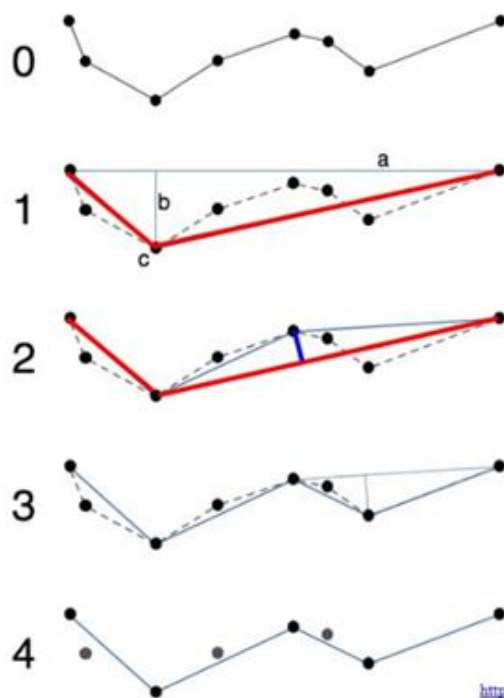


รูปที่ 5 ขั้นตอนการหาเส้นตัดและความสัมพันธ์ของเส้นตัดแนวนอน



รูปที่ 6 ตัวอย่างภาพที่ได้จากการใช้ Canny Edge Detection

## Douglas-Peucker Algorithm



An algorithm that:

- Reduces the number of points in a curve that is approximated by a series of points

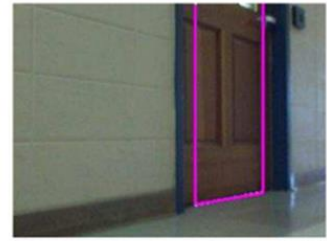
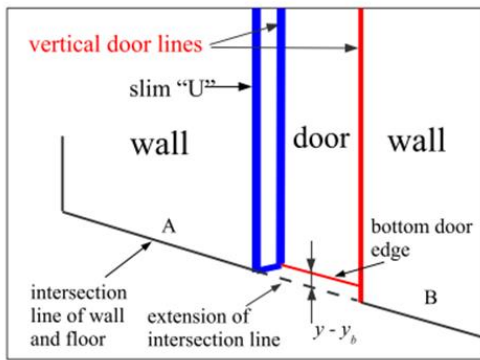
- Focus on maximum distance between the original curve and the simplified curve

- Finds a similar curve with fewer points

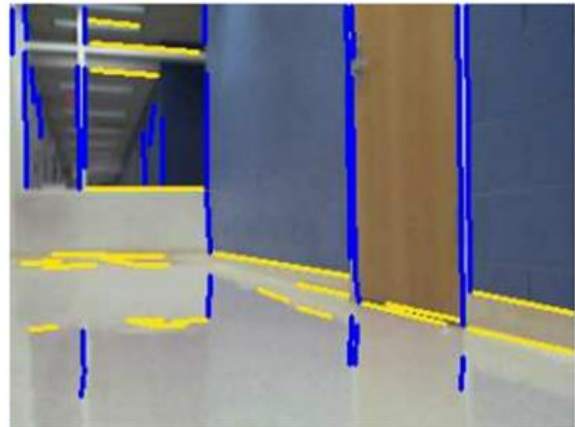
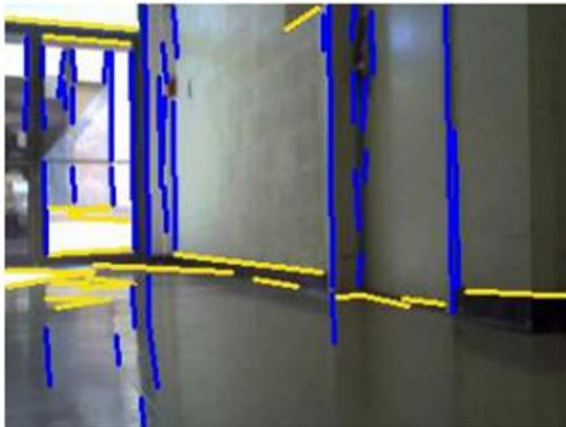
[http://en.wikipedia.org/wiki/Ramer%E2%80%99s%E2%80%93Douglas%E2%80%93Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer%E2%80%99s%E2%80%93Douglas%E2%80%93Peucker_algorithm)  
David Douglas & Thomas Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer 10(2), 112-122 (1973)

รูปที่ 7 ตัวอย่างการใช้งาน Douglas-Peucker algorithm

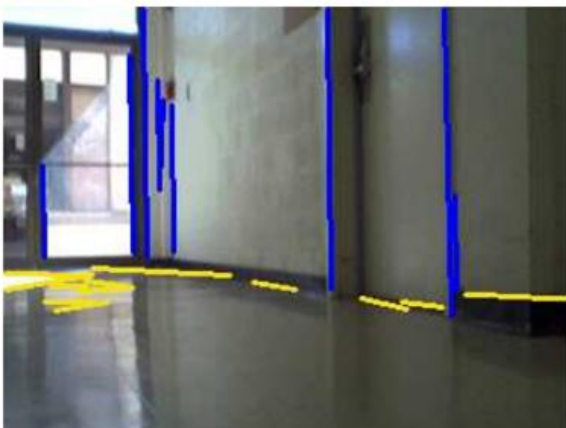




รูปที่ 8 คำอธิบายและผลลัพธ์ของวิธี Visual detection of lintel-occluded doors



รูปที่ 9 ผลลัพธ์จากการตรวจหาและแยกประเภทเส้นตัด



รูปที่ 10 ผลลัพธ์สุดท้ายจากการหาเส้นตัด

## ขั้นตอนที่ 2 : ให้คะแนนและประเมินเส้นตัด

เมื่อเราได้เส้นตัดที่ผ่านกระบวนการมาแล้วระดับหนึ่ง ซึ่งก็ยังมีเส้นที่ยังไม่เกี่ยวข้องอยู่ จึงจำเป็นต้องนำมาผ่านขั้นตอนการให้คะแนนแล้วประเมินว่าเส้นไหนควรจะเป็นเส้นที่นำมาใช้ตรวจหาพื้นได้ (รูปที่ 11) โดยมีวิธีการให้คะแนนทั้งหมด 3 แบบดังนี้

## 2.1 คะแนนโครงสร้าง

จากการสำรวจพบว่าความเข้มข้นของขอบที่เกิดขึ้นจากการสะท้อนบนพื้นมีแนวโน้มที่จะมีค่า Intensity สูง แต่ค่า Gradient Magnitude ต่ำ เพราะว่าการมัวโดยธรรมชาติเกิดขึ้นเนื่องจากการสะท้อนที่ไม่สมบูรณ์ของพื้น และจากข้อมูลรูปภาพ 400 รูป เลือกมา 800 จุดที่เป็นจุดจริง ๆ บนขอบกำแพง และจุดอื่นๆที่ถูกสุ่มมาแต่ไม่ได้อยู่บนขอบกำแพงเท่าๆกัน แล้วนำมาพลอตเป็นกราฟ แล้วนำข้อมูลที่ได้มาสร้างเส้น Support Vector Machine หรือ SVM จึงได้ออกมาดังรูปที่ 12 หลังจากนั้นนำมาสร้างค่า Threshold ด้วยสมการ

$$\tau_{LC} = \frac{1}{|\xi'|} \sum_{(x,y) \in \xi'} I(x,y).$$

โดยที่  $\xi'$  คือจุดที่อยู่เหนือเส้น SVM แล้วนำค่าจากรูปมาผ่าน Threshold นี้ดังสมการ

$$S(x,y) = (I(x,y) > \tau_{LC}).$$

ค่าที่ได้นำมาแทนในสมการ ซึ่งเป็นการหาผลรวมค่าระยะห่างระหว่างพิกเซล ดังนี้

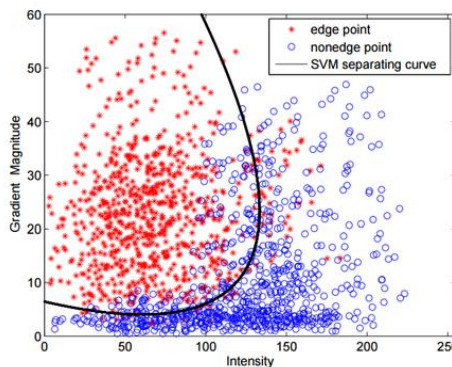
$$\phi_s(\ell_h) = \sum_{(x,y) \in \ell_h} d[(x,y), S],$$

สุดท้ายนำมา Normalize ด้วย Gaussian distribution ดังสมการนี้

$$\bar{\phi}_s(\ell_h) = \exp \left\{ -\frac{\phi_s(\ell_h)}{2\sigma_s^2} \right\},$$

where  $\sigma_s = 10$ .

จะได้มาซึ่งคะแนนโครงสร้าง



รูปที่ 12 กราฟระหว่าง Intensity และ Gradient Magnitude และเส้น SVM

## 2.2 คะแนนเส้นล่าง

เริ่มต้นด้วยการกรองเส้นแนวตั้งที่จุดต่ำสุดของเส้นอยู่เหนือครึ่งของรูป นำจุดต่ำสุดในแนวแกน X ของเส้นแนวตั้งดังกล่าวมาเชื่อมกัน จากนั้นนำเส้นแนวนอนมาคำนวณหาระยะห่างระหว่างแต่ละจุดกับเส้นเชื่อมดังกล่าว ดังสมการ

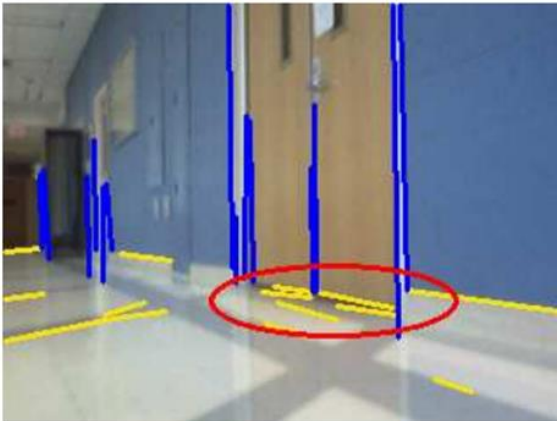
$$\phi_b(\ell_h) = \sum_{(x,y) \in \ell_h} d[(x,y), \ell_b],$$

สุดท้ายนำมา Normalize ด้วย Gaussian distribution ดังสมการนี้

$$\bar{\phi}_b(\ell_h) = \exp \left\{ -\frac{\phi_b(\ell_h)}{2\sigma_b^2} \right\},$$

where  $\sigma_b = 30$ .

จะได้มาซึ่งคะแนนเส้นล่าง



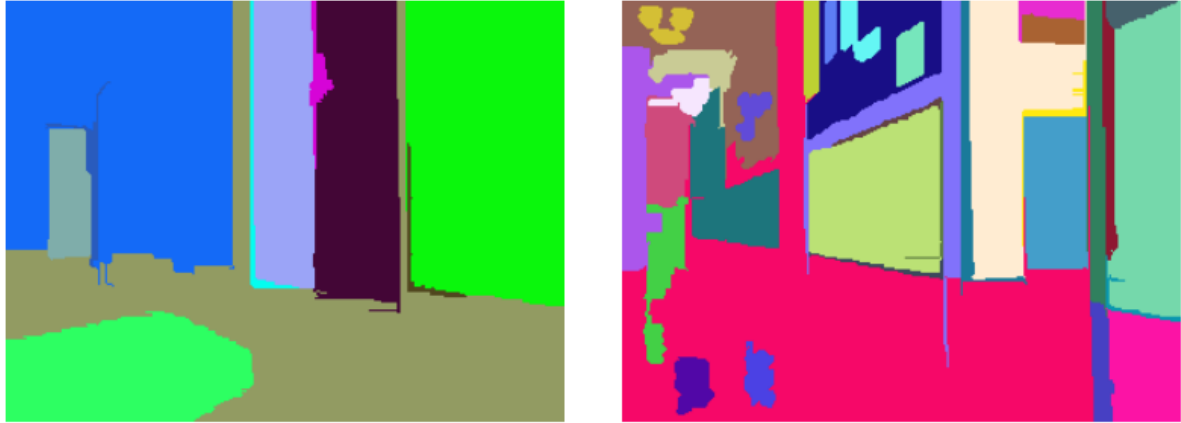
รูปที่ 13 เส้นแนวนอนที่อยู่ในวงสีแดงมีคะแนนสูงเนื่องจากอยู่ระหว่างจุดต่ำสุดของเส้นแนวตั้ง

## 2.3 คะแนนความเหมือน

เริ่มต้นด้วยการแบ่งส่วนของรูปภาพด้วย Graph-based Segmentation จะได้ผลลัพธ์ดังรูปที่ 14 แล้วนำมาคำนวณด้วยสมการ

$$\bar{\phi}_h(\ell_h) = \frac{|\mathcal{R}|}{|\mathcal{R}_{\max}|},$$

โดยที่  $|\mathcal{R}|$  หมายถึงจำนวนพิกเซลที่อยู่ใต้เส้น และ  $|\mathcal{R}_{\max}|$  ขนาดที่ใหญ่ที่สุดในส่วนที่ถูกแบ่งมา ที่เป็นแบบนี้เพราะว่าส่วนที่แบ่งได้ใหญ่ที่สุดมักจะเป็นพื้น จะได้มาซึ่งคะแนนความเหมือน



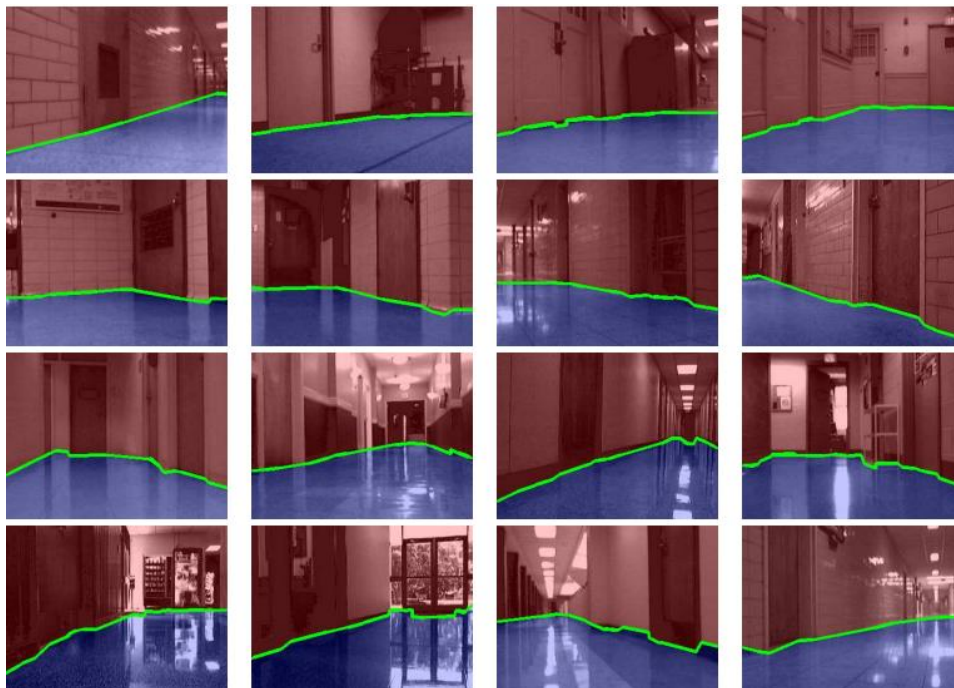
รูปที่ 14 ผลลัพธ์จากการแบ่งส่วนของรูปภาพด้วย Graph-based Segmentation

### ขั้นตอนที่ 3 : ตรวจสอบเขตแดนของพื้น

โดยนำเส้นทั้งหมดที่เหลื้อมาผ่าน Threshold สุดท้ายที่ได้มาจากรูปร่างข้อมูล ดังสมการนี้

$$\Phi_{total}(\ell_h) > \tau_\phi,$$

สุดท้ายนำเส้นทั้งหมดมาเชื่อมกันจะได้มาซึ่งเขตแดนของพื้นดังรูปที่ 15



รูปที่ 15 ตัวอย่างแสดงเขตแดนของพื้น