**Canny Edge Detection**

เป้าหมาย

ในบทนี้เราจะเรียนรู้

* แนวคิดการตรวจจับขอบ Canny
* ฟังก์ชัน OpenCV สำหรับที่: **cv2.Canny ()**

ทฤษฎี

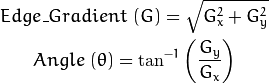
Canny Edge Detection เป็นอัลกอริทึมการตรวจหาขอบที่เป็นที่นิยม ได้รับการพัฒนาโดย John F. Canny ในปี 1986 เป็นขั้นตอนหลายขั้นตอนและเราจะดำเนินการผ่านแต่ละขั้นตอน

1. Noise Reduction

เนื่องจากการตรวจหาขอบมีความอ่อนไหวต่อสัญญาณรบกวนในภาพขั้นตอนแรกคือการขจัดเสียงรบกวนในภาพด้วยตัวกรองแบบเกาส์ 5x5 เราได้เห็นสิ่งนี้แล้วในบทก่อน ๆ

1. Finding Intensity Gradient of the Image

ภาพที่นุ่มนวลจะถูกกรองด้วยเค้ก Sobel ทั้งในแนวนอนและแนวตั้งเพื่อรับอนุพันธ์แรกในทิศทางแนวนอน ( G_x) และทิศทางแนวตั้ง ( G_y) จากภาพทั้งสองภาพนี้เราสามารถหาการไล่ระดับสีและทิศทางของขอบสำหรับแต่ละพิกเซลได้ดังนี้:



ทิศทางไล่ระดับสีจะตั้งฉากกับขอบเสมอ มีการปัดเศษให้เป็นหนึ่งในสี่มุมที่แสดงแนวเส้นทแยงมุมแนวนอนและแนวนอนสองเส้น

1. Non-maximum Suppression

หลังจากได้รับการไล่ระดับสีและทิศทางการสแกนภาพเสร็จสิ้นเพื่อลบพิกเซลที่ไม่พึงประสงค์ซึ่งอาจไม่เป็นขอบ สำหรับพิกเซลทุกพิกเซลจะได้รับการตรวจสอบว่าเป็นพิกเซลในละแวกใกล้เคียงที่สุดหรือไม่ในทิศทางของการไล่ระดับสี ตรวจสอบภาพด้านล่าง:

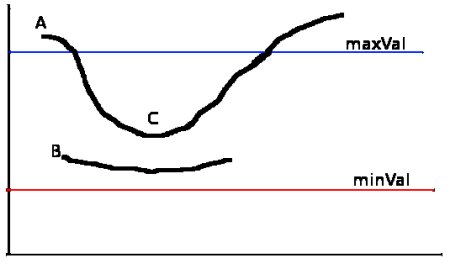


จุด A อยู่ที่ขอบ (ในแนวตั้ง) ทิศทางไล่โทนเป็นปกติที่ขอบ จุด B และ C อยู่ในทิศทางไล่ระดับสี ดังนั้นจุด A จะถูกตรวจสอบกับจุด B และ C เพื่อดูว่ามีรูปแบบสูงสุดแบบโลคอล ถ้าเป็นเช่นนั้นถือว่าเป็นขั้นตอนต่อไปมิฉะนั้นจะถูกระงับ (วางไว้ที่ศูนย์)

ในระยะสั้นผลที่ได้คือภาพไบนารีที่มี "ขอบบาง ๆ "

1. **Non-maximum Suppression**

ขั้นตอนนี้จะตัดสินใจว่าขอบทั้งหมดเป็นขอบที่แท้จริงและไม่เป็นเช่นนั้น สำหรับเรื่องนี้เราต้องสองค่าเกณฑ์*minVal*และMAXVALขอบที่มีการไล่ระดับความเข้มมากกว่า*maxVal*จะต้องเป็นขอบและด้านล่าง*minVal*จะตรวจสอบให้แน่ใจว่าไม่ใช่ขอบดังนั้นจึงถูกละทิ้ง ผู้ที่อยู่ระหว่างสองเกณฑ์เหล่านี้เป็นขอบที่แยกกันหรือไม่ใช่ขอบตามการเชื่อมต่อของพวกเขา หากเชื่อมต่อกับพิกเซล "แน่ใจว่าขอบ" พวกเขาจะถือว่าเป็นส่วนหนึ่งของขอบ มิเช่นนั้นจะถูกทิ้งด้วย ดูภาพด้านล่าง:



ขอบ A อยู่เหนือ maxVal ดังนั้นถือว่าเป็น "sure-edge" แม้ว่าขอบ C ต่ำกว่า maxVal จะเชื่อมต่อกับขอบ A เพื่อให้ถือว่าเป็นขอบที่ถูกต้องและเราได้รับเส้นโค้งเต็มรูปแบบ แต่ขอบ B แม้ว่าจะอยู่เหนือ minVal และอยู่ในบริเวณเดียวกับขอบของ C แต่จะไม่เชื่อมต่อกับ "ขอบที่แน่ใจ" เพื่อให้ถูกทิ้ง ดังนั้นจึงเป็นสิ่งสำคัญมากที่เราต้องเลือกminVal และ maxVal ตามเพื่อให้ได้ผลลัพธ์ที่ถูกต้อง

ขั้นตอนนี้จะ removes small pixels noises ขอบเป็นเส้นยาว

ดังนั้นสิ่งที่เราได้รับในที่สุดก็คือขอบที่แข็งแกร่งในภาพ

Canny Edge Detection ใน OpenCV

OpenCV ทำให้ทั้งหมดข้างต้นในการทำงานเดียว**cv2.Canny**() เราจะมาดูกันว่าจะใช้อย่างไร อาร์กิวเมนต์แรกเป็นภาพนำเข้าของเรา ข้อโต้แย้งที่สองและสามเป็นของเรา*minVal*และ*MAXVAL*ตามลำดับ อาร์กิวเมนต์ที่สามคือaperture\_sizeมีขนาดของเคอร์เนล Sobel ที่ใช้สำหรับค้นหาการไล่ระดับสีภาพ ค่าเริ่มต้นคือ 3. อาร์กิวเมนต์สุดท้ายคือ*L2gradient*ซึ่งระบุสมการหาค่า gradient ถ้าเป็นจริงมันใช้สมการข้างต้นที่ถูกต้องมากกว่ามิฉะนั้นจะใช้ฟังก์ชันนี้: ขอบ \ _Gradient \;  (G) = | G_x |  + | G_y |. โดยค่าเริ่มต้นมันเป็นเท็จ

