**ORB (แบบเร็วและแบบหมุนเร็ว)**

เป้าหมาย

ในบทนี้,

* เราจะดูพื้นฐานของ ORB

ทฤษฎี

ในฐานะที่เป็นผู้ที่ชื่นชอบ OpenCV สิ่งที่สำคัญที่สุดเกี่ยวกับ ORB คือมันมาจาก "OpenCV Labs" ขั้นตอนนี้ถูกนำขึ้นโดย Ethan Rublee, Vincent Rabaud, Kurt Konolige และ Gary R. Bradski ในกระดาษORBของพวกเขา: ทางเลือกที่มีประสิทธิภาพในการ SIFT หรือ SURFในปี 2011 ในฐานะที่เป็นชื่อกล่าวว่าเป็นทางเลือกที่ดีสำหรับ SIFT และ Surur ในการคำนวณ ต้นทุนการจับคู่ประสิทธิภาพและหลักสิทธิบัตร ใช่ SIFT และ SURF มีสิทธิบัตรและคุณควรจะจ่ายเงินเพื่อการใช้งาน แต่ ORB ไม่ได้ !!!

ORB เป็นแบบผสมผสานของเครื่องตรวจจับจุดสำคัญ FAST และคำอธิบายแบบย่อที่มีการปรับเปลี่ยนมากมายเพื่อเพิ่มประสิทธิภาพ ก่อนอื่นให้ใช้ FAST เพื่อค้นหาคีย์พอยน์แล้วใช้มาตรการแฮร์ริสมุมเพื่อหาจุดสูงสุด N ในหมู่พวกเขา นอกจากนี้ยังใช้พีระมิดเพื่อผลิตคุณสมบัติหลายองค์ประกอบ แต่ปัญหาหนึ่งคือ FAST ไม่ได้คำนวณการวางแนว ดังนั้นสิ่งที่เกี่ยวกับ invariance หมุน? ผู้เขียนเกิดการปรับเปลี่ยนต่อไป

คำนวณค่าความหนาแน่นถ่วงน้ำหนักของแพทช์ที่มีมุมตั้งอยู่ตรงกลาง ทิศทางของเวกเตอร์จากจุดมุมนี้ไปยัง centroid ช่วยในการวางแนว เพื่อปรับปรุงความไม่แปรปรวนของการหมุนช่วงเวลาจะคำนวณด้วย x และ y ซึ่งควรอยู่ในพื้นที่วงกลมรัศมีRซึ่งRเป็นขนาดของแพทช์

ตอนนี้สำหรับคำอธิบาย ORB ใช้คำอธิบายแบบย่อ แต่เราได้เห็นแล้วว่า BRIEF มีประสิทธิภาพต่ำเมื่อมีการหมุนเวียน ดังนั้นสิ่งที่ ORB ทำคือ "คัดท้าย" ย่อตามการวางแนวของคีย์พ็อพ สำหรับชุดคุณลักษณะของnการทดสอบไบนารีที่ตำแหน่ง (x_i, y_i)ให้กำหนด2 ครั้ง \ nเมตริกซ์Sซึ่งมีพิกัดของพิกเซลเหล่านี้ แล้วใช้การวางแนวของแพทช์, \ thetaเมทริกซ์การหมุนของมันจะถูกพบและหมุนSที่จะได้รับพา (หมุน) S_ \ thetaเวอร์ชั่น

ORB เปลี่ยนมุมให้เป็นมุมที่เพิ่มขึ้น2 \ pi / 30(12 องศา) และสร้างตารางการค้นหาของรูปแบบ BREMF ที่ถูกคอมพิวเตอร ตราบเท่าที่การวางแนวจุดสำคัญ\ thetaสอดคล้องกันระหว่างมุมมองชุดที่ถูกต้องของจุดS_ \ thetaจะใช้ในการคำนวณ descriptor ของมัน

BRIEF มีคุณสมบัติที่สำคัญที่แต่ละบิตมีความแปรปรวนมากและค่าเฉลี่ยใกล้ 0.5 แต่เมื่อมีการมุ่งเน้นทิศทาง keypoint จะสูญเสียทรัพย์สินนี้และกลายเป็นกระจายมากขึ้น ความแปรปรวนสูงทำให้คุณลักษณะเด่นชัดขึ้นเนื่องจากตอบสนองต่ออินพุตแตกต่างกัน อีกคุณสมบัติที่น่าพอใจคือการทดสอบไม่สัมพันธ์กันตั้งแต่นั้นการทดสอบแต่ละชิ้นจะมีผลต่อผลลัพธ์ เพื่อแก้ปัญหาเหล่านี้ทั้งหมด ORB จะค้นหาการทดสอบไบนารีทั้งหมดที่เป็นไปได้เพื่อหาคนที่มีความแปรปรวนสูงและใกล้เคียงกับ 0.5 รวมถึงไม่มีความสัมพันธ์กัน ผลที่ได้คือเรียกว่าrBRIEF

ใช้สำหรับการจับคู่แบบ descriptor multi-probe LSH ซึ่งใช้กับ LSH แบบเดิม กระดาษบอกว่า ORB เร็วกว่า SURF และ SIFT และ ORB descriptor ทำงานได้ดีกว่า SURF ORB เป็นทางเลือกที่ดีสำหรับอุปกรณ์ไฟฟ้าที่มีกำลังไฟต่ำสำหรับการเย็บภาพพาโนรามา ฯลฯ

ORB ใน OpenCV

ตามปกติเราต้องสร้างอ็อบเจ็กต์ ORB ด้วยฟังก์ชันcv2.ORB ()หรือใช้อินเทอร์เฟซทั่วไป feature2d มีพารามิเตอร์ที่เป็นตัวเลือก คนที่มีประโยชน์มากที่สุดคือn คุณลักษณะซึ่งหมายถึงจำนวนสูงสุดของคุณสมบัติที่จะถูกเก็บรักษาไว้ (โดยเริ่มต้น 500) scoreTypeซึ่งหมายถึงว่าแฮร์ริสทำคะแนนหรือคะแนนในการจัดอันดับของ FAST คุณสมบัติ (โดยค่าเริ่มต้นคะแนนแฮร์ริส) เป็นต้นพารามิเตอร์อื่นWTA\_Kตัดสินใจจำนวนจุด ที่ผลิตแต่ละองค์ประกอบของคำอธิบายสั้น ๆ ที่มุ่งเน้น ค่าเริ่มต้นคือสองคือเลือกสองจุดในแต่ละครั้ง ในกรณีที่การจับคู่, NORM\_HAMMINGระยะถูกนำมาใช้ ถ้า WTA\_K เป็น 3 หรือ 4 ซึ่งใช้เวลา 3 หรือ 4 คะแนนในการสร้างคำอธิบายแบบย่อจากนั้นระยะการจับคู่จะถูกกำหนดโดยNORM\_HAMMING2

ด้านล่างนี้เป็นรหัสง่ายๆที่แสดงการใช้ ORB

**import** **numpy** **as** **np**

**import** **cv2**

**from** **matplotlib** **import** pyplot **as** plt

img = cv2.imread('simple.jpg',0)

*# Initiate STAR detector*

orb = cv2.ORB()

*# find the keypoints with ORB*

kp = orb.detect(img,None)

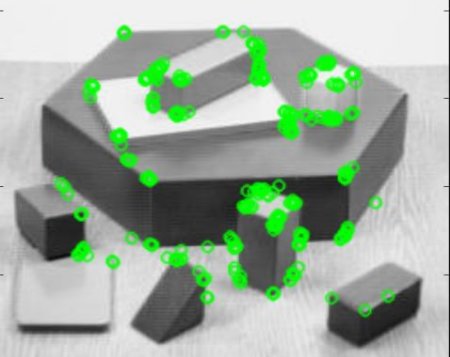
*# compute the descriptors with ORB*

kp, des = orb.compute(img, kp)

*# draw only keypoints location,not size and orientation*

img2 = cv2.drawKeypoints(img,kp,color=(0,255,0), flags=0)

plt.imshow(img2),plt.show()



การจับคู่คุณลักษณะ ORB เราจะทำในบทอื่น