Báo cáo chương 10

OpenCV

1. The OpenCV Python Interface
2. OpenCV Basics

* Reading and writing images: sử dụng hàm imread() và imwrite()
* Color spaces:
  + Trong opencv, ảnh được lưu ở kênh màu BGR, chúng ta có thể chuyển chúng thông qua các kênh màu khác sử dụng cvtColor().
  + Một số code hưu hiệu như:
    - cv2.COLOR\_BGR2GRAY
    - cv2.COLOR\_BGR2RGB
    - cv2.COLOR\_BGR2BGR
* Displaying images and results



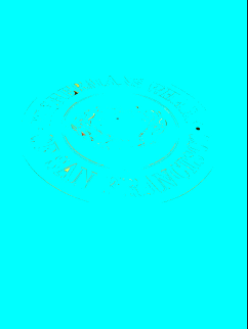
Hình 1 Integral image

A picture containing text, food, dark

Description automatically generated Calendar

Description automatically generated

Hình 2 Ngưỡng khác biệt cao nhất và thấp nhất là (6, 6, 6) trong hàm floodFill()



Hình 3 Ngưỡng khác biệt cao nhất là (40, 40, 40) trong hàm floodFill



Hình 4 SUFT feature

1. Processing Video

Sử dụng cv2.VideoCapture()

1. Tracking

* Oprical flow
  + Optical flow là một dịch chuyển ảnh của các đối tượng, ngữ cảnh, camera khi mà dịch chuyển giữa hai ảnh liên tiếp nhau. Nó là một trường vector 2D của phép dịch chuyển ảnh.
  + Optical flow dựa trên 3 giả thuyết chính:
    - Brightness constancy: cường độ sáng của các pixel của một đối tượng trong một ảnh không thay đổi giữa các ảnh liên tiếp.
    - Temporal regularity: Thời gian giữa các khung ảnh là đủ ngắn để xem như chuyển động giữa các ảnh sử dụng vi phân ( sử dụng để lấy ra phương trình trung tâm bên dưới
    - Spatial consistency: các pixel hàng xóm có chuyển động tương tự
  + Trong nhiều trường hợp các giả thuyết trên không thõa mãn, nhưng trong các chuyển động nhỏ và các bước thời gian ngắn giữa các ảnh nó là một mô hình tốt. Giả sử rằng một đối tượng pixel I(x, y, t) tại thời điểm t có cùng cường độ sáng tại thời điểm  sau dịch chuyển  có nghĩa rằng I(x y, t) = . Vi phân của ràng buộc này đưa ra một biểu thức luồng quang: . Trong đó v = [u, v] là một vector dịch chuyển và It là đạo hàm theo thời gian.Đối với mỗi điểm trong ảnh thì phương trình này chưa được xác định rõ và không thể giải quyết ( một phương trình với hai ẩn số trong v). Bằng cách thực thi một số nhất quán trong không gian, nó có thể có được một số giải pháp. Trong thuật toán Lucas-Kanade dưới đây, ta sẽ xem cách mà các giả thuyết được sử dụng.



* The lucas-Kanade algorithm
  + Việc theo dõi là quá trình theo dõi các đối tượng thông qua một chuỗi hình ảnh hoặc video. Hình thức cơ bản nhất của việc theo dõi là theo dõi các điểm quan tâm như là các góc. Một thuật toán phổ biến cho việc này là Lucas-Kanade tracking, cái mà sử dụng thuật toán luồng quang học (optical flow)
  + Lucsas-Kanade có thể được áp dụng với bất kì loại thuộc tính nào nhưng thường được cho các điểm góc tương tự như các điểm góc Harris. Hàm goodFeaturesToTrack() phát hiện các góc theo một thuật toán Shi và Tomasi, trong đó các góc là các điểm với hai giá trị riêng lớn của phương trình tensor cấu trúc (ma trận Harris) và trong đó giá trị riêng nhỏ hơn thì nằm trên một ngưỡng nào đó.
  + Biểu thức luồng quang học chưa được xác định ( có nghĩa rằng có nhiều tham số chưa biết) nếu được xem xét trên mỗi pixel. Sử dụng giả thuyết rằng các pixel hàng xóm có chuyển động cùng nhau nó có thể xếp nhiều phương trình với nhau để tạo thành một phương trình như sau: Text, letter

    Description automatically generated cho một vài hàng xóm của n pixel. Bây giờ có thể giải quyết bằng phương pháp bình phương tối thiểu. Thông thường, sự đóng góp của các pixel xung quanh được đánh trọng số để pixel ở xa ít ảnh hưởng hơn. Một trọng số Gaussian thường được chọn. Chúng ta có một liên hệ sau: Text

    Description automatically generated with medium confidence. Sử dụng phương pháp bình phương tối thiểu, vector chuyển động được cho bởi công thức: . Điều này chỉ có thể được giải quyết nếu A^TA khả nghịch, nó được xây dựng nếu được áp dụng các điểm góc Harris hoặc là “các thuộc tính tốt để theo dõi” của Shi-Tomasi. Đây là các để vector chuyển động được tính toán trong thuật toán theo dõi Lucas-Kanade.
  + Tiêu chí Lucas-Kanade theo dõi các công việc cho những thay đổi nhỏ. Để xử lý thay đổi lớn hơn hướng tiếp cận phân cấp được sử dụng. Trong trường hợp này, luồng quang học được tính tại từ phiên bản thô đến mịn của ảnh. Đây là cái hàm calcOpticalFlowPyrLK().
* Using the tracker:

A picture containing indoor, building, green, floor

Description automatically generated

A picture containing indoor, ceiling, floor, several

Description automatically generated

A picture containing indoor, ceiling, green, floor

Description automatically generated

A picture containing green, floor, lined

Description automatically generated

* Using generators:

A picture containing text, indoor

Description automatically generated

1. More example

* Impainting là việc tái tạo lại các phần bị mất hoặc xấu của một ảnh.