***Image stitching***

Image stitching hay photo stitching là quá trình kết hợp nhiều bức ảnh với vùng trùng lặp để tạo thành một bức ảnh toàn cảnh hay một bức ảnh có độ phân giải cao. Image titching yêu cầu phải xác định các vùng gần đúng và đồng nhất chúng thành bức ảnh hoàn chỉnh.

Ngày nay, nhiều máy ảnh kĩ thuật số có khả năng nối những bức ảnh ngay khi chụp.

Các giai đoạn của quá trình stitching: đăng kí ảnh,hiệu chỉnh ảnh, hòa trộn các bức ảnh.

Đăng kí ảnh bao gồm xác định các đặc trưng của các hình ảnh hoặc sử dụng phương pháp liên kết trực tiếp để tìm kiếm sự sắp xếp hình ảnh đó để giảm thiểu sự khác biệt giữa các pixel trùng lặp. khi sử dụng phương pháp liên kết trực tiếp thì cần phải hiệu chỉnh hình ảnh trước để đạt được kết quả tốt hơn.

Hiệu chỉnh ảnh nhằm mục đích giảm thiểu sự khác biệt giữa các mô hình ống kính lý tưởng và những ống kính máy ảnh được sử dụng, những thiếu sót như là: biến dạng, khiếm khuyết, sự khác biệt giữa các hình ảnh, họa tiết, các phản ứng quang học và quang sai màu.

Hòa trộn hình ảnh bao gồm việc thực hiện những điều chỉnh đã tìm ra trong giai đoạn hiệu chỉnh và kết hợp với remapping của các hình ảnh để tạo ra một phép chiếu đầu ra. Màu sắc giữa các hình ảnh được điều chỉnh để bù đắp cho những tiếp xúc khác biệt.

***Trích chọn đặc trưng***

Trong việc tạo một ảnh panorama thì ý tưởng chính của nó là xác định các điểm giống nhau của các ảnh và chiếu 1 trong 2 ảnh lên trước, sau đó nối các điểm giống nhau lại.

Các điểm giống nhau giữa các điểm được gọi là interest points.

Phép toán này được phát triển đầu tiên bởi Hans P.Moravec năm 1977 trong nghiên cứu “[automatic navigation of a robot through a clustered environment](http://www.stanford.edu/~learnest/cart.htm)”. Tại đây ông định nghĩa “interest of points” của một ảnh và kết luận rằng có thể dựa vào các “interest point” để nối các phần khác nhau của các bức ảnh.

Toán tử Moravec quan tâm đến các việc tìm kiếm các góc vì nó định nghĩa các “interest point” là các điểm có cường độ biến thiên lớn theo tất cả các hướng. Đây là trường hợp của các góc.

Toán tử tìm kiếm các “interest point” được Chris Harris và Mike Stephens phát triển tiếp năm 1988 với mục đích thiết lập môi trường cho robot dựa trên chuỗi hình ảnh. Giống như Moravec, họ tạo các phương thức để nối các điểm tương đương giữa các khung hình liên tiếp, nhưng họ quan tâm cả về các góc và các cạnh giữa các khung hình.

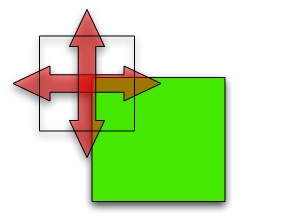
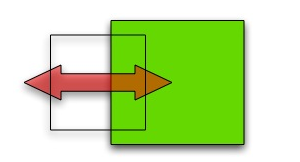
Harris và Stephens được cải tiến việc tìm kiếm góc của Moravec bằng cách xem xét sự khác biệt của các góc trực tiếp. Kĩ thuật tìm kiếm góc của Harris được thể hiện bằng việc tính trung bình các momen ma trận bằng cách tính Gradient của các ảnh, sau đó kết hợp với trị số đặc trưng của ma trận từ đó tính được một ước số góc. Dựa vào các ước số góc và chọn ra ước số lớn nhất để tìm kiếm vị trí góc.

Có những công nghệ tìm kiếm góc khác tốt hơn như SIFT và SURF nhưng đòi hỏi bản quyền và bằng sáng chế.

Giải thuật phát hiện góc dựa trên nguyên lý nếu bạn đặt một cửa sổ nhỏ trên một bức ảnh, nếu cửa sổ đó có chứa một góc, sau đó nếu cửa sổ đó được di chuyển theo bất kì hướng nào thì sẽ có sự thay đổi về cường độ.điều đó được minh họ trong hình dưới.

Nếu cửa sổ nằm ở trên vùng hình ảnh không chứa góc hay cạnh, sau khi di chuyển cửa sổ khung hình sẽ không có sự thay đổi về cường độ. Nếu cửa sổ nằm trên vùng có chứa cạnh thì nó sẽ có sự thay đổi về cường độ khi mà cửa sổ khung hình di chuyển theo một hướng.

Nếu cửa sổ khung hình có chứa một góc thì khi di chuyển cửa sổ khung hình theo bất kì hướng nào đều có sự thay đổi cường độ. Do đó ta biết được đâu là góc.



Giải thuật tìm kiếm góc của Harris minh chứng cho nguyên lý này.

Giải thuật:

Giả sử ta có ảnh xám I với mỗi điểm (u, v) có độ dịch chuyển là (x, y) ta có thể tính toán sai lệch khi cửa sổ khung hình dịch chuyển từ (u, v) sang (u+x, v+y) theo công thức sau:



Trong đó S(x, y) là tổng bình phương độ lệch, w(u, v) là cửa sổ khung hình tại vị trí (u, v), I(u, v) và I(u+x, v+y) là cường độ sáng của pixel tại các vị trí (u, v) và (u+x, v+y)

I(u+x, v+y) có thể khai triển Taylor như sau:



Trong đó Ix, Iy là các đạo hàm của I theo các thành phần x, y. Từ đó công thức S(x, y) có thể được viết lại như sau:



Nếu biểu diễn dưới dạng ma trận thì:



Trong đó A là một cấu trúc Tensor: ( là một ma trận nhận được từ hàm Gradient. Nó chứa các hướng đặc trưng(hướng trội) của gradient trong vùng xung quanh của một điểm, và mức độ của mỗi hướng đó)



Gọi λ1 và λ2 là các giá trị riêng của A. Khi đó:



Sẽ quyết định xem cửa sổ khung hình w(u, v) có chứa góc hay không. Với k là một ngưỡng cho trước (HarrisThreshold) để điều chỉnh số lượng góc cho mỗi ảnh, và thường có giá trị từ 0,01 đến 0,1. Nếu k lớn thì số lượng góc tìm thấy sẽ nhiều nhưng không chất lượng.

Nếu và thì cửa sổ khung hình tại (u, v) không chứa interest point.

Nếu và λ2 có giá trị khác 0 thì tìm thấy được cạnh.

Nếu cả λ1 và λ2 đều có các giá trị khác 0 thì cửa sổ khung hình tại (u, v) chứa góc.

SURF

SURF(Speeded Up Robust Features) được giới thiệu năm 2006 bởi nhóm nghiên cứu của Herbert Bay. Ý tưởng của thuật toán này tương đối giống với SIFT. SURF cũng dùng không gian scale để tìm kiếm điểm đặc trưng và các đặc trưng được mô tả bởi các vector có kèm thêm hướng. Thuật toán này có hai phần chính là “phát hiện” điểm đặc trưng, và “mô tả” điểm đặc trưng.

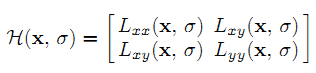
Phát hiện: Quá trình phát hiện điểm đặc trưng dựa trên không gian scale, và sử dụng mà trận Hessian ( chỉ sử dụng một xấp xỉ của ma trận này để đơn giản hơn trong việc tính toán). Các điểm đặc trưng là các điểm cực đại của hàm Hessian trong không gian scale.

Mô tả: mỗi điểm đặc trưng được thêm một hướng dùng để nhận dạng khi ảnh bị xoay, và được mô tả bằng một vector gồm 64 thành phần.

Cả hai bước trên đều sử dụng Integral Images để tăng tốc độ tính toán.

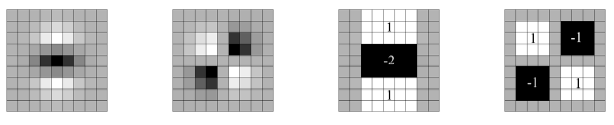
Phát hiện nhanh Hessian

Việc phát hiện đặc trưng của thuật toán SURF dựa vào ma trận Hessian, bởi vì đây là ma trận đảm bảo tốt về mặt thời gian cũng như độ chính xác. Trong phương pháp này, người ta có cho thêm một biến số scale σ vào ma trận Hessian để tạo không gian scale. Khi đó, với mỗi điểm X(x, y) của ảnh I, ma trận Hessian của X trong không gian scale là:



Với Lxx(**x**, σ) là đạo hàm bậc hai theo x của hàm Gaussian g(σ) tại điểm X của ảnh I. Tương tự như vậy, Lxy(**x**, σ) và Lyy(**x**, σ) lần lượt là đạo hàm bậc hai theo xy và yy của hàm Gaussian tại điểm X của ảnh I.

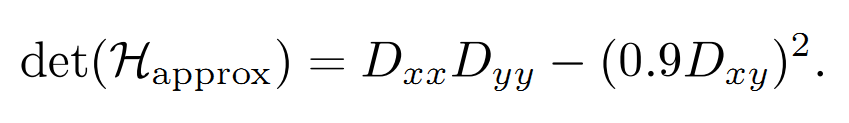
Hàm Gaussian là tối ưu trong phân tích không gian scale, nhưng trong thực tế nhiều trường hợp hàm Gaussian cần được rời rạc hóa và chia tách. Áp dụng trong phương pháp này, để đơn giản người ta chỉ dùng một xấp xỉ cho đạo hàm bậc hai của hàm Gaussian.



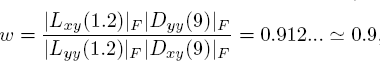
*Hai hình bên trái : Đạo hàm bậc 2 của Gaussian theo y và theo xy; Hai hình bên phải : xấp xỉ đạo hàm bậc 2 của Gaussian theo y và theo xy; vùng mầu xám tương ứng với giá trị bằng 0.*

Các hình chữ nhật 9x9 trong hình trên là các xấp xỉ của Đạo hàm bậc hai của hàm Gaussian với σ = 1.2 ( tỉ lệ scale nhỏ nhất ).

Gọi Dxx, Dyy, Dxy lần lượt là các xấp xỉ theo x, y, và xy của đạo hàm bậc hai hàm Gaussian. Ta được công thức:



Trong đó, w là bộ lọc được dùng để làm cân bằng cho biếu thức tính định thức Hessian. Và w được tính theo công thức:



Với |x|F là tiêu chuẩn Frobenius của ma trận x. Tiêu chuẩn Frobenius được định nghĩa như sau:

Không gian scale: Không gian scale giống như một kim tự tháp ảnh và chia làm nhiều mức khác nhau. Mỗi ảnh được lặp đi lặp lại với một hàm Gaussian và sau đó các mẫu được lưu trử ở các mức cao hơn của kim tự tháp.

Xác định vị trí điểm đặc trưng:

Để xác định vị trí điểm đặc trưng, trong mỗi mức độ của của không gian scale người ta sử dụng một ma trận 3x3, và mỗi điểm sẽ được so sánh với tất cả các láng giềng của nó.



Một điểm sẽ bị loại bỏ nếu nó không là cực đại trong số 26 láng giềng của nó (3x3x3). Điểm cực đại của ma trận Hessian được nội suy theo phương pháp mà Brown đã đề xuất.

Thêm hướng vào các điểm đặc trưng.

***Nối các đặc trưng***

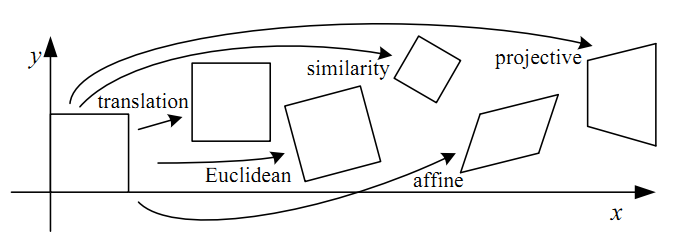
Khi đã xác định được các đặc trưng(các góc) từ các hình ảnh chúng ta sẽ thực hiện nối các đặc trưng tương đồng lại với nhau. Chúng ta sẽ sử dụng phương pháp tương quan tối đa để nối các đặc trưng đó lại với nhau. Bộ phân tích tương quan chéo làm việc bằng cách phân tích các điểm ảnh xung quanh tất cả các điểm đặc trưng trong ảnh thứ nhất và nối chúng với một cửa sổ các điểm ảnh xung quanh các điểm đặc trưng khác trong ảnh thứ hai. Các điểm có tương quan hai chiều lớn nhất sẽ được nối với nhau như các cặp tương ứng.

Khi đã xác định được các cặp điểm đặc trưng thì chúng ta cần xác định một mô hình để thực hiện việc dịch chuyển các điểm đặc trưng từ tập này sang tập khác.

***Mô hình chuyển động***

Trước khi chúng ta stitching image để tạo các ảnh panorama thì chúng ta cần phải xác định mối quan hệ toán học giữa các hình riêng rẽ.

Hình dưới là một số các mô hình chuyển động được sử dụng trong 2D phẳng.



=> biểu diễn các mô hình chuyển động này bằng toán học: sử dụng ma trận đồng nhất 3x3

X’ = (x’, y’,1) và x = (x, y, 1)

x’~Hx

~ : tương đương.

H là ma trận đồng nhất 3x3

D.O.F là khoảng cách giữa các đối tượng gần nhất và xa nhất trong một ảnh sắc nét chấp nhận được

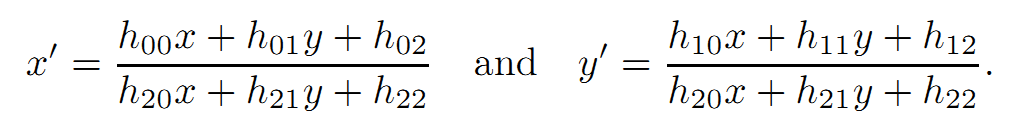
Translation(tịnh tiến) có 2 bậc tự do và thường dùng trong các bản vá lỗi nhỏ trong video và trong máy ảnh tức thời jitter.

Rotation(quay) và translation(tịnh tiến)(còn được gọi là chuyển động hình thể cứng hoặc phép biến đổi Euclidean) có 3 bậc tự do và được sử dụng trong mô hình chuyển động quay, ví dụ như các phần khác nhau của 1 bức ảnh lớn được quét trên một máy quét hình phẳng(flatbed scanner)

Scale(co giãn) và xoay hay còn gọi là phép biến đổi đồng dạng(similarity). phép biến đổi này bổ sung thêm một tỉ lệ co giãn nên có 4 bậc tự do. Mô hình này sử dụng tốt cho các loại máy ảnh phân tích chậm hoặc trong chức năng zoom của máy ảnh. phép biến đổi đồng dạng yêu cầu bảo toàn góc giữa các đường.

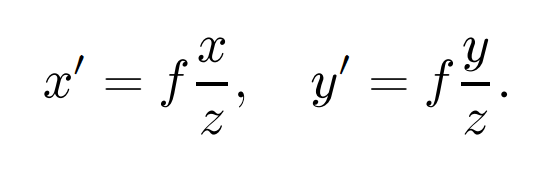
Biến đổi affine(afin) sử dụng ma trận 2x3(hoặc một ma trận tương đương 3x3 mà hàng cuối cùng là [0 0 1]) và có 6 bậc tự do. Mô hình này thường dùng để làm cho các hình ảnh bị biến dạng với các phép biến đổi phức tạp, và cũng dùng để vẽ các bề mặt 3D quan sát được theo luật gần xa bằng máy ảnh trực giao. Mô hình afin yêu cầu bảo toàn tính song song giữa các đường.

Phép chiếu(projective) có 8 bậc tư do và biến đổi theo luật xa gần (phép chiếu phối cảnh hoặc là homography). Phép biến đổi được biểu thị bằng một ma trận H có kích thước 3x3. Kết quả của phép nhân Hx phải được chuẩn hóa để đạt được kết quả không đồng nhất. có nghĩa là:



Phép biến đổi này đòi hỏi bảo toàn các đường thẳng. Mô hình này thích hợp cho việc biểu diễn các mô hình 3D hay quang cảnh 3D.

Trong 3D, quá trình xử lý phép chiều từ tọa độ 3D x = (x,y,z) thành x’ = (x’,y’,1) thông qua một ống kính lên mặt phẳng chiếu 2D với khoảng cách f dọc thep trục z.



Khi ta có hai ảnh 3D của một quang cảnh từ hai vị trí đặt máy ảnh khác nhau hoặc 2 hướng khác nhau. 1 điểm 3D p =(X, Y, Z, 1) được ánh xạ tới một tọa độ điểm ảnh x’0 thông qua sự kết hợp của một mô hình hình thể cứng(Euclidean) E0 và phép chiếu phối cảnh K0.



P0 là ma trận 3x4 và được gọi là ma trận camera. Nếu chúng ta có một điểm 2D x0, chúng ta chỉ có thể chiếu nó trong không gian 3D *ray.* Tuy nhiên, trong một cảnh phẳng chúng ta có một phương trình ñ0\*p+d0=0, có thể dùng phương trình này để chuyển từ p0 thành Ṕ và sau đó nó cho phép ta đảo ngược phép chiều 3D -> 2D.



H10 là một ma trận homography 3x3 và x1 và x0 là 2 tọa độ 2D đồng nhất. điều này cho thấy homography với 8 tham số đầu vào như là sự liên kết giữa các mô hình để tạo cảnh mosaic.

Homography là một phép chiếu phối cảnh, là loại biến đổi trong phép chiếu hình học.

Giả sử máy ảnh chỉ đơn giản là xoayvà tất cả mọi điểm ảnh đều xa máy ảnh. Khi đó chúng ta có một homography 3x3 với nhiều hạn chế.



System.drawing lớp matrix được đóng gói trong ma trận affine 3x3 biểu diễn các phép biến đổi hình học.

Trong Accord.Imaging lớp marixH đóng gói một ma trận đồng nhất 3x3 biểu diễn các phép chiếu.

Sự khác nhau giữa hai ma trận chỉ là các bậc tự do của mỗi ma trận. Mặc dù đều là ma trận 3x3, ma trận biến đổi hình học trong system.drawing có 6 bậc tự do, còn với ma trận phép chiếu trong Accord.Imaging có 8 bậc tự do. Giá trị cuối cùng trong ma trận có thể được hiểu là một tham số tỉ lệ và có thể được fix là 1. Tọa độ đồng nhất được sử dụng vì không thể thực hiện phép nhân ma trận trực tiếp trên phép chiếu phối cảnh.

Mỗi điểm ảnh có tọa độ là (x,y) đổi sang tọa độ đồng nhất là (x,y,w) và để đơn giản ta sẽ cho w cố định là 1.

=> tọa độ điểm ảnh là Ixy = (x,y,1)

Để thực hiện chiếu một hình ảnh ta thực hiện nhân ma trận Ixy với một ma trận chuẩn.

x =

=> để phục hồi lại tọa độ ban đầu ta có thể chia tọa độ I’xy cho w’ sau đó sẽ loại bỏ tham số thứ 3 sẽ thu đc tọa độ ban đầu.

(x’,y’,w’)/w’ = (x’/w’, y’/w’, 1)

Loại bỏ 1 => tọa độ Ixy = (x,y)

Vấn đề đặt ra là làm sao từ các điểm đặc trưng có thể thiết lập một ma trận homography. Để ước lượng một mô hình từ các dữ liệu thì ta sử dụng một phương thức gọi là RANSAC (RANdom SAmple Consensus).

Homography là một định nghĩa của toán học. đó là sự dịch chuyển sử dụng phép chiếu hình học, hay nói cách khác nó là sự kết hợp của cặp điểm trong phép chiếu phối cảnh. Ảnh thực trong không gian 3 chiều có thể biến đổi về không gian ảnh bằng phép chiều thông qua ma trận biến đổi Homography H. Nó không đảm bảo về kích thước và góc của vật được chiếu, nhưng bảo đảm về tỷ lệ.

Trong lĩnh vực thị giác máy, bất kì hai ảnh nào có cùng bề một phần chung đều liên quan đến homography. Nó có rất nhiều ứng dụng như: sửa ảnh, tính toán sự chuyển động, xoay, dịch chuyển giữa hai ảnh.

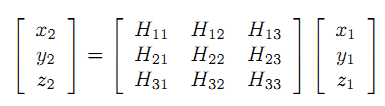
**X’ ~ HX**

X’*: là ảnh trong không gian camera.*

X*: là ảnh trong không gian thực.*

H*: là ma trận Homography. Ma trận H là ma trận đã được chứng minh là khả nghịch. Như vậy từ ảnh X’ muốn tái tạo lại ảnh X ta cần phải xác định ma trận Homography.*

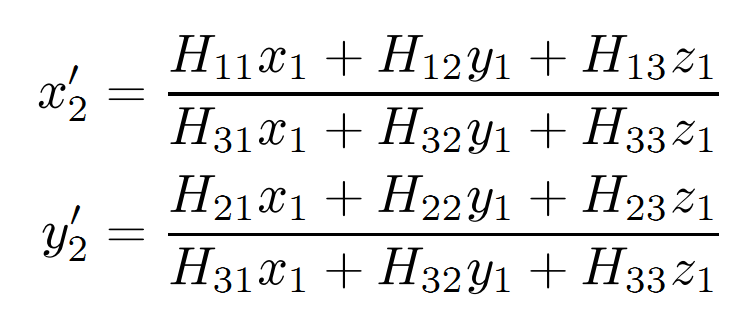
Trong tọa độ không đồng nhất ta có:



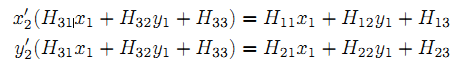
Công thức trên tương đương với công thức:

**X’ = HX**

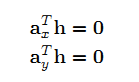
Với x2’ = x2/z2 và y2’ = y2/z2 ta có:



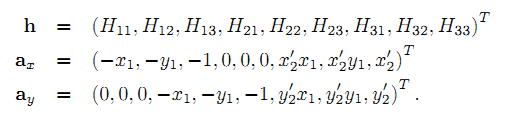
Để không mất tính tổng quát ta đặt z =1 ta được:



Từ 2 công thức trên ta có được công thức sau:



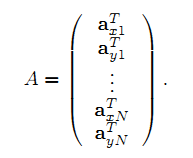
Với



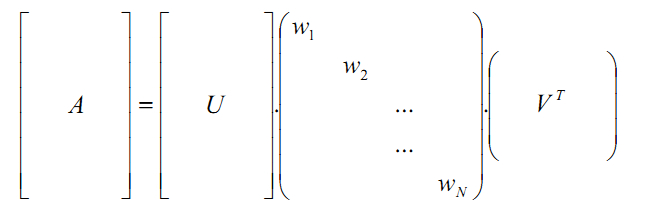
Với một tập hợp các điểm tương ứng ta có thể xác định được công thức sau:

Ah = 0

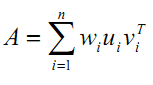
Với



*Sử dụng phương pháp SVD:* Phương pháp SVD dựa trên định lý trong đại số tuyến tính là bất kì ma trận A nào có kích thướng MxN mà , có thế được viết dưới dạng tích của một ma trận U trực giao theo cột có kích thước MxN, với một ma trận chéo W có kích thước NxN với các số trên đường chéo là không âm, và ma trận chuyển vị của một ma trận trực giao V có kích thước NxN

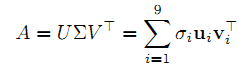


Nếu kí hiệu ui và vi là các tương ứng là các cột của U và V thì phương trình trên tương đương với



wi là các giá trị kì dị của A

Quay lại với bài toán tính Homography áp dụng công thức SVD, ta được:



Trong đó, các cột của ma trận V tương ứng với giá trị kì dị bằng 0 là nghiệm của hệ.

Do X và X’ có cùng một hướng nên

X’ x HX = 0

Các phần tử thứ j của ma trận H được viết là hjT, vì thế

Và X’T = (x’, y’, w’)T vì thế nên

X’HX =

Ta lại có: hjTX = XThj suy ra X’HX = 0 có thể viết thành

= 0

Thuật toán DLT

***RANSAC***

RANSAC là viết tắt của “RANdom SAmple Consensus”. Được công bố bởi Fischler và Bolles năm 1981. Thuật toán làm việc với một dữ liệu đầu vào với nhiều ngoại lai. Nó lặp đi lặp lại phương pháp ước lượng tham số để phù hợp với các mô hình toán học từ tập hợp các điểm dữ liệu chứa nhiều ngoại lai. Dữ liệu bao gồm các “inliers” và “outlier”, trong đó “inlier” là các dữ liệu được phân bổ và thiết lập các tham số mô hình, còn “outlier” là các dữ liệu không phù hợp với mô hình. Thêm vào đó các dữ liệu mới có thể có cả “outlier”. RANSAC giả định rằng với một tập các “inlier” (thường là nhỏ) thì có thể ước lượng được các thông số của mô hình với các dữ liệu này. Thuật toán có đầu ra là một sác xuất và nó ko thể đưa ra một kết quả hợp lý. Chỉ có thể kết luận rằng nếu sác xuất cao thì việc lặp đi lặp lại lớn.

///////////////////////////////

Có nhiều ứng dụng có sử dụng thuật toán này, như: giải quyết đồng thời các vấn đề tương ứng và ước lượng một ma trận có liên quan đến cặp stereo của máy ảnh.

/////////////////////////////

Thuật toán:

1. Lựa chọn ngẫu nhiên các điểm dữ liệu cần thiết để xác định các tham số mô hình.
2. Giải quyết các tham số của mô hình.
3. Xác định có bao nhiêu điểm từ tập hợp tất cả các điểm phù hợp với bước 2.
4. Nếu phần lớn các “inlier” trong tất cả các điểm vượt quá một ngưỡng T cho trước nào đó thì ước lượng lại mô hình tham số sử dụng tất cả các “inlier” đã được xác định trước đó và dừng thuật toán.
5. Nếu không thì quay lại bước 1 tới bước 4( tối đa là n lần)

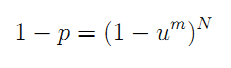
N được chọn là lớn nhất sao cho sác xuất p(thường là 0.99) là tối thiểu để một tập các mẫu ngẫu nhiên không chứa một “outlier”.

U đặc trưng cho sác xuất các dữ liệu được chọn là “inlier”.

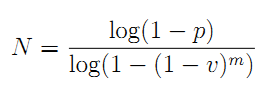
V = 1 – U là sác xuất của “outlier”.

m số lượng tối thiểu các điểm “inlier” cần phải lặp đi lặp lại.

Khi đó:



Từ đó suy ra:



Sau khi tính toán theo thuật toán trên ta xác định ma trận homography là tập các “inlier”(các điểm đặc trưng hoàn toàn chính xác tương quan giữa hai bức ảnh ) và loại bỏ các điểm “outlier”(các điểm đặc trưng không tương quan)

Sau khi tính toán được ma trận homography ta sẽ thực hiện việc hòa trộn hai hình ảnh lại với nhau.

/////////////////////////////////////////////

Ví dụ: Giả sử có 20 người trong đó có 19 người nghèo có thu nhập là 5 đồng và 1 người giàu có thu nhập là 1 tỷ. Nếu tính theo kiểu trung bình cộng thì bình quân mỗi người có thu nhập là khoảng 50 triệu đồng. Nếu tính theo kiểu trung vị thì mỗi người có thu nhập bình quân là 5 đồng. Cách thứ 2 cho phép giảm nhiễu, và nhiễu ở đây chính là người giàu. Phương pháp trung vị thực ra cũng chỉ hạn chế nhiễu chứ không loại bỏ hoàn toàn được nhiễu. Thuật toán RANSAC là một giải thuật khử nhiễu.

Ý tưởng cơ bản của thuật toán:

Tập dữ liệu đầu vào gồm hai loại dữ liệu là inlier và outlier. Trong đó:

Inlier là các dữ liệu chính xác, phù hợp với mô hình.

Outlier là các dữ liệu nhiễu, không phù hợp với mô hình.

Chọn n điểm ngẫu nhiên, tối thiểu để có thể xây dựng mô hình.

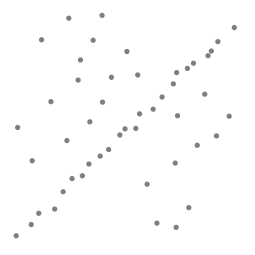
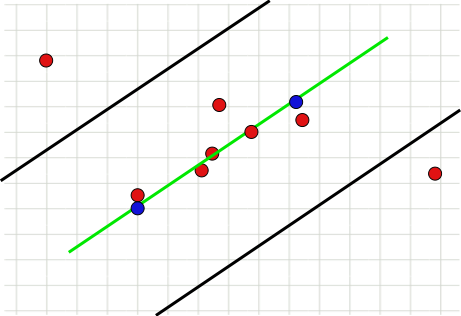
Từ các điểm đó có thể suy ra nhiều mô hình.

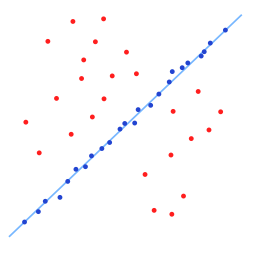
Đặt ra một ngưỡng nào đó, sau đó “bỏ phiếu”.

Hành động bỏ phiếu là hành động tính toán độ chính xác của mỗi điểm trong tập dữ liệu ban đầu với mô hình. Nếu vượt quá ngưỡng đã đặt ra thì gọi là outlier.

Sau khi “bỏ phiếu” sẽ là bước “kiểm phiếu”. Mô hình nào có nhiều phiếu bầu nhất sẽ được chọn là mô hình tốt nhất.

Ví dụ:



Thuật toán:

input:

data: một tập dữ liệu ban đầu.

model: một mô hình phù hợp với data.

n: số lượng dữ liệu nhỏ nhất để thiết lập nên mô hình.

k: số lần lặp lại của thuật toán.

t: một giá trị ngưỡng để xác định mốc tính toán cho mô hình

d: giá trị các điểm còn lại trong tập data đùng để kiểm chứng mô hình phù hợp với data( d = data - các điểm ngẫu nhiên )

output:

best\_model: các tham số mô hình tạo nên mô hình phù hợp nhất với data

best\_consensus\_set: Tập các điểm khớp với best\_model

best\_error: Tập các điểm outlier

iterations := 0

best\_model := nil

best\_consensus\_set := nil

best\_error := infinity

while(iterations < k){

maybe\_inliers := n chọn ngẫu nhiên n điểm từ data ban đầu.

maybe\_model là mô hình được tính toán từ tập maybe\_inliers đã chọn.

consensus\_set := maybe\_inliers

for(d){

nếu điểm đó có giá trị nhỏ hơn ngưỡng t thì thêm điểm đó vào tập consensus\_set

}

Nếu số lượng phần tử của consensus\_set > d (có nghĩa là đấy là mô hình tốt)

Thực hiện việc test model.

This\_model := là mô hình phù hợp với tất cả các điểm trong consensus\_set.

This\_error :=

If(This\_error < best\_error){

Ta đã tìm được mô hình tốt nhất kể từ khi lặp. Giữ cho đến khi tìm được mô hình tốt hơn.

best\_model := this\_model

best\_consensus\_set := consensus\_set

best\_error := this\_error

}

Interations++;

}

return best\_model, best\_consensus\_set, best\_error

1. SURF
2. Giới thiệu

SURF( *Speeded Up Robust Features* ) được giới thiệu năm 2006 bởi nhóm nghiên cứu của Herbert Bay. Ý tưởng của thuật toán này tương đối giống với SIFT. SURF cũng dùng không gian scale để tìm kiếm điểm đặc trưng và các đặc trưng được mô tả bởi các vector có kèm thêm hướng. Thuật toán này có hai phần chính là “phát hiện” điểm đặc trưng, và “mô tả” điểm đặc trưng.

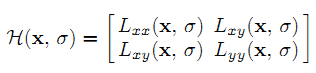
Phát hiện: Quá trình phát hiện điểm đặc trưng dựa trên không gian scale, và sử dụng ma trận Hessian ( chỉ sử dụng một xấp xỉ của ma trận này để đơn giản hơn trong việc tính toán). Các điểm đặc trưng là các điểm cực đại của hàm Hessian trong không gian scale. Người ta gọi bước này là phát hiện nhanh Hessian (*Fast Hessian Detector*).

Mô tả: mỗi điểm đặc trưng được thêm một hướng dùng để nhận dạng khi ảnh bị xoay, và được mô tả bằng một vector gồm 64 thành phần.

Cả hai bước trên đều sử dụng *Integral Images* để tăng tốc độ tính toán.

1. Phát hiện nhanh Hessian

Việc phát hiện đặc trưng của thuật toán SURF dựa vào ma trận Hessian, bởi vì đây là ma trận đảm bảo tốt về mặt thời gian cũng như độ chính xác. Trong phương pháp này, người ta có cho thêm một biến số scale σ vào ma trận Hessian để tạo không gian scale. Khi đó, với mỗi điểm X(x, y) của ảnh I, ma trận Hessian của X trong không gian scale là:



*Lxx(****x****, σ): đạo hàm bậc hai theo x của hàm Gaussian g(σ) tại điểm X của ảnh I*

*Lxy(****x****, σ) : đạo hàm bậc hai theo xy của hàm Gaussian g(σ) tại điểm X của ảnh I*

*Lyy(****x****, σ): đạo hàm bậc hai theo y của hàm Gaussian g(σ) tại điểm X của ảnh I*

Hàm Gaussian là tối ưu trong phân tích không gian scale, nhưng trong thực tế nhiều trường hợp hàm Gaussian cần được rời rạc hóa và chia tách. Áp dụng trong phương pháp này, để đơn giản người ta chỉ dùng một xấp xỉ cho đạo hàm bậc hai của hàm Gaussian.