**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---------\*\*\*---------

Báo cáo môn học

**Xử lý ảnh**

***Đề tài: Tìm hiểu về đặc trưng SIFT***

**Giáo viên hướng dẫn: TS. Hoàng Văn Hiệp**

**Sinh viên thực hiện :**

**1.** **Nguyễn Anh Tuấn Tùng - 20154247**

**2.** **Hoàng Hải Tú - 20154194**

**Hà Nội 06-2019**

Table of Contents

[1. Đặc trưng SIFT 3](#_Toc11848713)

[1.1. Giới thiệu 3](#_Toc11848714)

[1.2. Phát hiện các cực trị trong không gian tỉ lệ 4](#_Toc11848715)

[1.3. Định vị chính xác Keypoint: 8](#_Toc11848716)

[1.3.1. Loại bỏ các điểm tương phản thấp 9](#_Toc11848717)

[1.3.2. Loại bỏ các điểm biên 9](#_Toc11848718)

[1.4. Gán hướng 10](#_Toc11848719)

[1.5. Mô tả keypoint 11](#_Toc11848720)

[2. So khớp keypoint 14](#_Toc11848721)

[3. SURF 16](#_Toc11848722)

[3.1 Phát hiện điểm chú ý 16](#_Toc11848723)

[3.2 Trích xuất mô tả 16](#_Toc11848724)

[3.2.1 Định hướng điểm 16](#_Toc11848725)

[3.2.2 Mô tả dựa trên tổng số phản hồi sóng Haar 17](#_Toc11848726)

# 1. Đặc trưng SIFT

## 1.1. Giới thiệu

So khớp ảnh cho trước từ lâu vốn đã là một đề tài thiết yếu của lĩnh vực thị giác máy tính, bao gồm nhận diện vật thể hoặc khung cảnh, giải quyết các mô hình 3D từ nhiều ảnh và giám sát chuyển động…

SIFT (Scale-invariant feature transform) là một thuật toán phát hiện đặc trưng nhằm trích dẫn và mô tả các đặc trưng cục bộ của ảnh. Nó được nghiên cứu ở University of British Columbia, và được công bố bởi David Lowe vào 1999. Những đặc trưng được trích dẫn ra này là bất biến đối với thay đổi tỉ lệ và xoay ảnh, đồng thời chống lại khá tốt đối với thay đổi độ sáng và sự xoay nhẹ góc nhìn 3D (nhỏ hơn 50 degree). Những đặc trưng này được bố trí đồng đều ở cả miền không gian và miền tần số, giúp giảm thiểu tỉ lệ gây gián đoạn quá trình tìm kiếm ảnh bởi nhiễu đan xen và các vật che khuất.

SIFT, đặc biệt hoạt động như một thuật toán phát hiện đặc trưng (detector), đồng thời còn cho phép mô tả các đặc trưng đó (descriptor). Các đặc trưng được mô tả là vô cùng riêng biệt, nghĩa là nếu ta có một database rất lớn chứa các đặc trưng, thì khả năng tìm kiếm một đặc trưng trùng khớp với dữ liệu trong database đó là rất lớn.

SIFT được áp dụng trong nhận diện vật thể, điều khiển và định vị robot, làm mô hình 4D, nhận diện cử chỉ, giám sát chuyển động …

Các công đoạn chính của SIFT:

1. Phát hiện cực trị trong không gian tỉ lệ: Tìm kiếm trên toàn bộ các địa điểm và tỉ lệ của ảnh. Thuật toán sẽ sử dụng hàm Difference-of-Gaussian để tìm ra các điểm hấp dẫn tiềm tàng (interest point) mà là bất biến đối với tỉ lệ và chiều xoay.
2. Định vị chính xác Keypoint: Từ các vị trí tiềm tàng tìm ra ở bước 1, chúng ta sẽ tìm các keypoint. Các keypoint này chính là cực đại và cực tiểu của các ảnh Different-of-Gaussian mà ta đã dựng lên ở bước 1. Sau đó ta sẽ tính toán để chỉ chọn các keypoint ổn định, loại bỏ các điểm biên hoặc có độ tương phản yếu …
3. Gán hướng cho các keypoint: Với mỗi keypoint ta sẽ xác định cho chúng một hướng cố định. Các tính toán sau này sẽ được thực hiện dựa theo hướng đã được gán, giúp cho các đặc trưng này là bất biến đối với sự xoay ảnh.
4. Mô tả keypoint: Gradient cục bộ của ảnh sẽ được đo trong một vùng xung quanh các keypoint, ở tỉ lệ đã được chọn trước. Các keypoint này sẽ được mô tả bằng vector 128 chiều, rất mạnh khi chống lại các sự méo mó hình dạng cục bộ ảnh hay thay đổi độ sáng.

Đối với nhận diện ảnh, các đặc trưng SIFT được trích dẫn từ tập các ảnh mẫu, sau đó lưu vào database. Một ảnh mới sẽ được nhận diện bằng cách so sánh mỗi đặc trưng của ảnh mới này với các đặc trưng được lưu trữ, tìm kiếm các ứng cử viên trùng khớp với chúng dựa trên khoảng cách Euclide của các vector đặc trưng. Thuật toán nearest-neighbour được áp dụng để tìm kiếm các đặc trưng trùng khớp này một cách nhanh chóng.

## 1.2. Phát hiện các cực trị trong không gian tỉ lệ

Các vật thể trong thế giới thực đều chỉ có ý nghĩa ở một tỉ lệ nào đó. Ví dụ: Bạn có thể dễ dàng nhận thấy một cục đường nằm trên mặt bàn. Nhưng nhìn từ khía cạnh Ngân hà, cục đường đó chỉ đơn giản là … không tồn tại. Và không gian tỉ lệ (scale space) sinh ra nhằm mục đích mô tả khái niệm này trong lĩnh vực ảnh số.

Vậy scale space là gì? Để dựng lên scale space, ta sẽ lấy ảnh gốc để dần dần tạo ra các ảnh được lảm mờ. Đây là một ví dụ:

A picture containing mirror, bathroom, man, teeth

Description generated with very high confidence

*Mũ của mèo bị mờ dần, và râu ria của nó cũng vậy.*

Ta sẽ tìm kiếm các địa điểm trong ảnh mà là bất biến đối với sự thay đổi tỉ lệ, bằng cách lục tìm trên tất cả các tỉ lệ có thể, sử dụng chính hàm tỉ lệ scale space này.

Scale space của một ảnh, L(x,y,σ), được định nghĩa là tích chập của một ảnh đầu vào với toán tử làm mờ Gaussian G(x,y,σ):

L(x,y,σ) = G(x,y,σ)∗I(x,y)

Trong đó, G là:

A picture containing object

Description generated with high confidence

*Toán tử σ chính là độ làm mờ ta mong muốn (the scale operator)*

Để dựng lên scale space của ảnh, ta lấy ảnh gốc, dựng lên một chuỗi các ảnh bị làm mờ với độ làm mờ được nhân cấp tiến với hằng số k. Sau đó, ta resize ảnh còn một nửa bằng cách lấy mẫu xen kẽ theo hàng và cột (cứ một mẫu lấy thì bỏ một mẫu). Tiếp đến, ta lại làm mờ ảnh nhỏ này theo một chuỗi và lặp lai.

A picture containing cat, looking, indoor, bathroom

Description generated with very high confidence

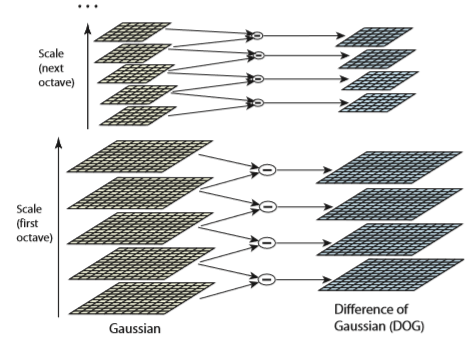
*Chuỗi các ảnh cùng kích cỡ được gọi là Octave. Trong một octave, các ảnh được làm mờ với các tỉ lệ khác nhau (lượng mờ được áp dụng).*

Để đạt được sự bất biến tỉ lệ một cách chính xác, nghiên cứu của Lindeberg (1994) đã chỉ ra rằng sự chuẩn hóa Laplacian of Gaussian với thừa số σ2 là cần thiết (Để ý toán tử σ2 dưới mẫu số của G, nếu ta loại bỏ được nó thì sẽ đạt được scale-invariant tuyệt đối!). Vậy nhưng, việc tính toán LoG là rất tốn kém (ta phải làm mờ ảnh bằng phép nhân chập với toán tử Gassian Blur như trên, sau đó tính toán đạo hàm bậc 2 của nó). Vậy nên ta sẽ chỉ áp dụng một phương pháp xấp xỉ, thay vì tính LoG của ảnh, ta tìm một tập các ảnh Different-of-Gaussian. DoG có thể được tính từ hiệu của 2 ảnh kế tiếp được khác nhau bởi tích số k lần độ làm mờ σ:

D(x,y,σ) = (G(x,y,kσ)−G(x,y,σ))∗I(x,y)

= L(x,y,kσ)−L(x,y,σ). (1)

Sau đó, ta lấy hiệu của mỗi 2 ảnh liền kề này để có được các ảnh DoG:



*Với mỗi octave của scale space, ảnh ban đầu được nhân chập với Gaussian Blur để tạo ra tập các ảnh scale space bên trái. Sau đó, các ảnh liên tiếp được trừ cho nhau để cho ra các ảnh DoG bên phải. Sau mỗi octave, ảnh Gaussian được lấy mẫu bằng ½ và lặp lại công đoạn.*

Hàm DoG cung cấp một cách ước lượng gần như chính xác đối với scale-normalized Laplacian of Gaussian, σ2∇2G, đã được chứng minh như sau:

Từ công thức cân bằng nhiệt:

A close up of a logo

Description generated with very high confidence

Bằng cách tính toán xấp xỉ ta lại có:

A screenshot of a cell phone

Description generated with high confidence

Vậy nên:

A close up of a mans face

Description generated with high confidence

Điều này cho thấy khi hàm DoG có tỉ lệ khác nhau bởi hằng thừa số k, nó đã tích hợp thêm tham số σ2 cần thiết cho sự bất biến tỉ lệ. Thừa số (k-1) là hằng số, nhưng sẽ không ảnh hưởng đến việc định vị cực đại và cực tiểu vì ta sẽ không kiểm tra các giá trị tại mỗi địa điểm. (nếu ta nhân toàn bộ ảnh với một hằng số, thì vị trí cực trị không thay đổi!).

**Phát hiện các cựu trị cục bộ:**

Để tìm ra các cự trị cục bộ của D(x,y,σ), mỗi một điểm mẫu sẽ được so sánh với 8 láng giềng gần nhất của nó trong ảnh hiện tại, cùng với 9 láng giềng ở tỉ lệ trên và 9 ở tỉ lệ dưới. Các điểm này là cực trị chỉ khi chúng lớn hoặc nhỏ hơn tất cả các láng giềng được so sánh. Quá trình này không tốn kém lắm bởi hầu hết các điểm không hợp lệ sẽ bị loại bỏ chỉ sau vài lần kiểm tra đầu tiên. Ảnh đầu tiên và ảnh cuối cùng sẽ không được tìm kiếm cực trị bởi không có đủ hàng xóm cần thiết để so sánh với chúng.

A close up of a logo

Description generated with very high confidence

Pixel đánh dấu X sẽ được so sánh với 26 láng giềng của nó.

Những điểm được đánh dấu sẽ là cực trị tương đối. Chúng là tương đối bởi cực trị không bao giờ nằm chính xác trong một pixel, mà sẽ nằm giữa các pixel với nhau. Vậy nên ở bước 2 chúng ta sẽ xác định các vị trí subpixel mà ở đó là cực trị.

## 1.3. Định vị chính xác Keypoint:

Từ các vị trí pixel tiềm tàng ta đã tìm thấy ở bước môt, giờ ta sẽ định vị chính xác các điểm cực trị. Ta sẽ tính toán khai triển Taylor của hàm scale space, D(x,y,σ), lấy vị trí của pixel ứng cử từ bước một làm gốc:

A picture containing object

Description generated with high confidence(2)

D và các đạo hàm của nó được tính toán tại điểm ứng cử và x = (x,y,σ)T là vị trí offset từ điểm này. Vị trí của cực trị thực sự, xˆ, được tính toán bằng cách lấy đạo hàm của hàm này và cho bằng không. Sau đó ta sẽ có:

A close up of a logo

Description generated with very high confidence(3)

Nếu offset xˆ lớn hơn 0.5 thì rất có thể cực trị nằm gần hơn ở một điểm mẫu khác. Trong trường hợp đó, điểm mẫu được thay đổi và việc nội suy được chuyển sang xung quanh điểm đó. Ngược lại thì offset xˆ được thêm vào vị trí của điểm ứng cử để thu được vị trí cực trị.

### 1.3.1. Loại bỏ các điểm tương phản thấp

Giá trị của hàm tại cực trị, D(xˆ), hữu dụng trong việc xác định và loại bỏ các cực trị mà có độ tương phản thấp. Thay (3) vào (2) ta thu được:

A close up of a logo

Description generated with very high confidence

Mọi cực trị với giá trị | D(xˆ)| nhỏ hơn 0.03 sẽ bị loại bỏ (giá trị pixel nằm trong đoạn [0,1].

### 1.3.2. Loại bỏ các điểm biên

Như ta đã biết, các điểm nằm dọc theo biên là không tốt đủ để được chọn làm đặc trưng. Để đạt được sự ổn định thì ta phải loại bỏ cả các điểm biên này.

Ý tưởng là tính toán 2 gradient ở keypoint, vuông góc với nhau. Dựa theo hình ảnh nằm xung quanh keypoint, 3 trường hợp có thể xảy ra:

* Đó là 1 vùng phẳng: Cả 2 gradient đều nhỏ
* Là một biên: Một gradient sẽ lớn (vuông góc với biên) và gradient còn lại sẽ nhỏ (dọc theo biên)
* Một góc: Cả 2 gradient sẽ lớn

Các góc là rất lý tưởng nên ta sẽ cho keypoint pass nếu cả 2 gradient đều lớn. Việc xác định này có thể thực hiện thông qua Hessian Matrix:

A close up of a logo

Description generated with very high confidence

Ta sẽ không cần tính cả 2 eigenvalue, bởi ta chỉ quan tâm đến tỉ lệ của chúng. Trace của H, Dxx + Dyy, cho ta tổng của 2 eigenvalue này, trong khi định thức của H, DxxDyy - Dxy2 cho ra tích của chúng. Giả sử eigenvalue nhỏ là b thì eigenvalue lớn hơn là a, với r = a/b là tỉ lệ của chúng. Qua tính toán ta có được:

Tr(H)2 / Det(H) = (r+1)2 / r

R phụ thuộc hoàn toàn chỉ vào tỉ số của các eigenvalue chứ không phải giá trị riêng của chúng. Đồng thời, khi r tăng thì R cũng tăng theo. Vậy nên, với giá trị ngưỡng rth, nếu R tại một keypoint lớn hơn (rth + 1)2 / rth thì keypoint đó sẽ bị loại bỏ. Lowe sử dụng giá trị rth = 10.

1.4. Gán hướng  
 Bằng việc gán một hướng cố định cho mỗi keypoint dựa trên đặc tính cục bộ của ảnh, thì keypoint descriptor sẽ được mô tả một cách tương đối dựa theo hướng cố định này, từ đó có thể nhận được sự bất biến về hướng xoay.

Đầu tiên thì ảnh đã được làm mờ Gaussian, L(x,y,σ), được sử dụng để mọi tính toán đều được thực hiện trong khuôn khổ bất biến tỉ lệ. Với một mẫu ảnh L(x,y) tại tỉ lệ σ, ta sẽ tính toán cường độ gradient m(x,y) và hướng θ(x,y) như sau:

A close up of a logo

Description generated with very high confidence

2 thông số này được tính toán cho mọi pixel xung quanh keypoint. Sau đó, một histogram được dựng lên.

Trong histogram này, 360o được chia thành 36 bin, mỗi bin 100. 2 thông số trên của mỗi pixel lân cận sẽ được thêm vào bin tương ứng, với giá trị được thêm vào có trọng số tương ứng với cường độ gradient của nó. Ví dụ: Tại 1 điểm ta đo được hướng gradient là 18.7590 thì nó sẽ được thêm vào bin 10-19o, và “lượng” được thêm vào tỉ lệ với cường độ gradient của nó. Ngoài ra, cửa sổ dùng để xác định vùng thu thập gradient quanh keypoint ta dùng Gaussian-weighted circular window với σ bằng 1.5 lần tỉ lệ của keypoint đó.

Sau khi ta đã xử lý hết các pixel xung quanh keypoint thì histogram này sẽ đạt được peak tại đâu đó. Ta sẽ lấy bin này làm hướng của keypoint.

Ngoài ra ta không chỉ lấy mỗi peak. Nếu tồn tại đỉnh nào lớn hơn hoặc bằng 80% giá trị của peak thì ta sẽ dựng lên một keypoint mới với cùng các giá trị x,y (địa điểm), σ (tỉ lệ) với keypoint sẵn có, chỉ có hướng là hướng của đỉnh nhỏ hơn này mà thôi.

Vậy nên việc xác định hướng có thể tách một keypoint thành nhiều keypoint.

Ví dụ:

A close up of a device

Description generated with high confidence

Trong histogram như trên, ta thấy đỉnh của nó ở tại 20-29o, vậy nên keypoint được gán hướng 3 (bin thứ 3). Nhưng quan sát ta thấy bin 31 cũng có giá trị vượt 80% giá trị đỉnh trên, vậy nên ta tạo một keypoint mới với cùng vị trí và tỉ lệ như keypoint đang xét và gán cho nó hướng 31.

## 1.5. Mô tả keypoint

Trong các công đoạn trước, ta đã gán vị trí, tỉ lệ và hướng cho mỗi keypoint. Các tham số này mô tả đặc điểm cục bộ của ảnh, vậy nên nó cung cấp sự bất biến đối với những tham số đó. Ở bước cuối cùng này, ta sẽ tạo ra các mô tả keypoint, hay còn gọi là vector đặc trưng 128 chiều, hoạt động như chữ ký của keypoint đó. Nghĩa là chúng có tính riêng biệt rất lớn, mạnh trong việc phân biệt các điểm đặc trưng trong một database vô cùng lớn, đồng thời bất biến một cách mạnh nhất có thể đối với các biến đổi còn lại: sự thay đổi cường độ sáng hoặc góc nhìn 3D.

Để thực hiện điều này, một cửa sổ kích thước 16x16 được áp vào xung quanh keypoint. Cửa sổ này được phân ra làm 16 cửa sổ 4x4 nhỏ hơn.

A screen shot of a social media post

Description generated with very high confidence

Trong mỗi cửa sổ 4x4, chúng ta tính toán cường độ gradient và hướng của chúng. Sau đó các hướng này được đặt vào một histogram gồm 8 bin:

A close up of a logo

Description generated with high confidence

Hướng nằm trong khoảng 0-44 độ được thêm vào bin thứ nhất. 45-89 thêm vào bin thứ 2 và cứ thế. Ngoài ra, khối lượng được thêm vào dựa trên cường độ của gradient (như mọi khi).

Nhưng điểm khác biệt là, khối lượng thêm vào bin còn dựa vào khoảng cách từ keypoint. Vậy nên các gradient càng xa keypoint thì lượng thêm vào càng nhỏ. Điều này có thể thực hiện bằng cách áp dụng một hàm Gaussian-weighting với σ giá trị bằng một nửa độ rộng cửa sổ mô tả. Ta nhân hàm Gaussian này với cường độ của hướng, và kết quả sẽ là: các cường độ càng xa trung tầm thì càng được làm nhỏ.

A picture containing photo, wall, indoor, monitor

Description generated with high confidence

Giải quyết xong với 16 pixel, chúng ta đã hoàn thành chuyển đổi 16 hướng hoàn toàn ngẫu nhiên vào 8 bin định trước. Chúng ta xử lý với toàn bộ 16 vùng 4x4, thu được tổng cộng 4x4x8 = 128 con số. Vậy nên ta nói các vector đặc trưng có 128 chiều.

Cuối cùng thì, vector đặc trưng được chỉnh sửa để giảm tác động của sự thay đổi cường độ sáng. Đầu tiên, vector được chuẩn hóa bằng cách chia cho độ dài của nó. Lí do là bởi, sự thay đổi độ tương phản của ảnh trong đó giá trị mỗi pixel được nhân với một hằng số cũng sẽ nhân gradient với hằng số đó, vậy nên sự thay đổi tương phản này có thể được loại bỏ bởi phép chuẩn hóa vector (chia cho độ dài vector mới thì ta vẫn sẽ thu được vector cũ). Còn sự thay đổi cường độ sáng trong đó một hằng số được cộng thêm vào các pixel sẽ không ảnh hưởng đến giá trị gradient, bởi chúng được tính từ độ chênh lệch giá trị các pixel. Vậy nên descriptor này là bất biến đối với sự thay đổi cường độ sáng.

Nhưng vấn đề ở đây là, các thay đổi phi tuyến về độ sáng cũng có thể xảy ra bởi sự bão hòa của camera hoặc các thay đổi cường độ sáng mà ảnh hưởng đến bề mặt 3D, khiến cho hướng bị lệch đi một khoảng. Ta có thể tránh điều này bằng cách làm giảm ảnh hưởng của các gradient có cường độ lớn: Giới hạn các giá trị của vector xuống còn 0.2 (các giá trị lớn hơn 0.2 được quy về 0.2), sau đó lại chuẩn hóa nó. Nghĩa là việc xem xét cường độ của các gradient lớn không còn quan trọng, mà thứ ta cần quan tâm là sự phân bố gradient theo các hướng.

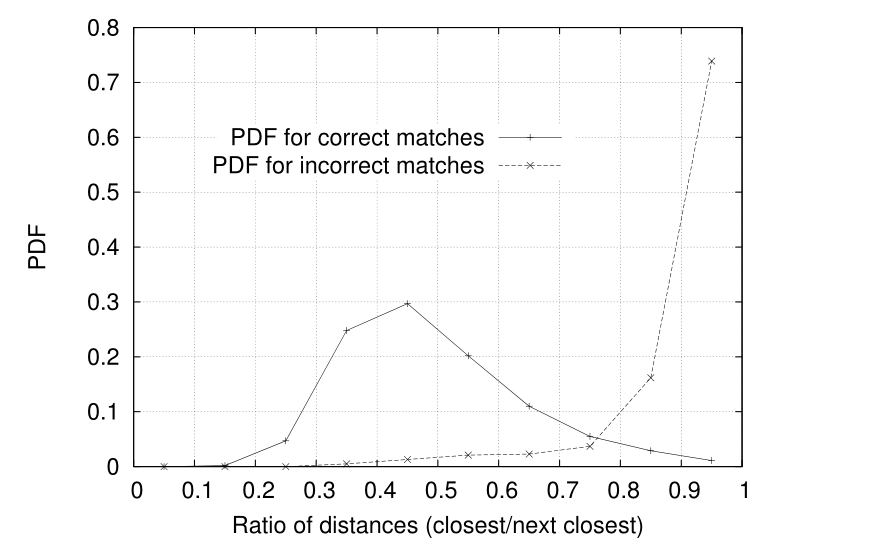
Sau khi có được các keypoint cùng với vector đặc trưng của chúng, ta có thể áp dụng vào nhiều lĩnh vực như nhận dạng ảnh hoặc giám sát chuyển động …

# 2. So khớp keypoint

Đối tượng so khớp tốt nhất của mỗi keypoint được xác định bởi điểm lân cận gần nhất trong dữ liệu keypoint thu được từ ảnh. Điểm lân cận gần nhất được định nghĩa là keypoint với khoảng cách Euclid tối thiểu cho vector mô tả bất biến tới keypoint đang xét.

Tuy nhiên, rất nhiều đặc tính của bức ảnh sẽ không có đối tượng so khớp chính xác vì chúng phát sinh từ lộn xộn nền (background clutter), hoặc là không được phát hiện. Do đó, sẽ là có ích nếu ta có thể loại bỏ các đặc tính mà không có đối tượng so khớp phù hợp trong dữ liệu một cách hiệu quả. Một ngưỡng toàn cục cho khoảng cách tới đặc tính gần nhất không hiệu quả, do một số vector mô tả rất khác biệt so với các vector mô tả khác.

Một cách đo đạc hiệu quả hơn, đó là so sách khoảng cách tới điểm lân cận gần nhất với khoảng cách tới điểm lân cận gần nhất thứ nhì. Nếu ta đang xử lý một đối tượng cụ thể trong một tấm ảnh và tìm ra được điểm lân cận gần nhất (so khớp chính xác), thì điểm lân cận gần nhất thứ nhì là điểm lân cận gần nhất tìm được sau quá trình xử lý một đối tượng khác với đối tượng đầu tiên (so khớp sai). Phương pháp đo đạc này thể hiện tốt, vì so khớp chính xác cần phải có điểm lân cận gần nhất gần hơn đáng kể so với so khớp sai.

  
*Hình: Xác suất khớp chính xác có thể được xác định bằng cách lấy tỷ lệ khoảng cách từ hàng xóm gần nhất với khoảng cách của lần thứ hai gần nhất. Sử dụng cơ sở dữ liệu gồm 40.000 điểm chính, đường liền nét hiển thị mật độ xác suất của tỷ lệ này đối với so khớp chính xác, trong khi đường chấm chấm là cho các kết quả so khớp sai.*

Hình vẽ cho thấy giá trị của phương pháp đo đạc này đối với dữ liệu hình ảnh thực. Các hàm mật độ xác suất cho các kết quả so khớp chính xác và sai được hiển thị theo tỷ lệ giữa lân cận gần nhất và gần nhất thứ nhì của mỗi keypoint. Các so khớp mà lân cận gần nhất chính xác có mật độ xác suất tập trung gần điểm có giá trị thấp hơn nhiều so với các so khớp không chính xác. Để triển khai nhận dạng đối tượng, ta loại tất cả các kết quả trong đó tỷ lệ khoảng cách lớn hơn 0,8, loại bỏ 90% các kết quả so khớp sai trong khi loại bỏ ít hơn 5% so khớp chính xác. Biểu đồ trên được tạo ra bằng cách khớp các hình ảnh sau khi thay đổi tỷ lệ và định hướng ngẫu nhiên, xoay độ sâu 30 độ và thêm 2% nhiễu hình ảnh, dựa trên cơ sở dữ liệu gồm 40.000 keypoint.

# 3. SURF

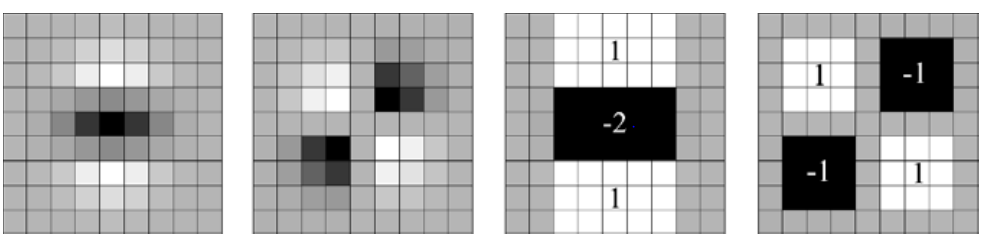
SURF, do Herbert Bay đề xuất, là một thuật toán lấy ý tưởng cơ bản từ SIFT, nhưng có những khác biệt nhất định. Giống như SIFT, SURF cũng có 2 tính năng là phát hiện điểm chú ý và trích xuất mô tả từ các điểm đó. Các đặc tả của SURF cũng có thể được dùng trong các mục đích tương tự SIFT như so khớp ảnh…

## 3.1 Phát hiện điểm chú ý

SURF sử dụng phương pháp tính xấp xỉ đạo hàm bậc 2 của Gaussian bằng bộ lọc 9×9 để tính nhanh định thức của ma trận Hessian của hàm Gaussian đối với từng điểm ảnh. Ma trận Hessian của ảnh I tại điểm ảnh x và scale δ là:

*H("x" ,δ)=[■(L\_xx ("x" ,δ)&L\_xy ("x" ,δ)@L\_xy ("x" ,δ)&L\_yy ("x" ,δ))], Lxx là đạo hàm bậc 2 của hàm Gaussian, tương tự với Lxy…*

Việc tính toán định thức của ma trận trên tiêu tốn khá nhiều tài nguyên và thời gian, do đó Bay và cộng sự đã đề xuất một cách tính xấp xỉ bằng cách đưa các đạo hàm về các bộ lọc số nguyên 9×9.



*Hình : 2 hình bên trái lần lượt là biểu diễn các đạo hàm Gaussian(Lyy và Lxy), 2 hình bên phải là các xấp xỉ của chúng.*

Sau khi tính toán xong, các định thức sẽ được tạo thành một bảng định thức. Sau đó kích thước của bộ lọc sẽ đươc tăng lên và thuật toán lặp lại vài lần để tạo thành 1 octave. Khác với sub-sampling của SIFT, SURF tăng kích thước các bộ lọc thay vì giảm kích cỡ ảnh để xử lý, do đó ảnh vẫn được giữ nguyên kích thước.

Với mỗi một scale của một octave, một khu vực lân cận 3×3×3 được xét với điểm trung tâm của khối sẽ là điểm được chọn nếu giá trị định thức tại ô đó là giá trị cực đai so với các điểm còn lại trong khu vực lân cận. Điểm này sau đó sẽ được nội suy để tìm ra vị trí điểm chú ý chính xác.

## 3.2 Trích xuất mô tả

### 3.2.1 Định hướng điểm

Để có được tính bất biến về góc xoay, định hướng của điểm chú ý cần được tim. Các phản hồi sóng Haar trên cả 2 phương ngang và dọc trong khu vực tròn xung quanh điểm chú ý với bán kính 6s (s là scale tại điểm chú ý đang xét) được tính toán và được đặt trọng số bởi một hàm Gaussian tại điểm chú ý đang xét, sau đó được chuyển thành các điểm trong hệ tọa độ 2 chiều, với phản hồi ngang là hoành độ và phản hồi dọc là tung độ.

Hướng chủ đạo sẽ được ước lượng bằng cách tính tổng của các phản hồi trong cửa sổ quay với độ lớn π/3, phản hổi dọc và ngang trong cửa sổ sẽ được cộng lại tương ứng, sau đó một vector hướng sẽ được dựng lên từ 2 tổng thu được. Vector lớn nhất sẽ là vector hướng chủ đạo. Kích thước của cửa sổ trượt là một tham số phải được lựa chọn cẩn thận để đạt được sự cân bằng mong muốn giữa độ mạnh và độ phân giải góc.

### 3.2.2 Mô tả dựa trên tổng số phản hồi sóng Haar

Để mô tả khu vực xung quanh điểm, một khu vực hình vuông được trích xuất, tập trung vào điểm chú ý và được định hướng dọc theo hướng như đã chọn ở trên. Kích thước của cửa sổ này là 20s.

Vùng chú ý được chia thành các tiểu vùng vuông nhỏ hơn 4×4 và đối với mỗi vùng, các phản ứng sóng con Haar được trích xuất tại vùng 5×5 các điểm mẫu. Các phản hồi được cân bằng với một hàm Gaussian (để mang lại sự mạnh mẽ hơn cho các biến dạng, tiếng ồn và biến đổi).