

TÌM HIỂU ĐẶC TRƯNG SURF

GVHD : TS. Hoàng Văn Hiệp

Sinh viên thực hiện: Nguyễn Thị Oanh 20163103
Nguyễn Hữu Tráng 20164196

NỘI DUNG

I. Tìm hiểu đặc trưng SURF

1. Giới thiệu đặc trưng SURF
2. Cài đặt thuật toán SURF trên MATLAB
3. Kết quả thử nghiệm

II. Kết luận

III. Tài liệu tham khảo

I. Tìm hiểu đặc trưng SURF

1. Giới thiệu đặc trưng SURF

- SURF (Speeded Up Robust Features) có nhiều ưu điểm và hiệu quả cao
- SURF lần đầu tiên được xuất bản bởi Herbert Bay
- SURF tìm ra các điểm đặc trưng cục bộ, bất biến đối với phép dịch (translation), phép xoay (rotation) và phép zoom-out (scaling) làm cơ sở cho đối sánh, nhận dạng ảnh.

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

- a. Chuyển ảnh đầu vào thành ảnh tích phân (Integral images)
- b. Tính toán Hessian matrix-based
- c. Xây dựng Scale-space
- d. Tìm các điểm đặc trưng (keypoints)
- e. Xác định chính xác điểm đặc trưng dựa trên nội suy lân cận
- f. Tính hướng tham chiếu cho mỗi điểm đặc trưng
- g. Xây dựng bộ mô tả điểm đặc trưng (keypoint descriptor)
- h. Thuật toán so khớp các keypoint (matching)

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

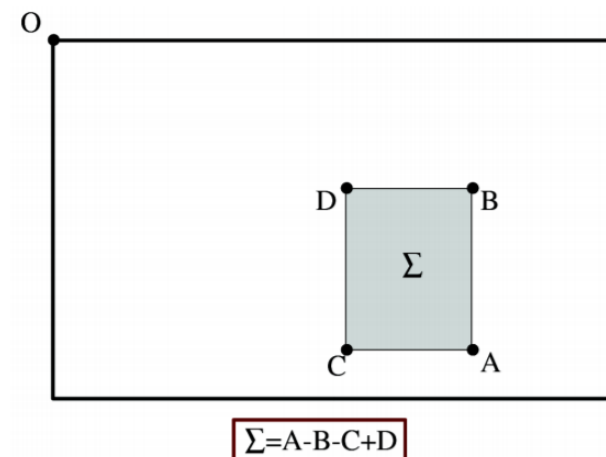
a. Chuyển ảnh đầu vào thành ảnh tích phân (Integral images)

Hình ảnh tích phân được sử dụng để tính tổng các giá trị (giá trị pixel) trong một hình ảnh nhất định

Cho phép tính nhanh các bộ lọc filter box.

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j)$$

$I_{\Sigma}(\mathbf{x})$ tại một vị trí $\mathbf{x} = (x, y)^T$ biểu thị tổng của tất cả các pixel trong hình ảnh đầu vào I trong một vùng hình chữ nhật được tạo bởi gốc và \mathbf{x} .



I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

b. Tính toán Hessian matrix-based

Cho một điểm $X = (x, y)$, ma trận Hessian $H(x, \sigma)$ tại X ở tỷ lệ σ được xác định là:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

Trong đó: $L_{xx}(X, \sigma)$ là tích chập của đạo hàm bậc hai Gaussian với ảnh I ở điểm X .

Tương tự: $L_{xy}(X, \sigma)$ và $L_{yy}(X, \sigma)$

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

b. Tính toán Hessian matrix-based

Để tính toán $\det(H)$, trước tiên chúng ta cần áp dụng tích chập với nhân Gaussian, sau đó là đạo hàm bậc hai.

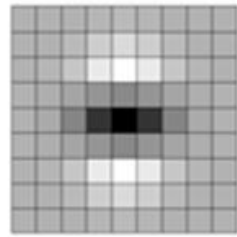
SURF lấy giá trị gần đúng (cả tích chập và đạo hàm bậc hai) bằng bộ filter box \Rightarrow chi phí thấp với việc sử dụng hình ảnh tích chập độc lập với kích thước (một phần lý do SURF nhanh)

1. Tìm hiểu đặc trưng SURF

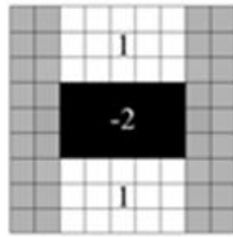
2. Cài đặt thuật toán SURF trên MATLAB

b. Tính toán Hessian matrix-based

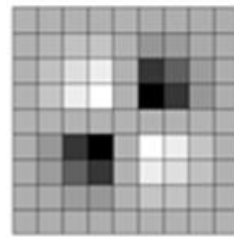
Box Filters 9x9 ~ sigma = 1.2



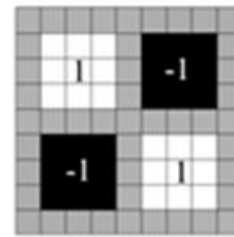
L_{yy}



D_{yy}



L_{xy}



D_{xy}

Biểu thị L_{xx} , L_{yy} , L_{xy} xấp xỉ D_{xx} , D_{yy} và $D_{xy} \Rightarrow \det(H)$ có thể biểu diễn (xấp xỉ) là:

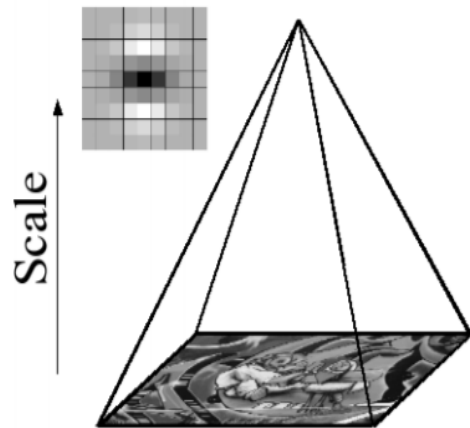
$$\det(\mathcal{H}_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2.$$

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \simeq 0.9,$$

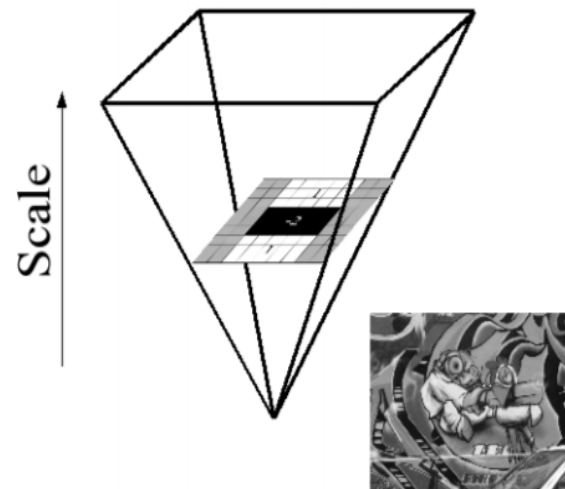
I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

c. Xây dựng Scale-space



SIFT: Giảm kích thước ảnh bằng cách làm mịn nhiều lần bằng gaussian rồi lấy mẫu một cách đệ quy



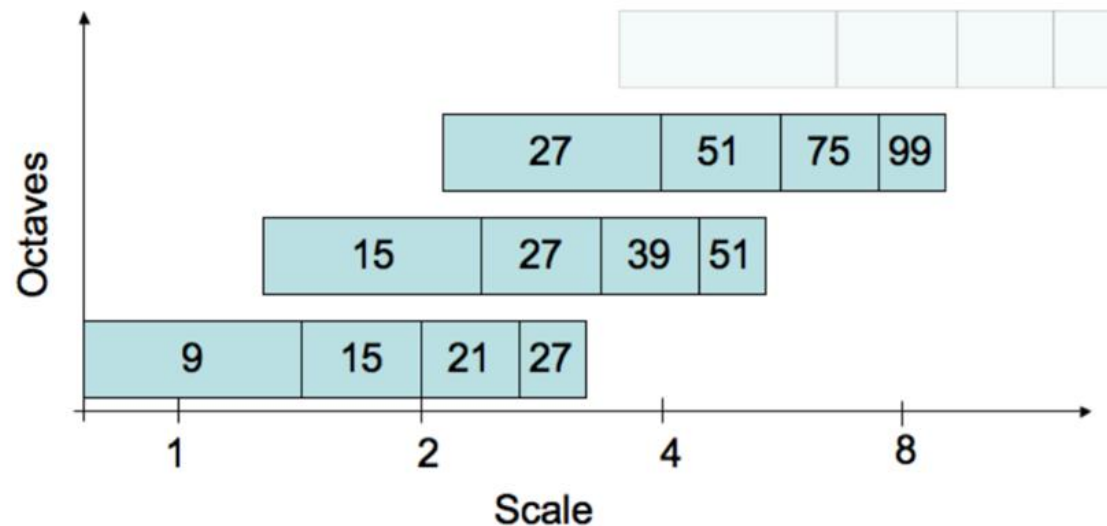
SURF: Tăng dần kích thước box filter và đồng thời tăng bước lấy mẫu trên các octave một cách song song, chi phí tính toán là constant

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

c. Xây dựng Scale-space

Với mỗi octave, thường sẽ dùng 4 bộ lọc box filters với kích thước tăng dần, minh họa:



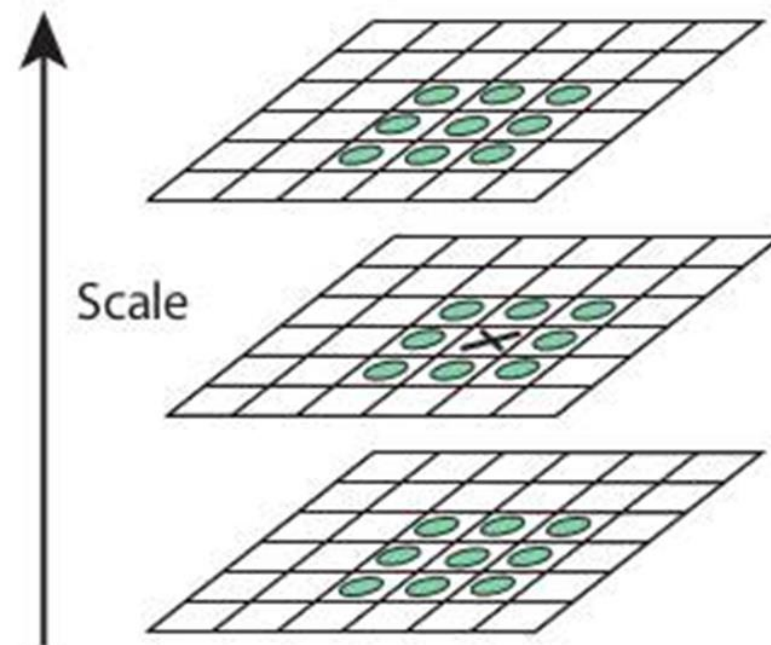
I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

d. Tìm các điểm đặc trưng (keypoints)

Duyệt qua từng octave:

- + Sử dụng 1 ma trận để đánh dấu vị trí các điểm đặc trưng
- + Tìm ra các điểm cực trị trong vùng 26 giá trị lân cận
- + Không xét scale đầu tiên và cuối cùng của mỗi octave, các điểm nằm trên biên ma trận
- + Chú ý: Trong mỗi octave ảnh có kích thước khác nhau nên cần tham chiếu về cùng tọa độ



I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

e. Xác định chính xác điểm đặc trưng dựa trên nội suy lân cận

- Sử dụng khai triển Taylor bậc 2 để nội suy lân cận.

- Ta sẽ tính cả Jacobian và Hessian để tính offset cho mỗi điểm đặc trưng tìm được ở mục d. Nếu offset lớn hơn 0.5 thì nó thực sự gần với một điểm hấp dẫn ở cấp pixel khác, cho nên sẽ bị loại bỏ đi

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

f. Tính hướng tham chiếu cho mỗi điểm đặc trưng

Để bất biến đối với phép quay, cố gắng xác định hướng các điểm key-point (điểm hấp dẫn).

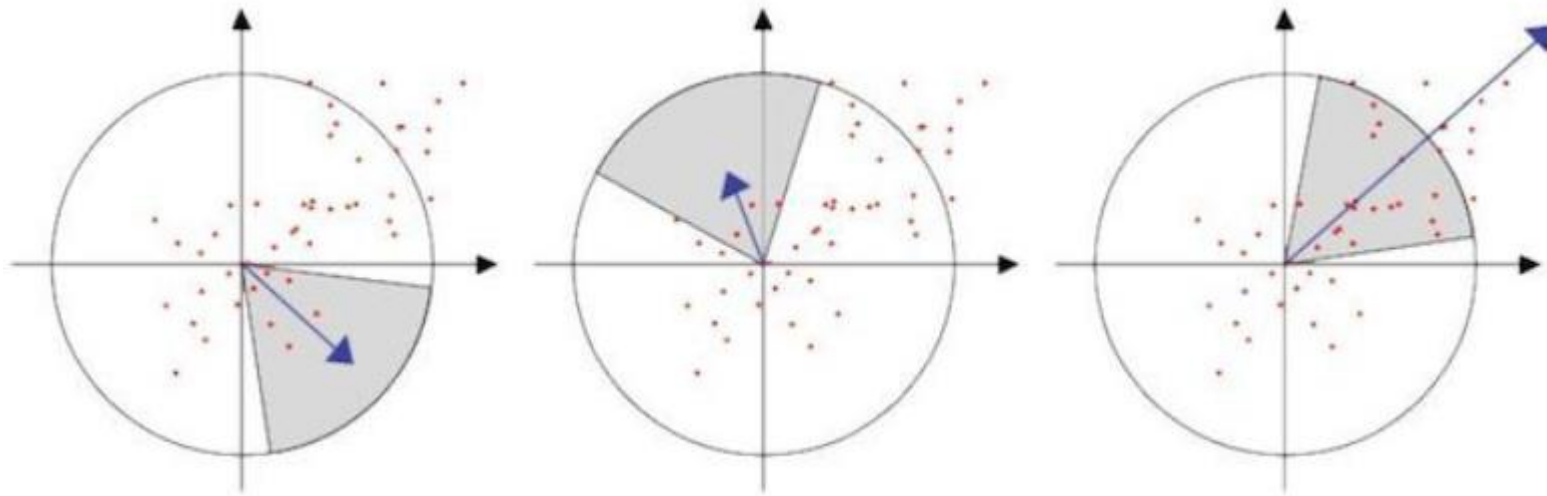
- + Tính toán phản hồi Haar-wavelet theo hướng x và y với vùng lân cận có bán kính $6s$ xung quanh điểm key-point, ở đây sử dụng ảnh tích hợp để tính nhanh

- + Tính tổng các phản hồi sóng theo x và y trong khu vực quét. Sau đó thay đổi hướng quét (thêm $\pi / 3$) và tính lại, cho đến khi tìm thấy hướng có giá trị tổng lớn nhất, hướng này là hướng chính của điểm key-point

I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

f. Tính hướng tham chiếu cho mỗi điểm đặc trưng

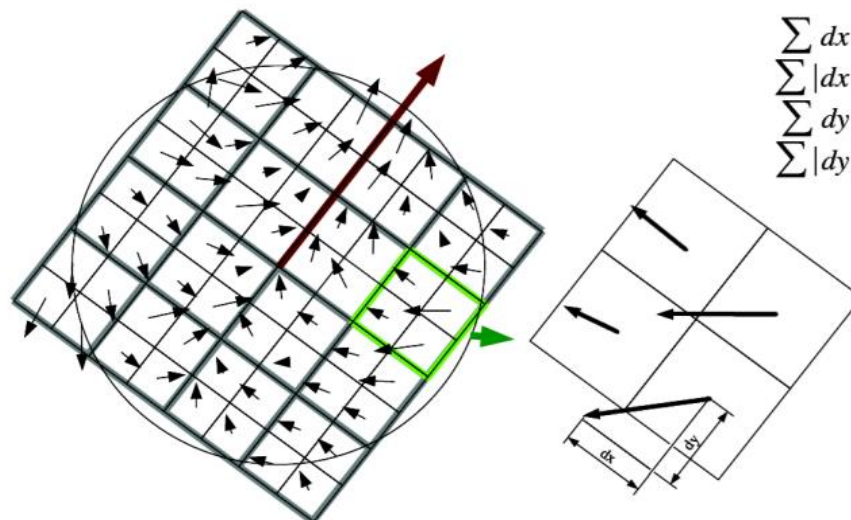


I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

g. Xây dựng bộ mô tả điểm đặc trưng (keypoint descriptor)

- Xây dựng một vùng hình vuông tập trung xung quanh điểm đặc trưng và được định hướng dọc theo hướng đã tính ở trên. Kích thước của cửa sổ này là 20s.

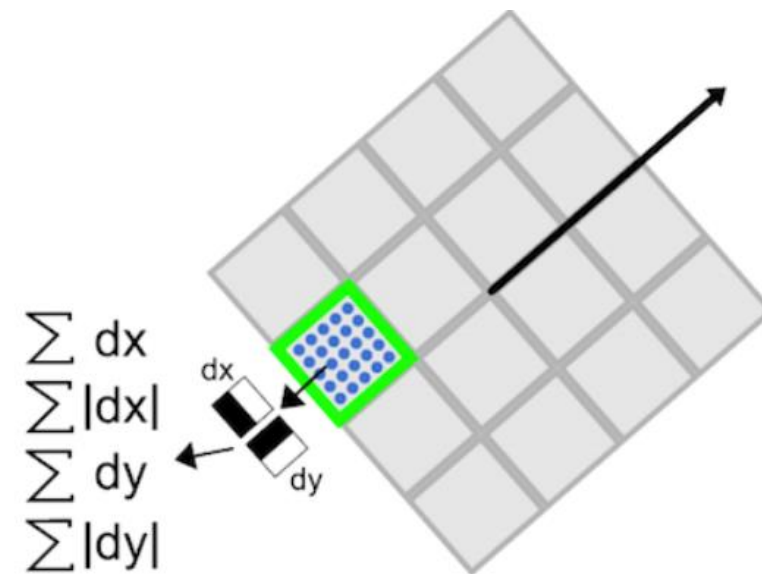


I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

g. Xây dựng bộ mô tả điểm đặc trưng (keypoint descriptor)

- Sau đó tính tổng các phản hồi dx và dy trong vùng con, ta được 2 trường đầu tiên trong vector mô tả. Để thể hiện thông tin về sự thay đổi cường độ, trích xuất thêm giá trị tuyệt đối của dx và dy và đưa vào vector mô tả \Rightarrow Mỗi vùng con 4×4 có 1 vector mô tả 4 chiều dạng $V = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$. Do đó, khi gộp tất cả các tiểu vùng lại ta sẽ có 1 vector mô tả 64 chiều.



I. Tìm hiểu đặc trưng SURF

2. Cài đặt thuật toán SURF trên MATLAB

h. Thuật toán so khớp các keypoint (matching)

- Tính khoảng cách Euclid giữa các điểm hấp dẫn tìm được ở 2 ảnh với nhau.
- Hiển thị 20 điểm khớp nhau nhất.

I. Tìm hiểu đặc trưng SURF

3. Kết quả thử nghiệm



I. Tìm hiểu đặc trưng SURF

3. Kết quả thử nghiệm



I. Tìm hiểu đặc trưng SURF

3. Kết quả thử nghiệm



I. Tìm hiểu đặc trưng SURF

3. Kết quả thử nghiệm



II. Kết luận

Sau một thời gian tìm hiểu và cài đặt thì kết quả nhóm đã cài đặt thành công thuật toán SURF. Kết quả khá khả quan trong đa số trường hợp. Tuy nhiên do thời gian có hạn nên còn nhiều thiếu sót, đặc biệt chưa xây dựng được ứng dụng hoàn chỉnh và thực tế nào. Nhóm mong nhận được sự đóng góp ý kiến từ thầy và các bạn để có thể cải tiến, hoàn thiện hơn

III. Tài liệu tham khảo

1. An Analysis of the SURF Method

link: https://www.ipol.im/pub/art/2015/69/article_lr.pdf

2. Introduction to SURF

link: <https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e>

3. Open SURF

link: <https://www.mathworks.com/matlabcentral/fileexchange/28300-opensurf-including-image-warp>

The image features a large, irregular blue ink splatter on a white background. The splatter has a textured, painterly appearance with various shades of blue and some darker spots. Centered within the blue area is the text "THANKS FOR YOUR WATCHING!" in a white, sans-serif, all-caps font. The text is arranged in two lines, with "THANKS FOR YOUR" on the top line and "WATCHING!" on the bottom line. The font is clean and modern, providing a strong contrast against the blue background.

THANKS FOR YOUR
WATCHING!