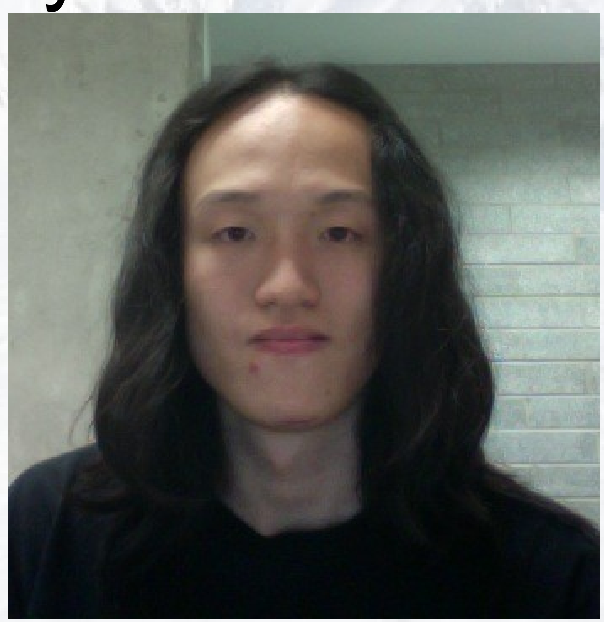
SURF: Speeded Up Robust Features

CRV Tutorial Day 2010

David Chi Chung Tam
Ryerson University

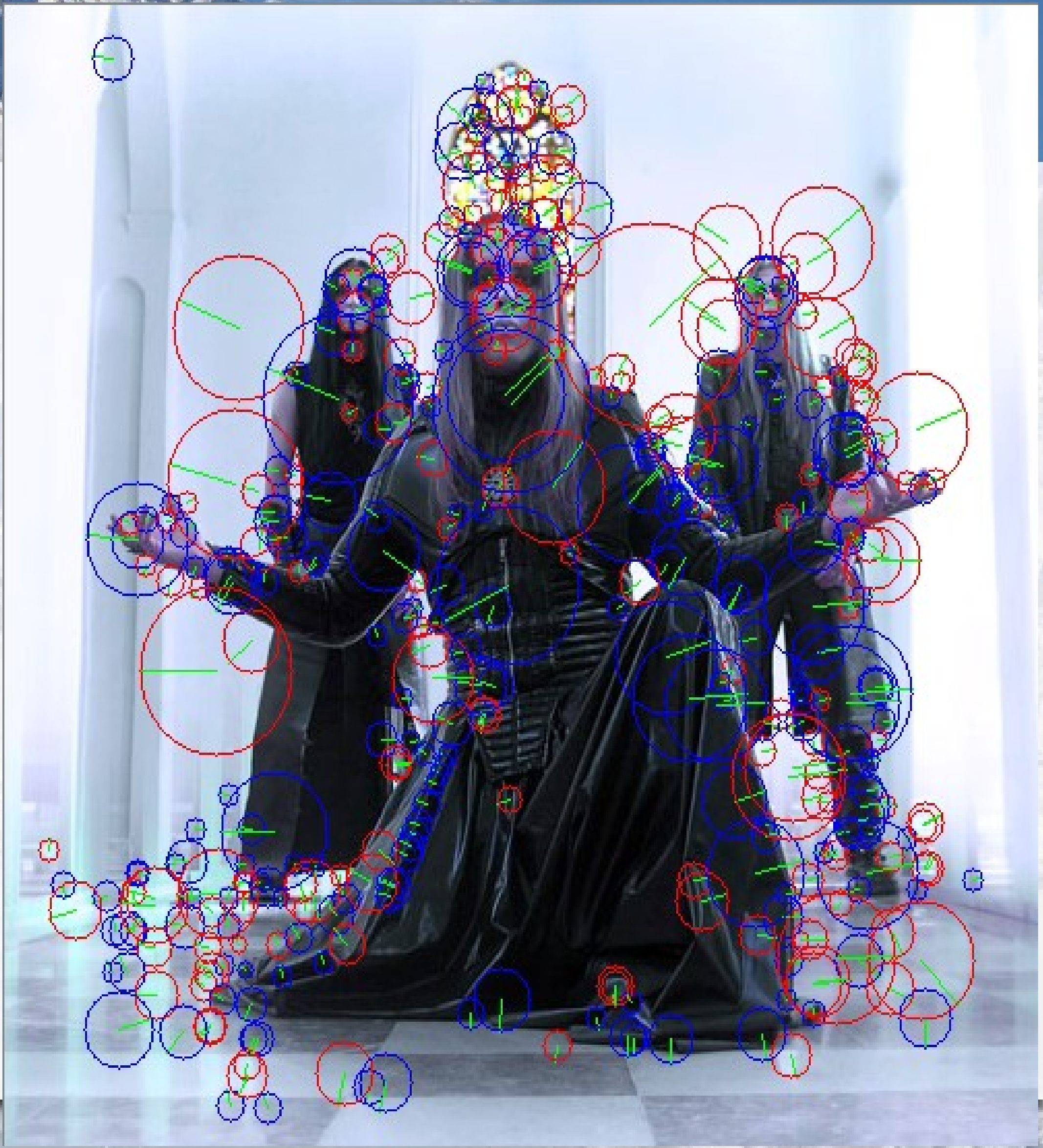


Goals of SURF

- A fast interest point detector and descriptor
  - Maintaining comparable performance with other detectors
  - High repeatability (reliability of finding same interest points under different viewing conditions)
- Builds upon concepts used in David Lowe's SIFT, but with better performance (according to authors of the SURF paper)
  - Tested in real-world applications

Example: Interest Points Detected

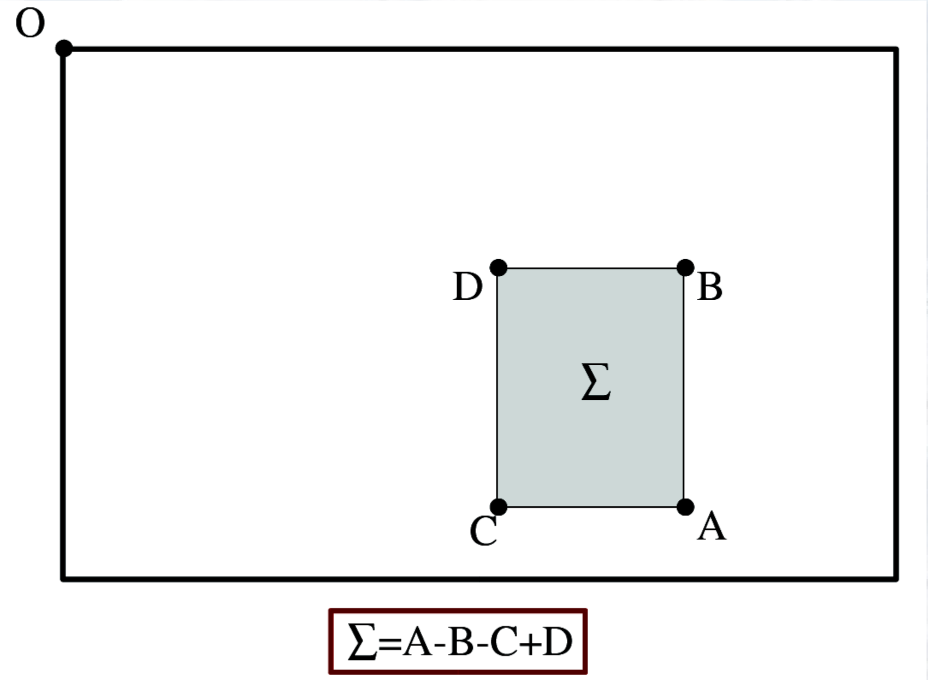
- OpenSURF example
- Circles are interest points detected
- Size of circles represent scales
- Green line: orientation
- Red: light on dark
- Blue: dark on light



Base image: behemoth.pl

Integral Images

- Reduces computation time significantly
- Calculate sum of pixel intensities in a rectangular region
- Only 3 additions needed:
  - Sum = A-B-C+D
- Calculation time independent of size



Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

Interest Points - Hessian Matrix

- SURF approximates the Hessian matrix
- Chosen for its good accuracy

Convolution of Gaussian second order derivatives with the image at point x $\frac{\delta}{\delta x^2} g(\sigma)$

Hessian Matrix

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

Hessian Matrix

- The Hessian matrix contains 2nd order derivatives – curvature =high values at 'hills' and 'valleys' (maxima and minima)

- Taylor Expansion:

Jacobian matrix

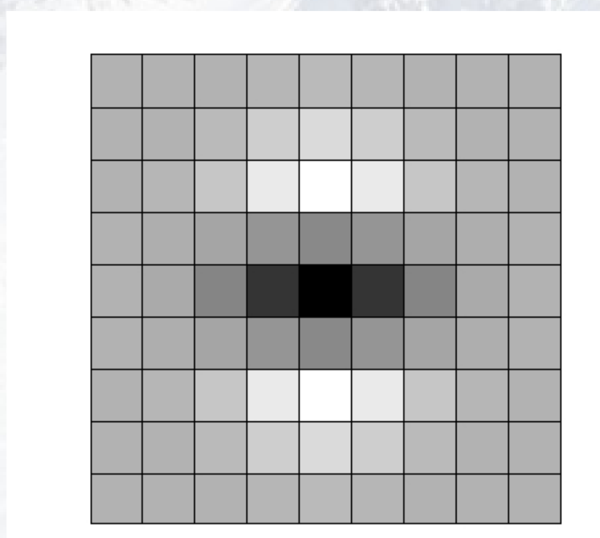
$$y = f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + J(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^{\mathrm{T}} H(\mathbf{x})\Delta\mathbf{x}$$

http://en.wikipedia.org/wiki/Hessian_matrix

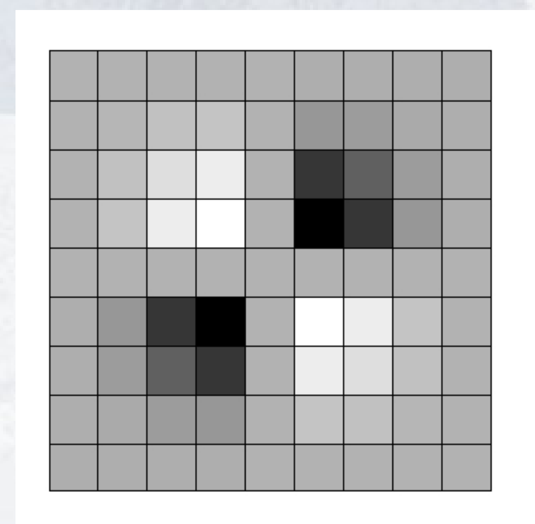
Gaussian Second Order Partial Derivatives

- Discretized and cropped for images
  - Replace f(x) with pixel intensities
- Computation costs increase as filter size increases

Y direction



XY direction

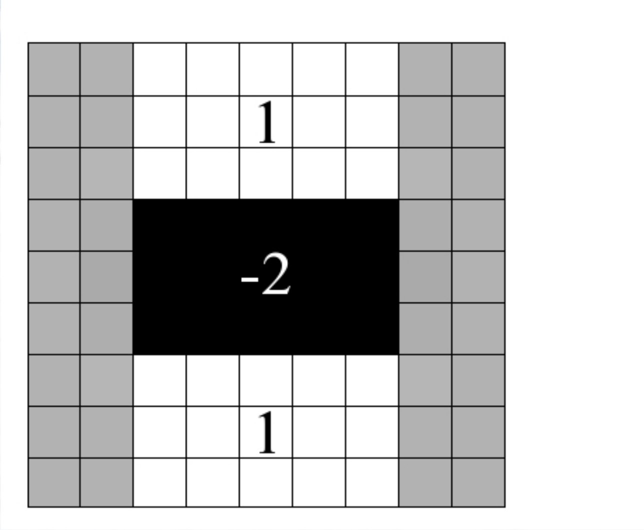


Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

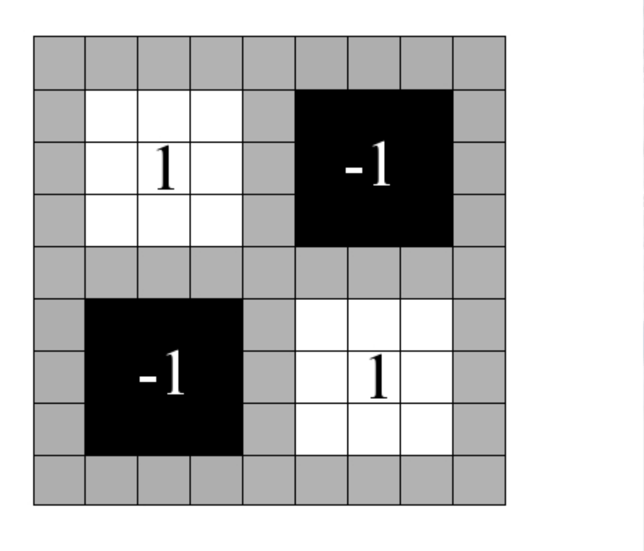
Approximation – Box Filters

- Box filters provide speed improvements far outweighing small performance decrease
- Filter size does not affect computational cost

Y direction



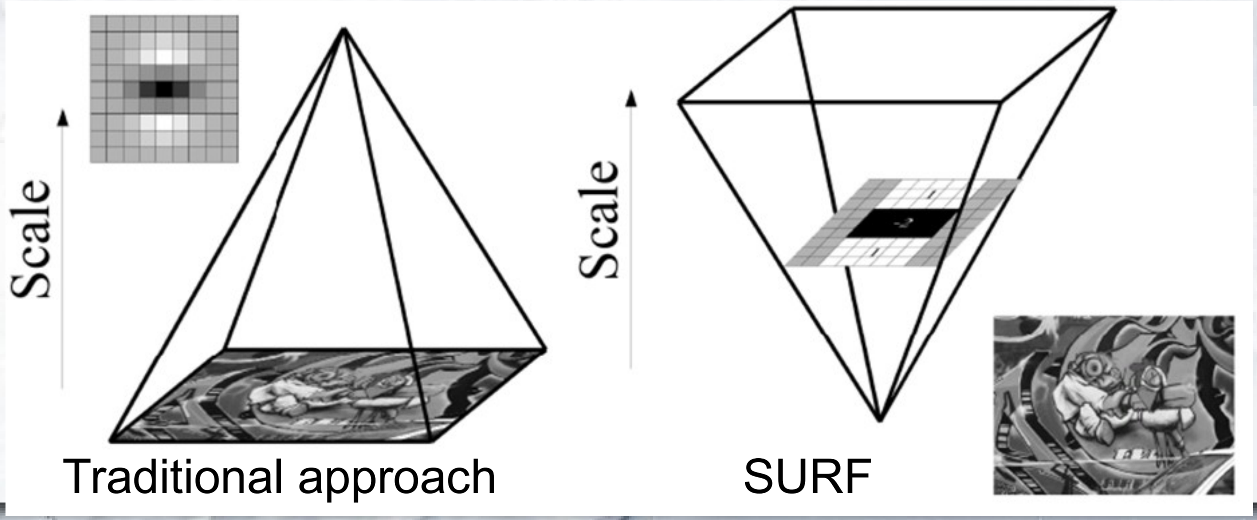
XY direction



Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

Scale Space

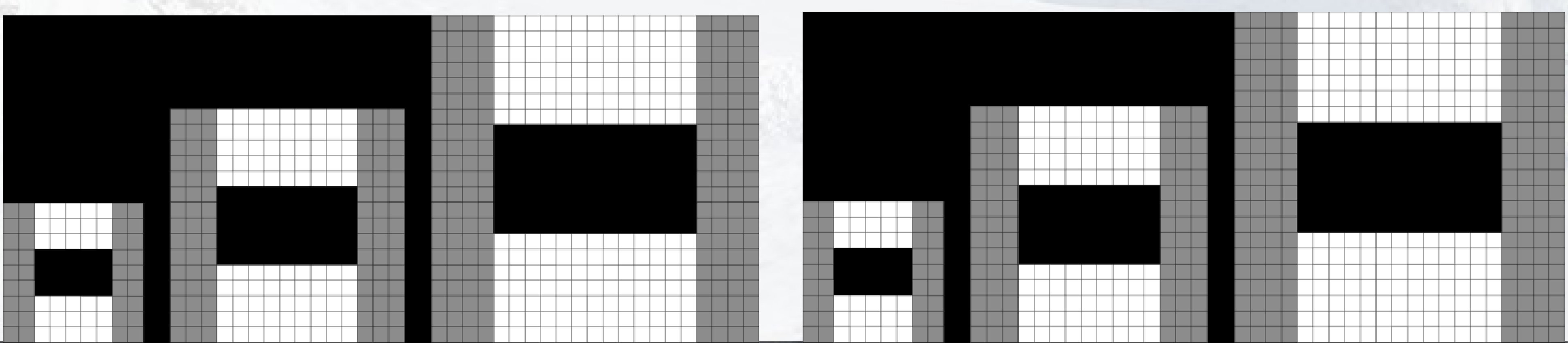
- A continuous function usable for finding extremas
- Using integral images allows scaling of filters without increasing computational cost



Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

Scale Space

- Divided into octaves – series of filter response maps with double increments on higher octave
- Begins with 9x9 filter, corresponding to $\sigma = 1.2$
- Increment of 6 or higher needed for preservation of filter structure



Chris Evans. Notes on the OpenSURF Library. January 18, 2009

Feature Descriptors

- Similar to SIFT (David Lowe)
- Distribution of intensity content within the interest point neighborhood
- First order Haar wavelet responses
  - Calculated in x and y direction instead of gradients
  - Use integral images to increase speed

Haar Wavelets

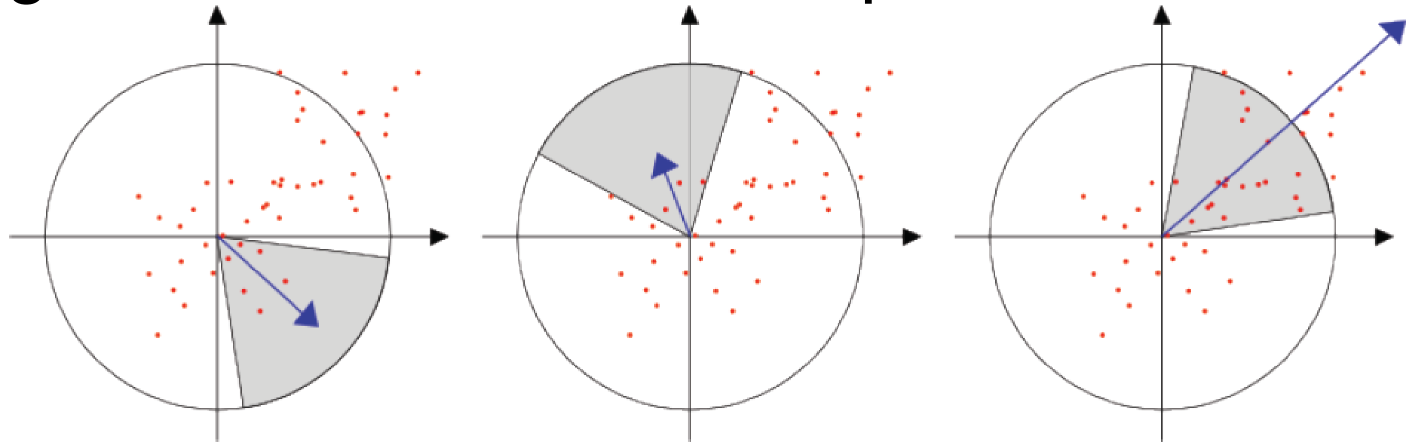
- Black: weight +1, White: weight -1
- Responses in x

- Responses in y

- For finding gradients

Orientation Assignment

- A window rotates around the origin that is 60 degrees wide
- Add up the responses within the window as the vector's length
- Longest vector is the interest point's orientation



Chris Evans. Notes on the OpenSURF Library. January 18, 2009

Some Implementations

- Official implementation (by authors of SURF paper)
  - closed source, non-commercial only
- OpenSURF by Chris Evans
  - Open Source, GPL, uses OpenCV
- OpenCV SURF
  - Included in OpenCV 2.0 and later - BSD
- GPU-based implementations
- etc.

References

- Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008
- Chris Evans. Notes on the OpenSURF Library. January 18, 2009 http://opensurf1.googlecode.com/files/OpenSURF.pdf
- http://en.wikipedia.org/wiki/Hessian_matrix