Feature Detection and Matching

簡韶逸 Shao-Yi Chien

Department of Electrical Engineering

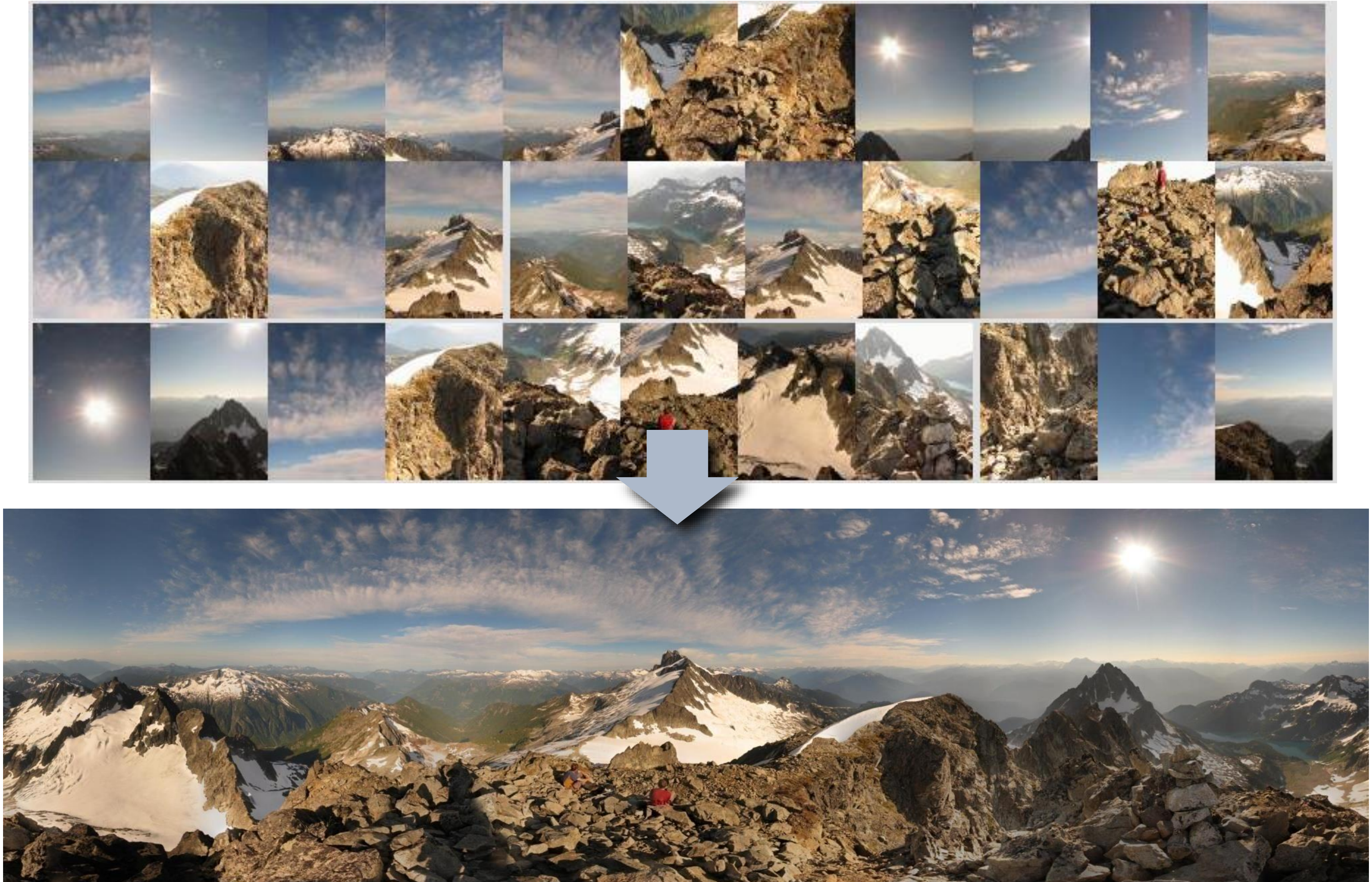
National Taiwan University

- References:
  - Slides from *Digital Visual Effects*, Prof. Y.-Y. Chuang, CSIE, National Taiwan University
  - Slides from *CE 5554 / ECE 4554: Computer Vision*, Prof. J.-B. Huang, Virginia Tech
  - Slides from *CSE 576 Computer Vision*, Prof. Steve Seitz and Prof. Rick Szeliski, U. Washington
  - Chap. 4 of *Computer Vision: Algorithms and Applications*
  - Reference papers

Outline

- The requirement for the features
- Points and patches
  - Feature detector
  - Feature descriptors
  - Feature matching
  - Feature tracking
  - SIFT
  - Applications
  - Recent features
- Edges and lines
- Appendix: MPEG-7 descriptors

How the Visual Cortex Represents the World?



Credit: Matt Brown

How the Visual Cortex Represents the World?

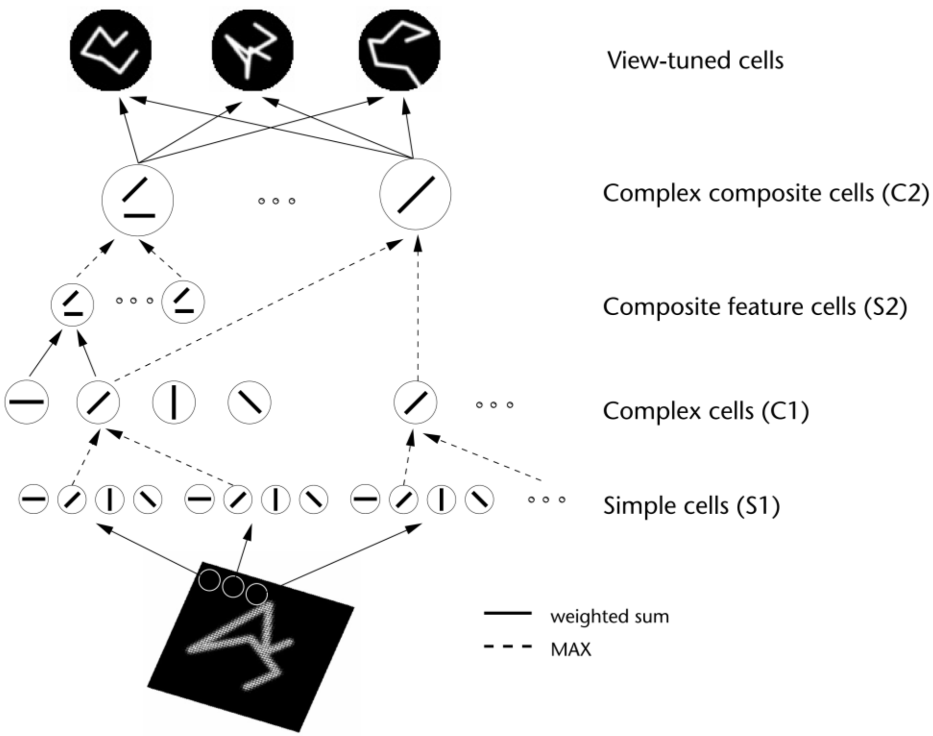
- The same place?

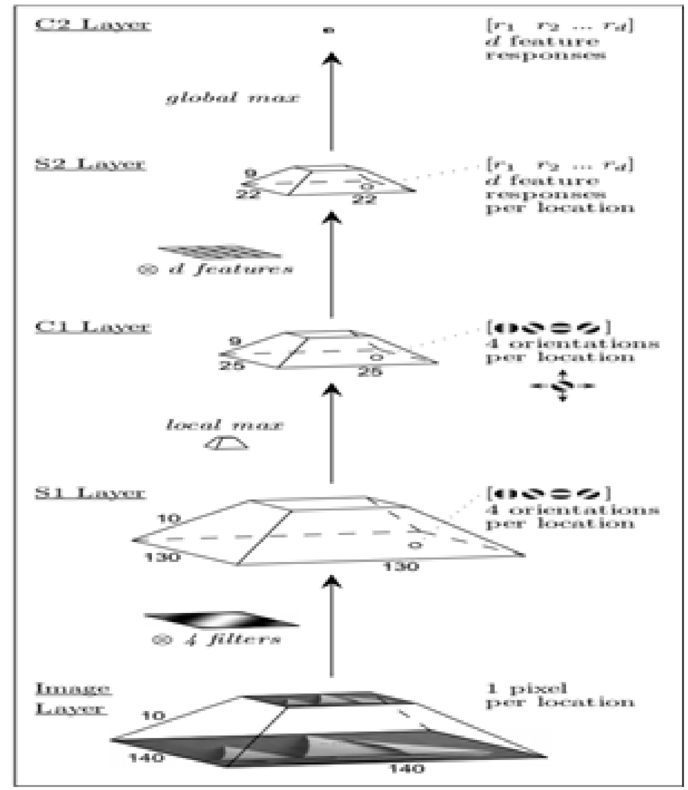
by Diva Sian

by scgbt

How the Visual Cortex Represents the World?

How the Visual Cortex Represents the World?





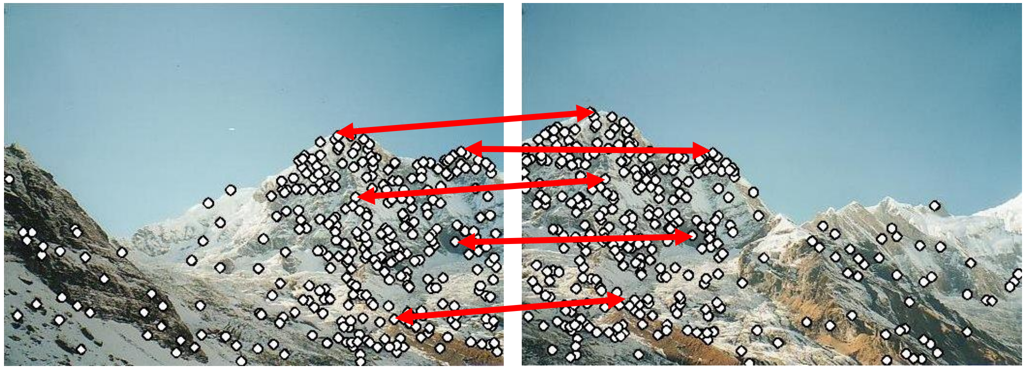
M. Riesenhuber and T. Poggio, “Why Can't a Computer be more Like a Brain?” *Nature Neuroscience*, vol. 2, no. 11, 1999.

The Requirement for the Features

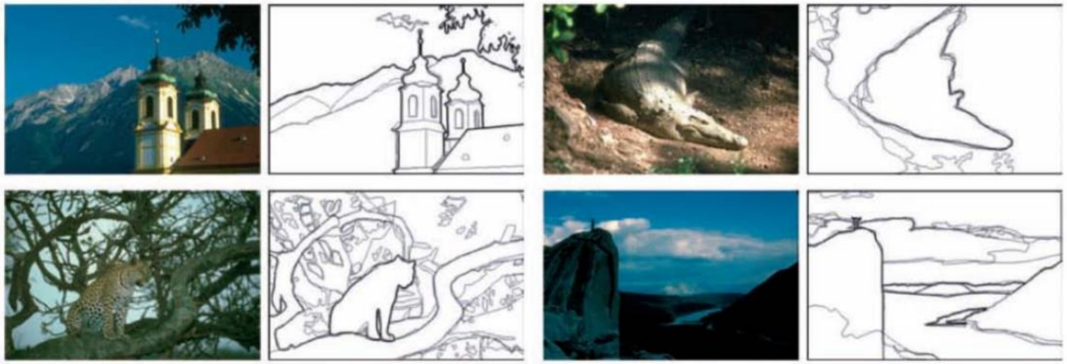
- We don’t make it by matching pixel values, but with some higher level information: **features**
- Requirements
  - **Invariant**: to lighting, color, rotation, scale, view angle…
  - **Locality**: features are local, so robust to occlusion and clutter (no prior segmentation)
  - **Distinctiveness**: individual features can be matched to a large database of objects
  - **Quantity**: many features can be generated for even small objects
  - **Efficiency**: close to real-time performance
  - …

Features

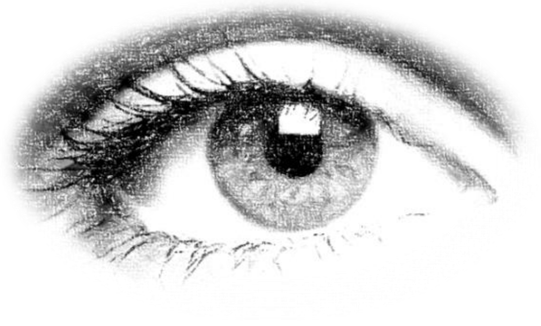
- Interest points



- Edge and lines



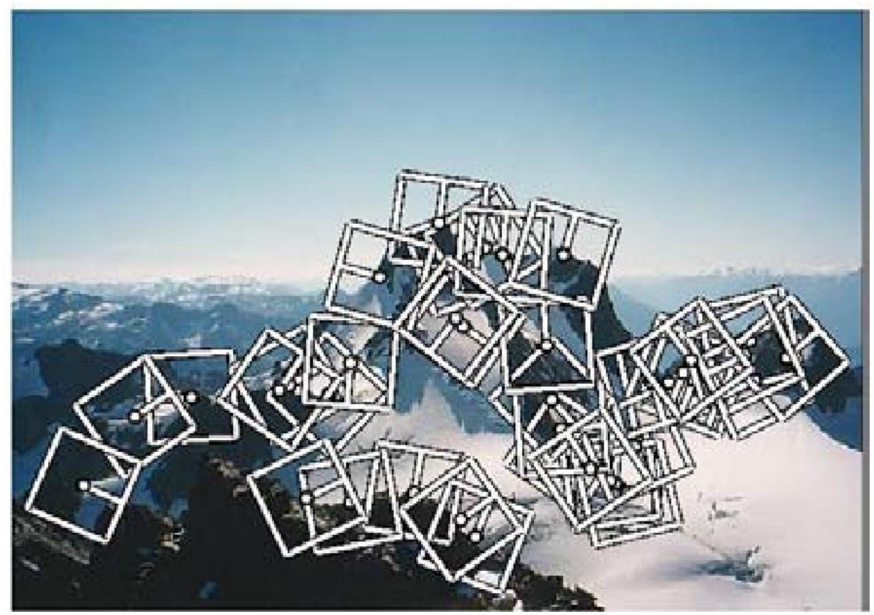
- Others



Points and Patches

Keypoint Detection and Matching Pipeline

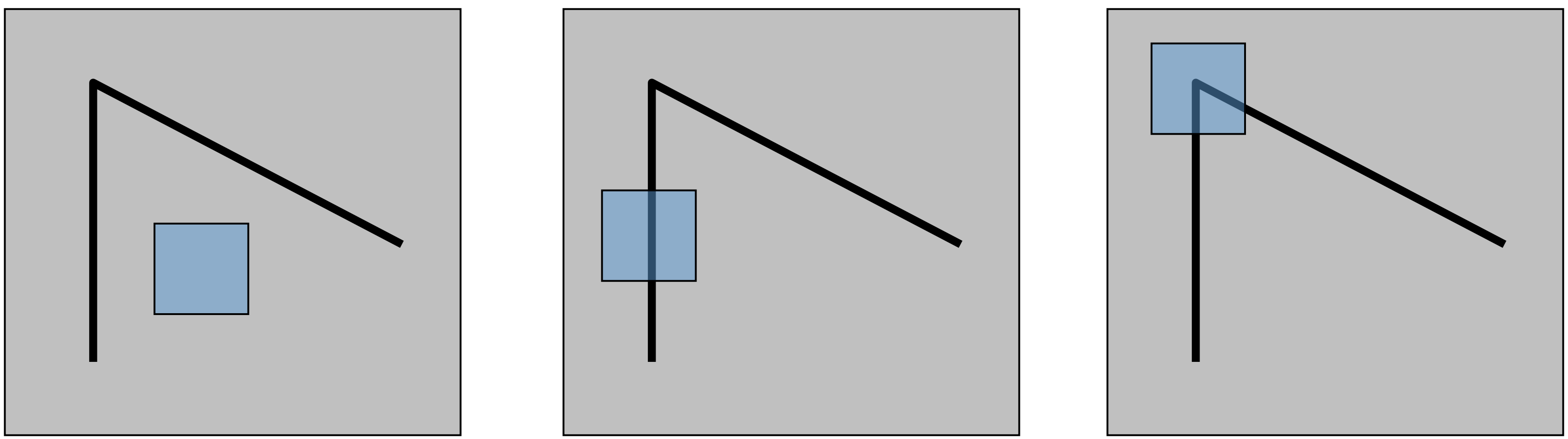
- Feature detection
- Feature description
- Feature matching
- Feature tracking



Feature Detection – Harris Corner Detector

Suppose we only consider a small window of pixels

- What defines whether a feature is a good or bad?

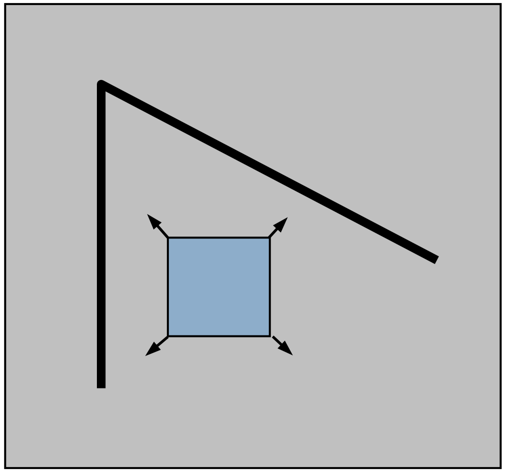


Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

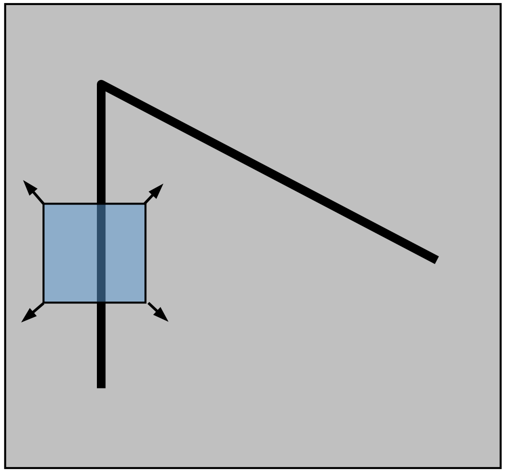
Feature Detection – Harris Corner Detector

Local measure of feature uniqueness

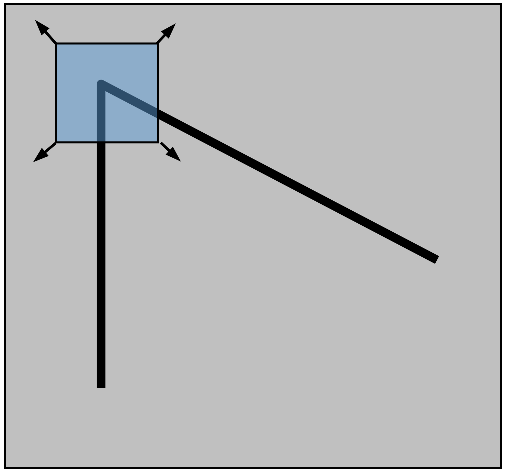
- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



"flat" region:
no change in all directions



"edge":
no change along the edge direction

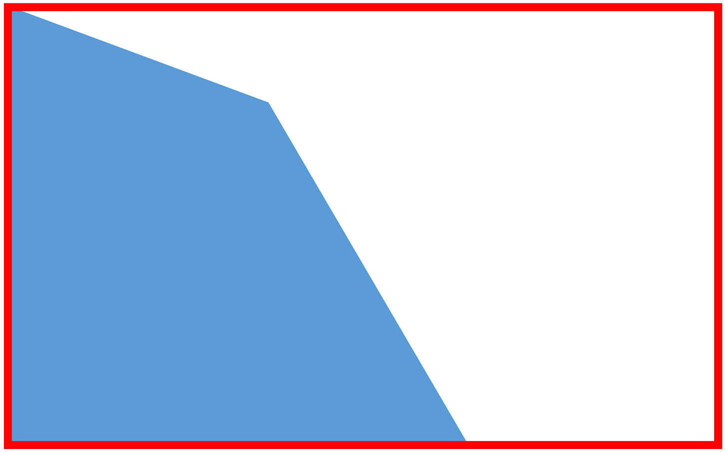


"corner":
significant change in all directions

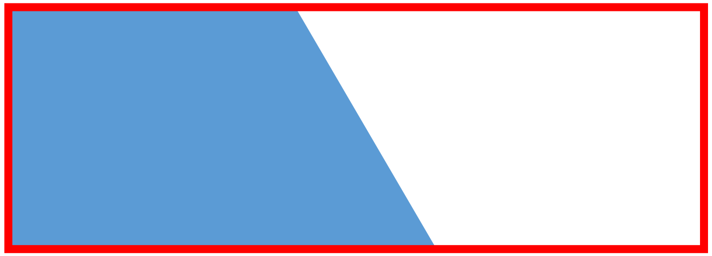
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

Aperture Problem

Aperture Problem



Aperture Problem



Feature Detection – Harris Corner Detector

- Change of intensity for the shift (*u*,*v*):

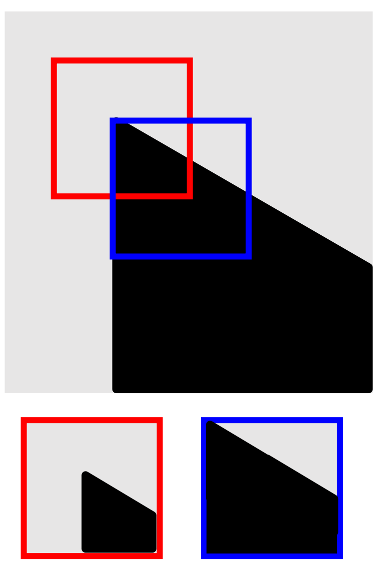
Auto-correlation function

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

window function

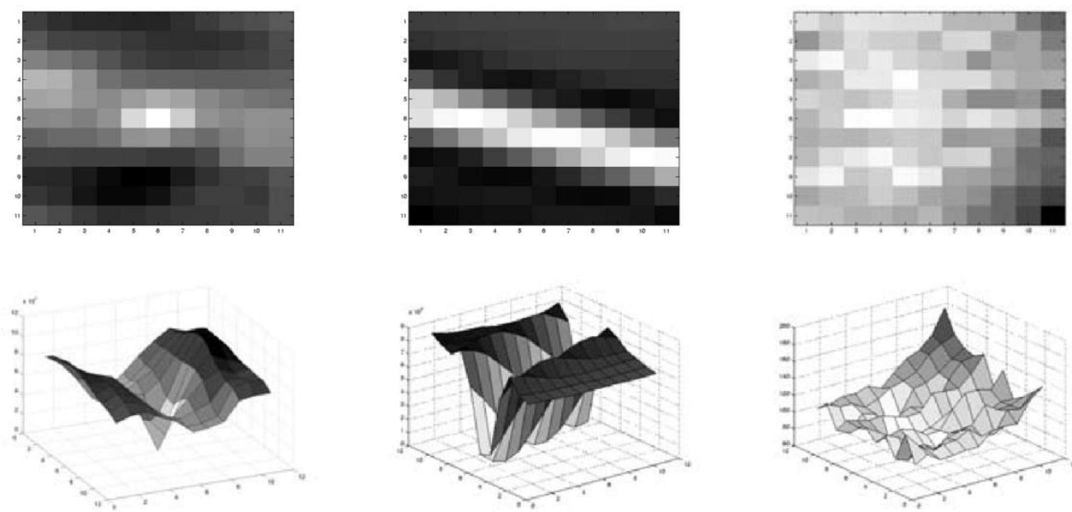
shifted intensity

intensity



$$w(x,y) = \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

Feature Detection – Harris Corner Detector



Strong Minimum

Strong Ambiguity

No Stable Minimum

E(u,v)

Feature Detection – Harris Corner Detector

- Small motion assumption → use Taylor Series expansion

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

$$= \sum_{x,y} w(x,y)[I(x,y) + \nabla I(x,y) \cdot (u,v) - I(x,y)]^2$$

$$= \sum_{x,y} w(x,y)\left[I_x(x,y)u + I_y(x,y)v\right]^2$$

$$= \sum_{x,y} w(x,y)\left[I_x^2\,(x,y)u^2 + I_y^2\,(x,y)v^2 + 2I_x(x,y)I_y(x,y)uv\right]$$

Feature Detection – Harris Corner Detector

$$E(u,v) = \sum_{x,y} w(x,y)\left[I_x^2\,(x,y)u^2 + I_y^2\,(x,y)v^2 + 2I_x(x,y)I_y(x,y)uv\right]$$

$$E(u,v) \cong \begin{bmatrix} u & v \end{bmatrix} \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix}$$

, where $\mathbf{M}$ is a 2×2 matrix computed from image derivatives:

$$\mathbf{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Eigenvectors of Symmetric Matrices

suppose $A \in \mathbf{R}^{n \times n}$ is symmetric, $i.e.$, $A = A^T$

fact: there is a set of orthonormal eigenvectors of A

$$A = Q\Lambda Q^T$$

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}$$

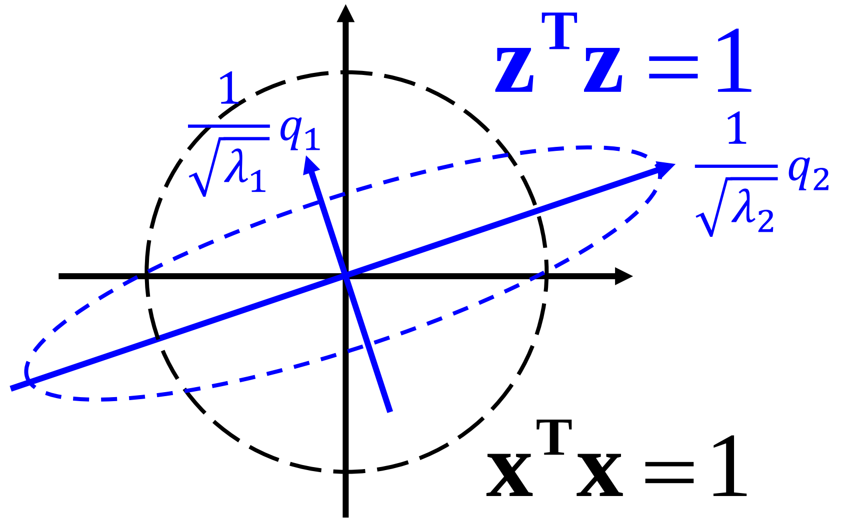
$$= \mathbf{x}^{\mathrm{T}}\mathbf{Q}\,\mathbf{\Lambda}\,\mathbf{Q}^{\mathrm{T}}\mathbf{x}$$

$$= \left(\mathbf{Q}^{\mathrm{T}}\mathbf{x}\right)^{\mathrm{T}}\mathbf{\Lambda}\left(\mathbf{Q}^{\mathrm{T}}\mathbf{x}\right)$$

$$= \mathbf{y}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{y}$$

$$= \left(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{y}\right)^{\mathrm{T}}\left(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{y}\right)$$

$$= \mathbf{z}^{\mathrm{T}}\mathbf{z}$$

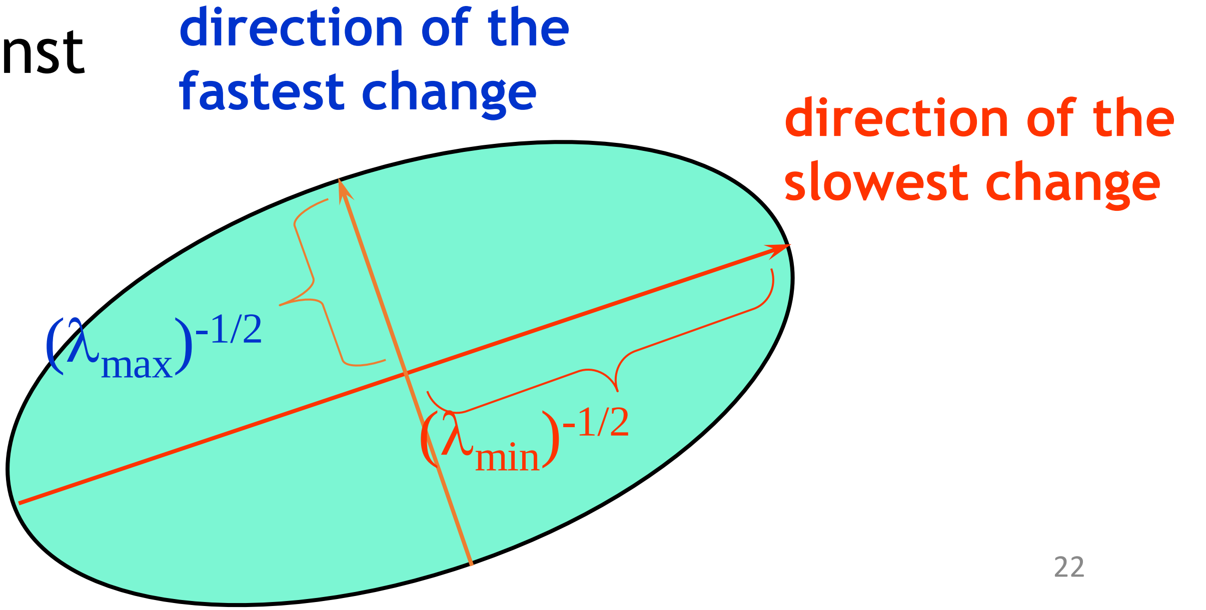


Feature Detection – Harris Corner Detector

Intensity change in shifting window: eigenvalue analysis

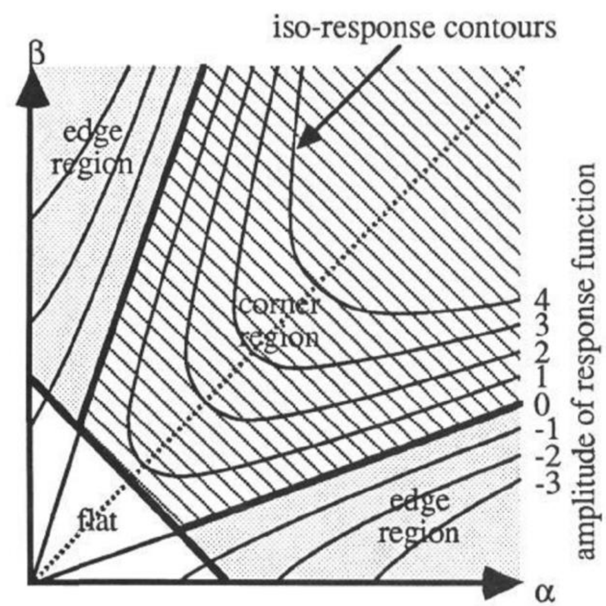
$$E(u,v) \cong [u,v]\,\mathbf{M}\begin{bmatrix} u \\ v \end{bmatrix} \qquad \lambda_1, \lambda_2 \text{ – eigenvalues of } \mathbf{M}$$

Ellipse *E(u,v)* = const



Feature Detection – Harris Corner Detector

- Feature scoring function
  - $\det(M) - \alpha \cdot \mathrm{trace}(M)^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$
  - $\lambda_0 - \alpha\lambda_1$
  - $\dfrac{\det M}{\mathrm{trace}\, M} = \dfrac{\lambda_0\lambda_1}{\lambda_0 + \lambda_1}$

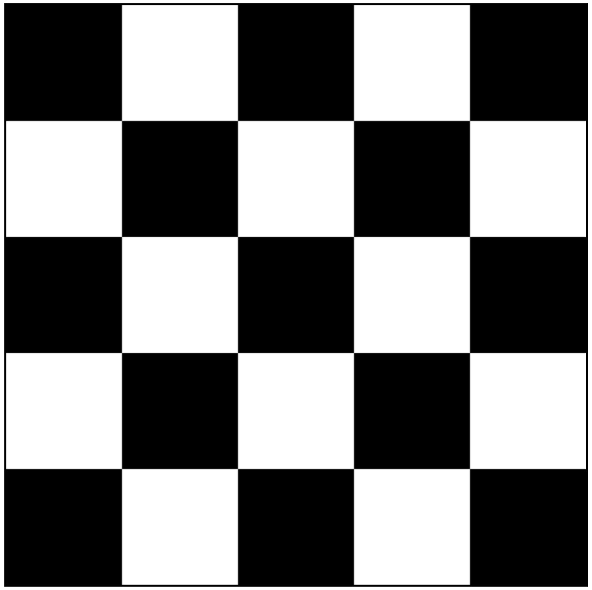


Ref: C. Harris and M.J. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vision Conference*, 1988.

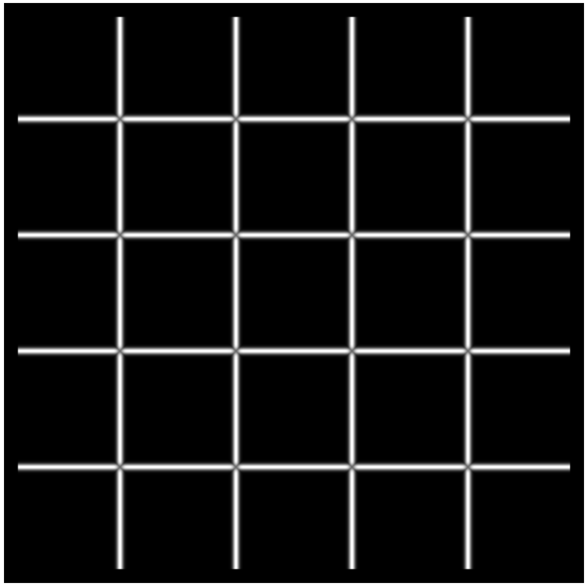
Feature Detection – Harris Corner Detector

Whole feature detection flow:

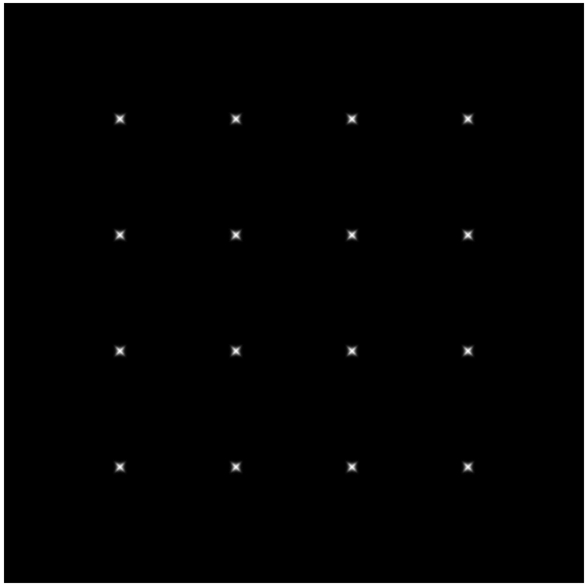
- Compute the gradient at each point in the image
- Create the ***M*** matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large Feature scoring function



I



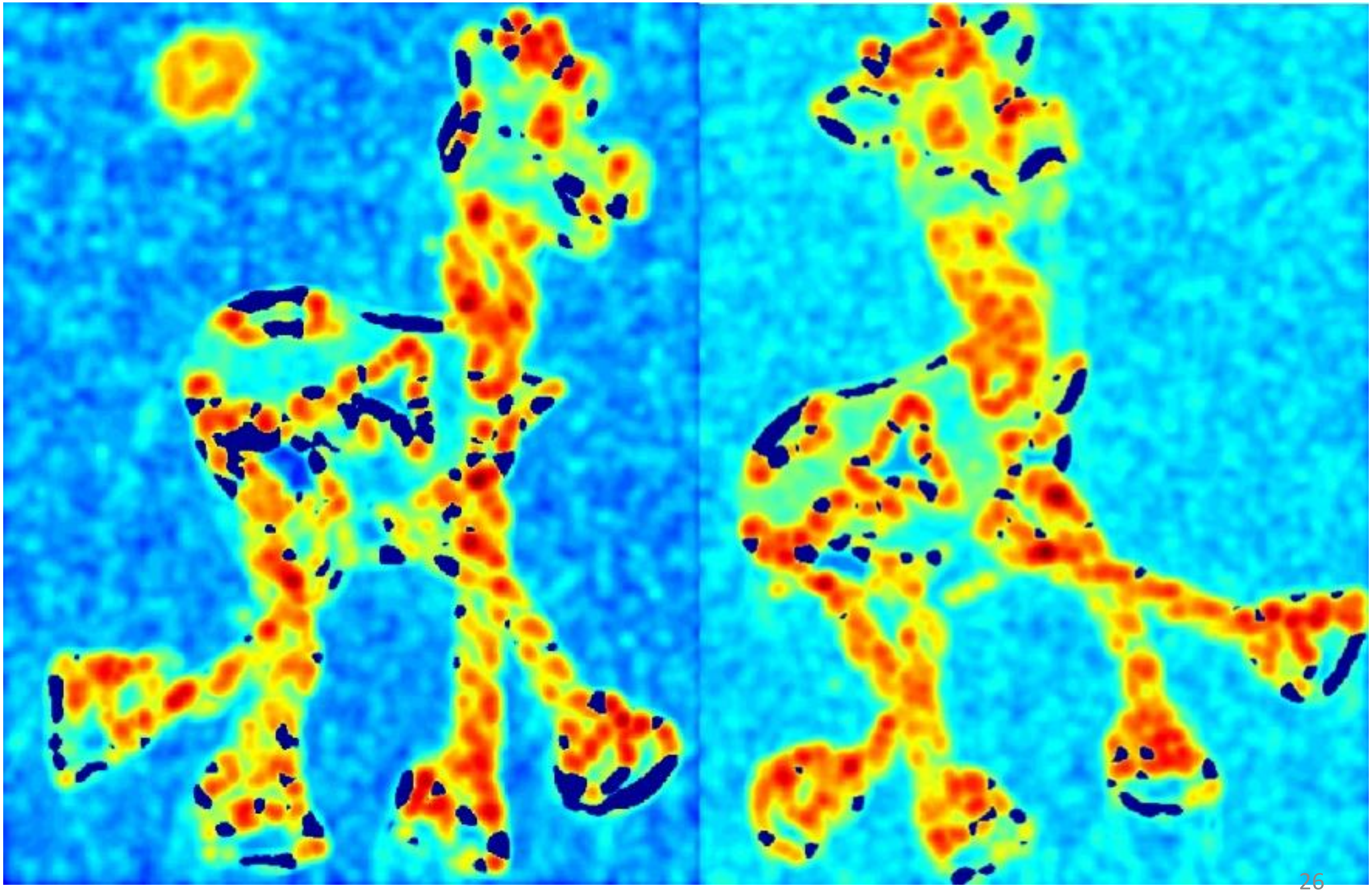
λ_0



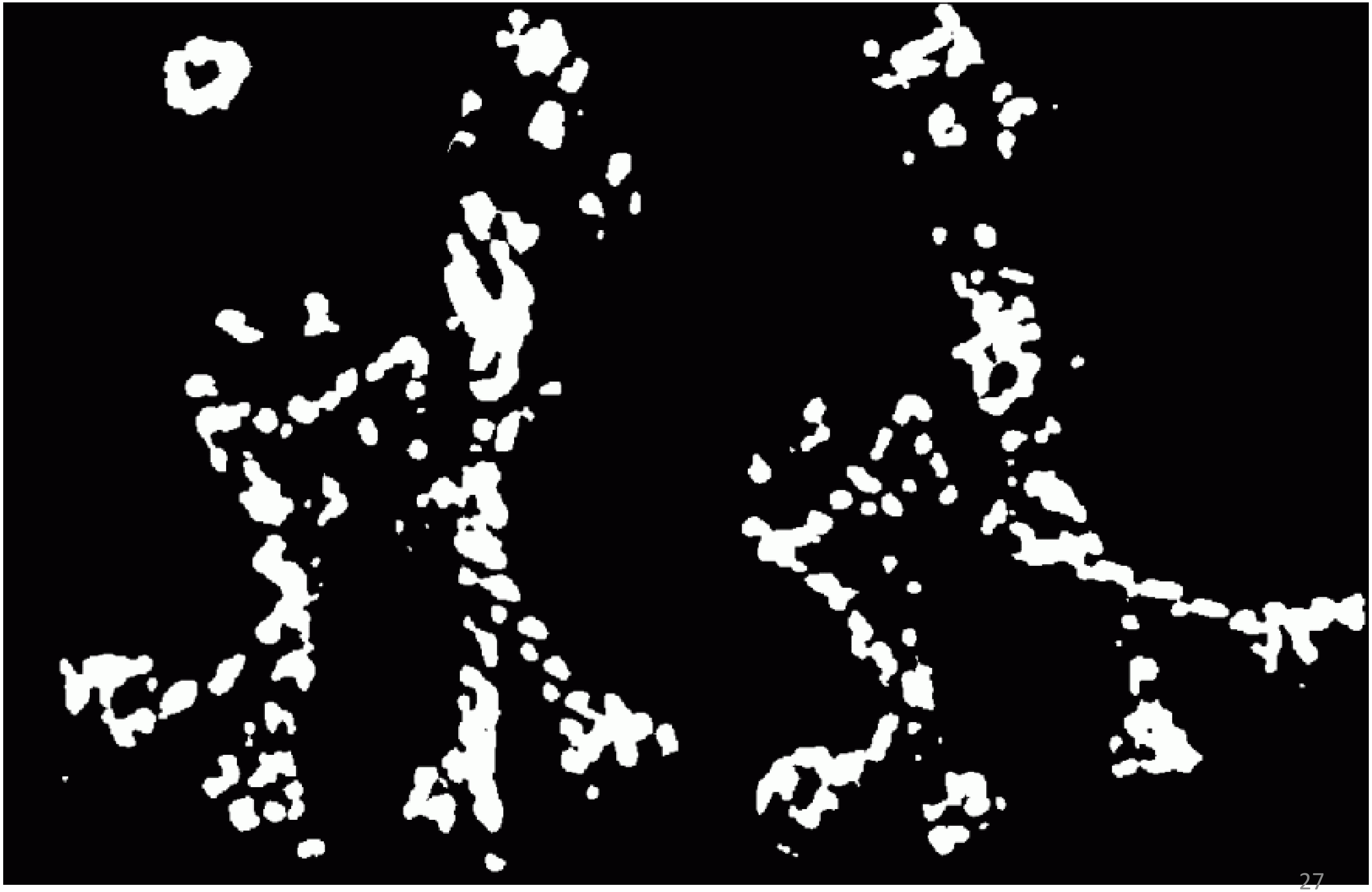
λ_1

Harris Detector Example

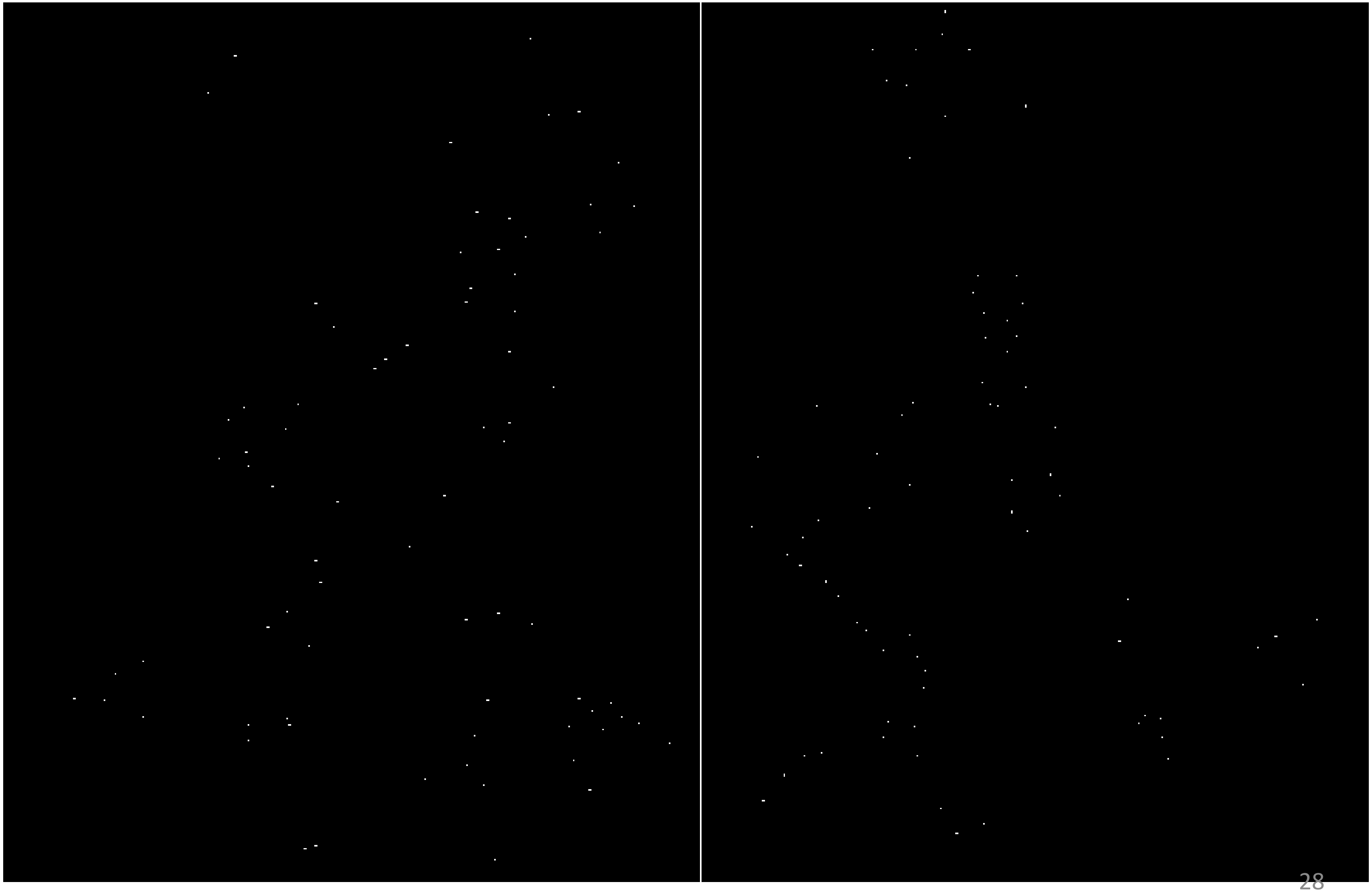
f value (red high, blue low)



Threshold (f > value)



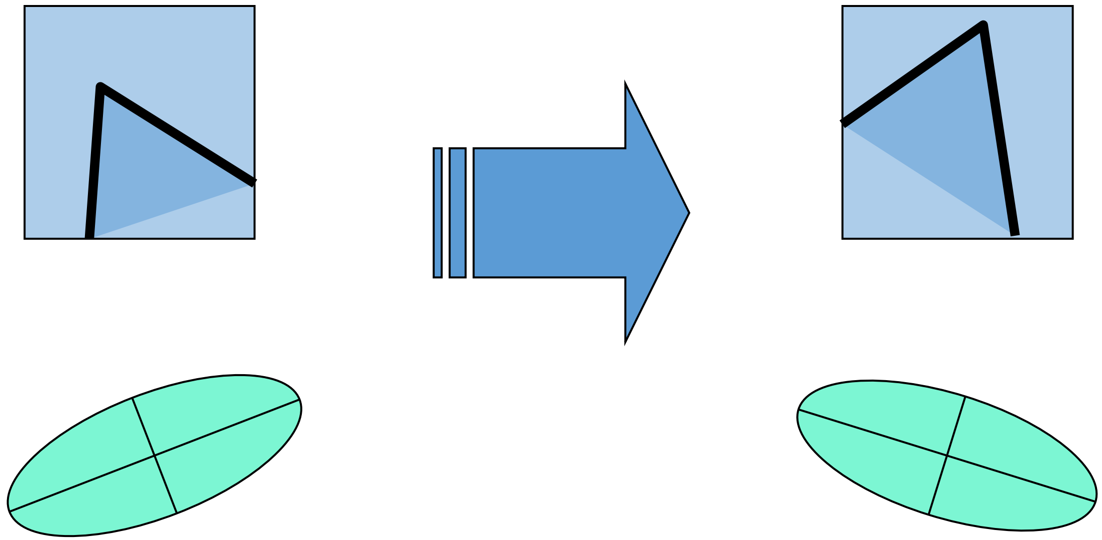
Find Local Maxima of f



Harris Features (in Red)

Harris Detector: Invariance Properties

- Rotation



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

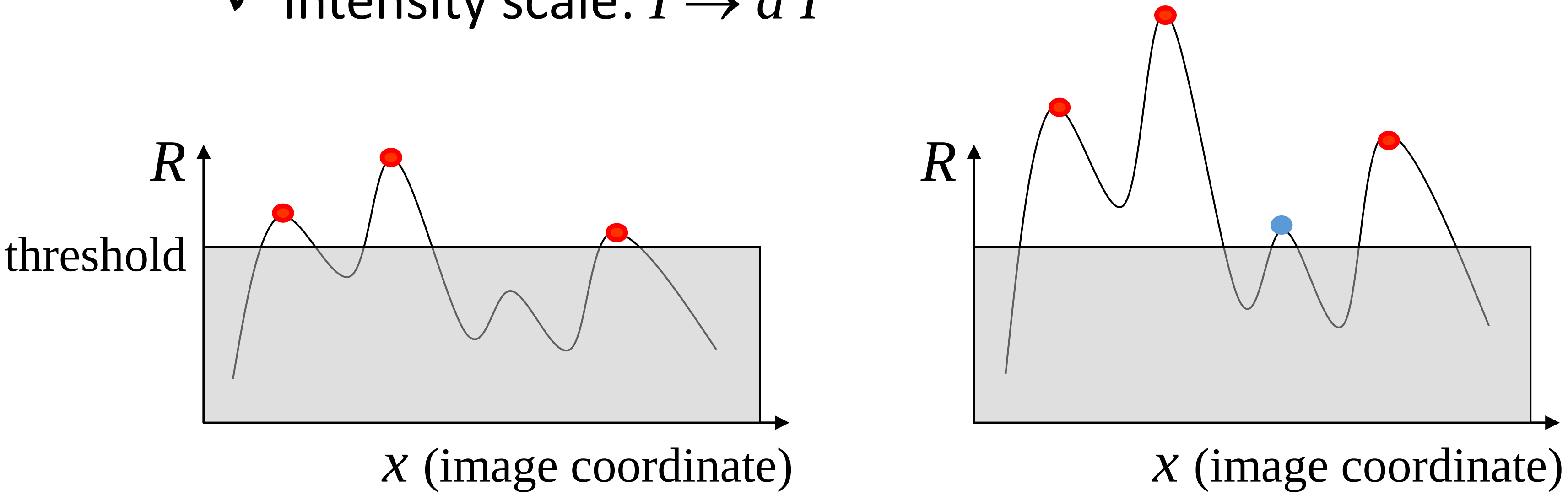
Corner response is invariant to image rotation

Harris Detector: Invariance Properties

- Affine intensity change: $I \rightarrow aI + b$

  ✓ Only derivatives are used =>
  invariance to intensity shift $I \rightarrow I + b$

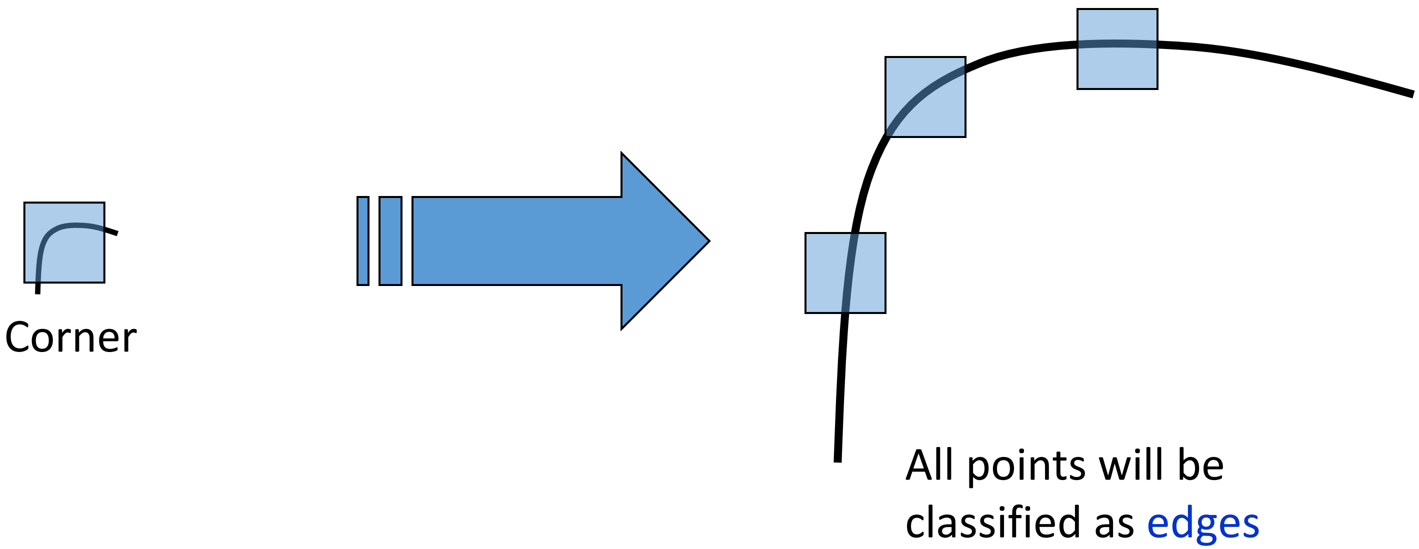
  ✓ Intensity scale: $I \rightarrow a\,I$



Partially invariant to affine intensity change

Harris Detector: Invariance Properties

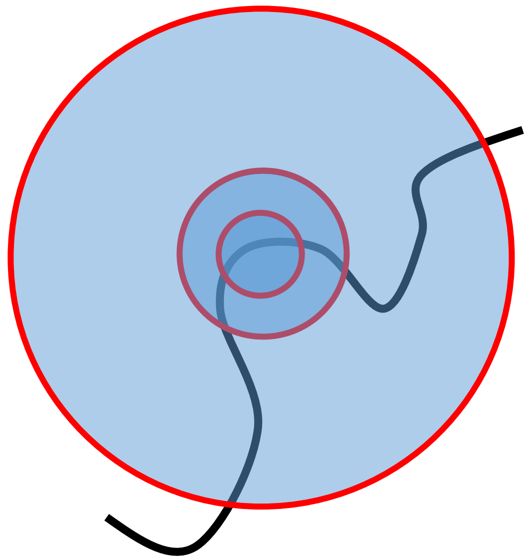
- Scaling



Not invariant to scaling

Scale Invariant Detection

Suppose you're looking for corners



Key idea: find scale that gives local maximum of f

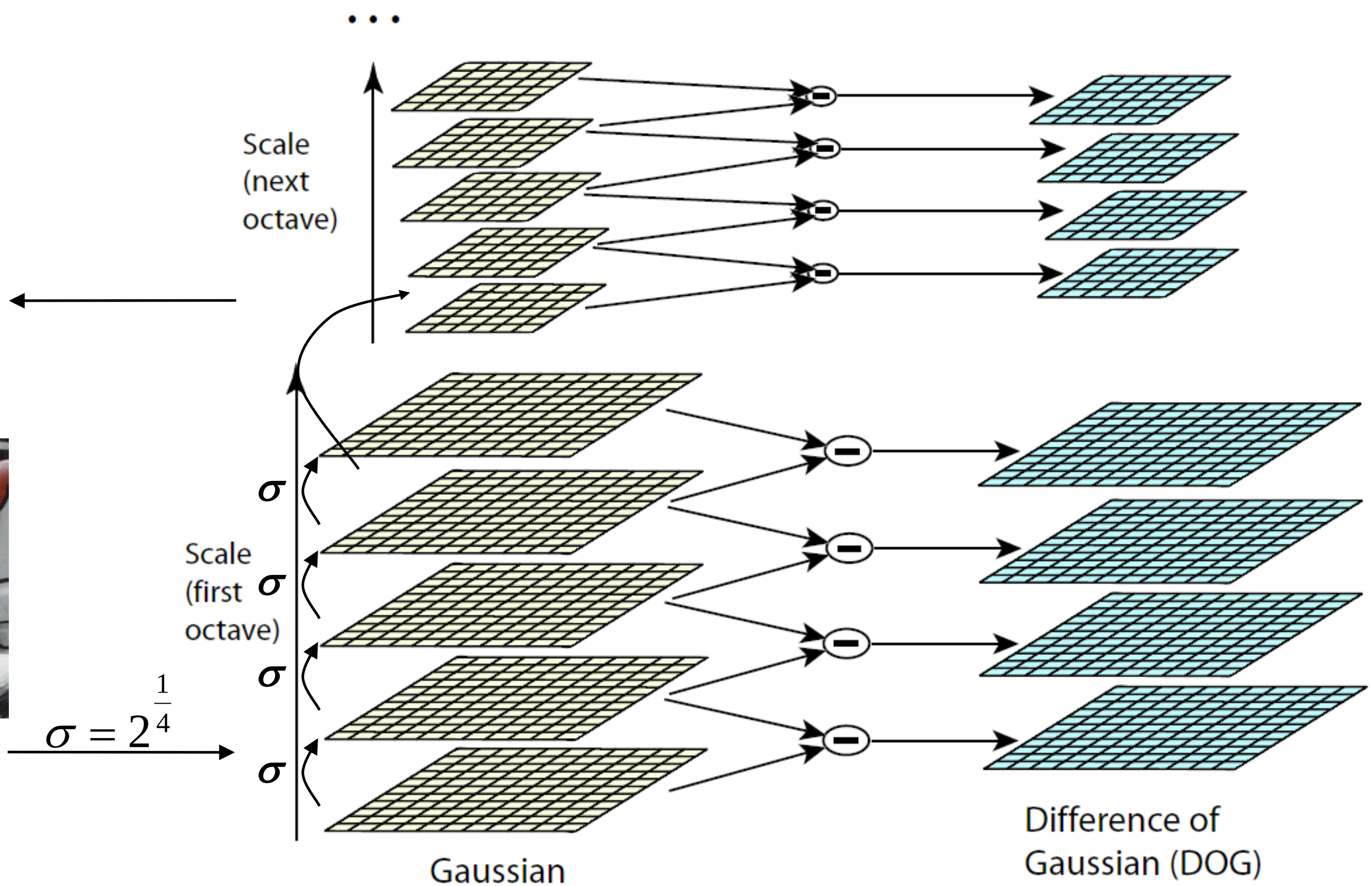
- in both position and scale
- One definition of f: the Harris operator

DoG – Efficient Computation

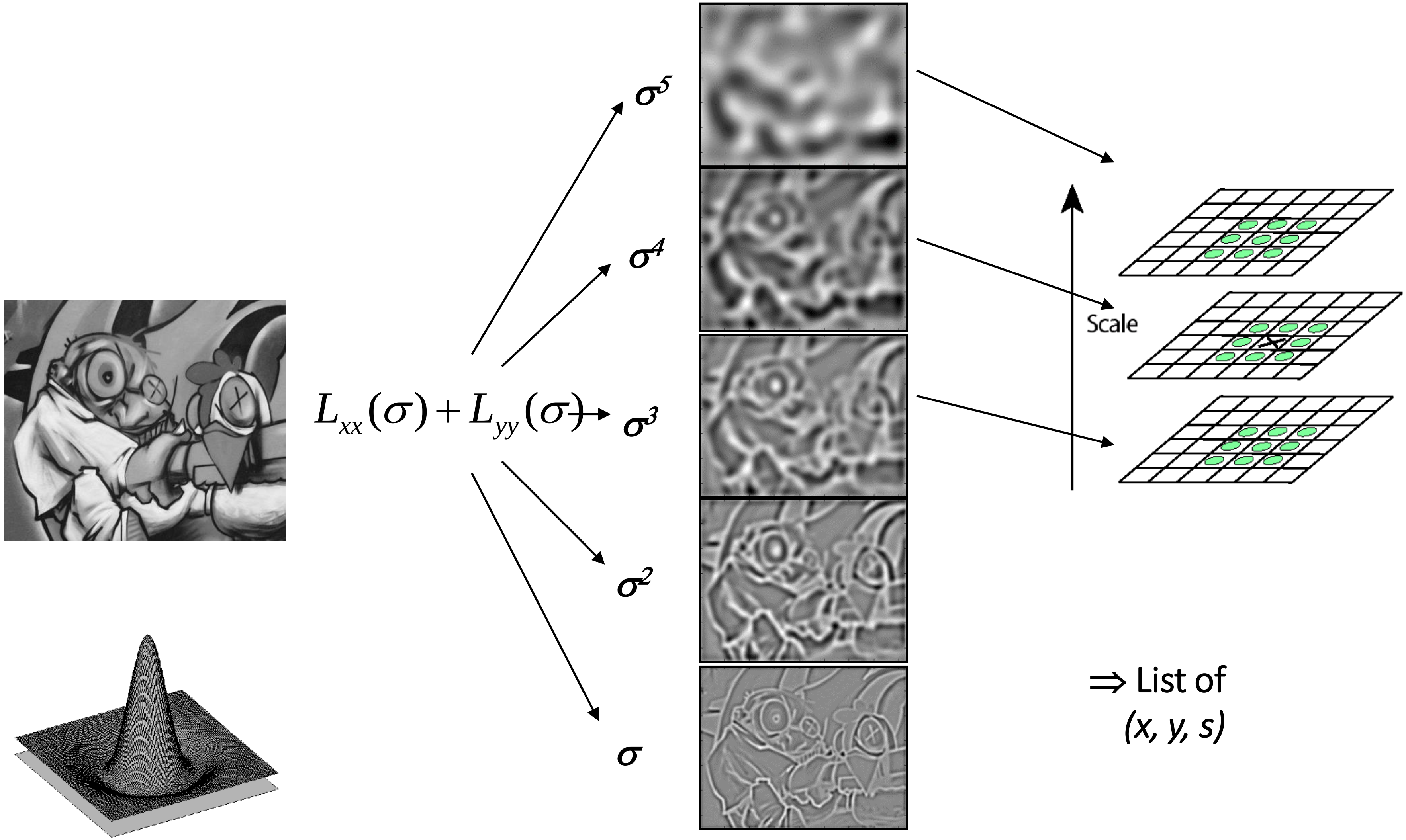
- Computation in Gaussian scale pyramid

Sampling with step σ^4 =2

Original image

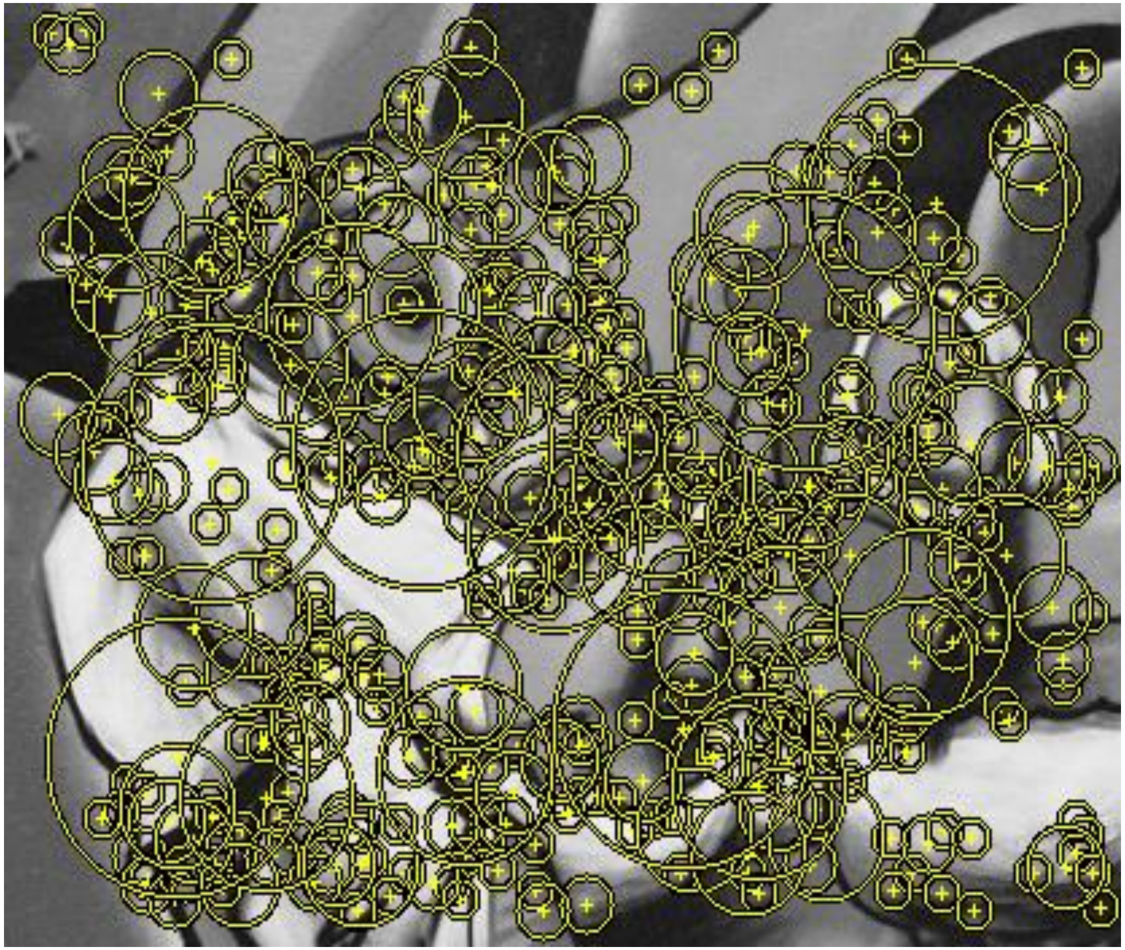


Find Local Maxima in Position-Scale Space of Difference-of-Gaussian



$\Rightarrow$ List of
(x, y, s)

Results: Difference-of-Gaussian

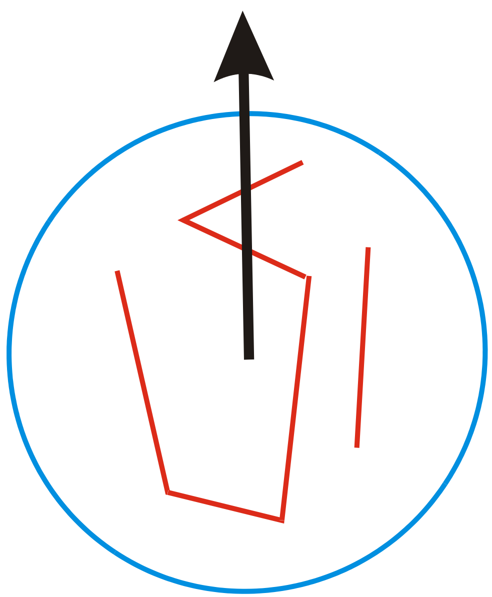


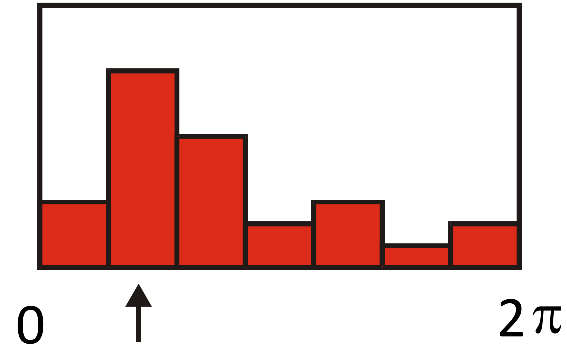
K. Grauman, B. Leibe

Orientation Normalization

[Lowe, SIFT, 1999]

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

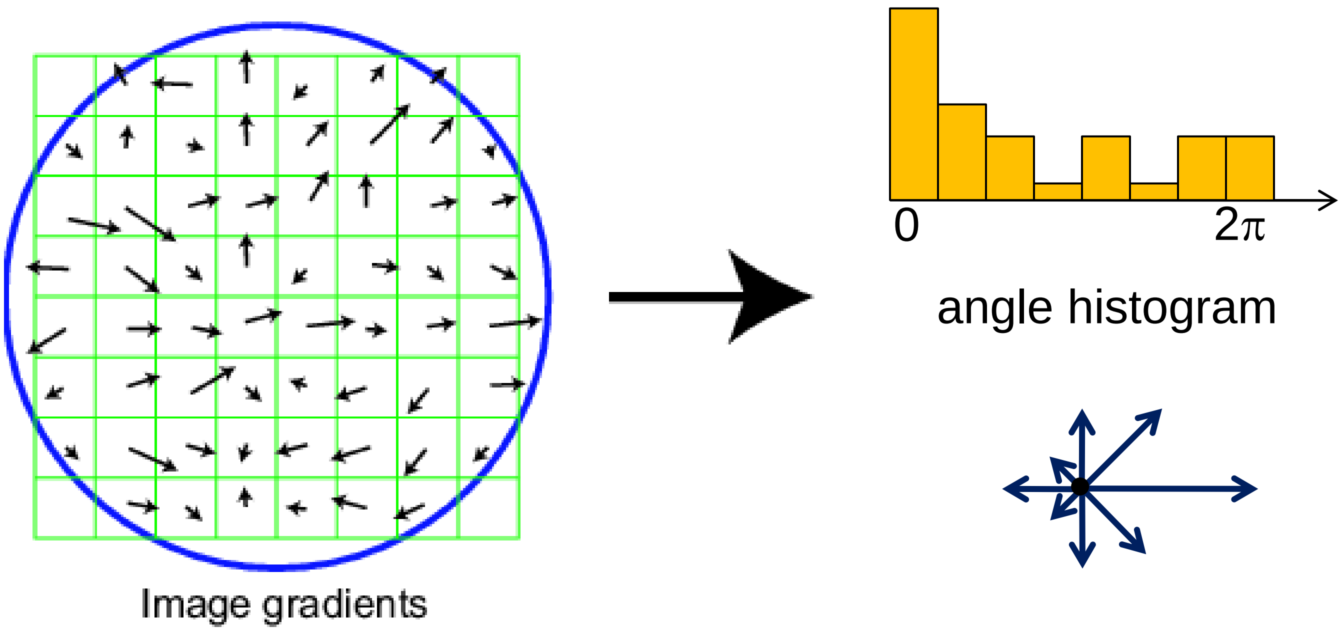




Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

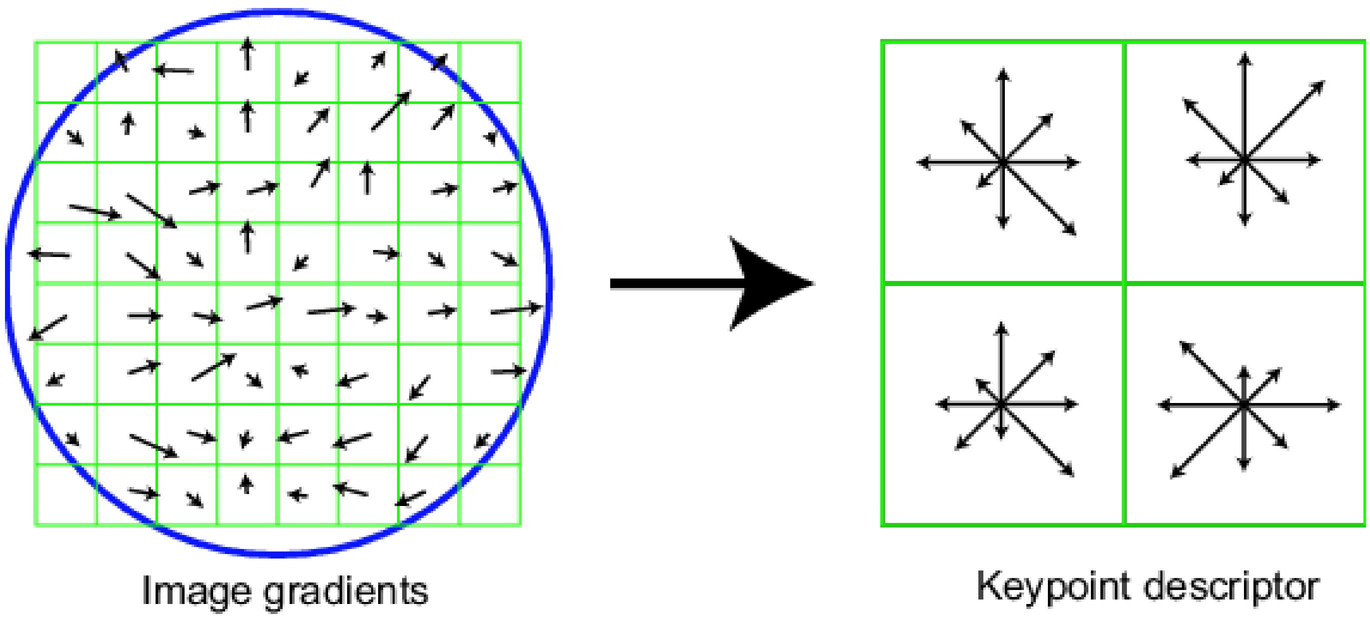


Adapted from slide by David Lowe

SIFT descriptor

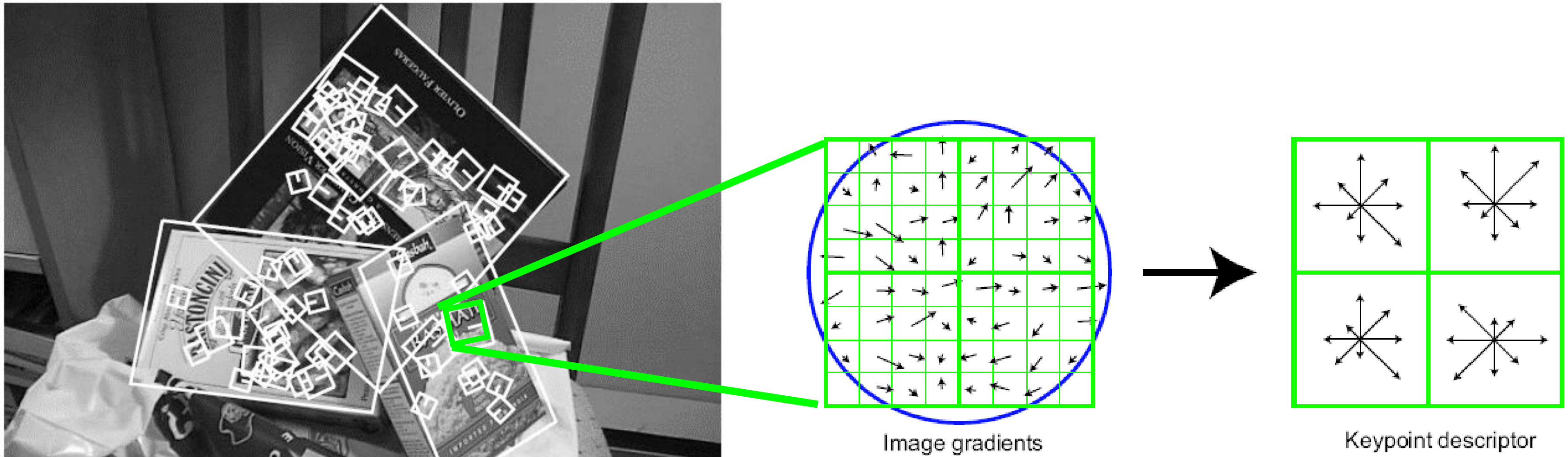
Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

Local Descriptors: SIFT Descriptor



Histogram of oriented gradients

- **Captures important texture information**
- **Robust to small translations / affine deformations**

[Lowe, ICCV 1999]

K. Grauman, B. Leibe

Details of Lowe's SIFT algorithm

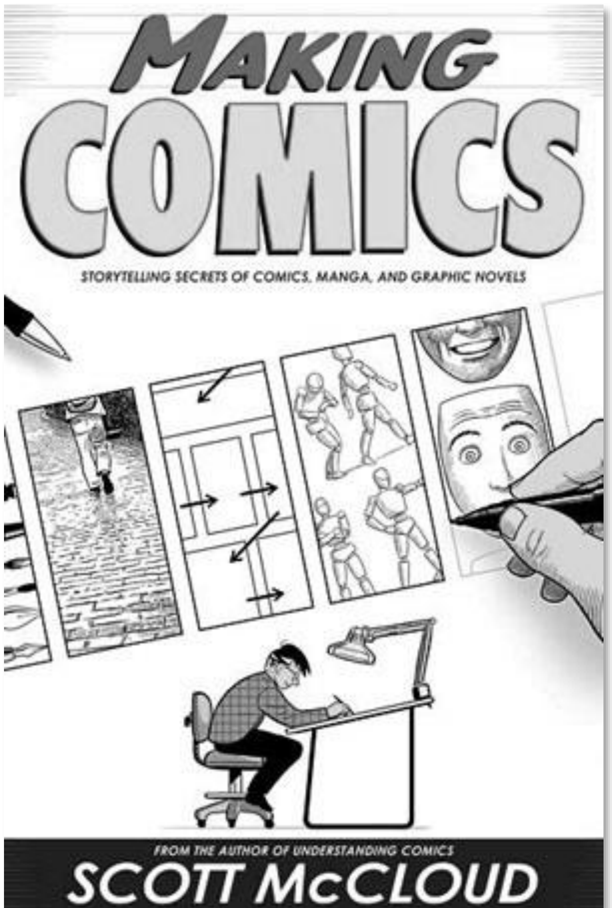
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$

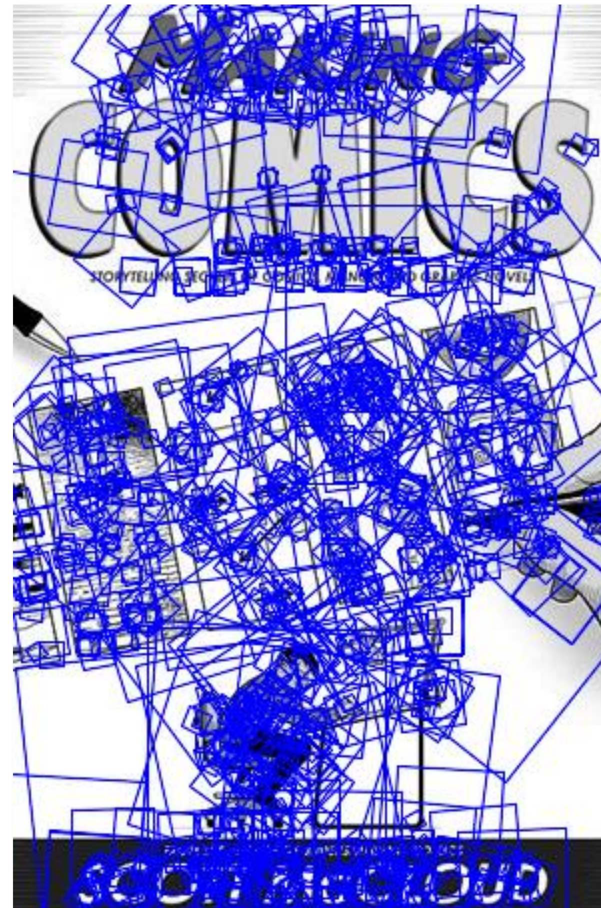
- Run DoG detector
  - Find maxima in location/scale space
  - Remove edge points
- Find all major orientations
  - Bin orientations into 36 bin histogram
    - Weight by gradient magnitude
    - Weight by distance to center (Gaussian-weighted mean)
  - Return orientations within 0.8 of peak
    - Use parabola for better orientation fit
- For each (x,y,scale,orientation), create descriptor:
  - Sample 16x16 gradient mag. and rel. orientation
  - Bin 4x4 samples into 4x4 histograms
  - Threshold values to max of 0.2, divide by L2 norm
  - Final descriptor: 4x4x8 normalized histograms

Ref: D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

SIFT Example



sift



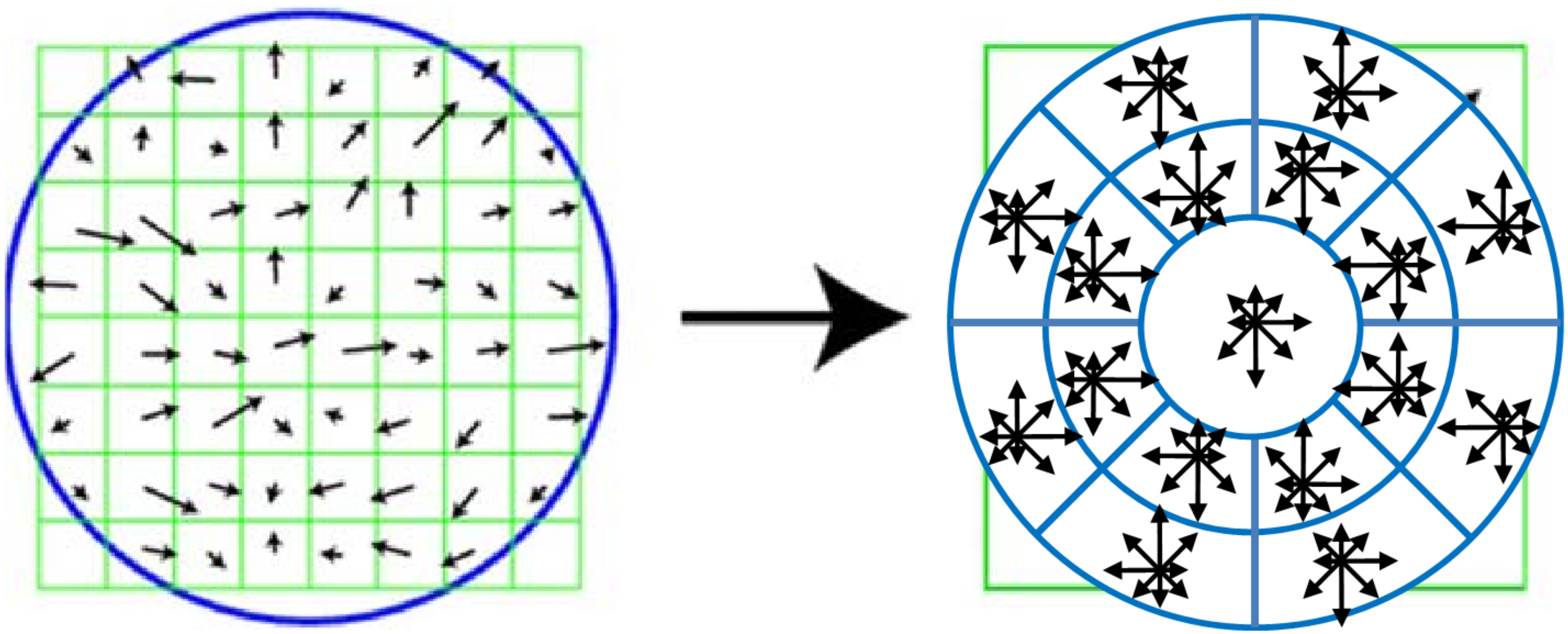
868 SIFT features

PCA SIFT

- 39 x 39 patch → 3042-D vector
- Dimension reduction to 36-D with principal component analysis (PCA)

Ref: Y. Ke and R. Sukthankar, “PCA-SIFT: a more distinctive representation for local image descriptors,” in *Proc. CVPR2004*.

Gradient Location-Orientation Histogram (GLOH)



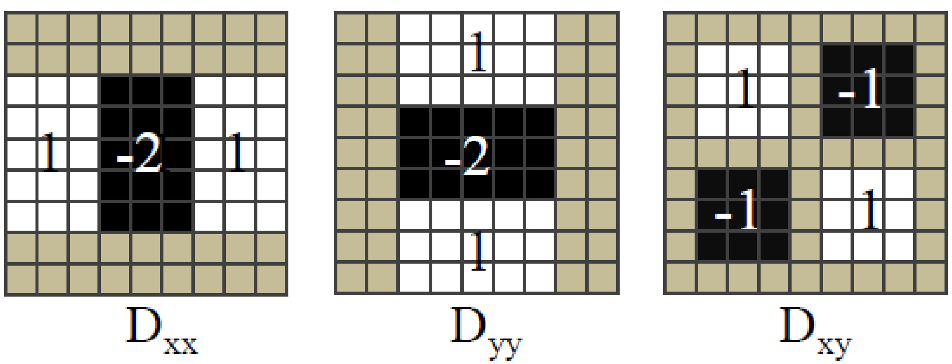
Ref: K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Tran. Pattern Analysis and Machine Intelligence*, 2005.

Speed Up Robust Feature (SURF)

- SURF detector
  - Hessian Matrix Based Interest Points

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

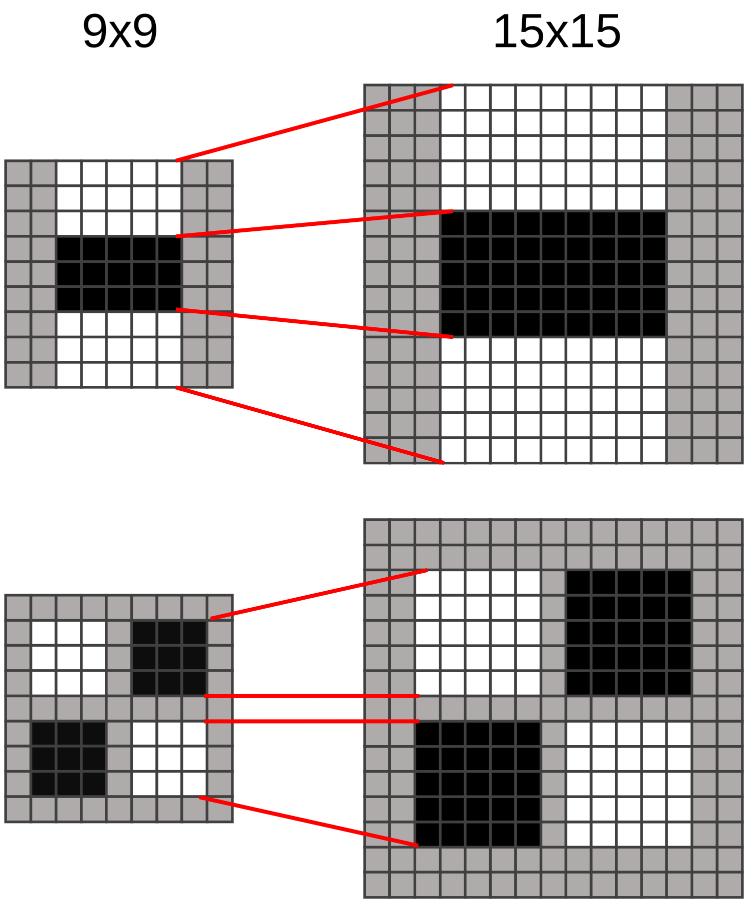
  - Approximation for Hessian Matrix
  - Low computational cost



Ref: T. Tuytelaars H. Bay, A. Ess and L. V. Gool, “SURF: Speeded up robust features," in *Proceedings of Computer Vision and Image Understanding (CVIU)*, 2008, vol. 110, pp. 346-359.

Speed Up Robust Feature (SURF)

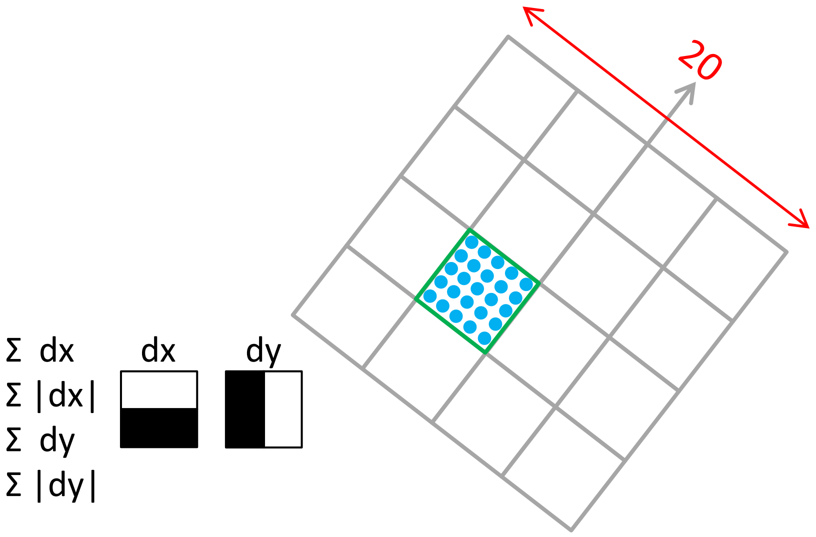
- SURF detector
  - Scale space representation



Octave	Filter size	Sampling Interval
1	9×9, 15×15, 21×21, 27×27	2
2	39×39, 51×51	4
3	75×75, 99×99	8
4	147×147, 195×195	16
5	291×291, 387×387	32

Speed Up Robust Feature (SURF)

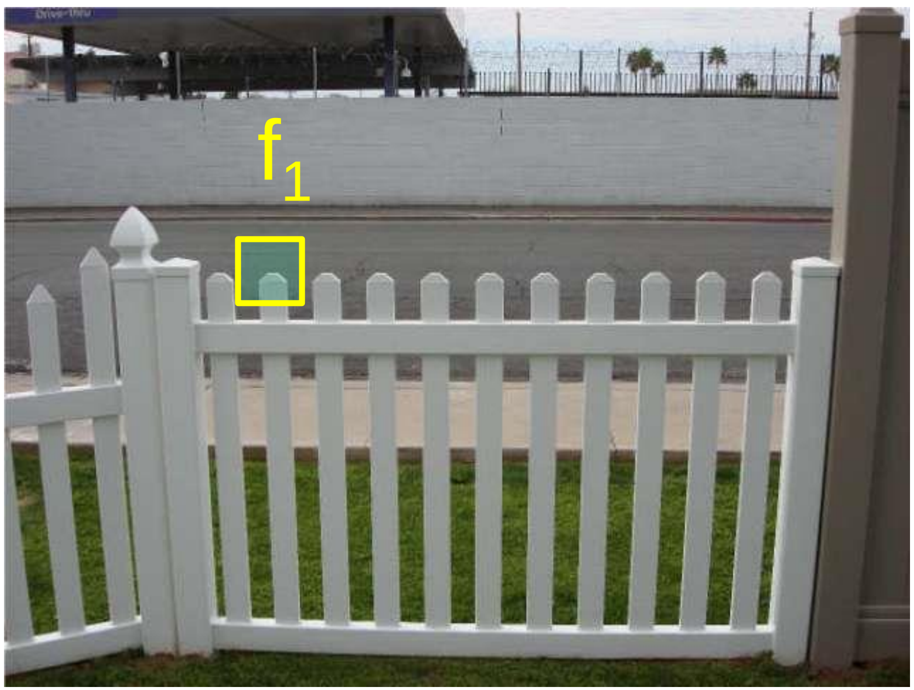
- Descriptor
  - Based on sum of Haar wavelet response
  - dx,dy : wavelet responses in x & y direction
  - 4x4 sub-region
  - Calculate Σdx , Σdy, Σ|dx|, Σ|dy|
  - 4*4*4 = 64 dimensions
  - 4*4*5*5=400 times calculation for an interest point
  - Irregular pattern



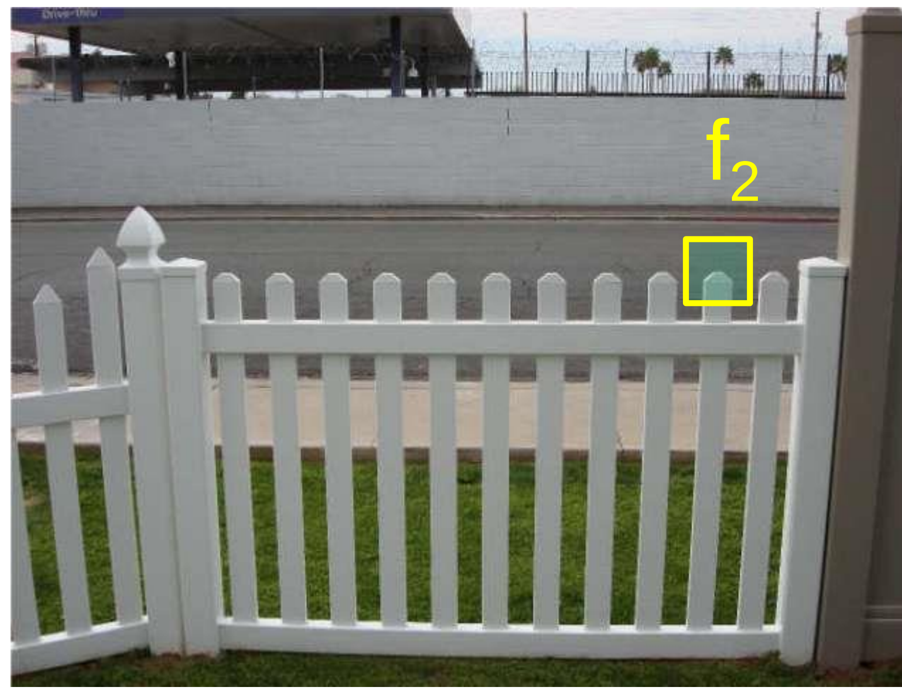
Feature Matching

How to define the difference between two features f_1, f_2?

- Simple approach is SSD(f_1, f_2)
  - sum of square differences between entries of the two descriptors
  - can give good scores to very ambiguous (bad) matches



I_1

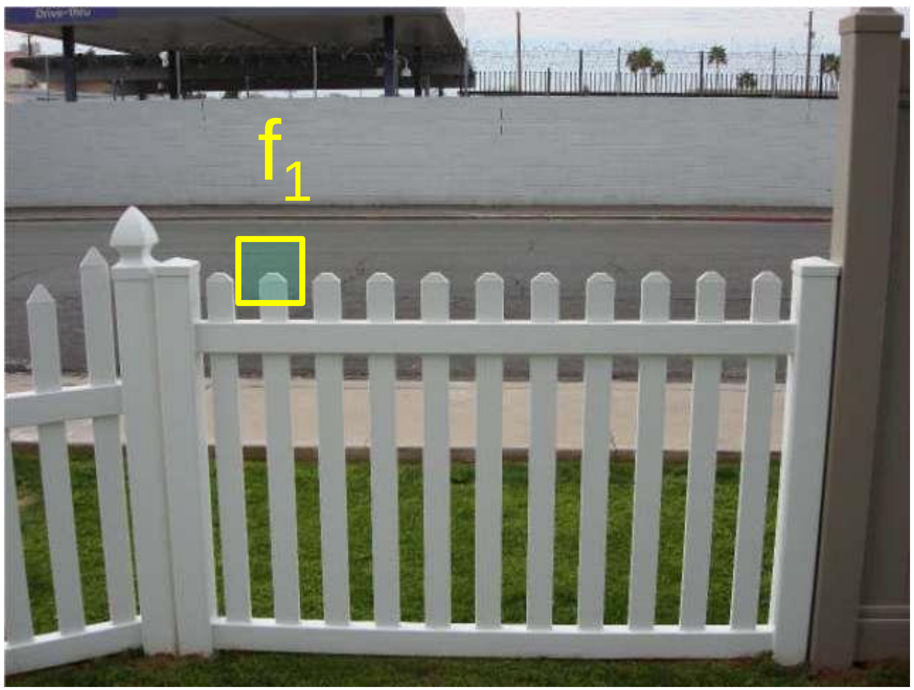


I_2

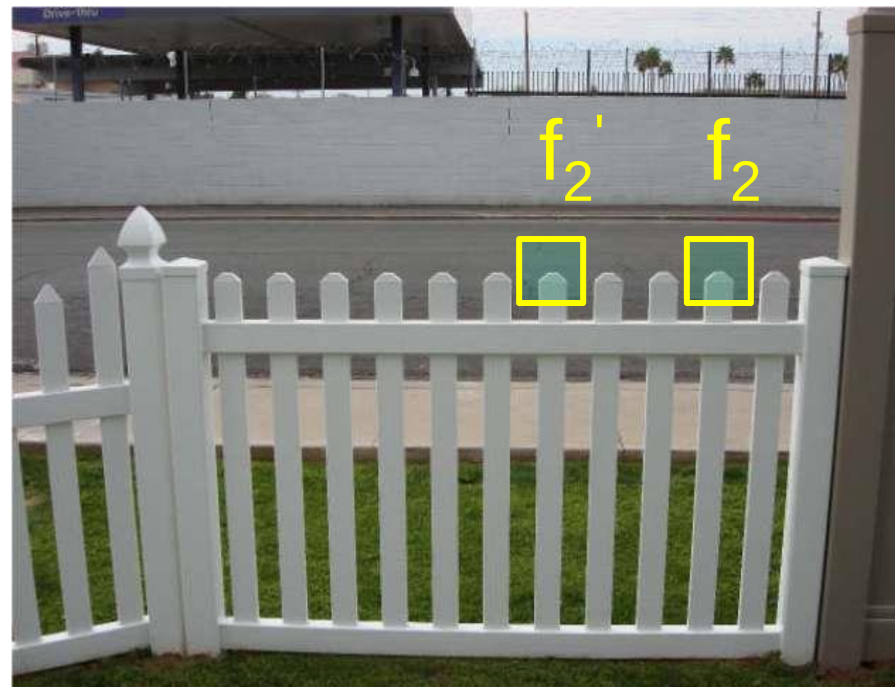
Feature Matching

How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = $SSD(f_1, f_2) / SSD(f_1, f_2')$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2^{nd} best SSD match to f_1 in I_2
  - gives small values for ambiguous matches



I_1



I_2

Feature Matching

- Matching? The difference < threshold
- How to evaluate?

	True matches	True non-matches	
Predicted matches	TP = 18	FP = 4	P' = 22
Predicted non-matches	FN = 2	TN = 76	N' = 78
	P = 20	N = 80	Total = 100

TPR = 0.90	FPR = 0.05

PPV = 0.82

ACC = 0.94

- TP: true positives
- FN: false negatives
- FP: false positives
- TN: true negatives

Feature Matching

- How to evaluate?

	True matches	True non-matches	
Predicted matches	TP = 18	FP = 4	P' = 22
Predicted non-matches	FN = 2	TN = 76	N' = 78
	P = 20	N = 80	Total = 100

TPR = 0.90	FPR = 0.05

PPV = 0.82
ACC = 0.94

- True positive rate (TPR), recall $$TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$
- False positive rate (FPR), false alarm $$FPR = \frac{FP}{FP + TN} = \frac{FP}{N}$$
- Positive predictive value (PPV), precision $$PPV = \frac{TP}{TP + FP} = \frac{TP}{P'}$$
- Accuracy (ACC) $$ACC = \frac{TP + TN}{P + N}$$

Feature Matching

- How to evaluate?

True positive rate (TPR)

$$TPR = \frac{TP}{TP + FN}$$

False positive rate (FPR)

$$FPR = \frac{FP}{FP + TN}$$

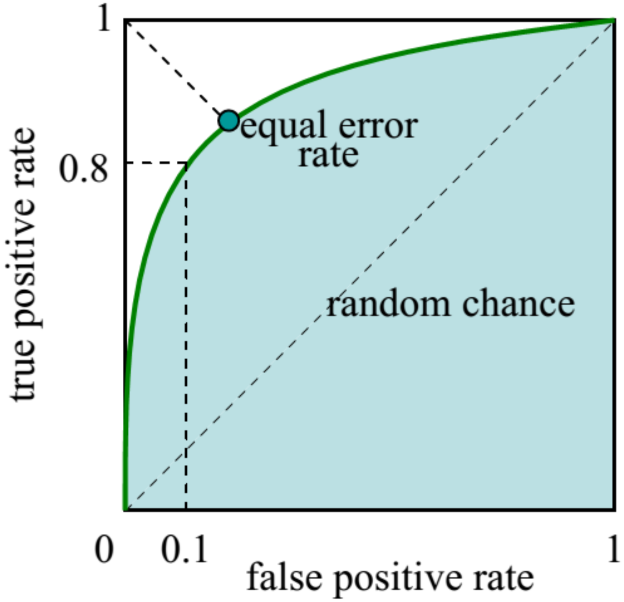
Positive predictive value (PPV)

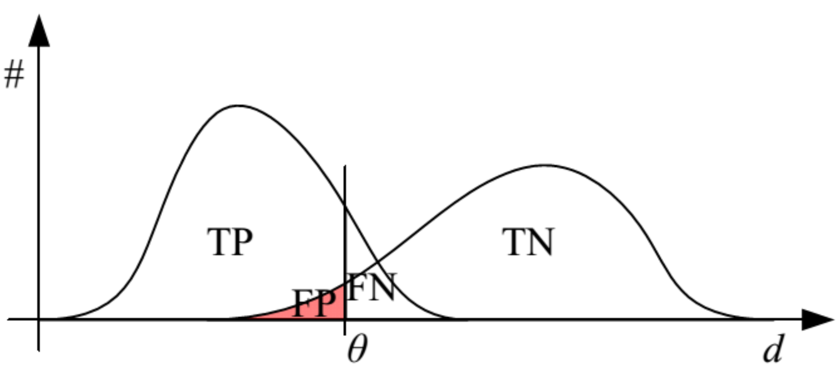
$$PPV = \frac{TP}{TP + FP}$$

Accuracy (ACC)

$$ACC = \frac{TP + TN}{P + N}$$

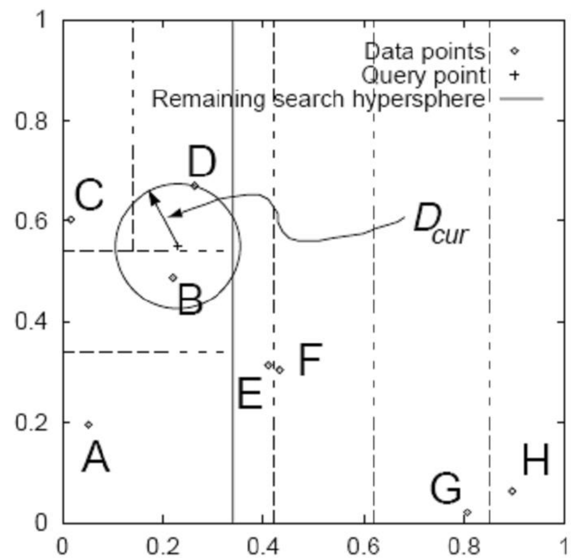
ROC curve (Receiver Operating Characteristic)

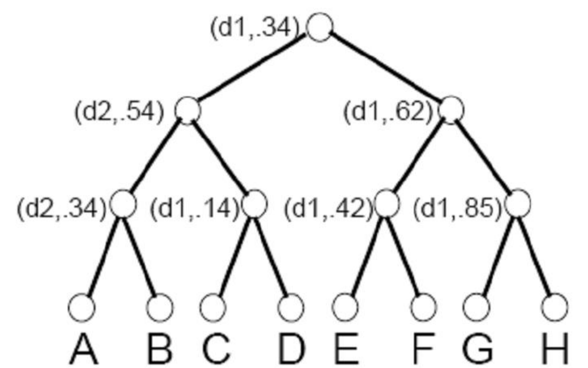




Feature Matching

- Efficient matching
  - Full search
  - Indexing structure
  - Multi-dimensional hashing
  - Locality sensitive hashing (LSH)
  - K-d tree



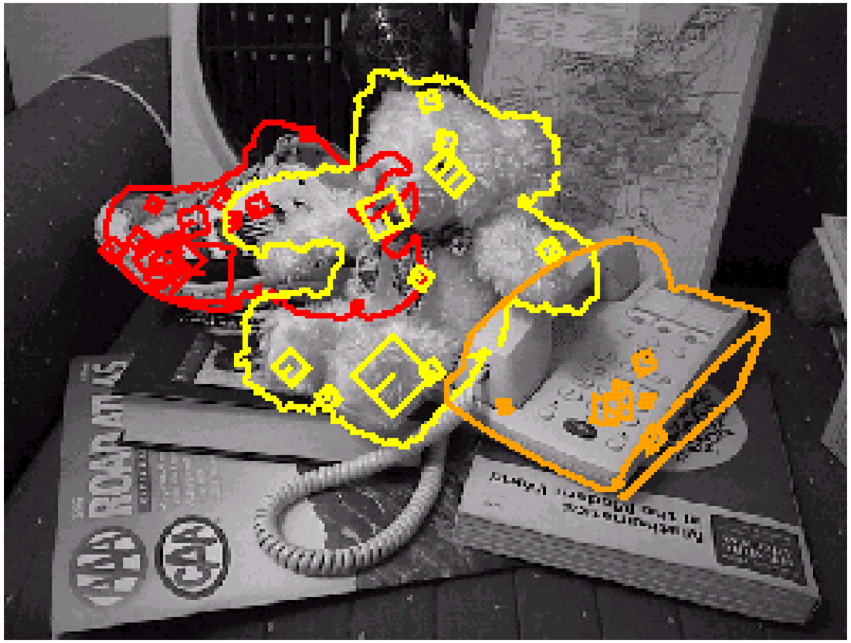


Applications

Features are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- … other

Object Recognition (David Lowe)



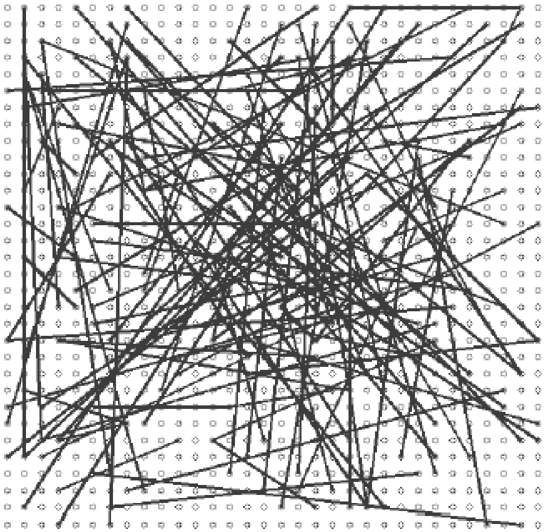
BRIEF (ECCV 2010)

- We define test τ on patch $\mathbf{p}$ of size $S \times S$ as

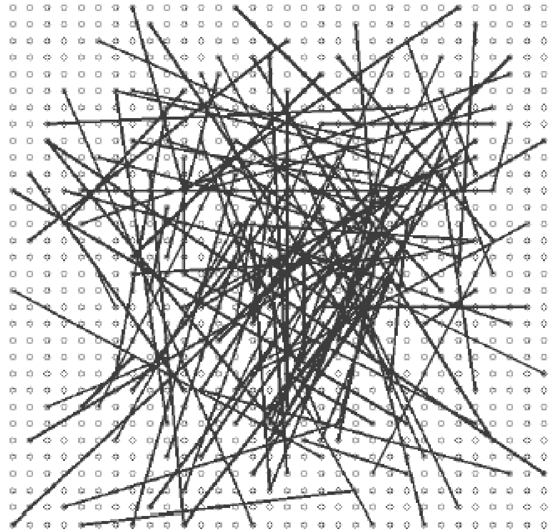
$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases}$$

  - where $\mathbf{p}(\mathbf{x})$ is the pixel intensity in a smoothed version of $\mathbf{p}$ at $\mathbf{x} = (u, v)$T.
- Choosing a set of n_d (x, y)-location pairs uniquely defines a set of binary tests.
  - We take our BRIEF descriptor to be the n_d-dimensional bitstring

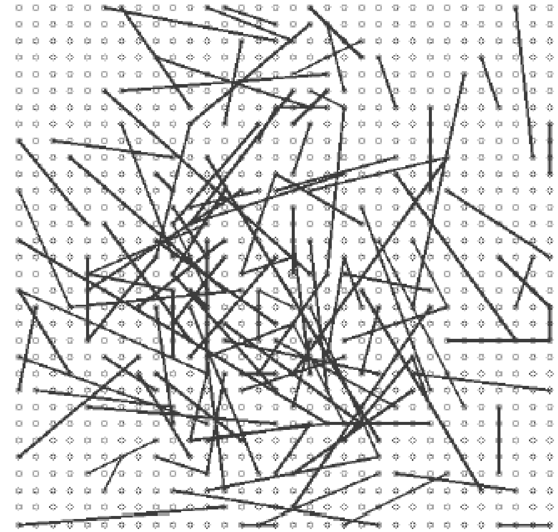
$$f_{n_d}(\mathbf{p}) := \sum_{1 \le i \le n_d} 2^{i-1}\, \tau(\mathbf{p};\mathbf{x}_i,\mathbf{y}_i)$$



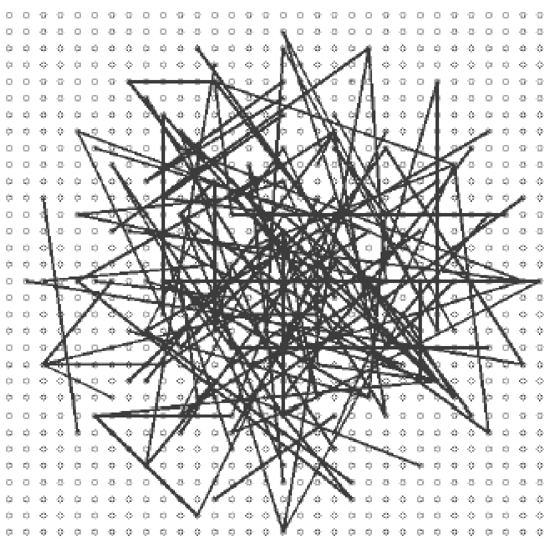
G I



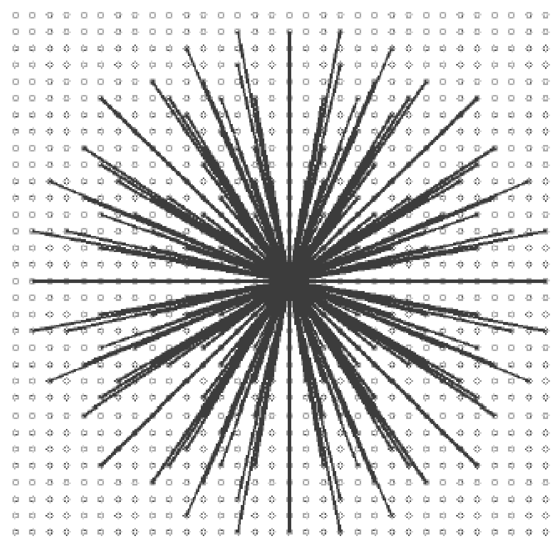
G II



G III



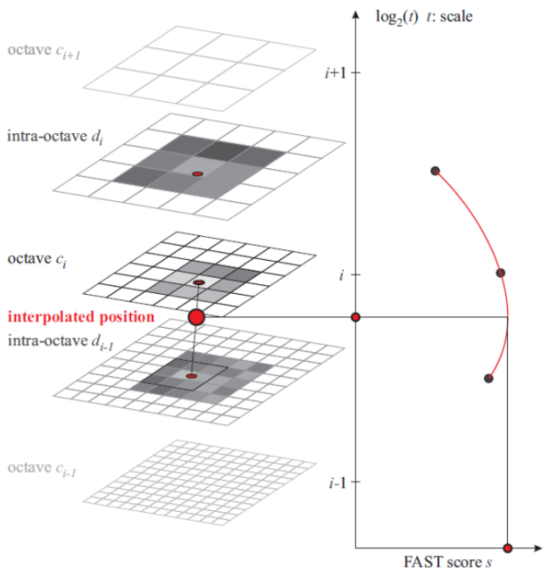
G IV



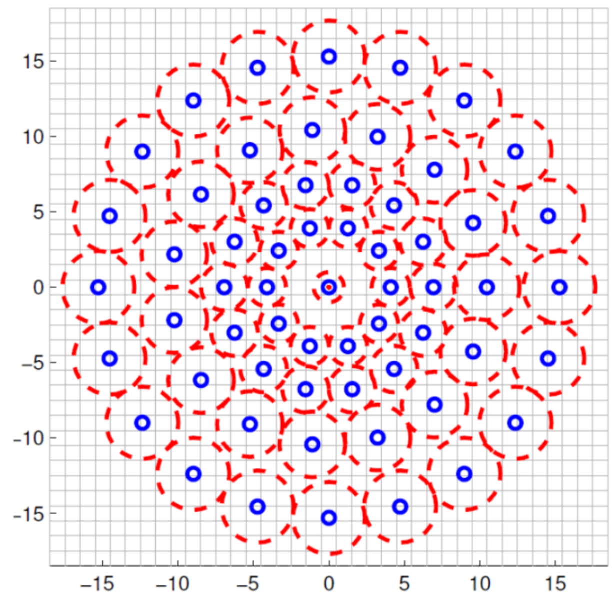
G V

BRISK (ICCV2011)

- Scale-space keypoint detection:



- Keypoint description:

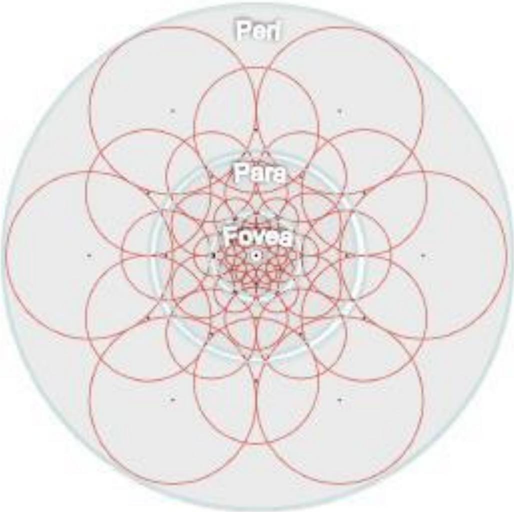


$$b = \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases}$$

$$\forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$$

FREAK (CVPR 2012)

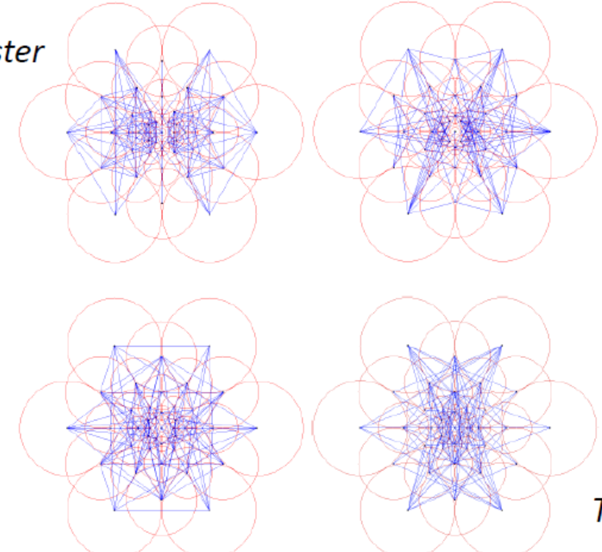
- Retinal sampling pattern
- Coarse-to-fine descriptor



$$F = \sum_{0 \leq a < N} 2^a T(P_a) \qquad T(P_a) = \begin{cases} 1 & \text{if } (I(P_a^{r_1}) - I(P_a^{r_2}) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

- How to select pairs?
  - Learn the best pairs from training data

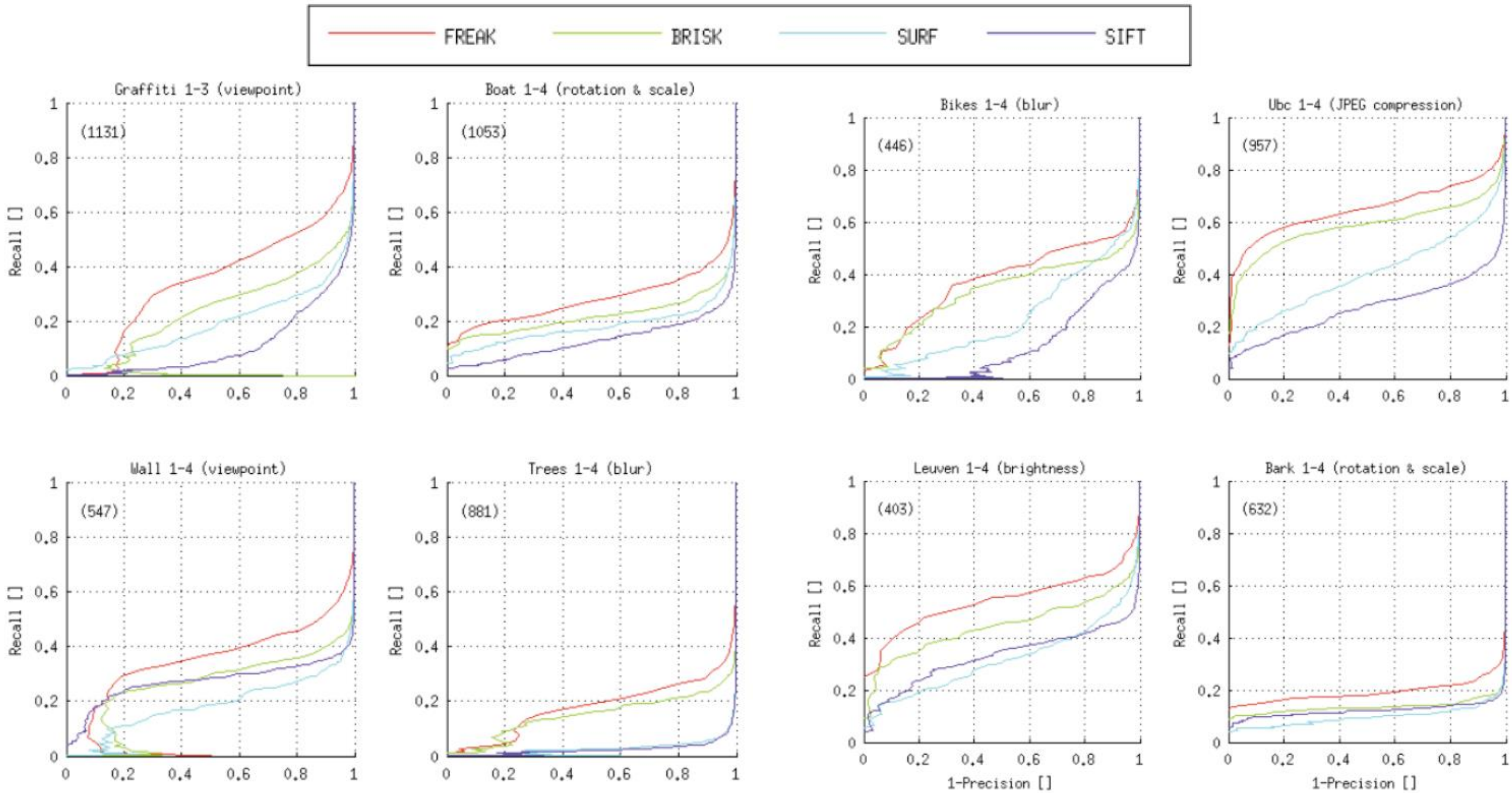
The 1st cluster



4 clusters
128 pairs per group

The last cluster

FREAK (CVPR 2012)



ORB: An efficient alternative to SIFT or SURF

- ORB = oFAST + rBRIEF

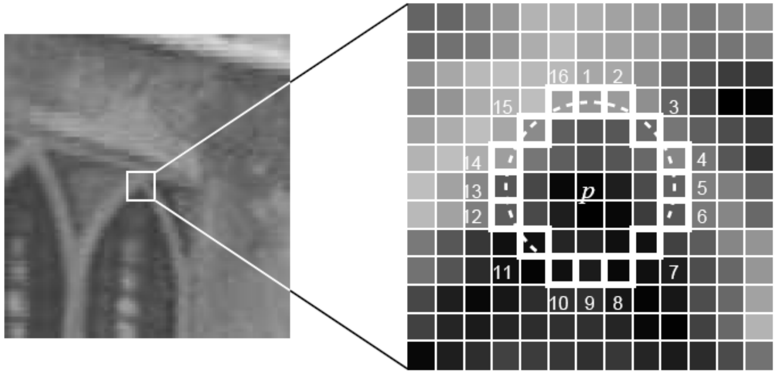
- oFAST: FAST Keypoint Orientation
- rBRIEF: Rotation-Aware Brief

E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. 2011 International Conference on Computer Vision*, Barcelona, 2011.

[1]Rosten, Edward, and Tom Drummond. "Machine learning for high-speed corner detection." Computer Vision–ECCV 2006.

FAST[1]

- Features from Accelerated Segment Test.
  - The segment test criterion operates by considering a circle of sixteen pixels around the corner candidate p.
  - The original detector classifies p as a corner if there exists a set of n contiguous pixels in the circle which are all brighter than the intensity of the candidate pixel $I_p + t$, or all darker than $I_p - t$.



Orientation by Intensity Centroid

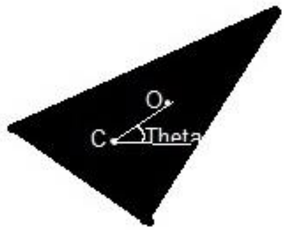
- Moments of a patch

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

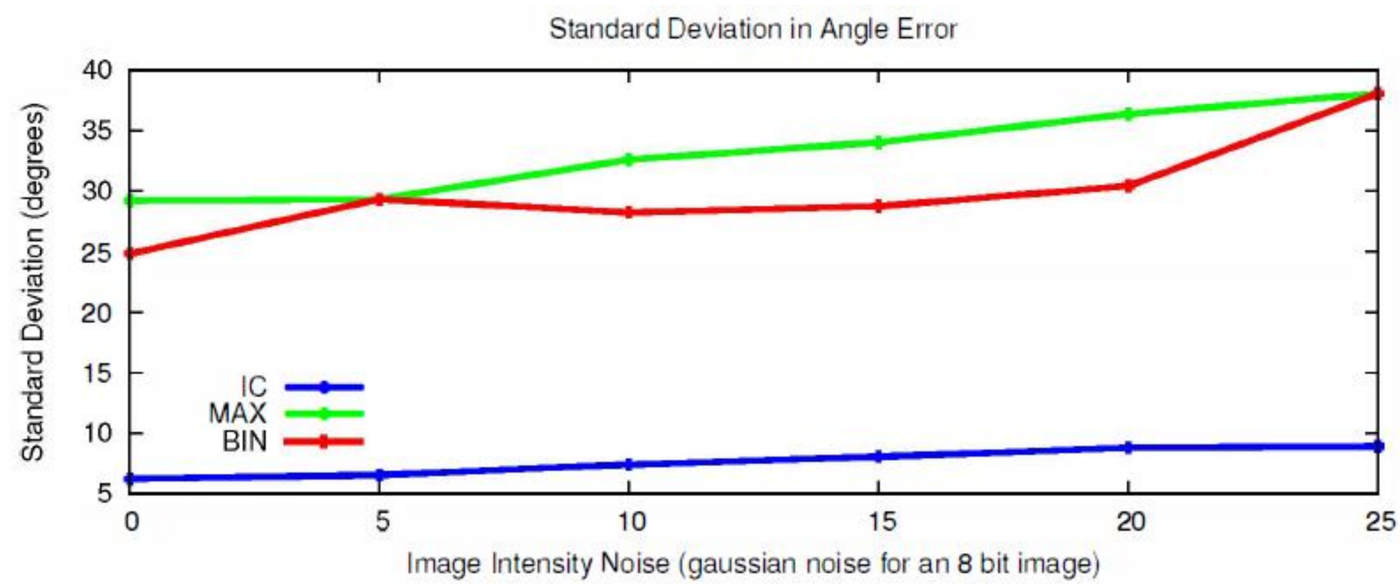
  - with these moments we may find the centroid

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

  - We can construct a vector from the corner's center, O, to the centroid, $\overrightarrow{OC}$.



Rotation Measure



- IC: intensity centroid
- MAX chooses the largest gradient in the keypoint patch
- BIN forms a histogram of gradient directions at 10 degree intervals, and picks the maximum bin.

Steered BRIEF

- Steer BRIEF according to the orientation of keypoints.

$$\mathbf{S} = \begin{pmatrix} \mathbf{x}_1, \ldots, \mathbf{x}_n \\ \mathbf{y}_1, \ldots, \mathbf{y}_n \end{pmatrix}$$

- Using the patch orientation θ and the corresponding rotation matrix R_θ, we construct a "steered" version S_θ of S:

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S}$$

- Now the steered BRIEF operator becomes

$$g_n(\mathbf{p}, \theta) := f_n(\mathbf{p}) | (\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{S}_\theta$$

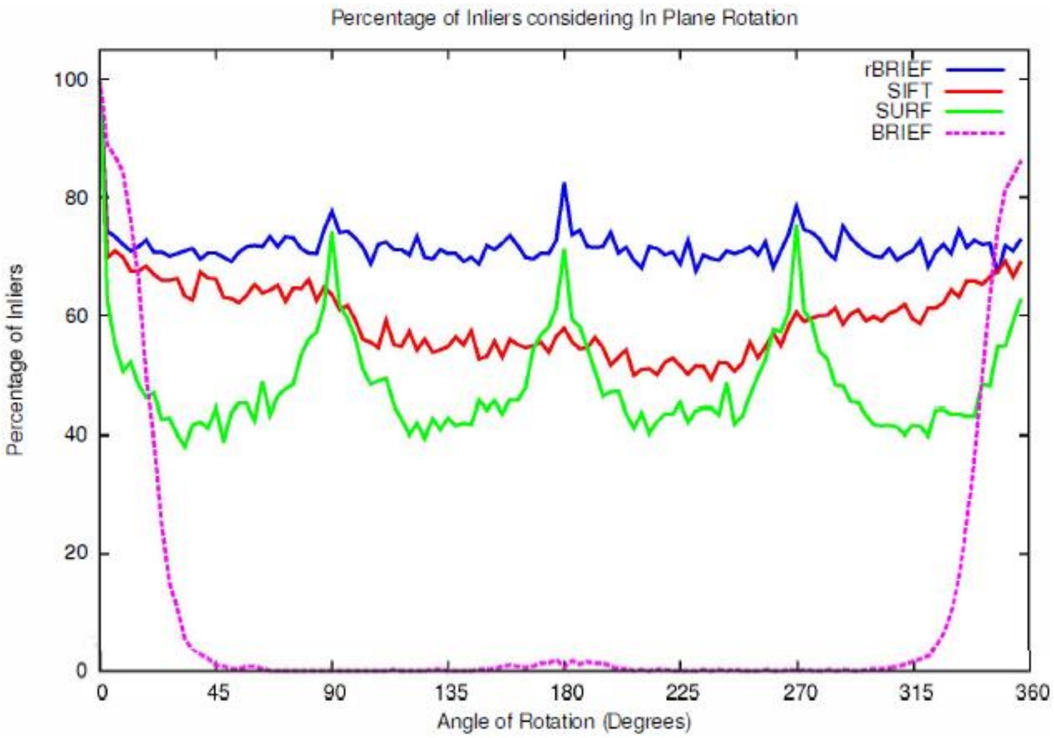
Learning Good Binary Features

- The algorithm is:

1. Run each test against all training patches.
2. Order the tests by their distance from a mean of 0.5, forming the vector T.
3. Greedy search:
   (a) Put the first test into the result vector R and remove it from T.
   (b) Take the next test from T, and compare it against all tests in R. If its absolute correlation is greater than a threshold, discard it; else add it to R.
   (c) Repeat the previous step until there are 256 tests in R. If there are fewer than 256, raise the threshold and try again.

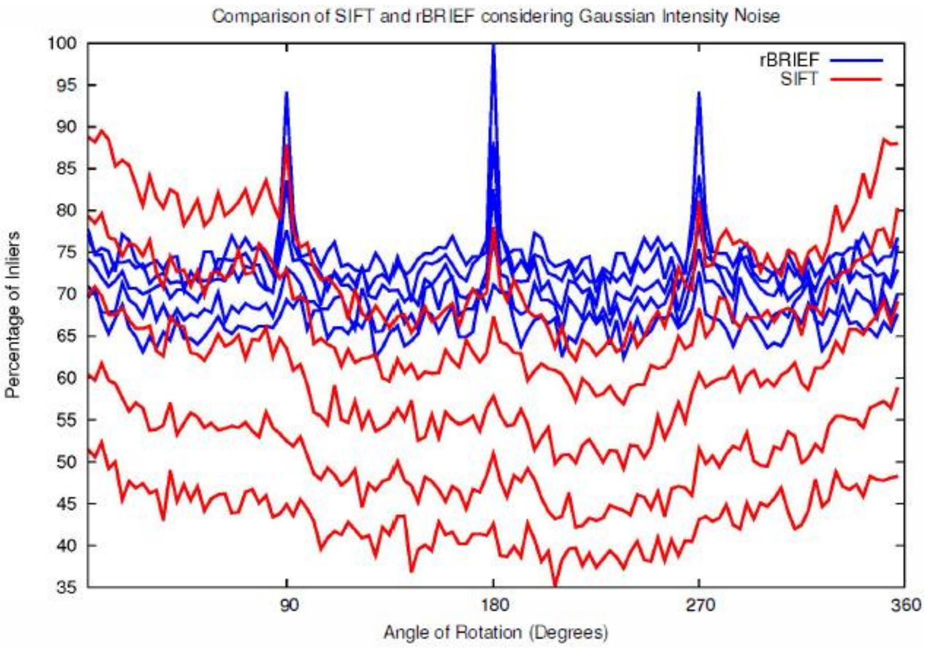
Results (1/3)

- Matching performance of SIFT, SURF, BRIEF with FAST, and ORB (oFAST +rBRIEF) under synthetic rotations with Gaussian noise of 10.



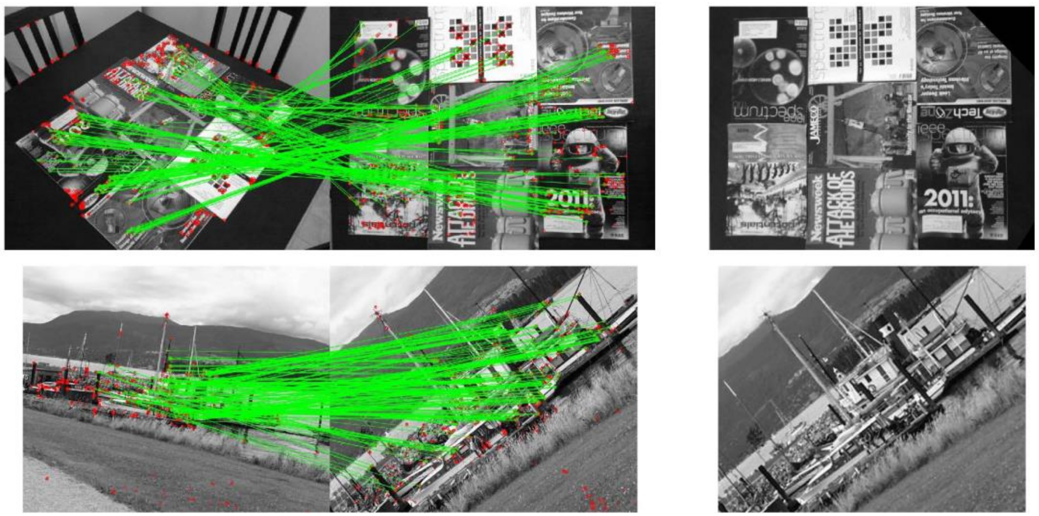
Results (2/3)

- Matching behavior under noise for SIFT and rBRIEF. The noise levels are 0, 5, 10, 15, 20, and 25. SIFT performance degrades rapidly, while rBRIEF is relatively unaffected.



Results (3/3)

- Test on real-world images:



	inlier %	N points
Magazines		
ORB	36.180	548.50
SURF	38.305	513.55
SIFT	34.010	584.15
Boat		
ORB	45.8	789
SURF	28.6	795
SIFT	30.2	714

Computation Time

- The ORB system breaks down into the following times per typical frame of size 640x480.

Intel i7 2.8 GHz

ORB:	Pyramid	oFAST	rBRIEF
Time (ms)	4.43	8.68	2.12

Detector	ORB	SURF	SIFT
Time per frame (ms)	**15.3**	217.3	5228.7

Pascal 2009 dataset
2686 images at 5 scales

OpenCV 2.4.9

- Detector
  - "FAST" – FastFeatureDetector
  - "STAR" – StarFeatureDetector
  - "SIFT" – SIFT (nonfree module)
  - "SURF" – SURF (nonfree module)
  - "ORB" – ORB
  - "BRISK" – BRISK
  - "MSER" – MSER
  - "GFTT" – GoodFeaturesToTrackDetector
  - "HARRIS" – GoodFeaturesToTrackDetector with Harris detector enabled
  - "Dense" – DenseFeatureDetector
  - "SimpleBlob" – SimpleBlobDetector

OpenCV 2.4.9

- Descriptor
  - "SIFT" – SIFT
  - "SURF" – SURF
  - "BRIEF" – BriefDescriptorExtractor
  - "BRISK" – BRISK
  - "ORB" – ORB
  - "FREAK" – FREAK

Edges and Lines

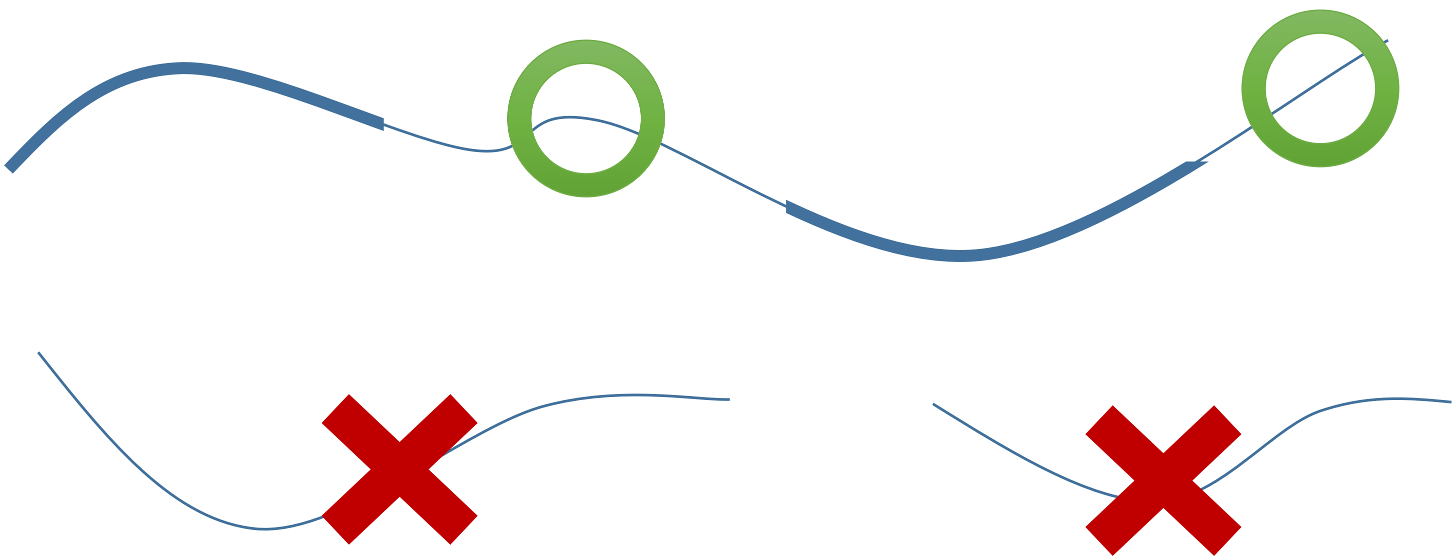
Y. Cao, C. Wang, L. Zhang and L. Zhang, "Edgel index for large-scale sketch-based image search," in *Proc. CVPR 2011*.

Edge Detection

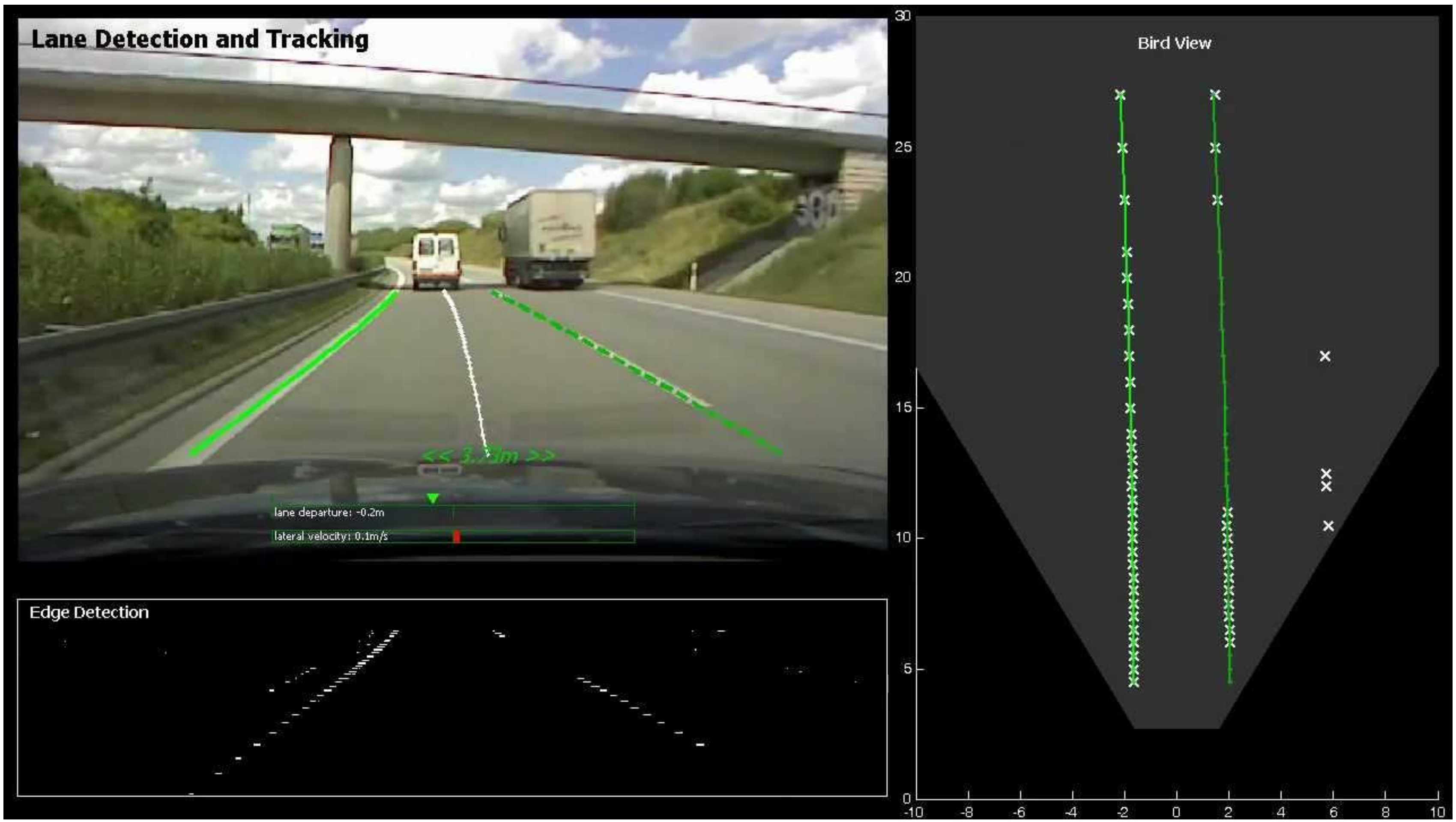
- Canny edge detector
  - The most widely used edge detector
  - The best you can find in existing tools like MATLAB, OpenCV…
- Algorithm:
  - Apply Gaussian filter to reduce noise
  - Find the intensity gradients of the image
  - Apply **non-maximum suppression** to get rid of false edges
  - Apply **double threshold** to determine potential edges
  - Track edge by **hysteresis**: suppressing weak edges that are not connected to strong edges

Hysteresis

- Find **connected components** from strong edge pixels to finalize edge detection

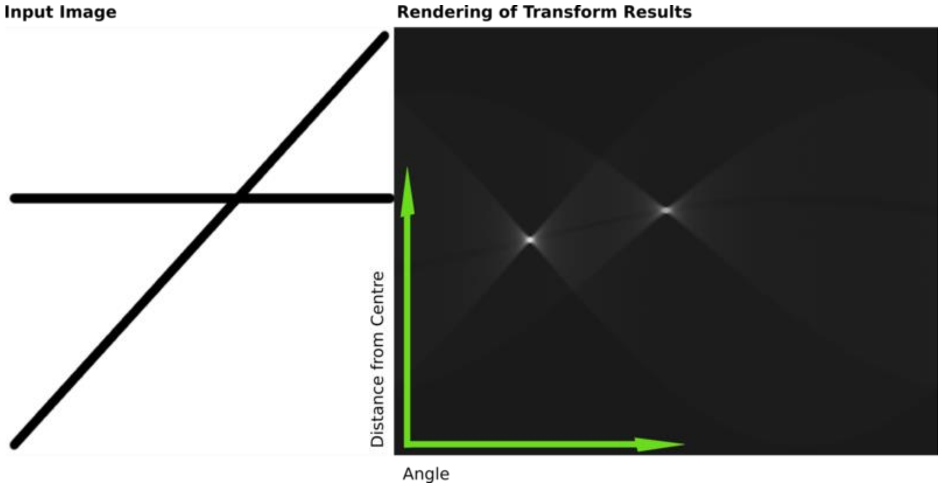


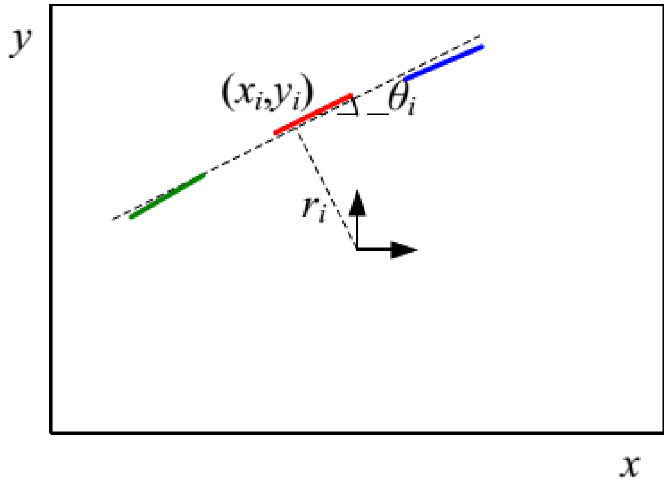
Hough Transform

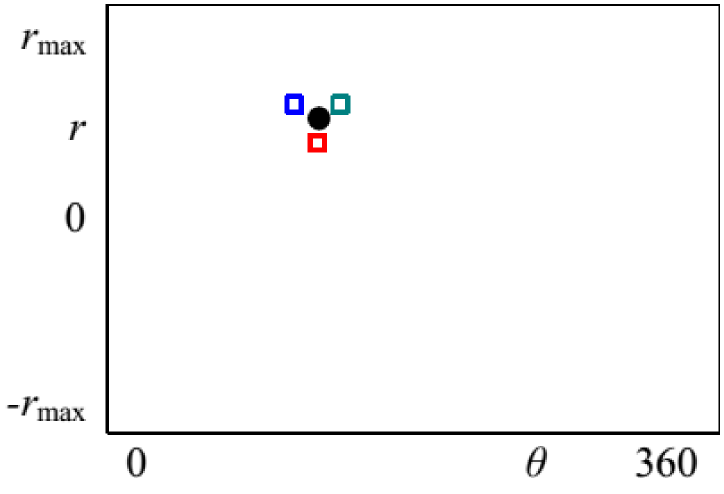


Hough Transform

- Vote in (θ, r) space
  - (Many choices)

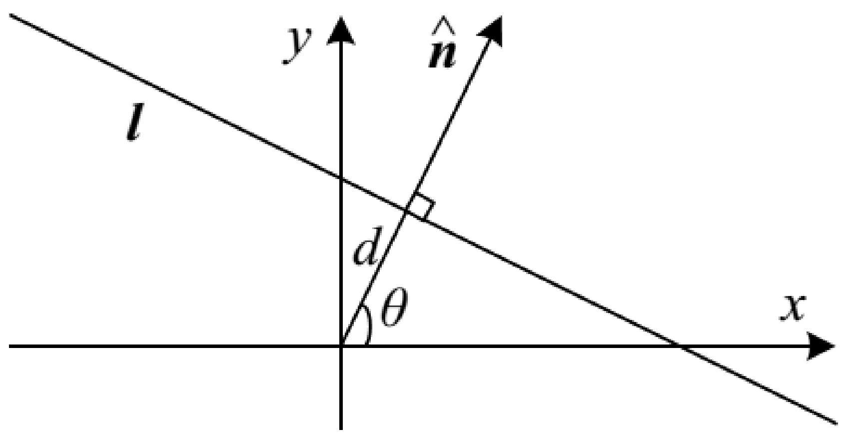


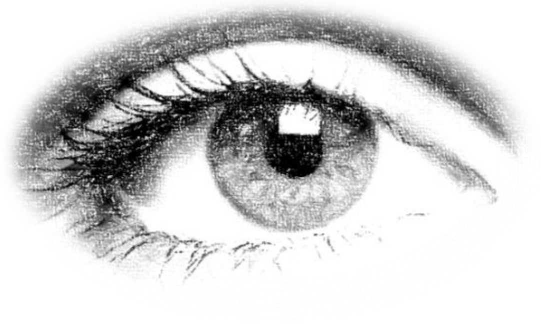




Hough Transform

- Clear the accumulator array
- For each detected edgel at location (x, y) and orientation $\theta = tan^{-1} n_y/n_x$, compute the value of $d = xn_x + yn_y$ and increment the accumulator corresponding to (θ, d)
- Find the peaks in the accumulator corresponding to lines
- Optionally re-fit the lines to the constituent edgels

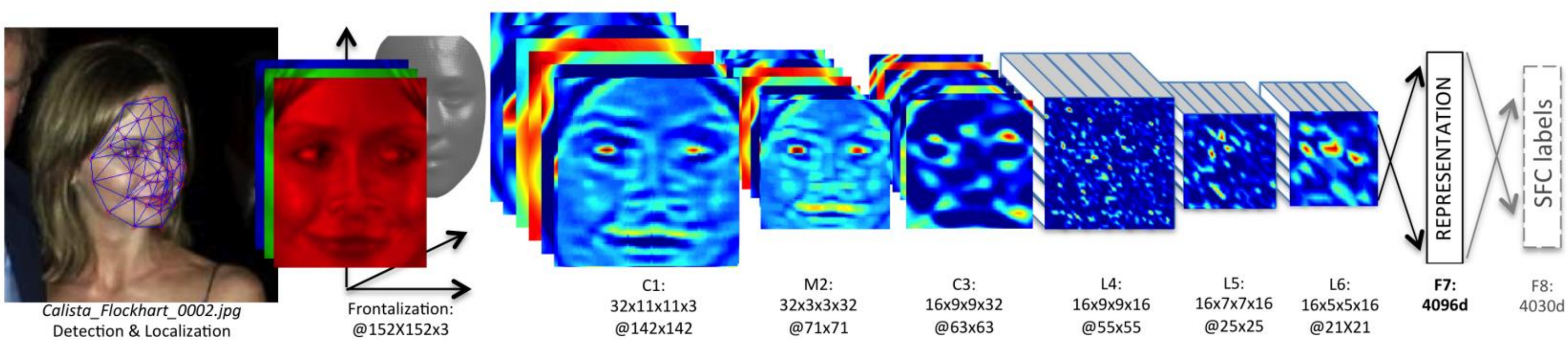




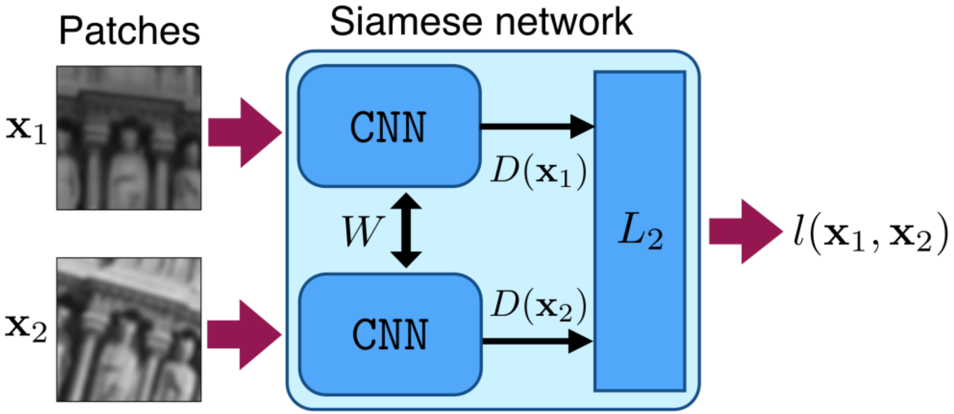
Deep Features

Deep Features

- Features extracted from Deep Neural Network
  - Ex. Deep Face (CVPR2014)



Deep Features



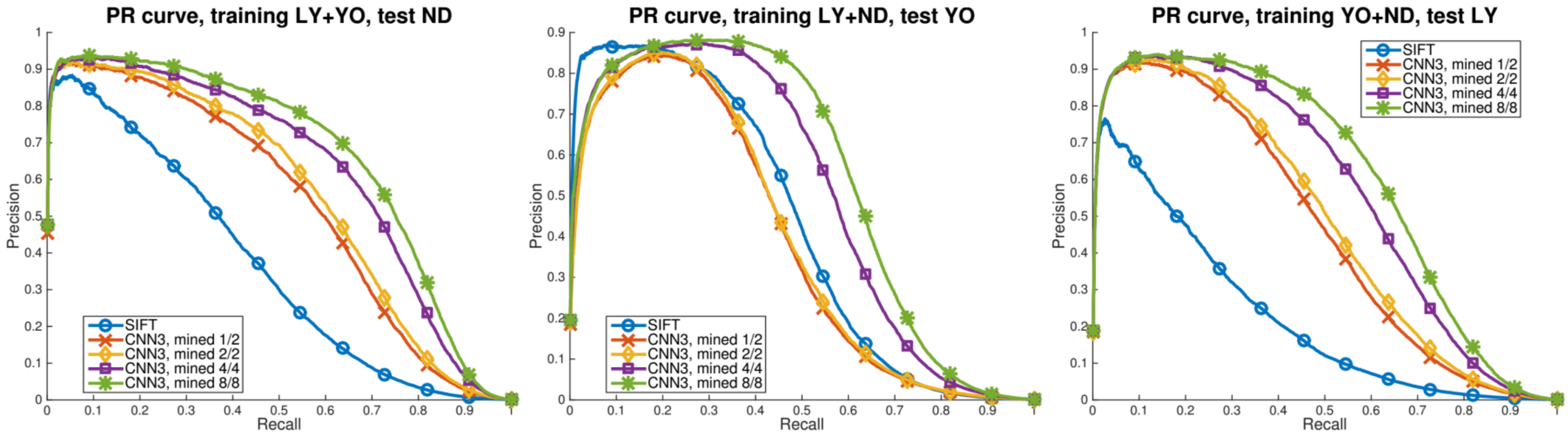
Layer	1	2	3
Input size	64×64	29×29	8×8
Filter size	7×7	6×6	5×5
Output channels	32	64	128
Pooling & Norm.tion	2×2	3×3	4×4
Nonlinearity	Tanh	Tanh	Tanh
Stride	2	3	4

Loss function:

$$l(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2, & p_1 = p_2 \\ \max(0, C - \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2), & p_1 \neq p_2 \end{cases}$$

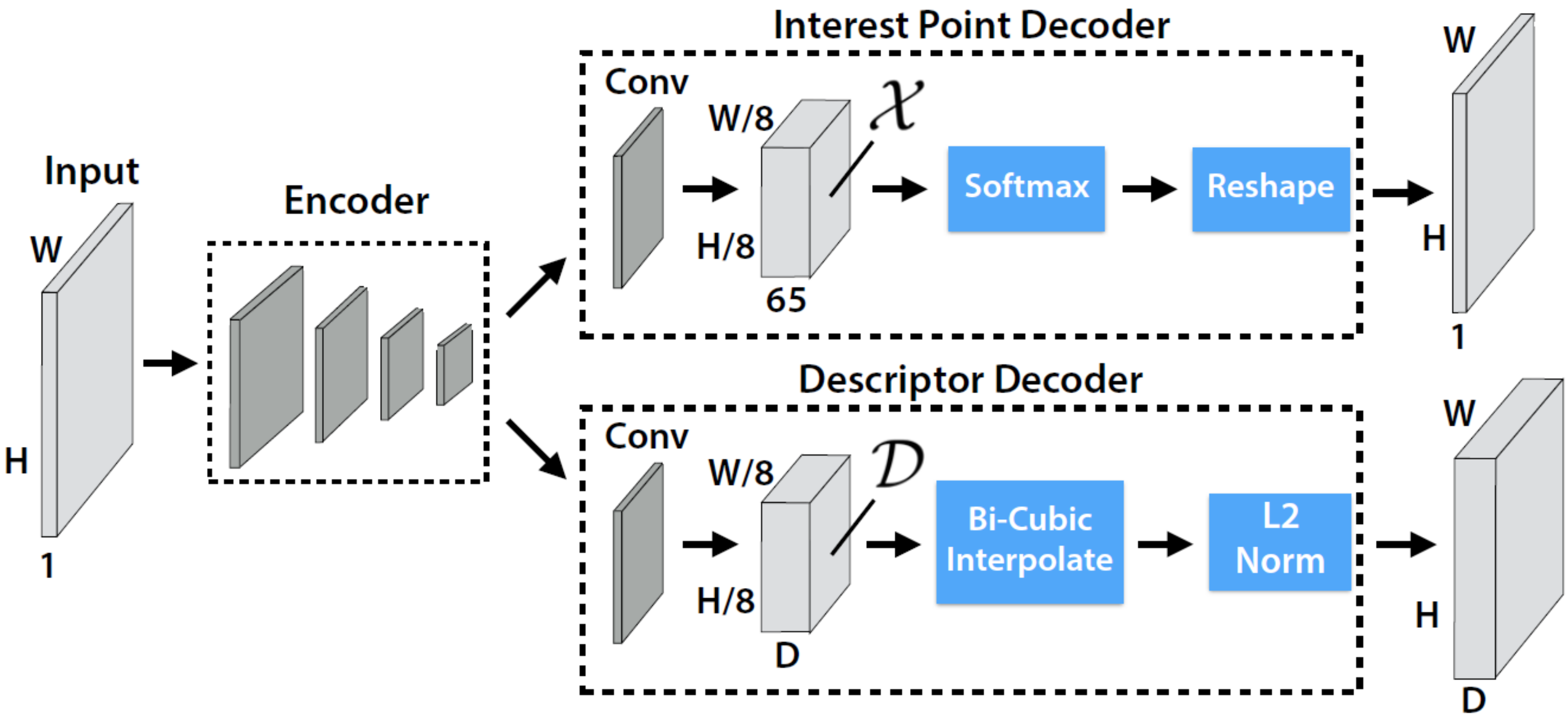
E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. ICCV 2015*.

Deep Features



E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. ICCV 2015*.

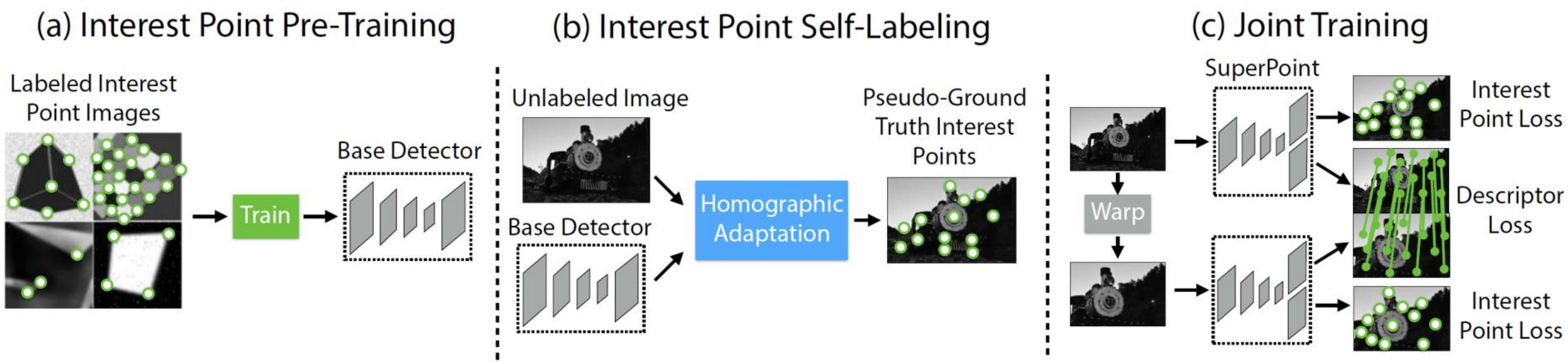
SuperPoint



Ref: D. DeTone, T. Malisiewicz and A. Rabinovich , "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

SuperPoint

- Self-supervised training



Ref: D. DeTone, T. Malisiewicz and A. Rabinovich , "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

Appendix: MPEG-7 Descriptors

Introduction

- **MPEG-7 is a standard for describing features of multimedia content**
- MPEG-7 provides the world's richest set of audio-visual descriptions
- Comprehensive scope of data interoperability
- Based on XML

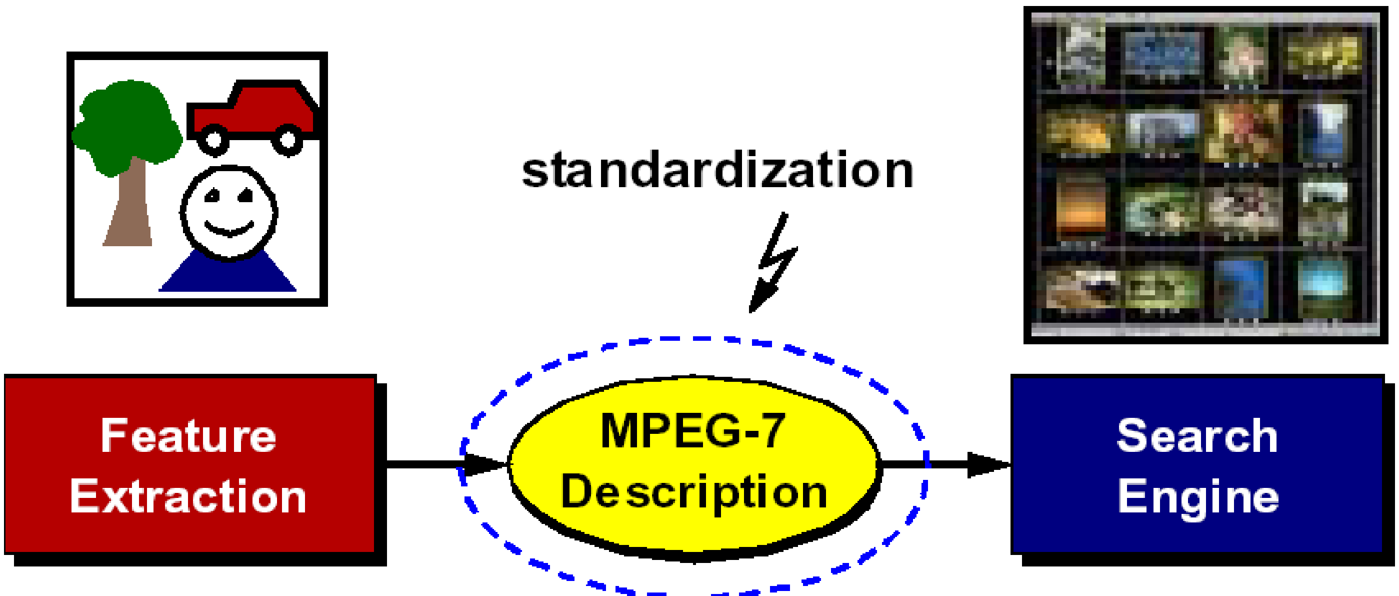
Introduction

- General visual descriptors
  - Color
  - Texture
  - Shape
  - Motion
- Domain-specific visual descriptors
  - Face recognition descriptors

What is Standardized?

- **Only define the descriptions**
  - Not standardize how to produce the descriptions
  - Not standardize how to use the descriptions
  - Only define what is needed for the interoperability of MPEG-7 enabled systems

What is Standardized?



Feature Extraction:
Content analysis (D, DS)
Feature extraction (D, DS)
Annotation tools (DS)
Authoring (DS)

MPEG-7 Scope:
Description Schemes (DSs)
Descriptors (Ds)
Language (DDL)
Ref: MPEG-7 Concepts

Search Engine:
Searching & filtering
Classification
Manipulation
Summarization Indexing

Color Descriptors

- Color Space Descriptor
- Dominant Color Descriptor
- Scalable Color Descriptor
- Group of Frames (or Pictures) Descriptor
- Color Structure Descriptor
- Color Layout Descriptor

Example: Dominant Color Descriptor (1)

- Compact description
- Browsing of image databases based on single or several color values
- Definition:
  - F = {($\mathbf{c}_i$, p_i, v_i), s}, (i = 1, 2, …, N) (N < 9)
  - $\mathbf{c}_i$: color value vector (default color space: RGB)
  - p_i : percentage ($\sum_i p_i = 1$)
  - v_i : optional color variance
  - s : spatial coherency

Dominant Color Descriptor (2)

- Binary syntax of DCD

Field	Number of Bits	Meaning
NumberofColors	3	Specifies number of dominant colors
SpatialCoherency	5	Spatial Coherency Value
Percentage[]	5	Normalized percentage associated with each dominant color
ColorVariance[][]	1	Color variance of each dominant color
Index[][]	1—12	Dominant color values

Dominant Color Descriptor (3)

- Extraction:
  - Clustering is performed in a perceptually uniform color space (Lloyd algorithm)
  - Distortion :

$$D_i = \sum_n h(n) \| x(n) - c_i \|^2, x(n) \in C_i$$

    - **x**(n) : the color vector at pixel n
    - h(n) : perceptual weight for pixel n
    - **c**$_i$: centroid of cluster C$_i$

$$c_i = \frac{\sum h(n) x(n)}{\sum h(n)}, x(n) \in C_i$$

Dominant Color Descriptor (4)

- Extraction:
  - The procedure is initialized with one cluster consisting of all pixels and one representative color computed as the centroid of the cluster
  - The algorithm then follows a sequence of centroid calculation and clustering steps until a stopping criterion (minimum distortion or maximum number of iterations)

Dominant Color Descriptor (5)

- Extraction:
  - Spatial coherency (s):
    - 4 connectivity connected component analysis
    - Individual spatial coherence: normalized average number of the connected pixels of each dominant color
    - s = $\sum_i p_i \cdot$ (individual spatial coherence)$_i$
    - S is nonuniformly quantized to 5 bits,
      31 means highest confidence
      1 means no confidence
      0 means not computed

Dominant Color Descriptor (6)

- Similarity Matching:
  - Number of representative colors is small, one can first search the database for each of the representative color separately, then combine.

Dominant Color Descriptor (7)

- Similarity Matching:
  - Consider 2 DCDs :
  - $$F_1 = \{(\mathbf{c}_{1i}, p_{1i}, v_{1i}), s_1\}, \quad (i = 1, 2, \ldots, N_1) \text{ and}$$
    $$F_2 = \{(\mathbf{c}_{2i}, p_{2i}, v_{2i}), s_2\}, \quad (i = 1, 2, \ldots, N_2).$$
  - Dissimilarity (D):

$$D^2(F_1, F_2) = \sum^{N_1} p_{1i}^2 + \sum^{N_2} p_{2j}^2 - \sum^{N_1} \sum^{N_2} 2a_{1i,2j} p_{1i} p_{2j}$$

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{\max} & d_{k,l} \le T_d \\ 0 & d_{k,l} > T_d \end{cases}$$

$d_{k,l}$: $d_{k,l} = \|c_k - c_l\|$ is the Euclidean distance between two colors

$$d_{\max} = \alpha T_d$$

Dominant Color Descriptor (8)

- Similarity Matching:
  - Dissimilarity (D_S):

$$D_S = w_1 \mathrm{abs}(s_1 - s_2) D + w_2 D$$

    $w_1 = 0.3$, $w_2 = 0.7$ (recommanded)
  - Dissimilarity (D_V):

$$D_V = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} p_{1i} p_{1j} f_{1i\ 1j} + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} p_{2i} p_{2j} f_{2i\ 2j} - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2 p_{1i} p_{2j} f_{1i\ 2j}$$

where
$$f_{x_i\ y_j} = \frac{1}{2\pi \sqrt{v_{x_i\ y_j}{}^{(l)} v_{x_i\ y_j}{}^{(u)} v_{x_i\ y_j}{}^{(v)}}} \times \exp\left[- \left(\frac{c_{x_i\ y_j}{}^{(l)}}{v_{x_i\ y_j}{}^{(l)}} + \frac{c_{x_i\ y_j}{}^{(u)}}{v_{x_i\ y_j}{}^{(u)}} + \frac{c_{x_i\ y_j}{}^{(v)}}{v_{x_i\ y_j}{}^{(v)}} \right) \Big/ 2 \right]$$

and
$$c_{x_i\ y_j}{}^{(l)} = (c_{x_i}{}^{(l)} - c_{y_j}^{(l)})^2,\ v_{x_i\ y_j}{}^{(l)} = (v_{x_i}{}^{(l)} + v_{y_j}{}^{(l)})$$

Dominant Color Descriptor (9)

- Similarity Matching Results:

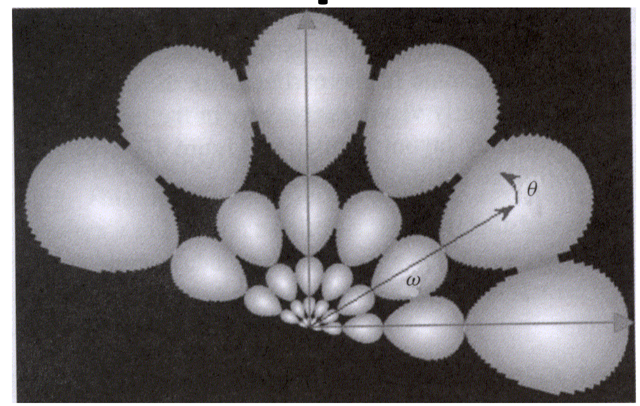
Table 13.3 ANMRR results for dominant color

Average number of colors	ANMRR(D)	ANMRR(D_S)	ANMRR(D_V)
3	0.31	0.30	0.25
5	0.25	0.21	0.16

Texture Descriptors

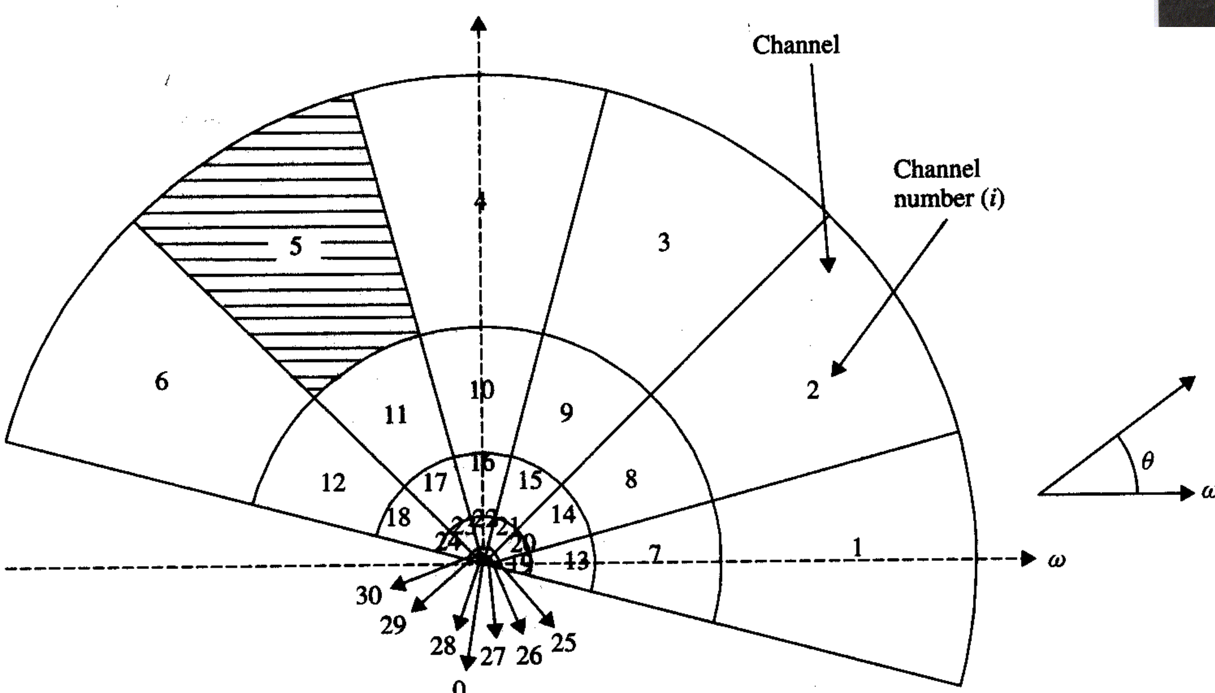
- Homogeneous Texture Descriptor (HTD)
- Texture Browsing Descriptor (TBD)
- Edge Histogram Descriptor (EHD)

Homogeneous Texture Descriptor



$$HTD = [\, f_{DC},\, f_{SD}, e_1, e_2, \ldots, e_{30}, d_1, d_2, \ldots d_{30}\,]$$

62 numbers (496 bits)



Texture feature channels modeled using the Gabor functions in the polar frequency domain

Channels used in computing the HTD

HTD Extraction

$$HTD = [\, f_{DC},\, f_{SD}, e_1, e_2, \ldots, e_{30}, d_1, d_2, \ldots d_{30}\,]$$

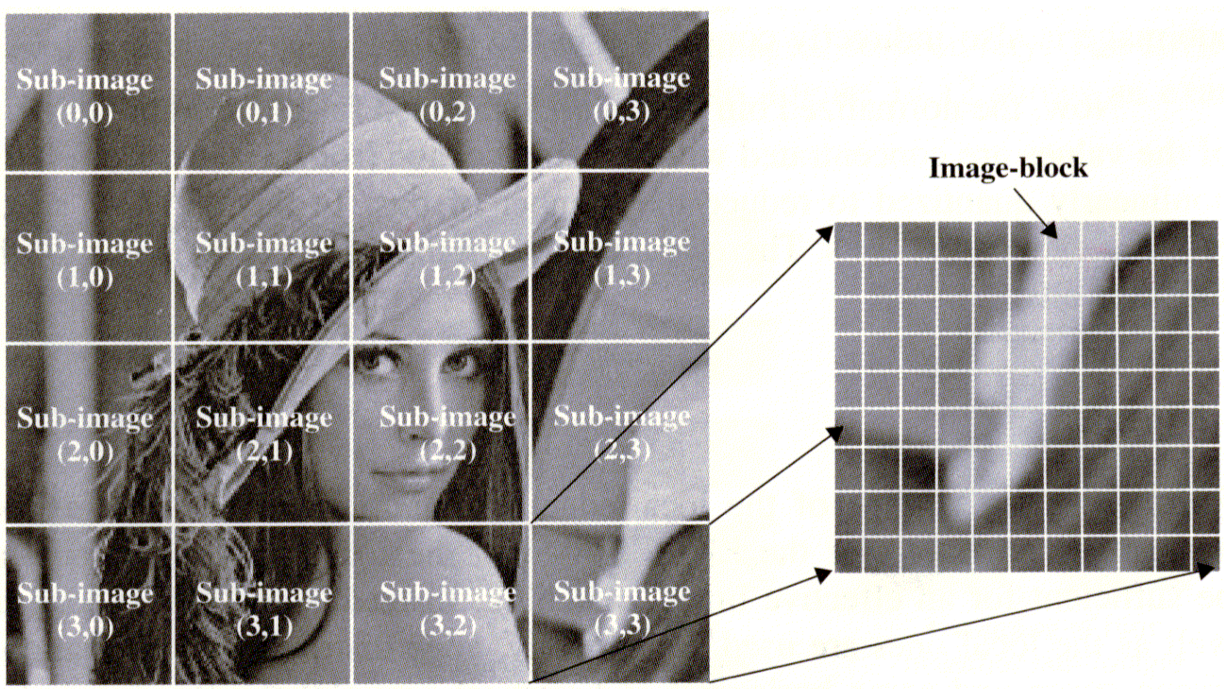
- On the basis of the frequency layout and the Gabor functions:
- e_i : log-scaled sum of the square of the Gabor-filtered Fourier transform coefficients of an image:

$$e_i = \log_{10}[1 + p_i]$$

$$where\ \ p_i = \sum_{\omega=0^+}^{1} \sum_{\theta=(0^\circ)^+}^{360^\circ} \left[G_{s,r}(\omega,\theta) |\omega| P(\omega,\theta) \right]^2$$

Edge Histogram Descriptor

- Divide the image into 4x4 subimages
- Block-based edge extraction



Semantics of the Histogram bins of the EHD

- 5 edge type x 16 subimages = 80 histogram bins

H_E	Semantics
h(0)	Relative population of vertical edges in subimage at (0,0)
h(1)	Relative population of horizontal edges in subimage at (0,0)
h(2)	Relative population of 45° edges in subimage at (0,0)
h(3)	Relative population of 135° edge in subimage at (0,0)
h(4)	Relative population of nondirectional edges in subimage at (0,0)
⋮	⋮
h(75)	Relative population of vertical edges in subimage at (3,3)
h(76)	Relative population of horizontal edges in subimage at (3,3)
h(77)	Relative population of 45° edges in subimage at (3,3)
h(78)	Relative population of 135° edges in subimage at (3,3)
h(79)	Relative population of nondirectional edges in subimage at (3,3)

Shape Descriptors

- Region-based descriptor
- Contour-based descriptor
- 3-D Shape Descriptor

Region v.s Contour (1/2)

Figure 15.1 Example of contour-based and region-based region similarity (© 2001 IEEE, from M. Bober; MPEG-7 Visual Descriptors, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, June 2001)

Region v.s Contour (2/2)

Figure 15.2 Example of shapes in which region-based shape is applicable

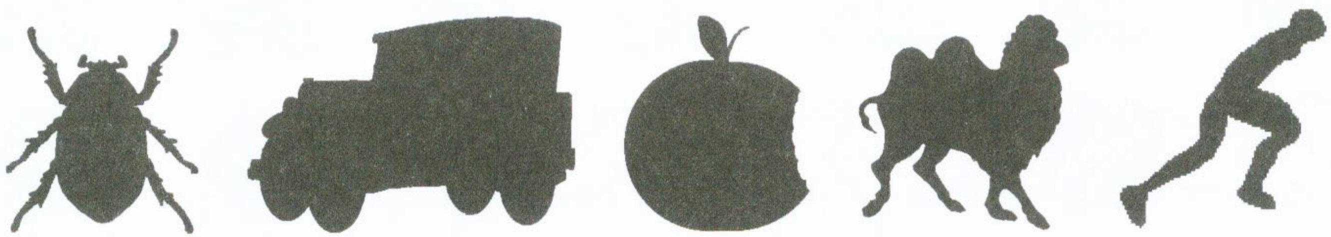


Figure 15.3 Examples of shapes in which contour-based shape is applicable

Goal of SDs

- Fast search and browsing
  - Concise manner
- Robust to
  - Scaling
  - Translation
  - Rotation

Region-based descriptor

- Take all pixels into account
- Project the shape onto the 2-D domain by using ART

Angular Radial Transform

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta)\rangle = \int_0^{2\pi}\int_0^1 V_{nm}^*(\rho,\theta)\, f(\rho,\theta)\,\rho\,\mathrm{d}\rho\,\mathrm{d}\theta$$

$$V_{nm}(\rho,\theta) = A_m(\theta)R_n(\rho)$$

$$A_m(\theta) = \frac{1}{2\pi}\exp(jm\theta)$$

$$R_n(\rho) = \begin{cases} 1 & n = 0 \\ 2\cos(\pi n\rho) & n \neq 0 \end{cases}$$

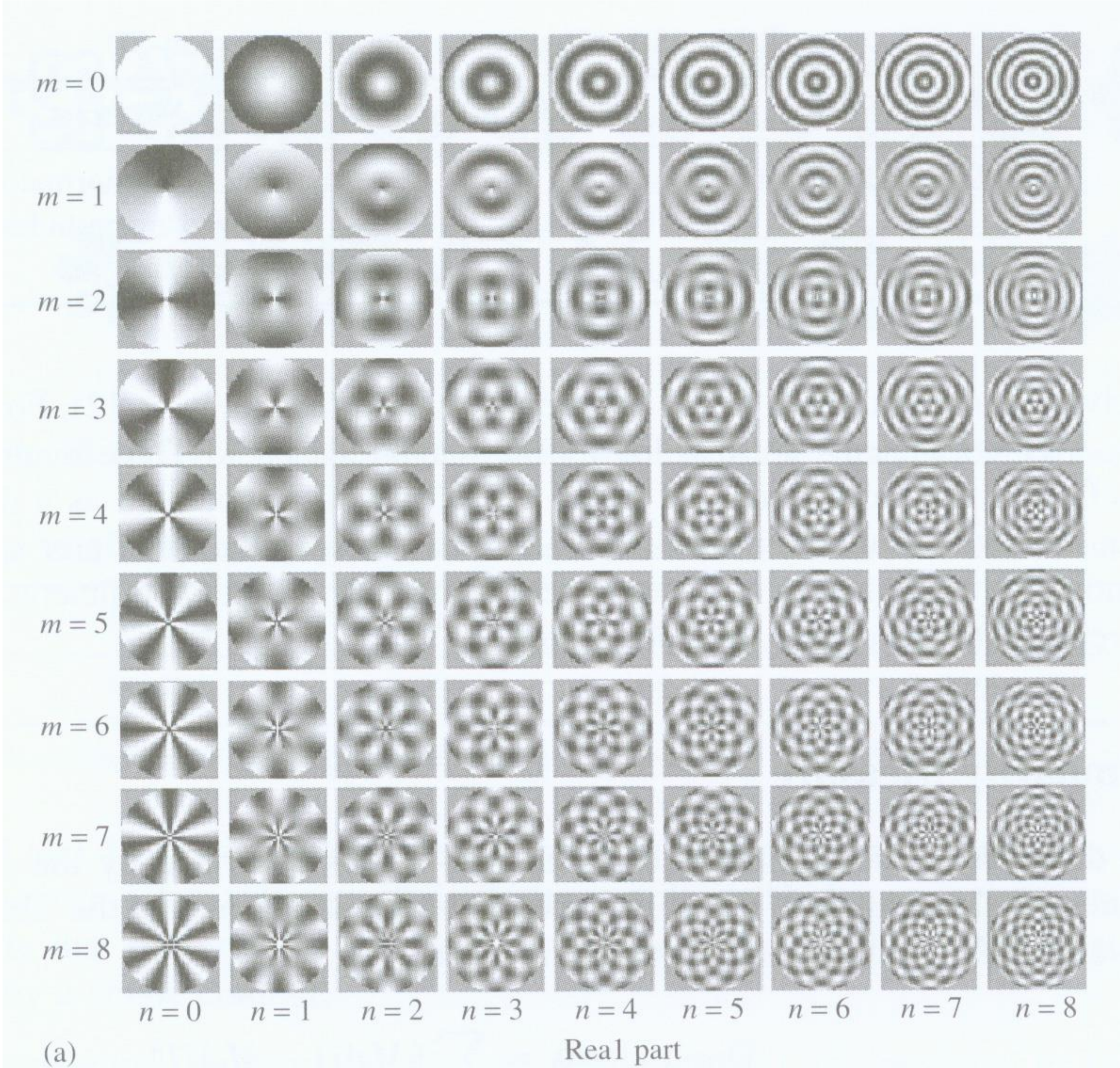
F: image function

V: ART basis

Define on unit circle in polar system

ART basis

Real part



ART basis

Imaginary part

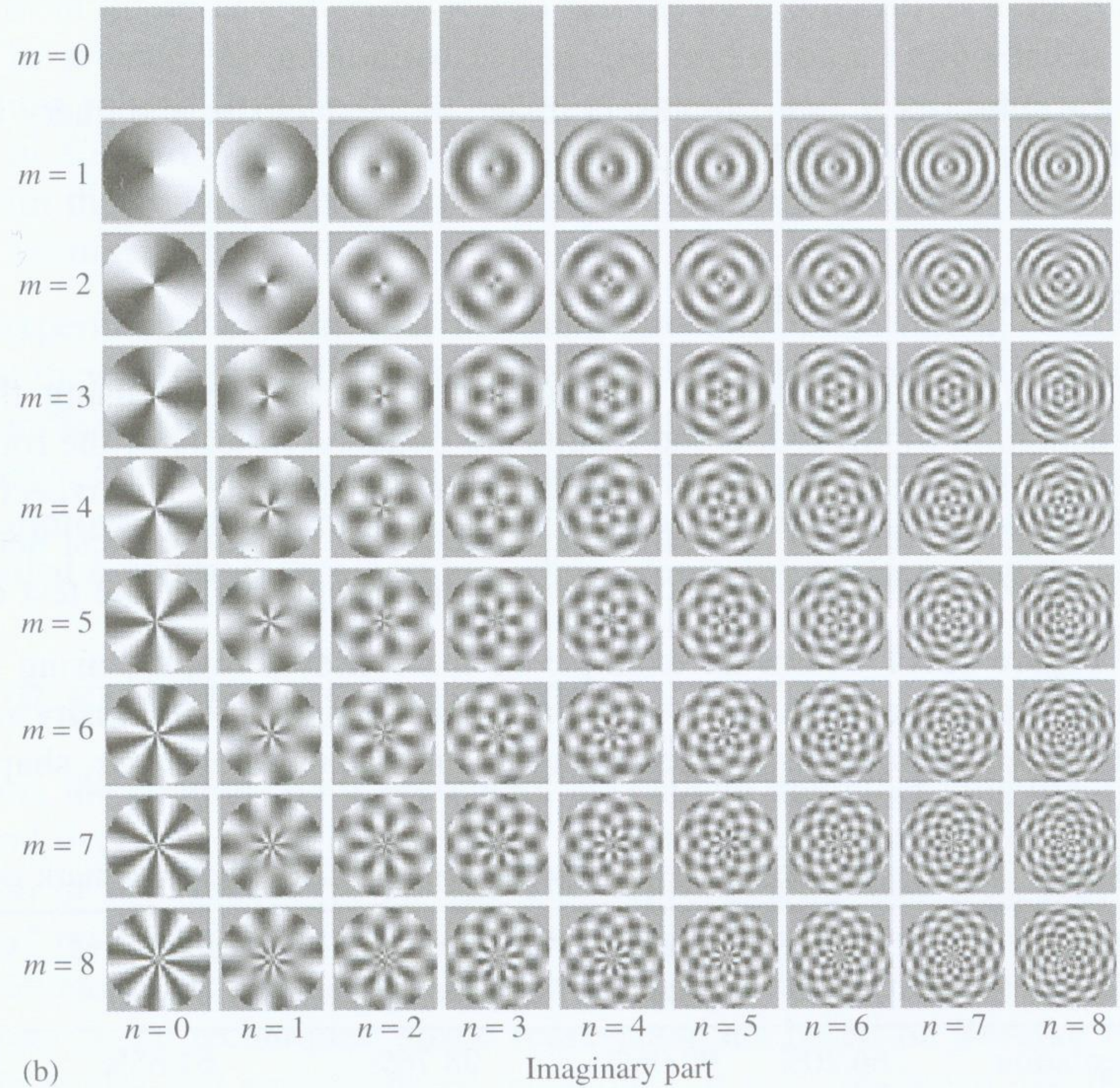


Figure 15.10 Real and imaginary parts of ART basis functions

Descriptor Representation

Table 15.1 Representation of the region-SD

Field	Number of bits	Meaning
Magnitude of ART[k]	4	An array of 35 normalized and quantized magnitudes of the shape coefficients.

Angular (m):0~11

Radial (n) :0~2

Normalize by F_{00}:

F_{nm}/F_{00} n=0~2,m=0~11

Contour-based descriptor

- Take only border pixels into account
- CSS representation

CSS Representation

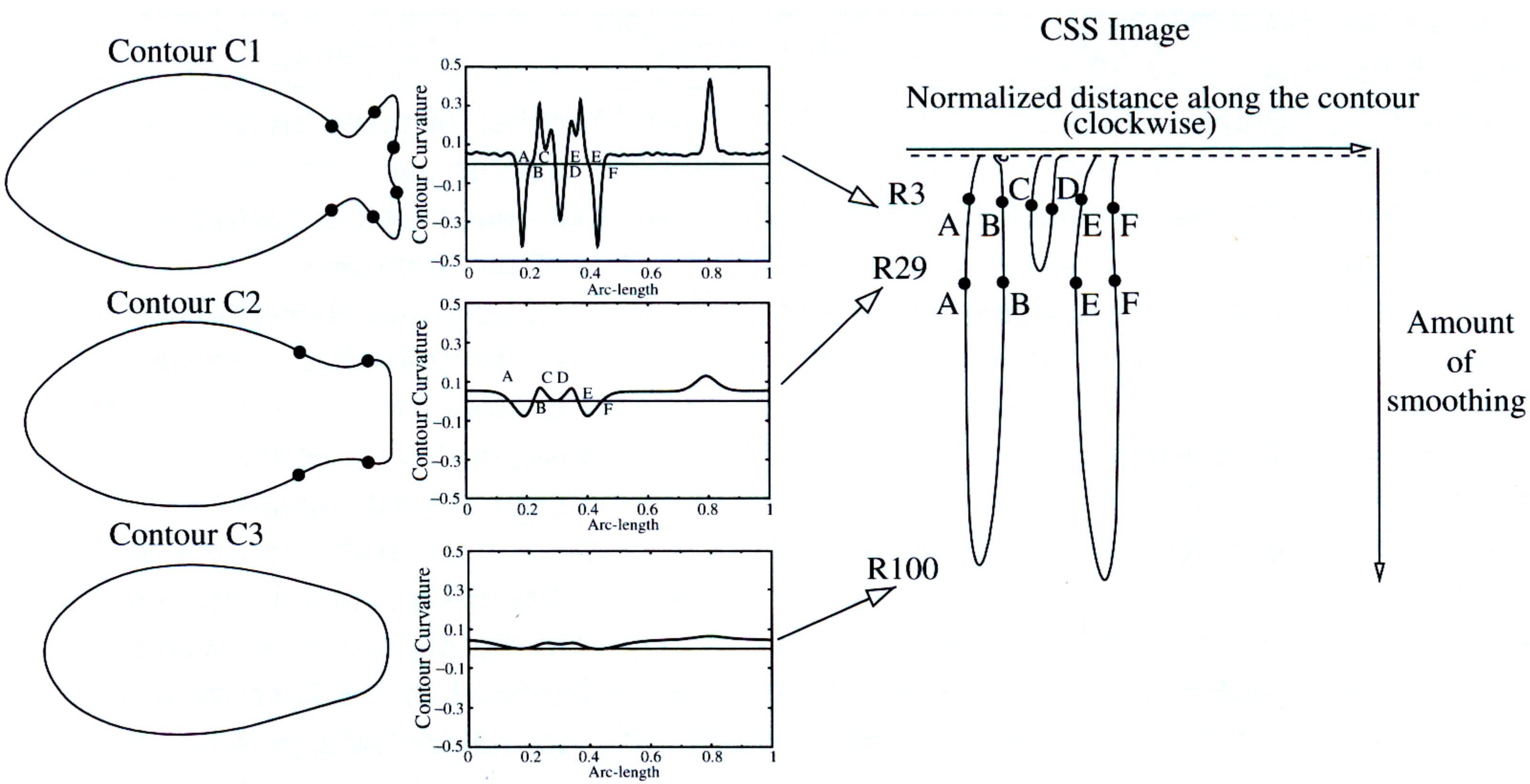


Figure 15.13 Extraction of the CSS image for the fish contour of Figure 15.12

Descriptor Representation

- Number of Peaks [6 bits]
- Global Curvature [2*6 bits]
- Prototype Curvature [2*6 bits]
- Highest Peak Y [7 bits]
- Peak X [6 bits]
- Peak Y [3 bits]

Motion Descriptors

- Motion Activity Descriptor
- Camera Motion Descriptor
- Motion Trajectory Descriptor
- Parametric Motion Descriptor