Epipolar Geometry, Feature Detection, and Feature Matching

Multi-View Geometry

- Different views of a scene are not unrelated
- Relationships exist between two, three and more cameras
- *Question: Given an image point in one image, how does this restrict the position of the corresponding image point in another image?*
- Reference: R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000

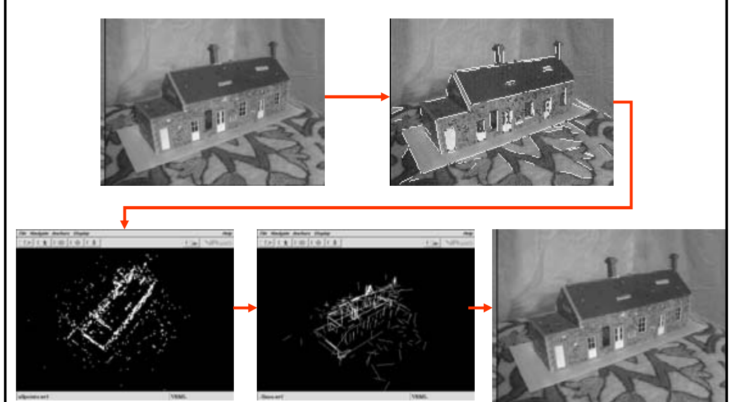
Three Questions

1. **Correspondence geometry:** Given an image point *X* in the first view, how does this constrain the position of the corresponding point *X´* in the second image?
2. **Camera geometry (motion):** Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, i=1,…,*n*, what are the cameras P and P´ for the two views?
3. **Scene geometry (structure):** Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras P, P´, what is the position of (their pre-image) *X* in the 3D world?

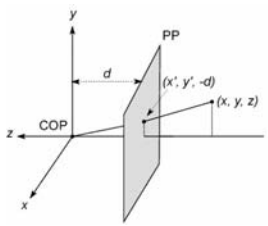
Applications: Stereopsis, 3D scene reconstruction, making panoramic images, structure from motion

Multi-View Stereo

[Fitzgibbon and Zisserman, 1998]



Modeling Camera Projection

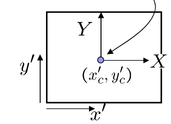


- The coordinate system
  - Pinhole camera model as an approximation
  - Put the pinhole (aka optical center, center of projection) at the origin
  - Put the image plane (projection plane) *in front* of the optical center

Camera Parameters

- A camera is described by several parameters
  - Translation T of the optical center from the origin of world coords
  - Rotation R of the image plane
  - focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
  - blue parameters are called "extrinsics," red are "intrinsics"
- Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$

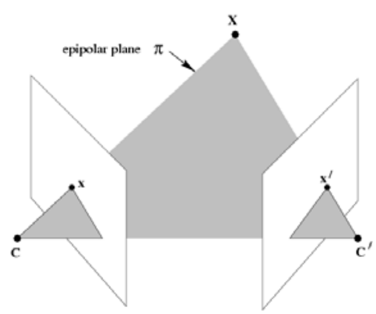


  - The projection matrix models the cumulative effect of all parameters
  - Useful to decompose into a series of operations

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

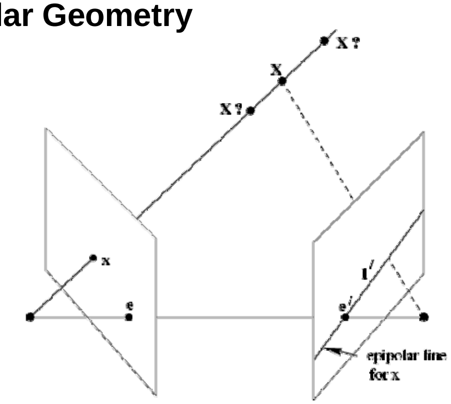
identity matrix

Epipolar Geometry



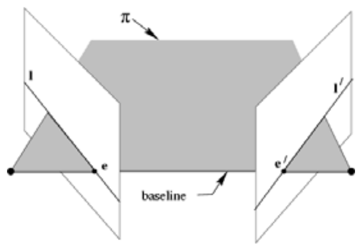
Co-Planarity Constraint: *C*, *C´*, *x*, *x´* and *X* are co-planar

Epipolar Geometry



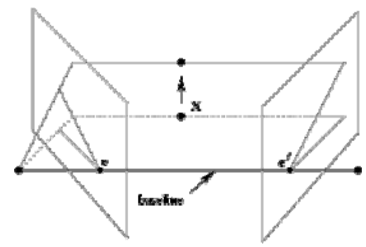
What if only C, C´, and *x* are known?
Answer: *x´* constrained to lie on **epipolar line** *l´*

Epipolar Geometry



All points on π project onto **epipolar lines** *l* and *l*′

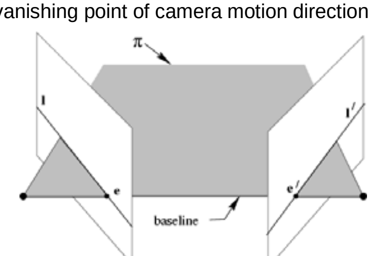
Epipolar Geometry



Family of planes π and lines *l* and *l*′ intersect in **epipoles** *e* and *e*′

Epipolar Geometry

epipoles *e*, *e*′
= intersection of baseline with image plane
= projection of optical center in other image
= vanishing point of camera motion direction

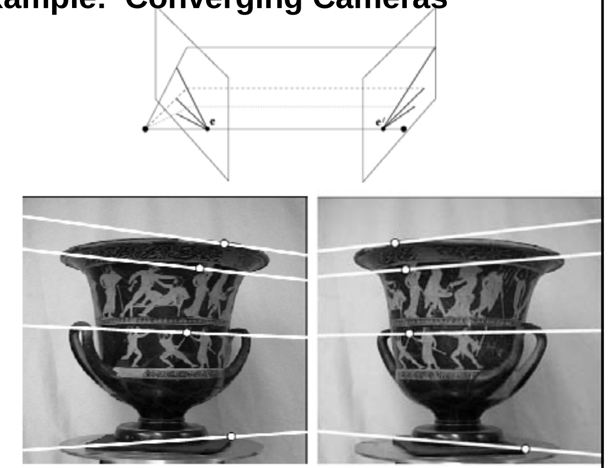


an epipolar plane = plane containing baseline (1D family)

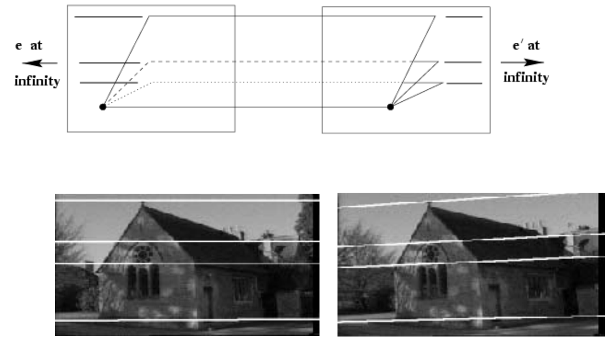
an epipolar line = intersection of epipolar plane with image
(always come in corresponding pairs)

- **Correspondence geometry:** Given an image point *x* in the first view, how does this constrain the position of the corresponding point *x*′ in the second image?
- **Epipolar geometry constrains search for x′ from 2D to 1D**

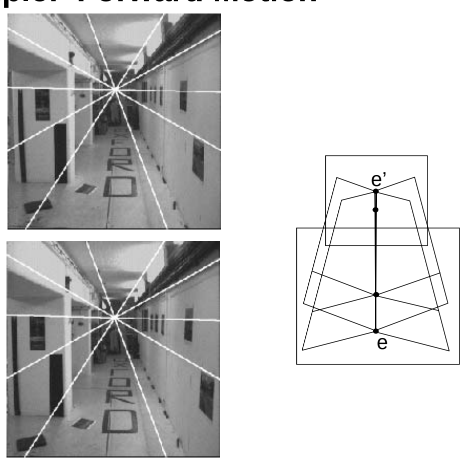
Example: Converging Cameras



Example: Parallel Cameras



Example: Forward Motion



Computing the Epipolar Geometry

- Given a scene point, define it in terms of 2 vectors wrt left and right cameras

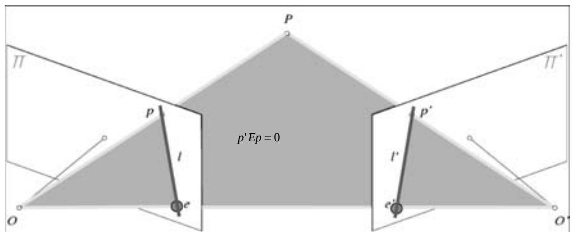
$$P' = RP + t$$

$$p = MP \quad \text{with} \quad M = [I \mid 0]$$

$$p' = M'P' \quad \text{with} \quad M' = [R \mid t]$$

- Co-planarity constraint: *P*, *P´* and *t* are co-planar, so their mixed product = 0

Calibrated Camera

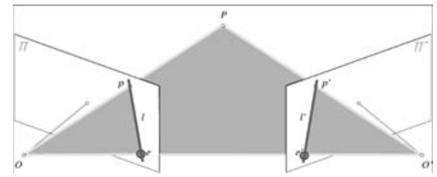


$\vec{0p}, \vec{00'}, \vec{0'p'}$ are co-planar ⇒ $p'[t\times(Rp)]=0$ with $\begin{cases} p=(u,v,1)^T \\ p'=(u',v',1)^T \end{cases}$

⇓

E = Essential matrix $\quad p'Ep=0 \quad$ with $\quad E=[t_\times]R=SR$

Uncalibrated Camera



p and p' in pixel coordinates correspond to $\hat{p}$ and $\hat{p}'$ in camera coordinates

$\hat{p} = M_{\text{int}}^{-1}\, p$ and $\hat{p}' = M'^{-1}_{\text{int}}\, p'$ ⇒ $p'^T\, F\, p = 0$

$\hat{p}'\, E\, \hat{p} = 0$

with $F = M'^{-T}_{\text{int}}\, E\, M^{-1}_{\text{int}}$

F = Fundamental matrix

Properties of Fundamental and Essential matrices

- Matrix is 3 x 3
- **Transpose:** If F is fundamental matrix of cameras (P, P′), then F^T is fundamental matrix of camera (P′,P)
- **Epipolar lines:** Think of *p* and *p′* as points in the projective plane. Then *F p* is projective line in the right image.
  That is $l' = F p \quad l = F^T p'$
- **Epipoles:** Since for any p the epipolar line $l' = F p$ contains the epipole e', so $(e'^T F)\, p=0$ for all *p*. Thus $e'^T F=0$ and $F e = 0$

Essential matrix, E

- Encodes information on the extrinisic camera parameters only
- E is of rank 2, since S has rank 2 (and R has full rank)
- Has only 5 degrees of freedom: 3 for rotation, 2 for translation (*t* can only be recovered up to a scale factor, meaning only the direction of translation can be obtained)

Fundamental matrix, F

- F is the unique 3x3 rank 2 matrix that satisfies $x'^T F x = 0$ for all $x \leftrightarrow x'$
- Encodes information on the intrinsic and extrinsic camera parameters
- F is of rank 2, since S has rank 2 (R and M and M′ have full rank)
- Has 7 degrees of freedom
  (There are 9 elements, but scaling is not significant and det F = 0)

Computing Fundamental Matrix from Point Correspondences

- The fundamental matrix is defined by the equation $\mathbf{x}'^{\mathrm{T}}_i \mathbf{F} \mathbf{x}_i = 0$ for any pair of corresponding points $\mathbf{x}_i$ and $\mathbf{x}'_i$ in the 2 images
- The equation for a pair of points $(x, y, 1)$ and $(x', y', 1)$ is: $x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0$
- For n point matches:

$$\mathbf{A}\,\mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0$$

Computing Fundamental Matrix from Point Correspondences

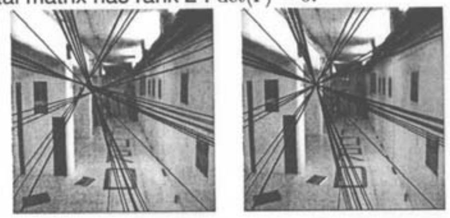
- We have a homogeneous set of equations $\mathbf{A}\,\mathbf{f} = 0$
- $\mathbf{f}$ can be determined only up to a scale, so there are 8 unknowns, and at least 8 point matchings are needed
  - hence the name "8 point algorithm"
- The least square solution is the singular vector corresponding the smallest singular value of $\mathbf{A}$, i.e. the last column of $\mathbf{V}$ in the SVD $\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^{\mathrm{T}}$

Normalized 8-Point Algorithm [Hartley, 1995]

1. Normalization: Center the image data at the origin and scale it so the mean squared distance between the origin and the data points is 2 pixels: $q_i = Tp_i$ and $q'_i = T'p'_i$
2. Solve linear system to compute F from conjugate pairs q_i and q'_i
3. Enforce rank-2 constraint by finding closest singular F′ to F
4. Denormalization: Output $F = T^{T} F' T'$

The singularity constraint

Fundamental matrix has rank 2 : $\det(F) = 0$.



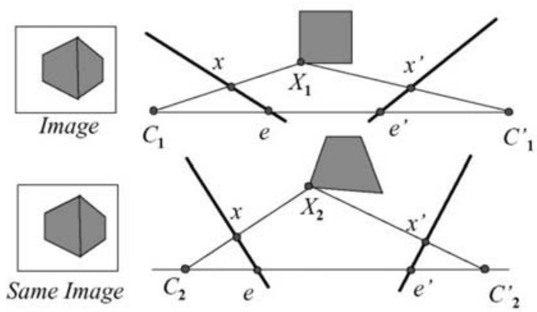
Left : Uncorrected F – epipolar lines are not coincident.

Right : Epipolar lines from corrected F.

Projective Reconstruction Theorem

- Assume we determine matching points x_i and x'_i. Then we can compute a unique fundamental matrix F
- The camera matrices M, M′ cannot be recovered uniquely
- Thus the reconstruction (X_i) is not unique
- There exists a projective transformation H such that

$$X_{2,i} = HX_{1,i,} \quad M_2 = M_1H^{-1} \quad M'_2 = M'_1H^{-1}$$



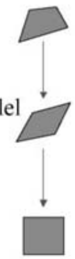
Projective Reconstruction Theorem (Consequences)

- We can compute a projective reconstruction of a scene from 2 views based on image correspondences alone
- We don't have to know anything about the calibration or poses of the cameras
- The true reconstruction is within a projective transformation **H** of the projective reconstruction: $\mathbf{X_{2i}} = \mathbf{H}\,\mathbf{X_{1i}}$

8

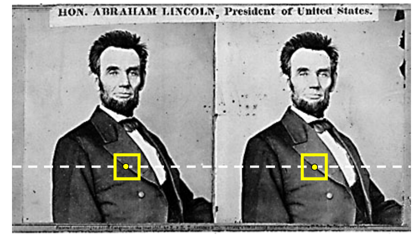
Stratified Reconstruction

- Begin with a projective reconstruction
- Refine it to an affine reconstruction
  - Parallel lines are parallel; ratios along parallel lines are correct
  - Reconstructed scene is then an affine transformation of the actual scene
- Then refine it to a metric reconstruction
  - Angles and ratios are correct
  - Reconstructed scene is then a scaled version of actual scene



12

3D Scene Reconstruction: Basic Stereo Algorithm



For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

Finding correspondences is relatively easy when baseline is small

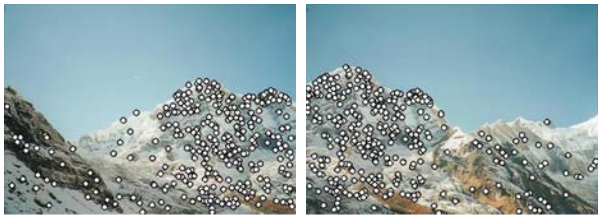
State of the Art in 3D Reconstruction (Structure from Motion)

- M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual modeling with a hand-held camera, *Int. J. Computer Vision* **59**(3), 2004
- M. Pollefeys and L. Van Gool. From images to 3D models, *Comm. ACM* **45**(7), 2002

M. Pollefeys and L. Van Gool, 3D from Video

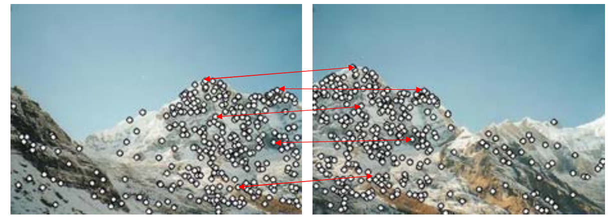
Wide Baseline Matching

- Camera networks usually have cameras that are far apart, making correspondence problem very difficult
- Feature-based approach: Detect feature points in both images



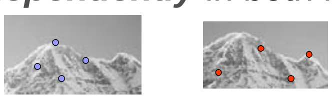
Matching with Features

- Detect feature points in both images
- Find corresponding pairs



Matching with Features

- Problem 1:
  - Detect the ***same*** point ***independently*** in both images

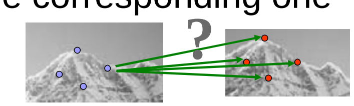


no chance to match!

We need a repeatable **detector**

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one



We need a reliable and distinctive **descriptor**

Properties of an Ideal Feature

- Local: features are local, so robust to occlusion and clutter (no prior segmentation)
- Invariant (or covariant) to many kinds of geometric and photometric transformations
- Robust: noise, blur, discretization, compression, etc. do not have a big impact on the feature
- Distinctive: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Accurate: precise localization
- Efficient: close to real-time performance

Applications

- Wide baseline matching without scene segmentation

Applications

- Recognition of specific objects

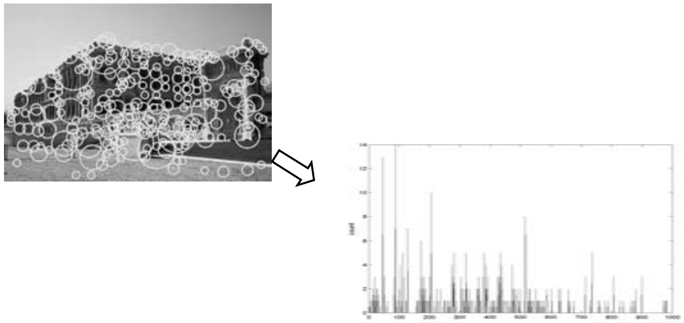
Rothganger et al. '03 Lowe et al. '02 Ferrari et al. '04

Applications

- Object class recognition
  - Bag-of-Word models
  - Constellation (graph) models

Recognition of object classes

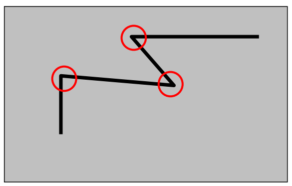
- Bag-of-visual-words image representation:



Recent Work on Feature Detectors

- Hessian
- Harris
- Lowe: SIFT (DoG)
- Mikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuytelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brady: Salient Regions
- Others

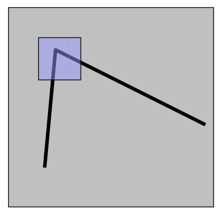
Harris “Corner”/Interest Point Detector



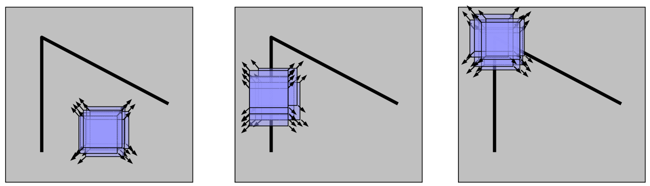
C. Harris, M. Stephens, “A Combined Corner and Edge Detector,” 1988

Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in response



Harris Detector: Basic Idea



“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in *all* directions

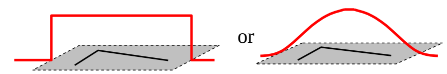
Harris Detector: Mathematics

Change of intensity for the shift [u,v]:

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

Window function — Shifted intensity — Intensity

Window function $w(x,y)$ =



1 in window, 0 outside — or — Gaussian

Harris Detector: Mathematics

Expanding $E(u,v)$ in a 2^{nd} order Taylor series, we have, for small shifts, [u,v], a *bilinear* approximation:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

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

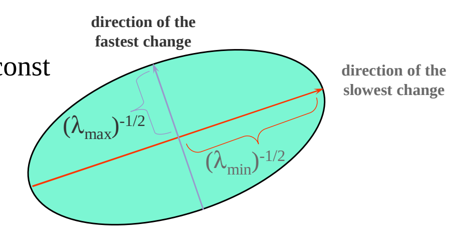
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

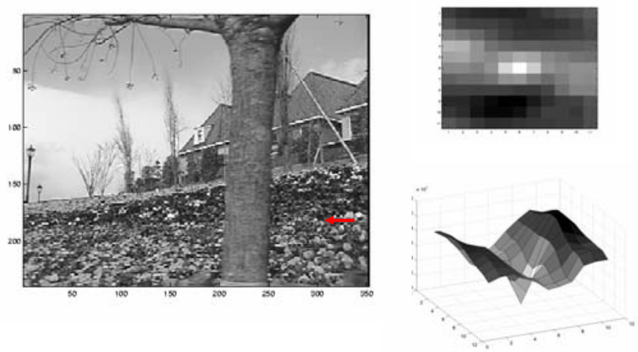
$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

λ_1, λ_2 – eigenvalues of M

Ellipse $E(u,v)$ = const

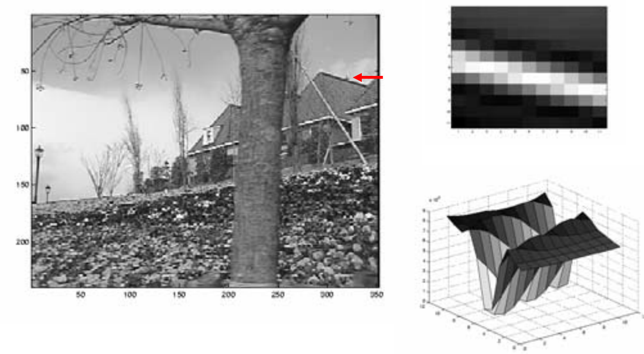


Selecting Good Features



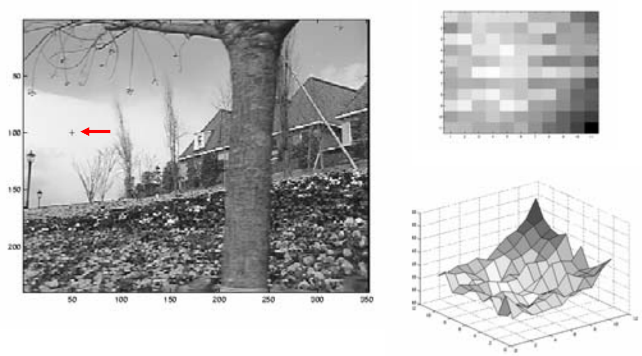
λ_1 and λ_2 both large

Selecting Good Features



large λ_1, small λ_2

Selecting Good Features

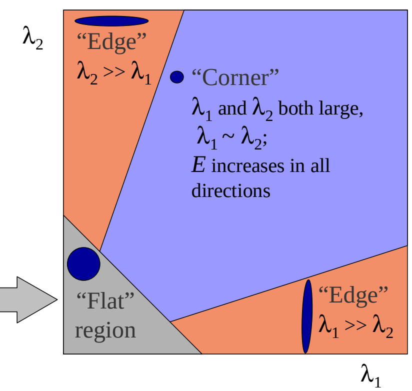


small λ_1, small λ_2

Harris Detector: Mathematics

Classification of image points using eigenvalues of M:

λ_1 and λ_2 are small; E is almost constant in all directions



Harris Detector: Mathematics

Measure of corner response:

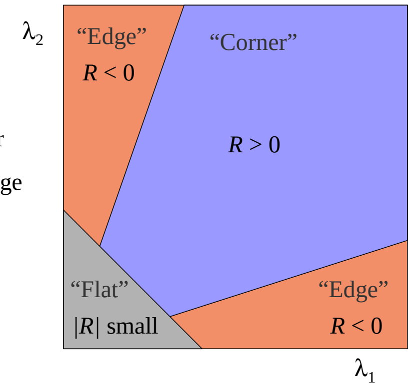
$$R = \det M - k\,(\operatorname{trace} M)^2$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

k is an empirically-determined constant; e.g., $k = 0.05$

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



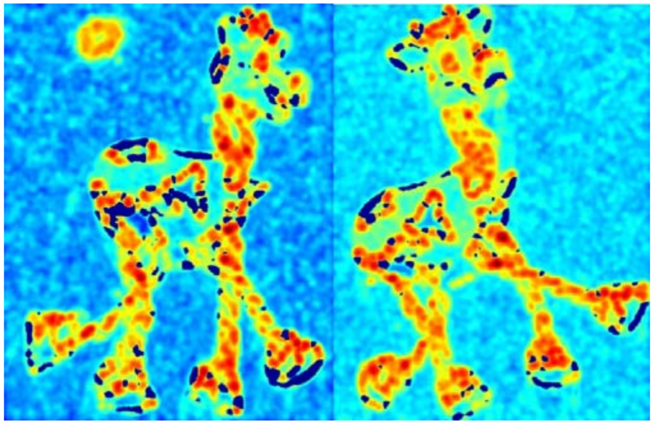
Harris Detector

- Algorithm:
  - Find points with large corner response function R (R > threshold)
  - Take the points of local maxima of R (for localization)

Harris Detector: Example

Harris Detector: Example

Compute corner response R

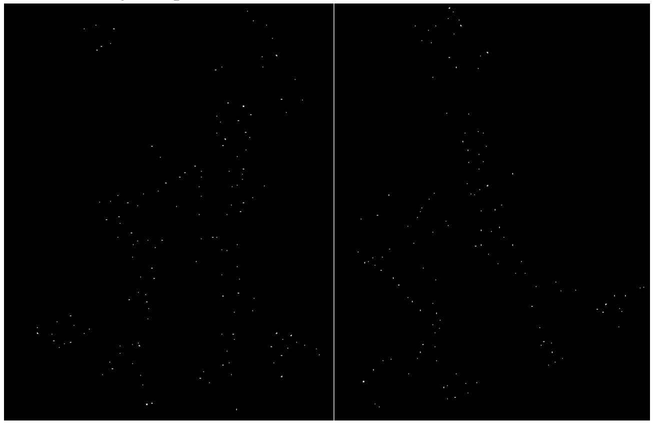


Harris Detector: Example

Find points with large corner response: $R >$ threshold

Harris Detector: Example

Take only the points of local maxima of R

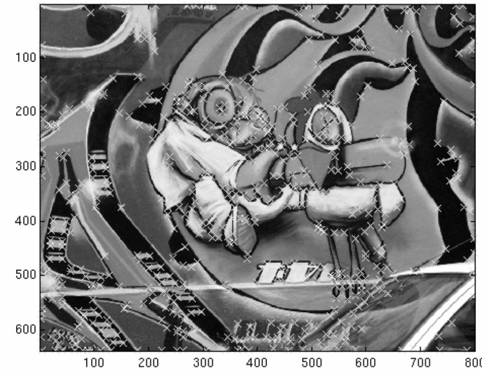


Harris Detector: Example

Harris Detector: Example

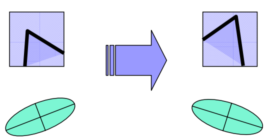
Interest points extracted with Harris (~ 500 points)

Harris Detector: Example



Harris Detector: Some Properties

- Rotation invariance

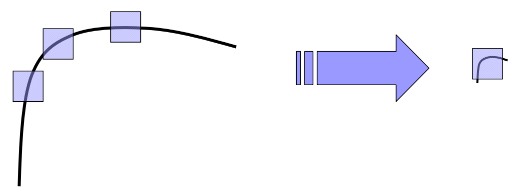


Ellipse rotates but its shape (i.e., eigenvalues) remains the same

Corner response R is invariant to image rotation

Harris Detector Properties: Scale Changes

- But **not** invariant to *image scale*



Fine scale: All points will be classified as edges

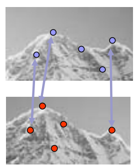
Coarse scale: Corner

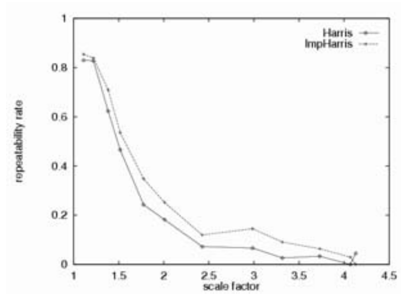
Harris Detector: Some Properties

- Quality of Harris detector for different scale changes

Repeatability rate:

$$\frac{\text{\# correct correspondences}}{\text{\# possible correspondences}}$$

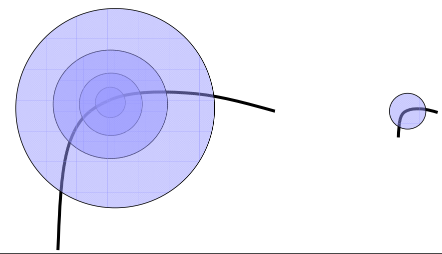




C. Schmid et al., "Evaluation of Interest Point Detectors," IJCV 2000

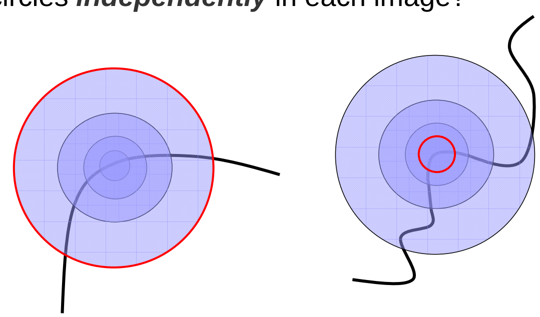
Scale Invariant Detection

- Consider regions (e.g., circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

- Problem: How do we choose corresponding circles *independently* in each image?

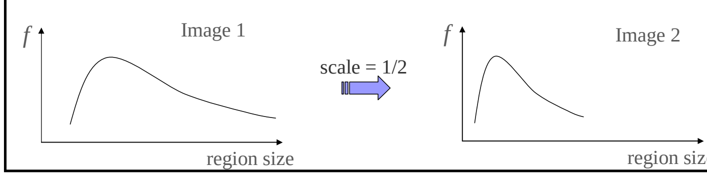


Scale Invariant Detection

- Solution:
  - Design a function on the region (circle) that is "scale invariant," i.e., the same for corresponding regions, even if they are at different scales

Example: Average intensity. For corresponding regions (even of different sizes) it will be the same

– For a point in one image, we can consider it as a function of region size (circle radius)

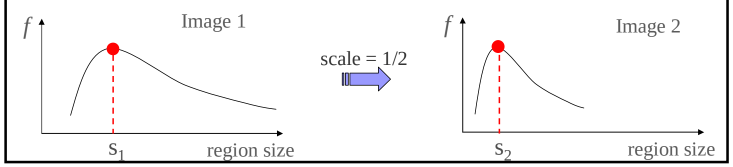


Scale Invariant Detection

- Common approach: Take a local maximum of this function

Observation: Region size, for which the maximum is achieved, should be *invariant* to image scale

Important: This scale invariant region size is found in each image **independently**!



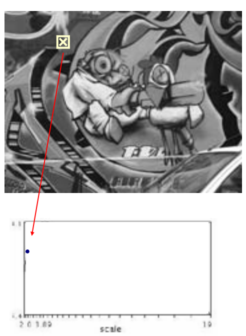
Automatic Scale Selection

Lindeberg et al., 1996

$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

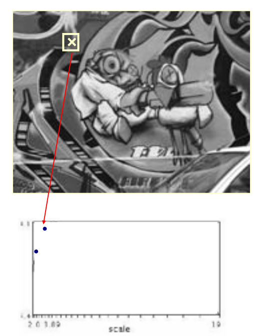
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

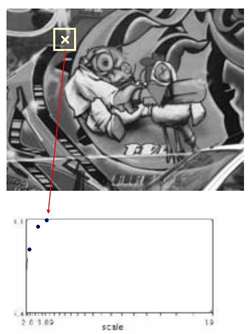
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

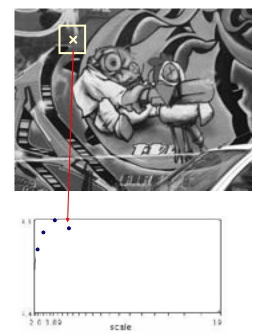
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

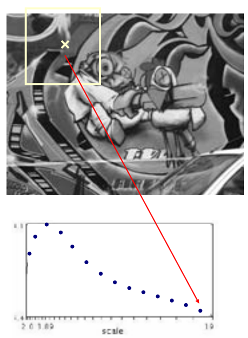
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

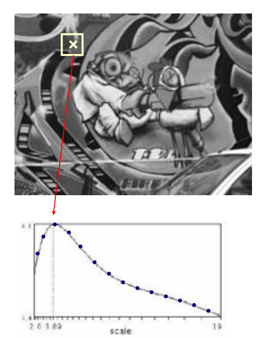
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

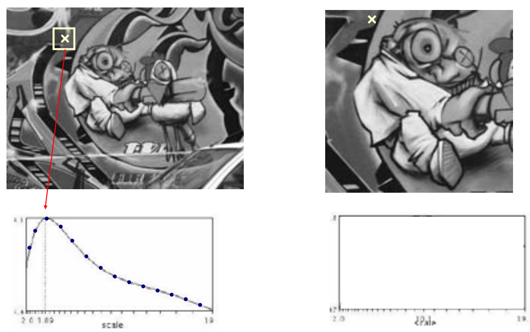
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$

Automatic Scale Selection

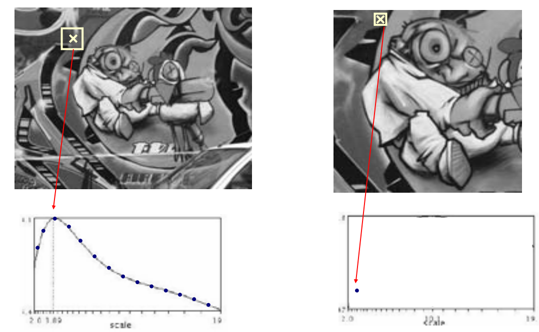
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

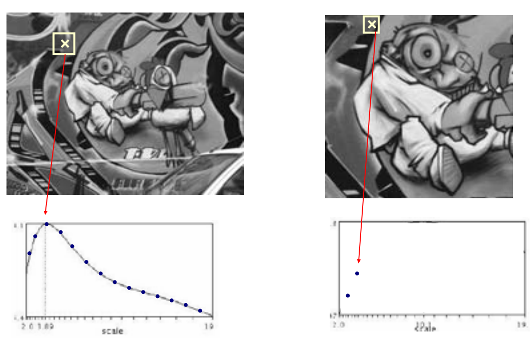
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

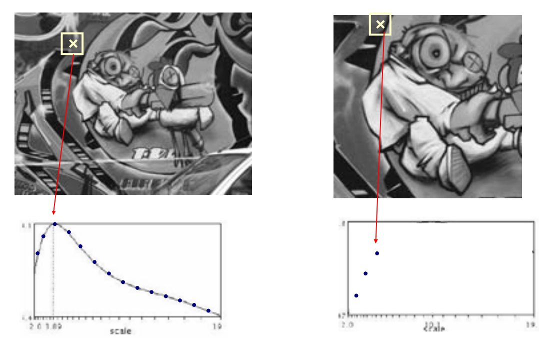
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

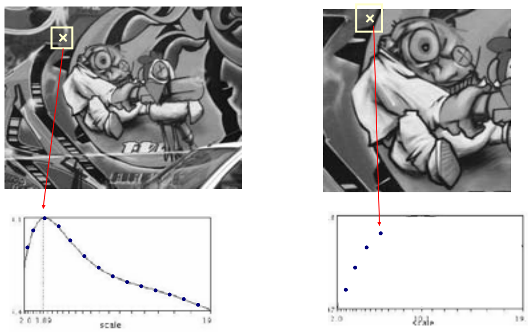
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \ldots i_m}(x,\sigma))$ $f(I_{i_1 \ldots i_m}(x',\sigma))$

Automatic Scale Selection

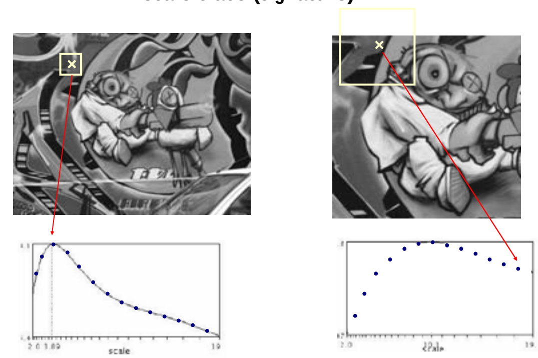
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Automatic Scale Selection

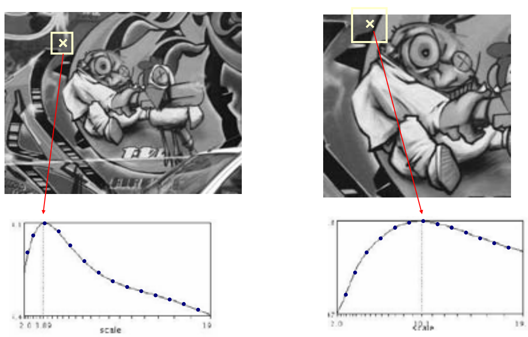
Function responses for increasing scale
Scale trace (signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Automatic Scale Selection

Function responses for increasing scale
Scale trace (signature)



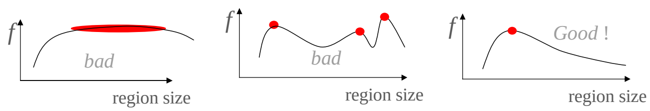
$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma'))$

Automatic Scale Selection

- Normalize: rescale to fixed size

Scale Invariant Detection

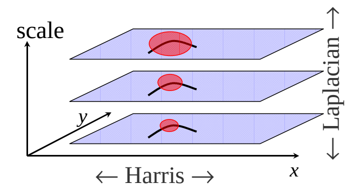
- A “good” function for scale detection has one stable sharp peak



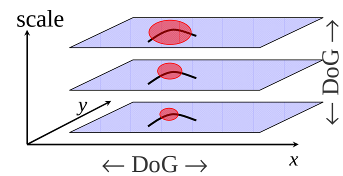
- For many images: a good function would be a one that responds to contrast (sharp local intensity change)

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maxima of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale



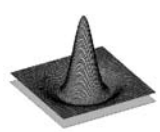
- SIFT keypoints[2]
  Find local extrema of:
  - Difference of Gaussians in space and scale

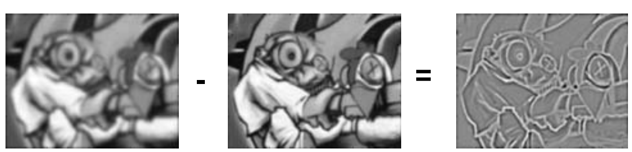


[1] K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points,” ICCV 2001
[2] D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints,” *IJCV*, 2004

Lowe’s SIFT (DoG) Detector

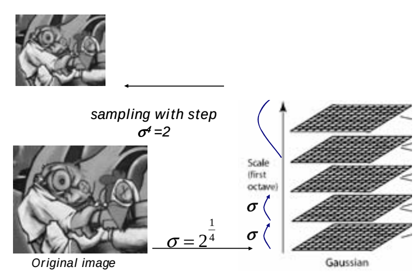
- Difference-of-Gaussian (DoG) as approximation of the Laplacian-of-Gaussian (LoG)

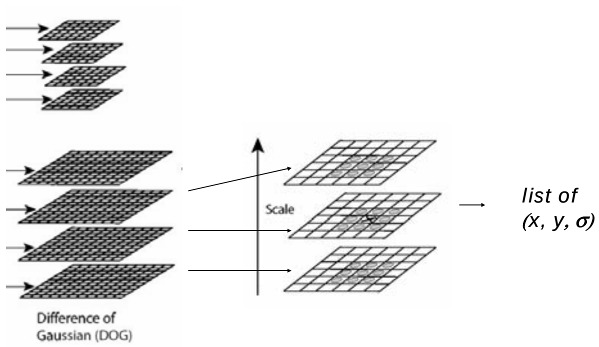




Lowe’s SIFT (DoG) Detector

- Difference-of-Gaussians as approximation of the Laplacian-of-Gaussian

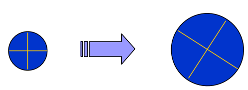




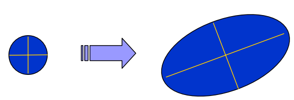
Lowe's SIFT (DoG) Detector

Affine Invariant Detection

- Previously we considered:
  Similarity transform (rotation + uniform scale)

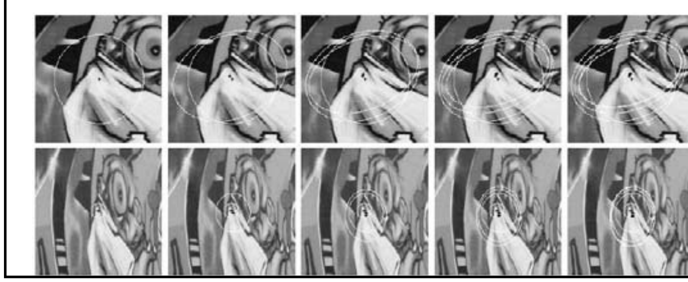


- Now we go on to:
  Affine transform (rotation + scale + skew)



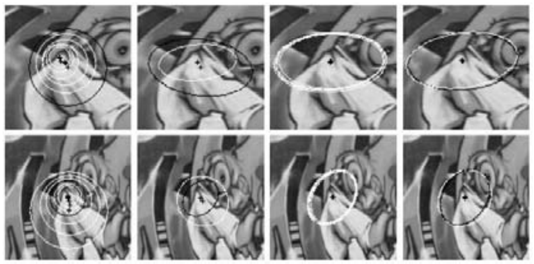
Affine Invariant Dection: Mikolajczyk's Harris-Affine Detector

- Initialization with Harris Laplace
- Estimate shape based on second moment matrix
- Use normalization / deskewing
- Iterative algorithm



Affine Invariant Detection: Mikolajczyk's Harris-Affine Detector

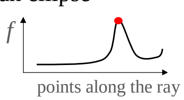
1. Detect multi-scale Harris points
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location



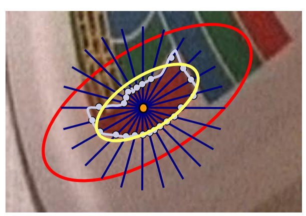
Harris-Affine

Affine Invariant Detection: Tuytelaars's Intensity-based Regions

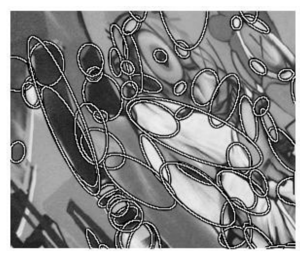
1. Select intensity extrema
2. Consider intensity profile along rays from each extremum point
3. Select maximum of invariant function $f(t)$ along each ray
4. Connect all local maxima
5. Compute *geometric moments* of orders up to 2 for this region
6. Fit an ellipse

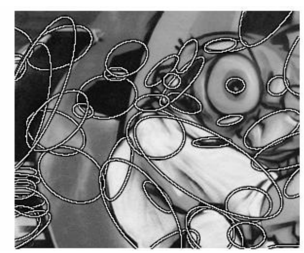


$$f(t) = \frac{abs(I_0 - I)}{\max\left(\frac{\int abs(I_0 - I)dt}{t}, d\right)}$$



Intensity-based Regions





Quantitative Comparisons of Feature Detectors

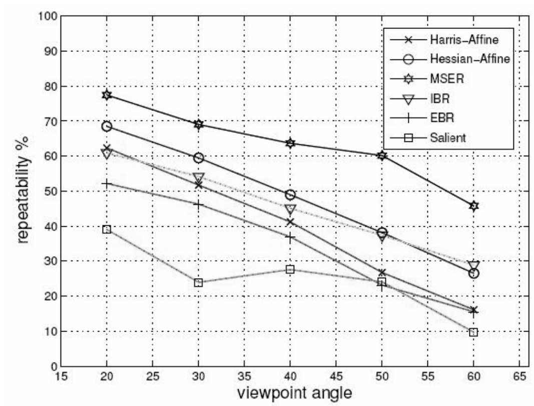
- Evaluation of interest point detectors, C. Schmid, R. Mohr and C. Bauckhage, *Int. J. Computer Vision* **37**(2), 2000
- Scale and affine invariant interest point detectors, K. Mikolajczyk and C. Schmid, *Int. J. Computer Vision* **60**(1), 2004
- A comparison of affine region detectors, K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, *Int. J. Computer Vision* **65**(1/2), 2005
- A survey on local invariant features, T. Tuytelaars and K. Mikolajczyk
- Evaluation on 3D objects (Moreels & Perona, ICCV, 2005)
- Evaluation on 3D objects (Fraundorfer & Bischof, ICCV, 2005)

Evaluation Criterion: Repeatability

- Repeatability rate: percentage of correctly corresponding points

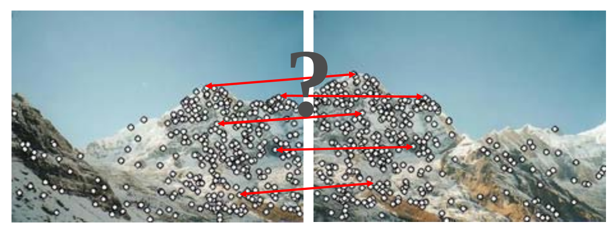
$$\text{repeatability} = \frac{\#\text{correspondences}}{\#\text{detected}} \cdot 100\%$$

Repeatability



Feature Point Descriptors

- We know how to detect points
- Next question: **How to match them?**

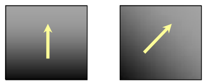


Point descriptor should be:
1. Invariant
2. Distinctive

Descriptors Invariant to Rotation

- Find local orientation

Dominant direction of gradient:

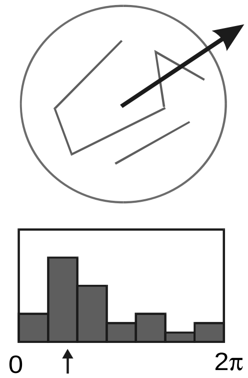


- Compute description relative to this orientation

[1] K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001
[2] D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004

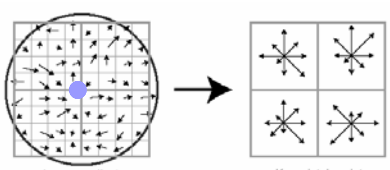
SIFT: Select Canonical Orientation

- Compute histogram of local gradient directions computed at selected scale of Gaussian pyramid in neighborhood of a keypoint
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)



SIFT Keypoint Feature Descriptor

- Descriptor overview:
  - Compute gradient orientation histograms on 4 x 4 neighborhoods, relative to the keypoint orientation using thresholded image gradients from Gaussian pyramid level at keypoint’s scale
  - Quantize orientations to 8 values
  - 2 x 2 array of histograms
  - SIFT feature vector of length 4 x 4 x 8 = 128 values for each keypoint
  - Normalize the descriptor to make it invariant to intensity change

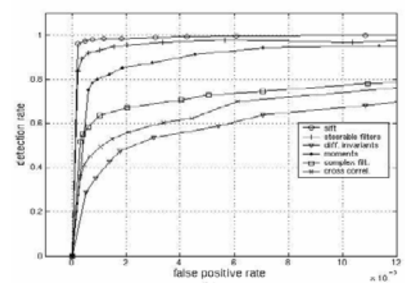


D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV 2004

SIFT – Scale Invariant Feature Transform[1]

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[1] D.Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” IJCV 2004
[2] K.Mikolajczyk, C.Schmid, “A Performance Evaluation of Local Descriptors,” CVPR 2003

References on Feature Descriptors

- A performance evaluation of local descriptors, K. Mikolajczyk and C. Schmid, *IEEE Trans. PAMI* **27**(10), 2005
- Evaluation of features detectors and descriptors based on 3D objects, P. Moreels and P. Perona, *Int. J. Computer Vision* **73**(3), 2007

Feature Detection and Description Summary

- Stable (repeatable) feature points can be detected regardless of image changes
  - Scale: search for correct scale as *maximum* of an appropriate function
  - Affine: approximate regions with *ellipses*
- Invariant and distinctive descriptors can be computed
  - Invariant *moments*
  - *Normalizing* with respect to scale and affine transformation
- Limited affine invariance for large viewpoint changes; no projective invariant methods
- Incorporate color, texture into descriptor