Visual Recognition: Instance Level Recognition

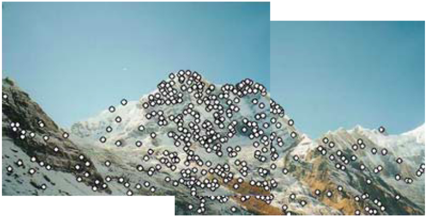
Raquel Urtasun

TTI Chicago

Jan 19, 2012

Local features for **instance-level** recognition

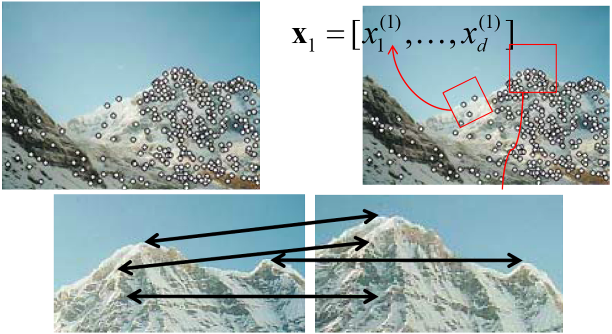
Application Example: Image stitching



[Source: K. Grauman]

Local features

- **Detection**: Identify the interest points.
- **Description**: Extract vector feature descriptor around each interest point.
- **Matching**: Determine correspondence between descriptors in two views.
- **Tracking**: alternative to matching that only searches a small neighborhood around each detected feature.



[Source: K. Grauman]

Goal: interest operator repeatability

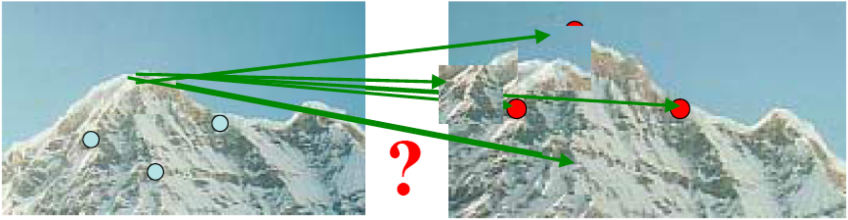
- We want to detect (at least some of) the same points in both images.
- We have to be able to run the detection procedure independently per image.

Figure: No chance to find the true matches

[Source: K. Grauman]

Goal: descriptor distinctiveness

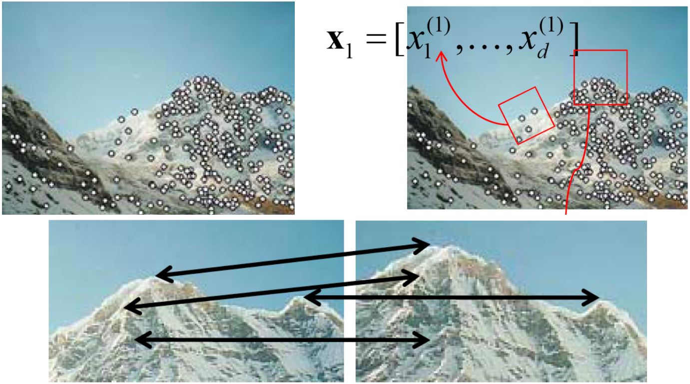
- We want to be able to reliably determine which point goes with which.
- Must provide some invariance to geometric and photometric differences between the two views.



[Source: K. Grauman]

Local features

- **Detection**: Identify the interest points.
- **Description**: Extract vector feature descriptor around each interest point.
- **Matching**: Determine correspondence between descriptors in two views.



[Source: K. Grauman]

What points to choose?

[Source: K. Grauman]

What points to choose?

- Textureless patches are nearly impossible to localize.
- Patches with large contrast changes (gradients) are easier to localize.

What points to choose?

- Textureless patches are nearly impossible to localize.
- Patches with large contrast changes (gradients) are easier to localize.
- But straight line segments at a single orientation suffer from the aperture problem, i.e., it is only possible to align the patches along the direction normal to the edge direction.

What points to choose?

- Textureless patches are nearly impossible to localize.
- Patches with large contrast changes (gradients) are easier to localize.
- But straight line segments at a single orientation suffer from the aperture problem, i.e., it is only possible to align the patches along the direction normal to the edge direction.
- Gradients in at least two (significantly) different orientations are the easiest, e.g., corners.

What points to choose?

- Textureless patches are nearly impossible to localize.
- Patches with large contrast changes (gradients) are easier to localize.
- But straight line segments at a single orientation suffer from the aperture problem, i.e., it is only possible to align the patches along the direction normal to the edge direction.
- Gradients in at least two (significantly) different orientations are the easiest, e.g., corners.

Corners as distinctive interest points

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

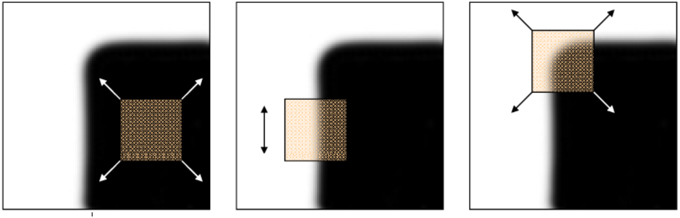


Figure: (left) flat region: no change in all directions, (center) edge: no change along the edge direction, (right) corner: significant change in all directions

[Source: Alyosha Efros, Darya Frolova, Denis Simakov]

A Simple Matching Criteria

- Compare two image patches using (weighted) summed square difference

$$E_{WSSD}(\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_1(\mathbf{p}_i + \mathbf{u}) - I_0(\mathbf{p}_i)]^2$$

with I_0 and I_1 two images being compared, $\mathbf{u}(u_x, u_y)$ a displacement vector, $w(\mathbf{p})$ a spatially varying weighting function, and the summation i is over all the pixels in the patch.

- We do not know which other image locations the feature will end up being matched against.

A Simple Matching Criteria

- Compare two image patches using (weighted) summed square difference

$$E_{WSSD}(\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_1(\mathbf{p}_i + \mathbf{u}) - I_0(\mathbf{p}_i)]^2$$

  with I_0 and I_1 two images being compared, $\mathbf{u}(u_x, u_y)$ a displacement vector, $w(\mathbf{p})$ a spatially varying weighting function, and the summation i is over all the pixels in the patch.

- We do not know which other image locations the feature will end up being matched against.
- We can only compute how stable this metric is with respect to small variations in position u by comparing an image patch against itself.

A Simple Matching Criteria

- Compare two image patches using (weighted) summed square difference

$$E_{WSSD}(\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_1(\mathbf{p}_i + \mathbf{u}) - I_0(\mathbf{p}_i)]^2$$

  with I_0 and I_1 two images being compared, $\mathbf{u}(u_x, u_y)$ a displacement vector, $w(\mathbf{p})$ a spatially varying weighting function, and the summation i is over all the pixels in the patch.

- We do not know which other image locations the feature will end up being matched against.
- We can only compute how stable this metric is with respect to small variations in position u by comparing an image patch against itself.
- This is the **auto-correlation function**

$$E_{AC}(\Delta\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_0(\mathbf{p}_i + \Delta u) - I_0(\mathbf{p}_i)]^2$$

A Simple Matching Criteria

- Compare two image patches using (weighted) summed square difference

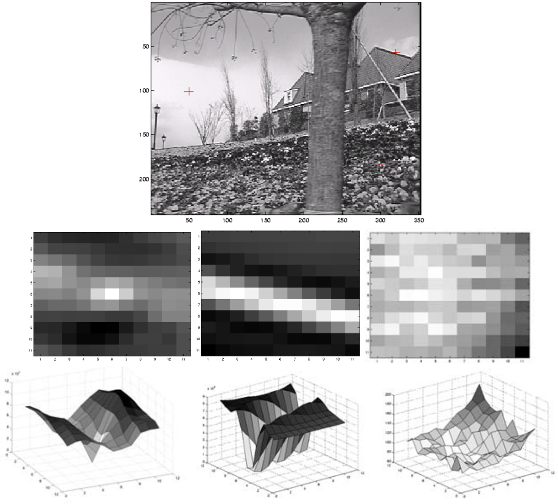
$$E_{WSSD}(\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_1(\mathbf{p}_i + \mathbf{u}) - I_0(\mathbf{p}_i)]^2$$

  with I_0 and I_1 two images being compared, $\mathbf{u}(u_x, u_y)$ a displacement vector, $w(\mathbf{p})$ a spatially varying weighting function, and the summation i is over all the pixels in the patch.

- We do not know which other image locations the feature will end up being matched against.
- We can only compute how stable this metric is with respect to small variations in position u by comparing an image patch against itself.
- This is the **auto-correlation function**

$$E_{AC}(\Delta\mathbf{u}) = \sum_i w(\mathbf{p}_i)[I_0(\mathbf{p}_i + \Delta u) - I_0(\mathbf{p}_i)]^2$$

Which one is better?



[Source: R. Szeliski]

How to select?

- Using a Taylor Series expansion $I_0(\mathbf{p}_i + \Delta\mathbf{u}) \approx I_0(\mathbf{p}_i) + \nabla I_0(\mathbf{p}_i)$ we can approximate the autocorrelation as

$$
\begin{aligned}
E_{AC}(\Delta\mathbf{u}) &= \sum_i w(\mathbf{p}_i)[I_0(\mathbf{p}_i + \Delta\mathbf{u}) - I_0(\mathbf{p}_i)]^2 \\
&\approx \sum_i w(\mathbf{p}_i)[I_0(\mathbf{p}_i) + \nabla I_0(\mathbf{p}_i)\Delta\mathbf{u} - I_0(\mathbf{p}_i)]^2 \\
&= \sum_i w(\mathbf{p}_i)[\nabla I_0(\mathbf{p}_i)\Delta\mathbf{u}]^2 \\
&= \Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u}
\end{aligned}
$$

  with

$$
\nabla I_0(\mathbf{p}_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)(\mathbf{p}_i)
$$

  the image gradient.

- Gradient can be computed with the filtering techniques we saw, e.g., derivatives of Gaussians.

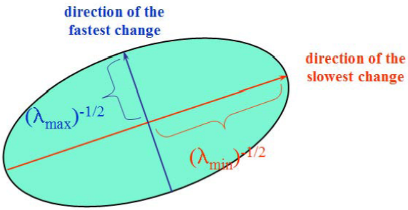
More on selection

- The autocorrelation is $E_{AC}(\Delta\mathbf{u}) = \Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u}$, with

$$\mathbf{A} = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

  where we have replaced the weighted summations with discrete convolutions with the weighting kernel w.

- **A** can be interpreted as a tensor where the outer products of the gradients are convolved with a weighting function.
- Eigenvalues a notion of uncertainty



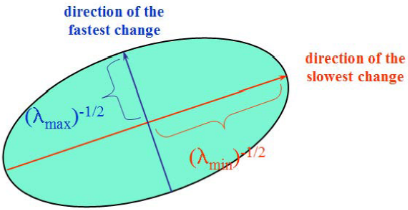
[Source: R. Szeliski]

More on selection

- The autocorrelation is $E_{AC}(\Delta \mathbf{u}) = \Delta \mathbf{u}^T \mathbf{A} \Delta \mathbf{u}$, with

$$\mathbf{A} = \sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

  where we have replaced the weighted summations with discrete convolutions with the weighting kernel w.
- **A** can be interpreted as a tensor where the outer products of the gradients are convolved with a weighting function.
- Eigenvalues a notion of uncertainty



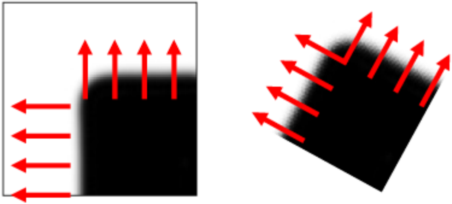
[Source: R. Szeliski]

Eigenvalues a notion of uncertainty

- $\mathbf{A}$ is symmetric

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \mathbf{U}^T \quad \text{with} \quad \mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

- The eigenvalues of $\mathbf{A}$ reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.
- How is this matrix for



[Source: R. Szeliski]

Local Feature Selection Criteria

- Shi and Tomasi, 94 proposed the smallest eigenvalue of **A**, i.e., $\lambda_0^{-1/2}$.
- Harris and Stephens, 88 is rotationally invariant and downweights edge-like features where $\lambda_1 \gg \lambda_0$

$$\det(\mathbf{A}) - \alpha \text{trace}(\mathbf{A})^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

Local Feature Selection Criteria

- Shi and Tomasi, 94 proposed the smallest eigenvalue of $\mathbf{A}$, i.e., $\lambda_0^{-1/2}$.
- Harris and Stephens, 88 is rotationally invariant and downweights edge-like features where $\lambda_1 \gg \lambda_0$

$$\text{det}(\mathbf{A}) - \alpha \text{trace}(\mathbf{A})^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

- Triggs, 04 suggested

$$\lambda_0 - \alpha \lambda_1$$

also reduces the response at 1D edges, where aliasing errors sometimes inflate the smaller eigenvalue.

Local Feature Selection Criteria

- Shi and Tomasi, 94 proposed the smallest eigenvalue of $\mathbf{A}$, i.e., $\lambda_0^{-1/2}$.
- Harris and Stephens, 88 is rotationally invariant and downweights edge-like features where $\lambda_1 \gg \lambda_0$

$$\det(\mathbf{A}) - \alpha \text{trace}(\mathbf{A})^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

- Triggs, 04 suggested

$$\lambda_0 - \alpha\lambda_1$$

  also reduces the response at 1D edges, where aliasing errors sometimes inflate the smaller eigenvalue.
- Brown et al, 05 use the harmonic mean

$$\frac{\det(\mathbf{A})}{\text{trace}(\mathbf{A})} = \frac{\lambda_0\lambda_1}{\lambda_0 + \lambda_1}$$

Local Feature Selection Criteria

- Shi and Tomasi, 94 proposed the smallest eigenvalue of $\mathbf{A}$, i.e., $\lambda_0^{-1/2}$.
- Harris and Stephens, 88 is rotationally invariant and downweights edge-like features where $\lambda_1 \gg \lambda_0$

$$\det(\mathbf{A}) - \alpha \text{trace}(\mathbf{A})^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$$

- Triggs, 04 suggested

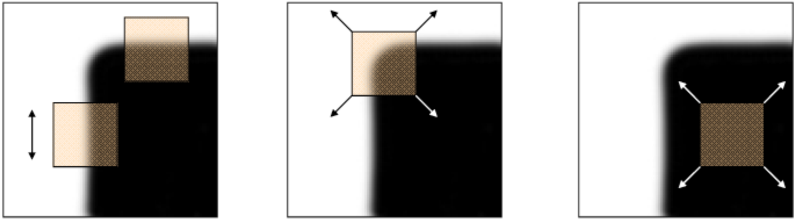
$$\lambda_0 - \alpha \lambda_1$$

  also reduces the response at 1D edges, where aliasing errors sometimes inflate the smaller eigenvalue.

- Brown et al, 05 use the harmonic mean

$$\frac{\det(\mathbf{A})}{\text{trace}(\mathbf{A})} = \frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$$

Type of responses



"edge":

$\lambda_1 >> \lambda_2$

$\lambda_2 >> \lambda_1$

"corner":

λ_1 and λ_2 are large, $\lambda_1 \sim \lambda_2$;

"flat" region

λ_1 and λ_2 are small;

[Source: K. Grauman]

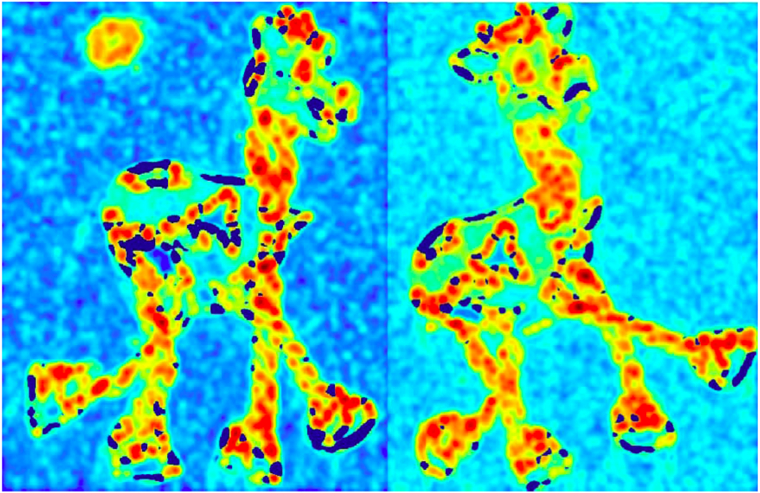
Harris Corner detector

1. Compute **A** for each image window to get their cornerness scores.
2. Find points whose surrounding window gave large corner response ($f >$ threshold).
3. Take the points of local maxima, i.e., perform non-maximum suppression.

Example

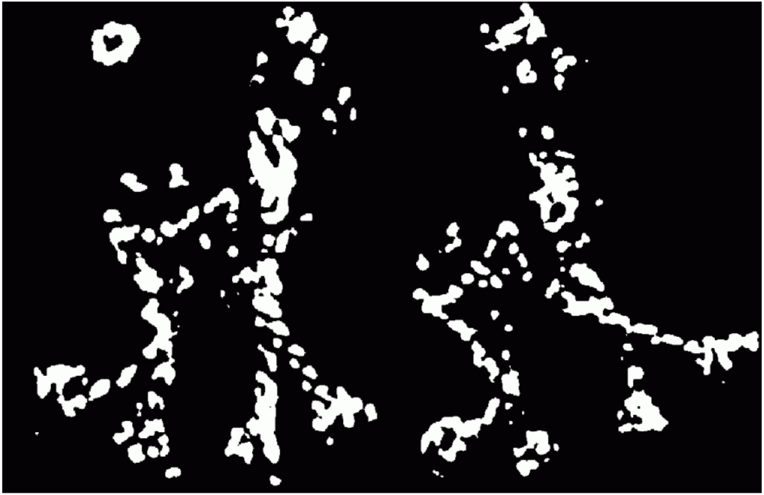
[Source: K. Grauman]

1) Compute Cornerness



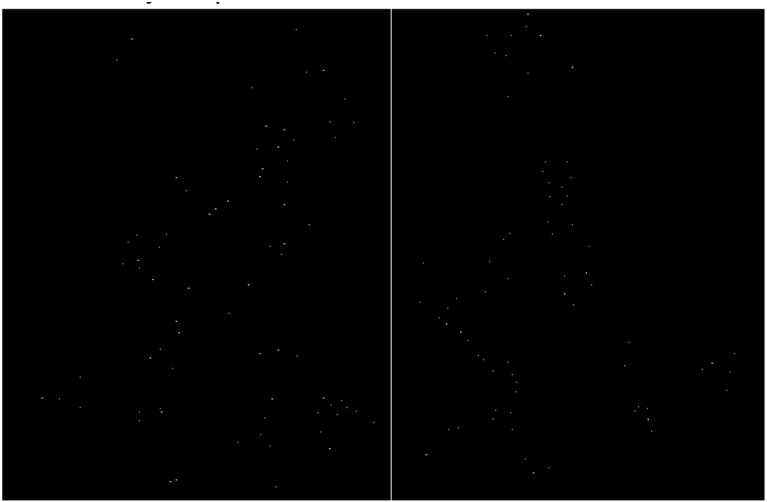
[Source: K. Grauman]

2) Find High Response



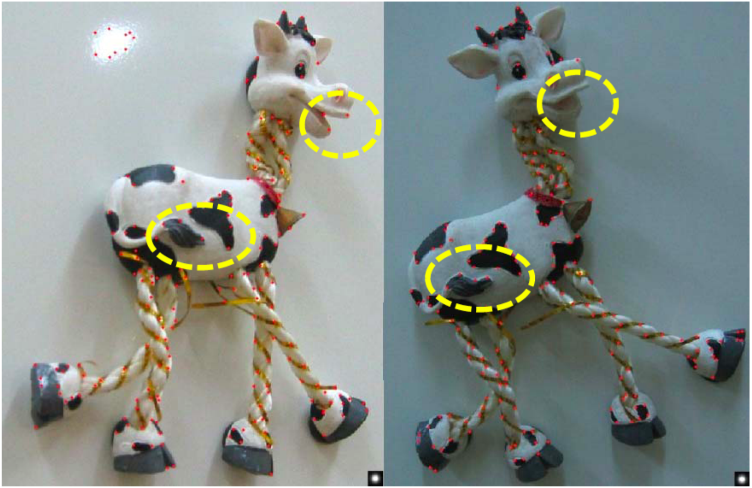
[Source: K. Grauman]

3) Non-maxima Suppresion



[Source: K. Grauman]

Results

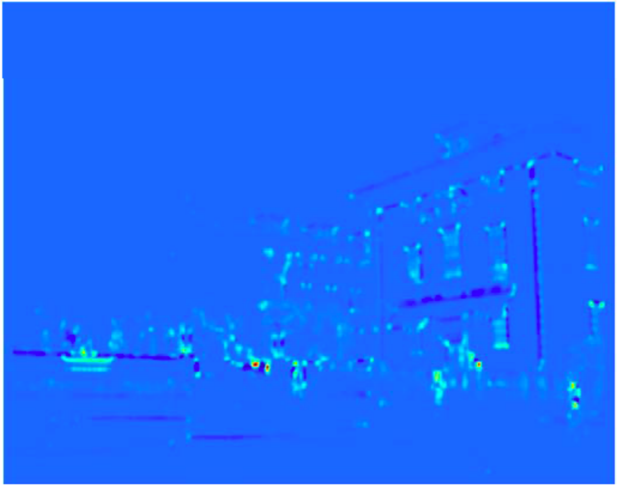


[Source: K. Grauman]

Another Example

[Source: K. Grauman]

Cornerness



[Source: K. Grauman]

Interest Points

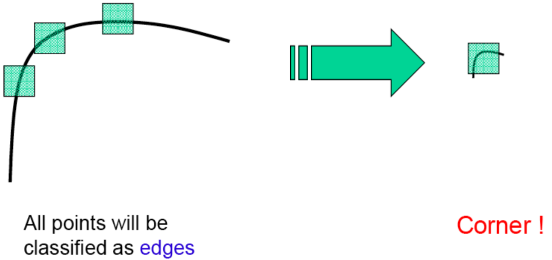
[Source: K. Grauman]

Properties of Harris Corner Detector

- Rotation invariant?

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} = \mathbf{U} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \mathbf{U}^T \quad \text{with} \quad \mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

- Scale Invariant?



[Source: K. Grauman]

Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

- Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.

Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

- Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
- When does this work?

Scale invariant interest points

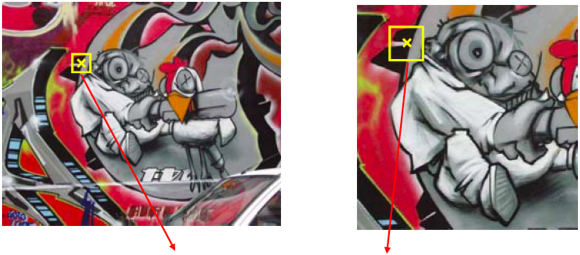
How can we independently select interest points in each image, such that the detections are repeatable across different scales?

- Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
- When does this work?
- More efficient to extract features that are stable in both location and scale.

Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

- Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
- When does this work?
- More efficient to extract features that are stable in both location and scale.
- Find scale that gives local maxima of a function f in both position and scale.

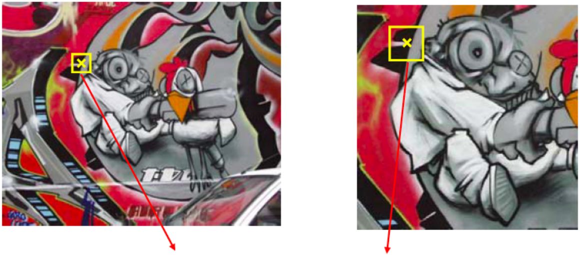


$$f(I_{i_1 \dots i_m}(x,\sigma)) = f(I_{i_1 \dots i_m}(x',\sigma'))$$

Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?

- Extract features at a variety of scales, e.g., by using multiple resolutions in a pyramid, and then matching features at the same level.
- When does this work?
- More efficient to extract features that are stable in both location and scale.
- Find scale that gives local maxima of a function f in both position and scale.



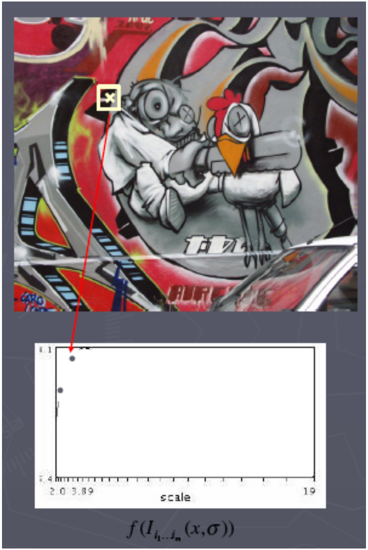
$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

Automatic Scale Selection

Function responses for increasing scale (scale signature).

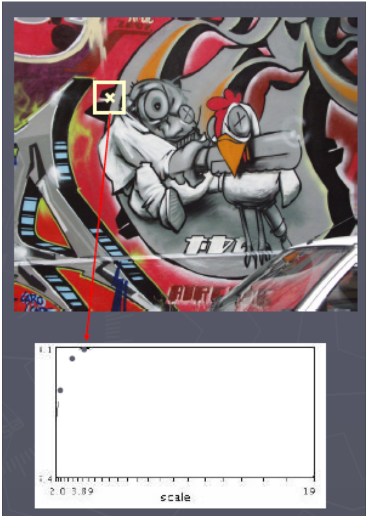
Automatic Scale Selection

Function responses for increasing scale (scale signature).



Automatic Scale Selection

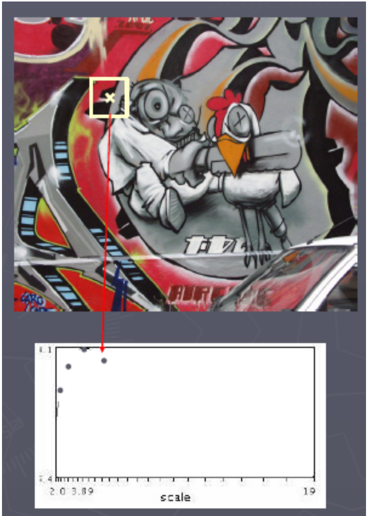
Function responses for increasing scale (scale signature).



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

Automatic Scale Selection

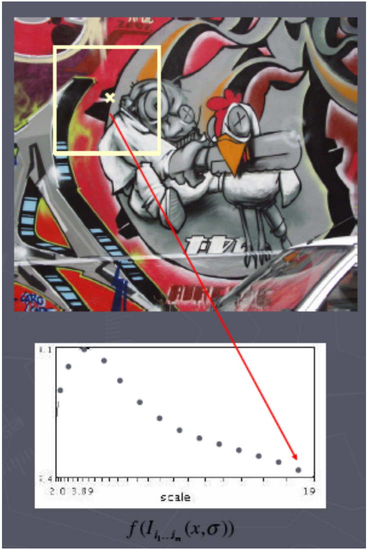
Function responses for increasing scale (scale signature).



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

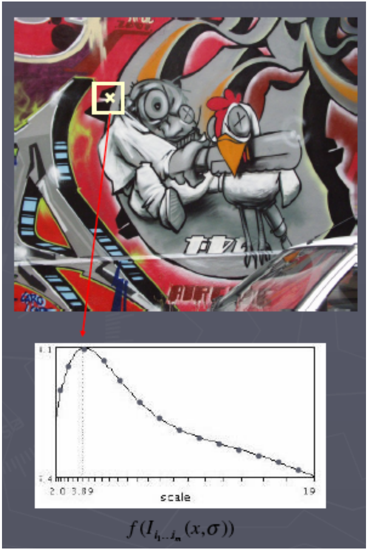
Automatic Scale Selection

Function responses for increasing scale (scale signature).



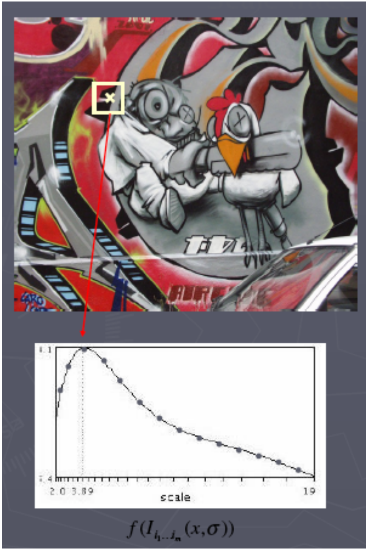
Automatic Scale Selection

Function responses for increasing scale (scale signature).



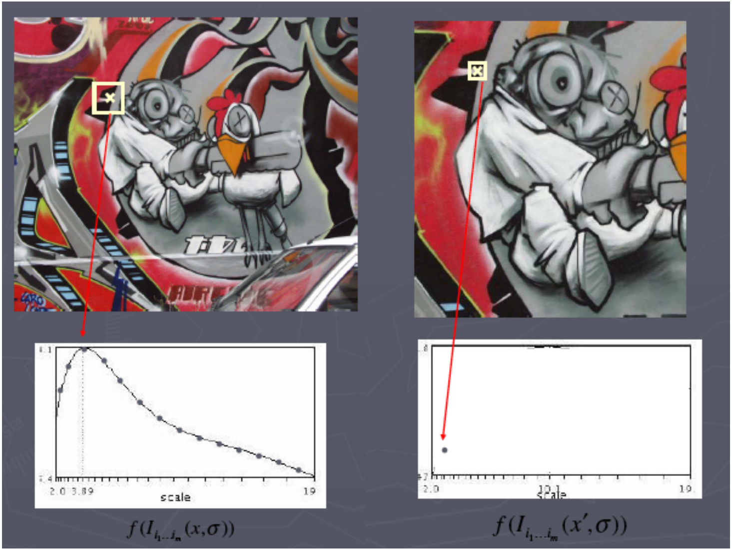
Automatic Scale Selection

Function responses for increasing scale (scale signature).



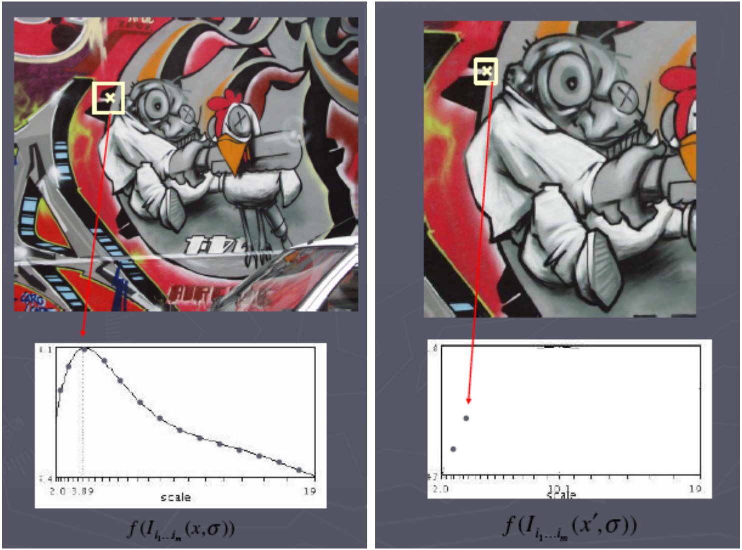
Automatic Scale Selection

Function responses for increasing scale (scale signature).

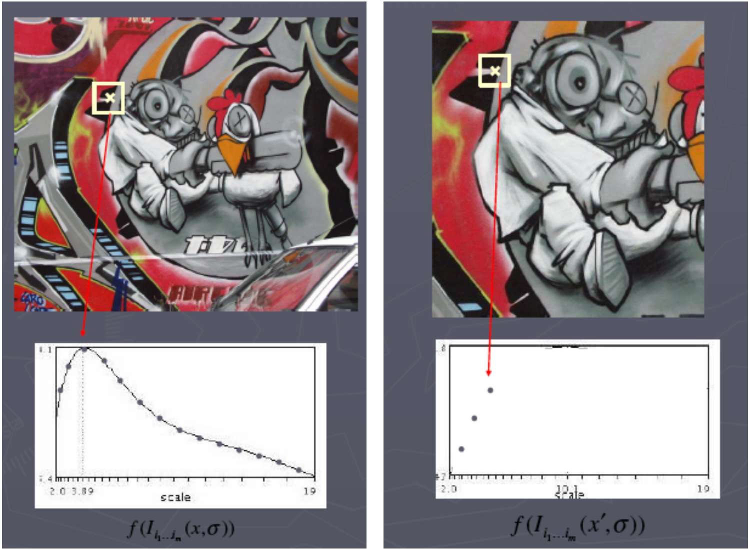


Automatic Scale Selection

Function responses for increasing scale (scale signature).

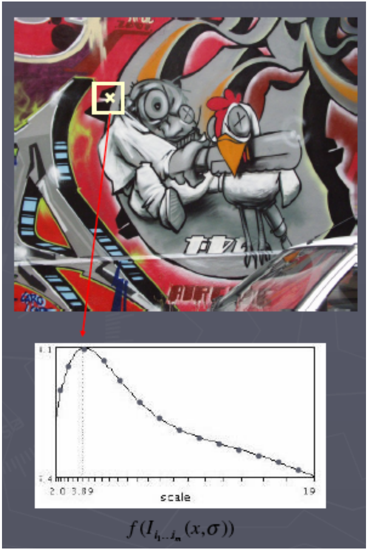


Function responses for increasing scale (scale signature).

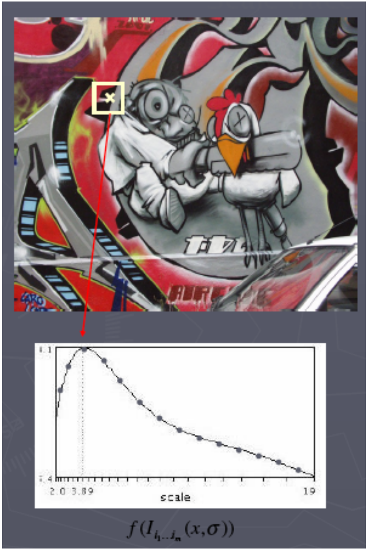


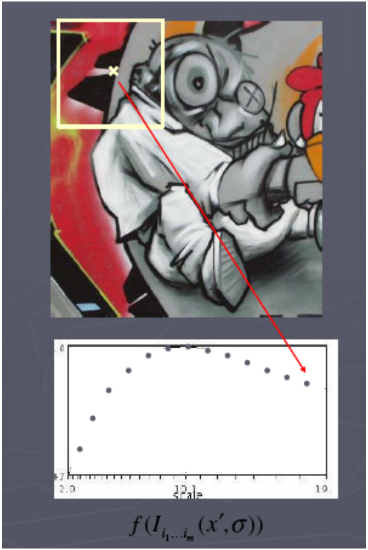
Automatic Scale Selection

Function responses for increasing scale (scale signature).



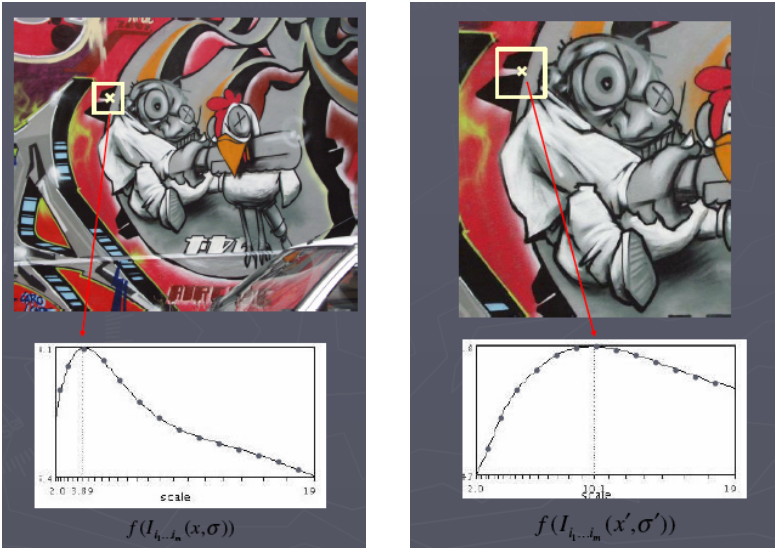
Function responses for increasing scale (scale signature).





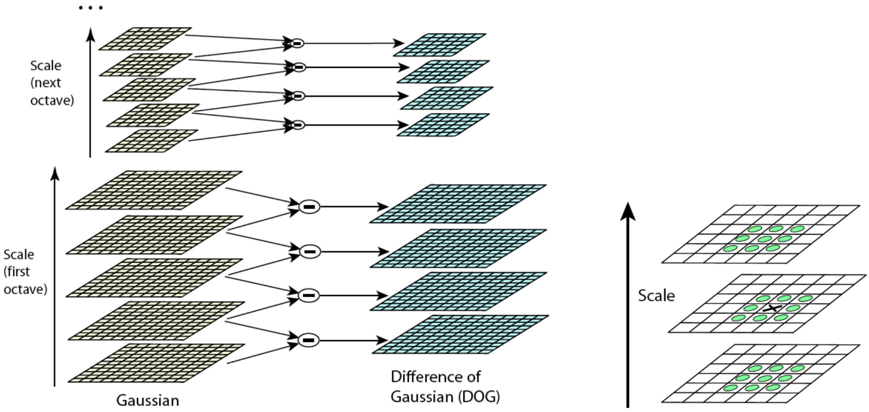
Automatic Scale Selection

Function responses for increasing scale (scale signature).



What can the signature function be?

- Lindeberg (1998): extrema in the Laplacian of Gaussian (LoG).
- Lowe (2004) proposed computing a set of sub-octave Difference of Gaussian filters looking for 3D (space+scale) maxima in the resulting structure.

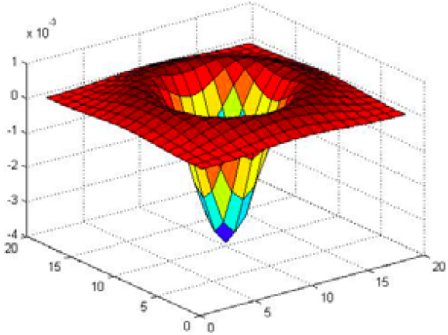


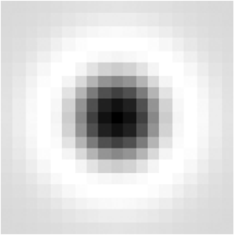
[Source: R. Szeliski]

Blob detection

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

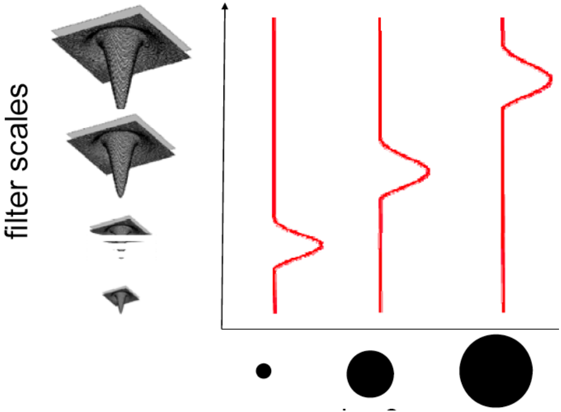




[Source: K. Grauman]

Blob detection in 2D: scale selection

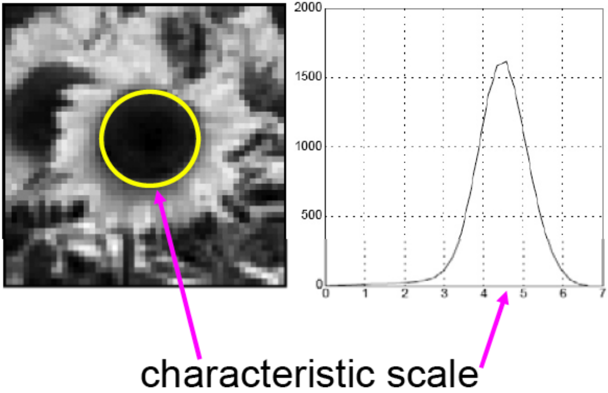
Laplacian-of-Gaussian = blob detector



[Source: B. Leibe]

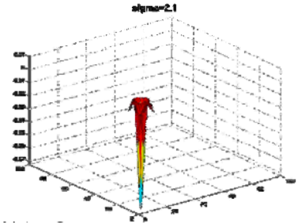
Characteristic Scale

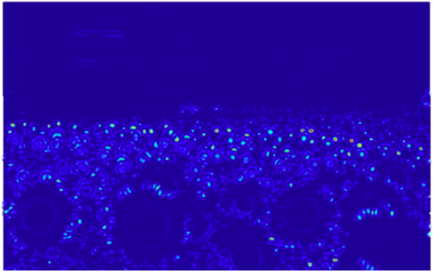
- We define the **characteristic scale** as the scale that produces peak of Laplacian response



[Source: S. Lazebnik]

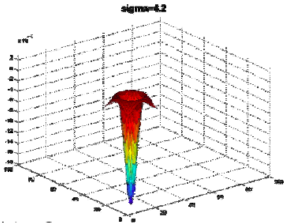
Example

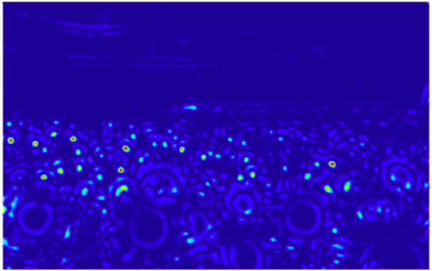




[Source: K. Grauman]

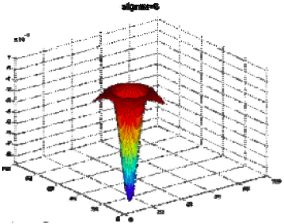
Example

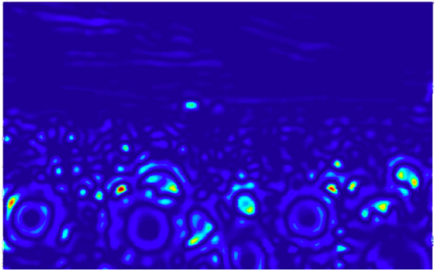




[Source: K. Grauman]

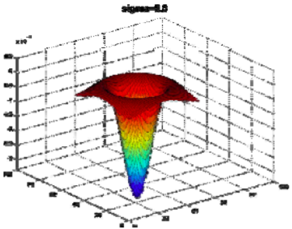
Example

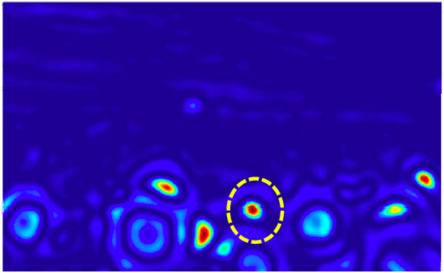




[Source: K. Grauman]

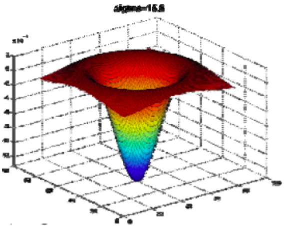
Example

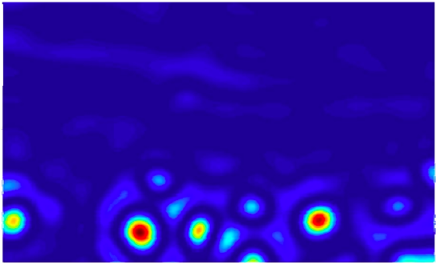




[Source: K. Grauman]

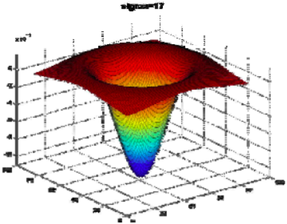
Example

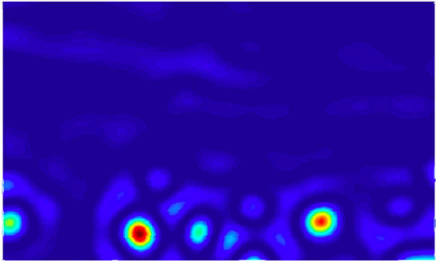




[Source: K. Grauman]

Example

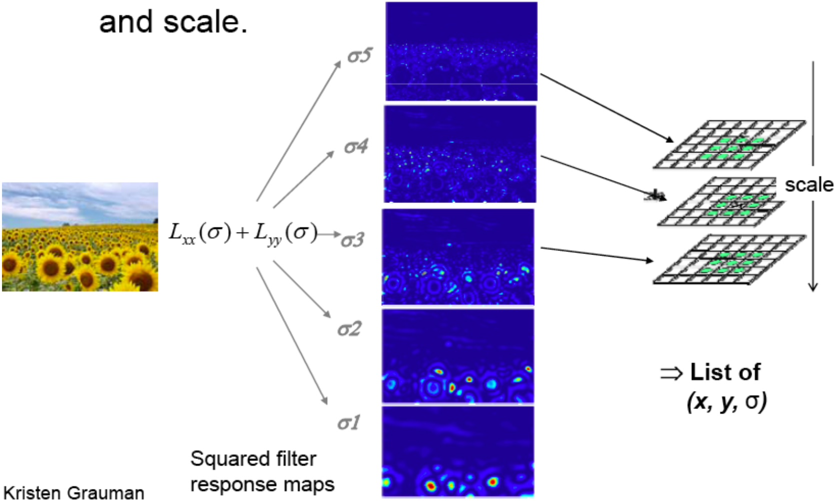




[Source: K. Grauman]

Scale invariant interest points

Interest points are local maxima in both position and scale.



Kristen Grauman

Example

[Source: S. Lazebnik]

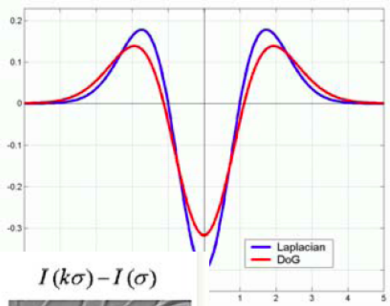
Fast approximation

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

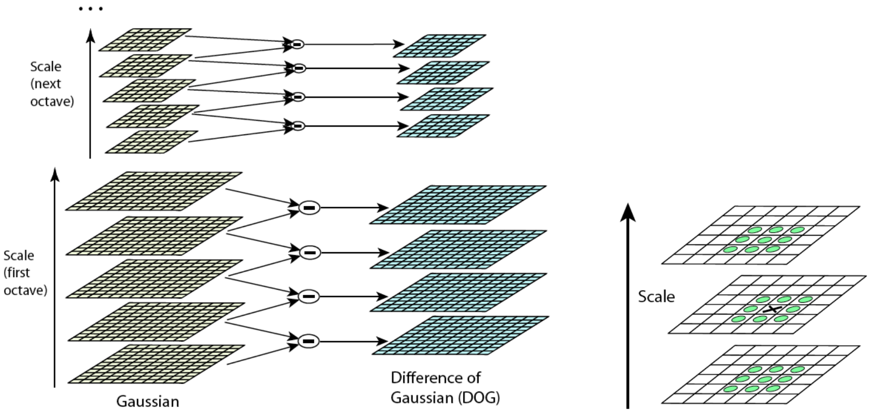


$I(k\sigma)$ $\quad$ $I(\sigma)$ $\quad$ $I(k\sigma) - I(\sigma)$

[Source: K. Grauman]

Lowe's DoG

- Lowe (2004) proposed computing a set of sub-octave Difference of Gaussian filters looking for 3D (space+scale) maxima in the resulting structure



[Source: R. Szeliski]

Laplacian vs Hessian

- Laplacian of Gaussians is scale invariant.
- Simple and efficient.
- But fires more on edges than determinant of hessian

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **Robust**: noise, blur, discretization, compression, etc. do not have a big impact on the feature.

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **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.

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **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.

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **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.

Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **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.

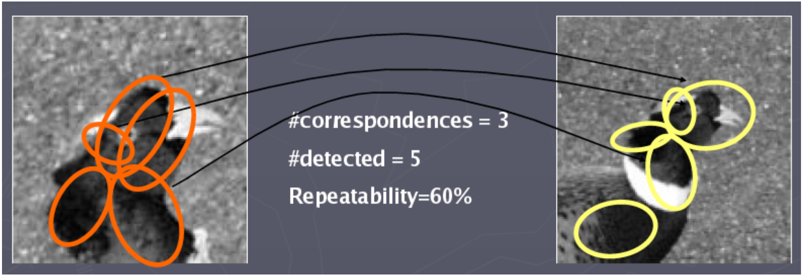
Properties of the ideal feature

- **Local**: features are local, so robust to occlusion and clutter (no prior segmentation).
- **Invariant**: to certain transformations, e.g, scale, rotation.
- **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.

A lot of other interest point detectors

- Hessian
- Lowe: DoG
- Lindeberg: scale selection
- Miikolajczyk & Schmid: Hessian/Harris-Laplacian/Affine
- Tuyttelaars & Van Gool: EBR and IBR
- Matas: MSER
- Kadir & Brrady: Salient Regions
- Speeded–Up Robust Features (SURF) of Bay et al.

- Repeatability rate: percentage of detected that have correct corresponding points
- What's the problem of this?



[Source: T. Tuyttellaars]

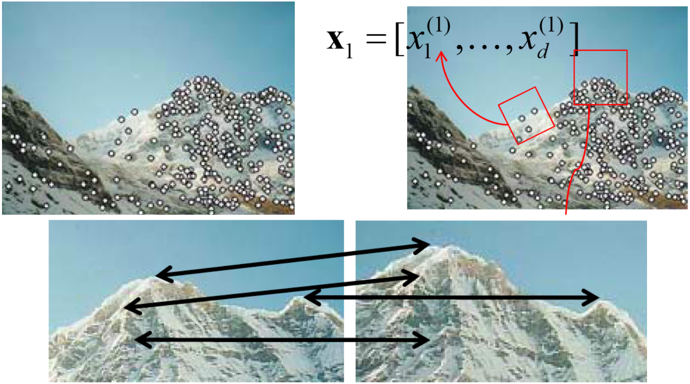
Evaluation criteria: repeatability

- Two points are in correspondence if the intersection over union is bigger than a certain threshold.

[Source: T. Tuyttellaars]

Local features

- **Detection**: Identify the interest points.
- **Description**: Extract vector feature descriptor around each interest point.
- **Matching**: Determine correspondence between descriptors in two views.



[Source: K. Grauman]

The ideal feature descriptor

- Repeatable (invariant/robust)
- Distinctive
- Compact
- Efficient

Invariances

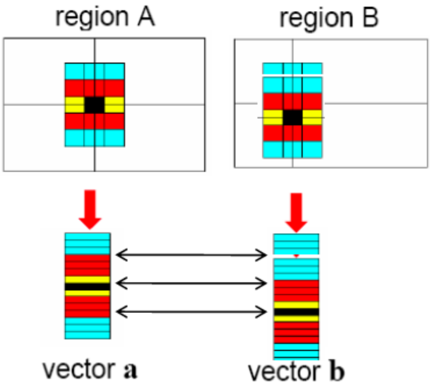
[Source: T. Tuytelaars]

Invariances

[Source: T. Tuytelaars]

Raw Pixels as Local Descriptrs

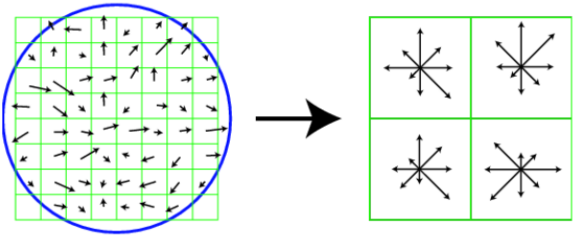
- The simplest way is to write down the list of intensities to form a feature vector, and normalize them (i.e., mean 0, variance 1).
- But this is very sensitive to even small shifts, rotations.



[Source: K. Grauman]

SIFT descriptor [Lowe 2004]

- Compute the gradient at each pixel in a 16×16 window around the detected keypoint, using the appropriate level of the Gaussian pyramid at which the keypoint was detected.
- Doweight gradients by a Gaussian fall-off function (blue circle) to reduce the influence of gradients far from the center.
- In each 4×4 quadrant, compute a gradient orientation histogram using 8 orientation histogram bins.



(a) image gradients

(b) keypoint descriptor

[Source: R. Szeliski]

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.
- To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.
- To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.
- Great engineering effort!

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.
- To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.
- Great engineering effort!
- Why subpatches?

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.
- To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.
- Great engineering effort!
- Why subpatches?
- Why does SIFT have some illumination invariance?

SIFT descriptor [Lowe 2004]

- To reduce the effects of location and dominant orientation misestimation, each of the original 256 weighted gradient magnitudes is softly added to nearby bins.
- The resulting 128 non-negative values form a raw version of the SIFT descriptor vector.
- To reduce the effects of contrast or gain (additive variations are already removed by the gradient), the 128-D vector is normalized to unit length.
- To further make the descriptor robust to other photometric variations, values are clipped to 0.2 and the resulting vector is once again renormalized to unit length.
- Great engineering effort!
- Why subpatches?
- Why does SIFT have some illumination invariance?

Making descriptor rotation invariant

- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation

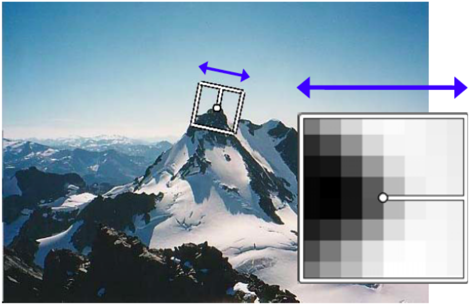


Figure: Figure from M. Brown

[Source: K. Grauman]

SIFT descriptor [Lowe 2004]

Extraordinarily robust matching technique

- Changes in viewpoint: up to about 60 degree out of plane rotation
- Changes in illumination: sometimes even day vs. night
- Fast and efficient: can run in real time
- Lots of code available

[Source: S. Seitz]

Example

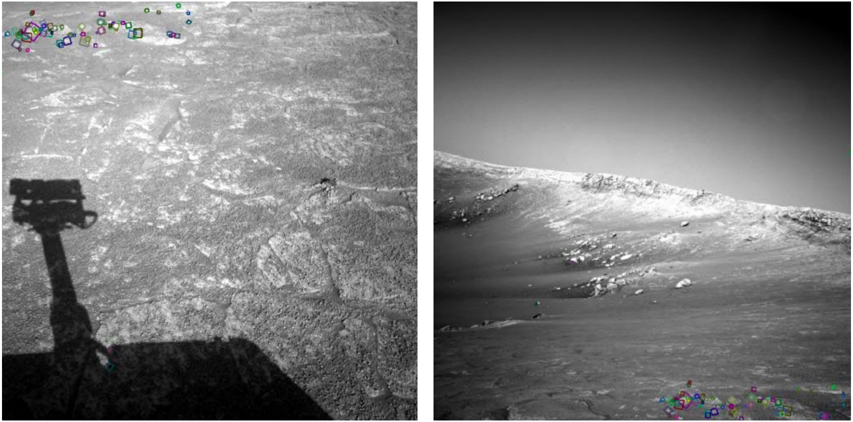


Figure: NASA Mars Rover images with SIFT feature matches

[Source: N. Snavely]

SIFT properties

Invant to

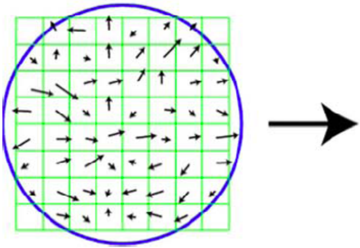
- Scale
- Rotation

Partially invariant to

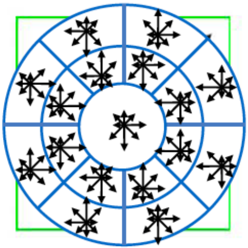
- Illumination changes
- Camera viewpoint
- Occlusion, clutter

Gradient location-orientation histogram (GLOH)

- Developed by Mikolajczyk and Schmid (2005): variant on SIFT that uses a log-polar binning structure instead of the four quadrants.
- The spatial bins are 11, and 15, with eight angular bins (except for the central region), for a total of 17 spatial bins and 16 orientation bins.
- The 272D histogram is then projected onto a 128D descriptor using PCA trained on a large database.



(a) image gradients



(b) keypoint descriptor

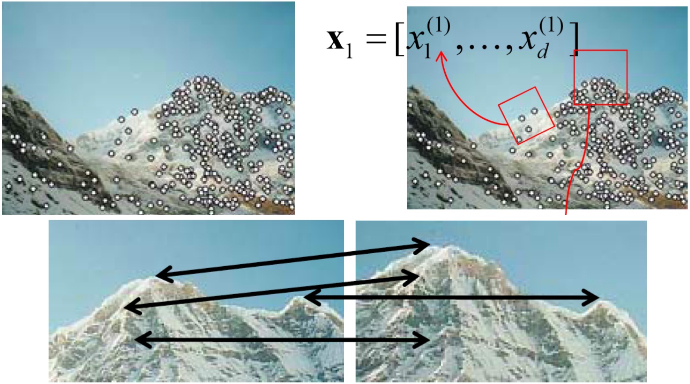
[Source: R. Szeliski]

Other Descriptors

- Steerable filters
- moment invariants
- complex filters
- shape contexts
- PCA-SIFT
- HOG
- SURF
- DAISY

Local features

- **Detection**: Identify the interest points.
- **Description**: Extract vector feature descriptor around each interest point.
- **Matching**: Determine correspondence between descriptors in two views.



[Source: K. Grauman]

Matching local features

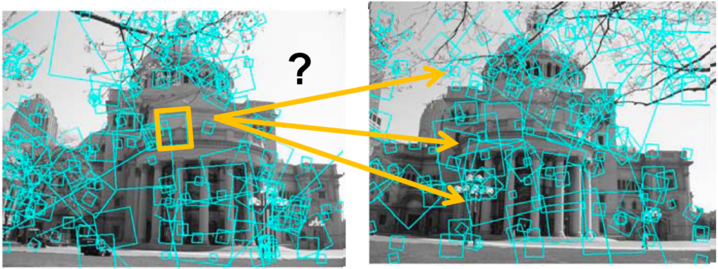
Once we have extracted features and their descriptors, we need to match the features between these images.

- Matching strategy: which correspondences are passed on to the next stage
- Devise efficient data structures and algorithms to perform this matching

Figure: Images from K. Grauman

Matching local features

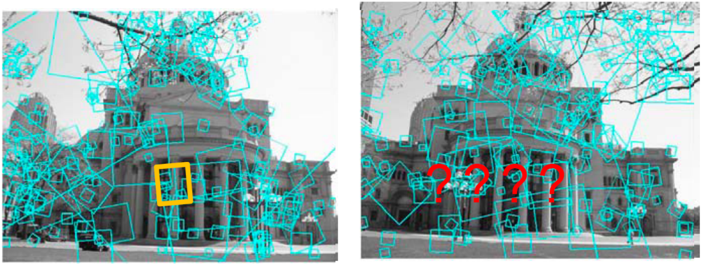
- To generate candidate matches, find patches that have the most similar appearance (e.g., lowest SSD)
- Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)



[Source: K. Grauman]

Ambiguous matches

- At what SSD value do we have a good match?
- To add robustness, consider ratio of distance to best match to distance to second best match
  - If low, first match looks good.
  - If high, could be ambiguous match.



[Source: K. Grauman]

Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor

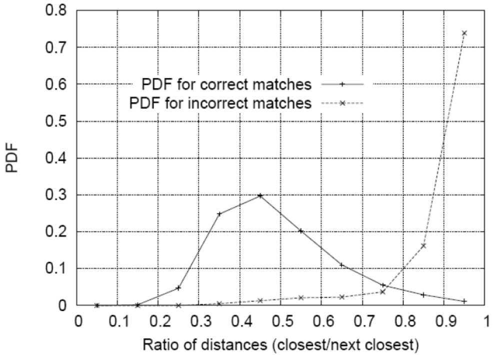


Figure: Images from D. Lowe

[Source: K. Grauman]

Which threshold to use?

- Setting the threshold too high results in too many false positives, i.e., incorrect matches being returned.
- Setting the threshold too low results in too many false negatives, i.e., too many correct matches being missed

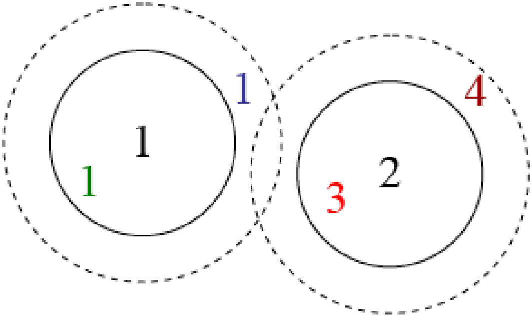


Figure: Images from R. Szeliski

How to quantize how good is our matching?

- TP: true positives, i.e., number of correct matches
- FN: false negatives, matches that were not correctly detected
- FP: false positives, proposed matches that are incorrect
- TN: true negatives, non-matches that were correctly rejected.

$$\text{True positive rate (recall)} \quad TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$

$$\text{True negative rate} \quad TNR = \frac{FP}{FP + TN} = \frac{FP}{N}$$

$$\text{positive predictive value (precision)} \quad PPV = \frac{TP}{TP + FP} = \frac{TP}{P'}$$

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

Measuring performance

- Any particular matching strategy (at a particular threshold or parameter setting) can be rated by the TPR and FPR numbers
- We want TPR=1 and FPR=0.
- As we vary the matching threshold, we obtain a family of such points, i.e., receiver operating characteristic (ROC curve)
- The closer this curve lies to the upper left corner, the better its performance.

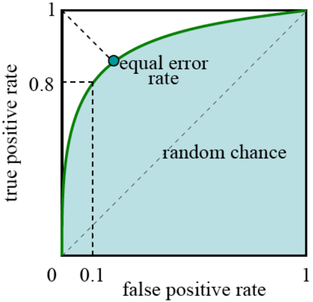


Figure: Images from R. Szeliski

Measuring performance

- Area under the curve (AUC) is a way to summarize ROC with 1 number.
- Mean average precision, which is the average precision (PPV) as you vary the threshold.
- The equal error rate is sometimes used as well.

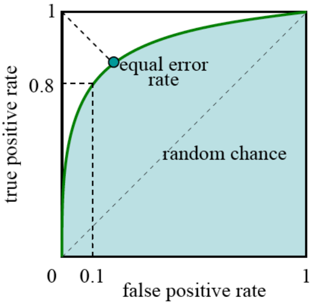


Figure: Images from R. Szeliski

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition

[Source: K. Grauman]

Wide Baseline Stereo

[Source: T. Tuytelaars]

Recognizing the Same Object

Schmid and Mohr 1997

Sivic and Zisserman, 2003

Rothganger et al. 2003

Lowe 2002

[Source: K. Grauman]

Motion Tracking

Figure: Images from J. Pilet

Summary

Interest point detection

- Harris corner detector
- Laplacian of Gaussian, automatic scale selection
- Difference of Gaussians

Invariant descriptors

- Rotation according to dominant gradient direction
- Histograms for robustness to small shifts and translations (SIFT descriptor)
- Polar coordinate descriptors GLOH.

Category-level recognition

Recognizing or retrieving specific objects

- Example: Visual search in feature films

Visually defined query

"Find this clock"

"Find this place"

"Groundhog Day" [Rammis, 1993]

[Source: J. Sivic]

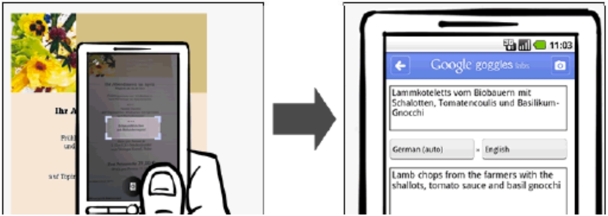
Recognizing or retrieving specific objects

- Example: Search photos on the web for particular places

Find these landmarks ...in these images and 1M more

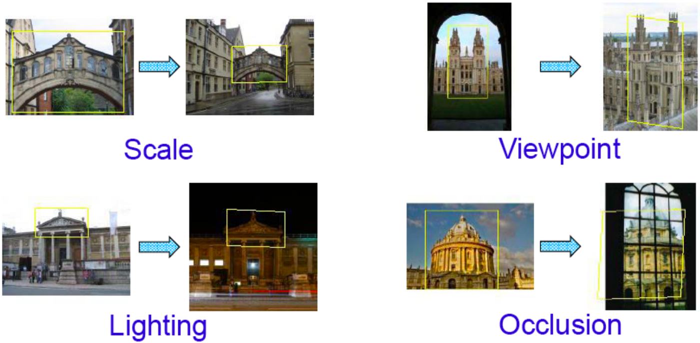
[Source: J. Sivic]


Google Goggles
Use pictures to search the web.
Watch a video
Get Google Goggles
Android (1.6+ required)
Download from Android Market.
Send Goggles to Android phone
New! iPhone (iOS 4.0 required)
Download from the App Store.
Send Goggles to iPhone
New!
Text
Landmarks
Books
Contact Info
Artwork
Wine
Logos


11:03
Google goggles
Lammkoteletts vom Biobauern mit Schalotten, Tomatencoulis und Basilikum-Gnocchi
German (auto)
English
Lamb chops from the farmers with the shallots, tomato sauce and basil gnocchi

Why is it difficult?

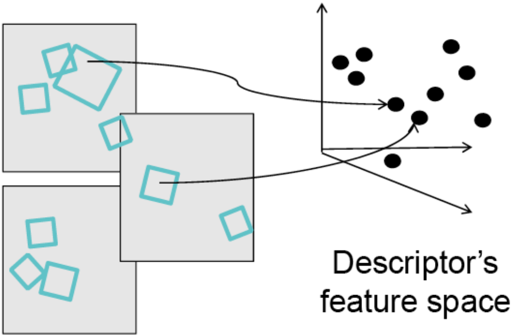
- Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion.
- We can't expect to match such varied instances with a single global template...



[Source: J. Sivic]

Indexing local features

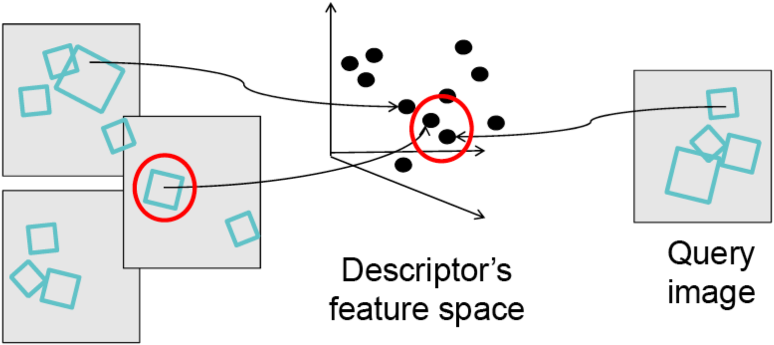
- Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



[Source: K. Grauman]

Indexing local features

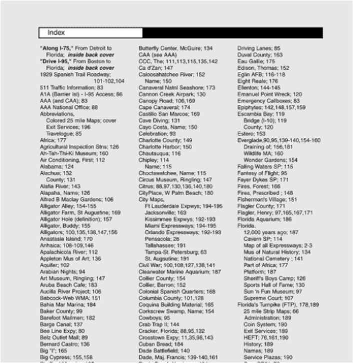
- It can have millions of features to search.



[Source: K. Grauman]

Indexing local features: inverted file index

- For text documents, an efficient way to find all pages on which a word occurs is to use an index.
- We want to find all images in which a feature occurs.
- To use this idea, well need to map our features to visual words.
- Why?

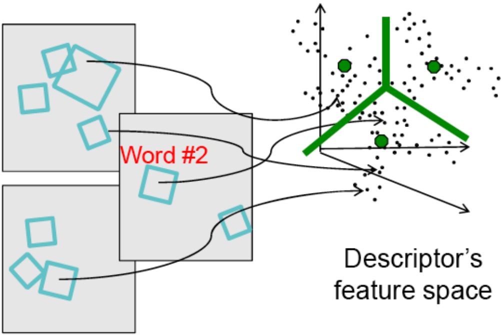


Index

[Source: K. Grauman]

Indexing local features: inverted file index

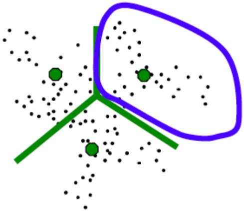
- Map high-dimensional descriptors to tokens/words by quantizing the feature space.
- Quantize via clustering, let cluster centers be the prototype words.
- Determine which word to assign to each new image region by finding the closest cluster.

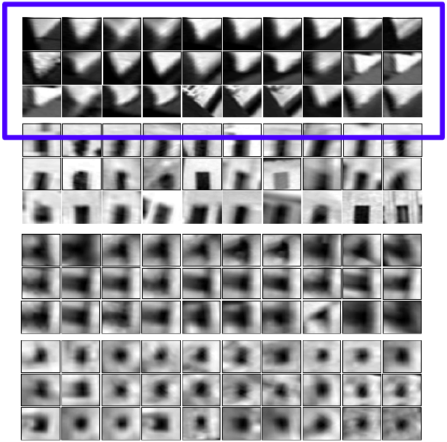


[Source: K. Grauman]

Visual words

- Each group of patches belongs to the same visual word.





Visual vocabulary formation issues

- Vocabulary size, number of words
- Sampling strategy: where to extract features?
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)