Local Feature Detectors

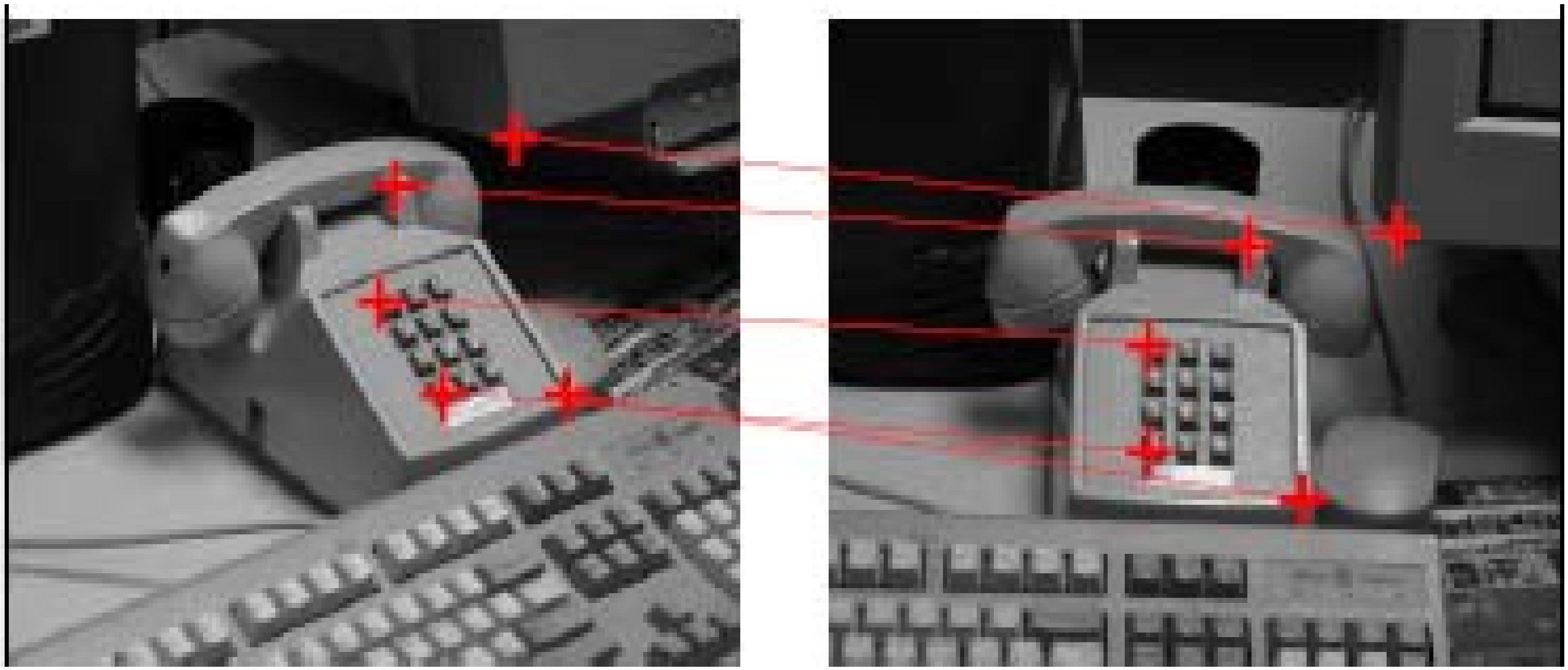
Selim Aksoy
Department of Computer Engineering
Bilkent University
saksoy@cs.bilkent.edu.tr

Slides adapted from Cordelia Schmid and David Lowe, CVPR 2003 Tutorial, and Matthew Brown, Microsoft Research.

Image matching

- Image matching is a fundamental aspect of many problems in computer vision.
  - Object or scene recognition
  - Solving for 3D structure from multiple images
  - Stereo correspondence
  - Image alignment & stitching
  - Image search
  - Motion tracking
- Find “interesting” pieces of the image.
  - Focus attention of algorithms
  - Speed up computation

Image matching applications



- Object recognition: Find correspondences between feature points in training and test images.
- 3D reconstruction: find correspondences between feature points in two images of the same scene.

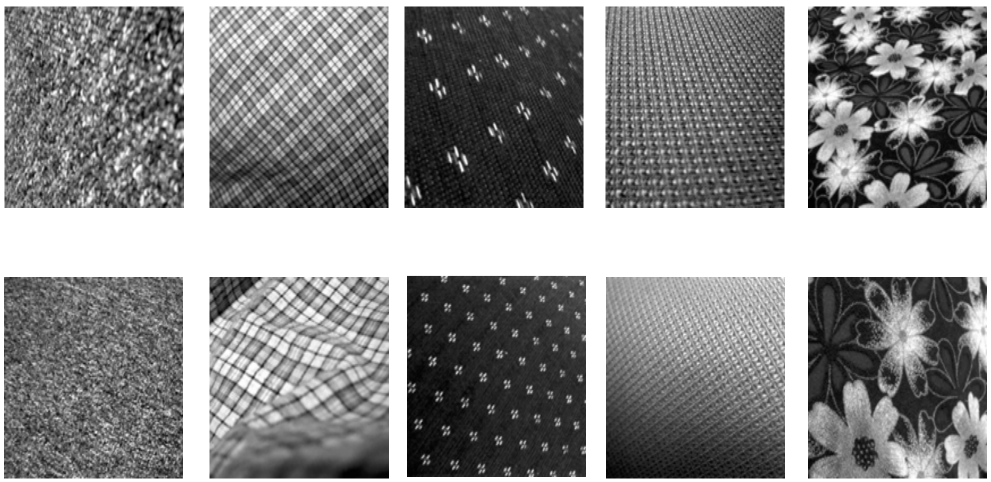
Adapted from Martial Hebert, CMU

Image matching applications

- Stereo correspondence

Image matching applications

- Recognition



Texture recognition

Car detection

Image matching applications

- 3D recognition

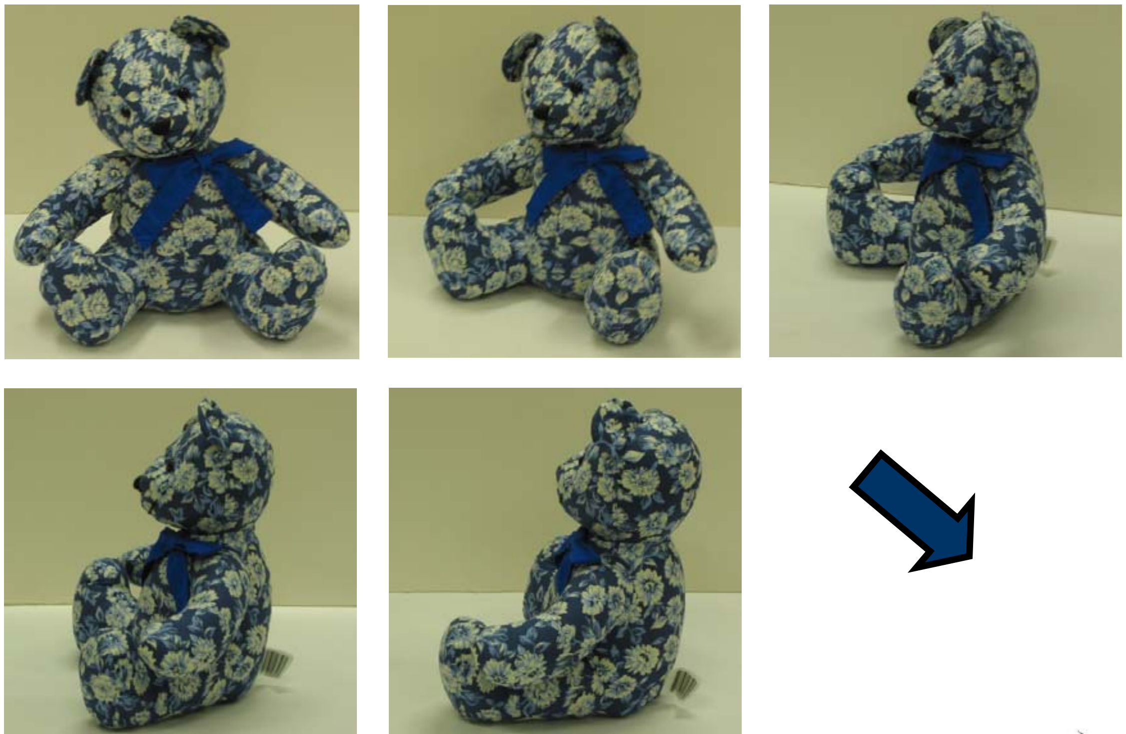
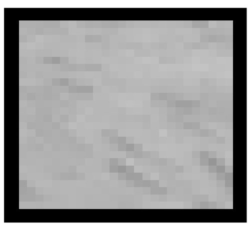
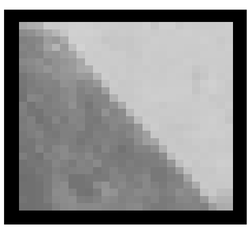


Image matching

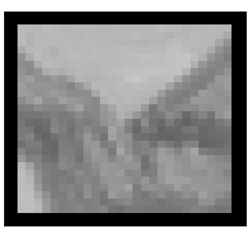
- Matching based on correlation alone.
- Matching based on edge pixels or line segments.
  - Not very discriminant.
- Solution: matching with interest points & correlation.
  - Discrete, reliable and meaningful.



0D structure
➡ not useful for matching



1D structure
➡ edge, can be localized in 1D,
subject to the aperture problem



2D structure
➡ corner, or **interest point**, can be
localized in 2D, good for matching

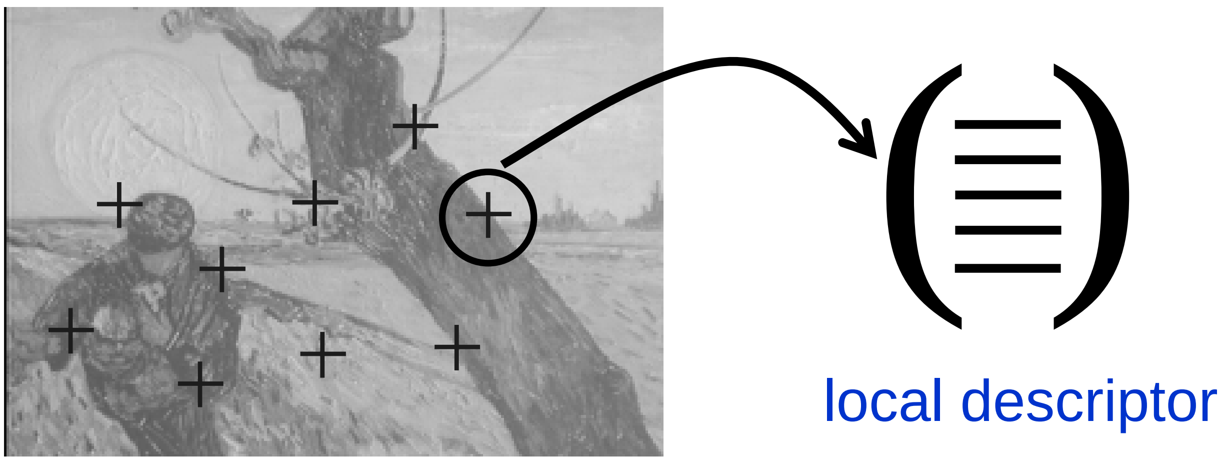
Adapted from Matthew Brown, Microsoft Research

Image matching

- There are three important requirements for feature points to have a better correspondence for matching:
  - Points corresponding to the same scene points should be extracted consistently over different views.
  - They should be invariant to image scaling, rotation and to change in illumination and 3D camera viewpoint.
  - There should be enough information in the neighborhood of the points so that corresponding points can be automatically matched.

Interest points

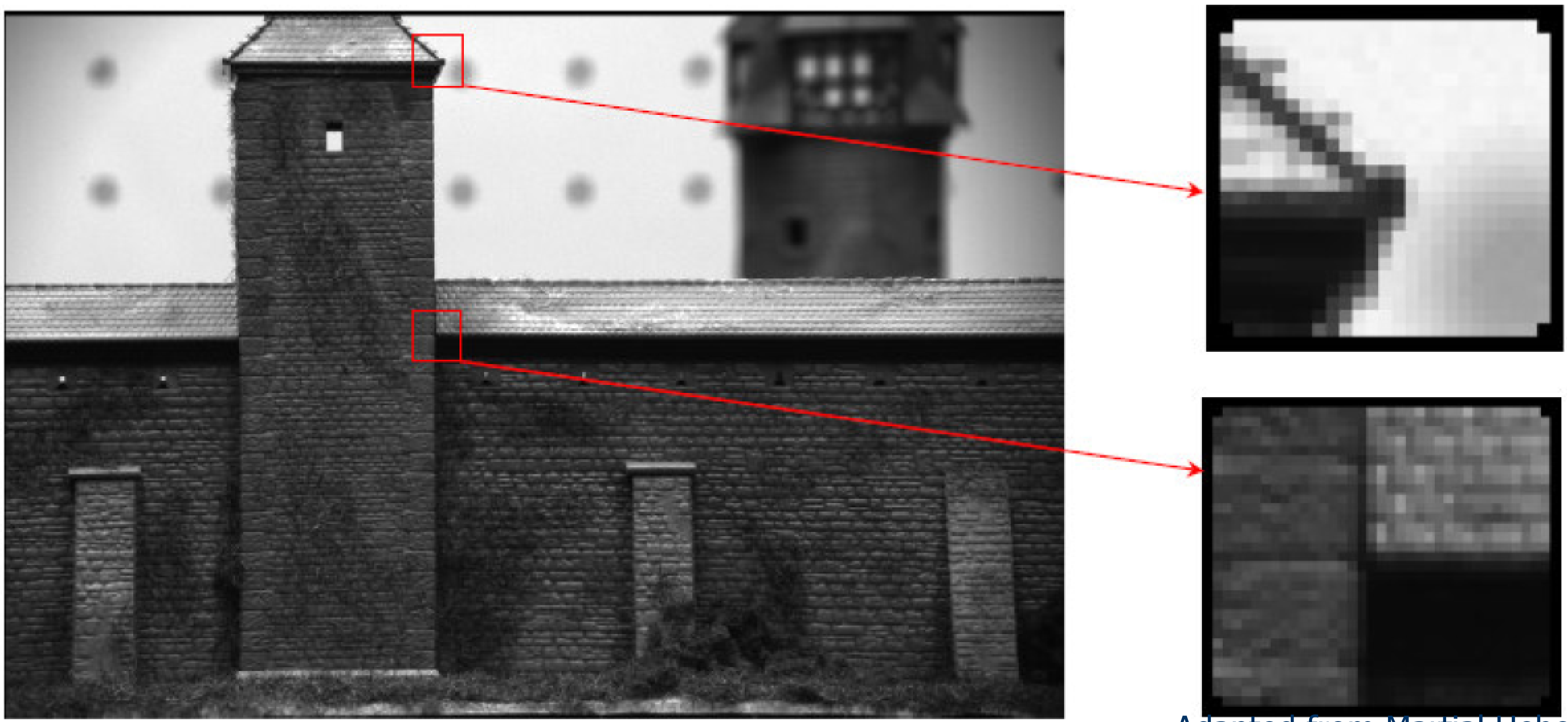
- Interest points: local invariant photometric descriptors.



- *Local* : robust to occlusion/clutter + no segmentation.
- *Photometric* : distinctive.
- *Invariant* : to image transformations + illumination changes.

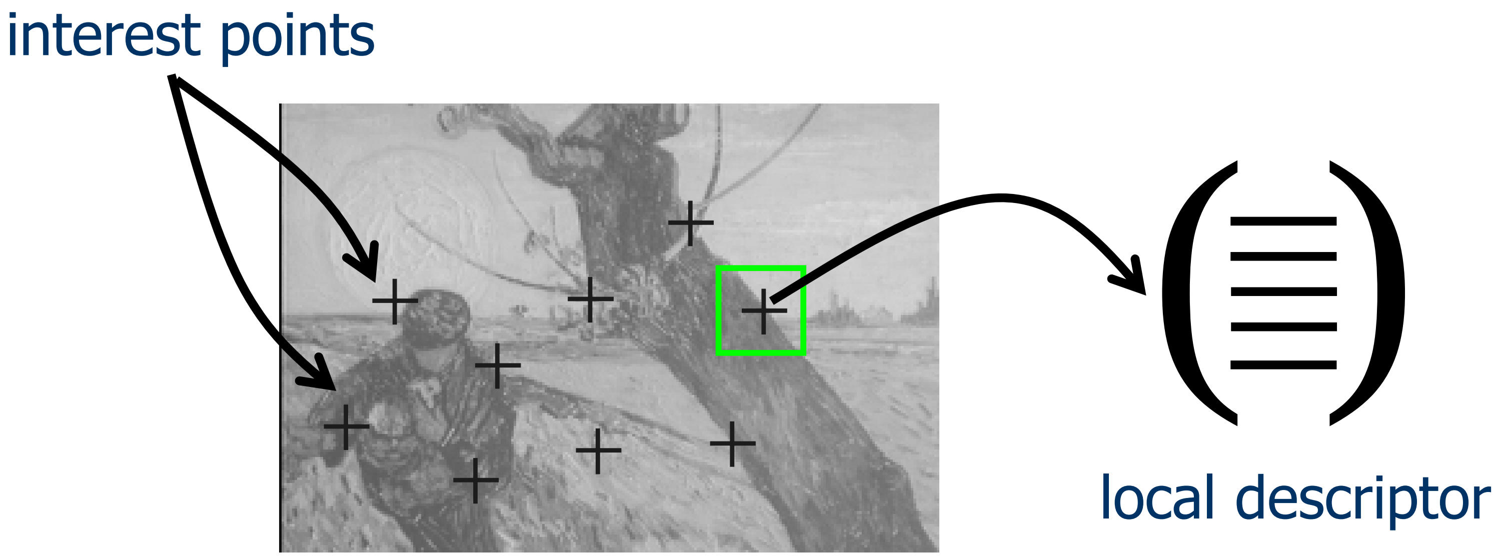
Interest points

- Intuitively junctions or contours.
- Generally more stable features over changes of viewpoint.
- Intuitively large variations in the neighborhood of the point in all directions.



Adapted from Martial Hebert, CMU

Overview of the approach



1. Extraction of interest points (characteristic locations).
2. Computation of local descriptors.
3. Determining correspondences.
4. Selection of similar images.

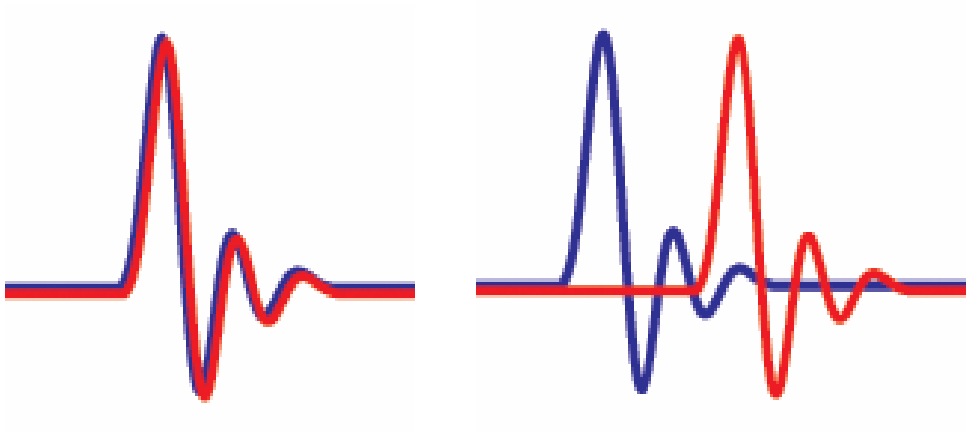
Harris detector

- Based on the idea of auto-correlation.

- Important difference in all directions
  → interest point.

Harris detector

- Auto-correlation function (ACF) measures the self similarity of a signal and is related to sum-of-square difference (SSD).



$$ACF = r(a) = \int_{-\infty}^{\infty} f(x) f(x-a) dx$$

$$SSD = e(a) = \int_{-\infty}^{\infty} (f(x) - f(x-a))^2 dx$$

Adapted from Matthew Brown, Microsoft Research

Harris detector

Auto-correlation function for a point (x, y) and a shift $(\Delta x, \Delta y)$

$$f(x,y) = \sum_{(x_k, y_k) \in W} \left(I(x_k, y_k) - I(x_k + \Delta x, y_k + \Delta y) \right)^2$$

Discrete shifts can be avoided with the auto-correlation matrix:

$$\text{with} \quad I(x_k + \Delta x, y_k + \Delta y) = I(x_k, y_k) + \left(I_x(x_k, y_k) \quad I_y(x_k, y_k) \right) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$f(x,y) = \sum_{(x_k, y_k) \in W} \left(\left(I_x(x_k, y_k) \quad I_y(x_k, y_k) \right) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right)^2$$

Harris detector

$$
= \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} \begin{bmatrix} \sum\limits_{(x_k, y_k) \in W} (I_x(x_k, y_k))^2 & \sum\limits_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) \\ \sum\limits_{(x_k, y_k) \in W} I_x(x_k, y_k) I_y(x_k, y_k) & \sum\limits_{(x_k, y_k) \in W} (I_y(x_k, y_k))^2 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}
$$

Auto-correlation matrix: M

Harris detector

- Auto-correlation matrix
  - captures the structure of the local neighborhood
  - measure based on eigenvalues of M
    - 2 strong eigenvalues → interest point (corner)
    - 1 strong eigenvalue → contour
    - 0 eigenvalue → uniform region
- Eigenvalues are proportional to the principal curvatures of the local auto-correlation function, and form a rotationally invariant description of M.
- Interest point detection:
  - threshold on the eigenvalues
  - local maximum for localization

Harris detector

- To measure the corner strength:

  $$R = \det(M) - k(\mathrm{trace}(M))2$$

  where

  $$\mathrm{trace}(M) = \lambda_1 + \lambda_2$$

  $$\det(M) = \lambda_1 \text{ x } \lambda_2$$

  (λ_1 and λ_2 are the eigenvalues of M).
- R is positive for corners, negative in edge regions, and small in flat regions.

Harris detector

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x . I_x \quad I_{y2} = I_y . I_y \quad I_{xy} = I_x . I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

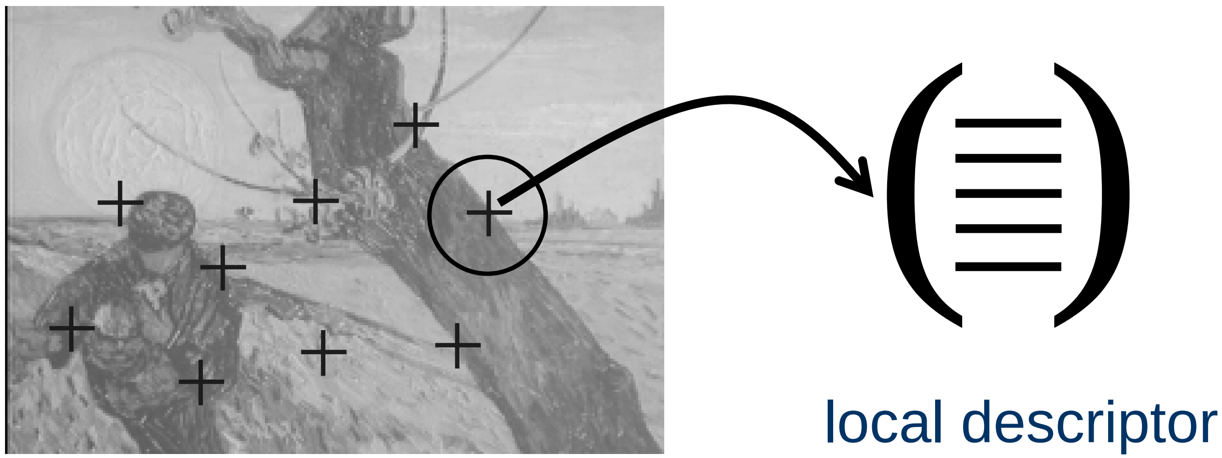
$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

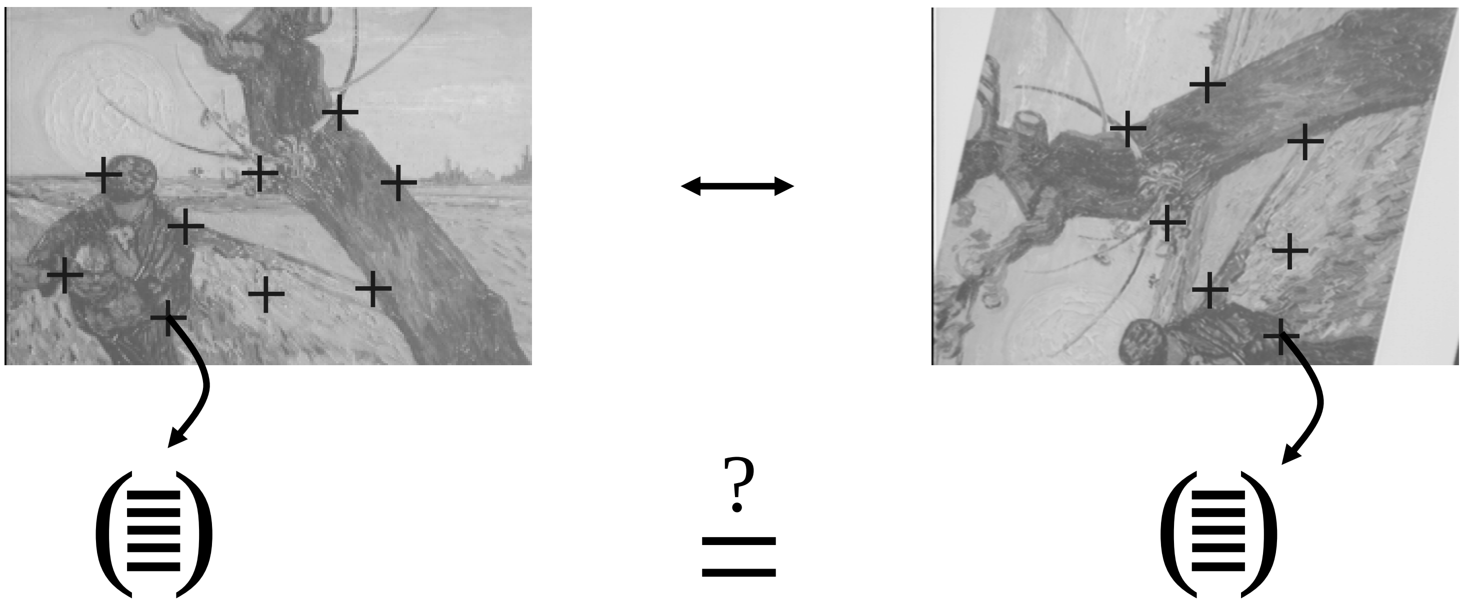
Adapted from Martial Hebert, CMU

Local descriptors



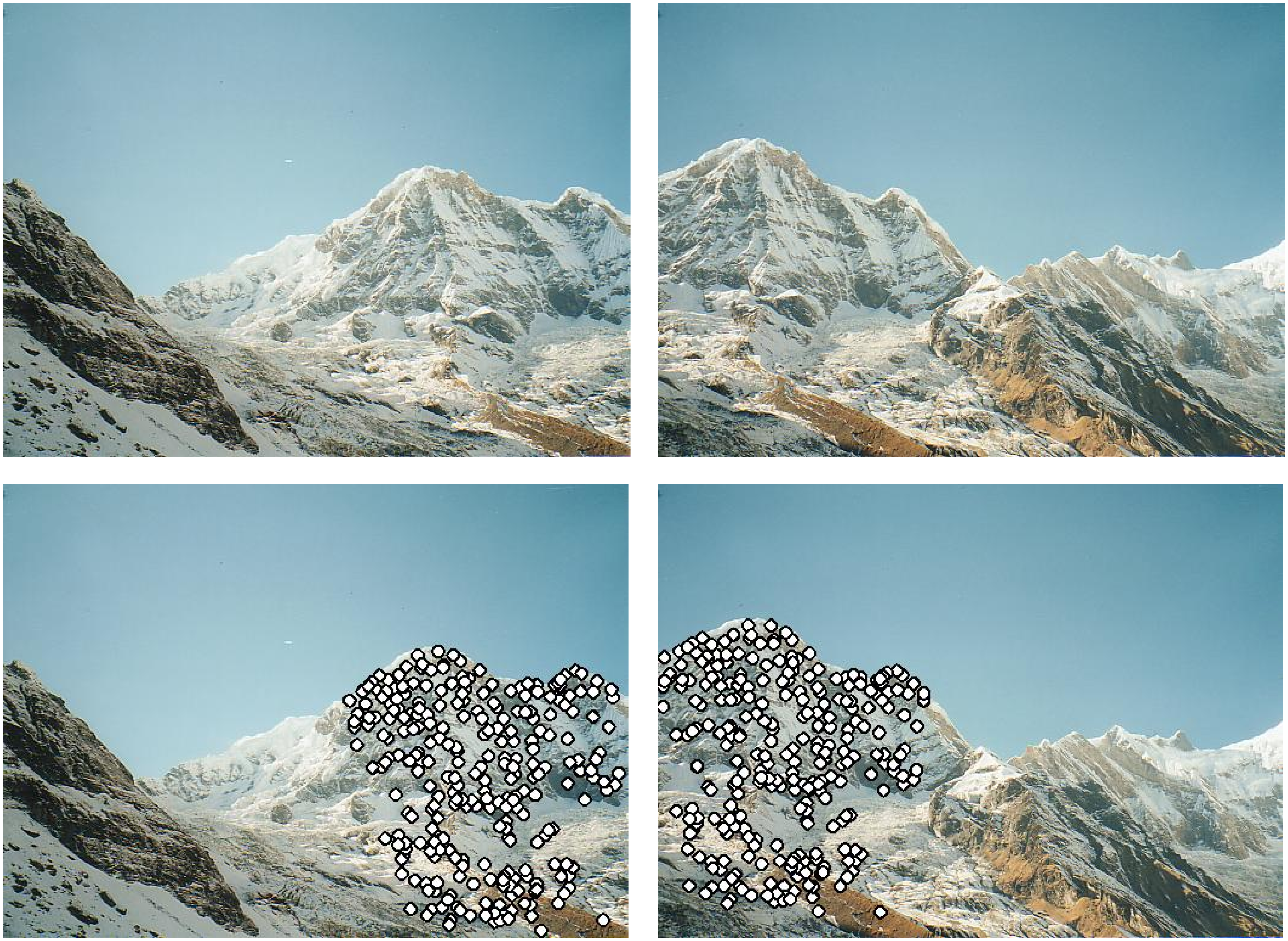
- Descriptors characterize the local neighborhood of a point.
  - Gray values can be used directly.
  - Gray value derivatives or differential invariants can also be used.
  - Values can be normalized for invariance to illumination.

Determining correspondences



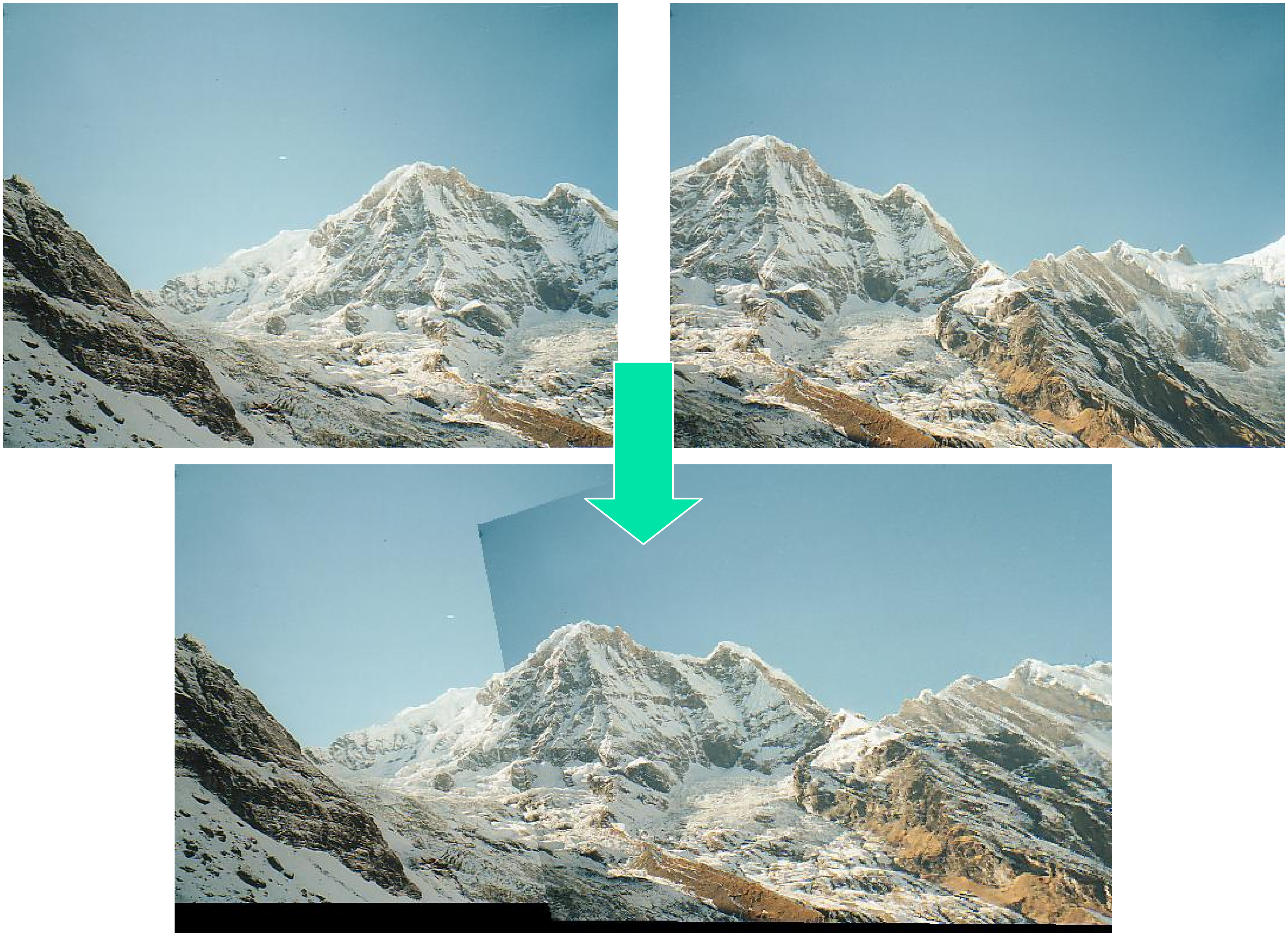
Vector comparison using a distance measure.

Matching examples



Adapted from Matthew Brown, Microsoft Research

Matching examples

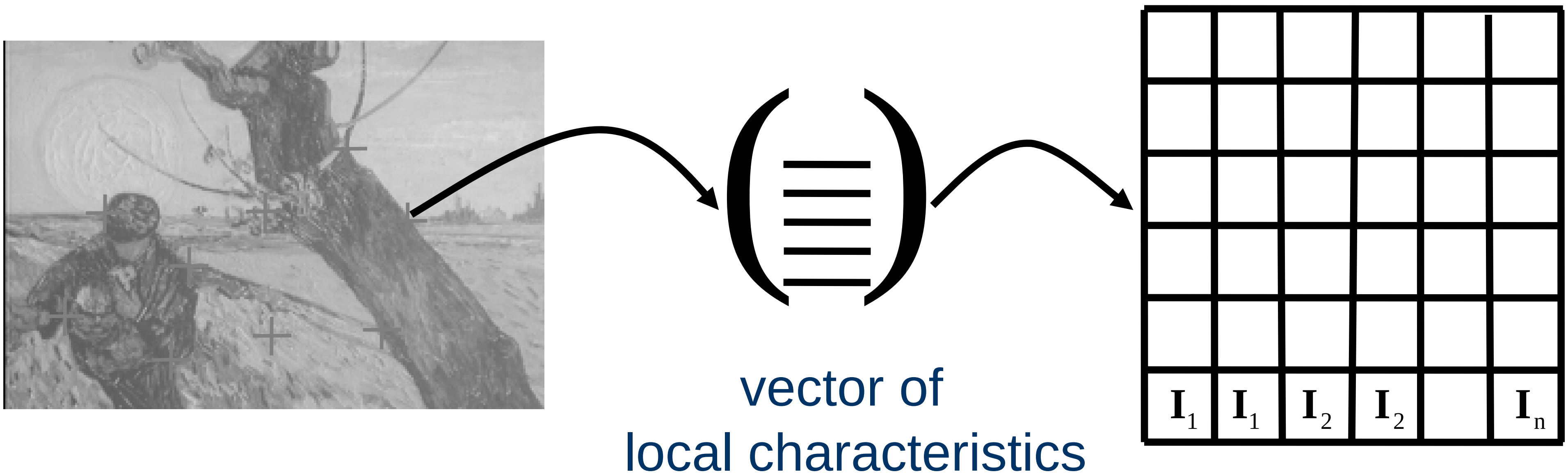


Retrieval examples

- Retrieval in a large database using:
  - voting algorithm
  - additional constraints
- Rapid access with an indexing mechanism.

Retrieval examples

- Voting algorithm:

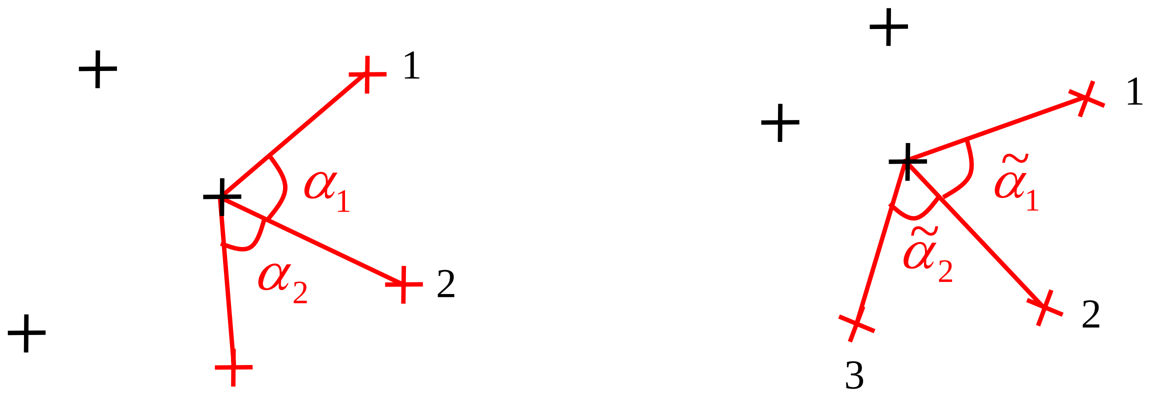


Retrieval examples

- Compute a set of invariant features V around each interest point for each image in the database.
- For a query image compute the same model.
- Compare the vectors for each of the interest points in the query image with all the models in the database.
- If distance is below some threshold then give a vote to the corresponding model.

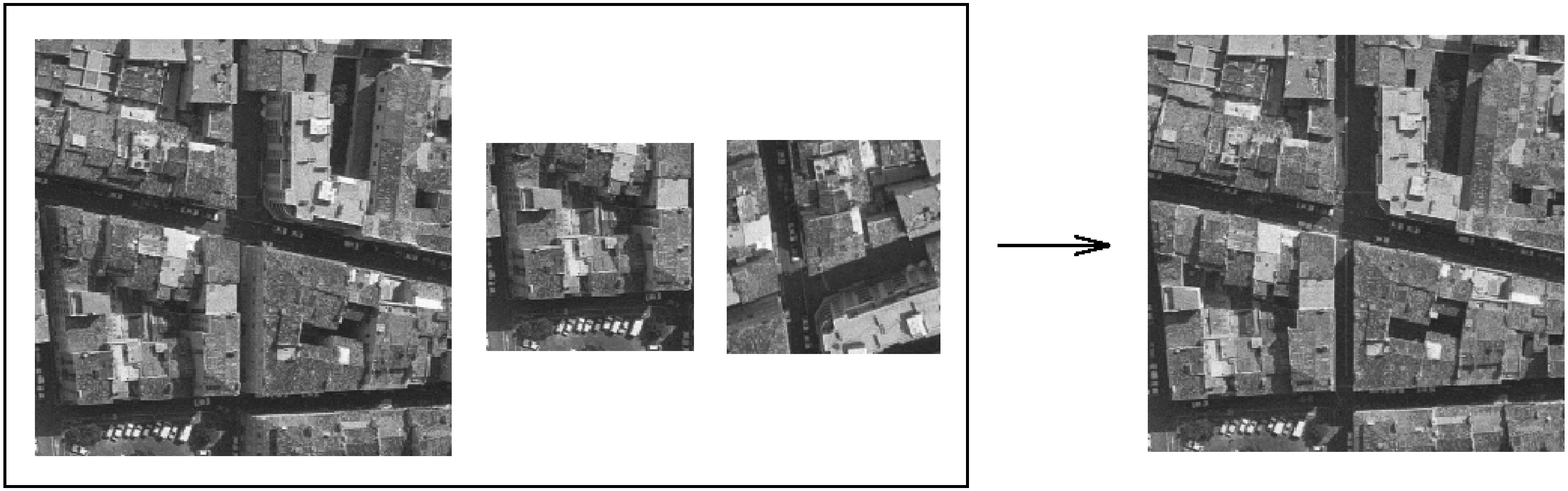
Retrieval examples

- Semi-local constraints
  - neighboring points should match
  - angles, length ratios should be similar



- Global constraints
- Robust estimation of the image transformation (homogaphy, epipolar geometry)

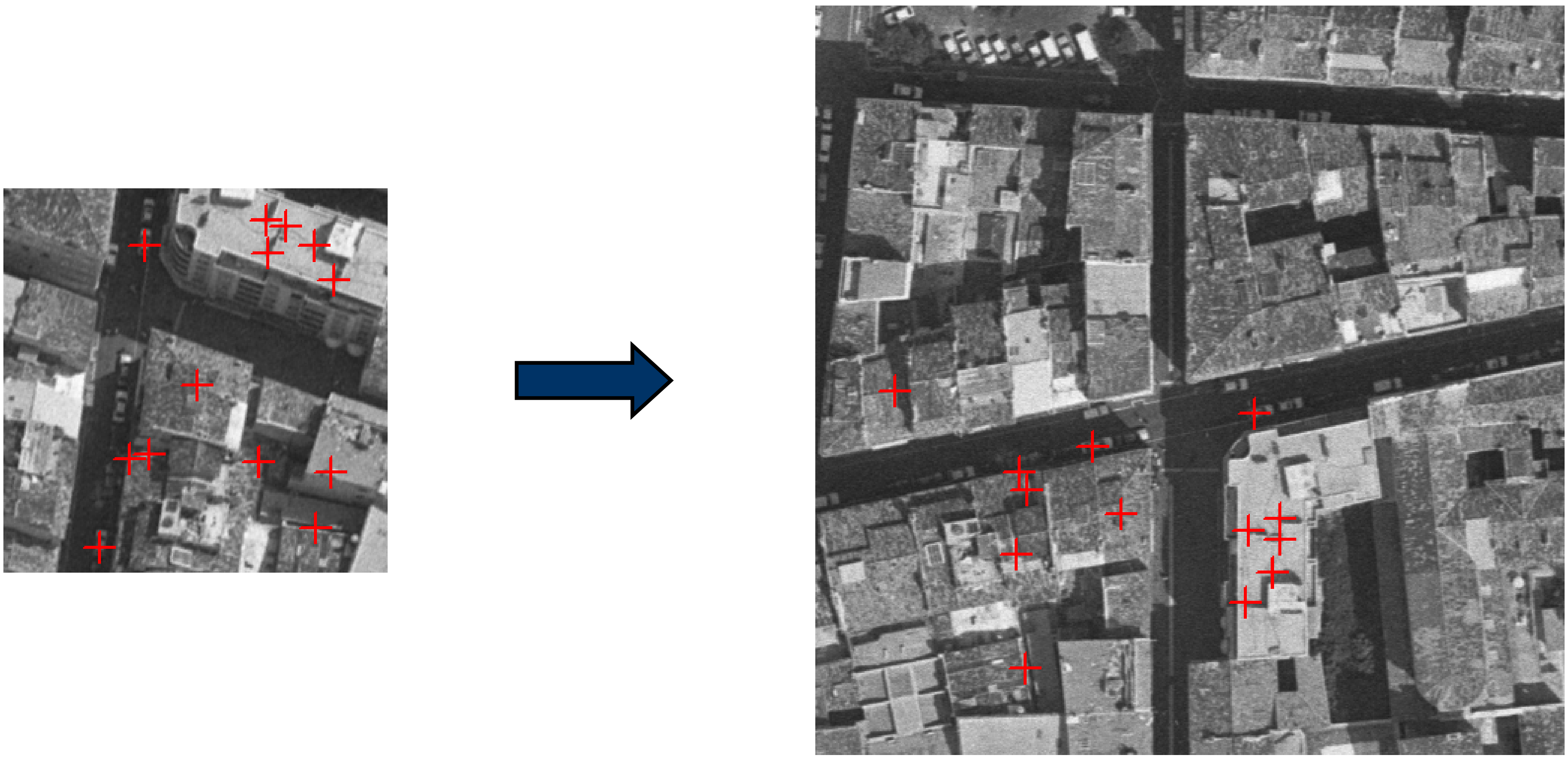
Retrieval examples



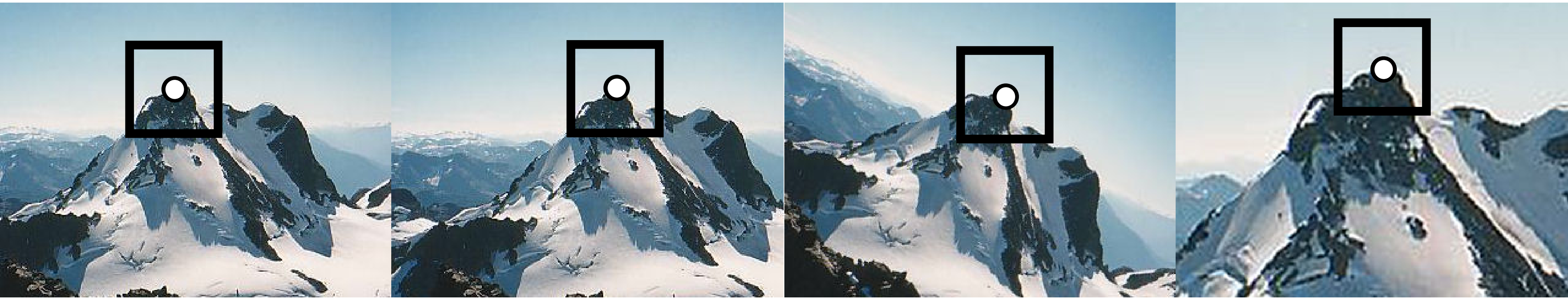
database with ~1000 images

The image on the right is correctly retrieved using any of the images on the left.

Retrieval examples



Invariance

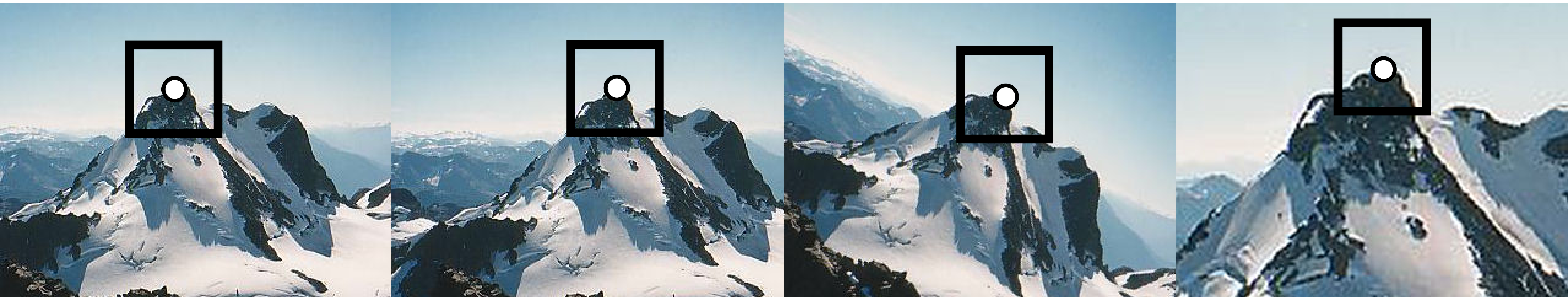


original translated rotated scaled

	Translation	Rotation	Scale
Is Harris invariant?	?	?	?
Is correlation invariant?	?	?	?

Adapted from Matthew Brown, Microsoft Research

Invariance



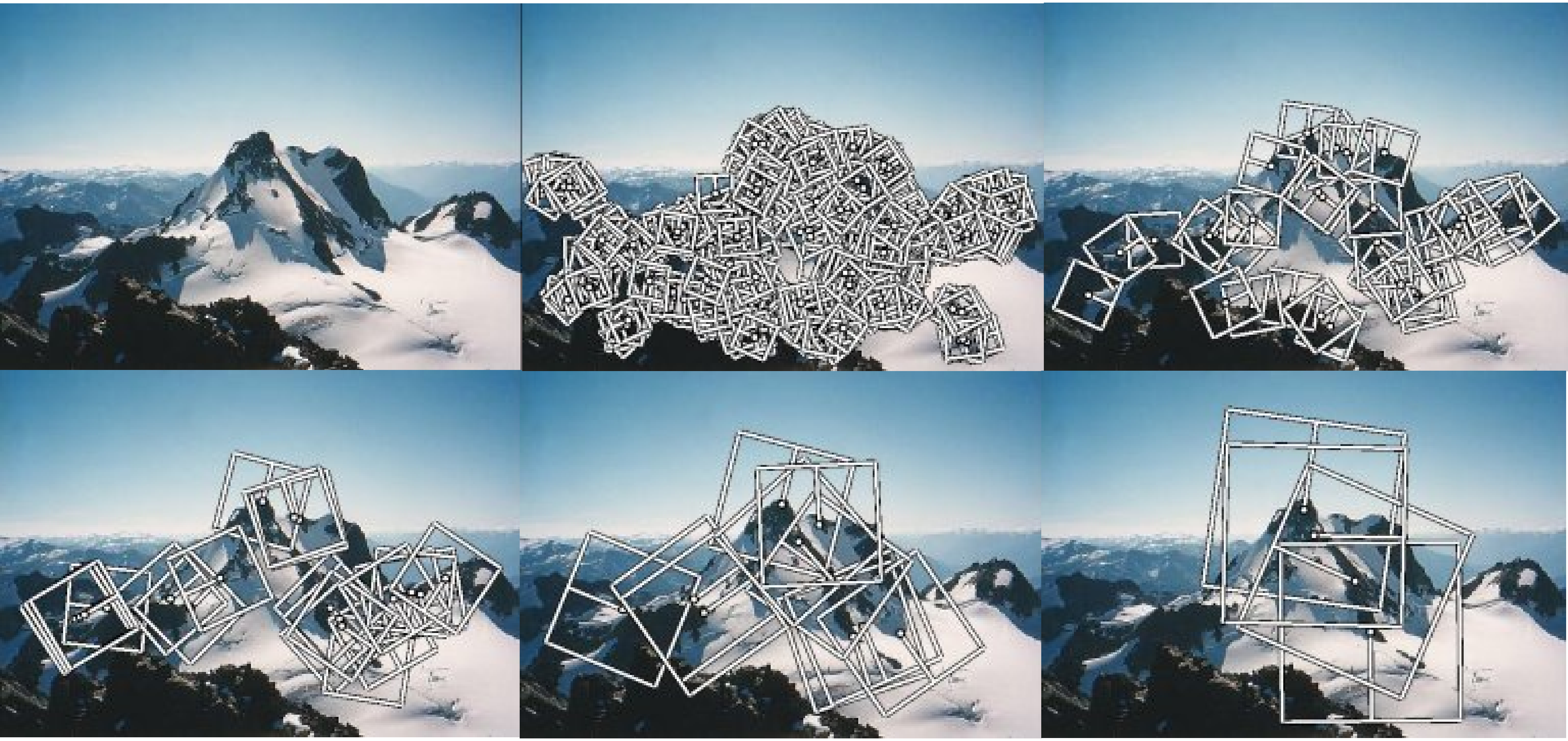
original translated rotated scaled

	Translation	Rotation	Scale
Is Harris invariant?	YES	YES	NO
Is correlation invariant?	YES	NO	NO

Adapted from Matthew Brown, Microsoft Research

Multi-scale oriented patches

- Extract oriented patches at multiple scales.



Adapted from Matthew Brown, Microsoft Research

Summary of the approach

- Very good results in the presence of occlusion and clutter:
  - local information
  - discriminant gray value information
  - invariance to image rotation and illumination
- No invariance to scale and affine changes.
- Solution for more general viewpoint changes:
  - local invariant descriptors to scale and rotation
  - extraction of invariant points and regions

Approach for matching and recognition

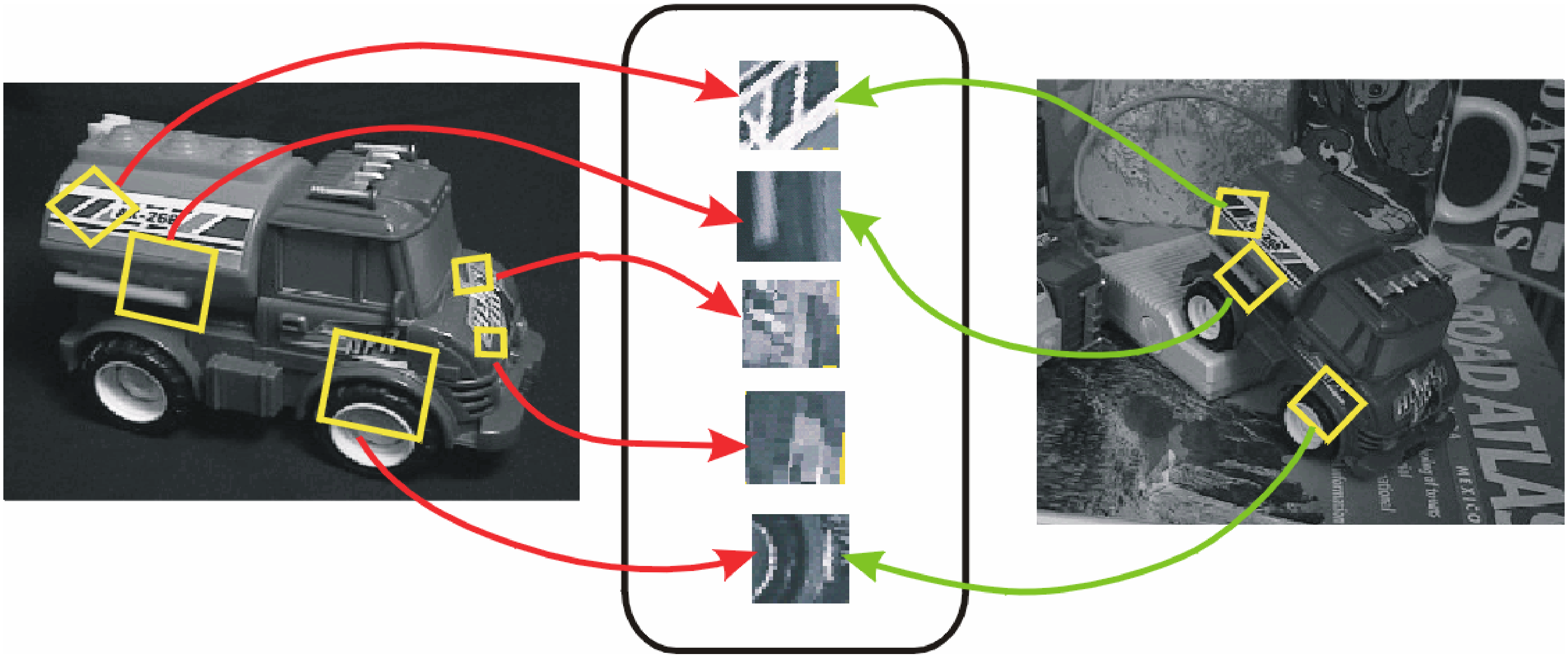
- Detection of interest points/regions
  - Harris detector (extension to scale and affine invariance)
  - Blob detector based on Laplacian
- Computation of descriptors for each point
  - Gray value patch, differential invariants, steerable filter, SIFT descriptor
- Similarity of descriptors
  - Correlation, Mahalanobis distance, Euclidean distance
- Semi-local constraints
  - Geometrical or statistical relations between neighborhood points
- Global verification
  - Robust estimation of geometry between images

SIFT (Scale Invariant Feature Transform)

- The original Harris operator was not invariant to scale and its descriptor was not invariant to rotation.
- For better image matching, David Lowe's goal was to develop an operator that is invariant to scale and rotation.
- The operator he developed is both a detector and a descriptor, and can be used for both image matching and object recognition.

Idea of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.



SIFT Features

Claimed advantages of SIFT

- **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.
- **Extensibility**: can easily be extended to wide range of differing feature types, with each adding robustness.

Overall approach for SIFT

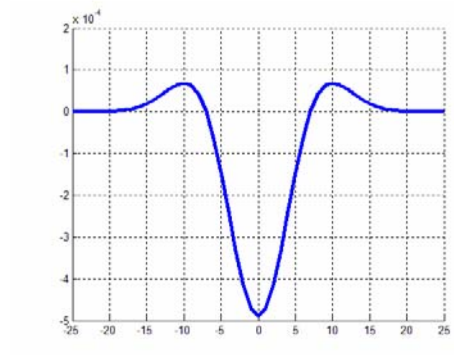
1. Scale space extrema detection
   - Search over multiple scales and image locations.
2. Keypoint localization
   - Fit a model to determine location and scale.
   - Select keypoints based on a measure of stability.
3. Orientation assignment.
   - Compute best orientation(s) for each keypoint region.
4. Keypoint description
   - Use local image gradients at selected scale and rotation to describe each keypoint region.

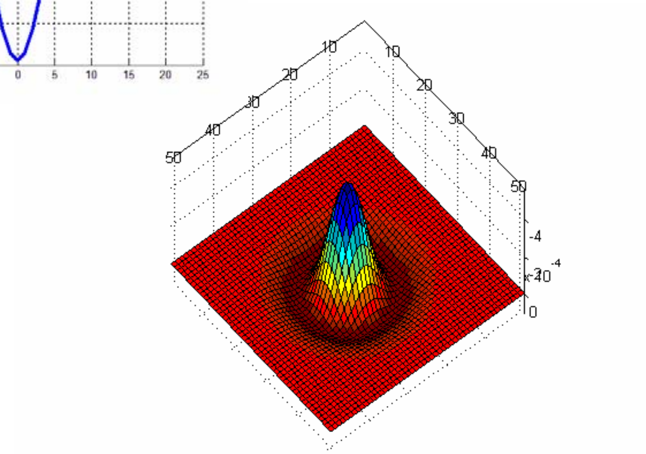
Scale space extrema detection

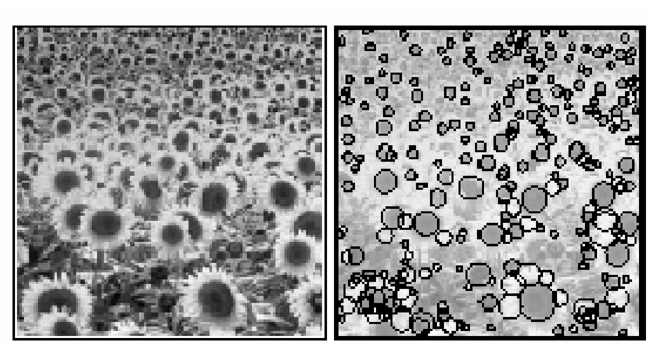
- Goal: Identify locations and scales that can be repeatably assigned under different views of the same scene or object.
- Method: search for stable features across multiple scales using a continuous function of scale.
- Prior work has shown that under a variety of assumptions, the best function is a Gaussian function.
- The scale space of an image is a function $L(x,y,\sigma)$ that is produced from the convolution of a Gaussian kernel (at different scales) with the input image.

Scale space interest points

- Laplacian of Gaussian kernel
  - Scale normalized
  - Proposed by Lindeberg
- Scale space detection
  - Find local maxima across scale/space
  - A good "blob" detector







$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

$$\nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

Scale space interest points

- Scale space function L
  - Gaussian convolution

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where σ is the width of the Gaussian

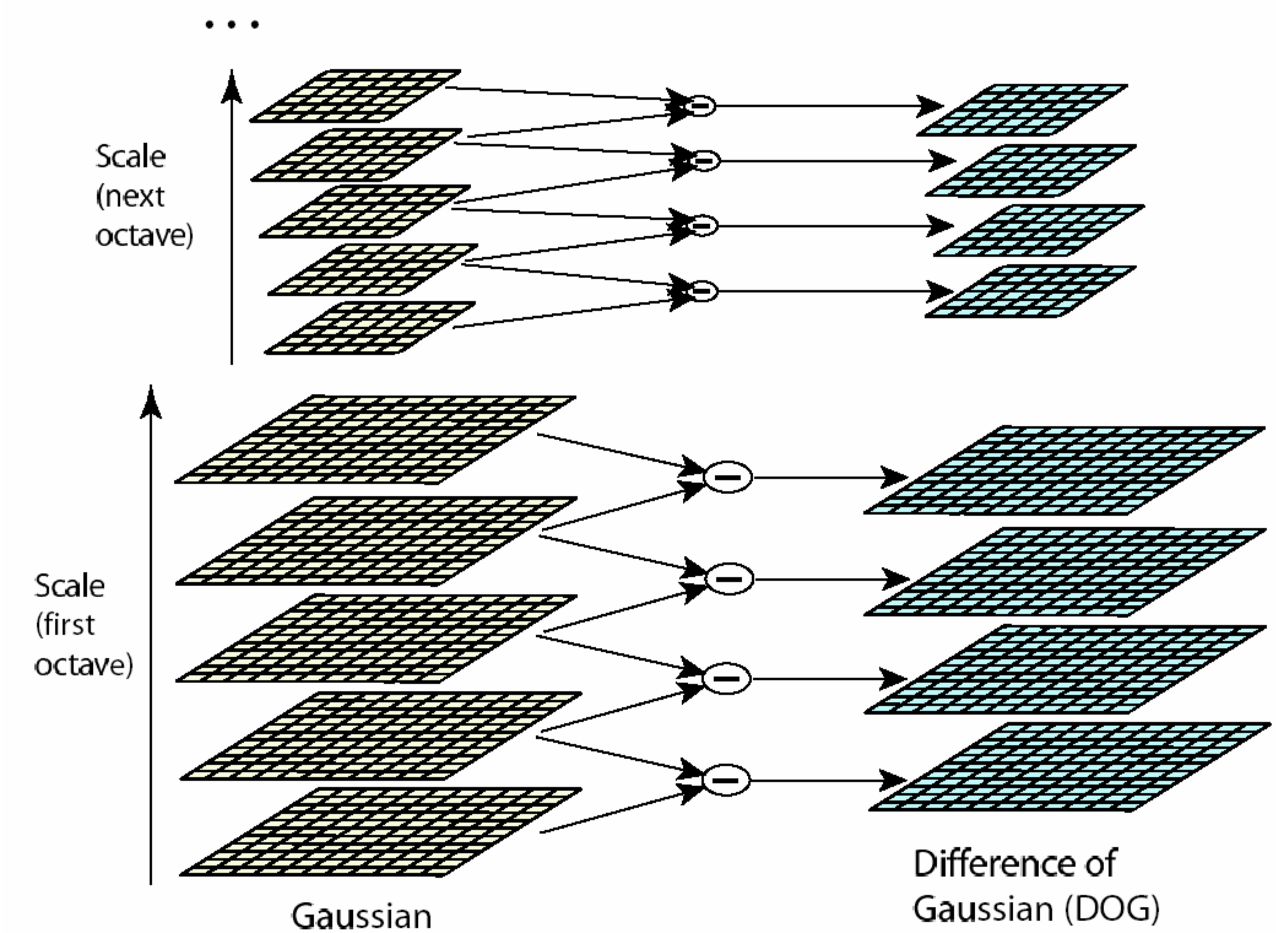
  - Difference of Gaussian kernel is a close approximate to scale-normalized Laplacian of Gaussian

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

2 scales: σ and kσ

  - Can approximate the Laplacian of Gaussian kernel with a difference of separable convolutions

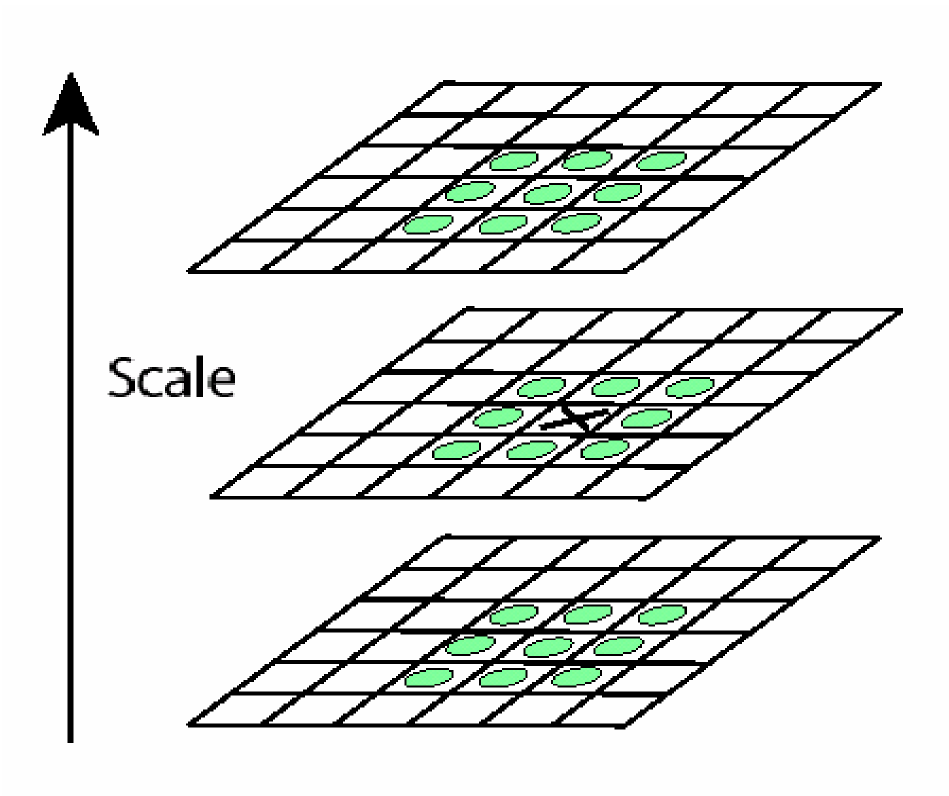
Lowe’s pyramid scheme



For each octave of scale space, the initial image is repeatedly convolved with Gaussian to produce the set of scale space images (left). Adjacent Gaussian images are subtracted to produce difference of Gaussian images (right). After each octave Gaussian image is downsampled by a factor of 2.

Keypoint localization

- Detect maxima and minima of difference of Gaussian in scale space.
- Each point is compared to its 8 neighbors in the current image and 9 neighbors each in the scales above and below.



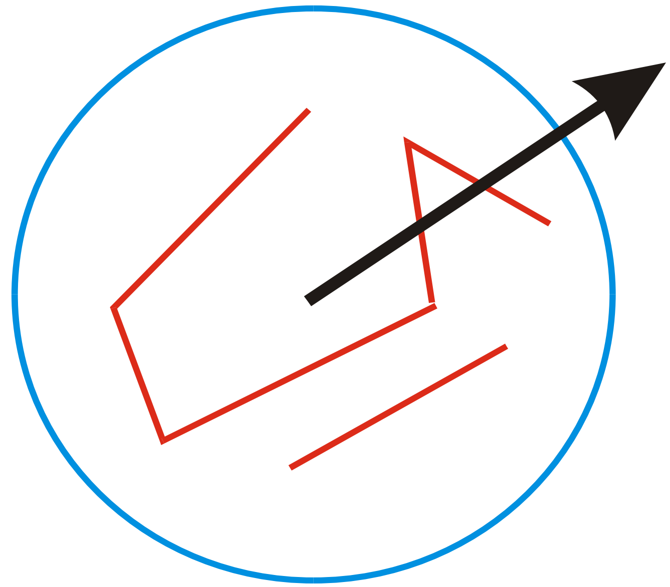
- Select only if it is greater or smaller than all the others.
- For each max or min found, output is the location and the scale.

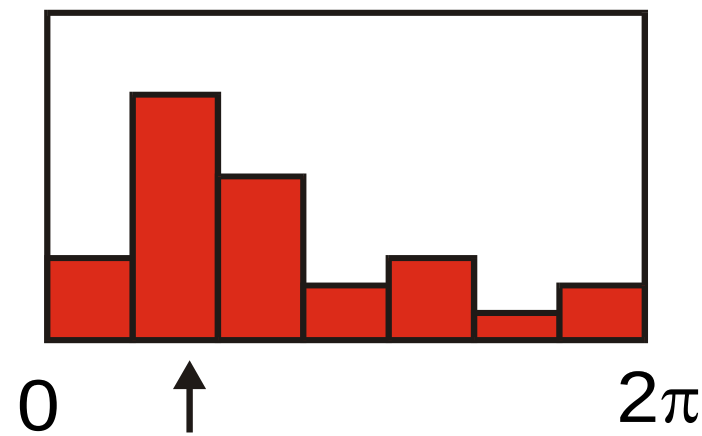
Keypoint localization

- Detailed keypoint determination
  - Sub-pixel and sub-scale location scale determination.
  - Ratio of principal curvature to reject edges and flats (like detecting corners).
- Once a keypoint candidate is found, perform a detailed fit to nearby data to determine
  - location, scale, and ratio of principal curvatures.
- In initial work keypoints were found at location and scale of a central sample point.
- In newer work, they fit a 3D quadratic function to improve interpolation accuracy.
- The Hessian matrix was used to eliminate edge responses.

Orientation assignment

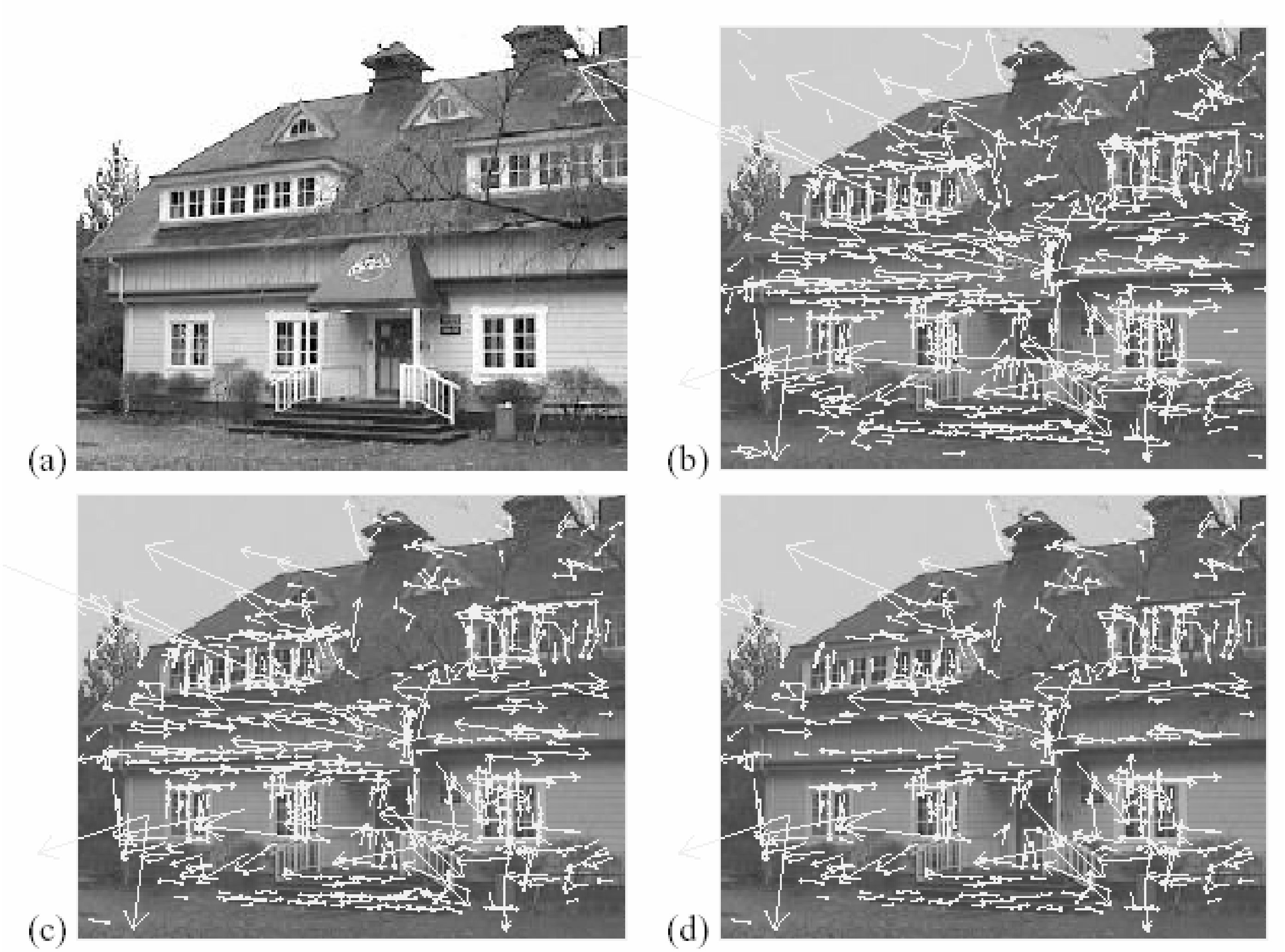
- Create histogram of local gradient directions computed at selected scale.
- Assign canonical orientation at peak of smoothed histogram.
- Each key specifies stable 2D coordinates (x, y, scale, orientation).





Example of keypoint detection

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach).



(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

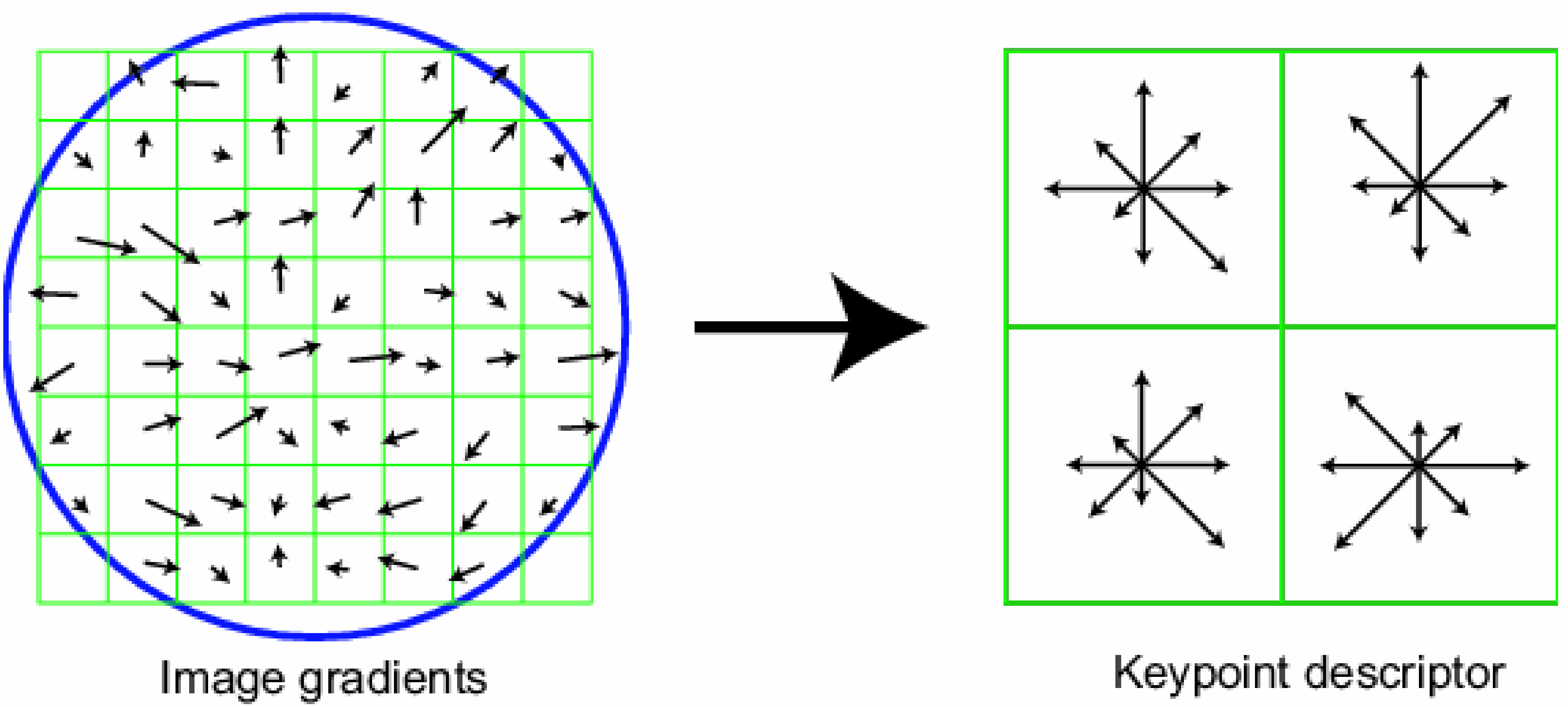
Keypoint descriptors

- At this point, each keypoint has
  - location,
  - scale,
  - orientation.
- Next step is to compute a descriptor for the local image region about each keypoint that is
  - highly distinctive,
  - invariant as possible to variations such as changes in viewpoint and illumination.

Lowe's keypoint descriptor

- Use the normalized circular region about the keypoint.
  - Rotate the window to standard orientation.
  - Scale the window size based on the scale at which the point was found.
- Compute gradient magnitude and orientation at each point in the region.
- Weight them by a Gaussian window overlaid on the circle.
- Create an orientation histogram over the 4x4 subregions of the window.
- 4x4 descriptors over 16x16 sample array were used in practice. 4x4 times 8 directions gives a vector of 128 values.

Lowe’s keypoint descriptor



In the paper, 4x4 arrays of 8 bin histogram is used, resulting in a total of 128 features for one keypoint (shown with 2x2 descriptors over 8x8 array).

Examples: image retrieval

change in viewing angle

• • •

> 5000 images

Examples: image retrieval

22 correct matches

Examples: image retrieval

change in viewing angle
+ scale change

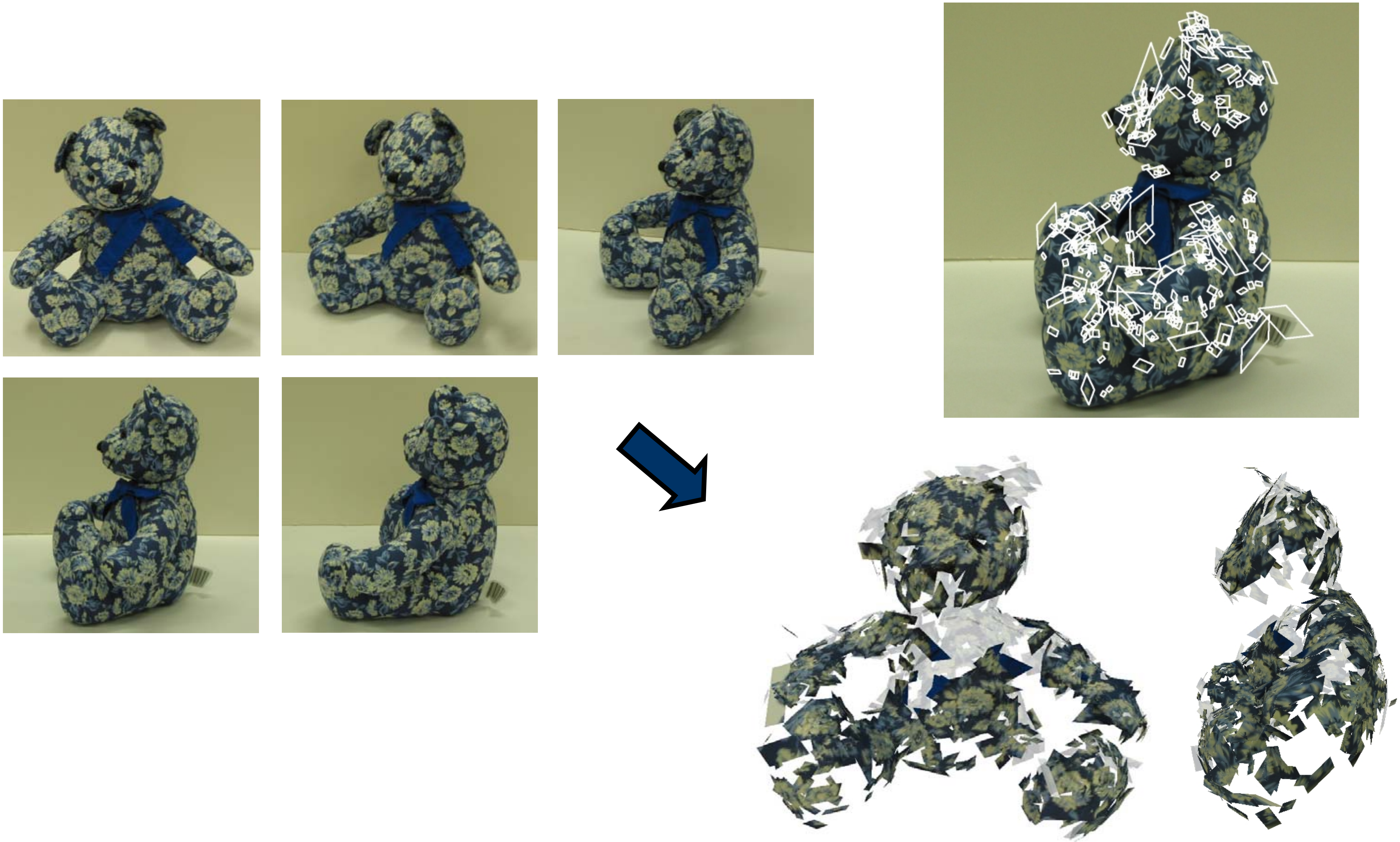
• • •

> 5000 images

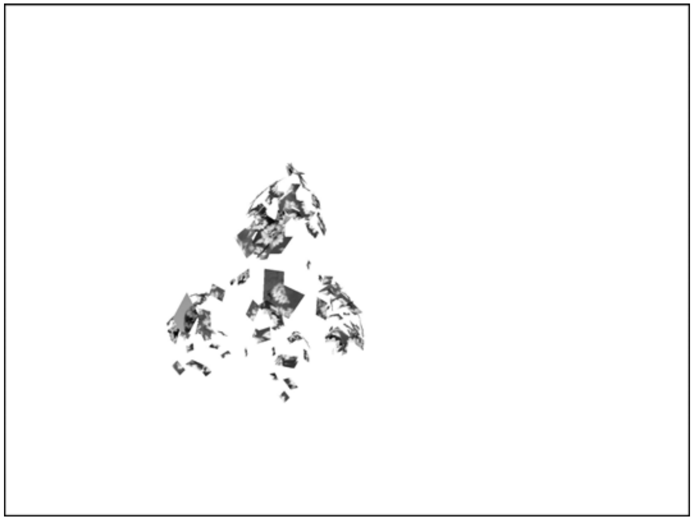
Examples: image retrieval

33 correct matches

Examples: 3D recognition



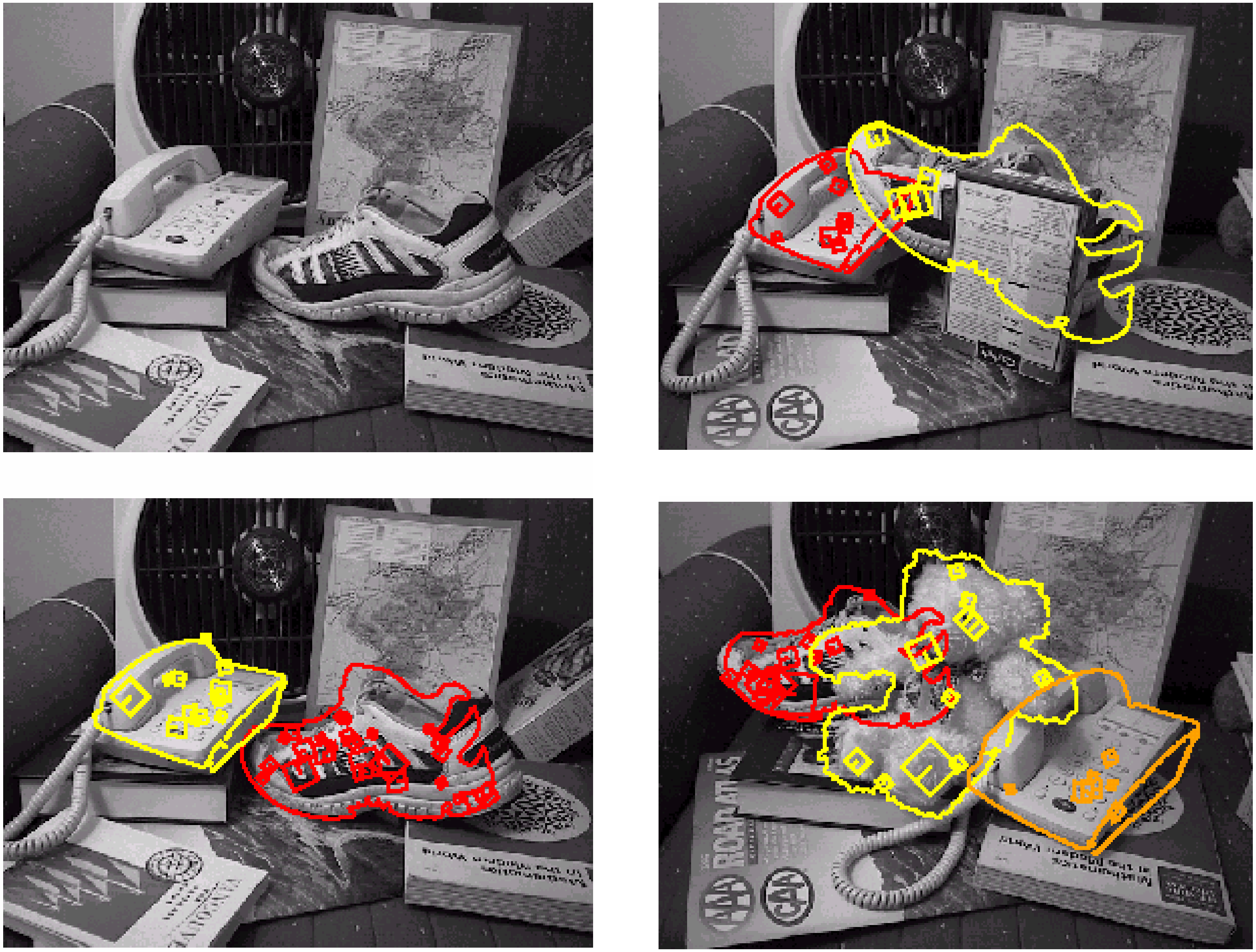
Examples: 3D recognition



3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints

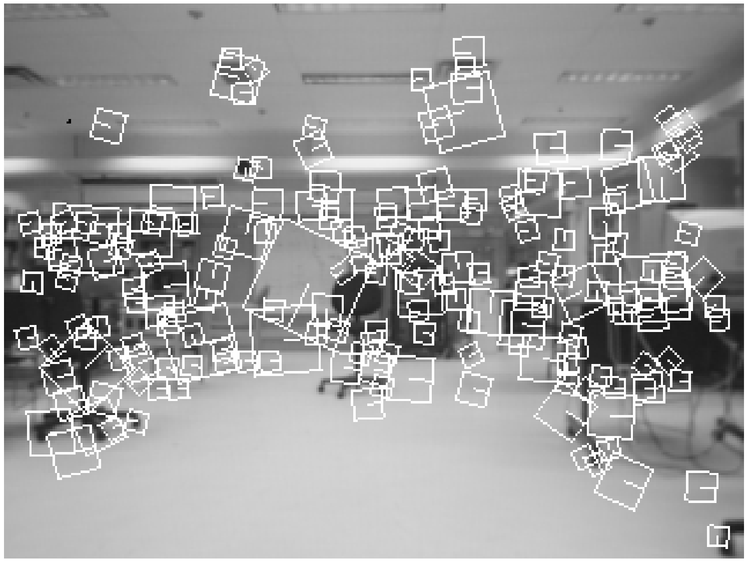
Examples: 3D recognition

Examples: 3D recognition



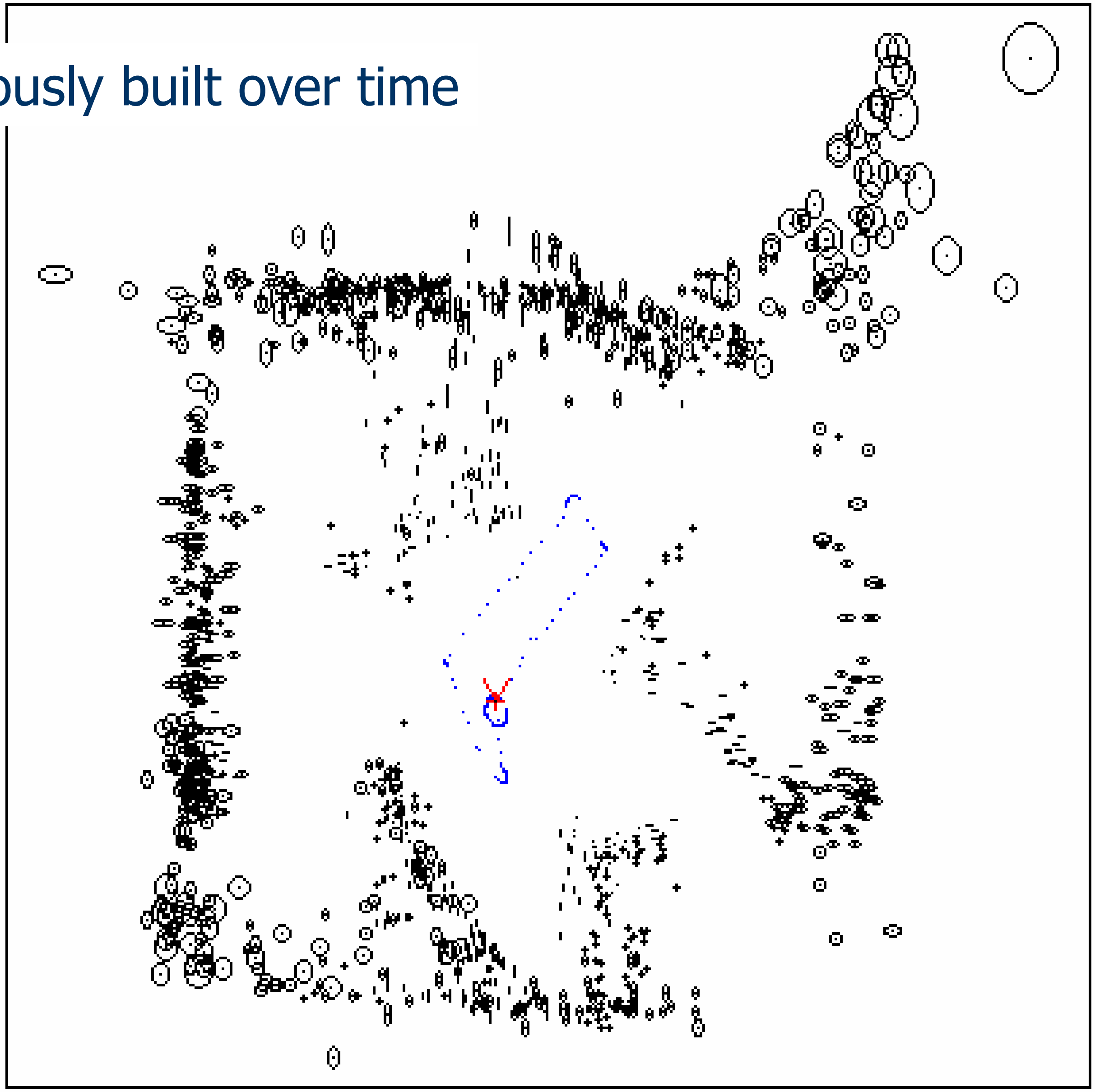
Examples: location recognition

Examples: robot localization



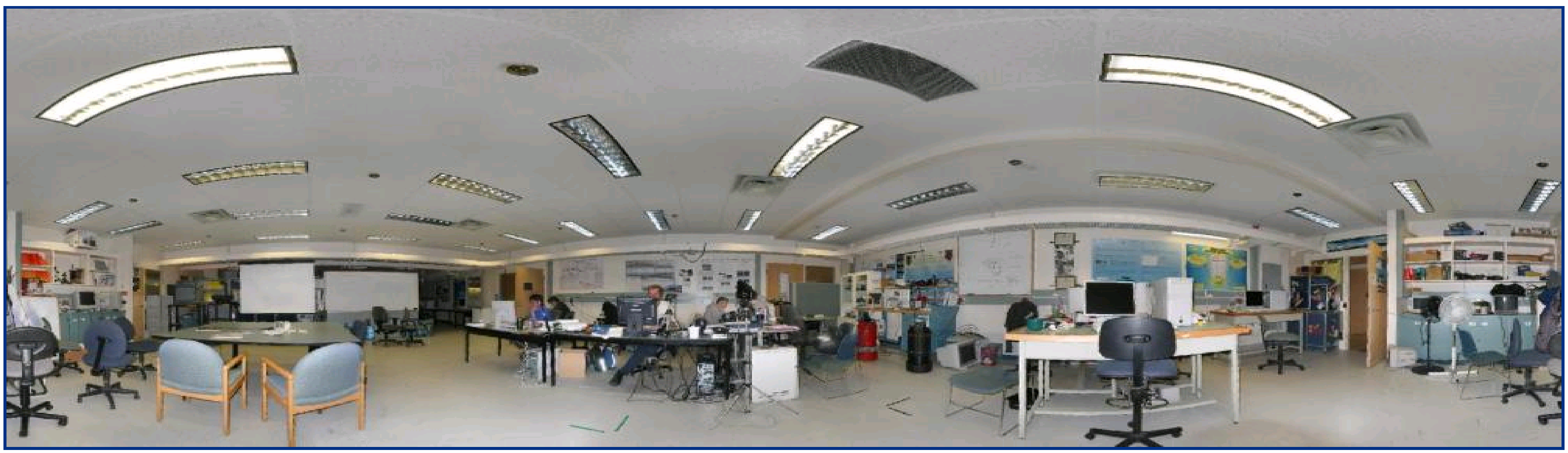
Examples: robot localization

Map continuously built over time



Examples: panaromas

- Matthew Brown and David Lowe
- Recognize overlap from an unordered set of images and automatically stitch together.
- SIFT features provide initial feature matching.
- Image blending at multiple scales hides the seams.



Panorama of Lowe's lab automatically assembled from 143 images

Examples: panaromas

Input images

Multiple panaromas from an unordered image set

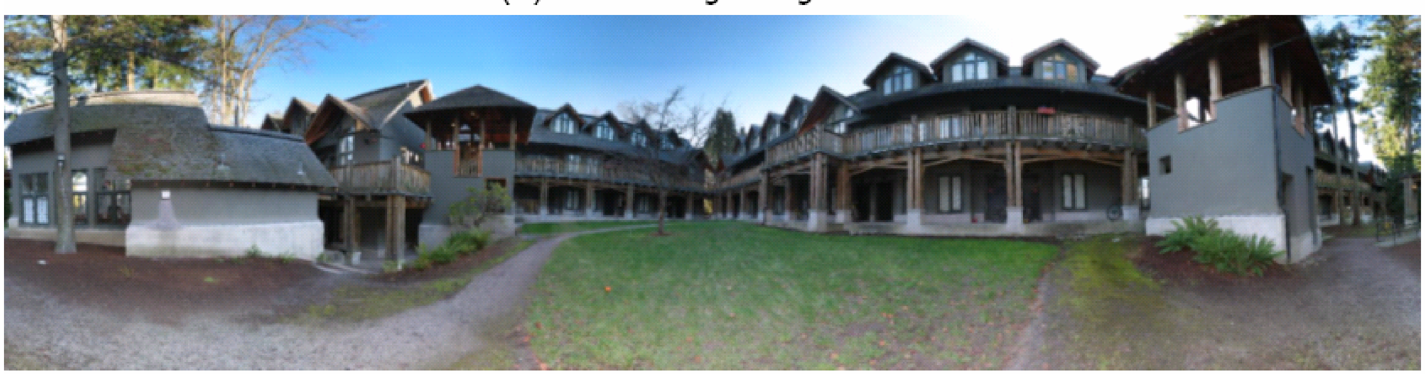
Output panorama 1

Examples: panaromas

Image registration and blending

(a) *40 of 80 images registered*

(b) *All 80 images registered*



(c) *Rendered with multi-band blending*