Image Matching

Liangliang Nan

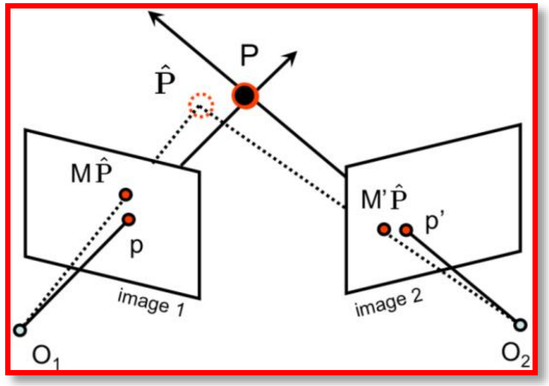
Today's Agenda

- Review last lecture
  - Reconstruct 3D Geometry
- Image matching

Triangulation

- Find coordinates of 3D points from projections in two views
  - The linear method
    - Easy to solve and very efficient
    - Any number of corresponding image points
    - Can handle multiple views
    - Used as initialization to advanced methods

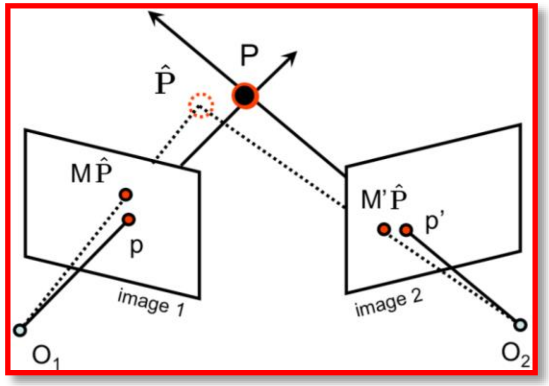
$$AP = 0 \qquad A = \begin{bmatrix} xM_3 - M_1 \\ yM_3 - M_2 \\ x'M_3' - M_1' \\ y'M_3' - M_2' \end{bmatrix}$$



Triangulation

- Find coordinates of 3D points from projections in two views
  - The linear method
    - Easy to solve and very efficient
    - Any number of corresponding image points
    - Can handle multiple views
    - Used as initialization to advanced methods
  - The non-linear method

$$AP = 0 \qquad A = \begin{bmatrix} xM_3 - M_1 \\ yM_3 - M_2 \\ x'M_3' - M_1' \\ y'M_3' - M_2' \end{bmatrix}$$

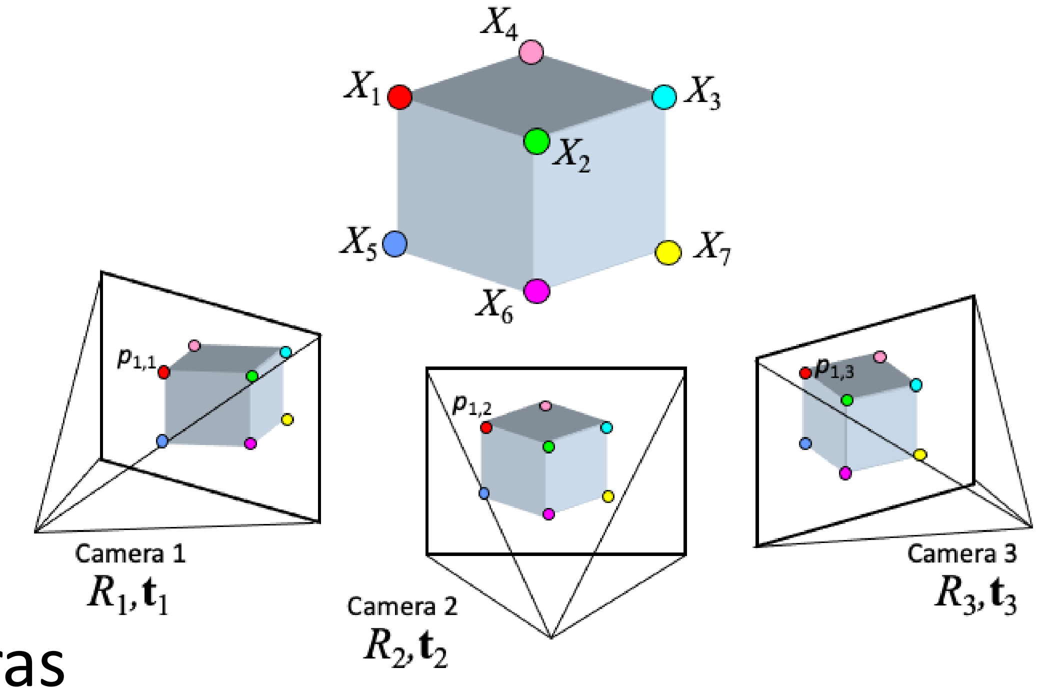


$$\min_{\hat{P}} \sum_i \|M\hat{P}_i - p_i\|^2$$

Reprojection error

Structure from Motion

- Structure
  – 3D geometry of the scene/object
- Motion
  – Camera locations and orientations
- Structure from Motion
  – Compute geometry from moving cameras
  – Simultaneously refine structure and motion



Bundle Adjustment

- Minimize sum of squared re-projection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j) - \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} \right\|^2$$

- Minimizing this function is called *bundle adjustment*
- Initialization
  - From chained 2-view reconstruction
    - Relative motion can be estimated from the corresponding images points
    - 3D points can be estimated from the relative motion using triangulation

Quizzes

- What are the differences between **bundle adjustment** and the **non-linear method for triangulation**?
- Given a camera that can **only translate** along a certain direction, can it be used to take images for 3D reconstruction?
- Given a camera that can **freely rotate only**, can it be used to take images for 3D reconstruction?

Today’s Agenda

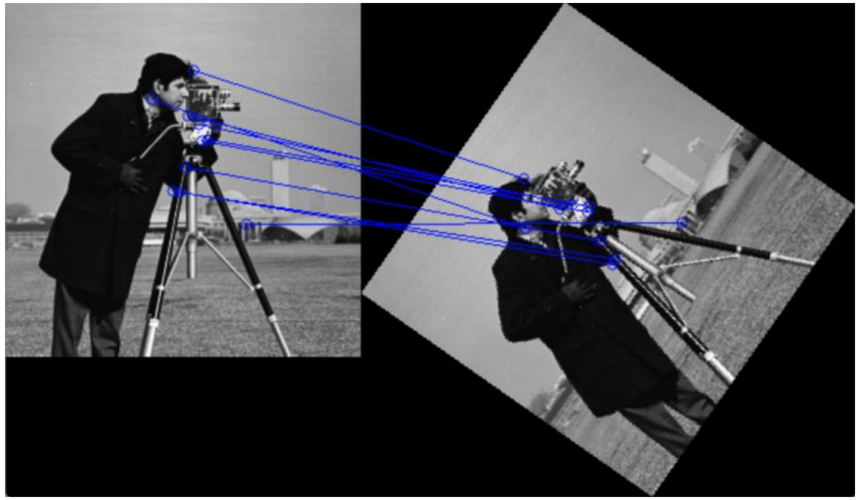
- Review last lecture
  - Reconstruct 3D Geometry
- Image matching

Imaging Matching

- Find Corresponding Image Points
  - Key points
  - Image descriptors
  - Matching

Key Points

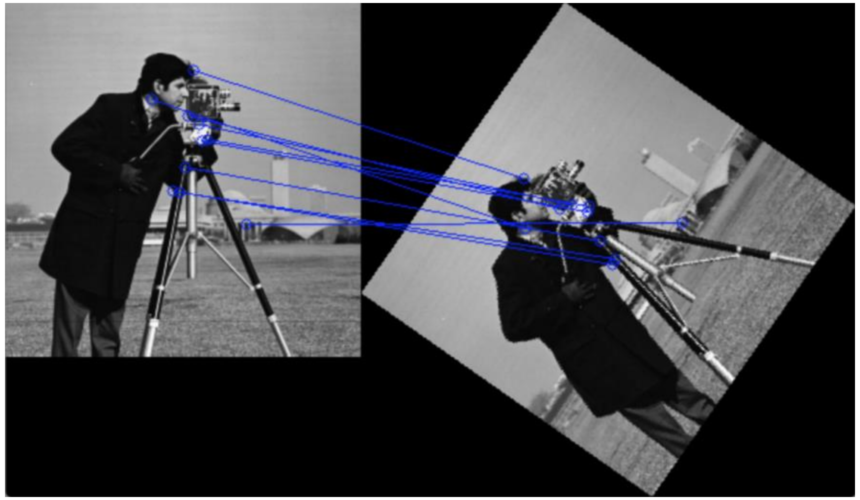
- Distinctive locations
- Interesting or stand out
- Remain unchanged
    - Rotate, translate, shrink/expand, distortion …



10 matching pairs

Descriptors

- What makes a key point different from other key points?
  - The way we **describe** the key points
  - High-dimensional vectors
- Feature (key point + descriptor)
  - e.g., **SIFT**, SURF



10 matching pairs

SIFT

David Lowe

FOLLOW

Professor Emeritus, Computer Science Dept., University of British Columbia
Verified email at cs.ubc.ca - Homepage

Computer Vision   Object Recognition

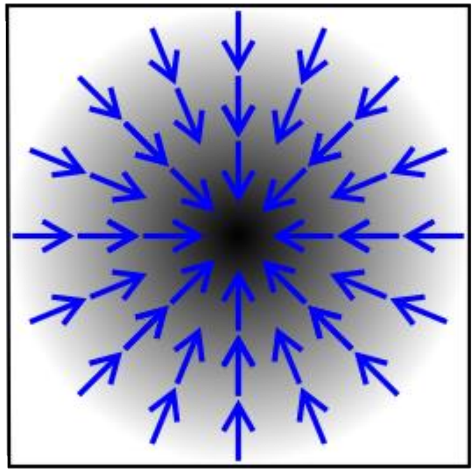
TITLE	CITED BY	YEAR
Distinctive image features from scale-invariant keypoints DG Lowe International journal of computer vision 60 (2), 91-110	75291	2004

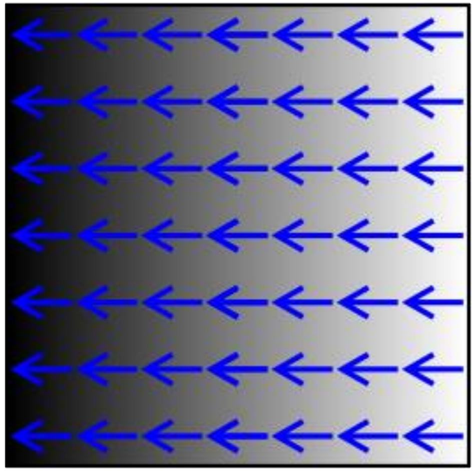
Motivation

- Scale invariant
  - Decompose the image into multiple scales and describe the key points at each scale

- Rotation invariant
  - Dominant orientation of the gradient directions

- Gradient
  - A vector
    - Direction: the one with the greatest rate of increase of the function value
    - Magnitude: how fast the function value increases



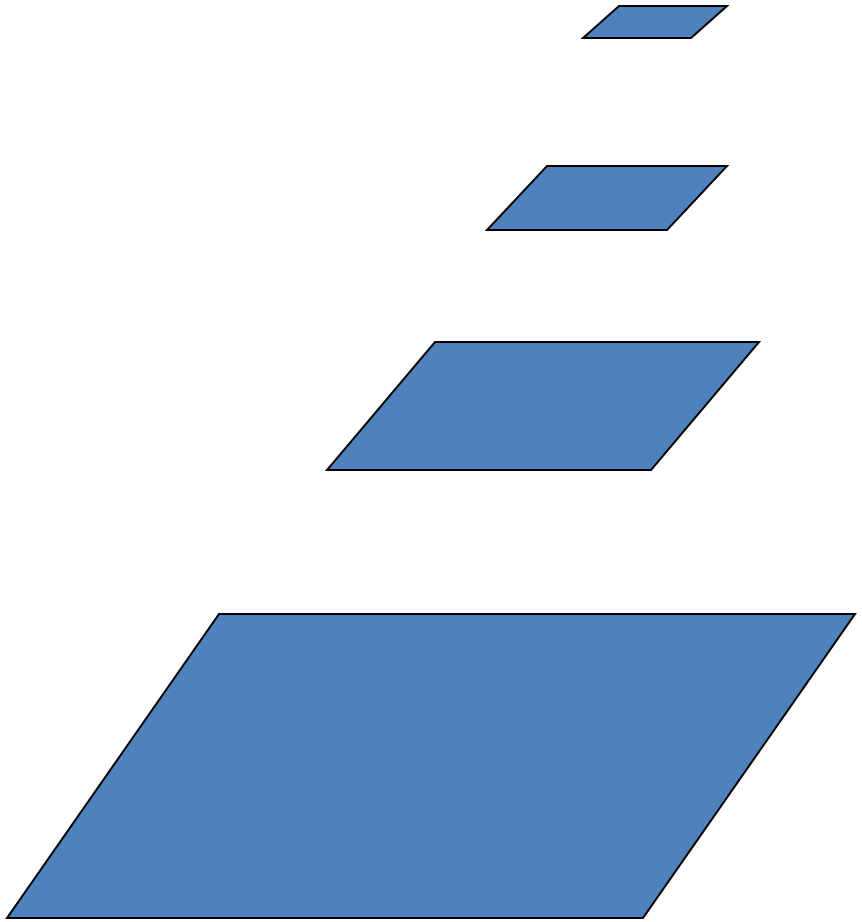


SIFT

- Overall procedure at a high level
  1. Scale-space extrema detection
     - Goal: Identify potential key points invariant to scale and orientation
     - Method: Search over all scales and image locations
  2. Key point localization
  3. Orientation assignment
  4. Key point description

SIFT

- Image Pyramids



And so on.

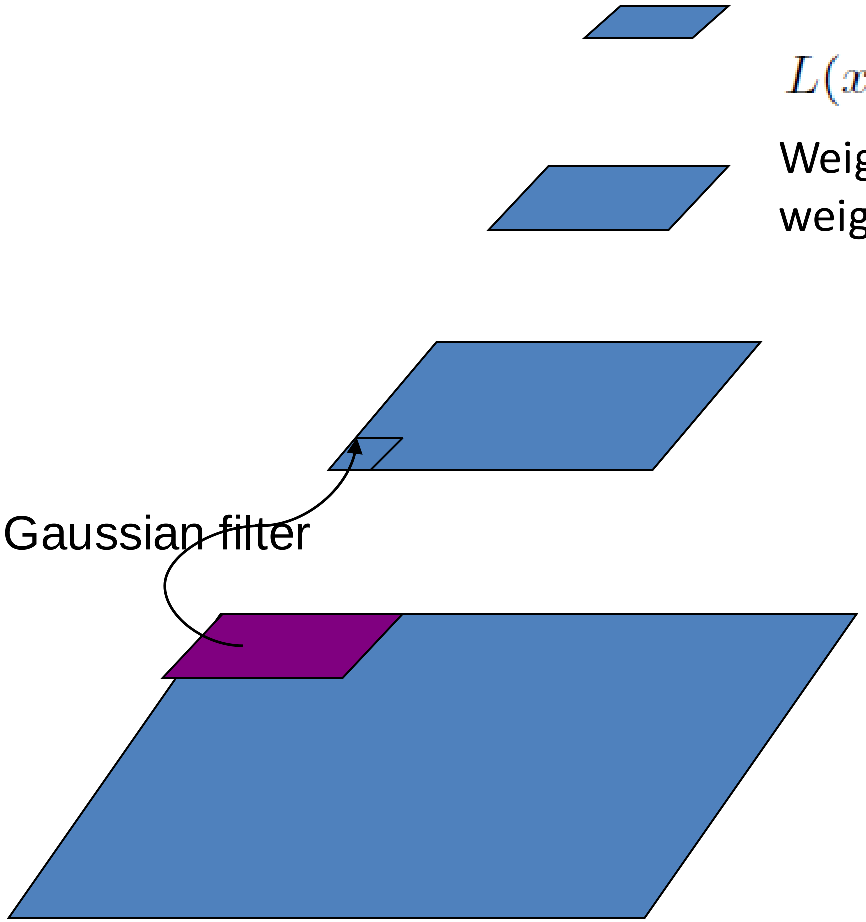
3rd level is derived from the 2nd level according to the same function

2nd level is derived from the original image according to some function

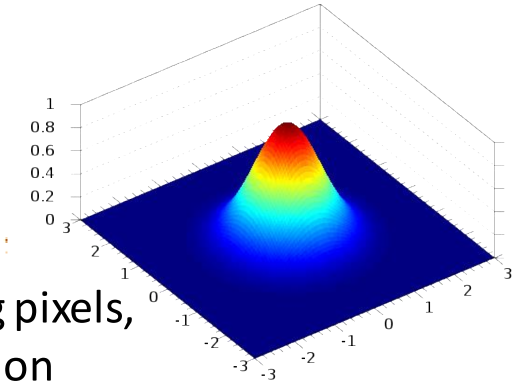
Bottom level is the original image.

SIFT

- Image Pyramids (Gaussian)



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



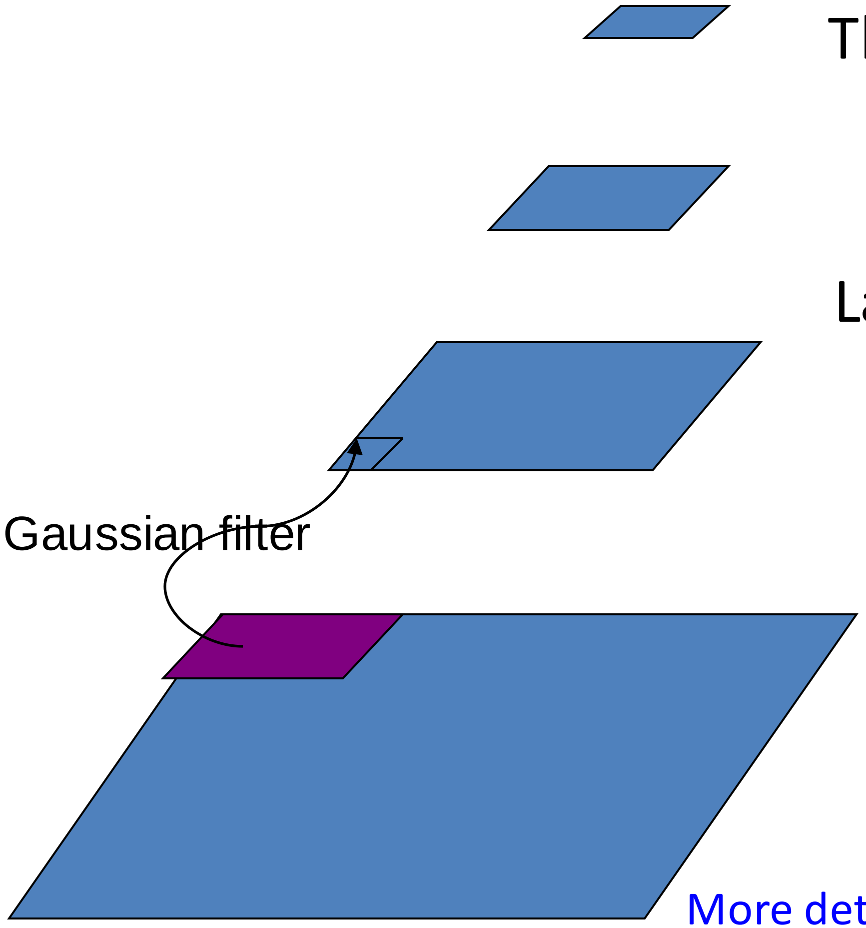
Weighted average of its neighboring pixels, weights specified by Gaussian function

At 2nd level, each pixel is the result of applying a Gaussian filter to the first level and then subsampling to reduce the size.

Bottom level is the original image.

SIFT

- Scale-space extrema detection

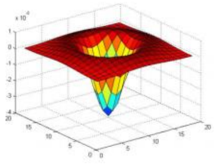


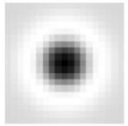
The scale space of an image

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

Laplacian of Gaussian (LoG)

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





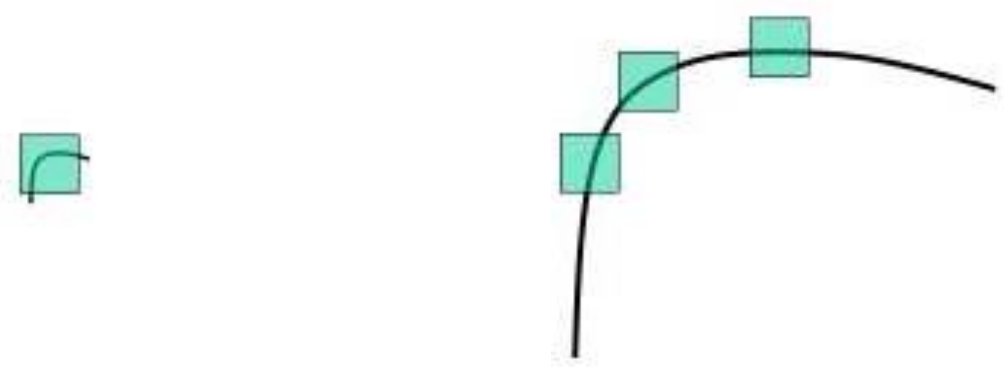
Compute a set of values at different scales

Find local maxima across scales (x, y, σ)

More details: https://en.wikipedia.org/wiki/Laplace_operator

SIFT

- Scale-space extrema detection
  - Local maxima across scale
  - Potential key point at (x, y) at scale σ



LoG (low sigma, **small corner**) > LoG (low sigma, large corner)
LoG (high sigma, small corner) < LoG (high sigma, **large corner**)

SIFT

- Overall procedure at a high level
  1. Scale-space extrema detection
  2. Key point localization
     - Motivation: potential key points have errors and outliers
     - Goal: refine to obtain the distinctive key points
  3. Orientation assignment
  4. Key point description

SIFT

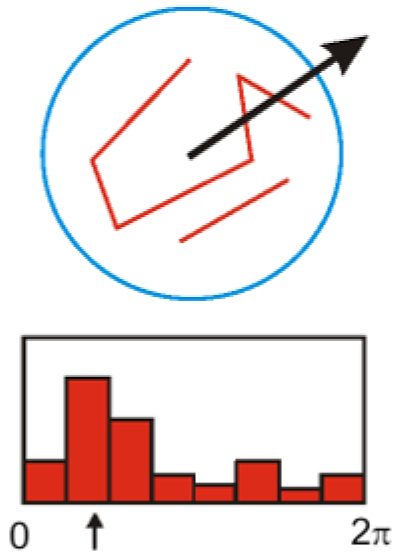
- Key point localization
  – A refinement step
    - Potential key points -> Distinctive key points
  – Eliminate low–contrast key points and edge key points
    - Comparing to a threshold (0.03 in the original paper)
    - Reject if smaller than the threshold
  – What remain are strong interest points
    - i.e., keypoints with higher confidence

SIFT

- Overall procedure at a high level
  1. Scale-space extrema detection
  2. Key point localization
  3. Orientation assignment
  4. Key point description

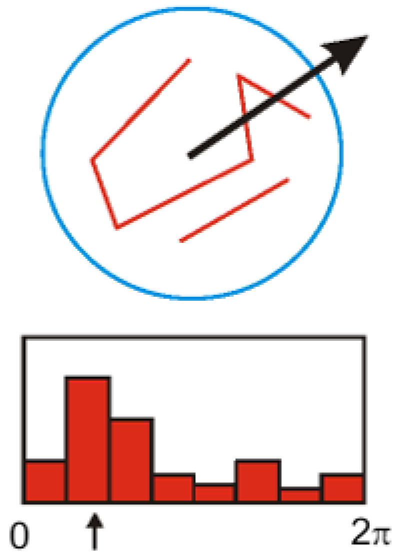
SIFT

- Orientation assignment
  - Assign one orientation to each key point based on local image gradient directions
    - Create a histogram of local gradient directions at the selected scale
    - Assign orientation at the peak of the smoothed histogram
    - Each key specifies (x, y, scale, orientation)



SIFT

- Orientation assignment
  - Assign one orientation to each key point based on local image gradient directions
    - Create a histogram of local gradient directions at the selected scale
    - Assign orientation at the peak of the smoothed histogram
    - Each key specifies (x, y, scale, orientation)
  - Rotation independence
    - Subtract key point rotation from each orientation

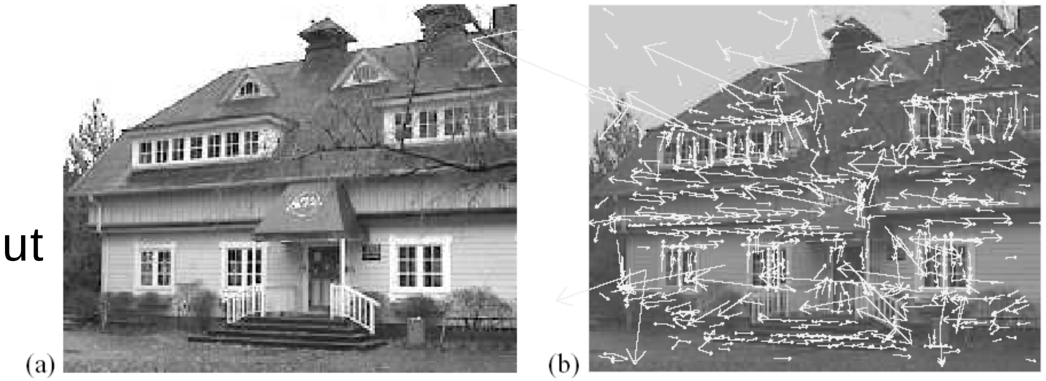


SIFT

- Orientation assignment

Key points are displayed as vectors indicating scale, orientation, and location

233 × 189 input



832 initial key points

Detected key points and their orientations

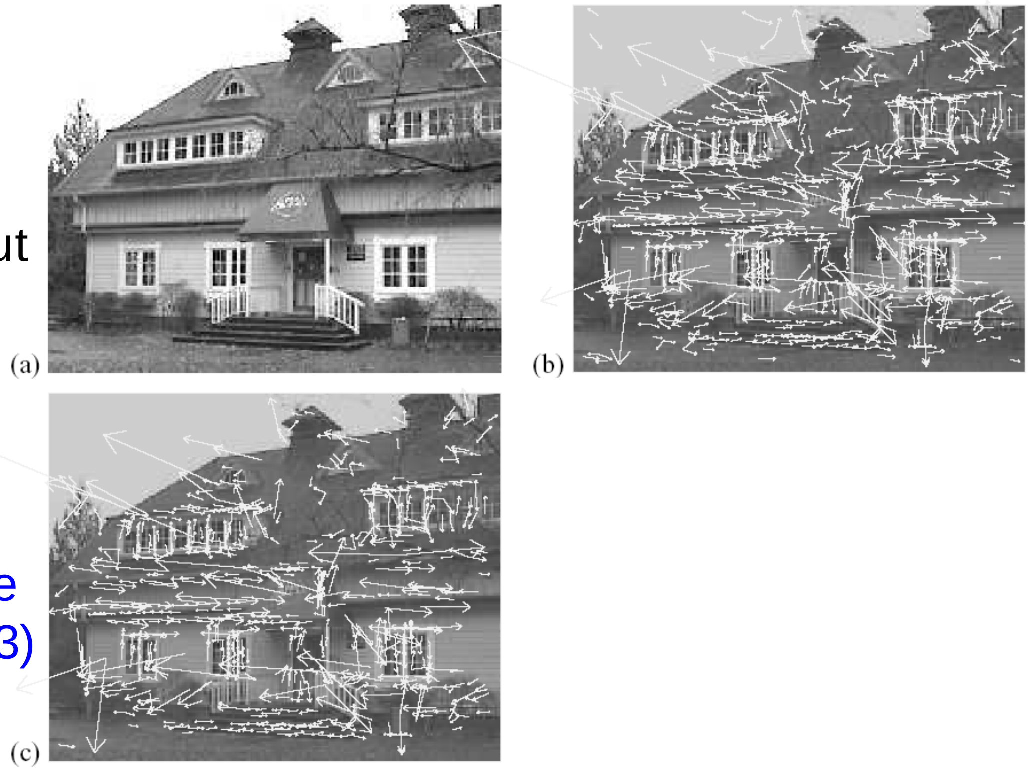
SIFT

- Stages of key point selection

Key points are displayed as vectors indicating scale, orientation, and location

233 × 189 input

832 initial key points



729 key points after the contrast threshold (0.03)

filtered out less distinctive key points

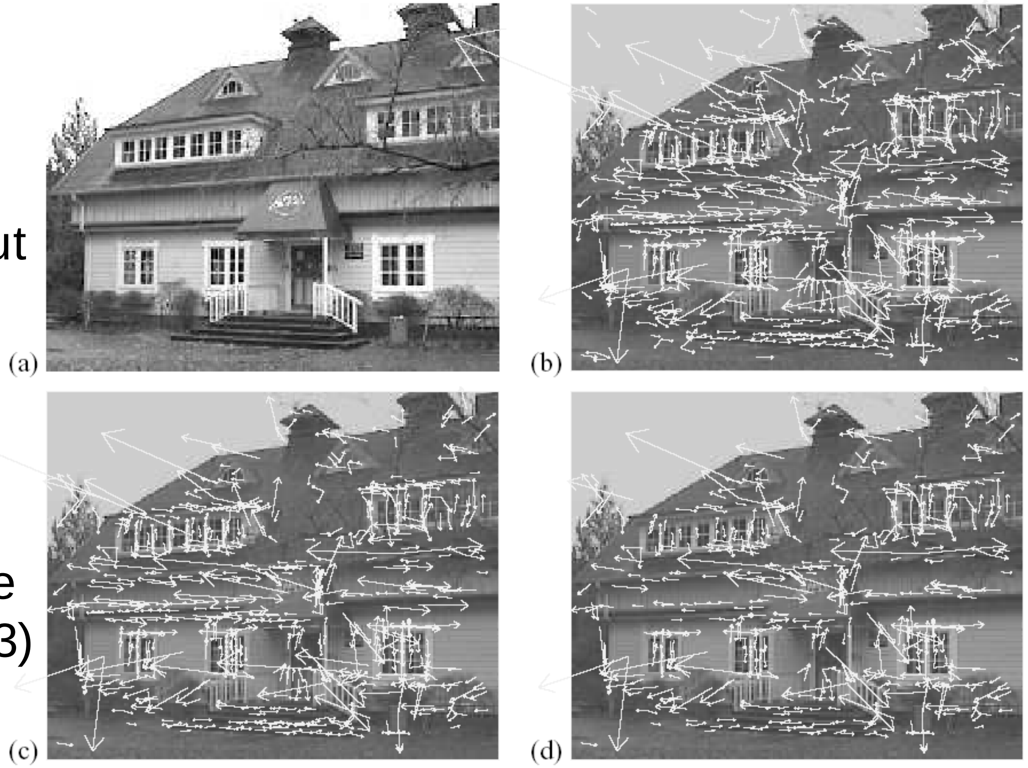
SIFT

- Stages of key point selection

Key points are displayed as vectors indicating scale, orientation, and location

233 × 189 input

832 initial key points



729 key points after the contrast threshold (0.03)

547 key points after the ratio threshold (75%)

SIFT

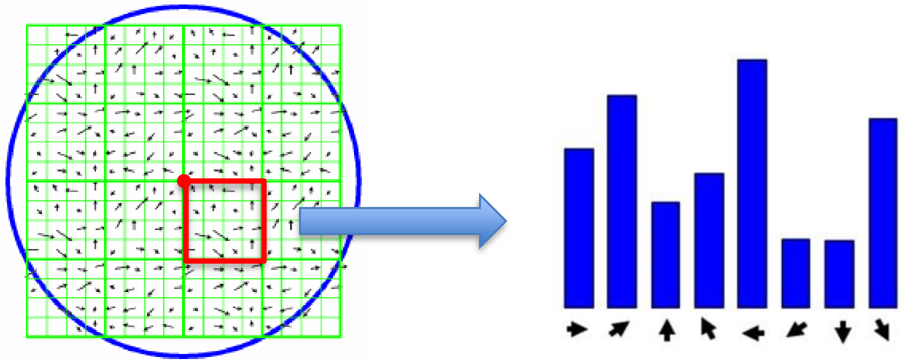
- Overall procedure at a high level
  1. Scale-space extrema detection
  2. Key point localization
  3. Orientation assignment
  4. Key point description

SIFT

- Key point description
  - Location (x, y)
  - Scale
  - Orientation
- Key point descriptor
  - Highly distinctive
  - Invariant to variations such as changes in viewpoint, distortion, illumination, etc.

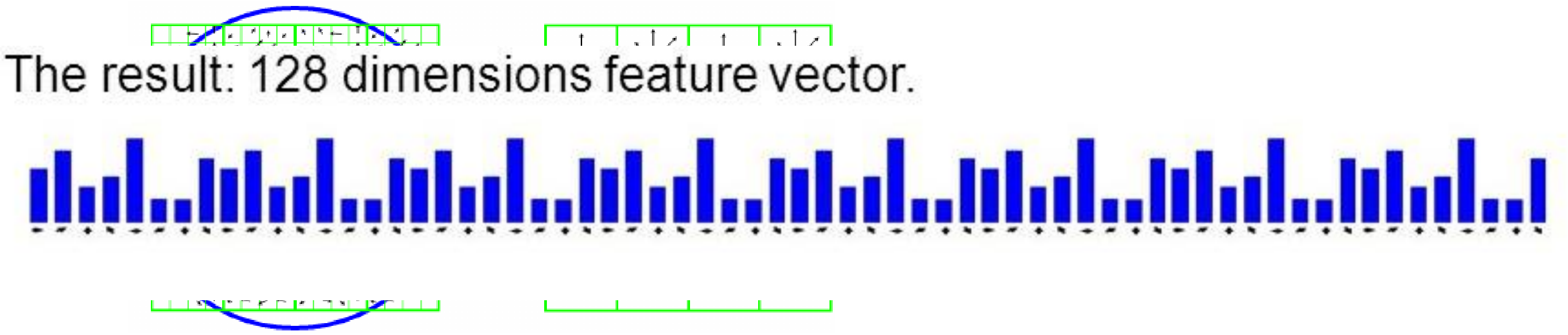
SIFT

- Key point description
  - – 16 x 16 neighborhood of the key point
  - – Divided into 16 sub-blocks of 4x4 size
  - – For each sub-block, create 8 bin orientation histogram
    - Value: sum of gradient magnitude at each direction



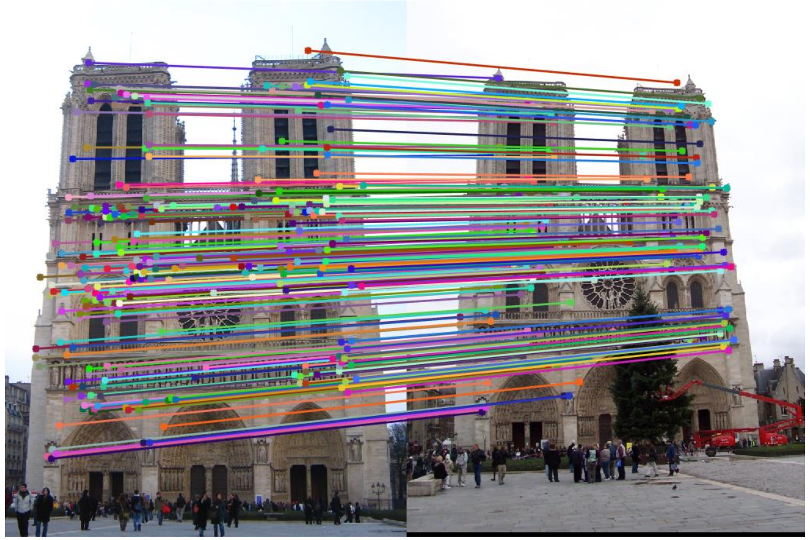
SIFT

- Key point description
  - 16 x 16 neighborhood of the key point
  - Divided into 16 sub-blocks of 4x4 size
  - For each sub-block, create 8 bin orientation histogram
  - Concatenate histograms of 16 sub-blocks



Feature Matching

- Matching key points
  - – Ideal case: find the nearest neighbor
  - – Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first



Feature Matching

- Matching key points
  - Ideal case: find the nearest neighbor
  - Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first

$$\text{Reject if } \frac{\text{closest−distance}}{\text{second−closest distance}} > 0.8$$

– Can eliminate about 90% of false matches while discards only 5% correct matches

Feature Matching

- Matching key points
  - Ideal case: find the nearest neighbor
  - Practice
    - Real-world images are very noisy
    - Second closest-match can be very near to the first

$$\text{Reject if } \frac{\text{closest−distance}}{\text{second−closest distance}} > 0.8$$

  - Can eliminate about 90% of false matches while discards only 5% correct matches
  - Sophisticate strategies
    - RANSAC

SIFT

- Characteristics of SIFT
  - **Locality**: features are local, so robust to occlusion and clutter
  - **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

SIFT

- Applications
  - Image stitching
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - …

For more details of the SIFT algorithm:

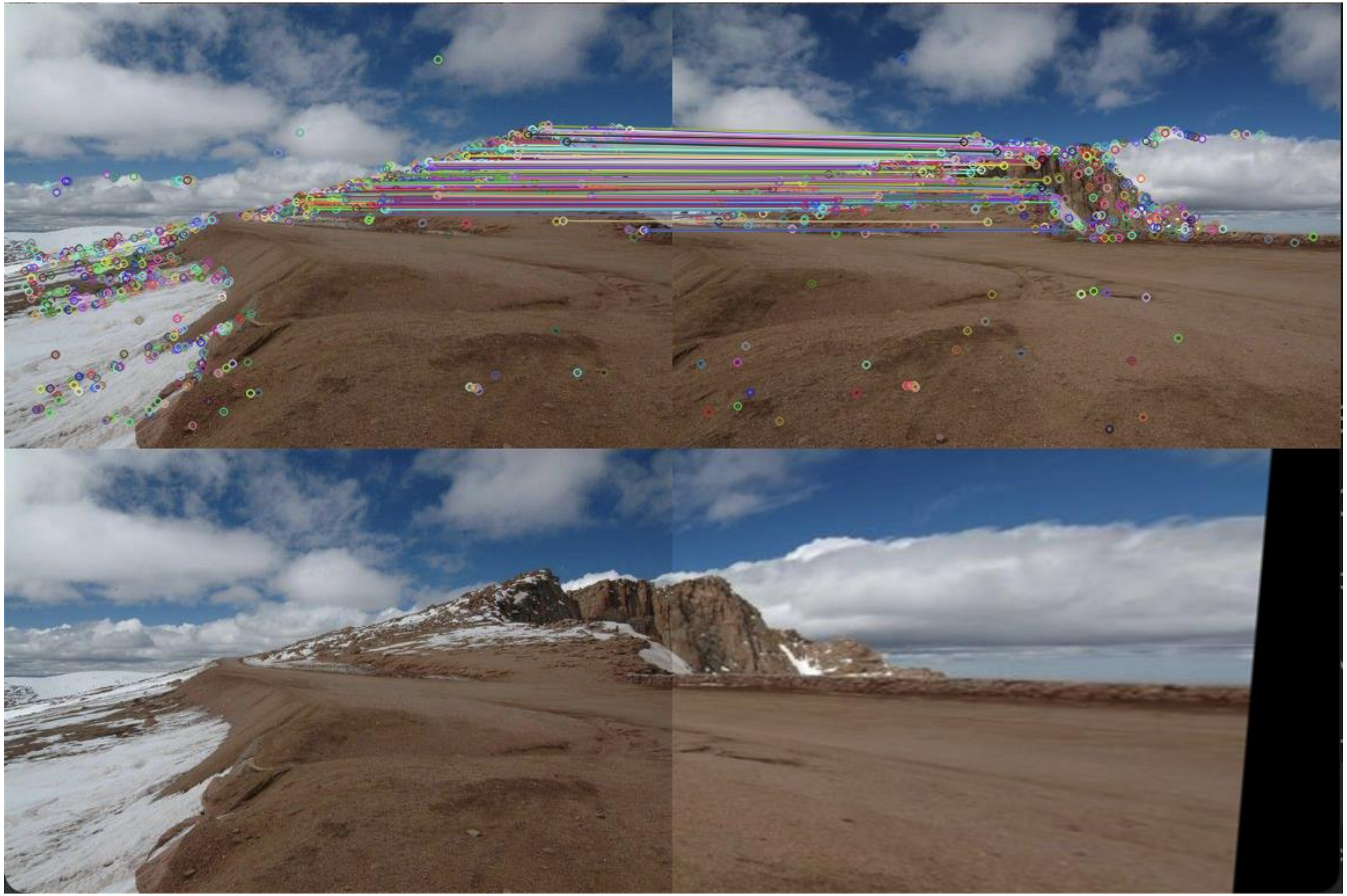
Distinctive image features from scale-invariant keypoints
D. Lowe. International journal of computer vision 60 (2), 91-110, 2004

Extensions or similar features

- SIFT (**S**cale-**I**nvariant **F**eature **T**ransform)
- SURF (**S**peeded-**U**p **R**obust **F**eatures)
- FAST (**F**eatures from **A**ccelerated **S**egment **T**est)
- BRIEF (**B**inary **R**obust **I**ndependent **E**lementary **F**eatures)
- ORB (**O**riented FAST and **R**otated **BRIEF**)
- …

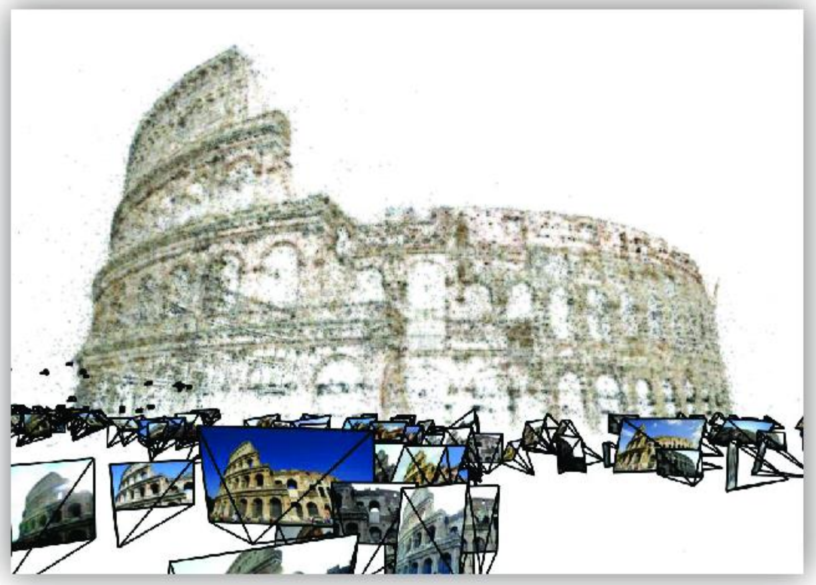
Lab: Image matching (code available)

Lab: Image matching (code available)



Next lecture

- Multi-view Stereo
  - Obtaining dense point clouds



Images + camera information

Dense 3d point cloud