SIFT: Scale-Invariant Feature Transform

Matthew Toews

ECSE 626

February 9, 2007

- Distinctive image features from scale-invariant keypoints
  David G. Lowe, IJCV, 60, 2 (2004), pp. 91-110.

2-Lecture Overview

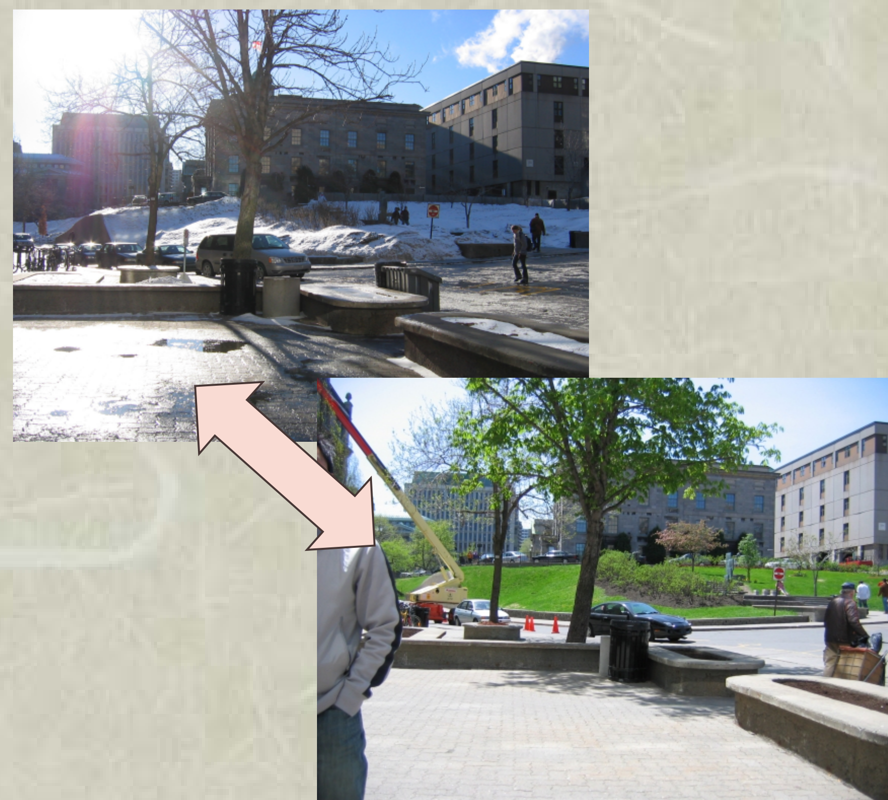
1) Image Features (February 9, 2007)
   - Corners, edges, SIFT.
   - **Single images**, matching.
2) Probabilistic Modeling (February 12, 2007)
   - Learning how features behave over **many images**.

Overview

- Image Matching via Local Features
- Scale-Invariant Feature Transform: SIFT
  - Detection
  - Description
  - Matching
- Applications & Examples

Image Matching

- Determine correspondence, or a mapping, between different images.



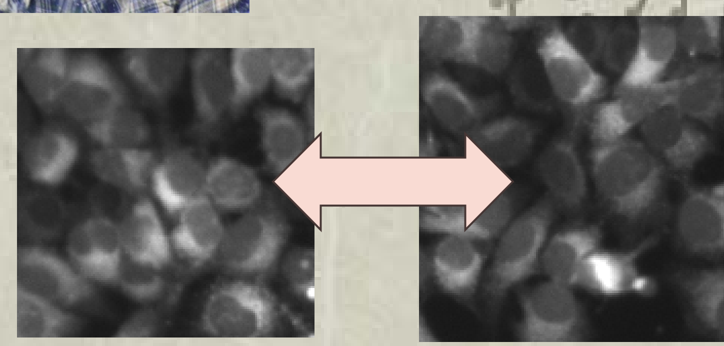


Image Matching – Difficulties

One-to-one?

- Defining a geometric mapping!

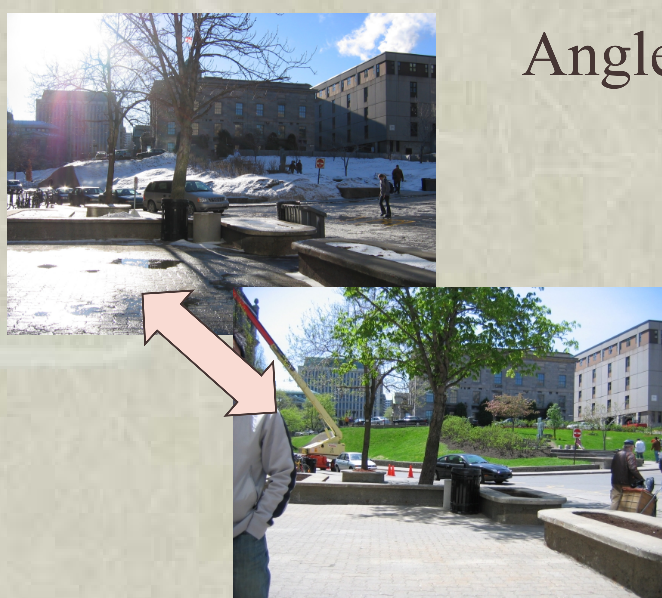
Continuous?

Many-to-one?

Diffeomorphic?

Bijective?

Angle-preserving?



Invertible?

Discrete?

Differentiable?

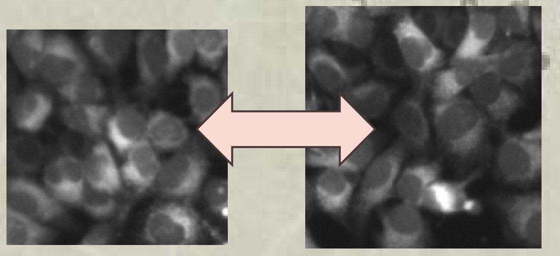


Image Matching – Difficulties

- Illumination change
- Geometrical deformation
- Viewpoint change
- Object/scene shape change
- Occlusion
- Ill-posedness: multiple solutions, no solutions

Image Matching via Local Features

- Mapping informative, discrete features between images.

1) Feature Detection

2) Feature Matching

Image Matching via Local Features

- Difficulties
  - Defining what sorts of features to detect.
  - Reliably detecting the same features in different images: *repeatability*.
  - Reliably matching the same features in different images.

Image Matching via Local Features

- Advantages
  - Robust:
    - Partial matching in the presence of occlusion.
  - Efficient:
    - No need to process entire images, just small windows.
    - Matching in the presence of image to translation, rotation, scale, lighting change.

Local Features in Vision: History

- 1970s:
  - Moravec: interest points.
- 1980s:
  - Harris: corner detectors.
  - Canny: edge detection.
- 1990s:
  - Shi: edge density.
  - Lindeberg: scale-space theory.
- 1990-2000s:
  - Lowe, Schmid, Carneiro, Kadir, efficient, robust scale-invariant feature detectors.

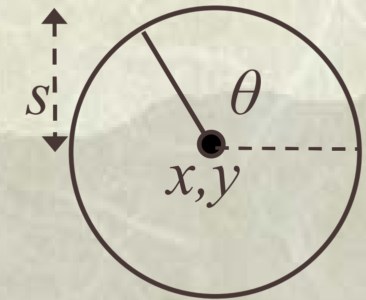
SIFT Features

- **SIFT**: Scale-Invariant Feature Transform
- **Idea**: identify the same image features in the presence of
  - Geometrical deformation: Translation, rotation, scale change.
  - Intensity deformation: linear intensity variation.

SIFT Features

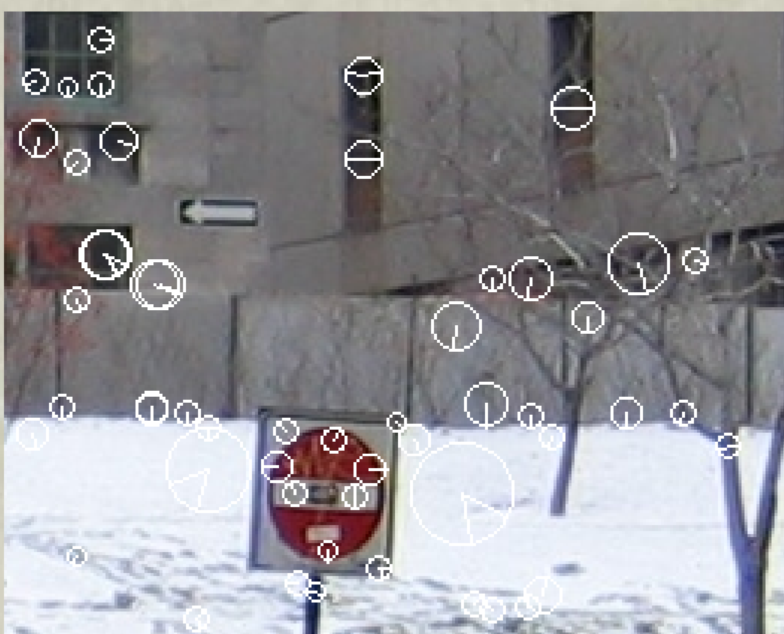
Geometry

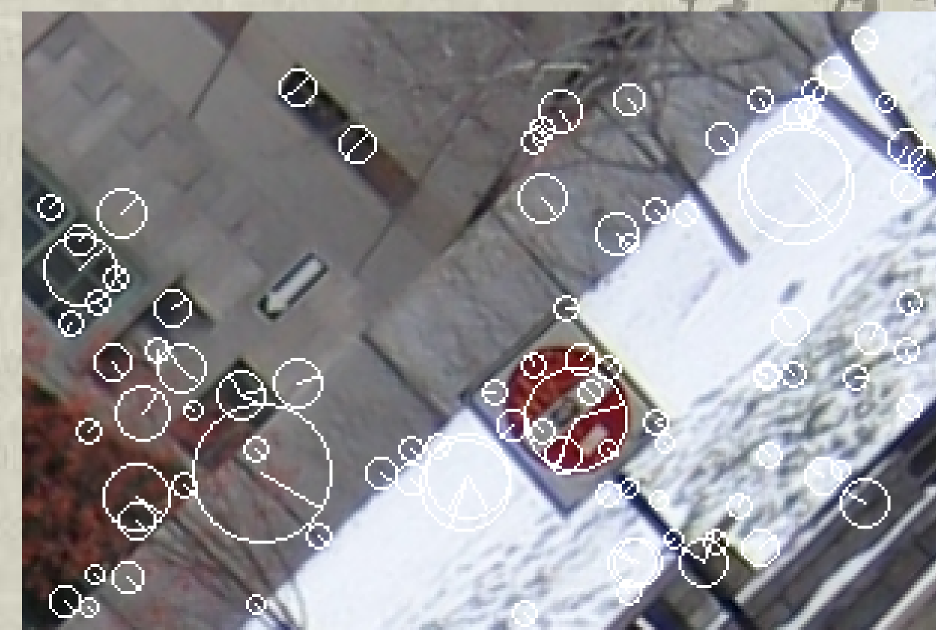
- Location x,y
- Orientation θ
- Scale s



Appearance

- Image intensity information
- I.e. Pixels, edges





SIFT Features

- Three Phases:
  1) Detection
  2) Description
  3) Matching

SIFT Feature Detection

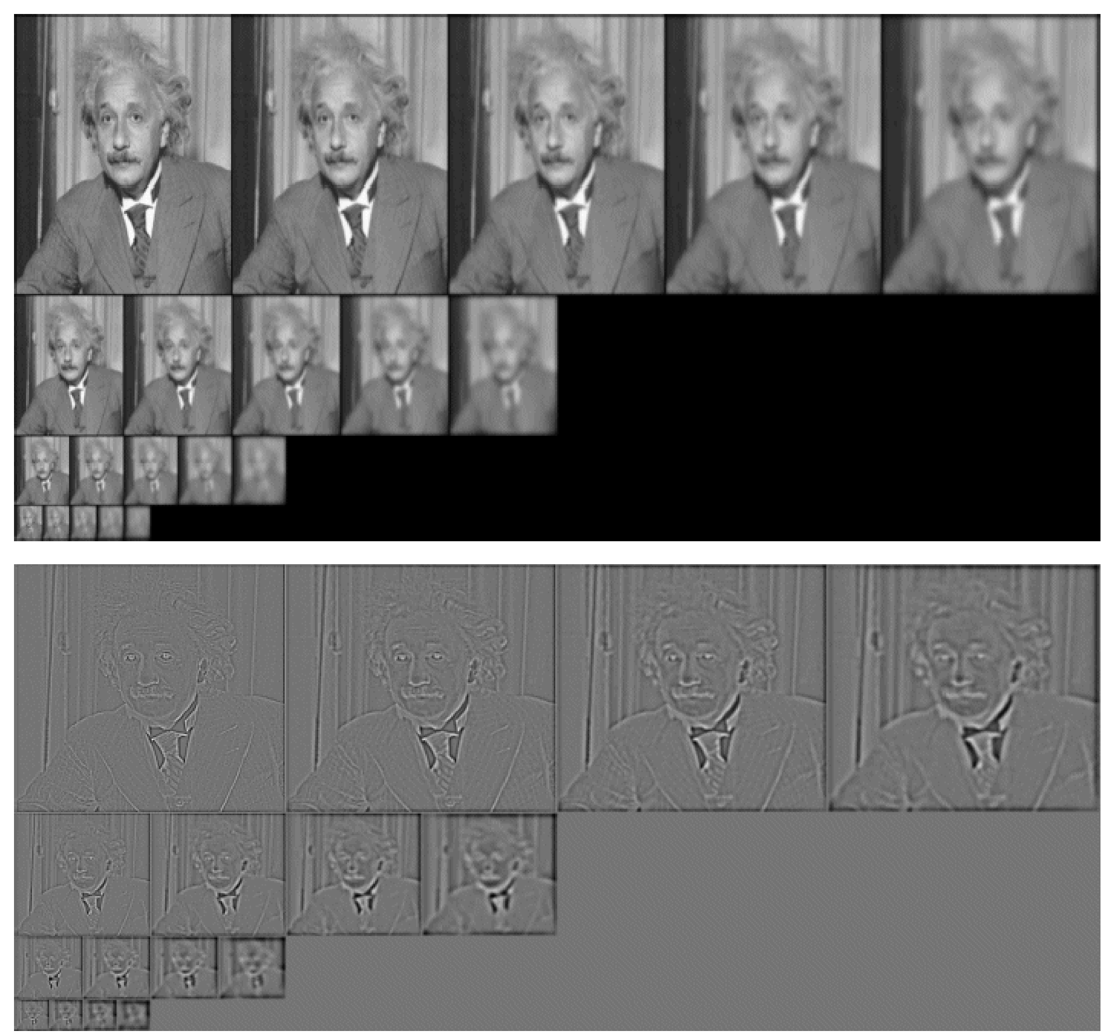
- **Purpose**: Automatically identify features in an image.

1) Create a Gaussian image scale space G(x,y,s).
2) Search for peaks in the derivative with respect to scale: dG(x,y,s)/ds.
3) Normalize features geometrically for matching.

Detection

Gaussian pyramid
G(x,y,s)

DOG pyramid
DOG(x,y,s)

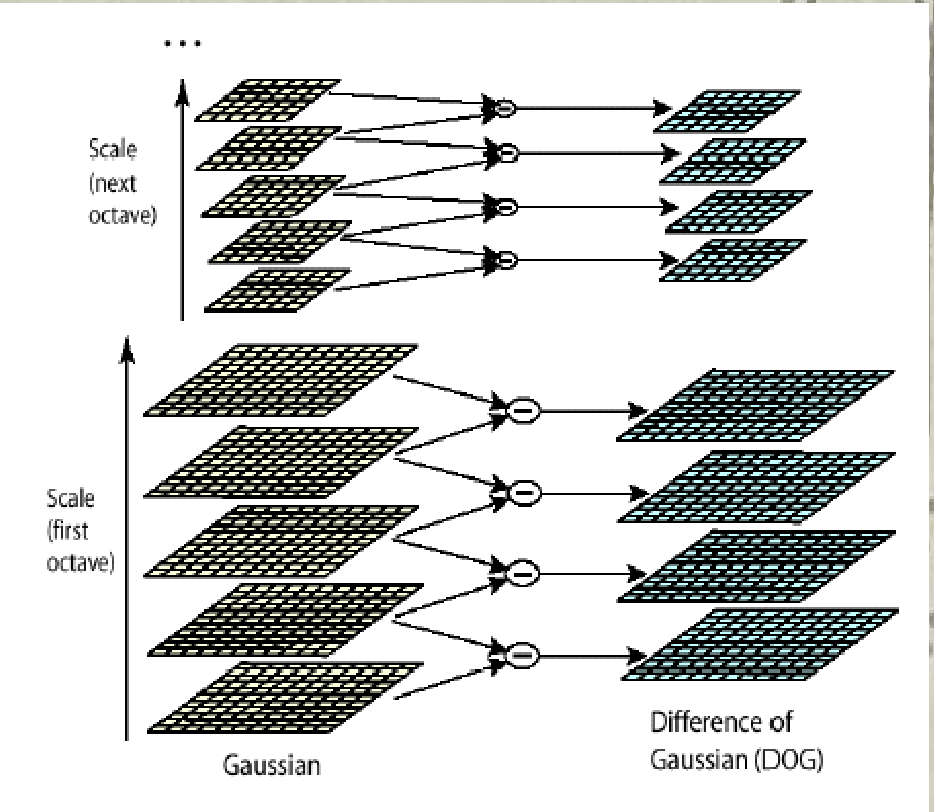


Detection: Why Gaussian?

- **Detailed answer**: scale-space theory.
  - Causality.
  - Non-creation of local extrema.
  - Semi-group structure.
- **Simple answer**: a scaled image should 'look' the same as the original.
  - $G(x,y, s_1+s_2) = G(x,y,s_1)*G(x,y,s_2)$

Detection: 1

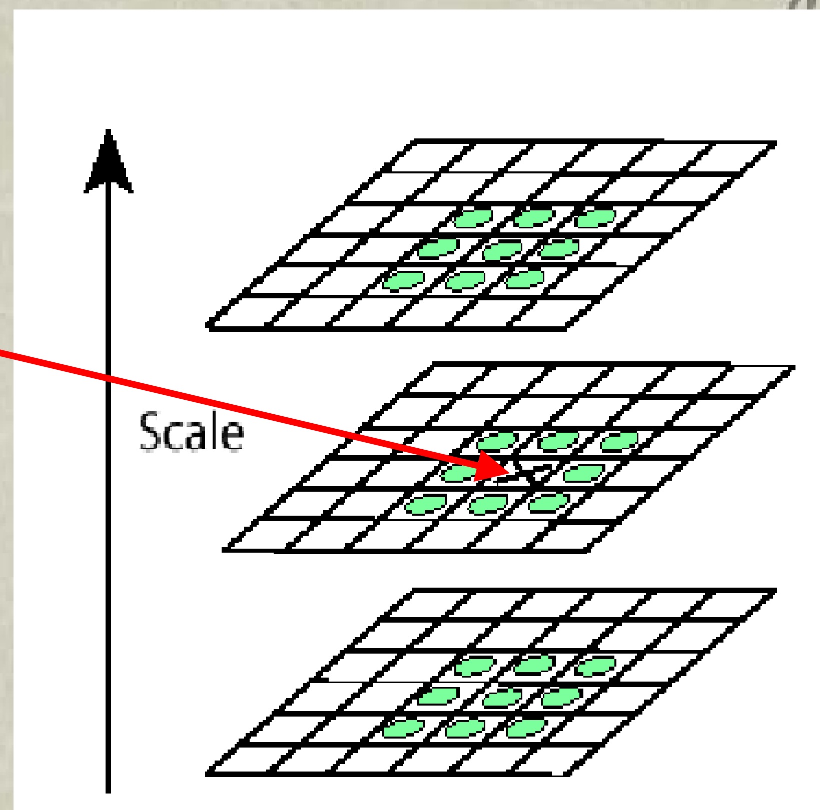
- Difference-of-Gaussian (DOG) Generation.



Detection: 2

- Extrema detection.

Max or min DOG(x,y,s)



Detection: 3

- Geometrical normalization. (for matching)

a) Normalize features according to scale: *scale invariance*.

b) Calculate dominant image orientations from image gradients.

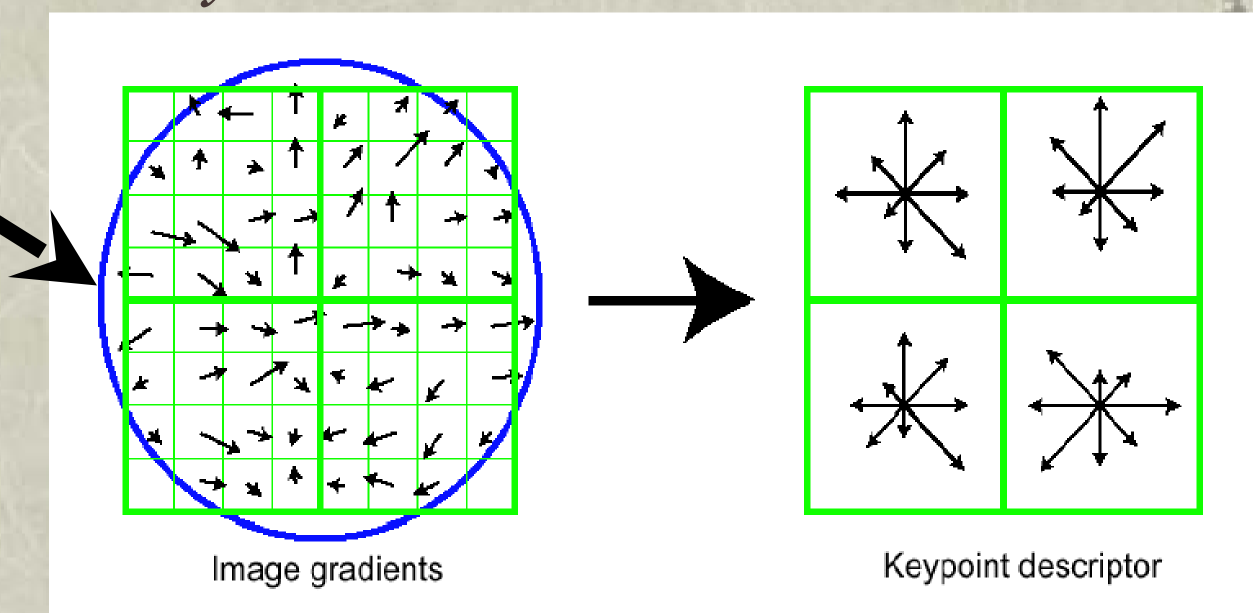
c) Normalize features according to orientation: *orientation invariance*.

SIFT Feature Description

- **Purpose**: Encode feature image content for feature matching.
  - Maximize feature distinctivness.
- **Many possibilities:**
  - Descriptions: image pixels, principle components…
  - Similarity measures: squared pixel differences, correlation, mutual information…

Description

- Encode using localized image gradient histograms.
- Normalize histogram bin magnitudes: *intensity invariance*.



SIFT Feature Matching

- **Purpose**: correctly match features in different images.
- **Step 1**: Nearest neighbour descriptor matching, distance thresholding.
- **Step 2**: Match validation via geometric consistency (Hough transform).

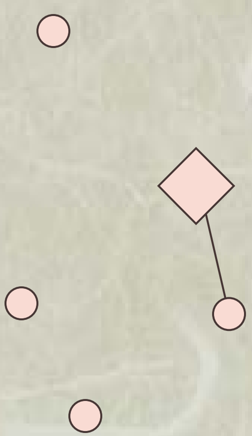
Matching

- Nearest neighbour descriptor matching.
- Euclidian distance measure.
  - Equivalent to normalized cross covariance for normalized descriptors.
  - Euclidean distance implies independent, identically distributed descriptor elements.

Match Distance Threshold

- **Purpose**: to discard false matches.

Training



Thres: nearest neighbor distance in set of unrelated features.

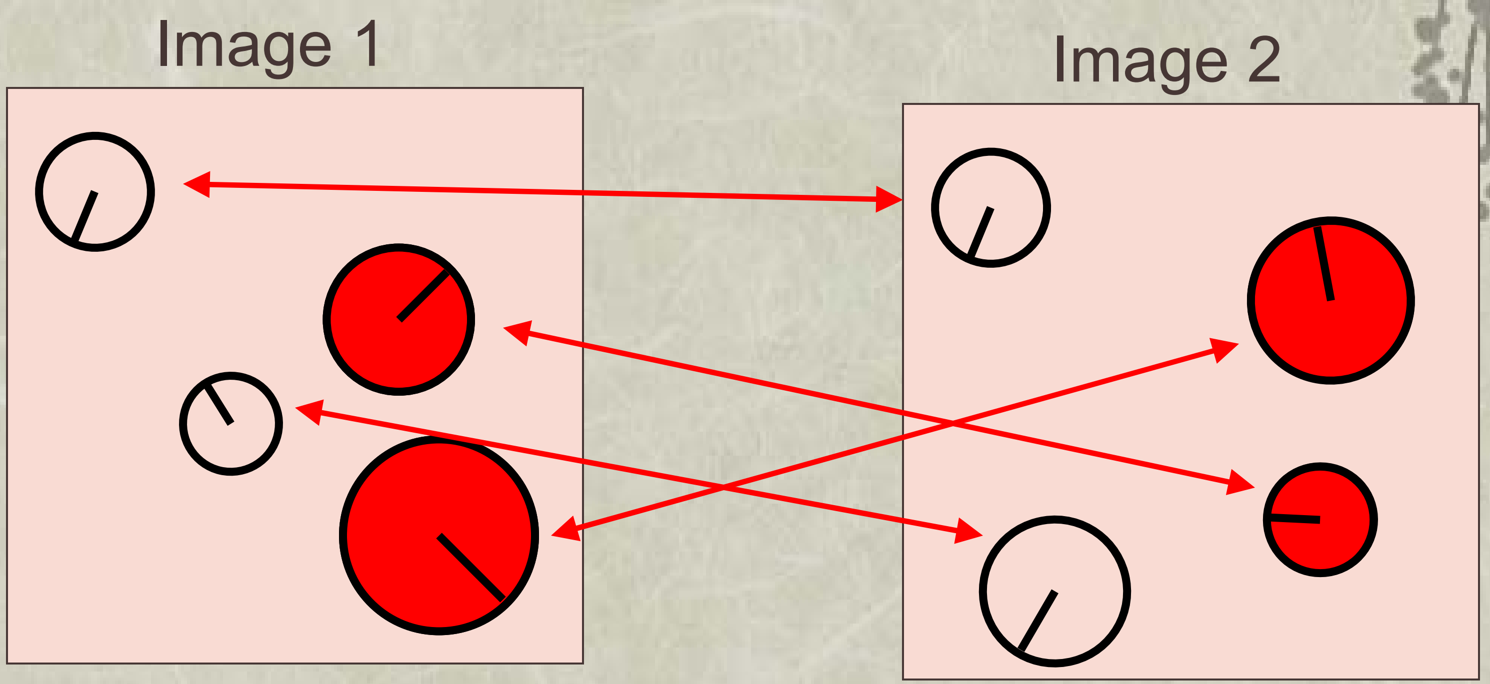
Matching

Dist: nearest neighbor distance in image of interest.

Rule: discard match if $\varepsilon Thres < Dist$

Match Geometrical Consistency

- **Purpose**: to validate true matches.
  - Hough transform, a voting technique.
  - Consider matches that agree geometrically.



Some Results

Some Results

It's Been Done

- Many scale-space feature detectors now exist.
  - Based on image blobs, edges, entropy, phase, color…
- Fast matching methods for database retrieval, view-based object recognition.
  - KD-tree data structure, O(log N) complexity.
  - 100,000s of object images.

Current Applications

- Automatic localization from cell phone camera images.
- Automated grocery checkout: cereal boxes, etc.
- 3D scene reconstruction, wide-baseline stereo.
- Probabilistic object appearance modeling.

Current Applications

- Used by AIBO to find his food supply!

- Recognition

Self-Charging

When the battery level gets lower than 40%,
AIBO starts self-charging. Using the black
and white pattern on the energy station pole
and its visual pattern recognition, AIBO can
locate and use the charger by itself.

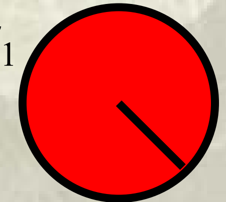
Future Work

- Dealing with large feature databases, ambiguity.
- Modeling abstract object class appearance.
  - i.e. faces, cars
  - locations
- Probabilistic appearance modeling.

Home Work

Knowing feature geometry g_1 and a transform matrix A, we can

g_1



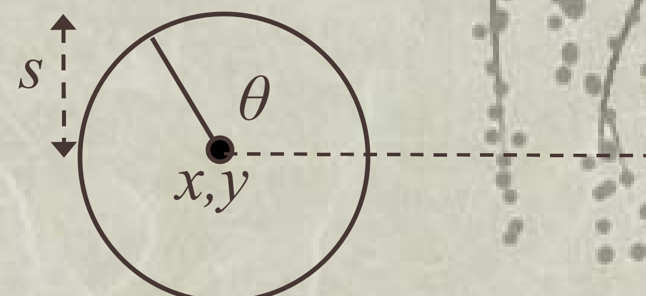
determine feature geometry g_2.

$g_2 = Ag_1$

What is A?

Hints:

$$\begin{bmatrix} \log s_2 \\ \theta_2 \\ x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} ? & 0 & 0 & 0 & ? \\ 0 & ? & 0 & 0 & ? \\ 0 & 0 & ? & ? & 0 \\ 0 & 0 & ? & ? & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \log s_1 \\ \theta_1 \\ x_1 \\ y_1 \\ 1 \end{bmatrix}$$



2-Lecture Overview

1) Image Features (February 9, 2007)
   - Corners, edges, SIFT.
   - **Single images**, matching.
2) Probabilistic Modeling (February 12, 2007)
   - Learning how features behave over **many images**.