CS294-43: Visual Object and Activity Recognition

Prof. Trevor Darrell

Jan 27th: Instance Recognition and Retrieval
Today

- SIFT
- Video Google
- Total Recall
- Photo Tourism
Correspondence

- Fundamental to many of the core vision problems
  - Recognition
  - Motion tracking
  - Multiview geometry
- Local features are the key


Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Local Features: Detectors vs. Descriptors

Detected Interest Points/Regions

Descriptors

<0 12 31 0 0 23 ...>

<5 0 0 11 37 15 ...>

<14 21 10 0 3 22 ...>

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Ideal Interest Points/Regions

- Lots of them
- Repeatable
- Representative orientation/scale
- Fast to extract and match

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Keypoint Localization

• Goals:
  - Repeatable detection
  - Precise localization
  - Interesting content

⇒ Look for two-dimensional signal changes

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Harris Detector [Harris88]

*Intuition:* Search for local neighborhoods where the image content has two main directions (eigenvectors).
Harris Detector [Harris88]

**Intuition:** Search for local neighborhoods where the image content has two main directions (eigenvectors).

1. Image derivatives
   \( g_x(\sigma_D) \), \( g_y(\sigma_D) \),

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
**Harris Detector** [Harris88]

*Intuition:* Search for local neighborhoods where the image content has two main directions (eigenvectors).

1. Image derivatives
   \( g_x(\sigma_D), \ g_y(\sigma_D), \)

2. Square of derivatives

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Harris Detector [Harris88]

Second moment matrix (autocorrelation matrix):

\[
\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix}
I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\
I_x I_y(\sigma_D) & I_y^2(\sigma_D)
\end{bmatrix}
\]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma_I) \)

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Harris Detector [Harris88]

Second moment matrix (autocorrelation matrix):

\[
\mu(\sigma_I, \sigma_D) = g(\sigma_I)^* \begin{bmatrix}
I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\
I_x I_y(\sigma_D) & I_y^2(\sigma_D)
\end{bmatrix}
\]

1. Image derivatives

2. Square of derivatives

3. Gaussian filter \( g(\sigma_I) \)

4. Cornerness function - both eigenvalues are strong

\[
har = \text{det}[\mu(\sigma_I, \sigma_D)] - \alpha[\text{trace}(\mu(\sigma_I, \sigma_D))] =
\]

\[
g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2
\]

5. Non-maxima suppression

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Harris Detector – Responses [Harris88]

Effect: A very precise corner detector.

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Harris Detector – Responses [Harris88]
Automatic Scale Selection

\[ f(\mathcal{I}_{i_1...i_m}(x, \sigma)) = f(\mathcal{I}_{i_1...i_m}(x', \sigma')) \]

Same operator responses if the patch contains the same image up to scale factor

How to find corresponding patch sizes?

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

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

\[
f(I_{i_1...i_m}(x', \sigma))
\]

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

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

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

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

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

\[ f(I_{i_1...i_m}(x', \sigma)) \]

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Laplacian-of-Gaussian (LoG) scale detection

- Laplacian also measures bandpass contrast...
- which ‘scale’ has most ‘contrast’?

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]

\( \Rightarrow \) List of \((x, y, s)\)

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Results: Laplacian-of-Gaussian

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Difference-of-Gaussian (DoG)

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

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
DoG - Efficient Computation

- Computation in Gaussian scale pyramid

Sampling with step $\sigma^4 = 2$

Original image

$\sigma = \frac{1}{2^4}$

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Results: Lowe’s DoG

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Finding Keypoints – Scale, Location

- **Harris-Laplacian**¹
  *Find local maximum of:*
  - Laplacian in scale
  - Harris corner detector in space (image coordinates)

- **SIFT**²
  *Find local maximum of:*
  - Difference of Gaussians in space and scale


Slide credit David Lee / David Lowe
Finding Keypoints – Orientation

- 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)

Slide credit David Lee / David Lowe
Finding Keypoints – Orientation

- Assign dominant orientation as the orientation of the keypoint
SIFT Descriptor

- 4x4 Gradient window
- Histogram of 4x4 samples per window in 8 directions
- Gaussian weighting around center (\( \sigma \) is 0.5 times that of the scale of a keypoint)
- 4x4x8 = 128 dimensional feature vector
SIFT Descriptor – Lighting changes

- Gains do not affect gradients
- Normalization to unit length removes contrast
- Saturation affects magnitudes much more than orientation
- Threshold gradient magnitudes to 0.2 and renormalize
Performance

- Very robust
  - 80% Repeatability at:
    - 10% image noise
    - 45° viewing angle
    - 1k-100k keypoints in database

- Best descriptor in [Mikolajczyk & Schmid 2005]’s extensive survey

- 606+ citations on Google Scholar already for [2004] paper

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Typical Usage

- For set of database images:
  1. Compute SIFT features
  2. Save descriptors to database

- For query image:
  1. Compute SIFT features
  2. For each descriptor:
     • Find closest descriptors (L2 distance) in database
  3. Verify matches
     • Geometry
     • Hough transform

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Nearest-neighbor matching to feature database

- Hypotheses are generated by approximate nearest neighbor matching of each feature to vectors in the database
  - SIFT use best-bin-first (Beis & Lowe, 97) modification to k-d tree algorithm
  - Use heap data structure to identify bins in order by their distance from query point

- Result: Can give speedup by factor of 1000 while finding nearest neighbor (of interest) 95% of the time

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
3D Object Recognition

- Only 3 keys are needed for recognition, so extra keys provide robustness

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Recognition under occlusion

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Test of illumination Robustness

- Same image under differing illumination

273 keys verified in final match

Slide credit: O. Pele, S. Thrun, J. Košlecká, N. Kumar
Location recognition

Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Image Registration Results

[Brown & Lowe 2003]
Slide credit: O. Pele, S. Thrun, J. Košecká, N. Kumar
Local Descriptors: SURF

- Fast approximation of SIFT idea
  - Efficient computation by 2D box filters & integral images
    - 6 times faster than SIFT
  - Equivalent quality for object identification

- GPU implementation available
  - Feature extraction @ 100Hz
    (detector + descriptor, 640×480 img)
  - [http://www.vision.ee.ethz.ch/~surf](http://www.vision.ee.ethz.ch/~surf)

[Bay, ECCV’06], [Cornelis, CVGPU’08]

Slide credit K. Grauman, B. Leibe AAAI08 Short Course
Scaling up: particular object retrieval

Example 1: Visual search in feature films

Visually defined query

“Find this clock”

“Find this place”

“Groundhog Day” [Rammis, 1993]

Slide credit: J. Sivic
Example II: Search photos on the web for particular places

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

Slide credit: J. Sivic
Why is it difficult?

Want to find the object despite possibly large changes in scale, viewpoint, lighting and partial occlusion.

Slide credit: J. Sivic
The need for large-scale visual search

Flickr: has 2 billion photographs, more than 1 million added daily

Company collections

Personal collections: 10000s of digital camera photos and mpegs

Vast majority will have minimal, if any, textual annotation. Yet text is the only common way of searching / accessing documents (e.g. Google / Live search)

Slide credit: J. Sivic
Recap: Image representation

- Image content is transformed into local features that are invariant to geometric and photometric transformations.
Affine covariant regions (Feb 10\textsuperscript{th})

- a region’s size and shape are not fixed, but
- automatically adapts to the image intensity to cover the same physical surface
- i.e. pre-image is the same surface region

Represent each region by the 128-dimensional SIFT descriptor vector

Slide credit: J. Sivic
Example

Slide credit: J. Sivic
Object recognition

Establish correspondences between object model image and target image by nearest neighbour matching on SIFT vectors

Slide credit: J. Sivic
Problem with matching on descriptors alone

• too much individual invariance
• each region can affine deform independently (by different amounts)
• use semi-local and global spatial relations to verify matches, e.g.:
  • common affine transformation [Lowe ‘99] (strong requirement)
  • locally similar affine transformation [Ferrari ‘04]
  • spatial neighbours match spatial neighbours [Schmid ‘97]
Example

Initial matches

Spatial consistency required

Slide credit: J. Sivic
Example of object recognition

1000+ descriptors per frame

- Shape adapted regions
- Maximally stable regions

Slide credit: J. Sivic
Match regions between frames using SIFT descriptors and spatial consistency

Multiple regions overcome problem of partial occlusion

- Shape adapted regions
- Maximally stable regions

Slide credit: J. Sivic
Visual search using local regions

Schmid and Mohr ’97 – 1k images
Sivic and Zisserman’03 – 5k images
Nister and Stewenius’06 – 50k images (1M)
Philbin et al.’07 – 100k images
Chum et al.’07 + Jegou and Schmid’07 – 1M images
Chum et al.’08 – 5M images

Index 1 billion (10^9) images
  – 200 servers each indexing 5M images?

Slide credit: J. Sivic
Outline of a retrieval strategy

1. Compute affine covariant regions in each frame independently
2. “Label” each region by a vector of descriptors based on its intensity
3. Finding corresponding regions is transformed to finding nearest neighbour vectors
4. Rank retrieved frames by number of corresponding regions
5. Verify retrieved frame based on spatial consistency

Slide credit: J. Sivic
Visual retrieval / search

Establish correspondences between object model image and images in the database by **nearest neighbour matching** on SIFT vectors

Slide credit: J. Sivic
Indexing local features

With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

- Low-dimensional descriptors: can use standard efficient data structures for nearest neighbor search

- High-dimensional descriptors: approximate nearest neighbor search methods more practical

- Inverted file indexing schemes

Slide credit: J. Sivic
Nearest-neighbor matching

Solve following problem for all feature vectors, \(x_j\), in the query image:

\[
\forall j \quad NN(j) = \arg \min_i ||x_i - x_j||
\]

where \(x_i\) are features in database images.

Nearest-neighbour matching is the major computational bottleneck

- Linear search performs \(dn\) operations for \(n\) features in the database and \(d\) dimensions
- No exact methods are faster than linear search for \(d>10\)
- Approximate methods can be much faster, but at the cost of missing some correct matches. Failure rate gets worse for large datasets.

Slide credit: J. Sivic
Indexing local features: approximate nearest neighbor search

Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]

Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability [Indyk & Motwani, 1998]

Slide credit: J. Sivic
K-d tree construction

Simple 2D example

Slide credit: Anna Atramentov
K-d tree query

Slide credit: Anna Atramentov
Approximate nearest neighbour K-d tree search

Key idea:
• Limit the number of neighbouring k-d tree bins to explore

• Search k-d tree bins in order of distance from query

• Requires use of a priority queue

Slide credit: J. Sivic
Fraction of nearest neighbors found

Results:
Speedup by several orders of magnitude over linear search

100,000 uniform points in 12 dimensions.
Approximate nearest neighbour
K-d tree search

- How to choose the dimension to split and the splitting point?
- Multiple randomized trees increase the chances of finding nearby points

Finding (approximate) nearest neighbours in $O(\log N)$ time

$N$ ... number of data points

Slide credit: J. Sivic
Indexing local features: approximate nearest neighbor search

Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]

Locality-Sensitive Hashing (LSH), a randomized hashing technique using hash functions that map similar points to the same bin, with high probability [Indyk & Motwani, 1998]

Slide credit: J. Sivic
Locality Sensitive Hashing (LSH)

- Choose a random projection
- Project points
- Points close in the original space remain close under the projection
- Unfortunately, converse not true
- Answer: use multiple quantized projections which define a high-dimensional “grid”

Slide credit: J. Sivic
Locality Sensitive Hashing (LSH)

- Cell contents can be efficiently indexed using a hash table
- Repeat to avoid quantization errors near the cell boundaries
- Point that shares at least one cell = potential candidate
- Compute distance to all candidates

Slide credit: J. Sivic
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, we’ll need to map our features to “visual words”.

Slide credit: J. Sivic
Object → Bag of ‘words’
Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that the retinal image was transmitted point by point to visual centers in the brain; the cerebral cortex was a movie screen, so to speak, upon which the image fell. Through the discoveries of Hubel and Wiesel we now know that the perception of a more complicated course of events is a step-wise analysis in a system of nerve cells stored in columns. In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $750bn, compared with a 18% rise in imports to $660bn. This is likely to further annoy the US, which has long argued that China's exports are unfairly cheap and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.

Analogy to documents

Analogy to documents

Analogy to documents

Analogy to documents
A clarification: definition of “BoW”

Looser definition
• Independent features

Slide credit L. Fei-Fei
A clarification: definition of “BoW”

Looser definition
  • Independent features

Stricter definition
  • Independent features
  • histogram representation

Slide credit L. Fei-Fei
learning

feature detection & representation

image representation

codewords dictionary

category models (and/or) classifiers

recognition

category model selection

classifiers

classification decision

Slide credit L. Fei-Fei
Visual words: main idea

Extract some local features from a number of images ...

E.g., SIFT descriptor space: each point is 128-dimensional

Slide credit: D. Nister
Visual words: main idea

Slide credit: D. Nister
Visual words: main idea

Slide credit: D. Nister
Visual words: main idea

Slide credit: D. Nister
Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space

- Quantize via clustering, let cluster centers be the prototype “words”

K. Grauman, B. Leibe
Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space

- Determine which word to assign to each new image region by finding the closest cluster center.

K. Grauman, B. Leibe
Visual words

Example: each group of patches belongs to the same visual word
Visual words

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.


Slide credit: J. Sivic
Inverted file index for images comprised of visual words

- Score each image by the number of common visual words (tentative correspondences)
- But: does not take into account spatial layout of regions
Clustered / quantization methods

- k-means (typical choice), agglomerative clustering, mean-shift,…

- Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  - Vocabulary tree [Nister & Stewenius, CVPR 2006]
Quantization using K-means

- K-means overview:

  1. Initialize cluster centres
  2. Find nearest cluster to each datapoint \((\textbf{slow})\) \(O(NK)\)
  3. Re-compute cluster centres as centroid

- K-means provably locally minimizes the sum of squared errors (SSE) between a cluster centre and its points

- But: The quantizer depends on the initialization.

- The nearest neighbour search is the bottleneck

Slide credit: J. Sivic
Approximate K-means

- Use the approximate nearest neighbour search (randomized forest of kd-trees) to determine the closest cluster centre for each data point.

- Original K-means complexity $= O(NK)$
- Approximate K-means complexity $= O(N \log K)$
- Can be scaled to very large $K$.

Slide credit: J. Sivic
Clustering / quantization methods

• k-means (typical choice), agglomerative clustering, mean-shift,…

• Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  • Vocabulary tree [Nister & Stewenius, CVPR 2006]
Example: Recognition with Vocabulary Tree

Tree construction:

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Training: Filling the tree

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Training: Filling the tree

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Training: Filling the tree

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Training: Filling the tree

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Training: Filling the tree

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree

Recognition

Verification on spatial layout

[Nister & Stewenius, CVPR’06]

Slide credit: David Nister
Vocabulary Tree: Performance

Evaluated on large databases
  • Indexing with up to 1M images

Online recognition for database of 50,000 CD covers
  • Retrieval in ~1s

Find experimentally that large vocabularies can be beneficial for recognition

[Nister & Stewenius, CVPR’06]
‘Bag of words’ model in text

Document = histogram of word frequencies ('bag of words' model)

$$d = \begin{bmatrix} \ldots & 0 & 1 & \ldots & 2 & 0 & \ldots \end{bmatrix}$$

Term-document matrix

Hofmann 2001

Slide credit: J. Sivic
“Bag of visual words”
Beyond Bag of Words

- Use the **position** and **shape** of the underlying features to improve retrieval quality

- Both images have many matches – which is correct?

Slide credit: J. Sivic
Beyond Bag of Words

- We can measure **spatial consistency** between the query and each result to improve retrieval quality

Many spatially consistent matches – **correct result**

Few spatially consistent matches – **incorrect result**

Slide credit: J. Sivic
Beyond Bag of Words

- Extra bonus – gives **localization** of the object

Slide credit: J. Sivic
Considered transformation

2D affine geometric transformation

\[
\begin{pmatrix}
  x' \\
  y'
\end{pmatrix} = \begin{bmatrix} H \end{bmatrix} \begin{pmatrix} x \\
  y \end{pmatrix} + \begin{pmatrix} t_x \\
  t_y \end{pmatrix}
\]

where \( H \) is a 2x2 non-singular matrix

Approximation to a planar homography (projective transformation)

Slide credit: J. Sivic
Review: Robust line estimation - RANSAC

Fit a line to 2D data containing outliers

There are two problems

1. a line fit which minimizes perpendicular distance
2. a classification into inliers (valid points) and outliers

Solution: use robust statistical estimation algorithm RANSAC (RANdom Sample Consensus) [Fishler & Bolles, 1981]
RANSAC robust line estimation

Repeat

1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

Choose the line with the largest number of inliers

- Compute least squares fit of line to inliers (regression)

Slide credit: J. Sivic
How many samples?

**Number of samples $N$**

- Choose $N$ so that, with probability $p$, at least one random sample is free from outliers
- e.g.:
  - $p=0.99$
  - outlier ratio: $e$

\[
\left(1 - \left(1 - e\right)^s\right)^N = 1 - p
\]

Probability a randomly picked point is an inlier

Probability of all points in a sample (of size $s$) are inliers

Source: M. Pollefeys
How many samples?

Number of samples $N$

- Choose $N$ so that, with probability $p$, at least one random sample is free from outliers
- e.g.:
  - $p = 0.99$
  - outlier ratio: $e$

$$\left(1 - (1 - e)^s\right)^N = 1 - p$$

$$N = \log(1 - p) / \log(1 - (1 - e)^s)$$

<table>
<thead>
<tr>
<th>$s$</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>17</td>
<td>458</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>19</td>
<td>35</td>
<td>4603</td>
</tr>
</tbody>
</table>
| 4   | 3  | 5   | 9   | 17  | 34  | 72  | 4.6e4|}
| 5   | 4  | 6   | 12  | 26  | 57  | 146 | 4.6e5|
| 6   | 4  | 7   | 16  | 37  | 97  | 293 | 4.6e6|
| 7   | 4  | 8   | 20  | 54  | 163 | 588 | 4.6e7|
| 8   | 5  | 9   | 26  | 78  | 272 | 1177| 4.6e8|

Source: M. Pollefeys
Example: line fitting

\[ p = 0.99 \]
\[ s = ? \]
\[ e = ? \]
\[ N = ? \]

Source: M. Pollefeys
Example: line fitting

\[ p = 0.99 \]
\[ s = 2 \]
\[ e = \frac{2}{10} = 0.2 \]

\[ N = 5 \]

Compare with exhaustively trying all point pairs:

\[ \binom{10}{2} = \frac{10 \times 9}{2} = 45 \]

<table>
<thead>
<tr>
<th>s</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>43</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>17</td>
<td>458</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>19</td>
<td>35</td>
<td>4603</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>17</td>
<td>34</td>
<td>72</td>
<td>4.6e4</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>26</td>
<td>57</td>
<td>146</td>
<td>4.6e5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>37</td>
<td>97</td>
<td>293</td>
<td>4.6e6</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>20</td>
<td>54</td>
<td>163</td>
<td>588</td>
<td>4.6e7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>78</td>
<td>272</td>
<td>1177</td>
<td>4.6e8</td>
</tr>
</tbody>
</table>

Source: M. Pollefeys
Algorithm summary – RANSAC robust estimation of 2D affine transformation

Repeat

1. Select 1 region to region correspondence (equivalent to 3 point correspondences)
2. Compute $H$ (2x2 matrix) + $t$ (2x1) vector for translation
3. Measure support (number of inliers within threshold distance, i.e. $d^2_{\text{transfer}} < t$)

\[ d^2_{\text{transfer}} = d(x, H^{-1}x')^2 + d(x', Hx)^2 \]

Choose the $(H, t)$ with the largest number of inliers
(Re-estimate $(H, t)$ from all inliers)

Slide credit: J. Sivic
Estimating spatial correspondences

1. Test each correspondence
Estimating spatial correspondences

2. Compute a planar affine transformation (6 dof)

Need just one correspondence
Estimating spatial correspondences

3. Score by number of consistent matches

Re-estimate full affine transformation (6 dof)

Slide credit: J. Sivic
Verification by spatial layout - overview

1. Query

2. Initial retrieval set (bag of words model)

3. Spatial verification (re-rank on # of inliers)

Slide credit: J. Sivic
Oxford buildings dataset

- Automatically crawled from **flickr**
- Consists of:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th># images</th>
<th># features</th>
<th>Descriptor size</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>1024 × 768</td>
<td>5,062</td>
<td>16,334,970</td>
<td>1.9 GB</td>
</tr>
<tr>
<td>ii</td>
<td>1024 × 768</td>
<td>99,782</td>
<td>277,770,833</td>
<td>33.1 GB</td>
</tr>
<tr>
<td>iii</td>
<td>500 × 333</td>
<td>1,040,801</td>
<td>1,186,469,709</td>
<td>141.4 GB</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1,145,645</td>
<td>1,480,575,512</td>
<td>176.4 GB</td>
</tr>
</tbody>
</table>

Slide credit: J. Sivic
Oxford buildings dataset

- Landmarks plus queries used for evaluation

- Ground truth obtained for 11 landmarks

- Evaluate performance by mean Average Precision

Slide credit: J. Sivic
Measuring retrieval performance: Precision - Recall

- **Precision**: % of returned images that are relevant

- **Recall**: % of relevant images that are returned

Slide credit: J. Sivic
Average Precision

- A good AP score requires both high recall and high precision
- Application-independent

Performance measured by mean Average Precision (mAP) over 55 queries on 100K or 1.1M image datasets

Slide credit: J. Sivic
Mean Average Precision variation with vocabulary size

<table>
<thead>
<tr>
<th>vocab size</th>
<th>bag of words</th>
<th>spatial</th>
</tr>
</thead>
<tbody>
<tr>
<td>50K</td>
<td>0.473</td>
<td>0.599</td>
</tr>
<tr>
<td>100K</td>
<td>0.535</td>
<td>0.597</td>
</tr>
<tr>
<td>250K</td>
<td>0.598</td>
<td>0.633</td>
</tr>
<tr>
<td>500K</td>
<td>0.606</td>
<td>0.642</td>
</tr>
<tr>
<td>750K</td>
<td>0.609</td>
<td>0.630</td>
</tr>
<tr>
<td>1M</td>
<td>0.618</td>
<td>0.645</td>
</tr>
<tr>
<td>1.25M</td>
<td>0.602</td>
<td>0.625</td>
</tr>
</tbody>
</table>

Slide credit: J. Sivic
Query Expansion in text

**In text:**

- Reissue top n responses as queries
- Pseudo/blind relevance feedback
- Danger of topic drift

**In vision:**

- Reissue *spatially verified* image regions as queries

---

Slide credit: J. Sivic
Query expansion in text - example

Original query: Hubble Telescope Achievements

Query expansion: Select top 20 terms from top 20 documents

Added terms:
- telescope
- hubble
- space
- nasa
- ultraviolet
- shuttle
- mirror
- telescopes
- earth
- discovery
- orbit
- flaw
- scientists
- launch
- stars
- universe
- mirrors
- light
- optical
- species

Example from: Jimmy Lin, University of Maryland

Slide credit: J. Sivic
Automatic query expansion

Visual word representations of two images of the same object may differ (due to e.g. detection/quantization noise) resulting in missed returns

Initial returns may be used to add new relevant visual words to the query

Strong spatial model prevents ‘drift’ by discarding false positives

[Chum, Philbin, Sivic, Isard, Zisserman, ICCV’07]
Visual query expansion - overview

1. Original query

2. Initial retrieval set

3. Spatial verification

4. New enhanced query

5. Additional retrieved images

Slide credit: J. Sivic
Query Expansion

Query Image

Originally retrieved image

Originally not retrieved

Slide credit: J. Sivic
Query Expansion

Slide credit: J. Sivic
Query Expansion

Slide credit: J. Sivic
Query Expansion
Query Expansion

New expanded query is formed as

- the average of visual word vectors of spatially verified returns
- only inliers are considered
- regions are back-projected to the original query image
Query Expansion

Query image

Originally retrieved

Retrieved only after expansion

Slide credit: J. Sivic
<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Oxford + Flickr1 dataset</th>
<th>Oxford + Flickr1 + Flickr2 dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OK</td>
<td>Junk</td>
</tr>
<tr>
<td>All Souls</td>
<td>78</td>
<td>111</td>
</tr>
<tr>
<td>Ashmolean</td>
<td>25</td>
<td>31</td>
</tr>
<tr>
<td>Balliol</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>Bodleian</td>
<td>24</td>
<td>30</td>
</tr>
<tr>
<td>Christ Church</td>
<td>78</td>
<td>133</td>
</tr>
<tr>
<td>Cornmarket</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>Hertford</td>
<td>24</td>
<td>31</td>
</tr>
<tr>
<td>Keble</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>Magdalen</td>
<td>54</td>
<td>103</td>
</tr>
<tr>
<td>Pitt Rivers</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Radcliffe Cam.</td>
<td>221</td>
<td>348</td>
</tr>
<tr>
<td>Total</td>
<td>539</td>
<td>838</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Slide credit: J. Sivic
After expansion

Average Precision histogram for 55 queries

Slide credit: J. Sivic
Other applications of local invariant features

Sony Aibo
(Evolution Robotics)

SIFT usage
- Recognize docking station
- Communicate with visual cards

Other uses
- Place recognition
- Loop closure in SLAM

Slide credit: David Lowe
Example Applications

Mobile tourist guide
- Self-localization
- Object/building recognition
- Photo/video augmentation

[Quack, Leibe, Van Gool, CIVR’08]
Web Demo: Movie Poster Recognition

50’000 movie posters indexed

Query-by-image from mobile phone available in Switzerland


K. Grauman, B. Leibe
Application: Image Auto-Annotation

Moulin Rouge

Old Town Square (Prague)

Tour Montparnasse

Colosseum

Left: Wikipedia image
Right: closest match from Flickr

[Quack CIVR’08]
Matching in large unordered datasets

Slide credit: J. Sivic
Matching in large unordered datasets

Slide credit: J. Sivic
Photo Tourism: Exploring Photo Collections in 3D

Noah Snavely
Steven M. Seitz

University of Washington

Richard Szeliski

Microsoft Research

© 2006 Noah Snavely

Thanks to the authors for making slides available!
Photo Tourism
Photo Tourism overview

System for interactive browsing and exploring large collections of photos of a scene. Computes viewpoint of each photo as well as a sparse 3d model of the scene.
Input Photos
Photo Tourism overview

Input photographs → Scene reconstruction → Photo Explorer
Scene reconstruction

- Automatically estimate
  - position, orientation, and focal length of cameras
  - 3D positions of feature points
Feature detection

- Detect features using SIFT [Lowe, IJCV 2004]
Feature detection

- Detect features using SIFT [Lowe, IJCV 2004]
SIFT Reminder

Image gradients

Keypoint descriptor
Pairwise feature matching

- Match features between each pair of images
Pairwise feature matching

- Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs

Fundamental matrix –

$F$ is a 3x3 matrix with rank 2 such that for:

Corresponding points in stereo pair $y_1$ and $y_2$

$$y_2^T F y_1 = 0.$$
Correspondence estimation

- Link up pairwise matches to form connected components of matches across several images
Structure from motion

\[ \text{minimize } f(R, T, P) \]
Incremental structure from motion

- Optimize parameters for two cameras and common points
- Find new image with most matches to existing points
- Initialize new camera using pose estimation
- Bundle adjust
- Add new points
- Bundle adjust
Incremental structure from motion
Reconstruction performance

- For photo sets from the Internet, 20% to 75% of the photos were registered
- Most unregistered photos belonged to different connected components

Running time:
- < 1 hour for 80 photos
- > 1 week for 2600 photos
Photo Tourism overview

Input photographs → Scene reconstruction → Photo Explorer
Navigation controls

- Free-flight navigation
- Object-based browsing
- Relation-based browsing
- Overhead map
Free Flight Navigation
Object-based browsing
Relation-based browsing

Find all similar images

Find all details

Find all zoom outs

Move left

Zoom in

Move right

Zoom out
Rendering
Rendering
Annotations
Saint Basil's Cathedral
Trafalgar Square
Rockefeller Center
Mount Rushmore
Great Wall Fly Through
## Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th># input</th>
<th># registered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trevi Fountain</td>
<td>466</td>
<td>360</td>
</tr>
<tr>
<td>Yosemite</td>
<td>325</td>
<td>1,893</td>
</tr>
<tr>
<td>Notre Dame</td>
<td>597</td>
<td>2,635</td>
</tr>
<tr>
<td>Prague</td>
<td>197</td>
<td>235</td>
</tr>
<tr>
<td>Great Wall</td>
<td>82</td>
<td>120</td>
</tr>
<tr>
<td>Trafalgar Square</td>
<td>1,893</td>
<td>278</td>
</tr>
</tbody>
</table>
Reconstruction running time

- **Great Wall**: 82 / 120 photos registered
  
  Running time: ~ 3 hours

- **Notre Dame**: 597 / 2,635 photos registered
  
  Running time: ~ 2 weeks
Advantages of 3D over 2D

- 3D geometry has multi-image consistency
- Can annotate point cloud directly
- Can import annotations from georeferenced sources (e.g., landmark databases)
- Can use depth as cue for rejecting outliers in selection
Contributions

- Automated system for registering photo collections in 3D for interactive exploration
- Structure from motion algorithm demonstrated on hundreds of photos from the Internet
- Photo exploration system combining new image-based rendering and photo navigation techniques
Limitations / Future work

• Not all photos can be reliably matched

→ Integrating GPS & other localization info.

• Structure from motion scalability

→ More efficient (sparse) algorithms

• Plane-based transitions lack parallax
Subsequent work

• Photo explorer scalability
  – Design client-server architecture for streaming images and geometry at required resolution
  – Scale to *all* of the world’s photos (and videos...)
  – Photosynth project at Microsoft Live Labs (live demo)
Today

- Lowe
- Video Google
- Total Recall
- Photo Tourism
Feb 3rd – Global features (HoG, Gist, Motion History, etc.)


Optional Readings:
