Announcements

• PS2 due Saturday 11:59 am

Approximating the Laplacian

• We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

\[ \nabla^2 \approx \sigma^2 \left( G_{\sigma}(x,y) + G_{\sigma}(x,y) \right) \]

(\text{Laplacian})

\[ \text{DoG} = G(x,y,k\sigma) - G(x,y,\sigma) \]

(\text{Difference of Gaussians})
Recap: Features and filters
Transforming and describing images; textures, colors, edges

Recap: Grouping & fitting
Clustering, segmentation, fitting; what parts belong together?

Recognition and learning
Recognizing objects and categories, learning techniques
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).

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

Indexing local features
• When we see close points in feature space, we have similar descriptors, which indicates similar local content.

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?
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”.

Text retrieval vs. image search

- What makes the problems similar, different?

Visual words: main idea

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

e.g., SIFT descriptor space: each point is 128-dimensional
Each point is a local descriptor, e.g. SIFT vector.

Visual words

- 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 center.
Visual words

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

Visual words and textons

• 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.

Leung & Malik 1999; Varma & Zisserman, 2002

Recall: Texture representation example

<table>
<thead>
<tr>
<th>Windows with</th>
<th>Dimension 1 (mean d/dx value)</th>
<th>Dimension 2 (mean d/dy value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean d/dx value</td>
<td>mean d/dy value</td>
<td></td>
</tr>
<tr>
<td>Win #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows with small gradient in both directions</th>
<th>Windows with primarily vertical edges</th>
<th>Both</th>
</tr>
</thead>
</table>
Visual vocabulary formation

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

Inverted file index

- Database images are loaded into the index mapping words to image numbers

Inverted file index

When will this give us a significant gain in efficiency?

- New query image is mapped to indices of database images that share a word.
• If a local image region is a visual word, how can we summarize an image (the document)?

Analogy to documents

Sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image

Hubel, Wiesel

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

China is forecasting a trade surplus of $610bn ($510bn) this year, a threefold increase on 2004’s $200bn. The Commerce Ministry said the surplus would be created from the rapid rise in exports, an increase in US trade with China, and the rising dollar.

The trade surplus has hardly decreased in the past decade, but US wants the Yuan to rise further to allow the US to rise further in value.
Bags of visual words

- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.

Comparing bags of words

- Rank frames by normalized scalar product between their occurrence counts—nearest neighbor search for similar images.

\[
\text{sim}(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^{V} d_j(i) \cdot q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} \cdot \sqrt{\sum_{i=1}^{V} q(i)^2}}
\]

for vocabulary of V words

Kristen Grauman

Bags of words for content-based image retrieval

Visually defined query

Find this clock

Find this place

"Groundhog Day" [Ramms, 1990]

Slide from Andrew Zisserman

Sivic & Zisserman, ICCV 2003
Scoring retrieval quality

Vocabulary Trees: hierarchical clustering for large vocabularies
Vocabulary Tree

• Training: Filling the tree

Slide credit: David Nister [Nister & Stewenius, CVPR'06]
What is the computational advantage of the hierarchical representation bag of words, vs. a flat vocabulary?
Vocabulary Tree

- Recognition
- RANSAC verification

(Nister & Stewenius, CVPR'06)
Slide credit: David Nister

Bags of words: pros and cons

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides vector representation for sets
+ good results in practice

- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear

Summary So Far

- Matching local invariant features: useful to provide matches to find objects and scenes.
- Bag of words representation: quantize feature space to make discrete set of visual words
- Inverted index: pre-compute index to enable faster search at query time
### Instance recognition

- Motivation – visual search
- **Visual words**
  - quantization, index, bags of words
- **Spatial verification**
  - affine; RANSAC, Hough
- **Other text retrieval tools**
  - tf-idf, query expansion
- **Example applications**

### Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?
Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?

Vocabulary size

Results for recognition task with 6347 images

Performance (%) vs. Nr of Leaf Nodes (10k, 100k, 1M, 10M)

Branching factors:
- 8
- 10
- 16

Kristen Grauman

Nister & Stewenius, CVPR 2006
Kristen Grauman

Instance recognition: remaining issues

- How to summarize the content of an entire image? And gauge overall similarity?
- How large should the vocabulary be? How to perform quantization efficiently?
- Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?
- How to score the retrieval results?
Spatial Verification

Both image pairs have many visual words in common.

Spatial Verification

Only some of the matches are mutually consistent

Spatial Verification: two basic strategies

• RANSAC
  − Typically sort by BoW similarity as initial filter
  − Verify by checking support (inliers) for possible transformations
    • e.g., “success” if find a transformation with > N inlier correspondences

• Generalized Hough Transform
  − Let each matched feature cast a vote on location, scale, orientation of the model object
  − Verify parameters with enough votes
Recall: Fitting an affine transformation

\[
\begin{bmatrix}
    x'_1 \\
    y'_1 \\
    x'_2 \\
    y'_2 \\
    \vdots
\end{bmatrix} =
\begin{bmatrix}
    m_1 & m_2 & m_3 & \cdots & m_n \\
    m_4 & m_5 & m_6 & \cdots & m_n \\
    m_7 & m_8 & m_9 & \cdots & m_n \\
    m_{10} & m_{11} & m_{12} & \cdots & m_n \\
    \vdots & \vdots & \vdots & \ddots & \vdots
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    y_1 \\
    0 \\
    0 \\
    \vdots
\end{bmatrix} =
\begin{bmatrix}
    x'_1 \\
    y'_1 \\
    x'_2 \\
    y'_2 \\
    \vdots
\end{bmatrix}
\]

Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.
**Video Google System**

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- Demo online at: [http://www.robots.ox.ac.uk/~vgg/research/��e/google/index.html](http://www.robots.ox.ac.uk/~vgg/research/藏e/google/index.html)

**Example Applications**

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

[Quack, Leibe, Van Gool, CVR'08]

**Application: Large-Scale Retrieval**

Query results from 5k Flickr images (demo available for 100k set)

[Philbin CVR'07]
Spatial Verification: two basic strategies

- **RANSAC**
  - Typically sort by BoW similarity as initial filter
  - Verify by checking support (inliers) for possible transformations
    - e.g., “success” if find a transformation with > N inlier correspondences

- **Generalized Hough Transform**
  - Let each matched feature cast a vote on location, scale, orientation of the model object
  - Verify parameters with enough votes
Voting: Generalized Hough Transform

- If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).

Model Novel image

Voting: Generalized Hough Transform

- A hypothesis generated by a single match may be unreliable.
- So let each match vote for a hypothesis in Hough space.

Model Novel image

Gen Hough Transform details (Lowe’s system)

- **Training phase:** For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame).
- **Test phase:** Let each match between a test SIFT feature and a model feature vote in a 4D Hough space:
  - Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location.
  - Vote for two closest bins in each dimension.
  - Find all bins with at least three votes and perform geometric verification:
    - Estimate least squares affine transformation.
    - Search for additional features that agree with the alignment.

Recall: difficulties of voting

- Noise/clutter can lead to as many votes as true target
- Bin size for the accumulator array must be chosen carefully
- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.

Gen Hough vs RANSAC

**GHT**
- Single correspondence -> vote for all consistent parameters
- Represents uncertainty in the model parameter space
- Linear complexity in number of correspondences and number of voting cells; beyond 4D vote space impractical
- Can handle high outlier ratio

**RANSAC**
- Minimal subset of correspondences to estimate model -> count inliers
- Represents uncertainty in image space
- Must search all data points to check for inliers each iteration
- Scales better to high-d parameter spaces
Questions?

See you Tuesday!