Local Features and Bag of Words Models

Computer Vision
CS 143, Brown

James Hays
Computer Engineering  Distinguished Lecture Talk

Compressive Sensing, Sparse Representations and Dictionaries: New Tools for Old Problems in Computer Vision and Pattern Recognition

Rama Chellappa, University of Maryland, College Park, MD 20742

Abstract: Emerging theories of compressive sensing, sparse representations and dictionaries are enabling new solutions to several problems in computer vision and pattern recognition. In this talk, I will present examples of compressive acquisition of video sequences, sparse representation-based methods for face and iris recognition, reconstruction of images and shapes from gradients and dictionary-based methods for object and activity recognition.

12:00 noon, Friday October 14, 2011, Lubrano Conference room, CIT room 477.
Previous Class

• Overview and history of recognition
Specific recognition tasks
Scene categorization or classification

- outdoor/indoor
- city/forest/factory/etc.
Image annotation / tagging / attributes

- street
- people
- building
- mountain
- tourism
- cloudy
- brick
- ...

Svetlana Lazebnik
Object detection

• find pedestrians
Image parsing
Today’s class: features and bag of words models

• Representation
  – Gist descriptor
  – Image histograms
  – Sift-like features

• Bag of Words models
  – Encoding methods
Image Categorization

Training Images

Training

Image Features

Classifier Training

Training Labels

Trained Classifier
Image Categorization

Training

Training Images

Image Features

Classifier Training

Training Labels

Trained Classifier

Testing

Test Image

Image Features

Trained Classifier

Prediction Outdoor

Derek Hoiem
Part 1: Image features

Training Images

Image Features

Classifier Training

Trained Classifier

Training Labels
Image representations

• Templates
  – Intensity, gradients, etc.

• Histograms
  – Color, texture, SIFT descriptors, etc.
Image Representations: Histograms

Global histogram

- Represent distribution of features
  - Color, texture, depth, ...

Images from Dave Kauchak
Image Representations: Histograms

Histogram: Probability or count of data in each bin

- **Joint histogram**
  - Requires lots of data
  - Loss of resolution to avoid empty bins

- **Marginal histogram**
  - Requires independent features
  - More data/bin than joint histogram
Image Representations: Histograms

Clustering

Use the same cluster centers for all images

Images from Dave Kauchak
Computing histogram distance

$$\text{histint}(h_i, h_j) = 1 - \sum_{m=1}^{K} \min \{ h_i(m), h_j(m) \}$$

Histogram intersection (assuming normalized histograms)

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{K} \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Chi-squared Histogram matching distance

Cars found by color histogram matching using chi-squared
Histograms: Implementation issues

• Quantization
  – Grids: fast but applicable only with few dimensions
  – Clustering: slower but can quantize data in higher dimensions

• Matching
  – Histogram intersection or Euclidean may be faster
  – Chi-squared often works better
  – Earth mover’s distance is good for when nearby bins represent similar values
What kind of things do we compute histograms of?

- Color
  - L*a*b* color space
  - HSV color space

- Texture (filter banks or HOG over regions)
What kind of things do we compute histograms of?
• Histograms of oriented gradients

SIFT – Lowe IJCV 2004
SIFT vector formation

• Computed on rotated and scaled version of window according to computed orientation & scale
  – resample the window

• Based on gradients weighted by a Gaussian of variance half the window (for smooth falloff)
SIFT vector formation

• 4x4 array of gradient orientation histograms
  – not really histogram, weighted by magnitude
• 8 orientations x 4x4 array = 128 dimensions
• Motivation: some sensitivity to spatial layout, but not too much.
Ensure smoothness

• Gaussian weight

• Trilinear interpolation
  – a given gradient contributes to 8 bins:
    4 in space times 2 in orientation
Reduce effect of illumination

- 128-dim vector normalized to 1
- Threshold gradient magnitudes to avoid excessive influence of high gradients
  - after normalization, clamp gradients >0.2
  - renormalize
Local Descriptors: Shape Context

Count the number of points inside each bin, e.g.:

Count = 4

Log-polar binning: more precision for nearby points, more flexibility for farther points.

Belongie & Malik, ICCV 2001
Shape Context Descriptor
Local Descriptors: Geometric Blur

- Compute edges at four orientations
- Extract a patch in each channel
- Apply spatially varying blur and sub-sample

Example descriptor

Berg & Malik, CVPR 2001

K. Grauman, B. Leibe
Self-similarity Descriptor

Figure 1. These images of the same object (a heart) do NOT share common image properties (colors, textures, edges), but DO share a similar geometric layout of local internal self-similarities.

Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007
Self-similarity Descriptor

Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007
Self-similarity Descriptor

Matching Local Self-Similarities across Images and Videos, Shechtman and Irani, 2007
Learning Local Image Descriptors, Winder and Brown, 2007

- Image Patch
- Smooth $G(x, \sigma)$
- T-Block Filter
- S-Block Pooling
- N-Block Normalize
- Descriptor

64x64 Pixels
~64x64 vectors of dimension $k$
$N$ histograms of dimension $k$

S1: SIFT grid with bilinear weights
S2: GLOH polar grid with bilinear radial and angular weights
S3: 3x3 grid with Gaussian weights
S4: 17 polar samples with Gaussian weights
Right features depend on what you want to know

• Shape: scene-scale, object-scale, detail-scale
  – 2D form, shading, shadows, texture, linear perspective

• Material properties: albedo, feel, hardness, ...
  – Color, texture

• Motion
  – Optical flow, tracked points

• Distance
  – Stereo, position, occlusion, scene shape
  – If known object: size, other objects
Things to remember about representation

• Most features can be thought of as templates, histograms (counts), or combinations

• Think about the right features for the problem
  – Coverage
  – Concision
  – Directness
Bag-of-features models
Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters

Origin 1: Texture recognition

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models


US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary
  Salton & McGill (1983)

2007-01-23: State of the Union Address
George W. Bush (2001-)

1962-10-22: Soviet Missiles in Cuba
John F. Kennedy (1961-63)

US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)

US Presidential Speeches Tag Cloud
http://chir.ag/phernalia/preztags/
Bag-of-features steps

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”
1. Feature extraction

- Regular grid or interest regions
1. Feature extraction

- Detect patches
- Normalize patch
- Compute descriptor

Slide credit: Josef Sivic
1. Feature extraction
2. Learning the visual vocabulary

Slide credit: Josef Sivic
2. Learning the visual vocabulary

Clustering

Slide credit: Josef Sivic
2. Learning the visual vocabulary

Clustering...

Visual vocabulary

Clustering

Slide credit: Josef Sivic
K-means clustering

- Want to minimize sum of squared Euclidean distances between points \( x_i \) and their nearest cluster centers \( m_k \)

\[
D(X, M) = \sum_{\text{cluster } k} \sum_{\text{point } i \text{ in cluster } k} (x_i - m_k)^2
\]

Algorithm:
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center
  - Recompute each cluster center as the mean of all points assigned to it
Clustering and vector quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Codebook can be learned on separate training set
  - Provided the training set is sufficiently representative, the codebook will be “universal”

- The codebook is used for quantizing features
  - A vector quantizer takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word
Example codebook

Appearance codebook

Source: B. Leibe
Another codebook

Source: B. Leibe
Visual vocabularies: Issues

• How to choose vocabulary size?
  • Too small: visual words not representative of all patches
  • Too large: quantization artifacts, overfitting

• Computational efficiency
  • Vocabulary trees
    (Nister & Stewenius, 2006)
But what about layout?

All of these images have the same color histogram
Spatial pyramid

Compute histogram in each spatial bin
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution

Lazebnik, Schmid & Ponce (CVPR 2006)
Scene category dataset

Multi-class classification results
(100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (vocabulary size: 16)</th>
<th>Strong features (vocabulary size: 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 (1 × 1)</td>
<td>45.3 ±0.5</td>
<td></td>
</tr>
<tr>
<td>1 (2 × 2)</td>
<td>53.6 ±0.3</td>
<td>56.2 ±0.6</td>
</tr>
<tr>
<td>2 (4 × 4)</td>
<td>61.7 ±0.6</td>
<td>64.7 ±0.7</td>
</tr>
<tr>
<td>3 (8 × 8)</td>
<td>63.3 ±0.8</td>
<td>66.8 ±0.6</td>
</tr>
</tbody>
</table>
Caltech101 dataset


Multi-class classification results (30 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (16)</th>
<th>Strong features (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0</td>
<td>15.5 ±0.9</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>31.4 ±1.2</td>
<td>32.8 ±1.3</td>
</tr>
<tr>
<td>2</td>
<td>47.2 ±1.1</td>
<td>49.3 ±1.4</td>
</tr>
<tr>
<td>3</td>
<td>52.2 ±0.8</td>
<td><strong>54.0</strong> ±1.1</td>
</tr>
</tbody>
</table>
Bags of features for action recognition

Space-time interest points