Computer Vision for HCI

HOG, Shape Context, and Template Matching

Outline

- Histogram of Gradients
- Shape Context
- Template Matching

Object Recognition

- Want a descriptor that can be employed to reliably recognize objects
- Observation: Local object appearance and shape often well-characterized by distribution of local orientations
  - Do not need precise knowledge (spatial position) of corresponding gradient or edge positions

Histogram of Oriented Gradients (HOG)

- Divide image into small cells (sub-images)
- Accumulate histograms of orientations of pixels within cells (computed from gradients)
- Concatenate histograms for final object representation
  - Can improve accuracy by normalizing histograms for better invariance to illumination
- Parameters vary depending on application
  - Will discuss Dalal 2005 (human detection)

Gradient Computation

- Use simple 1-D mask [-1 0 1]
  - Smoothing not used
  - Compute gradients for each color channel separately
    - Use the color with <u>largest</u> magnitude as pixel's gradient

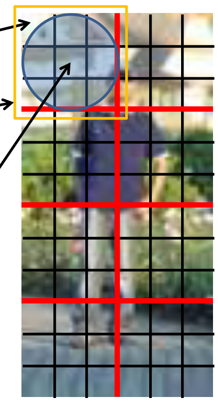
Input Image x gradient y gradient

Orientation Binning

- Each pixel contributes <u>weighted</u> vote for "**edge orientation**" histogram (1-D)
  - [Note: edge orientation is <u>orthogonal</u> to gradient direction ("along" edge) – rotation of gradient angle by 90º]
  - Weighted vote based on gradient magnitude
  - Orientation bins evenly spaced between 0º and 180º
- Votes accumulated over local spatial <u>cells</u>
- Group cells into larger spatial <u>blocks</u>

Rectangular Blocks

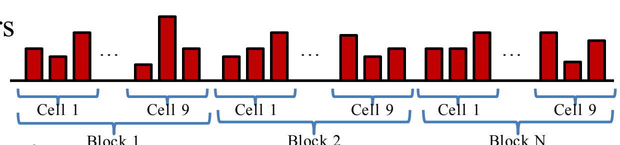
- Blocks generally square
- 3 parameters:
  - Number pixels per <u>cell</u>
  - Number cells per <u>block</u>
  - Number bins per gradient histogram
  - Dalal:
    - Cell = 6x6 pixels
    - Block = 3x3 cells
    - 9-bin histogram (0-180°)
- Reduce weight for pixels near border of <u>block</u> in cell histograms by applying Gaussian spatial window over <u>block</u>



HOG Descriptor

- <u>Local</u> variations in illumination and foreground/background contrast cause gradient strengths to vary
  - Normalize each block descriptor separately $v / \sqrt{\|v\|_2^2 + \varepsilon}$
- HOG descriptor is rasterized concatenation of normalized block descriptors

HOG Descriptor:



- Blocks could overlap
  - Then each cell could contribute to multiple block descriptors

Exploitation

- Person/non-person recognition
- Train classifier with image patches containing a person (positive samples) and image patches without a person (negative samples)
- Scan image (possibly at multiple scales) and compute confidence score from classifier
- Keep patches with score above threshold
- Perform non-maximal suppression

Person Detection Results

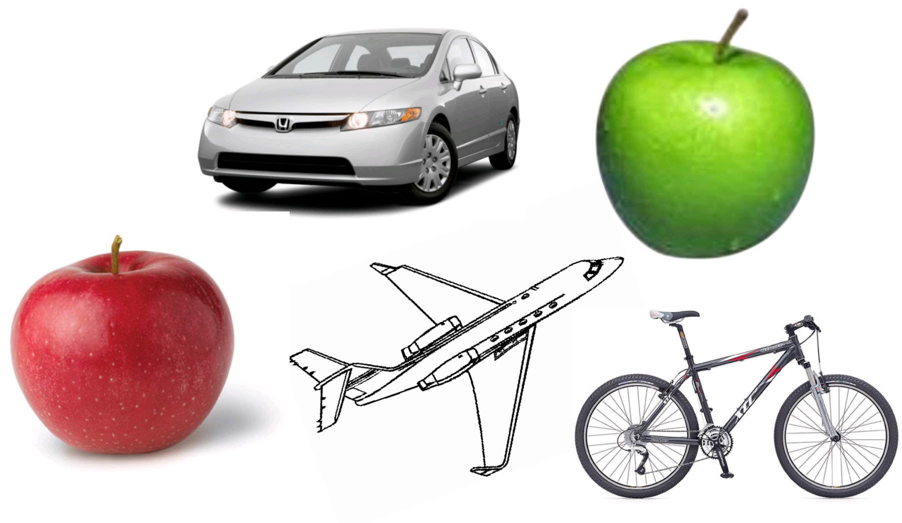
Histogram of Optical Flow (HOF)

- HOG focuses on static appearance information
- What about local motion information?
- Histogram of Optical Flow (HOF)
  - Analogous to HOG
  - Weighted quantization of flow vectors
  - Some approaches have separate bin for no motion

Outline

- Histogram of Gradients
- Shape Context
- Template Matching

Which objects have similar shapes?

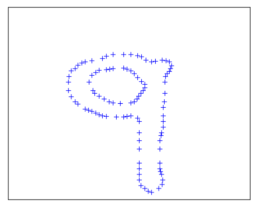


Shape Context

- "Shape context" is a descriptor to coarsely *describe the distribution of points along shape boundary* with respect to a given point
  - Inherently invariant to translation
  - Can make invariant to scale and rotation
  - **Empirically shown to be invariant to small nonlinear transformations, occlusions, and presence of outliers**
- Corresponding points on similar shapes will have similar shape contexts
- Can be incorporated in shape similarity measurement
  - For object recognition

Feature Points

- Represent shapes as discrete set of points on internal and external contours P = $\{p_1,\ldots,p_n\}$
  - Randomly sample points, ensuring a minimum distance between points
  - Do not have to be "key points"

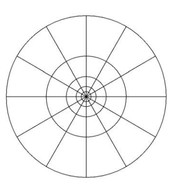
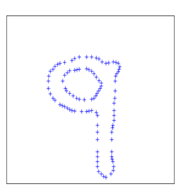
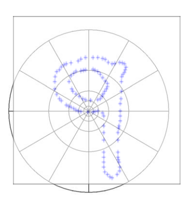
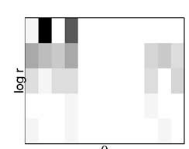



Shape Context

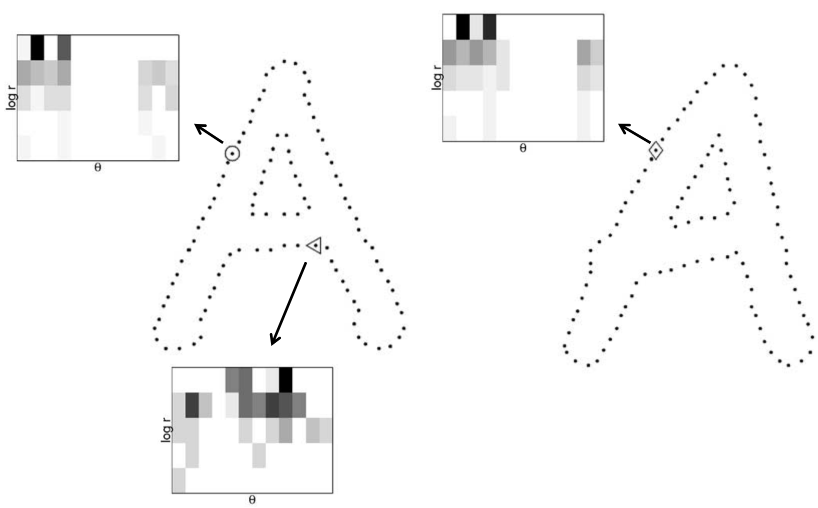
- For each point p_i compute coarse histogram h_i of the relative coordinates for the remaining n - 1 points

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \mathrm{bin}(k)\}$$

- Distribution of "relative" positions is robust, compact, and highly discriminative
- Use bins in log-polar space
  - 12 angular bins, 5 radial bins

Shape Context



Invariance and Robustness

- Representation is translation invariant
- Scale invariance achieved by normalizing all radial distances by mean distance of the point pairs
- Rotational invariance can be achieved by aligning shape contexts to tangent/gradient vectors for each point
  - Not always desirable (e.g., differentiating 6 from 9)
- Empirically shown to be robust to small nonlinear transformations, occlusions, and presence of outliers when transformation model is incorporated

Point Correspondence

- Compute cost of matching point p_i on first shape to point q_j on second shape using χ^2 statistic between **normalized** histograms

$$C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{n} \frac{\left(h_i(k) - h_j(k)\right)^2}{h_i(k) + h_j(k)}$$

Squared-distance inversely weighted by term frequencies to balance/equalize common/uncommon terms

- Minimize total matching cost having one-to-one mapping constraint

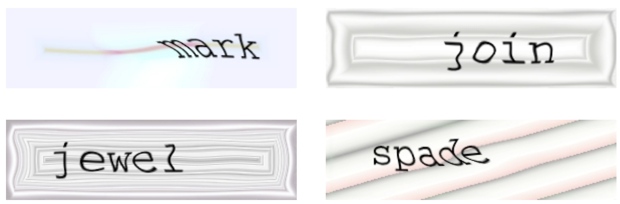
$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right)$$

  - Can add "dummy" nodes for outliers and when unequal number of points in shapes
  - Solve using Hungarian algorithm

Point Correspondence

Employing Shape Context

- Recognition, clustering, or classification based on match scores
- Additional techniques discussed in original paper
  - Handling deformations
- Breaking visual captcha



Outline

- Histogram of Gradients
- Shape Context
- Template Matching

Introduction

- Want to find areas of a search image that are similar to given template image T

Template Image T

Search Image

Best Matching Patch in Search Image

General Approaches

- Template-Based:
  - Utilize raw template (pixels) and find best matching patches in search image
    - Sum-of-absolute differences (SAD)
    - Sum-of-squared differences (SSD)
    - Normalized cross-correlation (NCC)

1) Sum-of-Absolute Differences
(SAD)

- Compute **absolute differences of pixel intensities** of template T and image patch P extracted from search image (note that P is same size as template T)

$$SAD(P,T) = \sum_{R,G,B} \sum_{x,y} \left| P(x,y) - T(x,y) \right|$$

- Compute SAD for all unique patch locations within the search image
- Keep patch with minimum SAD or patches with SAD less than given threshold

SAD Example

Template Image T

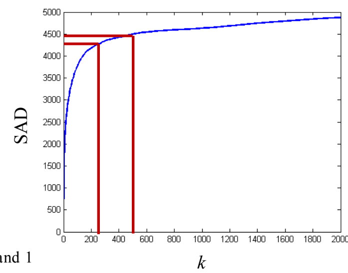
Search Image

Negative SAD, Origin is in center of patch

SAD Example

k^{th} best matching patch

$k = 1$ $k = 10$ $k = 50$ $k = 100$ $k = 150$ $k = 200$ $k = 250$ $k = 500$ $k = 1000$



Similar???

*Pixel values scaled between 0 and 1

2) Sum-of-Squared Differences (SSD)

- Similar to SAD, but replace absolute differences with **squared differences**

$$SSD(P,T) = \sum_{R,G,B} \sum_{x,y} \left(P(x,y) - T(x,y)\right)^2$$

- Compute SSD for all unique patches within the search image
- Keep patch with minimum SSD

SSD Example

Template Image T

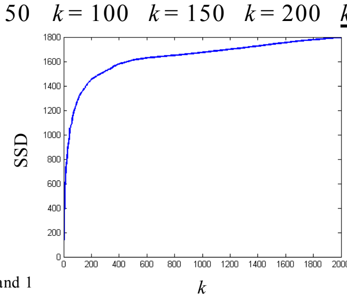
Search Image

Negative SSD,
Origin is in center of patch

SSD Example

k^{th} best matching patch

$k = 1$ $k = 10$ $k = 50$ $k = 100$ $k = 150$ $k = 200$ $k = 250$ $k = 500$ $k = 1000$



*Pixel values scaled between 0 and 1

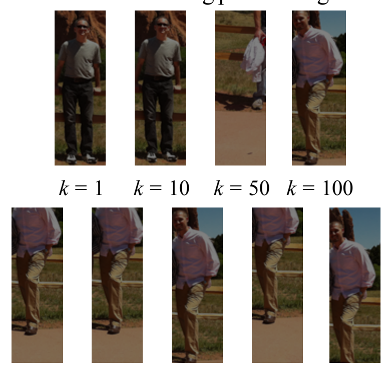
Illumination Changes

- SAD and SSD can work well if the template and search images have the same brightness
  - Problem: images can have varying illumination conditions

Template Image T

Search Image

k^{th} best matching patch using SSD



3) Normalized Cross-Correlation

(NCC)

- Normalize images to remove variations from illumination conditions

Mean of pixel values in patch
(each color computed independently)

$$NCC(P,T) = \sum_{R,G,B} \frac{1}{n-1} \sum_{x,y} \frac{\left(P(x,y)-\overline{P}\right)\cdot\left(T(x,y)-\overline{T}\right)}{\sigma_P \sigma_T}$$

Standard deviation of pixel values in patch
(each color computed independently)

Note: <u>larger</u> values of NCC <u>better</u>!

NCC Example

Template Image T

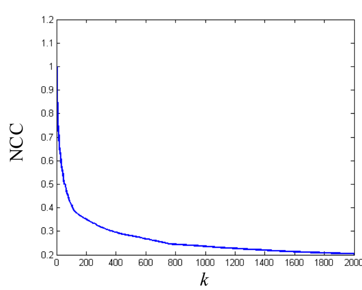
Search Image

NCC,
Origin is in center of patch

NCC Example

k^{th} best matching patch using **color** images

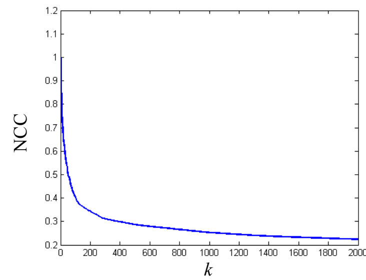
k = 1 | k = 10 | k = 50 | k = 100 | k = 150 | k = 200 | k = 250 | k = 500 | k = 1000



NCC Example

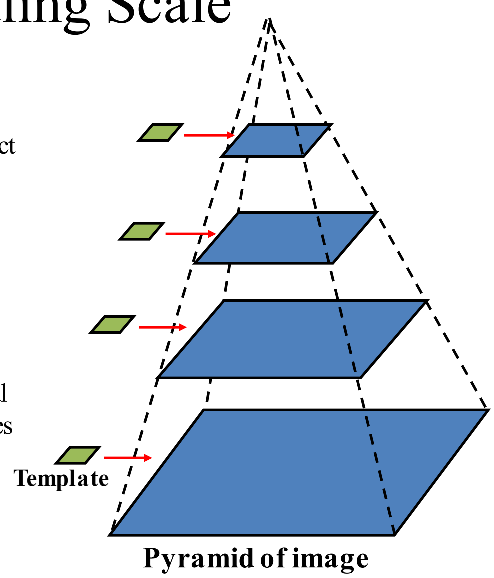
k^{th} best matching patch using **grayscale** images

$k = 1$ $k = 10$ $k = 50$ $k = 100$ $k = 150$ $k = 200$ $k = 250$ $k = 500$ $k = 1000$



Handling Scale

- Construct fixed-size template of the smallest size you want to detect
- Scan through image pyramid
  - Detects larger scales of object higher in the pyramid
- Efficient scanning method, with less pixels to examine overall
  - Instead of repeatedly scaling template and scanning original full-sized image multiple times



Summary

- Histogram of Gradients
  - Descriptor for object recognition
  - Local object appearance and shape often well-characterized by distribution of local orientations
- Shape Context
  - Descriptor to coarsely describe distribution of points along shape boundary with respect to a given point
  - Captcha
- Template Matching
  - SAD, SSD, NCC