Recap: Sliding-Window Object Detection

- If object may be in a cluttered scene, slide a window around looking for it.

Recap: Gradient-based Representations

- Consider edges, contours, and (oriented) intensity gradients
  - Summarize local distribution of gradients with histogram
    - Locally orderless: offers invariance to small shifts and rotations
    - Contrast-normalization: try to correct for variable illumination

Recap: Classifier Construction Choices...

- Nearest neighbor
  - Shakhnarovich, Viola, Darrell 2003
  - Berg, Berg, Malik 2005...
- Neural networks
  - LeCun, Bottou, Bengio, Hafler 1998
  - Rowley, Baluja, Kanade 1998
  - ...
Recap: Viola-Jones Face Detection

- "Rectangular" filters
- Feature output is difference between adjacent regions
- Efficiently computable with integral image; any sum can be computed in constant time
- Avoid scaling images to scale features directly for same cost

Recap: AdaBoost Feature+Classifier Selection

- Want to select the single rectangle feature and threshold that best separates positive (faces) and negative (non-faces) training examples, in terms of weighted error.

Linear Classifiers

- Find linear function to separate positive and negative examples
  - $\mathbf{x}$, positive: $\mathbf{x} \cdot \mathbf{w} + b \geq 0$
  - $\mathbf{x}$, negative: $\mathbf{x} \cdot \mathbf{w} + b < 0$

Support Vector Machines (SVMs)

- Discriminative classifier based on optimal separating hyperplane (i.e. line for 2D case)
- Maximize the margin between the positive and negative training examples
Support Vector Machines

- Want line that maximizes the margin.
  \[ x, \text{positive} (y_i = 1): \quad x_i \cdot w + b \geq 1 \]
  \[ x, \text{negative} (y_i = -1): \quad x_i \cdot w + b \leq -1 \]

- For support vectors, \( x_i \cdot w + b = \pm 1 \)
  \[ \sum_{i} \alpha_i y_i x_i \]

- Quadratic optimization problem:
  Minimize \( \frac{1}{2} w^T w \)
  Subject to \( y_i (w \cdot x_i + b) \geq 1 \)
  \[ \text{Support vectors} \]
  \[ \text{Margin} \]

C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*,
Data Mining and Knowledge Discovery, 1998

Finding the Maximum Margin Line

- Solution:
  \[ w = \sum \alpha_i y_i x_i \]
  \[ w \cdot x + b = \sum \alpha_i y_i x_i \cdot x + b \]

- Classification function:
  \[ f(x) = \text{sign} (w \cdot x + b) \]
  \[ = \text{sign} (\sum \alpha_i y_i x_i \cdot x + b) \]

  If \( f(x) < 0 \), classify as neg.,
  If \( f(x) > 0 \), classify as pos.

- Notice that this relies on an inner product between the test point \( x \) and the support vectors \( x_i \).

- (Solving the optimization problem also involves computing the inner products \( x_i \cdot x_j \) between all pairs of training points)

C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*,
Data Mining and Knowledge Discovery, 1998

Non-Linear SVMs: Feature Spaces

- General idea: The original input space can be mapped to some higher-dimensional feature space where the training set is separable:

  \[ \Phi: x \rightarrow \phi(x) \]

  More on that in the Machine Learning lecture...

Slide from Andrew Moore's tutorial: [http://www.autonlab.org/tutorials/svm.html](http://www.autonlab.org/tutorials/svm.html)

Some Often-Used Kernel Functions

- Linear:
  \[ K(x_i, x_j) = x_i^T x_j \]

- Polynomial of power \( p \):
  \[ K(x_i, x_j) = (1 + x_i^T x_j)^p \]

- Gaussian (radial-basis function):
  \[ K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \]

Slide from Andrew Moore's tutorial: [http://www.autonlab.org/tutorials/svm.html](http://www.autonlab.org/tutorials/svm.html)
Pedestrian detection with HoGs & SVMs

- Navneet Dalal, Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Summary: Sliding-Windows

- **Pros**
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes
  - Good detectors available (Viola & Jones, HOG, etc.)

- **Cons/Limitations**
  - High computational complexity
    - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
    - This puts tight constraints on the classifiers we can use.
    - If training binary detectors independently, this means cost increases linearly with number of classes.
    - With so many windows, false positive rate better be low

Limitations of Sliding Windows (continued)

- Not all objects are “box” shaped

Limitations (continued)

- Non-rigid, deformable objects not captured well with representations assuming a fixed 2D structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions

Limitations (continued)

- If considering windows in isolation, context is lost

Limitations (continued)

- In practice, often entails large, cropped training set (expensive)
- Requiring good match to a global appearance description can lead to sensitivity to partial occlusions
Topics of This Lecture

- Local Invariant Features
  - Motivation
  - Requirements, Invariances
- Keypoint Localization
  - Harris detector
  - Hessian detector
- Scale Invariant Region Selection
  - Automatic scale selection
  - Laplacian-of-Gaussian detector
  - Difference-of-Gaussian detector
  - Combinations
- Local Descriptors
  - Orientation normalization
  - SIFT

Motivation

- Global representations have major limitations
- Instead, describe and match only local regions
- Increased robustness to
  - Occlusions
  - Articulation
  - Intra-category variations

Application: Image Matching

Harder Case

Harder Still?

Answer Below (Look for tiny colored squares)
Application: Image Stitching

Procedure:
- Detect feature points in both images
- Find corresponding pairs

Use these pairs to align the images

General Approach

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Common Requirements

Problem 1:
- Detect the same point independently in both images

We need a repeatable detector!
Common Requirements

- Problem 1:
  - Detect the same point independently in both images
- Problem 2:
  - For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor!

Invariance: Geometric Transformations

Problem 1: Detect the same point independently in both images
Problem 2: For each point correctly recognize the corresponding one

Levels of Geometric Invariance

Requirements

- Region extraction needs to be repeatable and accurate
  - Invariant to translation, rotation, scale changes
  - Robust or covariant to out-of-plane (affine) transformations
  - Robust to lighting variations, noise, blur, quantization
- Locality: Features are local, therefore robust to occlusion and clutter.
- Quantity: We need a sufficient number of regions to cover the object.
- Distinctiveness: The regions should contain “interesting” structure.
- Efficiency: Close to real-time performance.

Many Existing Detectors Available

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Lindeberg ‘98], [Lowe ‘99]
- Harris-/Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-/Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- Others...

Those detectors have become a basic building block for many recent applications in Computer Vision.

Keypoint Localization

Goals:
- Repeatable detection
- Precise localization
- Interesting content

⇒ Look for two-dimensional signal changes
Finding Corners

- Key property:
  - In the region around a corner, image gradient has two or more dominant directions
- Corners are repeatable and distinctive


Corners as Distinctive Interest Points

- Design criteria
  - We should easily recognize the point by looking through a small window (locality)
  - Shifting the window in any direction should give a large change in intensity (good localization)

Harris Detector Formulation

- Change of intensity for the shift \([u,v]\):
  \[
  E(u,v) = \sum_{x,y} w(x,y) [I(x+u,y+v) - I(x,y)]^2
  \]

  \[\text{Window function } w(x,y) = \begin{cases} 1 & \text{in window} \\ 0 & \text{outside} \end{cases} \quad \text{or} \quad \text{Gaussian}\]

Harris Detector Formulation

- This measure of change can be approximated by:
  \[
  E(u,v) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}
  \]

  where \(M\) is a 2x2 matrix computed from image derivatives:
  \[
  M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}
  \]

  \[\text{Gradient with} \quad \text{times gradient} \quad \text{with respect to} \quad y \]

  \[
  M = \begin{bmatrix} \sum_{x,y} I_x I_y & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_x I_y & \sum_{x,y} I_x I_y \end{bmatrix} = \sum_{x,y} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix}
  \]

What Does This Matrix Reveal?

- First, let’s consider an axis-aligned corner:
What Does This Matrix Reveal?

- First, let’s consider an axis-aligned corner:
  \[
  M = \begin{bmatrix}
  \sum I_x^2 & \sum I_x I_y \\
  \sum I_x I_y & \sum I_y^2
  \end{bmatrix} = \begin{bmatrix}
  \lambda_1 & 0 \\
  0 & \lambda_2
  \end{bmatrix}
  \]

- This means:
  - Dominant gradient directions align with \(x\) or \(y\) axis
  - If either \(\lambda\) is close to 0, then this is not a corner, so look for locations where both are large.
- What if we have a corner that is not aligned with the image axes?

Interpreting the Eigenvalues

- Classification of image points using eigenvalues of \(M\):
  - Fast approximation:
    - Avoid computing the eigenvalues
    - \(\alpha\) constant (0.04 to 0.06)

Window Function \(w(x,y)\)

- Option 1: uniform window
  - Sum over square window:
    \[
    M = \sum_{x,y} I_x^2 I_y^2 + I_x I_y
    \]
  - Problem: not rotation invariant
- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum:
    \[
    M = g(\sigma_x, \sigma_y) I_x^2 I_y^2 + I_x I_y
    \]
  - Result is rotation invariant

General Case

- Since \(M\) is symmetric, we have \(M = RR^T\)
  - (Eigenvalue decomposition)
- We can visualize \(M\) as an ellipse with axis lengths determined by the eigenvalues and orientation determined by \(R\)

Corner Response Function

\[
R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2
\]

Summary: Harris Detector [Harris88]

- Compute second moment matrix (autocorrelation matrix)
  \[
  M(\sigma_x, \sigma_y) = \int g(x, y) I_x^2 I_y^2 + I_x I_y dy
  \]
- 1. Image derivatives
- 2. Square of derivatives
- 3. Gaussian filter \(g(\sigma)\)
- 4. Cornerness function - two strong eigenvalues
  \[
  R = \det(M(\sigma_x, \sigma_y)) - \alpha \text{trace}(M(\sigma_x, \sigma_y))^2
  = g(t_1^2) g(t_2^2) - g(t_1 t_2) [g(t_1^2) - g(t_2^2)]
  \]
- 5. Perform non-maximum suppression
Harris Detector: Workflow

- Compute corner responses $R$

- Take only the local maxima of $R$, where $R > \text{threshold}$.

Harris Detector – Responses [Harris88]

Effect: A very precise corner detector.
Harris Detector - Responses [Harris88]

- Results are well suited for finding stereo correspondences

Hessian Detector - Responses [Beaudet78]

- Effect: Responses mainly on corners and strongly textured areas.

Hessian Detector: Properties

- Rotation invariance?
- Scale invariance?

\[
\begin{bmatrix}
I_{xx} & I_{xy} \\
I_{xy} & I_{yy}
\end{bmatrix}
\]

\[
\det(\text{Hessian}(I)) = I_{xx}I_{yy} - I_{xy}^2
\]

In Matlab:

\[
\text{det}(I_{xx}, I_{xy}, I_{yy}) = (I_{xx}I_{yy} - I_{xy}^2)
\]

not invariant to image scale!

Corner response \( R \) is invariant to image rotation

Hessian determinant

\[
\text{Hessian}(I) = \begin{bmatrix}
I_{xx} & I_{xy} \\
I_{xy} & I_{yy}
\end{bmatrix}
\]

Note: these are 2nd derivatives!

Intuition: Search for strong derivatives in two orthogonal directions

Corner is invariant to image rotation!
Hessian Detector - Responses (Beaudet78)

For most local feature detectors, executables are available online:
- http://robots.ox.ac.uk/~vgg/research/affine
- http://www.cs.ubc.ca/~lowe/keypoints/
- http://www.vision.ee.ethz.ch/~surf

Affine Covariant Features

Affine Covariant Region Detectors

References and Further Reading

- Read David Lowe’s SIFT paper
  - D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60(2), pp. 91-110, 2004

- Good survey paper on Int. Pt. detectors and descriptors

- Try the example code, binaries, and Matlab wrappers
  - Good starting point: Oxford interest point page
    http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries