Computer Vision - Lecture 11
Local Features
10.12.2009

Bastian Leibe
RWTH Aachen
http://www.umic.rwth-aachen.de/multimedia
leibe@umic.rwth-aachen.de

Lecture Evaluation
- Please fill out the evaluation forms...

Talk Announcement
David J Fleet (University of Toronto) 15.12., 16:00h, UMIC 025
Physics-Based Models for Human Motion Analysis

The recovery and analysis of human motion from video is a key enabling technology for myriad applications (e.g., in man-machine interaction, computer graphics, biome-trics, and biomechanics). Many are confident that the problem will be solved, in part with the help of models for how people move. Current state-of-the-art models are usually learned from human motion capture data, but there are questions about whether such models will work well in unconstrained situations.

In this talk we advocate a new class of models, derived in part from principles of Newtonian dynamics and biomechanics. We describe two examples of such models: The first, inspired by low-dimensional passive-dynamic models of human locomotion, was designed for monocular tracking of walking people. The second uses physical principles to facilitate the inference of human muscle forces and one’s interactions with external surfaces. (Joint work with Marcus Brubaker and Leonid Sigal).

Course Outline
- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Object Categorization I
  - Sliding Window based Object Detection
- Local Features & Matching
  - Local Features - Detection and Description
  - Recognition with Local Features
- Object Categorization II
  - Part based Approaches
- 3D Reconstruction
- Motion and Tracking

Recap: Sliding-Window Object Detection
- If object may be in a cluttered scene, slide a window around looking for it.

- Essentially, this is a brute-force approach with many local decisions.

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: 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

Recap: AdaBoost

- Final classifier is combination of the weak classifiers
- Weights increased 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.

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, scale features directly for same cost

Recap: Classifier Construction Choices...

- Nearest neighbor
- Support Vector Machines
- Boosting
- Neural networks
- Conditional Random Fields

Recap: AdaBoost

- Weak Classifier 1
- Weak Classifier 2
- Weak classifier 3
- Final classifier is combination of the weak classifiers

Recap: Viola-Jones Face Detection

- Train cascade of classifiers with AdaBoost
- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
- Implementation available in OpenCV
  - http://sourceforge.net/projects/opencvlibrary/
Recap: Non-Linear SVMs

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

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

Slide from Andrew Moore’s tutorial: http://www.autonlab.org/tutorials/svm.html

Recap: Pedestrian Detection with HoG and SVMs

• Map each grid cell in the input window to a histogram counting the gradients per orientation.
• Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

[Dalal & Triggs, CVPR 2005]

Limitations of Sliding Windows (continued)

• Not all objects are “box” shaped

Slide credit: Kristen Grauman

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

Figure credit: Derek Hoiem

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

Image credit: Adam, Birdy, & Shimosaka

K. Grauman, B. Leibe
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

by Diva Sian
by swashford

Harder Case

by Diva Sian
by scgbt

Harder Still?

NASA Mars Rover images

Answer Below (Look for tiny colored squares)

NASA Mars Rover images with SIFT feature matches
(Figure by Noah Snavely)
Application: Image Stitching

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

Procedure:

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

General Approach

Common Requirements

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

No chance to match!

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!

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.

Invariance: Geometric Transformations

• Often modeled as a linear transformation:
  - Scaling + Offset

Invariance: Photometric Transformations

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

Many Existing Detectors Available

- Hessian & Harris [Beaudet ‘78], [Harris ‘88]
- Laplacian, DoG [Lindeberg ‘98], [Lowe 1999]
- Harris-Hessian-Laplace [Mikolajczyk & Schmid ‘01]
- Harris-Hessian-Affine [Mikolajczyk & Schmid ‘04]
- EBR and IBR [Tuytelaars & Van Gool ‘04]
- MSER [Matas ‘02]
- Salient Regions [Kadir & Brady ‘01]
- Others…
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

Harris Detector Formulation

\[ E(u, v) = \sum_{x,y} w(x, y) \left[ \nabla I(x, y)^T \nabla I(x + u, y + v) - \nabla I(x, y)^T \nabla I(x, y) \right]^2 \]

Window function
Shifted intensity
Intensity

Window function \( w(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \) or Gaussian


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)

"flat" region: no change in all directions
"edge": no change along the edge direction
"corner": significant change in all directions

Harris Detector Formulation

- This measure of change can be approximated by:

\[ E(u, v) = [u \ v] M [u \ v]^T \]

where \( M \) is a 2x2 matrix computed from image derivatives:

\[ M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \]

Sum over image region - the area we are checking for corner

\[ M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \]

Harris Detector Formulation

- Change of intensity for the shift \([u,v]\):

\[ E(u, v) = \sum_{x,y} w(x, y) \left[ I(x+u, y+v) - I(x, y) \right]^2 \]

Window function
Shifted intensity
Intensity

Window function \( w(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \) or Gaussian


What Does This Matrix Reveal?

- First, let’s consider an axis-aligned corner:

\[
M = R^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R
\]

(Eigenvalue decomposition)

We can visualize \(M\) as an ellipse with axis lengths determined by the eigenvalues and orientation determined by \(R\).

General Case

- Since \(M\) is symmetric, we have

\[
M = R^T \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R
\]

- 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 \(M\):

\[
\lambda_1, \lambda_2
\]

- \(\lambda_1\) and \(\lambda_2\) are large, \(\lambda_1 - \lambda_2\) increases in all directions

Corner Response Function

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

- Fast approximation
  - Avoid computing the eigenvalues
  - \(\alpha\): constant (0.04 to 0.06)

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

\[
M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}
\]

- Option 1: uniform window
  - Sum over square window
  - Problem: not rotation invariant

- Option 2: Smooth with Gaussian
  - Gaussian already performs weighted sum
  - Result is rotation invariant

Slide credit: Kristen Grauman
Slide credit: David Jacobs
Slide credit: Kristen Grauman
Summary: Harris Detector [Harris88]

- Compute second moment matrix (autocorrelation matrix)
  \[ M(\sigma) = g(\sigma) \begin{bmatrix} \mu_x^2 & \mu_x \mu_y & \mu_x \\ \mu_x \mu_y & \mu_y^2 & \mu_y \\ \mu_x & \mu_y & 1 \end{bmatrix} \]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma) \)
4. Cornerness function - two strong eigenvalues
   \[ R = \text{det}(M(\sigma)) - \alpha \text{trace}(M(\sigma)) \]
   \[ = g(I_{x0}^2)g(I_{y0}^2) - [g(I_{x1}I_{y0})]^2 - \alpha g(I_{x0}^2) + g(I_{y0}^2) \]
5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk

Harris Detector: Workflow

- Compute corner responses \( R \)
- Take only the local maxima of \( R \), where \( R > \text{threshold} \).
- Resulting Harris points

Slide adapted from Darya Frolova, Denis Simakov

Harris Detector - Responses [Harris88]

Effect: A very precise corner detector.

Slide credit: Krystian Mikolajczyk
**Perceptual and Sensory Augmented Computing**

**Hessian Detector - Responses [Harris88]**

- Results are well suited for finding stereo correspondences

**Harris Detector: Properties**

- Rotation invariance?
- Scale invariance?

- Corner response \( R \) is invariant to image rotation
- Not invariant to image scale!

**Hessian Detector [Beaudet78]**

- 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

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

In Matlab:

\[
I_{xx} * I_{yy} - (I_{xy})^2
\]
**Hessian Detector - Responses** [Beaudet78]

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

---

**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

---

**From Points to Regions...**

- The Harris and Hessian operators define interest points.
  - Precise localization
  - High repeatability
- In order to compare those points, we need to compute a descriptor over a region.
  - How can we define such a region in a scale invariant manner?
  - *I.e. how can we detect scale invariant interest regions?*

---

**Naïve Approach: Exhaustive Search**

- Multi-scale procedure
  - Compare descriptors while varying the patch size

---

**Naïve Approach: Exhaustive Search**

- Multi-scale procedure
  - Compare descriptors while varying the patch size
Naïve Approach: Exhaustive Search
- Multi-scale procedure
  - Compare descriptors while varying the patch size

Automatic Scale Selection
- Solution:
  - Design a function on the region, which is “scale invariant” (the same for corresponding regions, even if they are at different scales)
  - For a point in one image, we can consider it as a function of region size (patch width)

**Example:** average intensity. For corresponding regions (even of different sizes) it will be the same.

**scale = \( \frac{1}{2} \)**

**Region size**

**Important:** this scale invariant region size is found in each image independently!
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit: Krystian Mikolajczyk
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector

Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

Characteristic Scale

- We define the characteristic scale as the scale that produces peak of Laplacian response

\[
\text{Laplacian-of-Gaussian (LoG)} = \text{“blob” detector}
\]
LoG Detector: Workflow

Technical Detail

- We can efficiently approximate the Laplacian with a difference of Gaussians:
  \[ L = \sigma^2 (G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma)) \]
  (Laplacian)
  \[ DoG = G(x,y,k\sigma) - G(x,y,\sigma) \]
  (Difference of Gaussians)

Difference-of-Gaussian (DoG)

- Difference of Gaussians as approximation of the LoG
  - This is used e.g. in Lowe's SIFT pipeline for feature detection.
- Advantages
  - No need to compute 2nd derivatives
  - Gaussians are computed anyway, e.g. in a Gaussian pyramid.

Key point localization with DoG

- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses

Candidate keypoints: list of \((x,y,\sigma)\)
**DoG - Efficient Computation**

- Computation in Gaussian scale pyramid

**Results: Lowe’s DoG**

**Example of Keypoint Detection**

(a) 233x189 image
(b) 832 DoG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures (removing edge responses)

**Harris-Laplace [Mikolajczyk '01]**

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian (same procedure with Hessian ⇒ Hessian-Laplace)

**Summary: Scale Invariant Detection**

- **Given:** Two images of the same scene with a large scale difference between them.
- **Goal:** Find the same interest points independently in each image.
- **Solution:** Search for maxima of suitable functions in scale and in space (over the image).

- **Two strategies**
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation
  - These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).
Topics of This Lecture

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

Local Descriptors

- We know how to detect points
- Next question: How to describe them for matching?
- Point descriptor should be: 1. Invariant 2. Distinctive

Feature Descriptors

- Disadvantage of patches as descriptors:
  - Small shifts can affect matching score a lot
- Solution: histograms

Feature Descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.

Orientation Normalization: Computation

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

Slide adapted from David Lowe

Summary: SIFT

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint up to ~60 deg. out-of-plane rotation
  - Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://groups.csail.mit.edu/tideway/tutorial/tutorial_sift_sift

Slide credit: Steve Seitz

Working with SIFT Descriptors

- One image yields:
  - 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch
    - \([n \times 128\) matrix\]
  - \(n\) scale parameters specifying the size of each patch
    - \([n \times 1\) vector\]
  - \(n\) orientation parameters specifying the angle of the patch
    - \([n \times 1\) vector\]
  - \(n\) 2D points giving positions of the patches
    - \([n \times 2\) matrix\]

Slide credit: Steve Seitz

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
  - http://www.vision.ee.ethz.ch/~surf
  - [Bay, ECCV’06], [Cornelis, CVGPU’08]
- GPU implementation available
  - Feature extraction @ 100Hz (detector + descriptor, 640x480 img)

You Can Try It At Home...

- 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

http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries
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](http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries)