Computer Vision - Lecture 22

Repetition

03.02.2015

Announcements

• Exam
  - 1st Date: Monday, 23.02., 13:30 - 17:30h
  - 2nd Date: Thursday, 26.03., 09:30 - 12:30h
  - Closed-book exam, the core exam time will be 2h.
  - Admission requirement: 50% of the exercise points or passed test exam
  - We will send around an announcement with the exact starting times and places by email.

• Test exam
  - Date: Thursday, 05.02., 09:15 - 10:45h, room UMIC 025
  - Core exam time will be 1h
  - Purpose: Prepare you for the questions you can expect.
  - Possibility to collect bonus exercise points!

Announcements (2)

• Feedback to the lecture evaluation

Announcements (3)

• Today, I’ll summarize the most important points from the lecture.
  - It is an opportunity for you to ask questions...
  - ...or get additional explanations about certain topics.
  - So, please do ask.

• Today’s slides are intended as an index for the lecture.
  - But they are not complete, won’t be sufficient as only tool.
  - Also look at the exercises - they often explain algorithms in detail.

Repetition

• Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction

• Segmentation & Grouping

• Object Recognition

• Local Features & Matching

• Object Categorization

• 3D Reconstruction

• Motion and Tracking

Recap: Pinhole Camera

• (Simple) standard and abstract model today
  - Box with a small hole in it
  - Works in practice

Source: Forsyth & Ponce
Recap: Focus and Depth of Field

- Depth of field: distance between image planes where blur is tolerable

Thin lens: scene points at distinct depths come in focus at different image planes.

(Real camera lens systems have greater depth of field.)

Source: Shapiro & Stockman

Recap: Field of View and Focal Length

- As f gets smaller, image becomes more wide angle
  - More world points project onto the finite image plane

- As f gets larger, image becomes more telescopic
  - Smaller part of the world projects onto the finite image plane

Recap: Color Sensing in Digital Cameras

Estimate missing components from neighboring values (demosaicing)

Bayer grid

Source: Steve Seitz

Recap: Binary Processing Pipeline

- Convert the image into binary form
  - Thresholding

- Clean up the thresholded image
  - Morphological operators

- Extract individual objects
  - Connected Components Labeling

- Describe the objects
  - Region properties

Recap: Robust Thresholding

- Ideal histogram, light object on dark background

- Actual observed histogram with noise

Assumption here: only two modes

Source: Robyn Owens
Recap: Global Binarization [Otsu’79]

- Precompute a cumulative grayvalue histogram \( h \).
- For each potential threshold \( T \)
  1.) Separate the pixels into two clusters according to \( T \).
  2.) Compute both cluster means \( \mu_1(T) \) and \( \mu_2(T) \).
    Look up \( n_1 \), \( n_2 \) in \( h \).
  3.) Compute the between-class variance \( \sigma_{between}(T) \)
    \[ \sigma_{between}(T) = n_1(T)n_2(T)(\mu_1(T) - \mu_2(T))^2 \]
- Choose the threshold that maximizes \( T^* = \arg\max_T \sigma_{between}(T) \).

Recap: Background Surface Fitting

- Document images often contain a smooth gradient
  \( \Rightarrow \) Try to fit that gradient with a polynomial function

Recap: Dilation

- Definition
  "The dilation of \( A \) by \( B \) is the set of all displacements \( z \), such that \( (B) \) and \( A \) overlap by at least one element".
  \( (B) \) is the mirrored version of \( B \), shifted by \( z \).
- Effects
  - If current pixel \( z \) is foreground, set all pixels under \( (B) \) to foreground.
  \( \Rightarrow \) Expand connected components
  \( \Rightarrow \) Grow features
  \( \Rightarrow \) Fill holes

Recap: Erosion

- Definition
  "The erosion of \( A \) by \( B \) is the set of all displacements \( z \), such that \( (B) \) is entirely contained in \( A \)".
- Effects
  - If not every pixel under \( (B) \) is foreground, set the current pixel \( z \) to background.
  \( \Rightarrow \) Erode connected components
  \( \Rightarrow \) Shrink features
  \( \Rightarrow \) Remove bridges, branches, noise

Recap: Opening

- Definition
  Sequence of Erosion and Dilation
  \[ A \ast B = (A \ominus B) \oplus B \]
- Effect
  - \( A \ast B \) is defined by the points that are reached if \( B \) is rolled around inside \( A \).
  \( \Rightarrow \) Remove small objects, keep original shape.

Recap: Closing

- Definition
  Sequence of Dilation and Erosion
  \[ A \ast B = (A \oplus B) \ominus B \]
- Effect
  - \( A \ast B \) is defined by the points that are reached if \( B \) is rolled around on the outside of \( A \).
  \( \Rightarrow \) Fill holes, keep original shape.
Recap: Connected Components Labeling

- Process the image from left to right, top to bottom:
  1. If the next pixel to process is 1
     i. If only one of its neighbors (top or left) is 1, copy its label.
     ii. If both are 1 and have the same label, copy it.
     iii. If they have different labels
          Update the equivalence table.
          Copy the label from the left.
     iv. Otherwise, assign a new label.

- Re-label with the smallest of equivalent labels

Recap: Region Properties

- From the previous steps, we can obtain separated objects.
- Some useful features can be extracted once we have connected components, including
  - Area
  - Centroid
  - Extremal points, bounding box
  - Circularity
  - Spatial moments

Recap: Moment Invariants

- Normalized central moments
  \[ \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} \quad y = \frac{p + q}{2} + 1 \]
- From those, a set of invariant moments can be defined for object description.
  \[ \phi_1 = \eta_{20} + \eta_{02} \]
  \[ \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \]
  \[ \phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{12} - \eta_{03})^2 \]
  \[ \phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \]
- Robust to translation, rotation & scaling, but don’t expect wonders (still summary statistics).

Recap: Effect of Filtering

- Noise introduces high frequencies. To remove them, we want to apply a “low-pass” filter.
- The ideal filter shape in the frequency domain would be a box. But this transfers to a spatial sinc, which has infinite spatial support.
- A compact spatial box filter transfers to a frequency sinc, which creates artifacts.
- A Gaussian has compact support in both domains. This makes it a convenient choice for a low-pass filter.

Recap: Gaussian Smoothing

- Gaussian kernel
  \[ G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \]
- Rotationally symmetric
- Weights nearby pixels more than distant ones
   This makes sense as ‘probabilistic’ inference about the signal
- A Gaussian gives a good model of a fuzzy blob
Recap: Smoothing with a Gaussian

- Parameter \( \sigma \) is the “scale” / “width” / “spread” of the Gaussian kernel and controls the amount of smoothing.

```matlab
for sigma=1:3:10
    h = fspecial('gaussian', fsize, sigma);
    out = imfilter(im, h);
    imshow(out);
    pause;
end
```

Recap: Resampling with Prior Smoothing

- Note: We cannot recover the high frequencies, but we can avoid artifacts by smoothing before resampling.

Recap: The Gaussian Pyramid

- Basic idea
  - Replace each pixel by the median of its neighbors.

- Properties
  - Doesn’t introduce new pixel values
  - Removes spikes: good for impulse, salt & pepper noise
  - Nonlinear
  - Edge preserving

Recap: Derivatives and Edges...

- 1st derivative
- 2nd derivative

Recap: 2D Edge Detection Filters

- \( \nabla^2 \) is the Laplacian operator:

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]
Recap: Canny Edge Detector
1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   • Thin multi-pixel wide “ridges” down to single pixel width
4. Linking and thresholding (hysteresis):
   • Define two thresholds: low and high
   • Use the high threshold to start edge curves and the low
     threshold to continue them
• MATLAB:
  >> edge(image,'canny')
  >> help edge
adapted from D. Lowe, L. Fei-Fei

Recap: Edges vs. Boundaries
Edges useful signal to indicate occluding boundaries, shape.
Here the raw edge output is not so bad...
...but quite often boundaries of interest are fragmented, and we have extra
“clutter” edge points.
Slide credit: Kristen Grauman

Recap: Chamfer Matching
• Chamfer Distance
  - Average distance to nearest feature
    \[
    D_{\text{chamfer}}(T, T') = \frac{1}{|T|} \sum_{t \in T} d_t(t)
    \]
  - This can be computed efficiently by correlating the edge
    template with the distance-transformed image

Recap: Fitting and Hough Transform
Given a model of interest, we can overcome some of
the missing and noisy edges using fitting tech-
niques.
With voting methods like
the Hough transform, detected points vote on
possible model parame-
ters.

Recap: Hough Transform
• How can we use this to find the most likely parameters
  \((m, b)\) for the most prominent line in the image space?
  - Let each edge point in image space vote for a set of possible
    parameters in Hough space
  - Accumulate votes in discrete set of bins; parameters with the
    most votes indicate line in image space.
Slide credit: Steve Seitz
Recap: Hough Transform, Polar Parametrization

- Usual \((m,b)\) parameter space problematic: can take on infinite values, undefined for vertical lines.
- Point in image space \(\Rightarrow\) sinusoid segment in Hough space.

\[ d : \text{perpendicular distance from line to origin} \]
\[ \theta : \text{angle the perpendicular makes with the x-axis} \]

\[ x \cos \theta - y \sin \theta = d \]

Recap: Hough Transform for Circles

- Circle: center \((a,b)\) and radius \(r\)
  \[ (x-a)^2 + (y-b)^2 = r^2 \]
- For an unknown radius \(r\), unknown gradient direction

Recap: Generalized Hough Transform

- What if want to detect arbitrary shapes defined by boundary points and a reference point?

  At each boundary point, compute displacement vector: \(r = a - p\).
  For a given model shape: store these vectors in a table indexed by gradient orientation \(\theta\).

Recap: Gestalt Theory

- Gestalt: whole or group
  - Whole is greater than sum of its parts
  - Relationships among parts can yield new properties/features
- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

  "I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

  Max Wertheimer (1880-1943)

Recap: Gestalt Factors

- These factors make intuitive sense, but are very difficult to translate into algorithms.
Recap: Image Segmentation

- Goal: identify groups of pixels that go together

Recap: K-Means Clustering

- Basic idea: randomly initialize the k cluster centers, and iterate between the two following steps
  1. Randomly initialize the cluster centers, $c_1, ..., c_k$
  2. Given cluster centers, determine points in each cluster
     - For each point $p$, find the closest $c_i$. Put $p$ into cluster $i$
  3. Given points in each cluster, solve for $c_i$
     - Set $c_i$ to be the mean of points in cluster $i$
  4. If $c_i$ have changed, repeat Step 2

- Properties
  - Will always converge to some solution
  - Can be a “local minimum”
    - Does not always find the global minimum of objective function:
      \[ \sum_{i=1}^{k} \sum_{p \in c_i} |p - c_i|^2 \]

Recap: Expectation Maximization (EM)

- Goal
  - Find blob parameters $\theta$ that maximize the likelihood function:
    \[ p(data|\theta) = \prod_{n=1}^{N} p(x_n|\theta) \]
- Approach:
  1. E-step: given current guess of blobs, compute ownership of each point
  2. M-step: given ownership probabilities, update blobs to maximize likelihood function
  3. Repeat until convergence

Recap: Mean-Shift Algorithm

- Iterative Mode Search
  1. Initialize random seed, and window $W$
  2. Calculate center of gravity (the “mean”) of $W$:
    \[ \sum_{x \in W} \mathbb{I}(x) \]
  3. Shift the search window to the mean
  4. Repeat Step 2 until convergence

Recap: Mean-Shift Clustering

- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode

Recap: Mean-Shift Segmentation

- Find features (color, gradients, texture, etc)
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode
Recap: MRFs for Image Segmentation

- MRF formulation
  
  \[ E(x, y) = \sum_i \phi_i(x_i, y_i) + \sum_{i,j} \psi_{ij}(x_i, x_j) \]

  \( \phi(x_i, y_i) \): Unary potentials
  
  \( \psi_{ij}(x_i, x_j) \): Pairwise potentials

- Minimize the energy
  
  \[ \Rightarrow \text{Minimize the energy} \]

  \[ E(x, y) = \sum_i \phi_i(x_i, y_i) + \sum_{i,j} \psi_{ij}(x_i, x_j) \]

Recap: How to Set the Potentials?

- Unary potentials
  
  \[ \phi(x_i, y_i) \] = \text{Learn color distributions for each label}

  \[ \phi(x_i = 1, y_i) \]

  \[ \phi(x_i = 0, y_i) \]

Recap: Graph-Cuts Energy Minimization

- Solve an equivalent graph cut problem
  
  1. Introduce extra nodes: source and sink
  2. Weight connections to source/sink (\( t \)-links) by \( \phi(x_i = s) \) and \( \phi(x_i = t) \), respectively.
  3. Weight connections between nodes (\( n \)-links) by \( \psi(x_i, x_j) \).
  4. Find the minimum cost cut that separates source from sink.

  \[ \Rightarrow \text{Solution is equivalent to minimum of the energy.} \]

  \[ t \]-t Mincut can be solved efficiently

  - Dual to the well-known max flow problem
  - Very efficient algorithms available for regular grid graphs (1-2 MPixels/s)
  - Globally optimal result for 2-class problems

Recap: How to Set the Potentials?

- Pairwise potentials
  
  - Potts Model
    
    \[ \psi(x_i, x_j; \theta_{ij}) = \theta_{ij} \delta(x_i, x_j) \]

    simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.

    - Extension: "Contrast sensitive Potts model"
      
      \[ \psi(x_i, x_j, g_{ij}(y); \theta_{ij}) = \theta_{ij} g_{ij}(y) \delta(x_i, x_j) \]

      where
      
      \[ g_{ij}(y) = e^{-\beta |y_1 - y_2|^2} \]

    \[ \Rightarrow \text{Discourages label changes except in places where there is also a large change in the observations.} \]
Recap: When Can s-t Graph Cuts Be Applied?

- s-t graph cuts can only globally minimize binary energies that are submodular.

\[ E(L) = \sum_{t \in t-links} E_p(L_p) + \sum_{n \in n-links} E(L_p, L_q) \]

- Submodularity is the discrete equivalent to convexity.
  \[ \Rightarrow \text{Every local energy minimum is a global minimum.} \]
  \[ \Rightarrow \text{Solution will be globally optimal.} \]

Recap: Appearance-Based Recognition

- Basic assumption
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

\[ \Rightarrow \text{Fundamental paradigm shift in the 90's} \]

Recap: Comparison Measures

- Vector space interpretation
  - Euclidean distance

- Statistical motivation
  - Chi-square
  - Bhattacharyya

- Information-theoretic motivation
  - Kullback-Leibler divergence, Jeffrey's divergence

- Histogram motivation
  - Histogram intersection

- Ground distance
  - Earth Movers Distance (EMD)

Recap: Recognition Using Global Features

- E.g. histogram comparison

- "Nearest-Neighbor" strategy

Recap: Recognition Using Histograms

- Simple algorithm
  1. Build a set of histograms \( H = \{ h_i \} \) for each known object
  2. Build a histogram \( h_t \) for the test image.
  3. Compare \( h_t \) to each \( h_i \in H \)
  4. Select the object with the best matching score

\[ \Rightarrow \text{"Nearest-Neighbor" strategy} \]

Repetition

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
  - Global Representations
  - Subspace Representations
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking
Recap: Multidimensional Representations

- Combination of several descriptors
  - Each descriptor is applied to the whole image.
  - Corresponding pixel values are combined into one feature vector.
  - Feature vectors are collected in multidimensional histogram.

\[ D_1 \quad D_2 \quad \text{Lap} \]

Recap: Colored Derivatives

- Generalization: derivatives along
  - \( Y \) axis \( \rightarrow \) intensity differences
  - \( C_1 \) axis \( \rightarrow \) red-green differences
  - \( C_2 \) axis \( \rightarrow \) blue-yellow differences

- Application: Brand identification in video

First Applications Take Up Shape...

- Binary Segmentation
- Moment descriptors
- Skin color detection
- Line detection
- Circle detection
- Histogram based recognition
- Histogram based recognition

Recap: Subspace Methods

- Subspace methods
  - Reconstructive: PCA, ICA, NMF
  - Discriminative: LDA, SVM, CCA

Recap: Obj. Detection by Distance TO Eigenspace

- For each test image, compute the reprojection error
  - An \( n \)-pixel image \( x \in \mathbb{R}^n \) can be projected to the low-dimensional feature space \( y \in \mathbb{R}^m \) by \( y = Ux \).
  - From \( y \in \mathbb{R}^m \), the reconstruction of the point is \( U^Ty \).
  - The error of the reconstruction is \( \|x - U^Ty\| \).

- Accept a detection if this error is low.
  - Assumption: subspace is optimized to the target object (class).
  - Other classes are not represented well \( \Rightarrow \) large error.
Recap: Obj Identification by Distance IN Eigenspace

- Objects are represented as coordinates in an $n$-dim. eigenspace.
- Example:
  - 3D space with points representing individual objects or a manifold representing parametric eigenspace (e.g., orientation, pose, illumination).
- Estimate parameters by finding the NN in the eigenspace.

Recap: Eigenfaces

Recap: Restrictions of PCA

- PCA minimizes projection error
- PCA is “unsupervised” no information on classes is used
- Discriminating information might be lost

Recap: Local Feature Matching Pipeline

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

Recap: Requirements for Local Features

- 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!
Recap: Harris Detector \([\text{Harris88}]\)

- Compute second moment matrix (autocorrelation matrix)
  \[
  M(\sigma_x, \sigma_y) = g(\sigma_x) I_x(\sigma_y) \quad I_y(\sigma_y)
  \]
  1. Image derivatives
  2. Square of derivatives
  3. Gaussian filter \(g(\sigma)\)

4. Cornerness function - two strong eigenvalues
\[
R = \det(M(\sigma_x, \sigma_y)) - \alpha \text{trace}(M(\sigma_x, \sigma_y)) = g(I_x^2) g(I_y^2) - \alpha [g(I_x I_y)]^2 + g(I_x) + g(I_y)
\]

5. Perform non-maximum suppression

Recap: Harris Detector Responses \([\text{Harris88}]\)

Effect: A very precise corner detector.

Recap: Hessian Detector \([\text{Beaudet78}]\)

- Hessian determinant
\[
\text{Hessian}(I) = \begin{bmatrix}
  I_{xx} & I_{xy} \\
  I_{xy} & I_{yy}
\end{bmatrix}
\]
\[
\det(\text{Hessian}(I)) = I_{xx} I_{yy} - I_{xy}^2
\]
In Matlab:
\[
I_{xx} * I_{yy} - (I_{xy})^2
\]

Recap: Hessian Detector Responses \([\text{Beaudet78}]\)

Effect: Responses mainly on corners and strongly textured areas.

Recap: Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Recap: Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian
\[
L_\sigma(x, \sigma) + L_{\sigma'}(x, \sigma') > 0
\]
\[
\Rightarrow \text{List of } (x, y, \sigma)
\]
Recap: LoG Detector Responses

Efficient implementation
- Approximate LoG with a difference of Gaussians (DoG)

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

Candidate keypoints: list of \((x,y,\sigma)\)

Recap: Key point localization with DoG

Recap: Harris-Laplace [Mikolajczyk '01]

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

Harris points

Recap: SIFT Feature Descriptor

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: \(4 \times 4 \times 8 = 128\) dimensions

Recap: Recognition with Local Features

Image content is transformed into local features that are invariant to translation, rotation, and scale
- Goal: Verify if they belong to a consistent configuration
Recap: Fitting an Affine Transformation
- Assuming we know the correspondences, how do we get the transformation?

\[
\begin{bmatrix}
(x', y')_1 \\
(x', y')_2 \\
\vdots
\end{bmatrix} =
\begin{bmatrix}
(x, y)_1 \\
(x, y)_2 \\
\vdots
\end{bmatrix}
+ \begin{bmatrix}
m_1 & m_2 & m_3 & m_4 \\
m_5 & m_6 & m_7 & m_8 \\
m_9 & m_{10} & m_{11} & m_{12}
\end{bmatrix}
\begin{bmatrix}
t_1 \\
t_2 \\
\vdots
\end{bmatrix}
\]

Image coordinates

Homogenous coordinates

Matrix notation

Recap: Fitting a Homography
- Estimating the transformation

Solution:
- Null-space vector of \( A \)
- Corresponds to smallest eigenvector

\[
Ah = 0
\]

SVD

Minimizes least square error
Recap: RANSAC

RANSAC loop:
1. Randomly select a seed group of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find inliers to this transformation
4. If the number of inliers is sufficiently large, recompute least-squares estimate of transformation on all of the inliers
   • Keep the transformation with the largest number of inliers

Recap: RANSAC Line Fitting Example

• Task: Estimate the best line

Sample two points

Fit a line to them

Total number of points within a threshold of line.

Repeat, until we get a good result.
Recap: Feature Matching Example

• Find best stereo match within a square search window (here 300 pixels²)
• Global transformation model: epipolar geometry

before RANSAC  
after RANSAC

Recap: Generalized Hough Transform

• Suppose our features are scale- and rotation-invariant
  
  • Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
  
  • Of course, a hypothesis from a single match is unreliable.
  
  • Solution: let each match vote for its hypothesis in a Hough space with very coarse bins.

Application: Panorama Stitching

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

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: Classifier Construction: Many Choices...

Nearest Neighbor

Berg, Berg, Malik 2005, Chum, Zisserman 2007, Boiman, Shechtman, Irani 2008, ...

Neural networks


Boosting

Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006, Benenson 2012, ...

Support Vector Machines


Randomized Forests

Amit, Geman 1997, Breiman 2001, Lepeit, Fua 2006, Gall, Lempitsky 2009, ...

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: Non-Linear SVMs

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

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

Recap: HOG Descriptor Processing Chain

• SVM Classification
   Typically using a linear SVM

Object/Non-object

Linear SVM

Collect HOGs over detection window

Contrast normalize over overlapping spatial cells

Weighted vote in spatial & orientation cells

Compute gradients

Gamma compression

Image Window

Recap: HOG Cell Computation Details

• Gradient orientation voting
   Each pixel contributes to localized gradient orientation histogram(s)
   Vote is weighted by the pixel’s gradient magnitude

\[ \theta = \tan^{-1} \left( \frac{\partial I}{\partial x} / \frac{\partial I}{\partial y} \right) \]

\[ \| \nabla f \| = \sqrt{\left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2} \]

• Block-level Gaussian weighting
   An additional Gaussian weight is applied to each 2x2 block of cells
   Each cell is part of 4 such blocks, resulting in 4 versions of the histogram.
Recap: HOG Cell Computation Details (2)

- Important for robustness: Tri-linear interpolation
  - Each pixel contributes to (up to) 4 neighboring cell histograms
    - Weights are obtained by bilinear interpolation in image space:
      \[
      \begin{align*}
      R(x_i, y_i) &= w_i \left( 1 - \frac{x_i - x_k}{x_k - x_{k+1}} \right) \left( 1 - \frac{y_i - y_k}{y_k - y_{k+1}} \right) \\
      R(x_i, y_i) &= w_i \left( \frac{x_i - x_k}{x_k - x_{k+1}} \right) \left( 1 - \frac{y_i - y_k}{y_k - y_{k+1}} \right) \\
      R(x_i, y_i) &= w_i \left( 1 - \frac{x_i - x_k}{x_k - x_{k+1}} \right) \left( \frac{y_i - y_k}{y_k - y_{k+1}} \right) \\
      R(x_i, y_i) &= w_i \left( \frac{x_i - x_k}{x_k - x_{k+1}} \right) \left( \frac{y_i - y_k}{y_k - y_{k+1}} \right)
      \end{align*}
      \]
  - Contribution is further split over (up to) 2 neighboring orientation bins via linear interpolation over angles.

Recap: Non-Maximum Suppression

Recap: AdaBoost

Final classifier is combination of the weak classifiers

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 \(\rightarrow\) scale features directly for same cost

Application: Viola-Jones Face Detector

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/
Repetition

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

Recap: Identification vs. Categorization

- Find this particular object
- Recognize ANY car
- Recognize ANY cow

Recap: Visual Words

- Quantize the feature space into "visual words"
- Perform matching only to those visual words.

Recap: Bag-of-Word Representations (BoW)

- Object
- Bag of "words"

Recap: Categorization with Bags-of-Words

- Compute the word activation histogram for each image.
- Let each such BoW histogram be a feature vector.
- Use images from each class to train a classifier (e.g., an SVM).

Recap: Advantage of BoW Histograms

- Bag of words representations make it possible to describe the unordered point set with a single vector (of fixed dimension across image examples).
- Provides easy way to use distribution of feature types with various learning algorithms requiring vector input.
Limitations of BoW Representations

- The bag of words removes spatial layout.
- This is both a strength and a weakness.
- Why a strength?
- Why a weakness?

Recap: Part-Based Models

- Fischler & Elschlager 1973
- Model has two components
  - parts
  - structure
- (2D image fragments)
- (configuration of parts)

Recap: Implicit Shape Model - Representation

- Learn appearance codebook
  - Extract local features at interest points
  - Clustering \(\rightarrow\) appearance codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object

Recap: Implicit Shape Model - Recognition

Interest Points

Backprojected Hypotheses

Appearance codebook

3D Voting Space (continuous)

“Generalized Hough Transform with backprojection”

Spatial occurrence distributions

+ local figure-ground labels

Recap: Scale Invariant Voting

- Scale-invariant feature selection
  - Scale-invariant interest points
  - Rescale extracted patches
  - Match to constant-size codebook
- Generate scale votes
  - Scale as 3rd dimension in voting space
    \[
    \begin{align*}
    x_{scale} &= x_{img} - x_{wcc}(s_{img}/s_{wcc}) \\
    y_{scale} &= y_{img} - y_{wcc}(s_{img}/s_{wcc}) \\
    s_{scale} &= (s_{img}/s_{wcc})
    \end{align*}
    \]
  - Search for maxima in 3D voting space

Replication

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
  - Epipolar Geometry and Stereo Basics
  - Camera Calibration & Uncalibrated Reconstruction
  - Structure-from-Motion
- Motion and Tracking
Recap: What Is Stereo Vision?

• Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape.

Recap: Depth with Stereo - Basic Idea

• Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - Camera pose (calibration)
    - Point correspondence

Recap: Epipolar Geometry

• Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view.

• Epipolar constraint:
  - Correspondence for point \( p \) in \( \Pi \) must lie on the epipolar line \( l' \) in \( \Pi' \) (and vice versa).
  - Reduces correspondence problem to 1D search along conjugate epipolar lines.

Recap: Stereo Geometry With Calibrated Cameras

• Camera-centered coordinate systems are related by known rotation \( R \) and translation \( T \):
  \[
  X' = RX + T
  \]

Recap: Essential Matrix

\[
X' \cdot (T \times RX) = 0
\]
\[
X' \cdot (T \cdot RX) = 0
\]

Let \( E = T \cdot R \)

\[
X'^T E X = 0
\]

• This holds for the rays \( p \) and \( p' \) that are parallel to the camera-centered position vectors \( X \) and \( X' \), so we have:
  \[
p'^T E p = 0
  \]

• \( E \) is called the essential matrix, which relates corresponding image points [Longuet-Higgins 1981]

Recap: Essential Matrix and Epipolar Lines

Epipolar constraint: if we observe point \( p \) in one image, then its position \( p' \) in second image must satisfy this equation.

\[
l' = Ep
\]
(i.e., the line is given by: \( l' \times x = 0 \))

\[
l = E'^T p'
\]

\( l' \) is the coordinate vector representing the epipolar line for point \( p' \).
In practice, it is convenient if image scanlines are the epipolar lines.

Algorithm
- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection

This corresponds to the solution of $Ax = 0$ is the nullspace vector of $A$. This corresponds to the smallest singular vector of $A$.

For each pixel in the first image
- Find corresponding epipolar line in the right image
- Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
- Triangulate the matches to get depth information

This is easiest when epipolar lines are scanlines

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.
Recap: Calibrating a Camera

Goal
- Compute intrinsic and extrinsic parameters using observed camera data.

Main idea
- Place "calibration object" with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate \( P = P_{\text{in}} P_{\text{ext}} \)

This approach nicely generalizes to multiple cameras.

Recap: Camera Calibration (DLT Algorithm)

\[
\begin{bmatrix}
0^T & X_i^T & -y_i X_i^T \\
X_i^T & 0^T & -x_i X_i^T \\
\vdots & \vdots & \vdots \\
0^T & X_n^T & -y_n X_n^T \\
X_n^T & 0^T & -x_n X_n^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
\vdots \\
P_n
\end{bmatrix} = 0 \quad \text{Ap} = 0
\]

- \( P \) has 11 degrees of freedom.
- Two linearly independent equations per independent 2D/3D correspondence.
- Solve with SVD (similar to homography estimation)
  - Solution corresponds to smallest singular vector.
- 5 \( \frac{1}{2} \) correspondences needed for a minimal solution.

Recap: Epipolar Geometry

Camera matrix: \([I|0]\)
\( X = (u, v, w)^T \)
\( x = (u, v)^\prime \) system has coordinates \( Rx^\prime \) in the first one.

The vectors \( x_1, x_2, \text{ and } Rx^\prime \) are coplanar

Recap: Epipolar Geometry - Calibrated Case

- \( E x' \) is the epipolar line associated with \( x' \) (\( l = E x' \))
- \( E' x' \) is the epipolar line associated with \( x \) (\( l' = E' x \))
- \( E e' = 0 \) and \( E' e = 0 \)
- \( E \) is singular (rank two)
- \( E \) has five degrees of freedom (up to scale)

Essential Matrix (Longuet-Higgins, 1981)
The calibration matrices $K$ and $K'$ of the two cameras are unknown. We can write the epipolar constraint in terms of unknown normalized coordinates:

$$x^T E x' = 0 \quad x = K \hat{x}, \quad x' = K' \hat{x}'$$

$F x'$ is the epipolar line associated with $x'$ ($l = F x'$)

$F^T x$ is the epipolar line associated with $x$ ($l' = F^T x$)

$F e' = 0$ and $F e = 0$

$F$ is singular (rank two)

$F$ has seven degrees of freedom

Recap: Epipolar Geometry - Uncalibrated Case

The calibration matrices $K$ and $K'$ of the two cameras are unknown.

We can write the epipolar constraint in terms of unknown normalized coordinates:

$$x^T E x' = 0 \quad x = K \hat{x}, \quad x' = K' \hat{x}'$$

$F x'$ is the epipolar line associated with $x'$ ($l = F x'$)

$F^T x$ is the epipolar line associated with $x$ ($l' = F^T x$)

$F e' = 0$ and $F e = 0$

$F$ is singular (rank two)

$F$ has seven degrees of freedom

Recap: Epipolar Geometry - Uncalibrated Case

Recap: The Eight-Point Algorithm

$x = (u, v, 1)^T, x' = (u', v', 1)^T$

$$(w, w', 1) = 0$$

$$(a, a', b, b', c, c') = 0$$

1.) Solve with SVD. This minimizes

$$\sum (x_i^T F x_i)^2$$

2.) Enforce rank-2 constraint using SVD

$F$ is seven degrees of freedom.

Recap: The Eight-Point Algorithm

Recap: Normalized Eight-Point Alg.

1. Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels.

2. Use the eight-point algorithm to compute $F$ from the normalized points.

3. Enforce the rank-2 constraint using SVD.

$$F = U D V^T = U \begin{bmatrix} d_{11} & \cdots & d_{18} \\ \vdots & \ddots & \vdots \\ d_{18} & \cdots & d_{36} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{18} \\ \vdots & \ddots & \vdots \\ v_{18} & \cdots & v_{36} \end{bmatrix}$$

Set $d_{12}$ to zero and reconstruct $F$

4. Transform fundamental matrix back to original units: if $T$ and $T'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $T^T F T'$.

Recap: Normalized Eight-Point Alg.

Recap: Comparison of Estimation Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>B-point</th>
<th>Normalized B-point</th>
<th>Nonlinear least squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. Dist. 1</td>
<td>2.29 pix</td>
<td>0.92 pix</td>
<td>0.86 pix</td>
</tr>
<tr>
<td>Av. Dist. 2</td>
<td>2.18 pix</td>
<td>0.85 pix</td>
<td>0.80 pix</td>
</tr>
</tbody>
</table>

Perceptual and Sensory Augmented Computing
Recap: Epipolar Transfer
- Assume the epipolar geometry is known
- Given projections of the same point in two images, how can we compute the projection of that point in a third image?

\[ l_{1j} = F_{13} x_j \]
\[ l_{2j} = F_{23} x_j \]

Applications: 3D Reconstruction
- Given: \( m \) images of \( n \) fixed 3D points \( x_i = P_i X_j \), \( i = 1, ..., m \), \( j = 1, ..., n \)
- Problem: estimate \( m \) projection matrices \( P_i \) and \( n \) 3D points \( X_j \) from the \( mn \) correspondences \( x_{ij} \)

Recap: Structure from Motion Ambiguity
- If we scale the entire scene by some factor \( k \) and, at the same time, scale the camera matrices by the factor of \( 1/k \), the projections of the scene points in the image remain exactly the same.
- More generally: if we transform the scene using a transformation \( Q \) and apply the inverse transformation to the camera matrices, then the images do not change

\[ x = PX = (PQ^{-1})QX \]

Recap: Hierarchy of 3D Transformations
- Projective 15dof: Preserves intersection and tangency
- Affine 12dof: Preserves parallelism, volume ratios
- Similarity 7dof: Preserves angles, ratios of length
- Euclidean 6dof: Preserves angles, lengths

- With no constraints on the camera calibration matrix or on the scene, we get a projective reconstruction.
- Need additional information to upgrade the reconstruction to affine, similarity, or Euclidean.
Recap: Affine Structure from Motion

- Let’s create a $2m \times n$ data (measurement) matrix:
  \[
  \begin{bmatrix}
  \hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n} \\
  \hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  \hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn}
  \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}
  \]

  Cameras (2m \times 3)

- The measurement matrix $D = MS$ must have rank 3!


Slide credit: Svetlana Lazebnik

Recap: Affine Factorization

- Obtaining a factorization from SVD:

  \[
  D = U_3 \times \begin{bmatrix} W_1 & W_2 \end{bmatrix} \times V_3^T
  \]

  Possible decomposition:

  \[
  M = U_2 W_2 \times V_3 \quad S = W_1 \times V_3^T
  \]

  This decomposition minimizes $\|D-MS\|^2$

Slide credit: Martial Hebert

Recap: Projective Factorization

- If we knew the depths $z$, we could factorize $D$ to estimate $M$ and $S$.
- If we knew $M$ and $S$, we could solve for $z$.
- Solution: iterative approach (alternate between above two steps).

  \[
  D = MS \text{ has rank 4}
  \]

Recap: Sequential Projective SfM

- Initialize motion from two images SfM
- Initialize structure
- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image - *calibration*
  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera - *triangulation*
- Refine structure and motion: *bundle adjustment*

Slide credit: Svetlana Lazebnik

Recap: Estimating the Euclidean Upgrade

- Goal: Estimate ambiguity matrix $C$.
  - Orthographic assumption:
    \[
    a_i, a_2 = 0 \\
    a_i^T, a_2 = 0 \\
    a_i^T C a_2 = 0 \\
    a_i^T C a_3 = 1, \quad i = 1, \ldots, m
    \]

  This can be converted into a system of 3m equations:

  \[
  \begin{align*}
  \tilde{a}_1 \cdot \tilde{a}_2 &= 0 \\
  \tilde{a}_1 \cdot \tilde{a}_3 &= 0 \\
  \tilde{a}_2 \cdot \tilde{a}_3 &= 0 \\
  \end{align*}
  \]

  with $L = CC^T$

  this translates to

  $A_i A_i^T = I$

Slide adapted from S. Lazebnik, M. Hebert

Recap: Bundle Adjustment

- Non-linear method for refining structure and motion
- Minimizing mean-square reprojection error

  \[
  E(P, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, P X_j)
  \]

Slide credit: Svetlana Lazebnik
Recap: Estimating Optical Flow

- Given two subsequent frames, estimate the apparent motion field \( u(x, y) \) and \( v(x, y) \) between them.
- Key assumptions
  - Brightness constancy: projection of the same point looks the same in every frame.
  - Small motion: points do not move very far.
  - Spatial coherence: points move like their neighbors.

Recap: Lucas-Kanade Optical Flow

- Use all pixels in a \( K \times K \) window to get more equations.
- Least squares problem:
  \[
  \begin{bmatrix}
  I_x(p_1) & I_y(p_1) \\
  I_x(p_2) & I_y(p_2) \\
  \vdots & \vdots \\
  I_x(p_{25}) & I_y(p_{25})
  \end{bmatrix}
  \begin{bmatrix}
  u \\
  v
  \end{bmatrix}
  =
  \begin{bmatrix}
  I_x(p_1) \\
  I_x(p_2) \\
  \vdots \\
  I_x(p_{25})
  \end{bmatrix}
  \]
- Minimum least squares solution given by solution of
  \[
  (A^T A) \begin{bmatrix}
  d
  \end{bmatrix}
  = A^T b
  \]

Recap: Iterative Refinement

- Estimate velocity at each pixel using one iteration of LK estimation.
- Warp one image toward the other using the estimated flow field.
- Refine estimate by repeating the process.
- Iterative procedure
  - Results in subpixel accurate localization.
  - Converges for small displacements.

Recap: Coarse-to-fine Estimation

- Run iterative L-K
- Warp and upsample
Any Questions?

So what can you do with all of this?

Robust Object Detection & Tracking

Articulated Multi-Person Tracking

- Multi-Person tracking
  - Recover trajectories and solve data association
- Articulated Tracking
  - Estimate detailed body pose for each tracked person

Semantic 2D-3D Scene Segmentation

Integrated 3D Point Cloud Labels

Mining the World’s Images...
Tourist Guide Scenario

- Simply point the camera to any object/building of interest.
- Images are transmitted to a central server for recognition.
- Object-specific information is sent back to be displayed on the mobile phone.
- Mobile Augmented Reality fusion of graphics with real video.

Efficient Large-Scale Localization

3D Model, reconstructed from photos

Camera position

Query image