Computer Vision - Lecture 20

Repetition

31.01.2012

Bastian Leibe
RWTH Aachen
http://www.mmp.rwth-aachen.de

leibe@umic.rwth-aachen.de
Announcements

- Talk by Philippe Dreuw (Bosch Corporate Research)
  - “Research and Advance Engineering at Bosch - The Competence Center Computer Vision Systems”
  - 15.2., 14h in UMIC 025
Announcements (2)

• International Computer Vision Summer School (Sicily)
  - “Recognition, Registration, and Reconstruction in Images and Video”
  - Sicily, 15-21 July
  - Open to Master students, PhD students, young researchers
  - [http://svg.dmi.unict.it/icvss2012/](http://svg.dmi.unict.it/icvss2012/)

• If you’re interested...
  - Application deadline: (before) March 21 2012.
  - You will need a reference letter.
  - I can provide such a letter for a small number of interested people (based on the exam results).
  - You can apply to RWTH’s “Undergraduate Fund” program to get (partial) coverage of expenses if you’re accepted.
Announcements (Final)

• 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.

• Formal exam procedure
  ➢ Oral exam, form depends on M.Sc./Diplom specifics
  ➢ Procedure: 3 (or 4) questions, will have to answer 2 (3) of them
  ➢ For V4 exam, supplementary material also required.
Repetition

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

• 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

Bayer grid

Estimate missing components from neighboring values (demosaicing)

Source: Steve Seitz
Repetition

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

- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking
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

Image Source: D. Kim et al., Cytometry 35(1), 1999
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 \( h \)
      \[
      n_1(T) = \left| \{ I_{(x,y)} < T \} \right|, \quad n_2(T) = \left| \{ I_{(x,y)} \geq T \} \right|
      \]
  3.) Compute the between-class variance \( \sigma_{\text{between}}^2(T) \)
      \[
      \sigma_{\text{between}}^2(T) = n_1(T)n_2(T) \left[ \mu_1(T) - \mu_2(T) \right]^2
      \]
- Choose the threshold that maximizes
  \[
  T^* = \arg \max_T [\sigma_{\text{between}}^2(T)]
  \]
Recap: Background Surface Fitting

- Document images often contain a smooth gradient

⇒ 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 $(\hat{B})_z$ and $A$ overlap by at least one element”.
  - $(\hat{B})_z$ is the mirrored version of $B$, shifted by $z$)

- **Effects**
  - If current pixel $z$ is foreground, set all pixels under $(B)_z$ 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 $(B)_z$ is entirely contained in $A$.”

- **Effects**
  - If not every pixel under $(B)_z$ is foreground, set the current pixel $z$ to background.
  - Erode connected components
  - Shrink features
  - Remove bridges, branches, noise
Recap: Opening

• Definition
  - Sequence of Erosion and Dilation
    \[ A \circ B = (A \ominus B) \oplus B \]

• Effect
  - \( A \circ B \) is defined by the points that are reached if \( B \) is rolled around inside \( A \).
  - \( \Rightarrow \) Remove small objects, keep original shape.

Image Source: R.C. Gonzales & R.E. Woods
Recap: Closing

• Definition
  ➢ Sequence of Dilation and Erosion
    \[ A \cdot B = (A \oplus B) \ominus B \]

• Effect
  ➢ \( A \cdot B \) is defined by the points that are reached if \( B \) is rolled around on the outside of \( A \).
  ➢ Fill holes, keep original shape.

Image Source: R.C. Gonzales & R.E. Woods
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
           – Copy the label from the left.
           – Update the equivalence table.
     iv.) Otherwise, assign a new label.

- Re-label with the smallest of equivalent labels

Slide credit: J. Neira
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

\[ \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad \gamma = \frac{p + q}{2} + 1 \]

• From those, a set of *invariant moments* can be defined for object description.

\[
\begin{align*}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2
\end{align*}
\]

(Additional invariant moments \( \phi_5, \phi_6, \phi_7 \) can be found in the literature).

• Robust to translation, rotation & scaling, but don’t expect wonders (still summary statistics).
Repetition

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

• Segmentation & Grouping

• Object Recognition

• Local Features & Matching

• Object Categorization

• 3D Reconstruction

• Motion and Tracking
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
```

Slide credit: Kristen Grauman
Recap: Resampling with Prior Smoothing

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

Image Source: Forsyth & Ponce
Recap: The Gaussian Pyramid

\[
\begin{align*}
G_4 &= (G_3 \ast \text{gaussian}) \downarrow 2 \\
G_3 &= (G_2 \ast \text{gaussian}) \downarrow 2 \\
G_2 &= (G_1 \ast \text{gaussian}) \downarrow 2 \\
G_1 &= (G_0 \ast \text{gaussian}) \downarrow 2 \\
G_0 &= \text{Image}
\end{align*}
\]

Source: Irani & Basri
Recap: Median Filter

• 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

"zero crossings" of second derivative
Recap: 2D Edge Detection Filters

- \( \nabla^2 \) is the Laplacian operator:
  \[
  \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
  \]

Gaussian
\[
h_\sigma(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}
\]

Derivative of Gaussian
\[
\frac{\partial}{\partial x} h_\sigma(u, v)
\]

Laplacian of Gaussian
\[
\nabla^2 h_\sigma(u, v)
\]

Slide credit: Kristen Grauman
Repetition

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

• Segmentation & Grouping

• Object Recognition

• Local Features & Matching

• Object Categorization

• 3D Reconstruction

• Motion and Tracking

- Canny edge detector
- Chamfer matching
- Hough transform for lines
- Hough transform for circles
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

\[ D_{\text{chamfer}}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t) \]

- This can be computed efficiently by correlating the edge template with the distance-transformed image

[D. Gavrila, DAGM’99]
Recap: Fitting and Hough Transform

Given a model of interest, we can overcome some of the missing and noisy edges using fitting techniques.

With voting methods like the Hough transform, detected points vote on possible model parameters.
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.
Recap: Hough Transf. Polar Parametrization

- **Usual** \((m,b)\) **parameter space** problematic: can take on infinite values, undefined for vertical lines.

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

- **Point in image space** \(\Rightarrow\) **sinusoid segment in Hough space**

\(d\) : perpendicular distance from line to origin

\(\theta\) : angle the perpendicular makes with the \(x\)-axis

Slide credit: Steve Seitz
Recap: Hough Transform for Circles

- **Circle**: center \((a, b)\) and radius \(r\)
  \[
  (x_i - a)^2 + (y_i - b)^2 = r^2
  \]

- For an unknown radius \(r\), unknown gradient direction

Slide credit: Kristen Grauman
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: \( r = a - p_i \).

For a given model shape: store these vectors in a table indexed by gradient orientation \( \theta \).

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]

[Image showing the Hough transform process with boundary points and displacement vectors.
Slide credit: Kristen Grauman]
Repetition

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

- Segmentation & Grouping

- Object Recognition

- Local Features & Matching

- Object Categorization

- 3D Reconstruction

- Motion and Tracking
Recap: Color Sensing

- Electromagnetic spectrum

Slide credit: Svetlana Lazebnik
Recap: Color Perception

- Rods and cones act as filters on the spectrum
  - To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths
    - Each cone yields one number
  - Q: How can we represent an entire spectrum with 3 numbers?
  - A: We can’t! Most of the information is lost.
    - As a result, two different spectra may appear indistinguishable.
    - Such spectra are known as *metamers*.
Recap: RGB Color Space

- Primaries are monochromatic lights (for monitors, they correspond to the three types of phosphors)
- *Subtractive matching* required for some wavelengths

![RGB Color Space Diagram](image)

**RGB matching functions**

- $p_1 = 645.2$ nm
- $p_2 = 525.3$ nm
- $p_3 = 444.4$ nm

- Good for devices, but not for perception...
Recap: CIE XYZ Color Space

- Established in 1931 by the International Commission on Illumination
- Primaries are imaginary, but matching functions are everywhere positive
- 2D visualization: draw \((x,y)\), where 
  \[ x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z} \]

Matching functions

Slide credit: Svetlana Lazebnik
Recap: HSV Color Space

- **Hue, Saturation, Value** (Brightness)
- Nonlinear - reflects topology of colors by coding hue as an angle.
- Matlab: `hsv2rgb`, `rgb2hsv`.

Image from mathworks.com
Recap: Color as Low-Level Cue

for recognition


for content-based image retrieval (CBIR)

Blobworld system
Carson et al, 1999

Slide credit: Kristen Grauman
Recap: Color as Low-Level Cue

- Color histograms: Use distribution of colors to describe image
- No spatial information - invariant to translation, rotation, scale
Repetition

- Image Processing Basics
- Segmentation & Grouping
  - Segmentation and Grouping
  - Graph theoretic Segmentation
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

Gestalt factors

- K-Means & EM clustering
- Mean-shift clustering
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)

Untersuchungen zur Lehre von der Gestalt, Psychologische Forschung, Vol. 4, pp. 301-350, 1923
http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm
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, $c_1, \ldots, c_K$
  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:
      $$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$
Recap: Expectation Maximization (EM)

- **Goal**
  - Find blob parameters $\theta$ that maximize the likelihood function:
    $$P(data|\theta) = \prod_x P(x|\theta)$$

- **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

Slide credit: Steve Seitz
Recap: Mean-Shift Algorithm

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

Slide credit: Steve Seitz
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

Slide credit: Svetlana Lazebnik
Recap: Generic Clustering

• We have focused on ways to group pixels into image segments based on their appearance
  - Find groups; “quantize” feature space

• In general, we can use clustering techniques to find groups of similar “tokens”, provided we know how to compare the tokens.
  - *E.g.*, segment an image into the types of motions present
  - *E.g.*, segment a video into the types of scenes (shots) present
Repetition

- Image Processing Basics
- Segmentation & Grouping
  - Segmentation and Grouping
  - Graph theoretic Segmentation
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

Normalized cuts

Markov Random Fields

Graph cuts
Recap: Images as Graphs

- **Fully-connected graph**
  - Node (vertex) for every pixel
  - Link between *every* pair of pixels, \((p,q)\)
  - Affinity weight \(w_{pq}\) for each link (edge)
    - \(w_{pq}\) measures similarity
    - Similarity is *inversely proportional* to difference (in color and position...)

Slide credit: Steve Seitz
Recap: Normalized Cut (NCut)

- A minimum cut penalizes large segments
- This can be fixed by normalizing for size of segments
- The normalized cut cost is:

\[ N \text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \]

\[ \text{assoc}(A, V) = \text{sum of weights of all edges in } V \text{ that touch } A \]

\[ = \text{cut}(A, B) \left[ \frac{1}{\sum_{p \in A} w_{p,q}} + \frac{1}{\sum_{q \in B} w_{p,q}} \right] \]

- The exact solution is NP-hard but an approximation can be computed by solving a generalized eigenvalue problem.


Slide credit: Svetlana Lazebnik
Recap: NCuts: Overall Procedure

1. Construct a weighted graph $G=(V,E)$ from an image.
2. Connect each pair of pixels, and assign graph edge weights,
   $W(i, j) = \text{Prob. that } i \text{ and } j \text{ belong to the same region.}$
3. Solve $(D-W)y = \lambda Dy$ for the smallest few eigenvectors. This yields a continuous solution.
4. Threshold eigenvectors to get a discrete cut
   - This is where the approximation is made (we’re not solving NP).
5. Recursively subdivide if NCut value is below a pre-specified value.

NCuts Matlab code available at [http://www.cis.upenn.edu/~jshi/software/](http://www.cis.upenn.edu/~jshi/software/)
Recap: Markov Random Fields

\[ \Phi(x_i, y_i) \]

\[ \Psi(x_i, x_j) \]

Image patches

Scene patches

Image

Scene

Slide credit: William Freeman
Recap: Energy Formulation

- Joint probability
  \[ P(x, y) = \prod_i \Phi(x_i, y_i) \prod_{i,j} \Psi(x_i, x_j) \]

- Maximizing the joint probability is the same as minimizing the (negative) log
  \[ -\log P(x, y) = -\sum_i \log \Phi(x_i, y_i) - \sum_{i,j} \log \Psi(x_i, x_j) \]
  \[ E(x, y) = \sum_i \phi(x_i, y_i) + \sum_{i,j} \psi(x_i, x_j) \]

- This is similar to free-energy problems in statistical mechanics (spin glass theory). We therefore draw the analogy and call \( E \) an energy function.

- \( \phi \) and \( \psi \) are called potentials.
Recap: Energy Formulation

- **Energy function**

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

<table>
<thead>
<tr>
<th>Single-node potentials</th>
<th>Pairwise potentials</th>
</tr>
</thead>
</table>

- **Single-node potentials** $\phi$
  - Encode local information about the given pixel/patch
  - How likely is a pixel/patch to belong to a certain class (e.g. foreground/background)?

- **Pairwise potentials** $\psi$
  - Encode neighborhood information
  - How different is a pixel/patch’s label from that of its neighbor? (e.g. based on intensity/color/texture difference, edges)
Recap: Graph Cuts Energy Minimization

Regional bias example

Suppose $I^s$ and $I^t$ are given “expected” intensities of object and background

\[ D_p(s) \propto \exp \left( -\frac{\|I_p - I^s\|^2}{2\sigma^2} \right) \]
\[ D_p(t) \propto \exp \left( -\frac{\|I_p - I^t\|^2}{2\sigma^2} \right) \]

[Boykov & Jolly, ICCV’01]
Recap: Graph Cuts Energy Minimization

“expected” intensities of object and background can be re-estimated

\[ D_p(t) \propto \exp \left( -\| I_p - I^t \|^2 / 2\sigma^2 \right) \]

\[ D_p(s) \propto \exp \left( -\| I_p - I^s \|^2 / 2\sigma^2 \right) \]

EM-style optimization

Slide credit: Yuri Boykov

[Boykov & Jolly, ICCV’01]
Recap: When Can s-t Graph Cuts Be Applied?

- **s-t graph cuts** can only globally minimize **binary energies** that are **submodular**. 
  \[ E(L) = \sum_{p} E_{p}(L_{p}) + \sum_{pq \in N} E(L_{p}, L_{q}) \]

- Non-submodular cases can still be addressed with some optimality guarantees.
  - Current research topic

\[ E(s,s) + E(t,t) \leq E(s,t) + E(t,s) \]

Submodularity ("convexity")
Repetition

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

Appearance-based recognition

Histogram representations

Comparison measures

Probabilistic recognition

\[ \frac{D_x}{D_y} \]

\[ \text{Lap} \]

\[ \text{Histogram representations} \]

\[ \text{Comparison measures} \]

\[ \text{Probabilistic recognition} \]
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.

⇒ Fundamental paradigm shift in the 90’s
Recap: Recognition Using Global Features

- E.g. histogram comparison

![Test image](image1)

![Known objects](image2)
Recap: Comparison Measures

• Vector space interpretation
  - Euclidean distance

• Statistical motivation
  - Chi-square
  - Bhattacharyya

• Information-theoretic motivation
  - Kullback-Leibler divergence, Jeffreys divergence

• Histogram motivation
  - Histogram intersection

• Ground distance
  - Earth Movers Distance (EMD)
Recap: Recognition Using Histograms

• Simple algorithm
  1. Build a set of histograms $H=\{h_i\}$ for each known object
     ➢ More exactly, for each view of each object
  2. Build a histogram $h_t$ for the test image.
  3. Compare $h_t$ to each $h_i \in H$
     ➢ Using a suitable comparison measure
  4. Select the object with the best matching score
     ➢ Or reject the test image if no object is similar enough.

“Nearest-Neighbor” strategy
Recap: Histogram Backprojection

- „Where in the image are the colors we’re looking for?“
- Query: object with histogram $M$
- Given: image with histogram $I$
- Compute the „ratio histogram“: $R_i = \min \left( \frac{M_i}{I_i}, 1 \right)$
  - $R$ reveals how important an object color is, relative to the current image.
  - Project value back into the image (i.e. replace the image values by the values of $R$ that they index).
  - Convolve result image with a circular mask to find the object.
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.
Recap: Bayesian Recognition Algorithm

1. Build up histograms $p(m_k|o_n)$ for each training object.

2. Sample the test image to obtain $m_k$, $k \in K$.
   - Only small number of local samples necessary.

3. Compute the probabilities for each training object.

4. Select the object with the highest probability
   - Or reject the test image if no object accumulates sufficient probability.
Recap: Colored Derivatives

- Generalization: derivatives along
  - Y axis → intensity differences
  - C₁ axis → red-green differences
  - C₂ axis → blue-yellow differences

- Application:
  - Brand identification in video
First Applications Take Up Shape...

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

Image Source: http://www.flickr.com/photos/angelsk/2806412807/
Repetition

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

**PCA:** Distance TO eigenspace

**PCA:** Distance IN eigenspace

*B. Leibe*

*Fisher’s Linear Discriminant Analysis*
Recap: Subspace Methods

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

Representation:

\[ \text{image} = a_1 + a_2 + a_3 + \ldots \]

Classification and Regression:

Slide credit: Ales Leonardis
Recap: Obj. Detection by Distance TO Eigenspace

- Scan a window $\omega$ over the image and classify the window as object or non-object as follows:
  - Project window to subspace and reconstruct as earlier.
  - Compute the distance between $\omega$ and the reconstruction (reprojection error).
  - Local minima of distance over all image locations $\Rightarrow$ object locations
  - Repeat at different scales
  - Possibly normalize window intensity such that $|\omega|=1$. 

See Exercise 4.2!
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

Slide adapted from Ales Leonardis
Recap: Eigenfaces

Slide credit: Peter Belhumeur
Recap: Restrictions of PCA

- PCA minimizes projection error

- PCA is "unsupervised" no information on classes is used
- Discriminating information might be lost

Slide credit: Ales Leonardis
Recap: Fisher’s Linear Discriminant Analysis

- Maximize distance between classes
- Minimize distance within a class
- Criterion: \[ J(w) = \frac{w^T S_b w}{w^T S_w w} \]

\( S_b \) ... between-class scatter matrix
\( S_w \) ... within-class scatter matrix

- Vector \( w \) is a solution of a generalized eigenvalue problem:
  \[ S_b w = \lambda S_w w \]

- Classification function:
  \[ g(x) = w^T x + w_0 \Rightarrow \begin{cases} \text{Class 1} & \text{if } 0 \end{cases} \]
Recap: Fisherfaces

- Example Fisherface for recognition “Glasses/NoGlasses”
Repetition

• Image Processing Basics
• Segmentation & Grouping
• Object Recognition
• Local Features & Matching
   Local Features - Detection and Description
   Recognition with Local Features
• Object Categorization
• 3D Reconstruction
• Motion and Tracking
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 \textit{independently} in both images

- Problem 2:
  - For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor!

We need a repeatable detector!
Recap: Harris Detector [Harris88]

- Compute second moment matrix (autocorrelation matrix)

\[ M(\sigma_I, \sigma_D) = g(\sigma_I) \ast \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix} \]

1. Image derivatives
2. Square of derivatives
3. Gaussian filter \( g(\sigma) \)
4. Cornerness function - two strong eigenvalues

\[ R = \det[M(\sigma_I, \sigma_D)] - \alpha[\text{trace}(M(\sigma_I, \sigma_D))] \]

\[ = g(I_x^2) g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2 \]

5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk

B. Leibe
Recap: Harris Detector Responses [Harris88]

*Effect:* A very precise corner detector.

Slide credit: Krystian Mikolajczyk
Recap: Hessian Detector [Beaudet78]

- Hessian determinant

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

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

In Matlab:

\[
I_{xx} \cdot I_{yy} - (I_{xy})^2
\]

Slide credit: Krystian Mikolajczyk
Recap: Hessian Detector Responses [Beaudet78]

Effect: Responses mainly on corners and strongly textured areas.

Slide credit: Krystian Mikolajczyk
Recap: Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{i\ldots j_{m}}(x, \sigma)) \]

\[ f(I_{i\ldots j_{m}}(x', \sigma')) \]

Slide credit: Krystian Mikolajczyk
Recap: Laplacian-of-Gaussian (LoG)

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

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \]

\( \Rightarrow \) List of \((x, y, \sigma)\)

Slide adapted from Krystian Mikolajczyk
Recap: LoG Detector Responses
Recap: Key point localization with DoG

- 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 \((x, y, \sigma)\)

Image source: David Lowe
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

Harris-Laplace points

Slide adapted from Krystian Mikolajczyk
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: 4x4x8 = 128 dimensions

Repetition

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
  - Local Features - Detection and Description
  - Recognition with Local Features
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

Recognition pipeline

Fitting affine transformations & homographies

RANSAC

Gen. Hough Transform
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?

\[
\begin{bmatrix}
    x'_i \\
    y'_i \\
\end{bmatrix} = \begin{bmatrix}
    m_1 & m_2 \\
    m_3 & m_4 \\
\end{bmatrix}\begin{bmatrix}
    x_i \\
    y_i \\
\end{bmatrix} + \begin{bmatrix}
    t_1 \\
    t_2 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    m_1 & m_2 & 0 & 0 & 1 & 0 \\
    m_3 & m_4 & 0 & 0 & x_i & y_i \\
    t_1 & t_2 & \cdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
= \begin{bmatrix}
    x'_i \\
    y'_i \\
\end{bmatrix}
\]
Recap: Fitting a Homography

- Estimating the transformation

\[ x' = Hx \]
\[ x'' = \frac{1}{z'} x' \]
Recap: Fitting a Homography

• Estimating the transformation

\[
\begin{align*}
    h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13} - x_{A_i} h_{31} x_{B_i} - x_{A_i} h_{32} y_{B_i} - x_{A_i} &= 0 \\
    h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23} - y_{A_i} h_{31} x_{B_i} - y_{A_i} h_{32} y_{B_i} - y_{A_i} &= 0
\end{align*}
\]

\[
\begin{bmatrix}
    x_{B_1} & y_{B_1} & 1 & 0 & 0 & 0 & -x_{A_1} x_{B_1} & -x_{A_1} y_{B_1} & -x_{A_1} \\
    0 & 0 & 0 & x_{B_1} & y_{B_1} & 1 & -y_{A_1} x_{B_1} & -y_{A_1} y_{B_1} & -y_{A_1} \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\begin{bmatrix}
    h_{11} \\
    h_{12} \\
    h_{13} \\
    h_{21} \\
    h_{22} \\
    h_{23} \\
    h_{31} \\
    h_{32} \\
    1
\end{bmatrix} = 0
\]

\[
A h = 0
\]
Recap: Fitting a Homography

- Estimating the transformation
  - Solution:
    - Null-space vector of $A$
    - Corresponds to smallest eigenvector

$$Ah = 0$$

$$A = UDV^T = U \begin{bmatrix} d_{11} & \cdots & d_{19} \\ \vdots & \ddots & \vdots \\ d_{91} & \cdots & d_{99} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{v_{19}} \\ \vdots & \ddots & \vdots \\ v_{v_{91}} & \cdots & v_{v_{99}} \end{bmatrix}^T$$

$$h = \begin{bmatrix} v_{19}, \cdots, v_{99} \\ v_{99} \end{bmatrix}$$

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, re-compute 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
Recap: RANSAC Line Fitting Example

- Task: Estimate the best line
Recap: RANSAC Line Fitting Example

- Task: Estimate the best line

Fit a line to them
Recap: RANSAC Line Fitting Example

- Task: Estimate the best line

Total number of points within a threshold of line.

Slide credit: Jinxiang Chai
Recap: RANSAC Line Fitting Example

• Task: Estimate the best line

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

- Find best stereo match within a square search window (here 300 pixels$^2$)
- Global transformation model: epipolar geometry

before RANSAC

after RANSAC

Images from Hartley & Zisserman

Slide credit: David Lowe
Recap: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).

Slide credit: Svetlana Lazebnik
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

[Brown & Lowe, ICCV'03]
Repetition

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

Boosting

SVM

Sliding window principle

Car/non-car Classifier

Viola-Jones face detector

Train cascade of classifiers with AdaBoost

Selected features, thresholds, and weights

B. Leibe
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.

Slide credit: Kristen Grauman
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

Slide credit: Kristen Grauman
Recap: Classifier Construction Choices...

**Nearest neighbor**
- Shakhnarovich, Viola, Darrell 2003
- Berg, Berg, Malik 2005...

**Neural networks**
- LeCun, Bottou, Bengio, Haffner 1998
- Rowley, Baluja, Kanade 1998

**Support Vector Machines**
- Guyon, Vapnik
- Heisele, Serre, Poggio, 2001,...

**Boosting**
- Viola, Jones 2001,
- Torralba et al. 2004,
- Opelt et al. 2006,...

**Conditional Random Fields**
- McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003, ...

---

Slide credit: Kristen Grauman

Slide adapted from Antonio Torralba
Recap: AdaBoost

Final classifier is combination of the weak classifiers

Final classifier is linear combination of weak classifiers

Weights Increased

Weak Classifier 2

Weak Classifier 1

Weak classifier 3

B. Leibe

Slide credit: Kristen Grauman
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

Value at \((x,y)\) is sum of pixels above and to the left of \((x,y)\)

\[
D = 1 + 4 - (2 + 3) \\
= A + (A + B + C + D) - (A + C + A + B) \\
= D
\]

Integral image

Slide credit: Kristen Grauman

B. Leibe

[Viola & Jones, CVPR 2001]
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.

Resulting weak classifier:

\[ h_t(x) = \begin{cases} 
+1 & \text{if } f_t(x) > \theta_t \\
-1 & \text{otherwise} 
\end{cases} \]

For next round, reweight the examples according to errors, choose another filter/threshold combo.
Application: Viola-Jones Face Detector

- 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/]

Slide credit: Kristen Grauman
Recap: Support Vector Machines (SVMs)

- Discriminative classifier based on \textit{optimal separating hyperplane} (i.e. line for 2D case)
- Maximize the \textit{margin} between the positive and negative training examples

Slide credit: Kristen Grauman
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: 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]
Repetition

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

Bag-of-words representation

$O(N)$

Object

Bag of “words”
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.

Exact feature matching $\rightarrow$ Match to same visual word

Figure from Sivic & Zisserman, ICCV 2003

Slide adapted from Kristen Grauman
Recap: Bag-of-Word Representations (BoW)

Object → Bag of “words”

Source: ICCV 2005 short course, Li Fei-Fei
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).

Slide adapted from Kristen Grauman
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.

Slide credit: Kristen Grauman
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?
Repetition

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
  - Sliding Window based Object Detection
  - Bag-of-Words Approaches
  - Part based Approaches
- 3D Reconstruction
- Motion and Tracking
Recap: Part-Based Models

- Fischler & Elschlager 1973

- Model has two components
  - parts (2D image fragments)
  - structure (configuration of parts)
Recap: Implicit Shape Model - Representation

- Learn appearance codebook
  - Extract local features at interest points
  - Clustering \Rightarrow \text{appearance codebook}

- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object
Recap: Implicit Shape Model - Recognition

- Interest Points
- Matched Codebook Entries
- Probabilistic Voting

Backprojected Hypotheses → Backprojection of Maxima

“Generalized Hough Transform with backprojection”

3D Voting Space (continuous)

[Leibe, Leonardis, Schiele, SLCV’04; IJCV’08]
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
    \[
    \begin{align*}
    x_{vote} &= x_{img} - x_{occ}(s_{img}/s_{occ}) \\
    y_{vote} &= y_{img} - y_{occ}(s_{img}/s_{occ}) \\
    s_{vote} &= (s_{img}/s_{occ}).
    \end{align*}
    \]
  - Search for maxima in 3D voting space
**Repetition**

- 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

**Epipolar geometry**

**Image rectification**

**Disparity**

**Dense stereo matching**
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

Slide credit: Steve Seitz
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

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

\[ X' \cdot (T_x \ RX) = 0 \]

Let \( E = T_x R \)

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

- 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

\[
p' \, \text{Ep} = 0
\]

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

\[
l' = Ep
\]

is the coordinate vector representing the epipolar line for point \( p \)

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

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

is the coordinate vector representing the epipolar line for point \( p' \)

Slide credit: Kristen Grauman
Recap: Stereo Image Rectification

- 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
Recap: Dense Correspondence Search

- 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
  ⇒ Rectify images first

adapted from Svetlana Lazebnik, Li Zhang
Recap: Effect of Window Size

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with the same disparity.
Recap: Additional Constraints

- **Uniqueness**
  - For opaque objects, there is at most one match in the right image for every point in the left image.

- **Ordering**
  - Points on *same surface* (opaque object) will be in same order in both views.
Repetition

- 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

Camera models

\[ K = \begin{bmatrix} \alpha_x & s & x_0 \\ \alpha_y & y_0 & 1 \end{bmatrix} \]

Camera calibration

Triangulation

Essential matrix, Fundamental matrix

\[ x^T E x' = 0 \]
\[ x^T F x' = 0 \]

Eight-point algorithm

SVD!
Recap: A General Point

- Equations of the form
  \[ A x = 0 \]

- How do we solve them? (always!)
  - Apply SVD
  \[
  A = U D V^T = U \begin{bmatrix}
  d_{11} & \cdots & \\
  \vdots & \ddots & \\
  \end{bmatrix}
  \begin{bmatrix}
  v_{11} & \cdots & v_{1N} \\
  \vdots & \ddots & \\
  v_{N1} & \cdots & v_{NN}
  \end{bmatrix}^T
  \]
  - Singular values of \( A = \) square roots of the eigenvalues of \( A^T A \).
  - The solution of \( Ax=0 \) is the nullspace vector of \( A \).
  - This corresponds to the smallest singular vector of \( A \).

B. Leibe
Recap: Camera Parameters

- **Intrinsic parameters**
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew* (non-rectangular pixels)
  - *Radial distortion*

\[
K = \begin{bmatrix}
m_x & m_y & 0 \\
f & S & p_x \\
1 & f & p_y \\
1 & 1 & 1
\end{bmatrix}
\]

\[
P = K [ R \mid t ]
\]

- **Extrinsic parameters**
  - Rotation R
  - Translation t
    (both relative to world coordinate system)

- **Camera projection matrix**
  - General pinhole camera: 9 DoF
  - CCD Camera with square pixels: 10 DoF
  - General camera: 11 DoF
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 $P = P_{\text{int}} P_{\text{ext}}$
Recap: Camera Calibration (DLT Algorithm)

\[
\begin{bmatrix}
0^T & X_1^T & -y_1X_1^T \\
X_1^T & 0^T & -x_1X_1^T \\
\vdots & \vdots & \vdots \\
0^T & X_n^T & -y_nX_n^T \\
X_n^T & 0^T & -x_nX_n^T \\
\end{bmatrix}
\begin{pmatrix}
P_1 \\
P_2 \\
P_3 \\
\end{pmatrix} = 0 \\
Ap = 0
\]

- 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 ½ correspondences needed for a minimal solution.

- Two independent equations each in terms of three unknown entries of $X$.
- Stack equations and solve with SVD.
- This approach nicely generalizes to multiple cameras.

\[
\begin{align*}
\lambda_1 x_1 &= P_1 X \\
\lambda_2 x_2 &= P_2 X
\end{align*}
\]

\[
\begin{align*}
x_1 \times P_1 X &= 0 \\
x_2 \times P_2 X &= 0
\end{align*}
\]

\[
\begin{align*}
[x_{1\times}] P_1 X &= 0 \\
[x_{2\times}] P_2 X &= 0
\end{align*}
\]
Recap: Epipolar Geometry - Calibrated Case

Camera matrix: $[I|0]$

$X = (u, v, w, 1)^T$

$x = (u, v, w)^T$

Camera matrix: $[R^T | -R^T t]$

Vector $x'$ in second coord. system has coordinates $Rx'$ in the first one.

The vectors $x$, $t$, and $Rx'$ are coplanar
Recap: Epipolar Geometry - Calibrated Case

\[ x \cdot [t \times (Rx')] = 0 \quad \Rightarrow \quad x^T Ex' = 0 \quad \text{with} \quad E = [t \times] R \]

Essential Matrix
(Longuet-Higgins, 1981)
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:

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

$$x = K \hat{x}, \quad x' = K' \hat{x}'$$

Slide credit: Svetlana Lazebnik
Recap: Epipolar Geometry - Uncalibrated Case

\[ \hat{x}^T E \hat{x}' = 0 \quad \Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1} \]

\[ x = K \hat{x} \]

\[ x' = K' \hat{x}' \]

Fundamental Matrix
(Faugeras and Luong, 1992)
Recap: The Eight-Point Algorithm

\[
x = (u, v, 1)^T, \quad x' = (u', v', 1)^T
\]

\[
(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad (uu', uv', u, vv', vv', v, u', v', 1) = 0
\]

1.) Solve with SVD. This minimizes

\[
\sum_{i=1}^{N} (x_i^T F x_i')^2
\]

2.) Enforce rank-2 constraint using SVD

• Problem: poor numerical conditioning
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.

$$F = U D V^T = U \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & d_{33} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{31} & \cdots & v_{33} \end{bmatrix}^T$$

Set $d_{33}$ 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 $T^T F T'$.

[Harlty, 1995]
Recap: Comparison of Estimation Algorithms

<table>
<thead>
<tr>
<th></th>
<th>8-point</th>
<th>Normalized 8-point</th>
<th>Nonlinear least squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Av. Dist. 1</td>
<td>2.33 pixels</td>
<td>0.92 pixel</td>
<td>0.86 pixel</td>
</tr>
<tr>
<td>Av. Dist. 2</td>
<td>2.18 pixels</td>
<td>0.85 pixel</td>
<td>0.80 pixel</td>
</tr>
</tbody>
</table>

Slide credit: Svetlana Lazebnik
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?

\[
l_{31} = F_{13}^T x_1 \\
l_{32} = F_{23}^T x_2
\]
Recap: Active Stereo with Structured Light

- **Optical triangulation**
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
Applications: 3D Reconstruction
Repetition

- 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

\[ \begin{align*}
\mathbf{X}_1 & \rightarrow \mathbf{P}_1 \\
\mathbf{X}_2 & \rightarrow \mathbf{P}_2 \\
\mathbf{X}_3 & \rightarrow \mathbf{P}_3
\end{align*} \]

Projective ambiguity

Affine factorization

Euclidean upgrade

\[ a_1 \cdot a_2 = 0 \]
\[ |a_1|^2 = |a_2|^2 = 1 \]
Recap: Structure from Motion

• Given: $m$ images of $n$ fixed 3D points

  \[ x_{ij} = P_i X_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \]

• 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
  \[
  \begin{bmatrix}
  A & t \\
  v^T & v
  \end{bmatrix}
  \]
  Preserves intersection and tangency

- Affine 12dof
  \[
  \begin{bmatrix}
  A & t \\
  0^T & 1
  \end{bmatrix}
  \]
  Preserves parallelism, volume ratios

- Similarity 7dof
  \[
  \begin{bmatrix}
  s R & t \\
  0^T & 1
  \end{bmatrix}
  \]
  Preserves angles, ratios of length

- Euclidean 6dof
  \[
  \begin{bmatrix}
  R & t \\
  0^T & 1
  \end{bmatrix}
  \]
  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.

Slide credit: Svetlana Lazebnik
Recap: Affine Structure from Motion

- Let’s create a $2m \times n$ data (measurement) matrix:

$$D = \begin{bmatrix}
\hat{X}_{11} & \hat{X}_{12} & \cdots & \hat{X}_{1n} \\
\hat{X}_{21} & \hat{X}_{22} & \cdots & \hat{X}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{X}_{m1} & \hat{X}_{m2} & \cdots & \hat{X}_{mn}
\end{bmatrix} = \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_m
\end{bmatrix} \begin{bmatrix}
X_1 \\
X_2 \\
\vdots \\
X_n
\end{bmatrix}$$

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


Slide credit: Svetlana Lazebnik
Recap: Affine Factorization

- Obtaining a factorization from SVD:

\[ D = U_3 \times W_3 \times V_3^T \]

Possible decomposition:

\[ M = U_3 W_3^{1/2} \quad S = W_3^{1/2} V_3^T \]

This decomposition minimizes \(|D-MS|^2|\).
Recap: Projective Factorization

$$D = \begin{bmatrix} z_{11}x_{11} & z_{12}x_{12} & \cdots & z_{1n}x_{1n} \\ z_{21}x_{21} & z_{22}x_{22} & \cdots & z_{2n}x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1}x_{m1} & z_{m2}x_{m2} & \cdots & z_{mn}x_{mn} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_n \end{bmatrix}$$

Cameras (3m x 4)
Points (4 x n)

$$D = MS$$ has rank 4

- 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).

Slide credit: Svetlana Lazebnik
Recap: Sequential Projective SfM

- Initialize motion from two images using fundamental matrix
- 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*
Recap: Estimating the Euclidean Upgrade

- **Goal:** Estimate ambiguity matrix $C$
  - **Orthographic assumption:**
  - This can be converted into a system of $3m$ equations:
    \[
    \begin{align*}
    \hat{a}_{i1} \cdot \hat{a}_{i2} &= 0 \\
    |\hat{a}_{i1}| &= 1 \\
    |\hat{a}_{i2}| &= 1
    \end{align*}
    \]
    \[
    \begin{align*}
    a_{i1}^T C C^T a_{i2} &= 0 \\
    a_{i1}^T C C^T a_{i1} &= 1, \quad i = 1, \ldots, m \\
    a_{i2}^T C C^T a_{i2} &= 1
    \end{align*}
    \]

  - 1) Image axes are perpendicular $a_1 \cdot a_2 = 0$
  - 2) Scale is 1 $|a_1|^2 = |a_2|^2 = 1$

- This translates to
  \[
  L = C C^T 
  \]
  with
  \[
  A_i L A_i^T = I
  \]

---

Slide adapted from S. Lazebnik, M. Hebert
Recap: Bundle Adjustment

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

\[ E(P, X) = \sum_{i=1}^{m} \sum_{j=1}^{n} D(x_{ij}, P_i X_j)^2 \]
Repetition

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking
  - Motion and Optical Flow
  - Tracking with Linear Dynamic Models

\[
\begin{bmatrix}
\sum I_x I_x & \sum I_x I_y \\
\sum I_x I_y & \sum I_y I_y
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -
\begin{bmatrix}
\sum I_x I_t \\
\sum I_y I_t
\end{bmatrix}

\]

Lucas-Kanade optical flow

Coarse-to-fine estimation

Use for feature tracking
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:

\[
\begin{bmatrix}
I_x(p_1) & I_y(p_1) \\
I_x(p_2) & I_y(p_2) \\
\vdots & \vdots \\
I_x(p_{25}) & I_y(p_{25})
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = -
\begin{bmatrix}
I_t(p_1) \\
I_t(p_2) \\
\vdots \\
I_t(p_{25})
\end{bmatrix}
\]

where

\[A \quad d = b\]

\[
25 \times 2 \quad 2 \times 1 \quad 25 \times 1
\]

- Minimum least squares solution given by solution of

\[
(A^T A) \quad d = A^T b
\]

where

\[
2 \times 2 \quad 2 \times 1 \quad 2 \times 1
\]

Recall the Harris detector!
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

Gaussian pyramid of image 1

- $u=10$ pixels
- $u=5$ pixels
- $u=2.5$ pixels
- $u=1.25$ pixels

Gaussian pyramid of image 2

- $u=10$ pixels

Image 1

Image 2

Slide credit: Steve Seitz
Recap: Coarse-to-fine Estimation

Image 1
Gaussian pyramid of image 1

Run iterative L-K

Warp & upsample

Run iterative L-K

Image 2
Gaussian pyramid of image 2

Slide credit: Steve Seitz
Repetition

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking
  - Motion and Optical Flow
  - Tracking with Linear Dynamic Models

Tracking as inference

Prediction-Correction cycle

Linear motion models

Kalman filter
Recap: Tracking as Inference

- The hidden state consists of the true parameters we care about, denoted $X$.

- The measurement is our noisy observation that results from the underlying state, denoted $Y$.

- At each time step, state changes (from $X_{t-1}$ to $X_t$) and we get a new observation $Y_t$.

- Our goal: recover most likely state $X_t$ given
  - All observations seen so far.
  - Knowledge about dynamics of state transitions.
Recap: Tracking as Induction

- **Base case:**
  - Assume we have initial prior that predicts state in absence of any evidence: $P(X_0)$
  - At the first frame, *correct* this given the value of $Y_0 = y_0$

- **Given corrected estimate for frame $t$:**
  - Predict for frame $t+1$
  - Correct for frame $t+1
Recap: Prediction and Correction

- **Prediction:**

\[
P(X_t | y_0, \ldots, y_{t-1}) = \int P(X_t | X_{t-1})P(X_{t-1} | y_0, \ldots, y_{t-1})dX_{t-1}
\]

\[
\text{Dynamics model} \quad \text{Corrected estimate from previous step}
\]

- **Correction:**

\[
P(X_t | y_0, \ldots, y_t) = \frac{P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})}{\int P(y_t | X_t)P(X_t | y_0, \ldots, y_{t-1})dX_t}
\]

\[
\text{Observation model} \quad \text{Predicted estimate}
\]
Recap: Linear Dynamic Models

• Dynamics model
  - State undergoes linear transformation $D_t$ plus Gaussian noise
    \[ x_t \sim N(D_t x_{t-1}, \Sigma_{d_t}) \]

• Observation model
  - Measurement is linearly transformed state plus Gaussian noise
    \[ y_t \sim N(M_t x_t, \Sigma_{m_t}) \]
Recap: The Kalman Filter

Know corrected state from previous time step, and all measurements up to the current one → Predict distribution over next state.

Receive measurement

Know prediction of state, and next measurement → Update distribution over current state.

Time update

"Predict"

\[ P(X_t | y_0, \ldots, y_{t-1}) \]

Mean and std. dev. of predicted state:

\[ \mu_t^-, \sigma_t^- \]

Measurement update

"Correct"

\[ P(X_t | y_0, \ldots, y_t) \]

Mean and std. dev. of corrected state:

\[ \mu_t^+, \sigma_t^+ \]

Time advances: \( t++ \)

Slide credit: Kristen Grauman
Recap: General Kalman Filter (>1dim)

- What if state vectors have more than one dimension?

**PREDICT**

\[
x_t^- = D_t x_{t-1}^+
\]

\[
\Sigma_t^- = D_t \Sigma_{t-1}^+ D_t^T + \Sigma_d
\]

**CORRECT**

\[
K_t = \Sigma_t^- M_t^T \left( M_t \Sigma_t^- M_t^T + \Sigma_m \right)^{-1}
\]

\[
x_t^+ = x_t^- + K_t \left( y_t - M_t x_t^- \right)
\]

\[
\Sigma_t^+ = \left( I - K_t M_t \right) \Sigma_t^-
\]

More weight on residual when measurement error covariance approaches 0.

Less weight on residual as a priori estimate error covariance approaches 0.

For derivations, see F&P Chapter 17.3

Slide credit: Kristen Grauman
Any Questions?

So what can you do with all of this?
Mobile Object Detection & Tracking

[Ess, Leibe, Schindler, Van Gool, CVPR’08]
Mining the World’s Images...
Mobile Visual Search & Mobile AR

- 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

Query image

Camera position

[T. Sattler, B. Leibe, L. Kobbelt, ICCV’11]
Automatic Landmark Building Discovery
Any More Questions?

See you in *Machine Learning* next semester...