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/

• 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.)

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)

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
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 \( \sigma_{\text{between}}^2(T) = n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2 \).
- 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 \( (B \hat{z}) \) and \( A \) overlap by at least one element".
  - \( (B \hat{z}) \) is the mirrored version of \( B \), shifted by \( z \).
- Effects
  - If current pixel \( z \) is foreground, set all pixels under \( (B \hat{z}) \) to foreground.
  - Expand connected components
  - Grow features
  - Fill holes

Recap: Erosion

- Definition
  - "The erosion of \( A \) by \( B \) is the set of all displacements \( z \), such that \( (B \hat{z}) \) is entirely contained in \( A \)."
- Effects
  - If not every pixel under \( (B \hat{z}) \) 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
    \[ A \ast B = (A \ominus B) \oplus B \]
- Effect
  - \( A \ast B \) is defined by the points that are reached if \( B \) is rolled around inside \( A \).
  ⇒ 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 \( A \).
  ⇒ 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
     - If only one of its neighbors (top or left) is 1, copy its label.
     - If both are 1 and have the same label, copy it.
     - If they have different labels:
       - Copy the label from the left.
       - Update the equivalence table.
     - Otherwise, assign a new label.
  2. 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
  \[ \eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} \]
  \[ y = \frac{p + q + 1}{2} \]
- From those, a set of invariant moments can be defined for object description.
  \[ \phi_1 = \eta_{00}, \phi_2 = (\eta_{10} - \eta_{01})^2 + 4\eta_{11}^2, \]
  \[ \phi_3 = (\eta_{10} - 3\eta_{01})^2 + (3\eta_{11} - \eta_{01})^2, \]
  \[ \phi_4 = (\eta_{00} + \eta_{11})^2 + (\eta_{01} + \eta_{10})^2, \]
- 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

- High resolution
- Low resolution

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:
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: 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

Recap: Chamfer Matching
- Chamfer Distance
  - Average distance to nearest feature
  \[
  D_{chamfer}(T, I) = \frac{1}{|T|} \sum_{t \in T} d_f(t)
  \]
  - This can be computed efficiently by correlating the edge template with the distance-transformed image

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.

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.

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

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

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: \(r = a - p_i\).

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

Recap: Color Sensing

- Electromagnetic spectrum

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

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 \(x = X/(X+Y+Z)\), \(y = Y/(X+Y+Z)\)

Recap: HSV Color Space

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

Recap: Color as Low-Level Cue

- Color histograms: Use distribution of colors to describe image
- No spatial information - invariant to translation, rotation, scale
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 227 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:
    \[
    \sum_{i=1}^{k} \sum_{p \in \text{cluster } i} ||p - c_i||^2
    \]

Recap: Expectation Maximization (EM)

- Goal
  - Find blob parameters $\theta$ that maximize the likelihood function:
    \[
    P(\text{data}|\theta) = \prod P(x|\theta)
    \]

- 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

Recap: Mean-Shift Algorithm

- Iterative Mode Search
  1. Initialize random seed, and window $W$
  2. Calculate center of gravity (the "mean") of $W$
  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: 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

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...)

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:
  \[
  \text{Ncut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}
  \]
  \[
  \text{assoc}(A,V) = \text{sum of weights of all edges in } V \text{ that touch } A
  \]
  \[
  \text{cut}(A,B) = \frac{1}{\sum_{p \in A} w_{pq}} + \frac{1}{\sum_{q \in B} w_{pq}}
  \]
- The exact solution is NP-hard but an approximation can be computed by solving a generalized eigenvalue problem.

J. Shi and J. Malik, Normalized cuts and image segmentation. PAMI 2000
Recap: Graph Cuts Energy Minimization

Suppose \( I_o \) and \( I_b \) are given “expected” intensities of object and background

\[
D_x(s) = \exp \left(-\frac{1}{2\sigma^2} \| I_o - I_o \|^2 \right)
\]

\[
D_y(t) = \exp \left(-\frac{1}{2\sigma^2} \| I_b - I_b \|^2 \right)
\]

“expected” intensities of object and background \( I_o \) and \( I_b \) can be re-estimated

EM-style optimization

Recap: Energy Formulation

- Joint probability
  \[
P(x, y) = \prod_i \Phi(x_i, y_i) \prod_i \Psi(x_i, x_j)
\]

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

- 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: Markov Random Fields

- Energy function
  \[
  E(x, y) = \sum_i \phi(x_i, y_i) + \sum_j \psi(x_i, x_j)
  \]

- 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: 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 \((D-W)y = \lambda D y\) 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.
Recap: When Can s-t Graph Cuts Be Applied?

The graph cut energy function is:

\[ E(L) = \sum_{P} E_P(L_P) + \sum_{Q} E_{Q}(P,Q) \]

where:
- \( E_P(L_P) \) is the sum of energies over t-links
- \( E_{Q}(P,Q) \) is the sum of energies over n-links

- **s-t graph cuts** can only globally minimize binary energies that are **submodular**.
  
  \[ E(s,s) + E(t,t) \leq E(s,t) + E(t,s) \]
  
  - Non-submodular cases can still be addressed with some optimality guarantees.
  - Current research topic

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

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":
  \[ R_i = \min \left( \frac{M}{I}, 1 \right) \]
  - $R$ 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.
   \[ p(o_n|\text{Image}) = \frac{\prod_k p(m_k|o_n)p(o_n)}{\sum_n \prod_k p(m_k|o_n)p(o_n)} \]
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
  - $C_1$ axis → red-green differences
  - $C_2$ axis → blue-yellow differences
- Application:
  - Brand identification in video

First Applications Take Up Shape...

- Line detection
- Circle detection
- Binary Segmentation
- Skin color detection
- Moment descriptors

Repetition

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

Fisher’s Linear Discriminant Analysis
Recap: Subspace Methods

- **Reconstructive**
  - PCA, ICA, NMF
  - representation

- **Discriminative**
  - LDA, SVM, CCA
  - classification, regression

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 \( |\omega| = 1 \).

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).

Recap: Restrictions of PCA

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

Recap: Eigenfaces

- Estimate parameters by finding the NN in the eigenspace

Recap: Fisher’s Linear Discriminant Analysis

- Maximize distance between classes
- Minimize distance within a class
- Criterion: \( J(w) = \frac{w^T S_{bw}}{w^T S_{ww}} \)
  - \( S_{bw} \) between-class scatter matrix
  - \( S_{ww} \) within-class scatter matrix
- Vector \( w \) is a solution of a generalized eigenvalue problem: \( S_{bw} w = \lambda S_{ww} w \)
- Classification function:
  - Class 1: \( g(x) = w^T x + w_0 \) \( \leq 0 \) for Class 1
  - Class 2
Recap: Fisherfaces

- Example Fisherface for recognition “Glasses/NoGlasses”

Slide credit: Peter Belhumeur

Recap: Local Feature Matching Pipeline

1. Find a set of distinctive key points
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

Recap: Harris Detector

- Compute second moment matrix (autocorrelation matrix)

\[
M(\sigma_x, \sigma_y) = g(\sigma_x) \begin{bmatrix}
I_x^2(\sigma_x) & I_x I_y(\sigma_x) \\
I_x I_y(\sigma_x) & I_y^2(\sigma_y)
\end{bmatrix} g(\sigma_y)
\]

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

\[
R = \text{det}(M(\sigma_x, \sigma_y)) - \lambda \text{trace}(M(\sigma_x, \sigma_y))
\]

\[
g(t_x^2, t_y^2) - \lambda (g(t_x^2) + g(t_y^2))
\]

5. Perform non-maximum suppression

Recap: Harris Detector Responses

Effect: A very precise corner detector.
Recap: Hessian Detector [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:

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

Recap: Hessian Detector Responses [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

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

\[ \Rightarrow \text{List of } (x, y, \sigma) \]

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
Recap: Harris-Laplace [Mikolajczyk '01]

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

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)
  - Resulting descriptor: 4x4x8 = 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?

Recap: Fitting a Homography

• Estimating the transformation

Gen. Hough Transform
Recap: Fitting a Homography

- Estimating the transformation

$$Ah = 0$$

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

$$A = \begin{bmatrix} x_1 & y_1 & 1 \ \ 0 & 0 & 1 \ \ \vdots & \vdots & \vdots \ \ x_n & y_n & 1 \end{bmatrix}$$

$$Ah = 0$$

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
Recap: RANSAC Line Fitting Example

- Task: Estimate the best line

Recap: RANSAC Line Fitting Example

- Task: Estimate the best line

Recap: Feature Matching Example

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

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
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: 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: AdaBoost

- Weights Increased
- Final classifier is combination of the weak classifiers

Recap: Viola-Jones Face Detection

- 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

Slide credit: Kristen Grauman
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:

\[ h(x) = \begin{cases} +1 & \text{if } f(x) > 0_i \\ -1 & \text{otherwise} \end{cases} \]

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/

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: 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.

Code available: http://pascal.inrialpes.fr/soft/olt/

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: 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?
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: Part-Based Models

- Implicit Shape Model
  - Sliding Window based Object Detection
  - Bag-of-Words Approaches
  - Part based Approaches

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
- Matched Codebook Entries
- Probabilistic Voting

“Generalized Hough Transform with backprojection”

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
    \[ x_{scale} = x_{img} - x_{wc}(s_{img}/s_{wc}) \]
    \[ y_{scale} = y_{img} - y_{wc}(s_{img}/s_{wc}) \]
    \[ s_{scale} = (s_{img}/s_{wc}) \]
  - 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
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 \( E = TR \)

\[
X' \cdot TX = 0
\]
- This holds for the rays \( p \) and \( p' \) that are parallel to the camera-centered position vectors \( X \) and \( X' \), so we have: \( p'^T EP = 0 \)
- \( E \) is called the essential matrix, which relates corresponding image points [Longuet-Higgins 1981]
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 reprojecion

How do we solve them? (always!)
- 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

Reproductive

For each pixel in the first image

This corresponds to the smallest singular vector of $A$.

The solution of $Ax=0$ is the smallest singular vector of $A$. 

Recap: Effect of Window Size

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

Recap: Stereo Image Rectification

Recap: Dense Correspondence Search

Recap: Dense Correspondence Search

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.
Recap: Camera Parameters

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

- **Extrinsic parameters**
  - Rotation \( R \)
  - Translation \( t \)

- Camera projection matrix \( P = K[R | t] \)

- \( K \) 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.
- 5½ correspondences needed for a minimal solution.

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 \( P = P_{int} P_{ext} \)

Recap: Camera Calibration (DLT Algorithm)

\[
\begin{bmatrix}
0 & X_i^T - y_i X_i' \\
X_i & 0 & X_i' - x_i X_i' \\
\vdots & \vdots & \vdots \\
0 & X_i^T - y_i X_i' \\
X_i & 0 & X_i' - x_i X_i'
\end{bmatrix}
\begin{bmatrix}
P_x \\
P_y \\
P_z
\end{bmatrix}
= 0 \\
Ap = 0
\]

\( P \) has 11 degrees of freedom.
- Two linearly independent equations per independent 2D/3D correspondence.
- Solve with SVD
- Solution corresponds to smallest singular vector.
- 5½ 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.

Recap: Epipolar Geometry - Calibrated Case

- The vectors \( v_i \), \( v_i' \), and \( Rx' \) are coplanar.

Recap: Epipolar Geometry - Calibrated Case

\[ x' - [t \times (R x')] = 0 \]
\[ x' E x' = 0 \text{ with } E = [t] R \]

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:
  \[
  \hat{x}^T E \hat{x}' = 0 \quad \hat{x} = K \hat{x}, \quad \hat{x}' = K' \hat{x}'
  \]

Recap: The Eight-Point Algorithm

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

\[
(a_{11}, a_{12}, a_{13}, a_{14}, a_{21}, a_{22}, a_{23}, a_{24}, a_{31}, a_{32}, a_{33}, a_{34}, b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23}, b_{31}, b_{32}, b_{33}, b_{34})^T = 0
\]

1.) Solve with SVD. This minimizes \( \sum_{i=1}^{2n} (x_i^T F x_i)^2 \)
2.) Enforce rank-2 constraint using SVD

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.

\[
F = U D V^T
\]

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: Comparison of Estimation Algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>Av. Dist. 1</th>
<th>Av. Dist. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-point</td>
<td>2.33 pixels</td>
<td>2.18 pixels</td>
</tr>
<tr>
<td>Normalized B-point</td>
<td>0.92 pixel</td>
<td>0.85 pixel</td>
</tr>
<tr>
<td>Nonlinear least squares</td>
<td>0.68 pixel</td>
<td>0.80 pixel</td>
</tr>
</tbody>
</table>

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?
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

Recap: Structure from Motion

- Given: \( m \) images of \( n \) fixed 3D points
- 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
  - 15 dofs
  - Preserves intersection and tangency
- Affine
  - 12 dofs
  - Preserves parallelism, volume ratios
- Similarity
  - 7 dofs
  - Preserves angles, ratios of length
- Euclidean
  - 6 dofs
  - 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.
Let’s create a factorization from SVD:

Orthographic assumption:

\[ \mathbf{D} = \mathbf{A} \mathbf{X} \]

Refine structure and motion:

Goal: Estimate ambiguity matrix \( \mathbf{C} \)

Initialize structure

If we knew \( \mathbf{M} \) and \( \mathbf{S} \), we could solve for \( \mathbf{X} \).

This can be converted into a system of 3 equations:

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

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

Recap: Sequential Projective 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

Recap: Bundle Adjustment

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

Recap: Affine Structure from Motion

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

\[
\mathbf{D} = \begin{bmatrix}
\hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1n} \\
\hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{x}_{m1} & \hat{x}_{m2} & \cdots & \hat{x}_{mn}
\end{bmatrix} = \begin{bmatrix}
\mathbf{A}_1 \\
\mathbf{A}_2 \\
\vdots \\
\mathbf{A}_n
\end{bmatrix}
\]


Recap: Affine Factorization

- Obtaining a factorization from SVD:

\[
\mathbf{D} = \mathbf{U}_3 \mathbf{W}_3 \mathbf{V}_3^T
\]

Possible decomposition:

\[
\mathbf{D} = \mathbf{M} \mathbf{S}
\]

This decomposition minimizes \( |\mathbf{D} - \mathbf{M} \mathbf{S}|^2 \)

Recap: Projective Factorization

\[
\mathbf{D} = \begin{bmatrix}
\hat{z}_1 x_{11} & \hat{z}_1 x_{12} & \cdots & \hat{z}_1 x_{1n} \\
\hat{z}_2 x_{21} & \hat{z}_2 x_{22} & \cdots & \hat{z}_2 x_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{z}_m x_{m1} & \hat{z}_m x_{m2} & \cdots & \hat{z}_m x_{mn}
\end{bmatrix} = \begin{bmatrix}
\mathbf{P}_1 \\
\mathbf{P}_2 \\
\vdots \\
\mathbf{P}_n
\end{bmatrix}
\]

Points (4 \( \times \) n)

Cameras (3 \( \times \) m)

\( \mathbf{D} = \mathbf{M} \mathbf{S} \) has rank 4

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

Recap: Estimating the Euclidean Upgrade

- Goal: Estimate ambiguity matrix \( \mathbf{C} \)
  - Orthographic assumption:

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

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

Recap: Bundle Adjustment

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

\[
E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{n} D(x_{ik}, \mathbf{P}x_{jk})^2
\]
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:

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

- Minimum least squares solution given by solution of

\[
(ATA) d = ATb
\]

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 & upsample

Recall the Harris detector!
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

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:
  \[ P(X_t \mid y_0, \ldots, y_{t-1}) = \int P(X_t \mid X_{t-1})P(X_{t-1} \mid y_0, \ldots, y_{t-1}) \, dX_{t-1} \]
  - Dynamics model
  - Corrected estimate from previous step

- Correction:
  \[ P(X_t \mid y_0, \ldots, y_t) = \frac{P(Y_t \mid X_t)P(X_t \mid y_0, \ldots, y_{t-1})}{\int P(Y_t \mid X_t)P(X_t \mid y_0, \ldots, y_{t-1}) \, dX_t} \]
  - Observation model
  - Predicted estimate from previous step

Recap: Linear Dynamic Models

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

- Observation model
  - Measurement is linearly transformed state plus Gaussian noise
  \[ y_t \sim N(M_t x_t, \Sigma_m) \]

Recap: The Kalman Filter

- Know prediction of state, and next measurement update distribution over current state.
- Know corrected state from previous time step, and all measurements up to the current one
- Predict distribution over next state.

\[ P(X_{t+1} \mid y_0, \ldots, y_t) \]

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

Time advances: $t++$

Mean and std. dev. of corrected state:
\[ \mu_t^+, \Sigma_t^+ \]
Recap: General Kalman Filter (>1dim)

- What if state vectors have more than one dimension?

Predict

\[ x_t = D x_{t-1} \]
\[ \Sigma_t = D \Sigma_{t-1} D^T + \Sigma_n \]

Correct

\[ K_t = \Sigma_t M_t^T \left( M_t \Sigma_t M_t^T + \Sigma_m \right)^{-1} \]
\[ x_t = x_t + K \left( y_t - M x_t \right) \]
\[ \Sigma_t = (I - K_t M_t) \Sigma_t \]

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

Any Questions?

So what can you do with all of this?

Mobile Object Detection & Tracking

Mobile Visual Search & Mobile AR

Mobile Visual Search & Mobile AR

Efficient Large-Scale Localization

3D Model, reconstructed from photos

Query Image

Camera position

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.

Mining the World’s Images…
Automatic Landmark Building Discovery

Any More Questions?

See you in Machine Learning next semester...