

# Computer Vision - Lecture 22

## Repetition

09.02.2016

Bastian Leibe  
RWTH Aachen  
<http://www.vision.rwth-aachen.de>  
leibe@vision.rwth-aachen.de

## Announcements

- Exam
  - 1<sup>st</sup> Date: Monday, 29.02., 13:30 - 17:30h
  - 2<sup>nd</sup> Date: Thursday, 31.03., 09:30 - 12:30h
  - Closed-book exam, the core exam time will be 2h.
  - We will send around an announcement with the exact starting times and places by email.
- Test exam
  - Date: Thursday, 11.02., 14:15 - 15:45h, room UMIC 025
  - Core exam time will be 1h
  - Purpose: Prepare you for the questions you can expect.
  - Possibility to collect bonus exercise points!

B. Leibe

2

## Announcements (2)

- Feedback to the lecture evaluation

## Announcements (3)

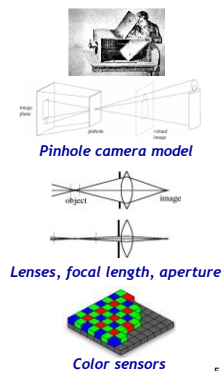
- Today, I'll summarize the most important points from the lecture.
  - It is an opportunity for you to ask questions...
  - ...or get additional explanations about certain topics.
  - So, please do ask.
- Today's slides are intended as an index for the lecture.
  - But they are not complete, won't be sufficient as only tool.
  - Also look at the exercises - they often explain algorithms in detail.

B. Leibe

4

## Repetition

- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

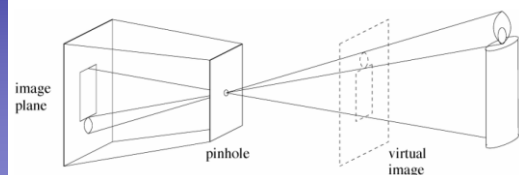


B. Leibe

5

## Recap: Pinhole Camera

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



Source: Forsyth & Ponce

B. Leibe

6

Computer Vision WS 15/16

## Recap: Focus and Depth of Field

Thin lens: scene points at distinct depths come in focus at different image planes.  
(Real camera lens systems have greater depth of field.)

"circles of confusion"

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

Source: Shapiro & Stockman B. Leibe 7

Computer Vision WS 15/16

## 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

B. Leibe from R. Durrainwami 8

Computer Vision WS 15/16

## Recap: Color Sensing in Digital Cameras

Bayer grid

Estimate missing components from neighboring values (demosaicing)

B. Leibe Source: Steve Seitz 9

Computer Vision WS 15/16

## Repetition

- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

B. Leibe 10

Computer Vision WS 15/16

## 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

B. Leibe Image Source: D. Kim et al., Cytometry 35(11), 1992 11

Computer Vision WS 15/16

## Recap: Robust Thresholding

Ideal histogram, light object on dark background

Actual observed histogram with noise

Assumption here: only two modes

Source: Robyn Owens B. Leibe 12

RWTH AACHEN  
UNIVERSITY

## Recap: Global Binarization [Otsu'79]

see  
Exercise 2.1!

- Precompute a cumulative grayvalue histogram  $h$ .
- For each potential threshold  $T$ 
  - Separate the pixels into two clusters according to  $T$ .
  - Compute both cluster means  $\mu_1(T)$  and  $\mu_2(T)$ .  
Look up  $n_1, n_2$  in  $h$   

$$n_1(T) = |\{I(x,y) < T\}|, \quad n_2(T) = |\{I(x,y) \geq T\}|$$
  - Compute the between-class variance  $\sigma_{\text{between}}^2(T)$   

$$\sigma_{\text{between}}^2(T) = n_1(T)n_2(T)[\mu_1(T) - \mu_2(T)]^2$$
- Choose the threshold that maximizes  

$$T^* = \arg \max_T [\sigma_{\text{between}}^2(T)]$$

Computer Vision WS 15/16

13

RWTH AACHEN  
UNIVERSITY

## Recap: Background Surface Fitting

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

Original image

Fitted surface

Shading compensation

Binarized result

Computer Vision WS 15/16

14

RWTH AACHEN  
UNIVERSITY

## Recap: Dilation

- Definition
  - "The dilation of  $A$  by  $B$  is the set of all displacements  $z$ , such that  $(B)_z$  and  $A$  overlap by at least one element".
  - $((B)_z)$  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

$A \oplus B_1$

$A \oplus B_2$

Computer Vision WS 15/16

15

RWTH AACHEN  
UNIVERSITY

## 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.
  - $\Rightarrow$  Erode connected components
  - $\Rightarrow$  Shrink features
  - $\Rightarrow$  Remove bridges, branches, noise

$A \ominus B_1$

$A \ominus B_2$

Computer Vision WS 15/16

16

RWTH AACHEN  
UNIVERSITY

## 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.

$A \circ B = \bigcup \{B_i | B_i \subseteq A\}$

$A \circ B = \bigcup \{B_i | B_i \subseteq A\}$

Computer Vision WS 15/16

17

RWTH AACHEN  
UNIVERSITY

## 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  $A$ .
  - $\Rightarrow$  Fill holes, keep original shape.

$A \cdot B = (A \oplus B) \ominus B$

$A \cdot B = (A \oplus B) \ominus B$

Computer Vision WS 15/16

18

Computer Vision WS 15/16

## Recap: Connected Components Labeling

- Process the image from left to right, top to bottom:
  - i.) If the next pixel to process is 1
    - 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

RWTH AACHEN UNIVERSITY

Slide credit: J. Neira

B. Leibe

19

Computer Vision WS 15/16

## 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

RWTH AACHEN UNIVERSITY

B. Leibe

20

Computer Vision WS 15/16

## Recap: Moment Invariants

see Exercise 2.3!

- Normalized central moments
 
$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^p}, \quad \gamma = \frac{p+q}{2} + 1$$
- From those, a set of *invariant moments* can be defined for object description.
 

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

(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).

RWTH AACHEN UNIVERSITY

B. Leibe

21

Computer Vision WS 15/16

## Repetition

- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

RWTH AACHEN UNIVERSITY

B. Leibe

22

Computer Vision WS 15/16

## 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.

RWTH AACHEN UNIVERSITY

B. Leibe

23

Computer Vision WS 15/16

## Recap: Gaussian Smoothing

see Exercise 2.4!

- 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

RWTH AACHEN UNIVERSITY

B. Leibe

24

**Recap: Smoothing with a Gaussian**

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

```

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

```

Slide credit: Kristen Grauman B. Leibe

**Recap: Resampling with Prior Smoothing**

Artifacts!

no smoothing

Gaussian  $\sigma = 1$

Gaussian  $\sigma = 2$

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

Computer Vision WS 15/16 B. Leibe Image Source: Forsyth & Ponce 26

**Recap: The Gaussian Pyramid**

Low resolution

High resolution

$G_4 = (G_3 * \text{gaussian}) \downarrow 2$

$G_3 = (G_2 * \text{gaussian}) \downarrow 2$

$G_2 = (G_1 * \text{gaussian}) \downarrow 2$

$G_1 = (G_0 * \text{gaussian}) \downarrow 2$

$G_0 = \text{Image}$

Source: Irani & Bassi 27

**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

Median value

Sort

Replace

Computer Vision WS 15/16 B. Leibe 28

**Recap: Derivatives and Edges...**

1st derivative

2nd derivative

“zero crossings” of second derivative

Maxima of first derivative

29

**Recap: 2D Edge Detection Filters**

Gaussian

Derivative of Gaussian

Laplacian of Gaussian

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

$\frac{\partial}{\partial x} h_\sigma(u, v)$

$\nabla^2 h_\sigma(u, v)$

- $\nabla^2$  is the Laplacian operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

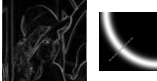
see Exercise 2.5f

Computer Vision WS 15/16 B. Leibe 30


Computer Vision WS 15/16

## Repetition

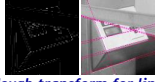
- Image Processing Basics
  - Image Formation
  - Binary Image Processing
  - Linear Filters
  - Edge & Structure Extraction
- Segmentation & Grouping
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking




Canny edge detector



Chamfer matching



Hough transform for lines



Hough transform for circles

31

B. Leibe

Computer Vision WS 15/16



## Recap: Canny Edge Detector

see Exercise 2.6!

- Filter image with derivative of Gaussian
- Find magnitude and orientation of gradient
- Non-maximum suppression:
  - Thin multi-pixel wide "ridges" down to single pixel width
- 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:**

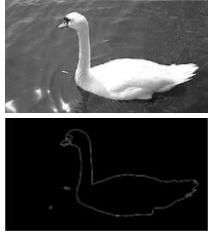
```
>> edge(image, 'canny');
>> help edge
```

adapted from D. Lowe, L. Fei-Fei

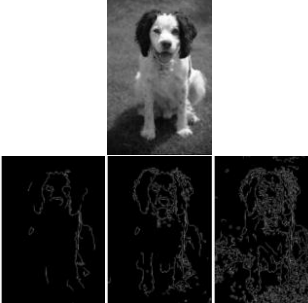
Computer Vision WS 15/16

## 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.

33

Slide credit: Kristen Grauman



Computer Vision WS 15/16

## Recap: Chamfer Matching

- Chamfer Distance
  - Average distance to nearest feature

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

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

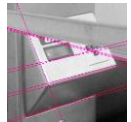




Edge image      Distance transform image

[D. Gavrilu, DAGM'99]


Computer Vision WS 15/16


## 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.

35

Slide credit: Kristen Grauman

Computer Vision WS 15/16

## Recap: Hough Transform

see Exercise 3.1!

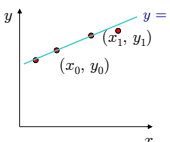
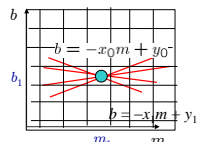


Image space



Hough (parameter) space

- 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.

36

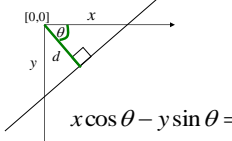
Slide credit: Steve Seitz



Computer Vision WS 15/16

## Recap: Hough Transf. Polar Parametrization

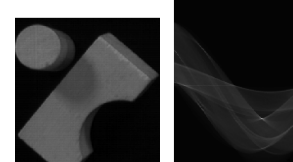
- Usual  $(m,b)$  parameter space problematic: can take on infinite values, undefined for vertical lines.



$d$  : perpendicular distance from line to origin  
 $\theta$  : angle the perpendicular makes with the x-axis

$$x \cos \theta - y \sin \theta = d$$

- Point in image space  $\Rightarrow$  sinusoid segment in Hough space



Slide credit: Steve Seitz

Computer Vision WS 15/16

## Recap: Hough Transform for Circles

see Exercise 3.1!

- 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

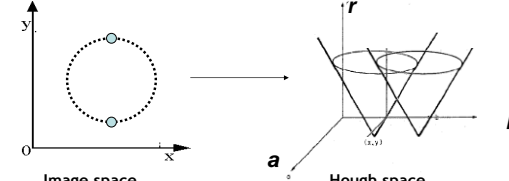


Image space

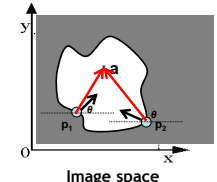
Hough space

Slide credit: Kristen Grauman

Computer Vision WS 15/16

## 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  $\theta$ .

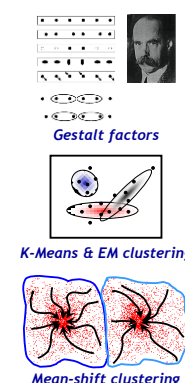
[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]

Slide credit: Kristen Grauman

Computer Vision WS 15/16

## Repetition

- Image Processing Basics
- Segmentation & Grouping
  - Segmentation and Grouping
  - Segmentation as Energy Minimization
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking



Gestalt factors

K-Means & EM clustering

Mean-shift clustering

B. Leibe

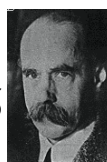
Computer Vision WS 15/16

## 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)

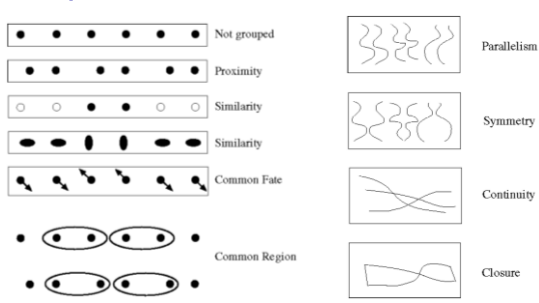


Untersuchungen zur Lehre von der Gestalt, *Psychologische Forschung*, Vol. 4, pp. 301-350, 1923  
<http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm>

B. Leibe

Computer Vision WS 15/16

## Recap: Gestalt Factors



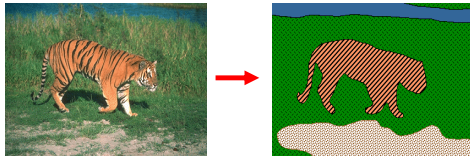
- These factors make intuitive sense, but are very difficult to translate into algorithms.

B. Leibe

Image source: Forsyth & Ponce

## Recap: Image Segmentation

- Goal: identify groups of pixels that go together



## Recap: K-Means Clustering

- Basic idea: randomly initialize the  $k$  cluster centers,  $c_1, \dots, c_k$ , and iterate between the two following steps

1. Randomly initialize the cluster centers,  $c_1, \dots, c_k$
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

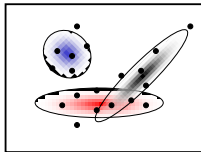


### Properties

- > 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)

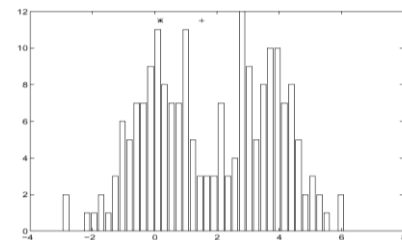


- Goal
  - > Find blob parameters  $\theta$  that maximize the likelihood function:

$$p(\text{data}|\theta) = \prod_{n=1}^N p(\mathbf{x}_n|\theta)$$

- Approach:
  1. E-step: given current guess of blobs, compute ownership of each point
  2. M-step: given ownership probabilities, update blobs to maximize likelihood function
  3. Repeat until convergence

## Recap: Mean-Shift Algorithm

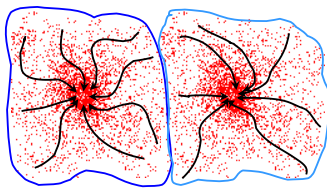


### Iterative Mode Search

1. Initialize random seed, and window  $W$
2. Calculate center of gravity (the "mean") of  $W$ :  $\sum_{x \in W} xH(x)$
3. Shift the search window to the mean
4. Repeat Step 2 until convergence

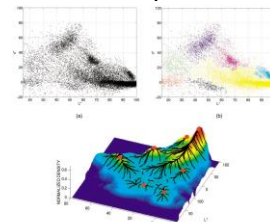
## Recap: Mean-Shift Clustering

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



## Recap: Mean-Shift Segmentation

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





Computer Vision WS 15/16

## Repetition

- Image Processing Basics
- Segmentation & Grouping
  - Segmentation and Grouping
  - Segmentation as Energy Minimization
- Object Recognition
- Local Features & Matching
- Object Categorization
- 3D Reconstruction
- Motion and Tracking

Markov Random Fields

Graph cuts

49

B. Leibe

Computer Vision WS 15/16

## Recap: MRFs for Image Segmentation

- MRF formulation
  - Unary potentials  $\phi(x_i, y_i)$
  - Pairwise potentials  $\psi(x_i, x_j)$

⇒ Minimize the energy

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

Data (D)    Unary likelihood    Pair-wise Terms    MAP Solution

50

Slide adapted from Phil Torr

Computer Vision WS 15/16

## Recap: Energy Formulation

- Energy function
 
$$E(\mathbf{x}, \mathbf{y}) = \sum_i \underbrace{\phi(x_i, y_i)}_{\text{Unary potentials}} + \sum_{i,j} \underbrace{\psi(x_i, x_j)}_{\text{Pairwise potentials}}$$
- Unary 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)

51

B. Leibe

Computer Vision WS 15/16

## Recap: How to Set the Potentials?

- Unary potentials
  - E.g. color model, modeled with a Mixture of Gaussians
 
$$\phi(x_i, y_i; \theta_\phi) = \log \sum_k \theta_\phi(x_i, k) p(k|x_i) \mathcal{N}(y_i; \bar{y}_k, \Sigma_k)$$

⇒ Learn color distributions for each label

52

B. Leibe

Computer Vision WS 15/16

## Recap: How to Set the Potentials?

- Pairwise potentials
  - Potts Model
 
$$\psi(x_i, x_j; \theta_\psi) = \theta_\psi \delta(x_i \neq x_j)$$
    - Simplest discontinuity preserving model.
    - Discontinuities between any pair of labels are penalized equally.
    - Useful when labels are unordered or number of labels is small.
  - Extension: "Contrast sensitive Potts model"
 
$$\psi(x_i, x_j, g_{ij}(y); \theta_\psi) = \theta_\psi g_{ij}(y) \delta(x_i \neq x_j)$$

where

$$g_{ij}(y) = e^{-\beta \|y_i - y_j\|^2} \quad \beta = 2 / \text{avg}(\|y_i - y_j\|^2)$$

⇒ Discourages label changes except in places where there is also a large change in the observations.

53

B. Leibe

Computer Vision WS 15/16

## Recap: Graph-Cuts Energy Minimization

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

⇒ Solution is equivalent to minimum of the energy.
- s-t Mincut can be solved efficiently
  - Dual to the well-known max flow problem
  - Very efficient algorithms available for regular grid graphs (1-2 MPixels/s)
  - Globally optimal result for 2-class problems

54

see Exercise 3.4!

## Recap: When Can s-t Graph Cuts Be Applied?

$$E(L) = \sum_p E_p(L_p) + \sum_{p,q \in N} E(L_p, L_q)$$

Unary potentials      Pairwise potentials  
t-links                      n-links       $L_p \in \{s, t\}$

- s-t graph cuts can only globally minimize **binary energies** that are **submodular**. [Boros & Hummer, 2002, Kolmogorov & Zabih, 2004]

$$E(L) \text{ can be minimized by s-t graph cuts} \iff E(s,s) + E(t,t) \leq E(s,t) + E(t,s)$$

Submodularity ("convexity")

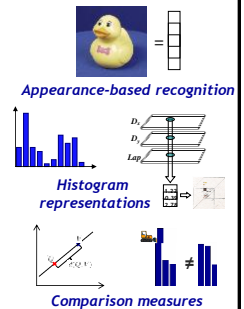
- Submodularity is the discrete equivalent to convexity.
  - Implies that every local energy minimum is a global minimum.
  - ⇒ Solution will be globally optimal.

B. Leibe

55

## Repetition

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

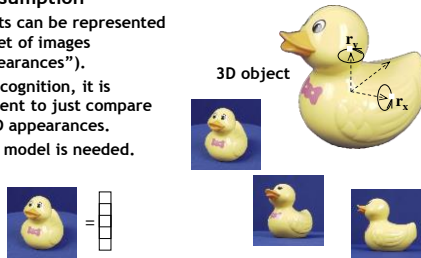


B. Leibe

56

## 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.



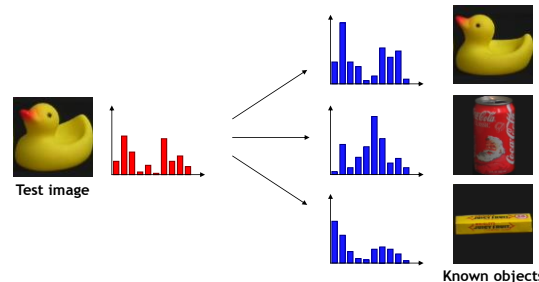
⇒ Fundamental paradigm shift in the 90's

B. Leibe

57

## Recap: Recognition Using Global Features

- E.g. histogram comparison



B. Leibe

58

## 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)



59

## Recap: Recognition Using Histograms

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

"Nearest-Neighbor" strategy

B. Leibe

60

Computer Vision WS 15/16

## Recap: Multidimensional Representations

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

$D_x$   
 $D_y$   
 $Lap$

61

Computer Vision WS 15/16

## Recap: Colored Derivatives

- Generalization: derivatives along
  - Y axis → intensity differences
  - C<sub>1</sub> axis → red-green differences
  - C<sub>2</sub> axis → blue-yellow differences
- Application:
  - Brand identification in video

62

Computer Vision WS 15/16

## First Applications Take Up Shape...

Line detection

Histogram based recognition

Circle detection

Binary Segmentation

Skin color detection

Moment descriptors

Image Source: <http://www.flickr.com/photos/angelisk/2806412807/>

63

Computer Vision WS 15/16

## 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

$$M(\sigma_1, \sigma_2) = g(\sigma_1) * \begin{bmatrix} I_x'(\sigma_1) & I_x I_y'(\sigma_1) \\ I_y I_x'(\sigma_1) & I_y'(\sigma_1) \end{bmatrix}$$

Harris & Hessian detector

$$Hes(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

Laplacian scale selection

Difference-of-Gaussian (DoG)

SIFT descriptor

64

Computer Vision WS 15/16

## Recap: Local Feature Matching Pipeline

- Find a set of distinctive key-points
- Define a region around each keypoint
- Extract and normalize the region content
- Compute a local descriptor from the normalized region
- Match local descriptors

Similarity measure

$d(f_A, f_B) < T$

65

Computer Vision WS 15/16

## Recap: Requirements for Local Features

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

We need a repeatable detector!

We need a reliable and distinctive descriptor!

66

Computer Vision WS 15/16

## Recap: Harris Detector [Harris88]

see Exercise 5.2!

- Compute second moment matrix (autocorrelation matrix)

$$M(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

- Image derivatives
- Square of derivatives
- Gaussian filter  $g(\sigma)$

4. Cornerness function - two strong eigenvalues

$$R = \det[M(\sigma_I, \sigma_D)] - \alpha [\text{trace}(M(\sigma_I, \sigma_D))]^2$$

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

5. Perform non-maximum suppression

Slide credit: Krystian Mikolajczyk B. Leibe

Computer Vision WS 15/16

## Recap: Harris Detector Responses [Harris88]

Effect: A very precise corner detector.

Slide credit: Krystian Mikolajczyk

Computer Vision WS 15/16

## Recap: Hessian Detector [Beaudet78]

see Exercise 5.1!

- Hessian determinant

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

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

In Matlab:

$$I_{xx} * I_{yy} - (I_{xy})^2$$

Slide credit: Krystian Mikolajczyk B. Leibe

Computer Vision WS 15/16

## Recap: Hessian Detector Responses [Beaudet78]

Effect: Responses mainly on corners and strongly textured areas.

Slide credit: Krystian Mikolajczyk

Computer Vision WS 15/16

## Recap: Automatic Scale Selection

- Function responses for increasing scale (scale signature)

Slide credit: Krystian Mikolajczyk B. Leibe

Computer Vision WS 15/16

## Recap: Laplacian-of-Gaussian (LoG)

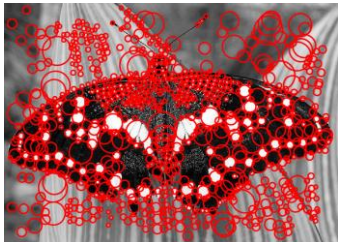
- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

⇒ List of  $(x, y, \sigma)$

Slide adapted from Krystian Mikolajczyk B. Leibe

Computer Vision WS 15/16

## Recap: LoG Detector Responses



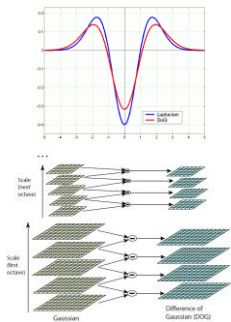
Slide credit: Svetlana Lazebnik B. Leibe

73

Computer Vision WS 15/16

## 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



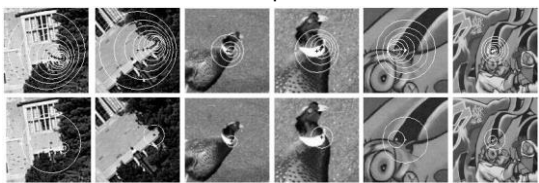
Candidate keypoints:  
list of  $(x, y, \sigma)$

Image source: David Lowe

Computer Vision WS 15/16

## Recap: Harris-Laplace [Mikolajczyk '01]

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



Harris points

Harris-Laplace points

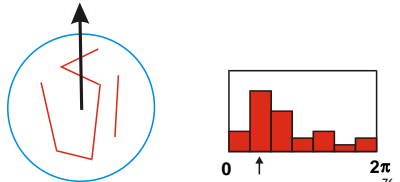
Slide adapted from Krystian Mikolajczyk B. Leibe

75

Computer Vision WS 15/16

## Recap: Orientation Normalization

- Compute orientation histogram [Lowe, SIFT, 1999]
- Select dominant orientation
- Normalize: rotate to fixed orientation

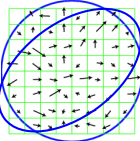


Slide adapted from David Lowe

Computer Vision WS 15/16

## Recap: Affine Adaptation

- Problem:
  - Determine the characteristic shape of the region.
  - Assumption: shape can be described by "local affine frame".
- Solution: iterative approach
  - Use a circular window to compute second moment matrix.
  - Compute eigenvectors to adapt the circle to an ellipse.
  - Recompute second moment matrix using new window and iterate...




Slide adapted from Svetlana Lazebnik B. Leibe

77

Computer Vision WS 15/16

## Recap: Iterative Affine Adaptation



1. Detect keypoints, e.g. multi-scale Harris
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, IJCV 60(1):63-86, 2004. Slide credit: Tinne Tuytelaars

78



**Recap: Affine-Inv. Feature Extraction**

Extract affine regions → Normalize regions → Eliminate rotational ambiguity → Compare descriptors

Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

79

**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

David G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV 60 (2), pp. 91-110, 2004.

Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

80

**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

Computer Vision WS 15/16

B. Leibe

81

**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

Local Features, e.g. SIFT

Computer Vision WS 15/16

Slide credit: David Lowe

B. Leibe

82

**Recap: Indexing features**

Detect or sample features → Describe features → List of positions, scales, orientations → Associated list of d-dimensional descriptors → Index each one into pool of descriptors from previously seen images → Match to quantized descriptors (visual words)

⇒ Shortlist of possibly matching images + feature correspondences

Computer Vision WS 15/16

Slide credit: Kristen Grauman

B. Leibe

83

**Recap: Fast Indexing with Vocabulary Trees**

- **Recognition**

Geometric verification

[Nister & Stewenius, CVPR'06]

Computer Vision WS 15/16

Slide credit: David Nister

B. Leibe

84



Computer Vision WS 15/16

## Recap: Fitting an Affine Transformation

- Assuming we know the correspondences, how do we get the transformation?

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

$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} x'_i \\ y'_i \\ \vdots \end{bmatrix}$$

B. Leibe

85

Computer Vision WS 15/16

## Recap: Fitting a Homography

- Estimating the transformation

Homogenous coordinates:  $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Image coordinates:  $\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \begin{bmatrix} 1/z' \\ 1/z' \\ 1/z' \end{bmatrix}$

Matrix notation:  $x' = Hx$ ,  $x'' = \frac{1}{z'} x'$

$$x_A = \frac{h_{11}x_B + h_{12}y_B + h_{13}}{h_{31}x_B + h_{32}y_B + 1}, y_A = \frac{h_{21}x_B + h_{22}y_B + h_{23}}{h_{31}x_B + h_{32}y_B + 1}$$

Slide credit: Krystian Mikolajczyk

B. Leibe

86

Computer Vision WS 15/16

## Recap: Fitting a Homography

- Estimating the transformation

see Exercise 5.5!

$$\begin{bmatrix} h_{11}x_B + h_{12}y_B + h_{13} - x_A h_{11} - x_A h_{12}y_B - x_A \\ h_{21}x_B + h_{22}y_B + h_{23} - y_A h_{11} - y_A h_{12}y_B - y_A \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \end{bmatrix}$$

$$Ah = 0$$

Slide credit: Krystian Mikolajczyk

B. Leibe

87

Computer Vision WS 15/16

## Recap: Fitting a Homography

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

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

$h = \begin{bmatrix} v_{19} & \dots & v_{99} \end{bmatrix}^T$  Minimizes least square error

Slide credit: Krystian Mikolajczyk

B. Leibe

88

Computer Vision WS 15/16

## Recap: RANSAC

see Exercise 6.2!

**RANSAC loop:**

- Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
- Compute transformation from seed group
- Find *inliers* to this transformation
- 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

Slide credit: Kristen Grauman

B. Leibe

89

Computer Vision WS 15/16

## Recap: RANSAC Line Fitting Example

- Task: Estimate the best line

Slide credit: Jinxing Chai

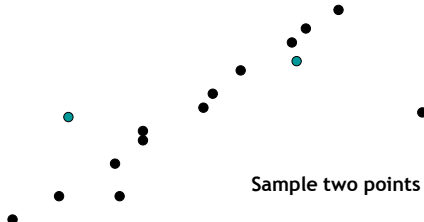
B. Leibe

90

Computer Vision WS 15/16

## Recap: RANSAC Line Fitting Example

- Task: Estimate the best line



Sample two points

Slide credit: Jinyang Chai

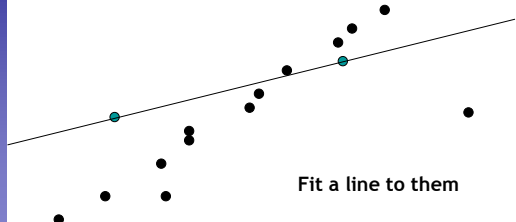
B. Leibe

91

Computer Vision WS 15/16

## Recap: RANSAC Line Fitting Example

- Task: Estimate the best line



Fit a line to them

Slide credit: Jinyang Chai

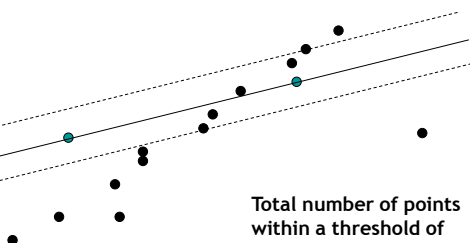
B. Leibe

92

Computer Vision WS 15/16

## Recap: RANSAC Line Fitting Example

- Task: Estimate the best line



Total number of points within a threshold of line.

Slide credit: Jinyang Chai

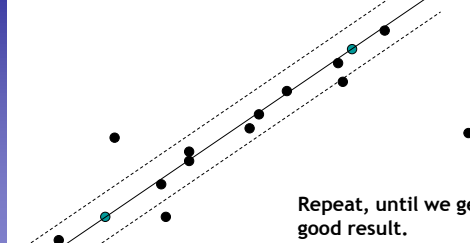
B. Leibe

93

Computer Vision WS 15/16

## Recap: RANSAC Line Fitting Example

- Task: Estimate the best line



Repeat, until we get a good result.

Slide credit: Jinyang Chai

B. Leibe

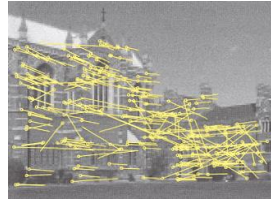
94

Computer Vision WS 15/16

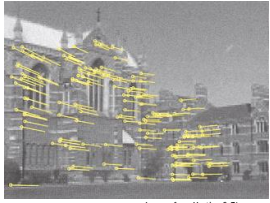
## Recap: Feature Matching Example

- Find best stereo match within a square search window (here 300 pixels<sup>2</sup>)
- Global transformation model: epipolar geometry

before RANSAC



after RANSAC



Images from Hartley & Zisserman

Slide credit: David Lowe

B. Leibe

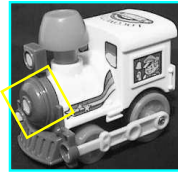
95


Computer Vision WS 15/16

## Recap: Generalized Hough Transform

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

model





Slide credit: Svetlana Lazebnik

B. Leibe

96

Computer Vision WS 15/16

## 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.

model

Slide credit: Svetlana Lazebnik

B. Leibe

97

Computer Vision WS 15/16

## Application: Panorama Stitching

see Panorama Demo!

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

B. Leibe

98

Computer Vision WS 15/16

## Repetition

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

Sliding window principle

Boosting

SVM

HOG detector

Viola-Jones face detector

B. Leibe

99

Computer Vision WS 15/16

## Recap: Sliding-Window Object Detection

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

Car/non-car Classifier

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

Slide credit: Kristen Grauman

B. Leibe

100

Computer Vision WS 15/16

## Recap: Gradient-based Representations

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

- Summarize local distribution of gradients with histogram

Slide credit: Kristen Grauman

B. Leibe

101

Computer Vision WS 15/16

## Recap: Classifier Construction: Many Choices...

Nearest Neighbor

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

Neural networks

LeCun, Bottou, Bengio, Haffner 1998, Rowley, Baluja, Kanade 1998, ...

Boosting

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

Support Vector Machines

Vapnik, Schölkopf 1995, Papageorgiou, Poggio '01, Dalal, Triggs 2005, Vedaldi, Zisserman 2012

Randomized Forests

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

Slide adapted from Kristen Grauman

B. Leibe

102

Computer Vision WS 15/16

## 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

Slide credit: Kristen Grauman B. Leibe

103

Computer Vision WS 15/16

## 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:

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

104

Computer Vision WS 15/16

## Recap: HOG Descriptor Processing Chain

- SVM Classification
  - Typically using a linear SVM

Slide adapted from Naveet Dalal

105

Computer Vision WS 15/16

## Recap: HOG Cell Computation Details

- Gradient orientation voting
  - Each pixel contributes to localized gradient orientation histogram(s)
  - Vote is weighted by the pixel's gradient magnitude
$$\theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

$$\|\nabla f\| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}$$
- Block-level Gaussian weighting
  - An additional Gaussian weight is applied to each  $2 \times 2$  block of cells
  - Each cell is part of 4 such blocks, resulting in 4 versions of the histogram.

106

Computer Vision WS 15/16

## Recap: HOG Cell Computation Details (2)

- Important for robustness: **Tri-linear interpolation**
  - Each pixel contributes to (up to) 4 neighboring cell histograms
  - Weights are obtained by **bilinear interpolation in image space**:
 
$$h(x_1, y_1) \leftarrow w \cdot \left( 1 - \frac{x - x_1}{x_2 - x_1} \right) \left( 1 - \frac{y - y_1}{y_2 - y_1} \right)$$

$$h(x_1, y_2) \leftarrow w \cdot \left( 1 - \frac{x - x_1}{x_2 - x_1} \right) \left( \frac{y - y_1}{y_2 - y_1} \right)$$

$$h(x_2, y_1) \leftarrow w \cdot \left( \frac{x - x_1}{x_2 - x_1} \right) \left( 1 - \frac{y - y_1}{y_2 - y_1} \right)$$

$$h(x_2, y_2) \leftarrow w \cdot \left( \frac{x - x_1}{x_2 - x_1} \right) \left( \frac{y - y_1}{y_2 - y_1} \right)$$
  - Contribution is further split over (up to) 2 neighboring orientation bins via **linear interpolation over angles**.

107

Computer Vision WS 15/16

## Recap: Non-Maximum Suppression

Image source: Naveet Dalal, PhD Thesis

108

Computer Vision WS 15/16

## Recap: AdaBoost

Weak Classifier 1

Weak Classifier 2

Weak Classifier 3

Weights Increased

Final classifier is combination of the weak classifiers

Slide credit: Kristen Grauman

B. Leibe

109

Computer Vision WS 15/16

## Recap: Viola-Jones Face Detection

"Rectangular" filters

Feature output is difference between adjacent regions

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

Integral image

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

B. Leibe

110

Computer Vision WS 15/16

## 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.

Slide credit: Kristen Grauman

B. Leibe

111

Computer Vision WS 15/16

## Application: Viola-Jones Face Detector

Train cascade of classifiers with AdaBoost

Apply to each subwindow

Slide credit: Kristen Grauman

B. Leibe

112

Computer Vision WS 15/16

## Repetition

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

Bag-of-words representation

Activation histogram

Implicit Shape Model

Slide credit: Kristen Grauman

B. Leibe

113

Computer Vision WS 15/16

## Recap: Identification vs. Categorization

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

Slide credit: Kristen Grauman

B. Leibe

114



**Recap: Visual Words**

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

Exact feature matching → Match to same visual word

Slide adapted from Kristen Grauman  
Figure from Sivic & Zisserman, ICCV 2003

**Recap: Bag-of-Word Representations (BoW)**

Object → Bag of "words"

116  
B. Leibe  
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).

Violins

Slide adapted from Kristen Grauman  
B. Leibe

**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  
B. Leibe

**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?

Slide adapted from Bill Freeman  
B. Leibe

**Recap: Part-Based Models**

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

120  
B. Leibe



Computer Vision WS 15/16

## 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

Appearance codebook

Spatial occurrence distributions  
+ local figure-ground labels

B. Leibe

Computer Vision WS 15/16

## Recap: Implicit Shape Model - Recognition

Interest Points

Matched Codebook Entries

Probabilistic Voting

"Generalized Hough Transform with backprojection"

3D Voting Space (continuous)

Backprojected Hypotheses

Backprojection of Maxima

122

(Leibe, Leonardis, Schiele, SLCV'04; UCV'08)

Computer Vision WS 15/16

## 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 3<sup>rd</sup> dimension in voting space

$$x_{vote} = x_{img} - x_{occ}(s_{img}/s_{occ})$$

$$y_{vote} = y_{img} - y_{occ}(s_{img}/s_{occ})$$

$$s_{vote} = (s_{img}/s_{occ})$$

- Search for maxima in 3D voting space

Search window

B. Leibe

Computer Vision WS 15/16

## Repetition

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

Convolutional Neural Networks

Convolution layers

Pooling layers

AlexNet, VGGNet, GoogLeNet

124

Computer Vision WS 15/16

## Recap: Convolutional Neural Networks

"LeNet" architecture

- Neural network with specialized connectivity structure
  - Stack multiple stages of feature extractors
  - Higher stages compute more global, more invariant features
  - Classification layer at the end

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, [Gradient-based learning applied to document recognition](#), Proceedings of the IEEE 86(11): 2278-2324, 1998.

Slide credit: Svetlana Lazebnik

B. Leibe

125

Computer Vision WS 15/16

## Recap: CNN Structure

- Feed-forward feature extraction
  1. Convolve input with learned filters
  2. Non-linearity
  3. Spatial pooling
  4. (Normalization)
- Supervised training of convolutional filters by back-propagating classification error

Feature maps

Normalization

Spatial pooling

Non-linearity

Convolution (Learned)

Input Image

Slide credit: Svetlana Lazebnik

B. Leibe

126

Computer Vision WS 15/16

## Recap: Intuition of CNNs

- Convolutional net**
  - Share the same parameters across different locations
  - Convolutions with learned kernels
- Learn multiple filters**
  - E.g. 1000x1000 image
  - 100 filters
  - 10x10 filter size
  - ⇒ only 10k parameters
- Result: Response map**
  - size: 1000x1000x100
  - Only memory, not params!

Slide adapted from Marc'Aurelio Ranzato. B. Leibe. Image source: Yann LeCun.

Computer Vision WS 15/16

## Recap: Convolution Layers

**Naming convention:**

HEIGHT  
WIDTH  
DEPTH

- All Neural Net activations arranged in 3 dimensions
  - Multiple neurons all looking at the same input region, stacked in depth
  - Form a single [1x1xdepth] depth column in output volume.

Slide credit: FeiFei Li, Andrei Karpathy. B. Leibe.

Computer Vision WS 15/16

## Recap: Activation Maps

Activations: one filter = one depth slice (or activation map)

5x5 filters

Each activation map is a depth slice through the output volume.

Activation maps

Slide adapted from FeiFei Li, Andrei Karpathy. B. Leibe.

Computer Vision WS 15/16

## Recap: Pooling Layers

Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2

6	8
3	4

- Effect:**
  - Make the representation smaller without losing too much information
  - Achieve robustness to translations

Slide adapted from FeiFei Li, Andrei Karpathy. B. Leibe.

Computer Vision WS 15/16

## Recap: Effect of Multiple Convolution Layers

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Slide credit: Yann LeCun. B. Leibe.

Computer Vision WS 15/16

## Recap: AlexNet (2012)

- Similar framework as LeNet, but
  - Bigger model (7 hidden layers, 650k units, 60M parameters)
  - More data (10<sup>6</sup> images instead of 10<sup>3</sup>)
  - GPU implementation
  - Better regularization and up-to-date tricks for training (Dropout)

A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](#), NIPS 2012.

Image source: A. Krizhevsky, I. Sutskever and G.F. Hinton. NIPS 2012.

## Recap: VGGNet (2014/15)

- **Main ideas**
  - Deeper network
  - Stacked convolutional layers with smaller filters (+ nonlinearity)
  - Detailed evaluation of all components
- **Results**
  - Improved ILSVC top-5 error rate to 6.7%.

ConvNet Configuration				
A	A-LRN	B	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	19 weight layers
conv-3-64	conv-3-64	input (224 x 224 RGB images)		
	LRN	conv-3-64	conv-3-64	conv-3-64
conv-3-128	conv-3-128	maxpool	conv-3-128	conv-3-128
		conv-3-128	conv-3-128	conv-3-128
conv-3-256	conv-3-256	maxpool	conv-3-256	conv-3-256
conv-3-256	conv-3-256	conv-3-256	conv-3-256	conv-3-256
		conv-3-256	conv-3-256	conv-3-256
conv-3-512	conv-3-512	maxpool	conv-3-512	conv-3-512
conv-3-512	conv-3-512	conv-3-512	conv-3-512	conv-3-512
		conv-3-512	conv-3-512	conv-3-512
conv-3-512	conv-3-512	maxpool	conv-3-512	conv-3-512
conv-3-512	conv-3-512	conv-3-512	conv-3-512	conv-3-512
		conv-3-512	conv-3-512	conv-3-512
		maxpool	conv-3-512	conv-3-512
		FC-4096	conv-3-512	conv-3-512
		FC-4096	conv-3-512	conv-3-512
		FC-1000	conv-3-512	conv-3-512
		softmax	conv-3-512	conv-3-512

B. Leibe

Image source: Simonyan & Zisserman

133

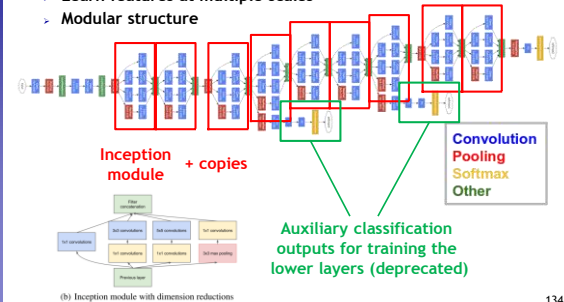
Slide credit: Andrei Karnathy

B. Leibe

135

## Recap: GoogLeNet (2014)

- Ideas:
  - Learn features at multiple scales
  - Modular structure



B. Leibe

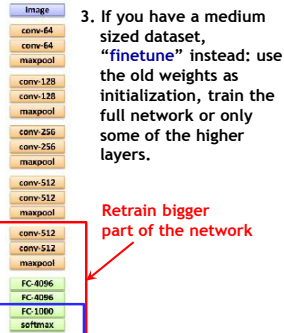
Image source: Szepedy et al.

134

## Recap: Transfer Learning with CNNs

1. Train on ImageNet
2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

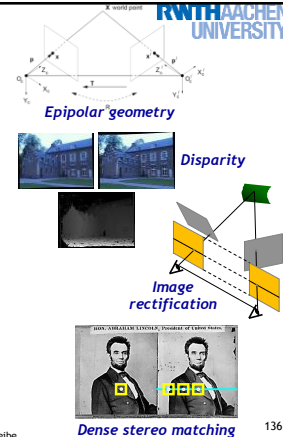
I.e., replace the Softmax layer at the end \

Retrain bigger  
part of the network

B. Leibe

## 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



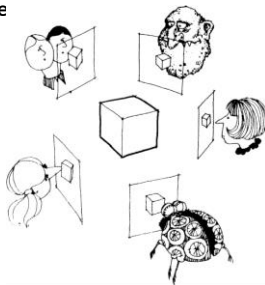
B. Leibe

### Dense stereo matching

136

## Recap: What Is Stereo Vision?

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

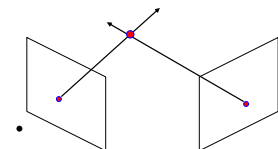


B. Leibe

Slide credit: Svetlana Lazebnik, Steve Seitz

137

## Recap: Depth with Stereo - Basic Idea



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

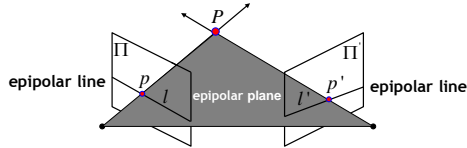
B. Leibe

Slide credit: Steve Seitz

138

## 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.

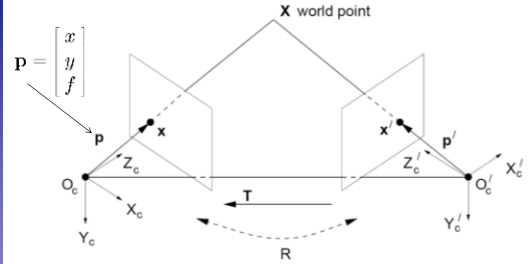
Computer Vision WS 15/16

Slide adapted from Steve Seitz

B. Leibe

139

## Recap: Stereo Geometry With Calibrated Cameras



- Camera-centered coordinate systems are related by known rotation  $R$  and translation  $T$ :

$$X' = RX + T$$

Computer Vision WS 15/16

Slide credit: Kristen Grauman

B. Leibe

140

## Recap: Essential Matrix

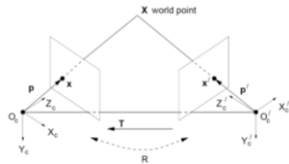
$$X' \cdot (T \times RX) = 0$$

$$X' \cdot (T \times RX) = 0$$

Let  $E = T \times R$

$$X'^T E X = 0$$

- 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]



Computer Vision WS 15/16

Slide credit: Kristen Grauman

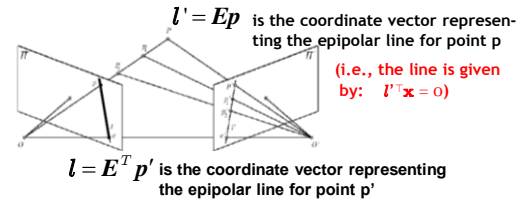
B. Leibe

141

## Recap: Essential Matrix and Epipolar Lines

$$p'^T E p = 0$$

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



$l = E^T p'$  is the coordinate vector representing the epipolar line for point  $p'$

Computer Vision WS 15/16

Slide credit: Kristen Grauman

B. Leibe

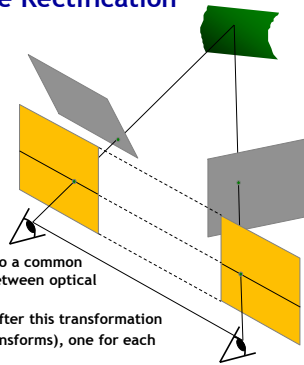
142

## 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

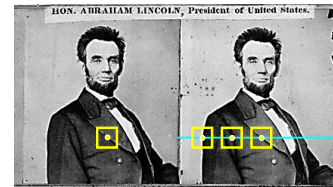


Computer Vision WS 15/16

Slide adapted from Li Zhang C. Loop & Z. Zhang, Computing Rectifying Homographies for Stereo Vision, CVPR'99

143

## Recap: Dense Correspondence Search



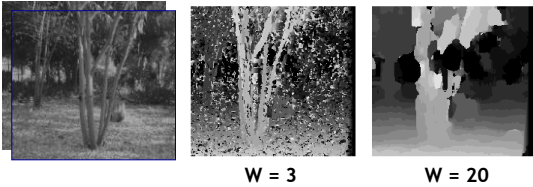
- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
  - Triangulate the matches to get depth information
- This is easiest when epipolar lines are scanlines
  - ⇒ Rectify images first

Computer Vision WS 15/16

adapted from Svetlana Lazebnik, Li Zhang

144

## 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.

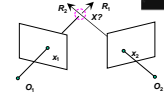
## Repetition

- 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

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Camera models

Camera calibration



Triangulation

Essential matrix, Fundamental matrix

$$x^T E x' = 0$$

$$x^T F x' = 0$$

$$\begin{bmatrix} u_1^T & u_2^T & u_3^T & u_4^T & u_5^T & u_6^T & u_7^T & u_8^T & u_9^T & u_{10}^T & u_{11}^T & u_{12}^T \\ v_1^T & v_2^T & v_3^T & v_4^T & v_5^T & v_6^T & v_7^T & v_8^T & v_9^T & v_{10}^T & v_{11}^T & v_{12}^T \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \\ F_{41} \\ F_{42} \\ F_{43} \end{bmatrix} = 0$$

Eight-point algorithm

## Recap: A General Point

- Equations of the form

$$Ax = 0$$

- How do we solve them? (always!)

- Apply SVD

$$A = UDV^T = U \begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & d_{NN} \end{bmatrix} \begin{bmatrix} v_{11} & \dots & v_{1N} \\ \vdots & \ddots & \vdots \\ v_{N1} & \dots & v_{NN} \end{bmatrix}^T$$

Singular values Singular vectors

- 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  $A$ .

## Recap: Camera Parameters

- Intrinsic parameters

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

$$K = \begin{bmatrix} m_x & 0 & p_x \\ 0 & m_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & s & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s' & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic parameters

- Rotation  $R$
- Translation  $t$  (both relative to world coordinate system)

- Camera projection matrix

$$P = K[R | t]$$

- General pinhole camera: 9 DoF
- CCD Camera with square pixels: 10 DoF
- General camera: 11 DoF

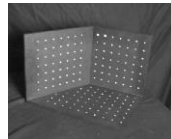
## Recap: Calibrating a Camera

### 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  $P = P_{\text{int}} P_{\text{ext}}$



P?

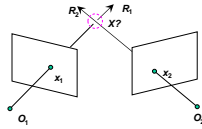
## Recap: Camera Calibration (DLT Algorithm)

$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \vdots & \vdots & \vdots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \quad Ap = 0$$

- $P$  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 1/2 correspondences needed for a minimal solution.

## Recap: Triangulation - Lin. Alg. Approach

see  
Exercise 6.3!



$$\begin{aligned} \lambda_1 x_1 &= P_1 X & x_1 \times P_1 X &= 0 & [x_1]_{\times} P_1 X &= 0 \\ \lambda_2 x_2 &= P_2 X & x_2 \times P_2 X &= 0 & [x_2]_{\times} P_2 X &= 0 \end{aligned}$$

- 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.

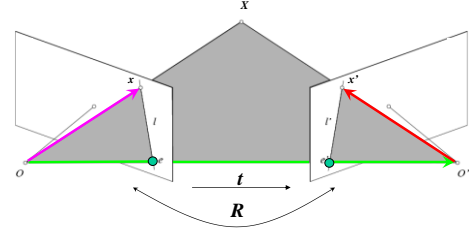
Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

151

## 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  $x$ ,  $l$ , and  $Rx'$  are coplanar

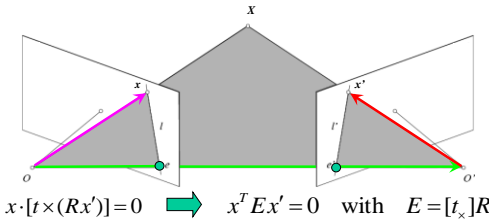
Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

152

## Recap: Epipolar Geometry - Calibrated Case



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

Essential Matrix  
(Longuet-Higgins, 1981)

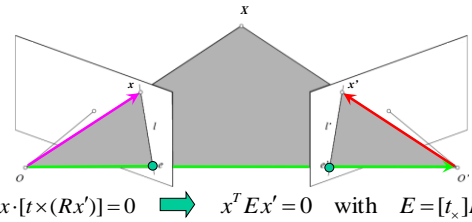
Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

153

## Recap: Epipolar Geometry - Calibrated Case



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

- $E x'$  is the epipolar line associated with  $x'$  ( $l = E x'$ )
- $E^T x$  is the epipolar line associated with  $x$  ( $l' = E^T x$ )
- $E e' = 0$  and  $E^T e = 0$
- $E$  is singular (rank two)
- $E$  has five degrees of freedom (up to scale)

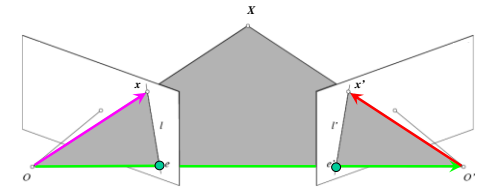
Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

154

## 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:

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

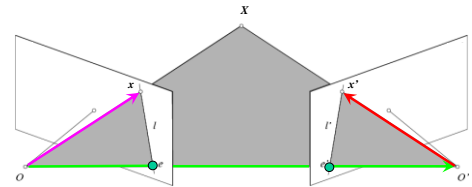
Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

155

## Recap: Epipolar Geometry - Uncalibrated Case



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

$$\begin{aligned} x &= K \hat{x} \\ x' &= K' \hat{x}' \end{aligned}$$

Fundamental Matrix  
(Faugeras and Luong, 1992)

Computer Vision WS 15/16

Slide credit: Svetlana Lazebnik

B. Leibe

156



### Recap: Epipolar Geometry - Uncalibrated Case

$\hat{x}^T E \hat{x}' = 0 \Rightarrow x^T F x' = 0$  with  $F = K^{-T} E K'^{-1}$

- $F x'$  is the epipolar line associated with  $x'$  ( $l = F x'$ )
- $F^T x$  is the epipolar line associated with  $x$  ( $l' = F^T x$ )
- $F e' = 0$  and  $F^T e = 0$
- $F$  is singular (rank two)
- $F$  has seven degrees of freedom

Slide credit: Svetlana Lazebnik

### Recap: The Eight-Point Algorithm

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

$$(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \Rightarrow (u u', u v', u, v u', v v', v, u', v', 1) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

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

Slide credit: Svetlana Lazebnik

### Recap: Normalized Eight-Point Alg.

- 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.
- Use the eight-point algorithm to compute  $F$  from the normalized points.
- Enforce the rank-2 constraint using SVD. Set  $d_{33}$  to zero and reconstruct  $F$
- Transform fundamental matrix back to original units: if  $T$  and  $T'$  are the normalizing transformations in the two images, then the fundamental matrix in original coordinates is  $T^T F T'$ .

Slide credit: Svetlana Lazebnik

### Recap: Comparison of Estimation Algorithms

	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

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?

$l_{31} = F^T_{13} x_1$

$l_{32} = F^T_{23} x_2$

Slide credit: Svetlana Lazebnik

### Applications: 3D Reconstruction

Slide credit: Svetlana Lazebnik

Computer Vision WS 15/16

## 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

Projective ambiguity

Affine factorization

Projective factorization

Euclidean upgrade

B. Leibe

163

Computer Vision WS 15/16

## 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  $x_{ij}$

Slide credit: Svetlana Lazebnik

B. Leibe

164

Computer Vision WS 15/16

## 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

$$x = PX = (PQ^{-1})QX$$

Slide credit: Svetlana Lazebnik

B. Leibe

165

Computer Vision WS 15/16

## Recap: Hierarchy of 3D Transformations

Transformation	Dof	Matrix	Properties
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	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$	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

B. Leibe

166

Computer Vision WS 15/16

## Recap: Affine Structure from Motion

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

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

Points ( $3 \times n$ )

Cameras ( $2m \times 3$ )

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

C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: A factorization method*, IJCV, 9(2):137-154, November 1992.

Slide credit: Svetlana Lazebnik

B. Leibe

167

Computer Vision WS 15/16

## Recap: Affine Factorization

- Obtaining a factorization from SVD:

Possible decomposition:

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

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

Slide credit: Martial Hebert

168

Computer Vision WS 15/16

## Recap: Projective Factorization

RWTH AACHEN UNIVERSITY

$$D = \begin{bmatrix} z_{11} \mathbf{x}_{11} & z_{12} \mathbf{x}_{12} & \cdots & z_{1n} \mathbf{x}_{1n} \\ z_{21} \mathbf{x}_{21} & z_{22} \mathbf{x}_{22} & \cdots & z_{2n} \mathbf{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} \mathbf{x}_{m1} & z_{m2} \mathbf{x}_{m2} & \cdots & z_{mn} \mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}$$

Points (4 × n)  
Cameras (3m × 4)

$D = MS$  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

Computer Vision WS 15/16

## Recap: Sequential Projective SfM

RWTH AACHEN UNIVERSITY

- 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*

Points  
Cameras

Slide credit: Svetlana Lazebnik

Computer Vision WS 15/16

## Recap: Estimating the Euclidean Upgrade

RWTH AACHEN UNIVERSITY

- Goal: Estimate ambiguity matrix  $C$ 
  - Orthographic assumption:
 
$$M \rightarrow MC, S \rightarrow C^{-1}S$$

1) Image axes are perpendicular  $\mathbf{a}_1 \cdot \mathbf{a}_2 = 0$   
2) Scale is 1  $|\mathbf{a}_1|^2 = |\mathbf{a}_2|^2 = 1$

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

$$\begin{cases} \hat{a}_{i1} \cdot \hat{a}_{i2} = 0 \\ |\hat{a}_{i1}| = 1 \\ |\hat{a}_{i2}| = 1 \end{cases} \Leftrightarrow \begin{cases} \mathbf{a}_i^T C C^T \mathbf{a}_{i2} = 0 \\ \mathbf{a}_i^T C C^T \mathbf{a}_{i1} = 1, \\ \mathbf{a}_i^T C C^T \mathbf{a}_{i2} = 1 \end{cases} \quad \text{with } L = C C^T$$

this translates to  $A_i L A_i^T = I$

Slide adapted from S. Lazebnik, M. Hebert

Computer Vision WS 15/16

## Recap: Bundle Adjustment

RWTH AACHEN UNIVERSITY

- Non-linear method for refining structure and motion
- Minimizing mean-square reprojection error
 
$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j)$$

Diagram illustrating bundle adjustment with cameras  $P_1, P_2, P_3$  and points  $X_j$  being projected onto image planes.

Slide credit: Svetlana Lazebnik

Computer Vision WS 15/16

## Repetition

RWTH AACHEN UNIVERSITY

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

Motion field

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

$A^T A \quad A^T b$

Lucas-Kanade optical flow

Gaussian pyramid  
Coarse-to-fine estimation

Slide credit: Svetlana Lazebnik

Computer Vision WS 15/16

## Recap: Estimating Optical Flow

RWTH AACHEN UNIVERSITY

Diagram showing two frames  $I(x, y, t-1)$  and  $I(x, y, t)$  with points moving between them.

- 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.

Slide credit: Svetlana Lazebnik

## Recap: Lucas-Kanade Optical Flow

- Use all pixels in a  $K \times K$  window to get more equations.
- Least squares problem:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \quad \begin{matrix} A & d = b \\ 25 \times 2 & 2 \times 1 & 25 \times 1 \end{matrix}$$

- Minimum least squares solution given by solution of  $(A^T A) d = A^T b$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \quad \begin{matrix} A^T A & A^T b \end{matrix}$$

Recall the Harris detector!

Computer Vision WS 15/16

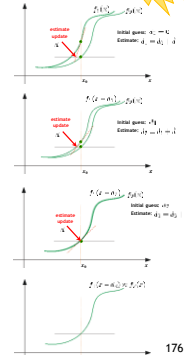
Slide adapted from Svetlana Lazebnik

B. Leibe

175

## 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.



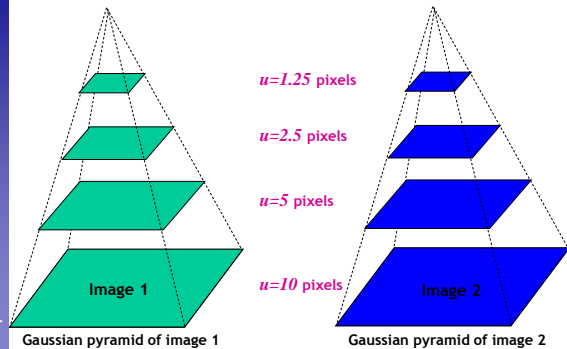
Computer Vision WS 15/16

Slide adapted from Steve Seitz

B. Leibe

176

## Recap: Coarse-to-fine Estimation



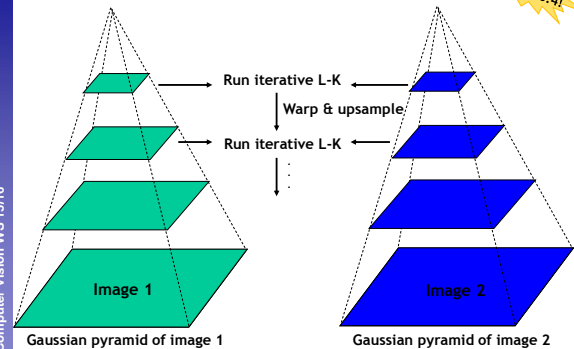
Computer Vision WS 15/16

Slide credit: Steve Seitz

B. Leibe

177

## Recap: Coarse-to-fine Estimation



Computer Vision WS 15/16

Slide credit: Steve Seitz

B. Leibe

178

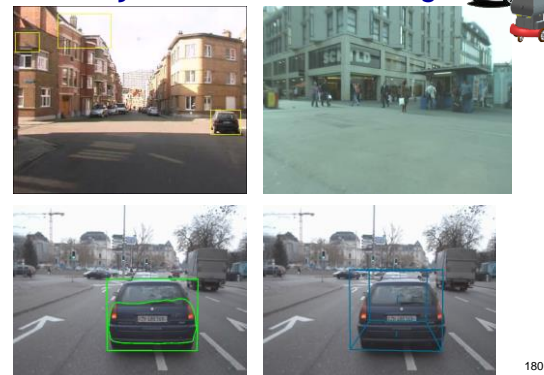
## Any Questions?

So what can you do with all of this?

Computer Vision WS 15/16

179

## Robust Object Detection & Tracking



Computer Vision WS 15/16

180

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Applications for Driver Assistance Systems

183

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Articulated Multi-Person Tracking

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

182

[Gammeter, Ess, Jaeggli, Schindler, Leibe, Van Gool, ECCV'08]

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Semantic 2D-3D Scene Segmentation

183

B. Leibe [G. Floros, B. Leibe, CVPR'12]

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Integrated 3D Point Cloud Labels

184

B. Leibe [G. Floros, B. Leibe, CVPR'12]

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Mining the World's Images...

185

Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

Automatic Landmark Building Discovery

186

Any More Questions?

*Good luck for the exam!*