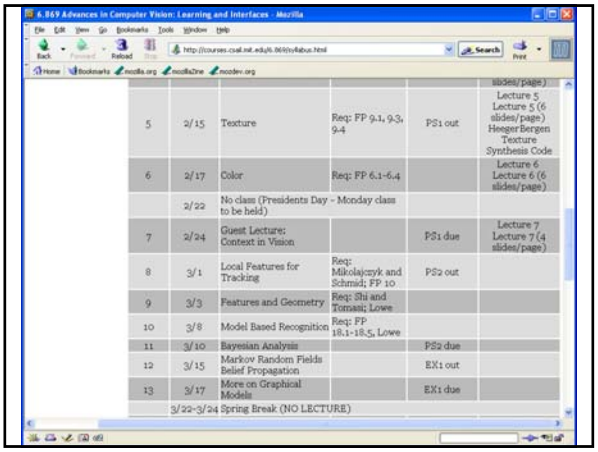
6.869 Advances in Computer Vision

Prof. Bill Freeman

March 1, 2005



6.869 Advances in Computer Vision: Learning and Interfaces - Mozilla

http://courses.csail.mit.edu/6.869/syllabus.html

					slides/page)
5	2/15	Texture	Req: FP 9.1, 9.3, 9.4	PS1 out	Lecture 5 Lecture 5 (6 slides/page) HeegerBergen Texture Synthesis Code
6	2/17	Color	Req: FP 6.1-6.4		Lecture 6 Lecture 6 (6 slides/page)
	2/22	No class (Presidents Day - Monday class to be held)			
7	2/24	Guest Lecture: Context in Vision		PS1 due	Lecture 7 Lecture 7 (4 slides/page)
8	3/1	Local Features for Tracking	Req: Mikolajczyk and Schmid; FP 10	PS2 out	
9	3/3	Features and Geometry	Req: Shi and Tomasi; Lowe		
10	3/8	Model Based Recognition	Req: FP 18.1-18.5, Lowe		
11	3/10	Bayesian Analysis		PS2 due	
12	3/15	Markov Random Fields Belief Propagation		EX1 out	
13	3/17	More on Graphical Models		EX1 due	
	3/22-3/24	Spring Break (NO LECTURE)			

Local Features

Matching points across images important for:
- object identification (instance recognition)
- object (class) recognition
- pose estimation
- stereo (3-d shape)
- motion estimate
- stitching together photographs into a mosaic
- etc

Today

Interesting points, correspondence.

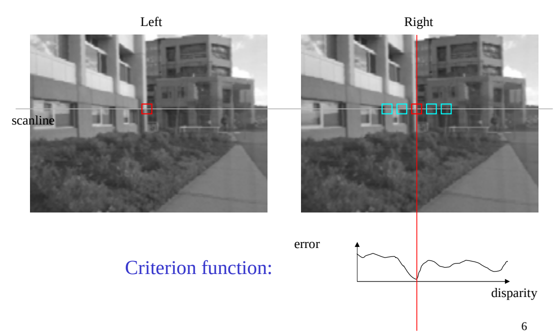
Scale and rotation invariant descriptors [Lowe]

Correspondence using window matching

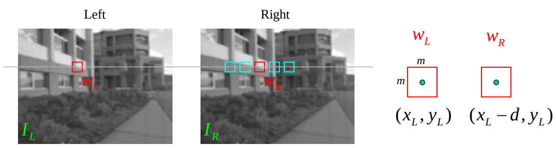
Points are highly individually ambiguous…

More unique matches are possible with small regions of image.

Correspondence using window matching



Sum of Squared (Pixel) Differences



w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

$$W_m(x,y) = \{u,v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x,y,d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$

Image Normalization

- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2}$$ Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}$$ Normalized pixel

Images as Vectors

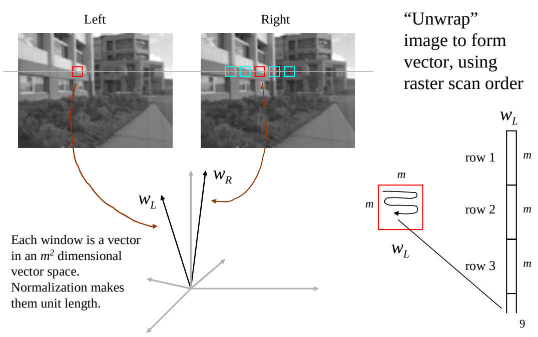
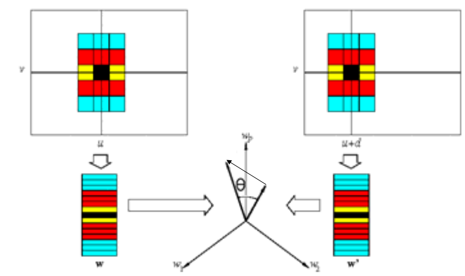


Image windows as vectors



Possible metrics

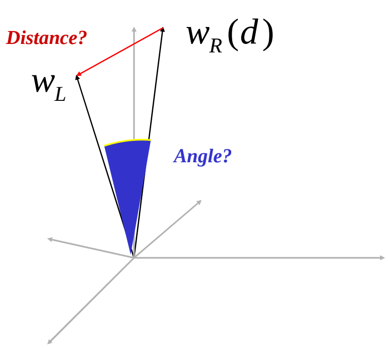
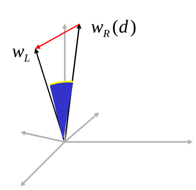


Image Metrics



(Normalized) Sum of Squared Differences

$$C_{\mathrm{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$
$$= \|w_L - w_R(d)\|^2$$

Normalized Correlation

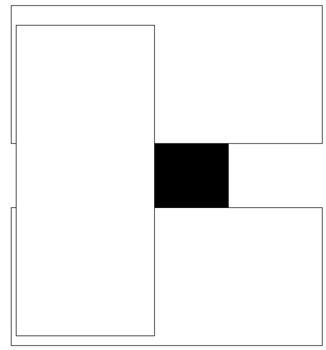
$$C_{\mathrm{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$
$$= w_L \cdot w_R(d) = \cos\theta$$

$$\boxed{d^* = \arg\min_d \|w_L - w_R(d)\|^2 = \arg\max_d w_L \cdot w_R(d)}$$

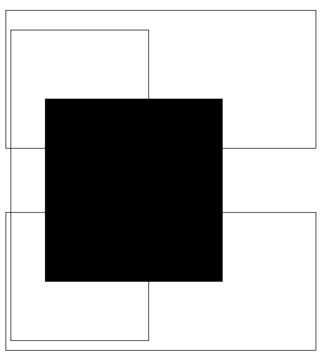
Local Features

Not all points are equally good for matching…

13

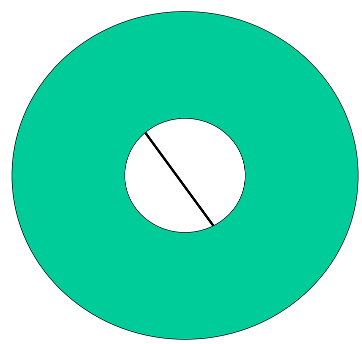


14



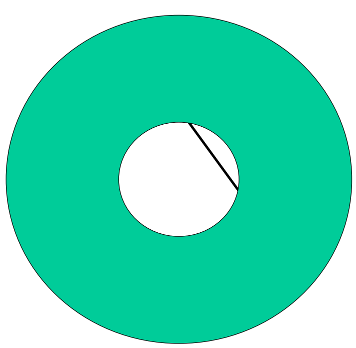
15

Aperture Problem and Normal Flow



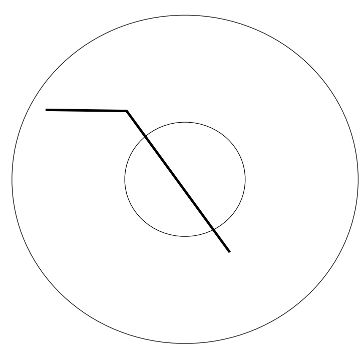
16

Aperture Problem and Normal Flow



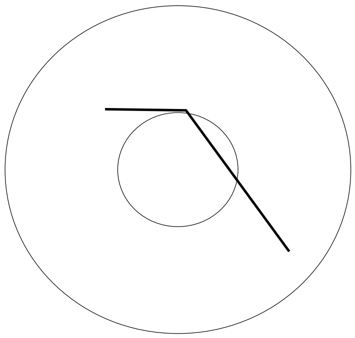
17

Aperture Problem and Normal Flow

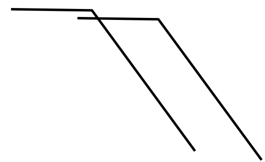


18

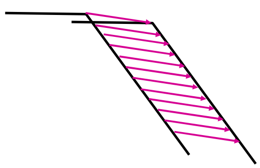
Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



Aperture Problem and Normal Flow



(Review) Differential approach: Optical flow constraint equation

Brightness should stay constant as you track motion

$$I(x+u\delta t, y+v\delta t, t+\delta t) = I(x,y,t)$$

1st order Taylor series, valid for small δt

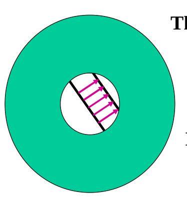
$$I(x,y,t) + u\delta t I_x + v\delta t I_y + \delta t I_t = I(x,y,t)$$

Constraint equation

$$\boxed{uI_x + vI_y + I_t = 0}$$

"BCCE" - Brightness Change Constraint Equation

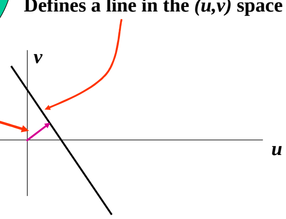
Aperture Problem and Normal Flow



The gradient constraint:

$$I_x u + I_y v + I_t = 0$$
$$\nabla I \bullet \vec{U} = 0$$

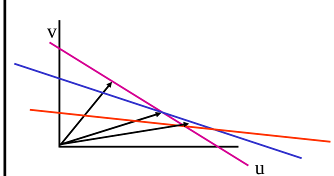
Defines a line in the *(u,v)* space



Normal Flow:

$$u_\perp = -\frac{I_t}{|\nabla I|}\frac{\nabla I}{|\nabla I|}$$

Combining Local Constraints



$$\nabla I^1 \bullet U = -I_t^1$$
$$\nabla I^2 \bullet U = -I_t^2$$
$$\nabla I^3 \bullet U = -I_t^3$$

etc.

Lucas-Kanade: Integrate gradients over a Patch

Assume a single velocity for all pixels within an image patch

$$E(u,v) = \sum_{x,y\in\Omega} \left(I_x(x,y)u + I_y(x,y)v + I_t\right)^2$$

Solve with:

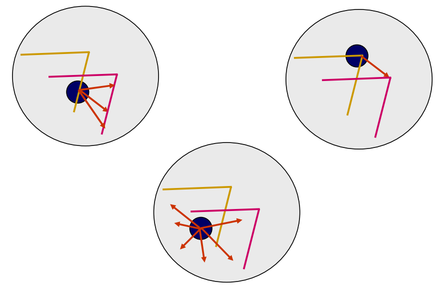
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}$$

On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I \nabla I^T\right)\vec{U} = -\sum \nabla I I_t$$

25

Local Patch Analysis



26

Selecting Good Features

- What's a "good feature"?
  - Satisfies brightness constancy
  - Has sufficient texture variation
  - Does not have too much texture variation
  - Corresponds to a "real" surface patch
  - Does not deform too much over time

27

Good Features to Track

$$\underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{\mathbf{u}} = \underbrace{-\begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix}}_{\mathbf{b}}$$

When is This Solvable?

- **A** should be invertible
- **A** should not be too small due to noise
  - eigenvalues λ_1 and λ_2 of **A** should not be too small
- **A** should be well-conditioned
  - λ_1/λ_2 should not be too large (λ_1 = larger eigenvalue)

Both conditions satisfied when *min*(λ_1, λ_2) > c

28

Harris detector

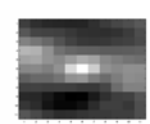
Auto-correlation matrix

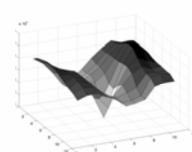
$$\begin{bmatrix} \sum_{(x_k,y_k)\in W} (I_x(x_k,y_k))^2 & \sum_{(x_k,y_k)\in W} I_x(x_k,y_k) I_y(x_k,y_k) \\ \sum_{(x_k,y_k)\in W} I_x(x_k,y_k) I_y(x_k,y_k) & \sum_{(x_k,y_k)\in W} (I_y(x_k,y_k))^2 \end{bmatrix}$$

- Auto-correlation matrix
  - captures the structure of the local neighborhood
  - measure based on eigenvalues of this matrix
    - 2 strong eigenvalues => interest point
    - 1 strong eigenvalue => contour
    - 0 eigenvalue => uniform region
- Interest point detection
  - threshold on the eigenvalues
  - local maximum for localization

29

Selecting Good Features

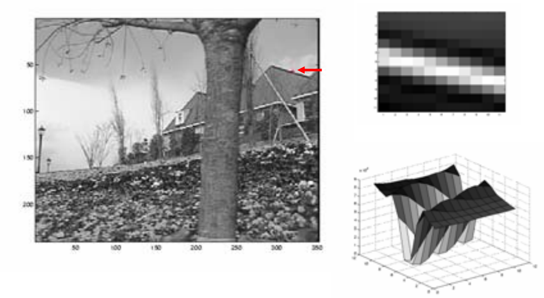




λ_1 and λ_2 are large

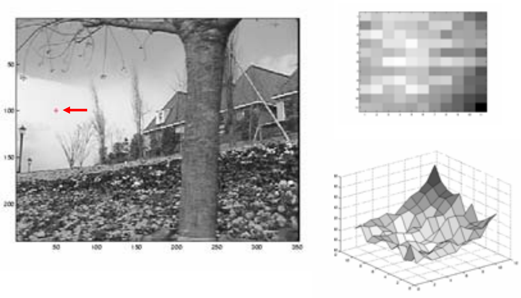
30

Selecting Good Features



large λ_1, small λ_2 31

Selecting Good Features



small λ_1, small λ_2 32

Today

Interesting points, correspondence.

Scale and rotation invariant descriptors [Lowe]

33

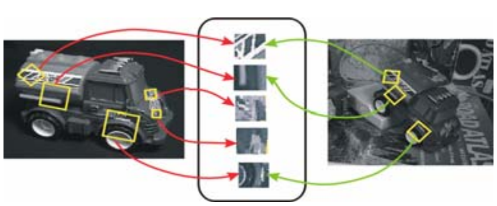
CVPR 2003 Tutorial

Recognition and Matching Based on Local Invariant Features

David Lowe
Computer Science Department
University of British Columbia

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



SIFT Features

35

Advantages of invariant local features

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to wide range of differing feature types, with each adding robustness

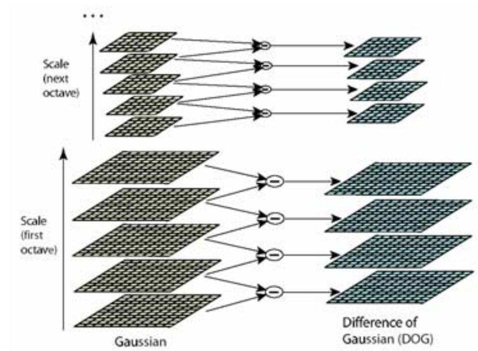
Scale invariance

Requires a method to repeatably select points in location and scale:

- The only reasonable scale-space kernel is a Gaussian (Koenderink, 1984; Lindeberg, 1994)
- An efficient choice is to detect peaks in the difference of Gaussian pyramid (Burt & Adelson, 1983; Crowley & Parker, 1984 – but examining more scales)
- Difference-of-Gaussian with constant ratio of scales is a close approximation to Lindeberg's scale-normalized Laplacian (can be shown from the heat diffusion equation)

37

Scale space processed one octave at a time



38

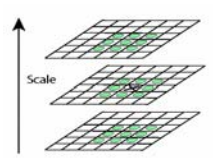
Key point localization

- Detect maxima and minima of difference-of-Gaussian in scale space
- Fit a quadratic to surrounding values for sub-pixel and sub-scale interpolation (Brown & Lowe, 2002)
- Taylor expansion around point:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

- Offset of extremum (use finite differences for derivatives):

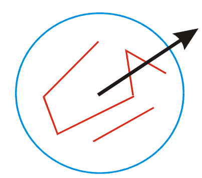
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$

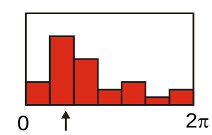


39

Select canonical orientation

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)

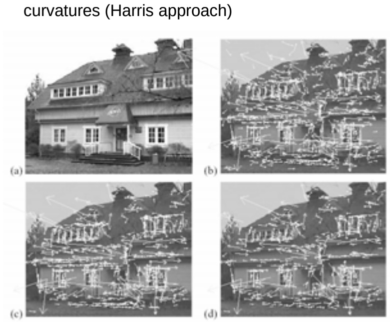




40

Example of keypoint detection

Threshold on value at DOG peak and on ratio of principle curvatures (Harris approach)

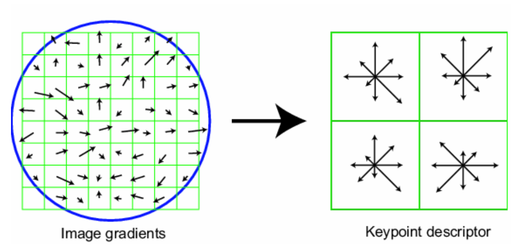


(a) 233x189 image
(b) 832 DOG extrema
(c) 729 left after peak value threshold
(d) 536 left after testing ratio of principle curvatures

41

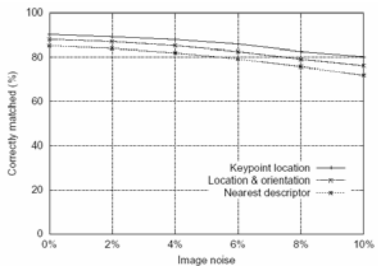
SIFT vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



Feature stability to noise

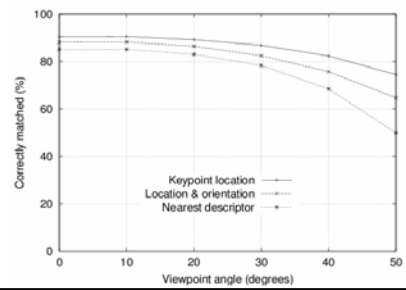
- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features



43

Feature stability to affine change

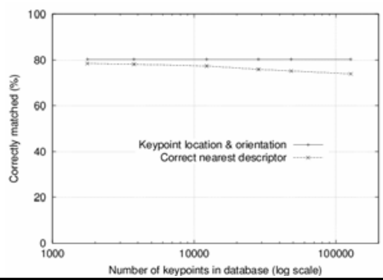
- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features



44

Distinctiveness of features

- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match



45

Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affi ne transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.

46

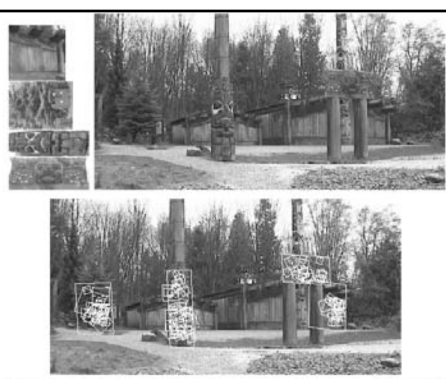


Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affi ne transform used for recognition.

A good SIFT features tutorial

http://www.cs.toronto.edu/~jepson/csc2503/tutSIFT04.pdf
By Estrada, Jepson, and Fleet.

48

An application of SIFT features in my own research…

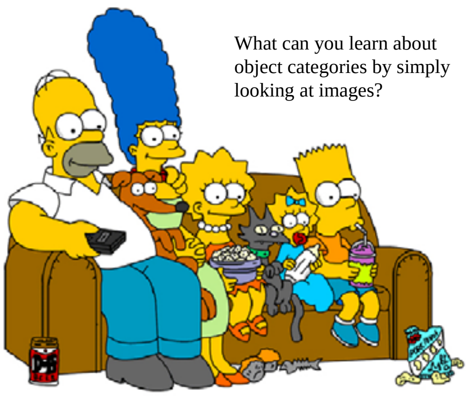
49

The couch potato project:
Learning from looking at images.

Bill Freeman, MIT
Joint work with: Josef Sivic, Andrew Zisserman (Oxford); Bryan Russell (MIT), Alyosha Efros (CMU).

December 18, 2004

50



What can you learn about object categories by simply looking at images?

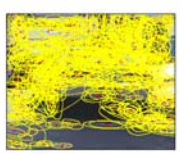
Labelled training databases

Labelling object classes in images is tedious, and can introduce biases.

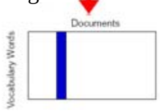
52

Overview of our Method

Find words



Form histograms

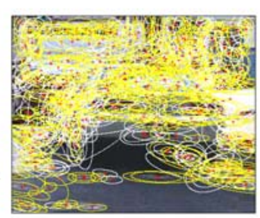


Discover topics

53

Extracting Words

- Find interest points using shape adapted (white) and maximally stable (yellow) regions
- Map ellipses to a circle
- Compute SIFT descriptor over circle



54

SIFT (scale invariant feature transforms)

David Lowe,
IJCV 2004

Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.

55

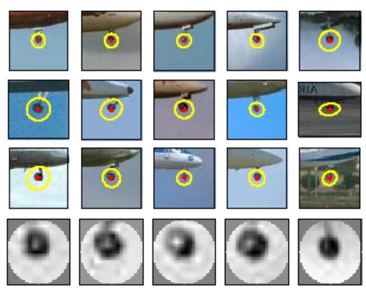
Visual words

- Vector quantize SIFT descriptors to a vocabulary of 2237 “visual words”.
- Heuristic design of descriptors makes these words somewhat invariant to:
  - Lighting
  - 2-d Orientation
  - 3-d Viewpoint

56

Examples of visual words

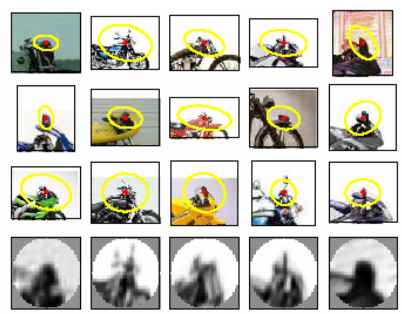
(a)



57

More visual words

(b)



58

Polysemy—the same word with different meanings

(a)

(b)

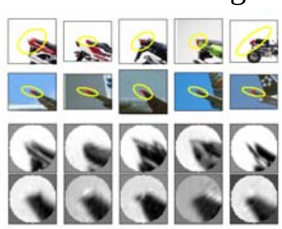


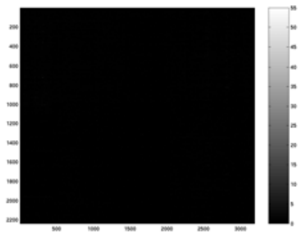
Figure 3: **Polysemy.** Example of a single visual word corresponding to two different (but locally similar) parts on two different object categories. (a) Top row shows occurrences of this visual word on the motorbike category, bottom row on the airplane category. The parts tend to occur consistently on different categories, i.e. this visual word fires mostly on the motorbike saddle and the airplane wing. (b) Corresponding normalized frames. Note the similarity of the normalized patches.

59

Experiment E

Observation matrix – experiment E

Visual word #

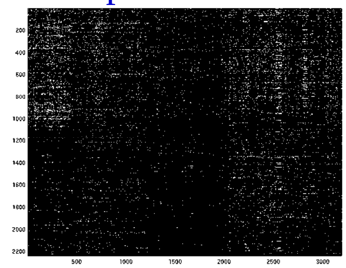


Frame #

13.8 % non-zero entries

Binarized observation matrix – experiment E

Visual word #



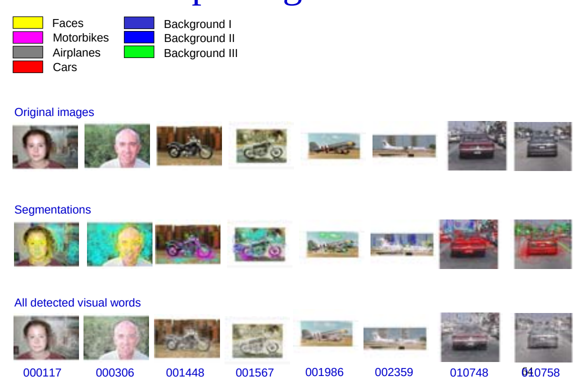
Frame #

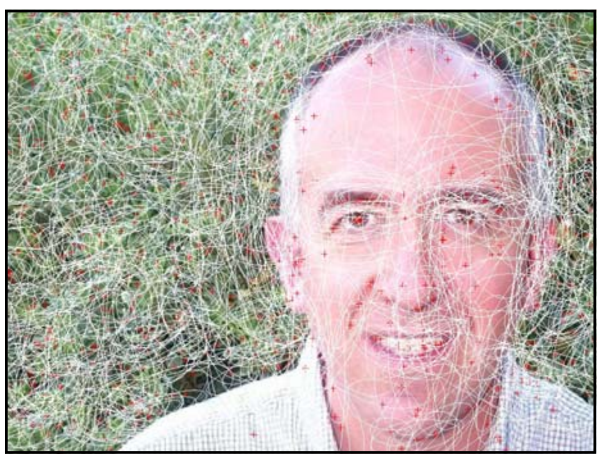
13.8 % non-zero entries

Results: All Experiments

Ex	Categories	pLSA		LDA		Texture	
		%	#	%	#	%	#
A	1,2 ub	100	1	99	7	91	53
B	1-3 ub	100	2	96	40	94	55
C	1-3	97	56	96	71	91	170
D	1-4	98	70	87	365	72	1060
E	1-4 + bg	78	931	77	970	73	1174
F	1-5,7-8 + bg	59	1515	64	1458	47	2093

Example segmentations



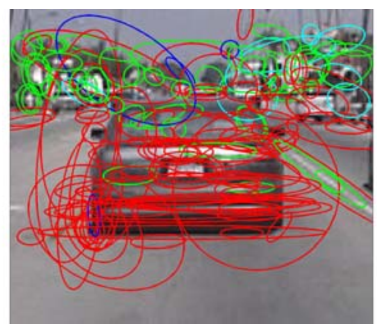


68

69

Faces
Motorbikes
Airplanes
Cars

Background I
Background II
Background III



70

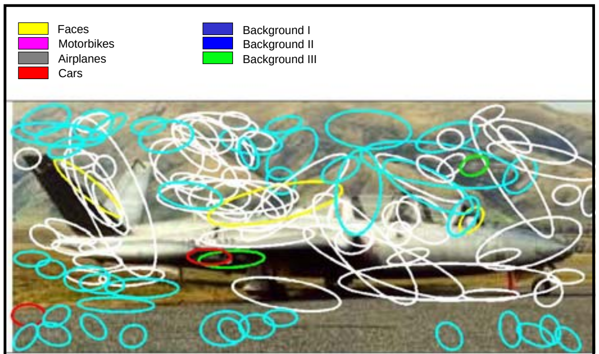


Figure 11: Multiple objects an image. **(a) pLSA example:** Two objects are present in this image: a motorbike (topic 1 - green) and a car (topic 6 - red). The learned mixture coefficients $P(z|d)$ are 0.41 (motorbikes - green), 0.02 (bg I - magenta), 0.16 (face - yellow), 0.19 (bg II - cyan), 0.04 (bg III - blue), 0.14 (cars - red), 0.02 (airplane - black). In total there are 740 elliptical regions in this image of which 95 (72 unique visual words) are shown (have $P(z|w, d)$ above 0.8). **(b) LDA example:** Two objects are present in this image. a face (yellow) and a car (red). The learned mixing weights θ are 0.19 car (red), 0.07 motorbike (green), 0.16 airplane (black), 0.14 background (blue), 0.44 face (yellow).

74