

Generalized Hough Transform, line fitting

Introduction to Computer Vision
CSE 152
Lecture 11a

CSE152, Winter 2013

Intro Computer Vision

Announcements

- Assignment 2: Due Thursday
- Midterm: Thursday, February 14 in class,
We'll talk about it at the end of class

CSE152, Winter 2013

Intro Computer Vision

Happy Mardi Gras



CSE152, Winter 2013

Intro Computer Vision

Canny Edge Detector

1. Smooth image by filtering with a Gaussian
2. Compute gradient at each point in the image.
3. At each point in the image, compute the direction of the gradient and the magnitude of the gradient.
4. Perform non-maximal suppression to identify candidate edgels.
5. Trace edge chains using hysteresis thresholding.

CSE152, Winter 2013

Intro Computer Vision

So, to detect corners

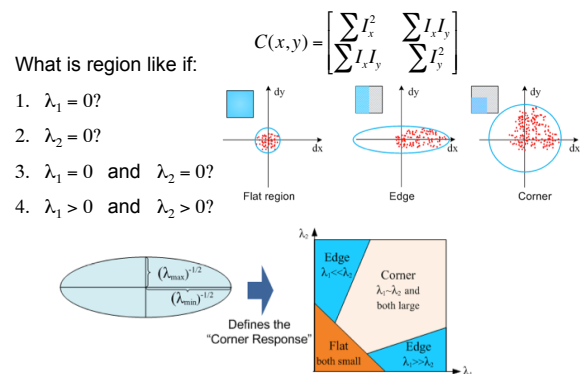
- Filter image with a Gaussian.
- Compute the gradient everywhere.
- Move window over image and construct C over the window.
- Use linear algebra to find λ_1 and λ_2 .
- If they are both big, we have a corner.
 1. Let $e(x,y) = \min(\lambda_1(x,y), \lambda_2(x,y))$
 2. (x,y) is a corner if it's local maximum of $e(x,y)$ and $e(x,y) > \tau$

Parameters: Gaussian std. dev, window size, threshold

CSE152, Winter 2013

Intro Computer Vision

Corner Detection

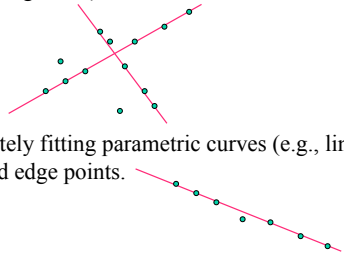


CSE152, Winter 2013

Intro Computer Vision

What to do with edges?

- Segment linked edge chains into curve features (e.g., line segments).
- Group unlinked or unrelated edges into lines (or curves in general).
- Accurately fitting parametric curves (e.g., lines) to grouped edge points.



CSE152, Winter 2013

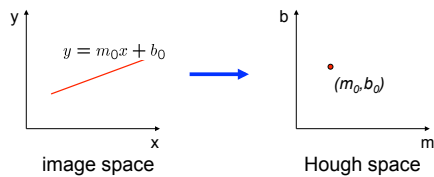
Intro Computer Vision

Hough Transform [Patented 1962]

CSE152, Winter 2013

Intro Computer Vision

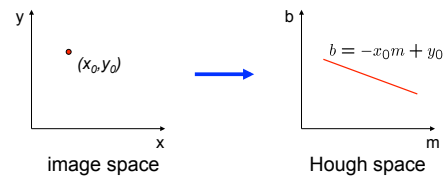
Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
 - given a set of points (x,y), find all (m,b) such that $y = mx + b$

Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
 - given a set of points (x,y), find all (m,b) such that $y = mx + b$
- What does a point (x_0, y_0) in the image space map to?
 - A: the solutions of $b = -x_0m + y_0$
 - this is a line in Hough space

Hough Transform Algorithm

- Typically use a different parameterization

$$d = x \cos \theta + y \sin \theta$$
 - d is the perpendicular distance from the line to the origin
 - θ is the angle this perpendicular makes with the x axis
 - Why?
- Basic Hough transform algorithm
 - Initialize $H[d, \theta] = 0$; H is called accumulator array
 - for each edge point $I[x, y]$ in the image
 - for $\theta = 0$ to 180

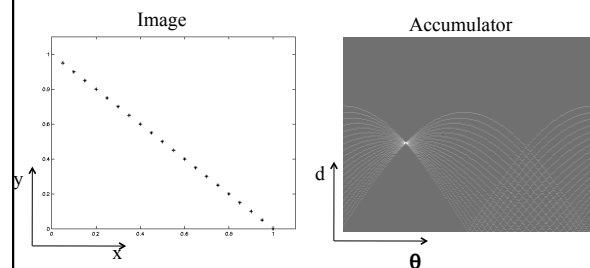
$$d = x \cos \theta + y \sin \theta$$

$$H[d, \theta] += 1$$
 - Find the value(s) of (d, θ) where $H[d, \theta]$ is the global maximum
 - The detected line in the image is given by $d = x \cos \theta + y \sin \theta$
- What's the running time (measured in # votes)?

CSE152, Winter 2013

Intro Computer Vision

Hough Transform: 20 colinear points

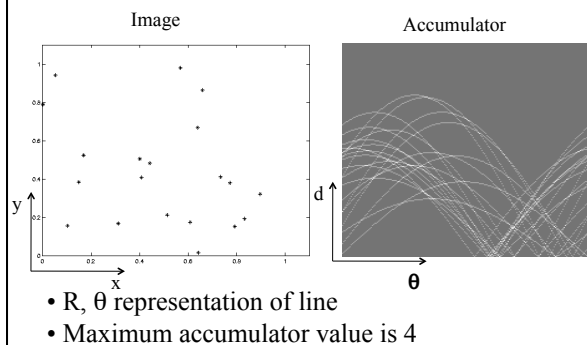


- R, θ representation of line
- Maximum accumulator value is 20

CSE152, Winter 2013 Note: accumulator array range: theta: 0-360, d: positive

Intro Computer Vision

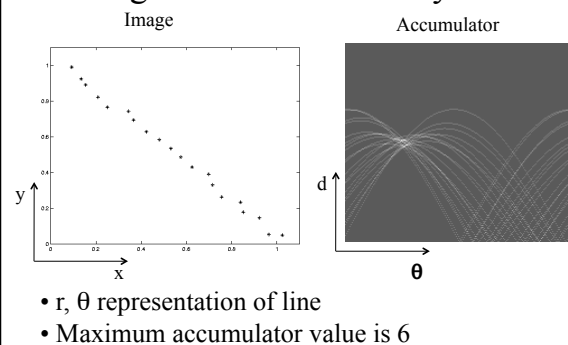
Hough Transform: Random points



CSE152, Winter 2013

Intro Computer Vision

Hough Transform: "Noisy line"



CSE152, Winter 2013

Intro Computer Vision

Hough Transform for Curves

Generalized Hough Transform

The H.T. can be generalized to detect any curve that can be expressed in parametric form:

$$y = f(x, a_1, a_2, \dots, a_p)$$

or

$$g(x, y, a_1, a_2, \dots, a_p) = 0$$

- $a_1, a_2, \dots, a_p$ are the parameters
- The parameter space is p -dimensional
- The accumulating array is LARGE!

CSE152, Winter 2013

Intro Computer Vision

Example: Finding circles

Equation for circle is

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

where the parameters are the circle's center (x_c, y_c) and radius r .

Three dimensional generalized Hough space.

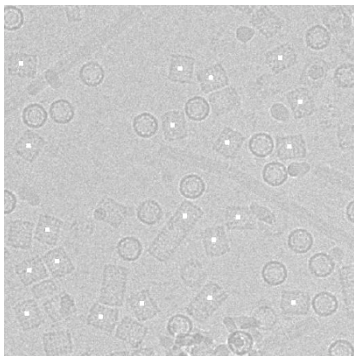
Given an edge point (x, y) ,

1. Loop over all values of (x_c, y_c) ,
2. Compute r
3. Increment $H(x_c, y_c, r)$

CSE152, Winter 2013

Intro Computer Vision

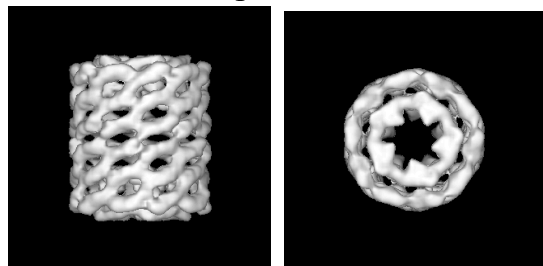
TEM Image of Keyhole Limpet Hemocyanin with detected particles



CSE152, Winter 2013

Intro Computer Vision

3D Maps of KLH



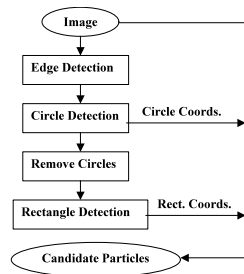
Three-dimensional maps of KLH at a resolution of 23.5 Å reconstructed using particles extracted either manually or automatically as described in the text. (a), (b) The side- and top- view of a 3D map reconstructed from a set of 1042 manually selected particle images.

CSE152, Winter 2013

Intro Computer Vision

Processing in Stage 1 for KLH

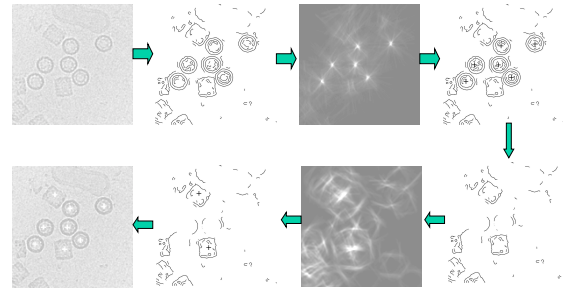
- Canny edge detection.
- A sequence of ordered Hough transforms (HT's) is applied in order from the computationally simplest one to the most complex one.
- Edges covered by the detected shapes are removed immediately from edge images following the application of the last HT.



CSE152, Winter 2013

Intro Computer Vision

Picking KLH Particles in Stage 1

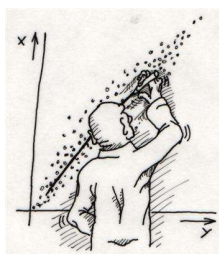


Zhu et al., IEEE Transactions on Medical Imaging, 2003

CSE152, Winter 2013

Intro Computer Vision

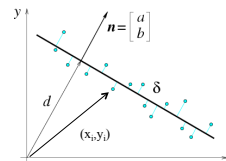
Line Fitting



CSE152, Winter 2013

Intro Computer Vision

Line Fitting



Given n points (x_i, y_i) , estimate parameters of line

$$ax_i + by_i - d = 0$$

subject to the constraint that

$$a^2 + b^2 = 1$$

Problem: minimize

$$E(a, b, d) = \sum_{i=1}^n (ax_i + by_i - d)^2$$

Cost Function:
Sum of squared distances
between each point and the line

with respect to (a, b, d) .

1. Minimize E with respect to d :

$$\frac{\partial E}{\partial d} = 0 \Rightarrow d = \frac{1}{n} \sum_{i=1}^n ax_i + by_i = a\bar{x} + b\bar{y}$$

where $(\bar{x}, \bar{y})$ is the
mean of the data points

CSE152, Winter 2013

Intro Computer Vision

Line fitting cont.

2. Substitute d back into E

$$E = \sum_{i=1}^n [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = \|\mathcal{U}\mathbf{n}\|^2 \quad \text{where } \mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$$

where $\mathbf{n} = (a \ b)^T$.

3. Minimize $E = \mathbf{n}^T \mathcal{U}^T \mathcal{U} \mathbf{n} = \mathbf{n}^T \mathbf{S} \mathbf{n}$ with respect to a, b subject to the constraint $\mathbf{n}^T \mathbf{n} = 1$. Note that $\mathbf{S}$ is given by

$$\mathbf{S} = \mathcal{U}^T \mathcal{U} = \begin{pmatrix} \sum_{i=1}^n x_i^2 - n\bar{x}^2 & \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} \\ \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} & \sum_{i=1}^n y_i^2 - n\bar{y}^2 \end{pmatrix}$$

And it's a real, symmetric, positive definite

CSE152, Winter 2013

Intro Computer Vision

Line Fitting – Finished

4. This is a constrained optimization problem in $\mathbf{n}$. Solve with Lagrange multiplier

$$\mathbf{L}(\mathbf{n}) = \mathbf{n}^T \mathbf{S} \mathbf{n} - \lambda (\mathbf{n}^T \mathbf{n} - 1)$$

Take partial derivative (gradient) w.r.t. $\mathbf{n}$ and set to 0.

$$\nabla \mathbf{L} = 2\mathbf{S} \mathbf{n} - 2\lambda \mathbf{n} = 0$$

or

$$\mathbf{S} \mathbf{n} = \lambda \mathbf{n}$$

$\mathbf{n} = (a, b)$ is an Eigenvector of the symmetric matrix $\mathbf{S}$ (the one corresponding to the smallest Eigenvalue).

5. d is computed from Step 1.

CSE152, Winter 2013

Intro Computer Vision

Midterm Thursday, May 5

- In class
- Full period
- Coverage – everything up to this point including readings
- “Cheat sheet” – you can prepare a one sided sheet of notes. It must be hand written. (After the midterm, save your sheet since you can use the other side for the final).
- No calculators.

CSE152, Winter 2013

Intro Computer Vision

Incomplete list of topics covered...

- Human visual system
 - Physiology – from eye to brain
 - Phenomenological
 - Function
- Camera models
- Factors in producing images
- Projection models
 - Perspective
 - Orthographic
- Homogenous Coordinates,
- Vanishing points
- Lenses, Distortion
- Sensors
- Quantization/Resolution
- Illumination
- Reflectance
 - BRDF
 - Lambertian
 - Specular
 - Phong
- Color
 - Light Spectrum
 - Reflectance, source
 - Sensor response
 - Color spaces
 - Chromaticity, YUV, RGB

CSE152, Winter 2013

Intro Computer Vision

Topics cont.

- Binary Vision
 - Thresholding
 - Neighborhoods
 - Connected component exploration
 - Features, moments
- Noise
 - Additive, Gaussian noise
- Filtering, linear, convolution with Kernel
 - Averaging/smoothing
 - Sharpening
 - Derivatives
 - Gaussian filter
 - Separability
- Edges & Edge detection
- Edge sources
- Canny
 - Gaussian derivatives
 - Magnitude, orientation
 - Non-maximal suppression
 - Linking/thresholding
- Hough Transform
- Generalized Hough transform
- Line fitting

CSE152, Winter 2013

Intro Computer Vision

Shape-from-X

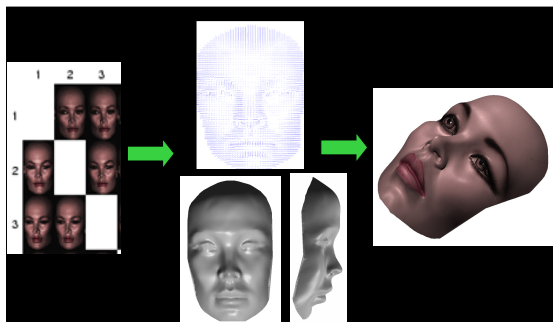
(i.e., Reconstruction)

- Methods for estimating 3-D shape from image data. X can be one (or more) of many cues.
 - Stereo (two or more views, known viewpoints)
 - Motion (moving camera or object)
 - Shading
 - Changing lighting (Photometric Stereo)
 - Texture variation
 - Focus/blur
 - Prior knowledge/context
 - structured light/lasers

CSE152, Winter 2013

Intro Computer Vision

Example: Helmholtz Stereo Depth + Normals + BRDF



CSE152, Winter 2013

Intro Computer Vision

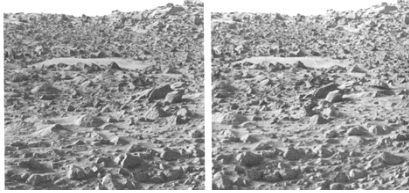
Stereo

CSE152, Winter 2013

Intro Computer Vision

Binocular Stereopsis: Mars

Given two images of a scene where relative locations of cameras are known, estimate depth of all common scene points.

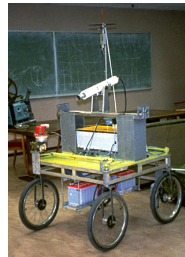


Two images of Mars

CSE152, Winter 2013

Intro Computer Vision

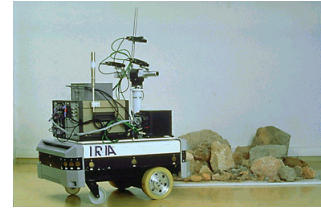
An Application: Mobile Robot Navigation



The Stanford Cart,
H. Moravec, 1979.

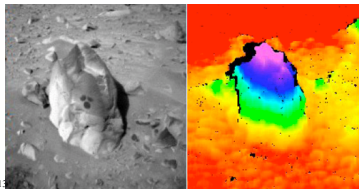
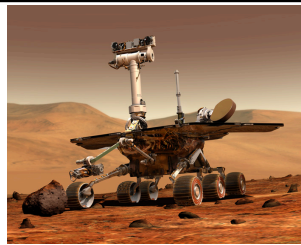
Courtesy O. Faugeras and H. Moravec.
CSE152, Winter 2013

The INRIA Mobile Robot, 1990.



Intro Computer Vision

Mars Exploratory Rovers: Spirit and Opportunity



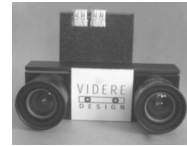
CSE152, Winter 2013

Intro Computer Vision

Commercial Stereo Heads



Trinocular stereo



Binocular stereo

CSE152, Winter 2013

Intro Computer Vision