CENG 789 – Digital Geometry Processing

13- Least-Squares, RANSAC, Hough Transform

Prof. Dr. Yusuf Sahillioğlu

Computer Eng. Dept, MIDDLE EAST TECHNICAL UNIVERSITY, Turkey

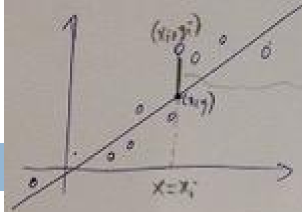
Outline

- ✓ In this class we will fit primitives to geometric data through
  - ✓ Least-squares.
  - ✓ RANSAC.
  - ✓ Hough Transform.
    - ✓ Surface Recon. ~ arbitrary models, not just basic primitives.
- ✓ Bonus material on some other geometric problems (distance, intersection) appended.

For x=xi, where
I have data, a&b
give me y, which
is different from yi.

Minimize sum of
these differences.

Least-Squares Line Fitting (vertical distance) # Best fit to all
data points is the IDEA



$$d = y_i - y = y_i - ax - b = y_i - ax_i - b$$ // minimize // vertical // distance to // line

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

$$E(a,b) = \sum_{i=1}^{n} (y_i - ax_i - b)(y_i - ax_i - b) = \sum_{i=1}^{n} -ay_ix_i - by_i - ax_iy_i + a^2x_i^2 + abx_i - by_i + abx_i + b^2$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} -x_iy_i - x_iy_i + 2ax_i^2 + bx_i + bx_i = \sum_{i=1}^{n} (-2x_i)(y_i - ax_i - b) = 0$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} -y_i + ax_i - y_i + ax_i + 2b = \sum_{i=1}^{n} (-2)(y_i - ax_i - b) = 0$$

$$-2\sum_{i=1}^{n} x_iy_i + 2a\sum_{i=1}^{n} x_i^2 + 2b\sum_{i=1}^{n} x_i = 0 \Rightarrow \sum_{i=1}^{n} x_iy_i = a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i$$

$$-2\sum_{i=1}^{n} y_i + 2a\sum_{i=1}^{n} x_i + 2b\cdot n = 0 \Rightarrow \sum_{i=1}^{n} y_i = a\sum_{i=1}^{n} x_i + bn$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_iy_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

$$A \quad x = b \Rightarrow x = A^{-1} \cdot b$$

Least-Squares Polynomial Fitting (vertical distance)

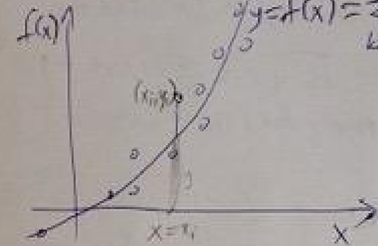
$$\hookrightarrow y = f(x) = \sum_{k=0}^{m} a_k x^k$$

$$E(a_0, a_1, a_2, \ldots, a_m) = \sum_{i=1}^{n} (y_i - f(x_i))^2 \Rightarrow$$ get m+1 eqs by setting $\partial E / \partial a_k = 0$ for $k = 0, \ldots, m$ — linearly on $a_0, a_1, \ldots, a_m$

Soln is a linear sys bcs f(x) depends

Least-Squares Parabola Fitting (vertical distance) $\rightarrow$ a special polynomial

$$y = f(x) = \sum_{k=0}^{2} a_k x^k = a_0 + a_1 x + a_2 x^2$$



$$E(a_0, a_1, a_2) = \sum_{i=1}^{n} (y_i - y)^2$$

$x = x_i$

$$= \sum_{i=1}^{n} (y_i - f(x))^2$$

$$= \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

In general, $E(a, b, c, \ldots) = \sum_i (y_i - f_{abc\ldots}(x_i))^2 \Rightarrow \frac{\partial E}{\partial a} = \sum_i -2(y_i - f_{abc\ldots}(x_i))\, \partial f / \partial a$ by chain rule

$$\frac{d(x^3+2x^2)^{1/2}}{dx} = \frac{1}{2}(x^3+2x^2)^{-1/2}(3x^2+4x)$$

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{n} -2(y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0 \quad \sum_{i=1}^{n} y_i = a_0 n + a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{n} -2x_i(y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0 \quad \sum_{i=1}^{n} x_i y_i = a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + a_2 \sum_{i=1}^{n} x_i^3$$

$$\frac{\partial E}{\partial a_2} = \sum_{i=1}^{n} -2x_i^2(y_i - a_0 - a_1 x_i - a_2 x_i^2) = 0 \quad \sum_{i=1}^{n} x_i^2 y_i = a_0 \sum_{i=1}^{n} x_i^2 + a_1 \sum_{i=1}^{n} x_i^3 + a_2 \sum_{i=1}^{n} x_i^4$$

$$\sum_{i=1}^{n} \begin{bmatrix} 1 & x_i & x_i^2 \\ x_i & x_i^2 & x_i^3 \\ x_i^2 & x_i^3 & x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} y_i \\ x_i y_i \\ x_i^2 y_i \end{bmatrix}$$

Solving Linear Systems

$$Ax = b \Rightarrow x = A^{-1} b$$

If A is not square or not full rank, A is singular and A^{-1} sucks. Look for an LS soln,

$$\min_x \|Ax - b\|^2 = \min_x (Ax-b)^T(Ax-b) = (Ax)^T(Ax) - 2(Ax)^T b + b^T b = x^T A^T A x - 2x^T A^T b + b^T b$$

$$\partial/\partial x = 2A^T A x - 2A^T b = 0 \Rightarrow A^T A x = A^T b \Rightarrow x = \underline{(A^T A)^{-1} A^T} b$$

pseudo-inv of A.

Full rank: columns of A are linearly independent. (or rows)

If A has full rank, $A^T A$ also has full rank, hence invertible

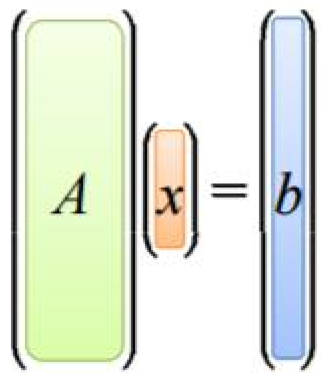
square even if A is $m \times n$

$m > n$: more eqs than unknowns $\Rightarrow$ overdetermined. Each eq typically

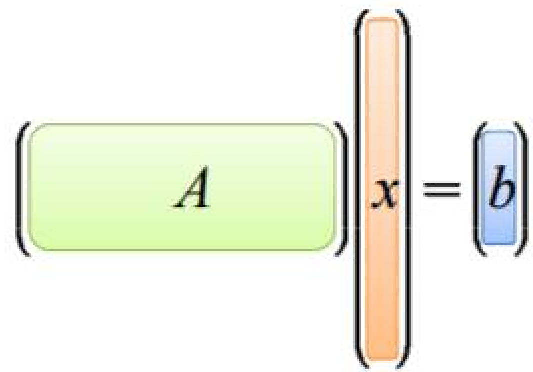
unknowns & eqs same ➔ single unique soln: $x=A^{-1}b$.

Matrix not always invertible

– Not square



Over determined



Under determined

$x=1$, $x=2$ //no exact soln.

$x=1$, $2x=2$ //exact soln.

$x+y+z=1$, $x+y+z=3$ //no soln.

$x+y+z=1$, $x+y+2z=3$ //inf. many

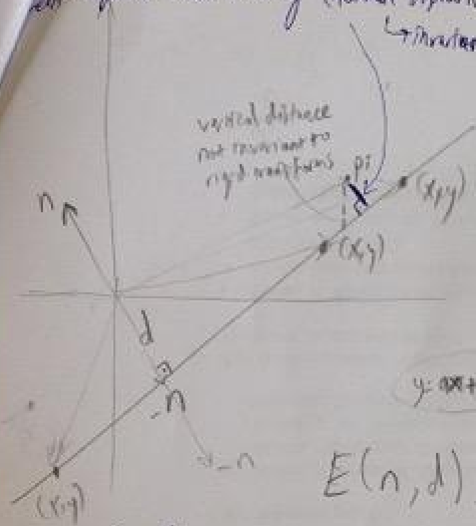
Use least-squares for approx. or exact solutions in either case.

$$\min_{x} \|Ax-b\|^2$$

ast-Square Line Fitting (normal distance)

↳ invariant under rigid transformations of point set P

$n_0x_0 + n_1x_1 + d = 0$ (4)



$n \cdot x = d$

$n \cdot x = -d \Rightarrow nx + d = 0$

line equation $n^T x + d = 0$

$y = ax + b$ $\quad \frac{x}{a} + \frac{y}{b} = 1$ $\quad x\cos\theta + y\sin\theta = r$

$$E(n,d) = \sum_i (n^T p_i + d)^2, \text{ s.t. } \|n\| = 1$$

Solve for d

$$E(n,d) = \sum_i (n^T p_i + d)^T (n^T p_i + d) = \sum_i (n^T p_i)^T (n^T p_i) + 2(n^T p_i)^T d + d^T d$$

$$= \sum_i p_i^T n n^T p_i + 2 p_i^T n d + d^T d$$

$$\frac{\partial E}{\partial d} = 2\sum_i p_i^T n + 2\sum_i d = 0 \quad nd = -\sum_i p_i^T n = -\sum_i n^T p_i = -n^T \sum_i p_i$$

$$\Rightarrow d = -n^T \frac{\sum_i p_i}{n} \Rightarrow d = -n^T \bar{p}$$

→ mean of point set P

Solve for n $\quad E(n) = \sum_i (n^T p_i - n^T \bar{p})^2 = \sum_i (n^T \tilde{p}_i)^2$, s.t. $\|n\| = 1$ where $\tilde{p}_i = p_i - \bar{p}$

→ penalize n not satisfying $\|n\| = 1$

$$E(n) = \sum_i (n^T \tilde{p}_i)^2 + \lambda(1 - n^T n)$$

→ $\|n\| \gg 1$ would be naturally penalized by this square

$$= \sum_i (n^T \tilde{p}_i)^T (n^T \tilde{p}_i) + \lambda(1 - n^T n)$$

$$\sum (n^T \tilde{p}_i)(n^T \tilde{p}_i)^T + \lambda(1 - n^T n) = \sum_i n^T \tilde{p}_i \tilde{p}_i^T n + \lambda(1 - n^T n) = \longrightarrow$$

$$E(n) = n^T \underbrace{\left(\sum_i \tilde{p}_i \tilde{p}_i^T\right)}_{\substack{C\ :\ \text{covariance matrix} \\ 3\times3}} \cdot n + \lambda(1 - n^T n) = n^T C n + \lambda(1 - n^T n)$$

(5)

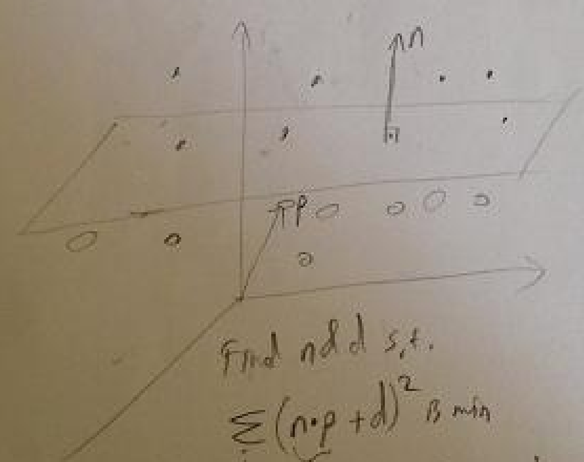
$$\frac{\partial E}{\partial n} = 2Cn - 2\lambda n = 0 \Rightarrow Cn = \lambda n$$

n is one of the eigenvectors of C

To minimize E, select the smallest eigvec (the one corresponding to smallest eigvalue λ)

$E(n) = \cancel{n^T \lambda n} + \lambda - \cancel{n^T \lambda n}$ // substitute $Cn = \lambda n$

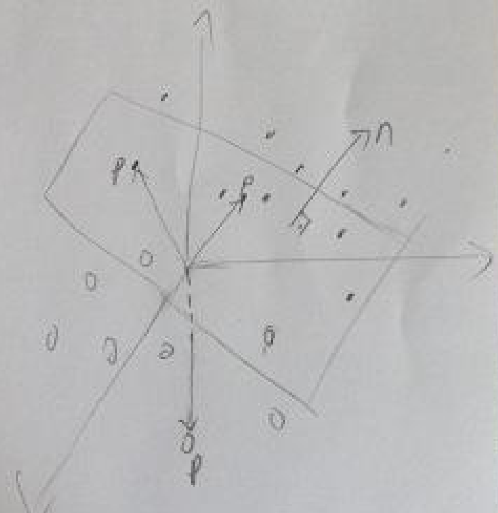
* YOU CAN ALSO CONSIDER THIS n & d AS LEAST-SQUARES PLANE FITTING to 3D points



Find n & d s.t.

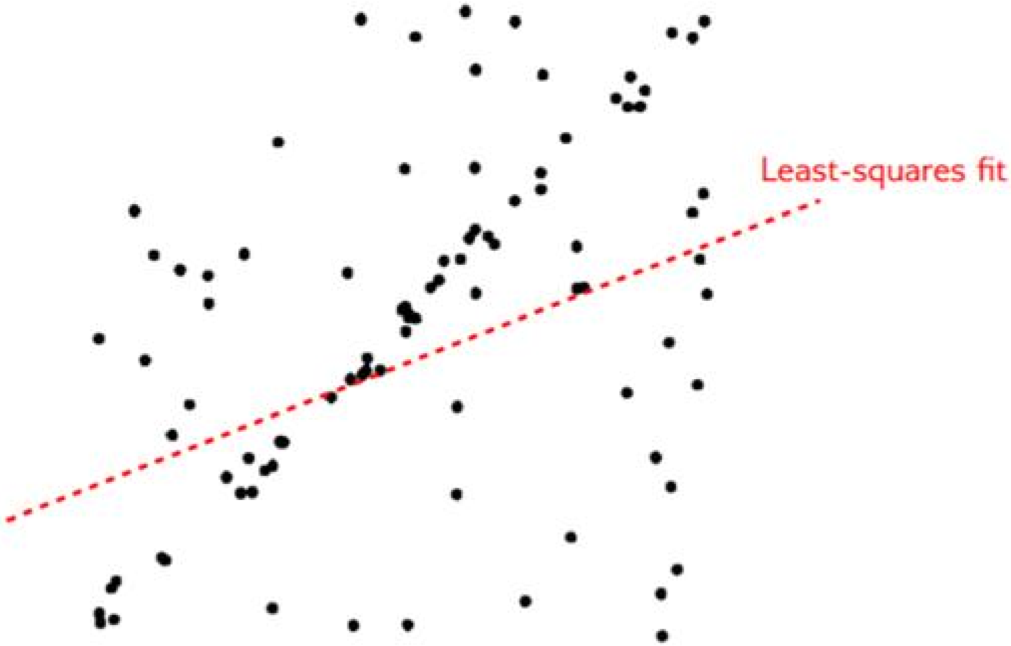
$\sum_i (n \cdot p + d)^2$ is min

$n^T p = -d$ if p on plane



Other Fitting Approaches

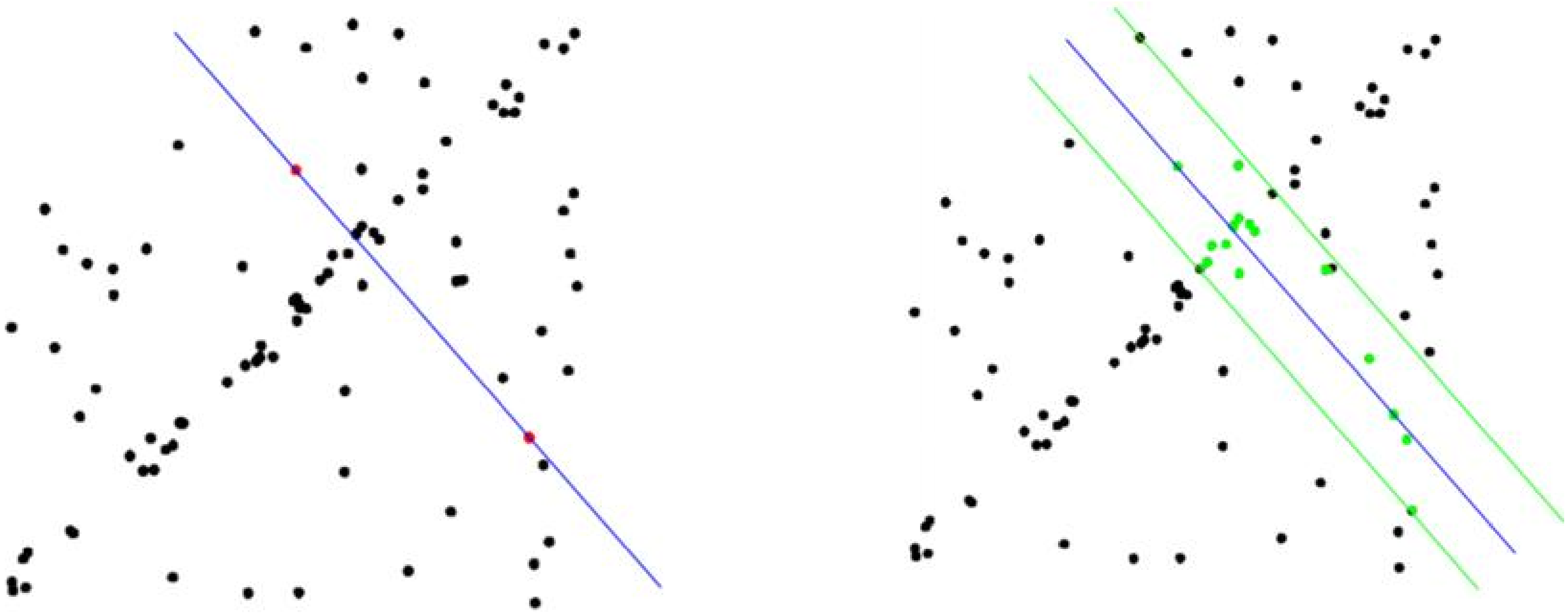
- ✓ RANSAC: Randomized Sample Consensus.
- ✓ Robust to outliers ☺.
- ✓ Not good for models represented w/ many parameters ☹.



- ✓ Here least-squares approach fails due to outliers.

Other Fitting Approaches

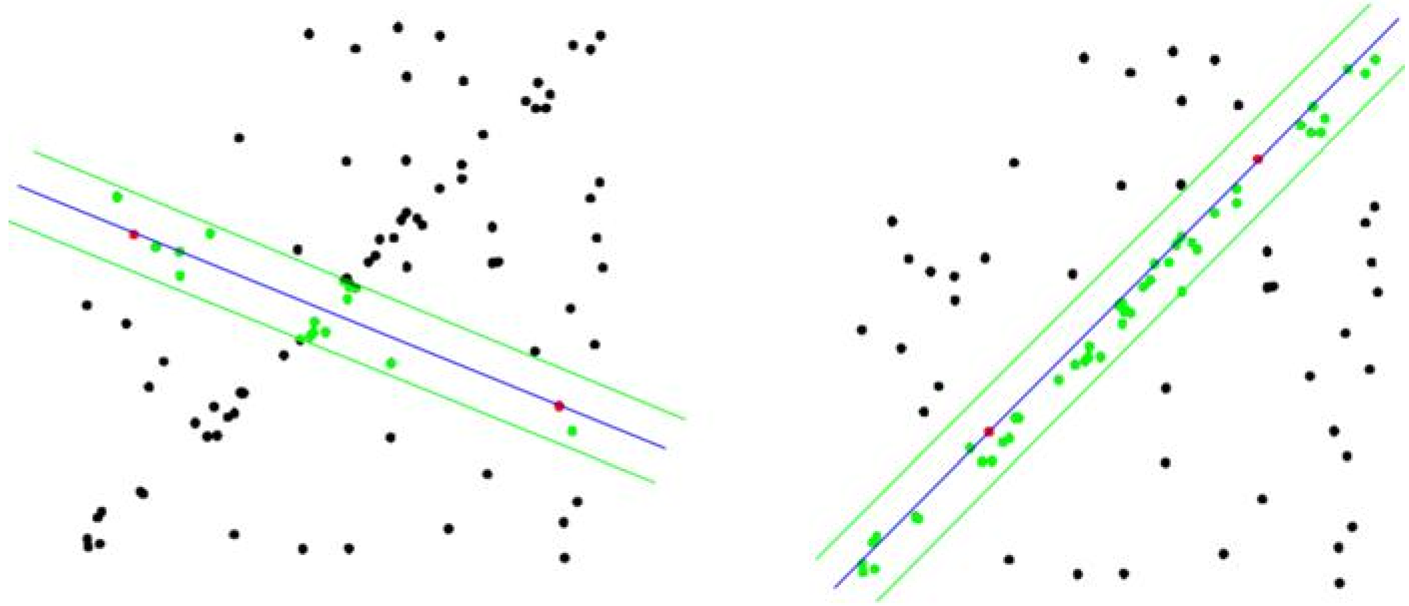
- ✓ RANSAC: Randomized Sample Consensus.
- ✓ Robust to outliers ☺.
- ✓ Not good for models represented w/ many parameters ☹.



- ✓ Fit model to d (=2 for line-fitting) random pnts, check # inliers (low).

Other Fitting Approaches

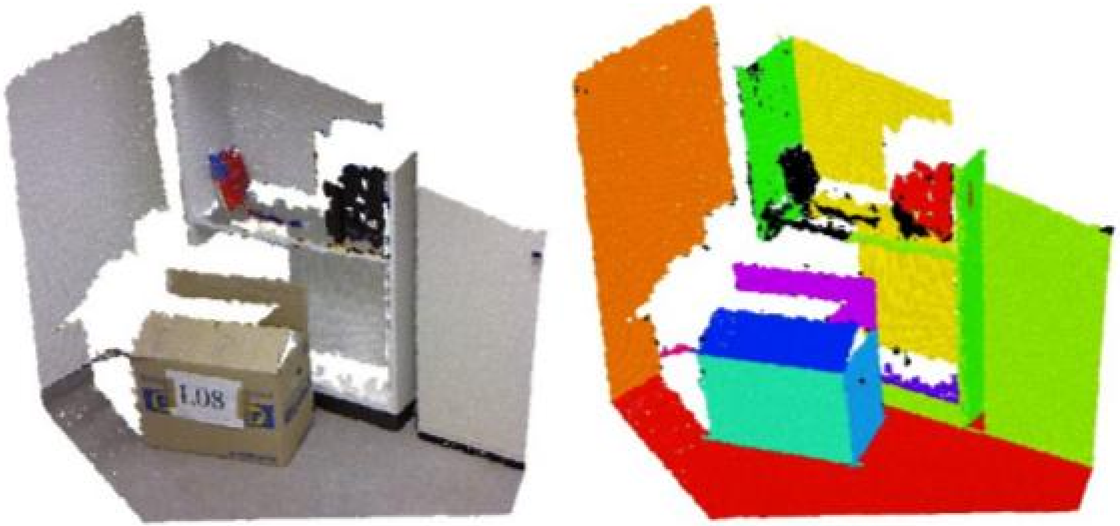
- ✓ RANSAC: Randomized Sample Consensus.
- ✓ Robust to outliers ☺.
- ✓ Not good for models represented w/ many parameters ☹.



- ✓ Threshold for inliers ☹.
- ✓ Sufficient inliers at right so stop.

Other Fitting Approaches

- ✓ RANSAC: Randomized Sample Consensus.
- ✓ Robust to outliers ☺.
- ✓ Not good for models represented w/ many parameters ☹.



- ✓ Popular in dominant plane detection in 3D scenes.

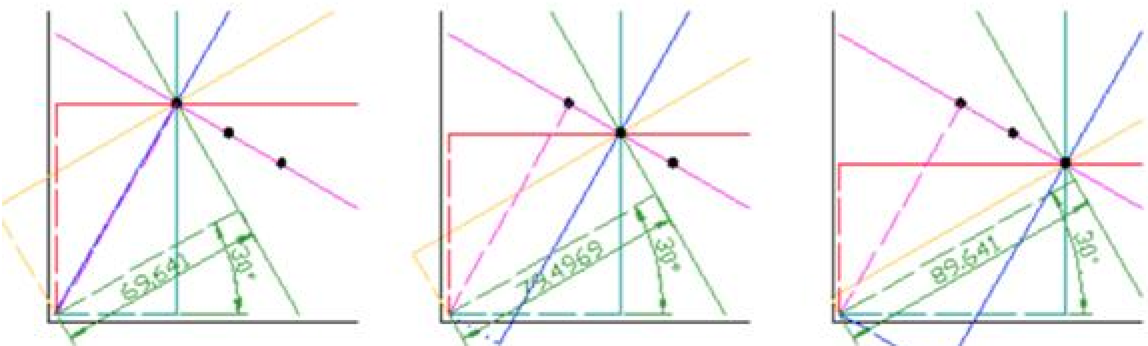
Other Fitting Approaches

- ✓ Hough Transform: each sample votes for all models (line, plane, ..) it supports. Vote is cast in the space of model parameters. Look for the model parameters w/ many votes.
- ✓ Robust to outliers ☺.
- ✓ Not good for models represented w/ many parameters ☹.

- ✓ Possible optimization for point clouds: at each point, vote only for planes that are roughly aligned with the estimated local normal.

Other Fitting Approaches

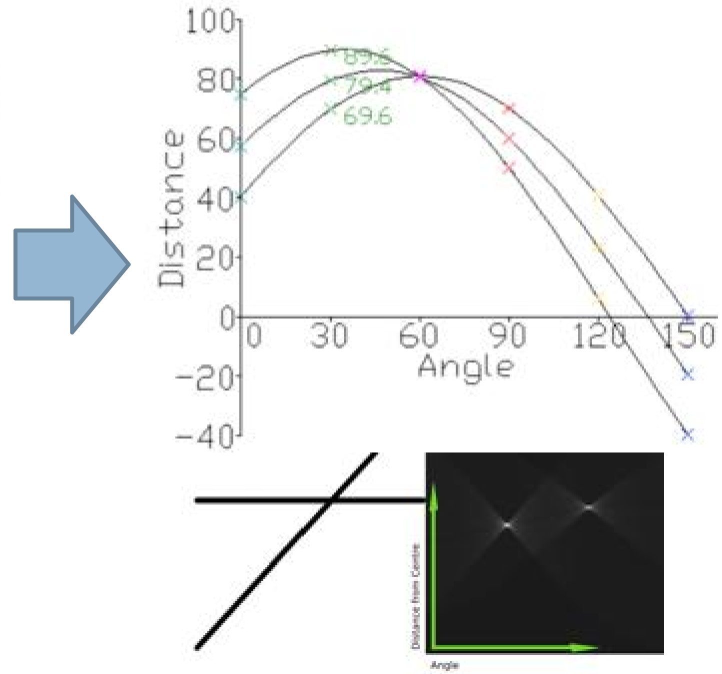
- ✓ Hough Transform: each sample votes for all models (line, plane, ..) it supports. Look for the model parameters w/ many votes.
- ✓ Cool demo by Wikipedia (parameters: angle & distance from origin):



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

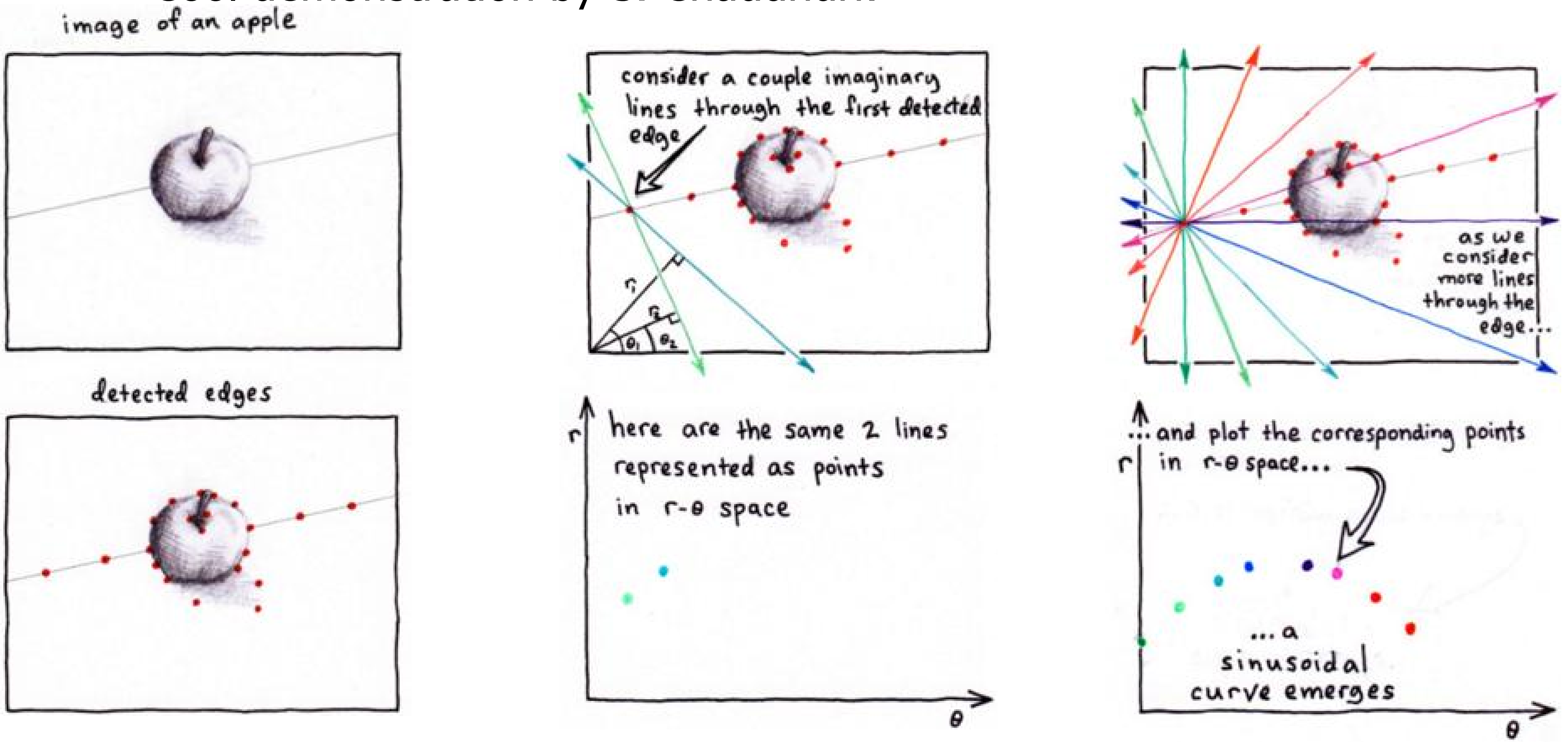
Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	−19.5

Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	−39.6



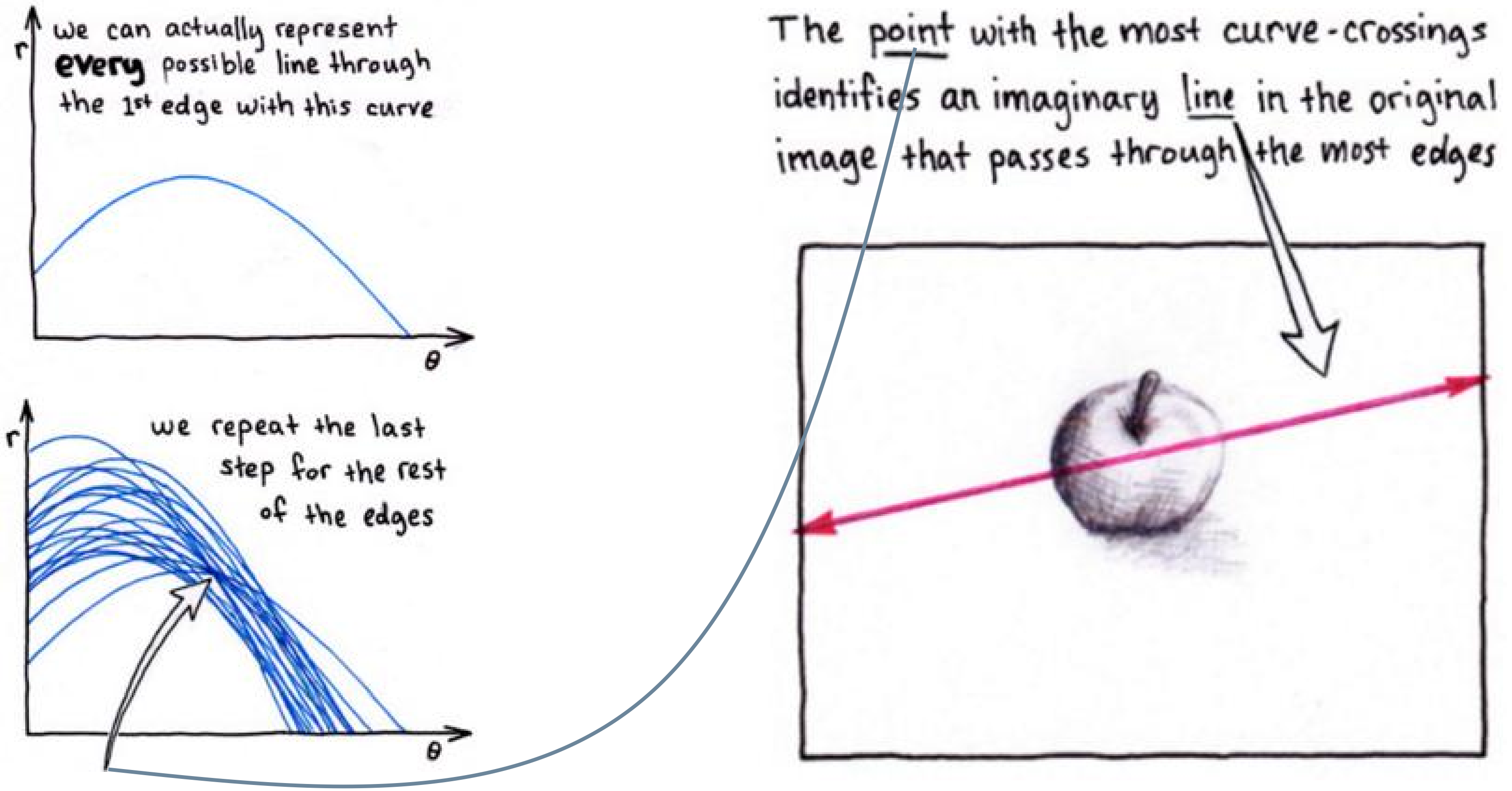
Other Fitting Approaches

- ✓ Hough Transform: each sample votes for all models (line, plane, ..) it supports. Look for the model parameters w/ many votes.
- ✓ Cool demonstration by S. Chaudhuri:



Other Fitting Approaches

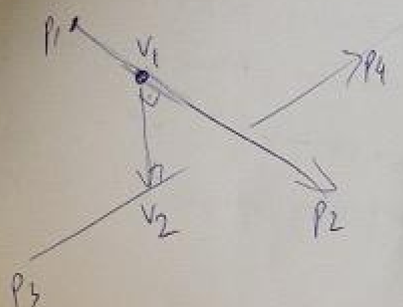
- ✓ Hough Transform: each sample votes for all models (line, plane, ..) it supports. Look for the model parameters w/ many votes.
- ✓ Cool demonstration by S. Chaudhuri:



Fitting Approaches Summary

- ✓ Least-squares.
  - ✓ Closed-form solution – superfast.
  - ✓ Robust to noise.
  - ✓ Not robust to outliers.
- ✓ RANSAC.
  - ✓ Robust to noise and outliers.
  - ✓ Support models w/ a moderate # parameters, 1-8.
- ✓ Hough transform.
  - ✓ Robust to noise and outliers.
  - ✓ Support models w/ a few # parameters, 1-4.

d = Distance between 2 lines in 3D = ?   (1)



$$(p_2-p_1)\cdot(v_2-v_1)=0 \qquad a\cdot x=0$$

$$(p_4-p_3)\cdot(v_2-v_1)=0 \qquad b\cdot x=0$$

$$a_0x_0+a_1x_1+a_2x_2=0$$
$$b_0x_0+b_1x_1+b_2x_2=0$$

2 eqs, 3 unknowns :((underdetermined)

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$A \quad x = b$

$x = A^{-1}b$ → then there's a unique soln.

If A is nonsingular (invertible), the rows of A are linearly independent, so are the cols

A is singular → det(A) = 0 → either no solutions – inconsistent or infinite " – consistent

A is singular bcs I can write row3 using row2

Rank of A: # of lin. indep. rows or cols (whichever smaller)

col1 using col2 & 3

$a_1\alpha + a_2\beta = a_0$

$b_1\alpha + b_2\beta = b_0$

solve for α & β.

$$v_1=(p_1+t(p_2-p_1))$$
$$v_2=(p_3+s(p_4-p_3))$$

$$v_2-v_1=p_3+sp_4-sp_3-p_1-tp_2+tp_1$$

$$(p_2-p_1)\cdot(\quad)=0$$
$$(p_4-p_3)\cdot(\quad)=0$$

2 eqs
2 unknowns :)

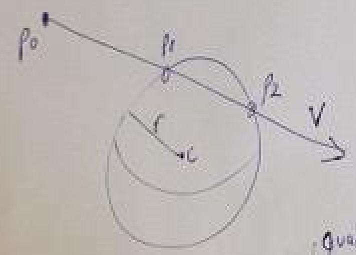
Solve for s & t

Find v_1 using t

" v_2 " s

$$d=\|v_2-v_1\|=\sqrt{(v_2-v_1)_0^2+(v_2-v_1)_1^2+(v_2-v_1)_2^2}=\sqrt{(v_2-v_1)^T(v_2-v_1)}$$

Intersection between a ray and a sphere?



A point on ray: $p_0 + t \cdot v$

If this makes r-distance to c, then it is on.

$$\|p_0 + t\cdot v - c\|^2 = r^2$$

p1 & p2

tangent

no solutions

quadratic in t; solving for t gives 2, 1 or 0 solutions

$$a = p_0 - c \Rightarrow \|a + tv\|^2 = r^2$$

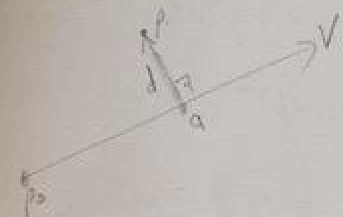
$$(a_0+tv_0 \;\; a_1+tv_1 \;\; a_2+tv_2)\begin{pmatrix}a_0+tv_0\\ a_1+tv_1\\ a_2+tv_2\end{pmatrix} = (a_0+tv_0)^2 + (a_1+tv_1)^2 + (a_2+tv_2)^2 = r^2$$

$(v_0^2+v_1^2+v_2^2)$

$$= a_0^2 + 2a_0tv_0 + t^2v_0^2 + a_1^2 + 2a_1tv_1 + t^2v_1^2 + a_2^2 + 2a_2tv_2 + t^2v_2^2 = V\cdot t^2 + W\cdot t + A - r^2 = 0$$

②

d = Distance between a point and line



$(p-a)\cdot v = 0$

$\downarrow$

$p_0 + t\cdot v$

$(p - p_0 - tv)\cdot v = 0$

$$\overbrace{\begin{pmatrix} b_0 - t\cdot v_0 \\ b_1 - t\cdot v_1 \\ b_2 - t\cdot v_2 \end{pmatrix}}^{b} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = b_0v_0 - tv_0^2 + b_1v_1 - tv_1^2 + b_2v_2 - tv_2^2 = 0$$

$$\rightarrow t = \frac{b_0v_0 + b_1v_1 + b_2v_2}{v_0^2 + v_1^2 + v_2^2}$$

$a = p_0 + t\cdot v$ known ✓ $\Rightarrow d = \|p - a\|$

d = Distance between a point and plane

$d = (p - p_0)\cdot n$ ✓ (unit vector) $= \frac{(p-p_0)\cdot n}{\|n\|}$ (in general)

Important to keep n unit but not (p-p0); see next slide

Alternative Solution: $(p' - p_0)\cdot n = 0$

$(p + \alpha\cdot n - p_0)\cdot n = 0$

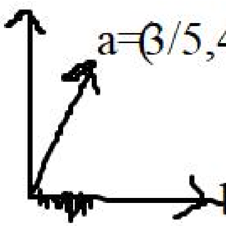
$(p - p_0 + \alpha n)\cdot n = 0$

$(p - p_0)\cdot n + \alpha\cdot n\cdot n = 0$

$$\alpha = \frac{(p_0 - p)\cdot n}{\|n\|^2}$$

$$d = \|p - p'\| = \|p - p - \alpha n\| = -\alpha\|n\| = \frac{-(p_0 - p)\cdot n}{\|n\|^2}\cdot\|n\| = \frac{(p - p_0)\cdot n}{\|n\|}$$ ✓

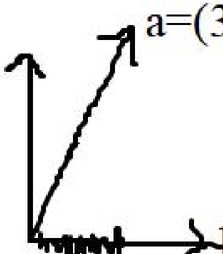
a DOT b is the scalar projection of vector a in the direction of UNIT vector b; so important to keep b unit but not a



a=(3/5,4/5) (unit)

b=(1,0)

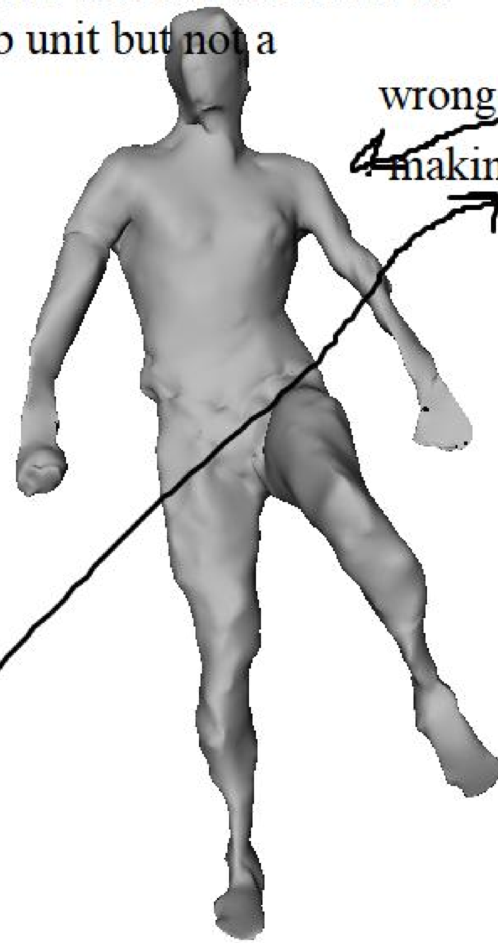
3/5*1+4/5*0=3/5



a=(3,4)

b=(1,0)

3*1+4*0=3 (don't make a unit!!!!!!!)



wrong dotPrdc

making a unit

Potential Project Topics

- ✓ Compare least-squares, RANSAC, and Hough plane fits to 3D data.
- ✓ Implement the Least-Squares Meshes paper or any other related paper.
- ✓ Hough transform meets Deep learning; implement the paper titled Deep Hough Voting for 3D Object Detection in Point Clouds.