Mathematical Foundation for Robotics

Mayank Mittal

AE640A: Autonomous Navigation

January 31, 2018

Course Announcements

- Assignment 1 due on February 5 (next week)
- No mid-semester examinations
- **Project Proposal submission** due ~~January 30~~ **February 2**
  - **Research problem**: description of the problem
  - **Technical approach:** technical plan and explain the project design
  - **Work Plan**: provide timeline and tasks assigned to each team member
  - **Expected Outcome**: describe how you are planning to demonstrate your project
  - **References** (Optional): You may describe what other people have done to solve similar problems and how your approach is different from the existing work

Outline

- Least Square Estimation
- RANSAC Algorithm
- Concept of Random Variables
  - Probability Mass Function (PMF): Joint and Marginal
  - Independent Random Variables
  - Expectation (Mean) of Random Variable
  - Variance, Covariance
  - Bayes rule
- Gaussian / Normal Distribution
- Introduction to Bayesian Framework

Least Square Estimation

Consider the system model:

$$Y = X\beta + \epsilon$$

such that β is the parameter vector which needs to be estimated, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is the noise/error in the observations which needs to be minimized.

Here, Y is the observations taken and X is the system design matrix.

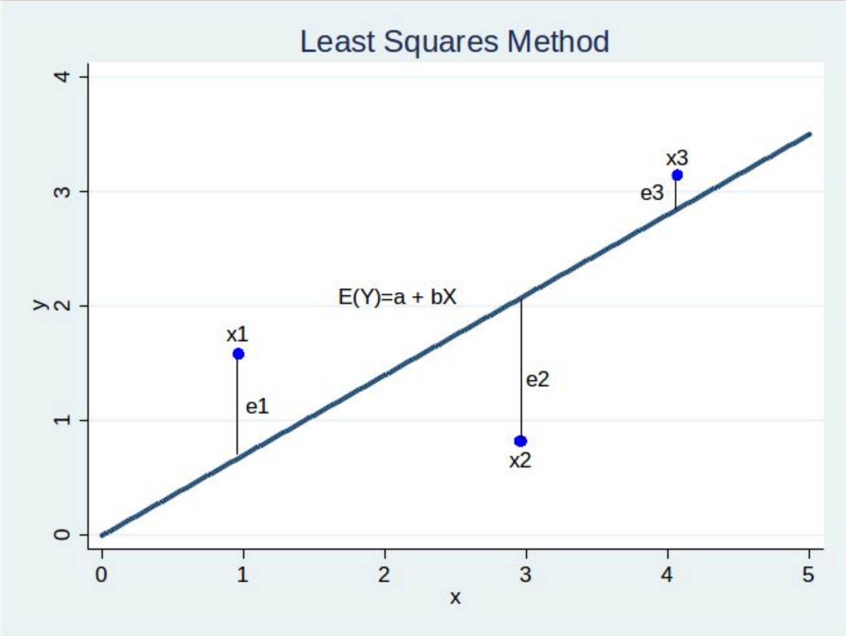


Image Credits: Robert Collins, CSE486, Penn State

Least Square Estimation

Least squares method minimizes the sum of squares of errors (deviations of individual data points form the regression line)

That is, the least square estimation problem can be written as:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\epsilon^T \epsilon)$$

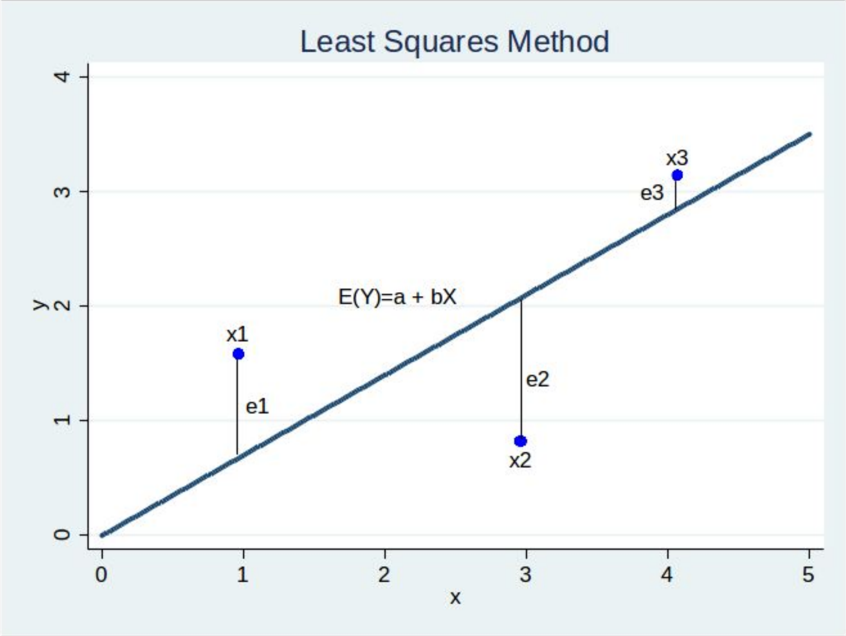


Image Credits: Robert Collins, CSE486, Penn State

Least Square Estimation

$$Y = X\beta + \epsilon$$

$$\implies \quad E(Y) = \hat{Y} = X\beta$$

$$\implies \quad \epsilon = Y - \hat{Y} = Y - X\beta$$

$$\implies \quad \epsilon^T\epsilon = (Y - X\beta)^T(Y - X\beta)$$

$$\implies \quad \epsilon^T\epsilon = Y^TY - 2Y^TX\beta + \beta^TX^TX\beta$$

To minimize the objective function $\epsilon^T\epsilon$, we shall evaluate the partial derivative of it w.r.t. β and equate it to zero

Least Square Estimation

$$\epsilon^T \epsilon = Y^T Y - 2Y^T X\beta + \beta^T X^T X\beta$$

To minimize the objective function $\epsilon^T \epsilon$, we shall evaluate the partial derivative of it w.r.t. β and equate it to zero

$$\frac{\partial \epsilon^T \epsilon}{\partial \beta} = -2X^T Y + 2X^T X\beta = 0$$

$$\implies \quad X^T X\beta = X^T Y$$

$$\implies \quad \beta = (X^T X)^- X^T Y$$

Least Square Estimation: Drawbacks

- Least squares estimation is sensitive to outliers, so that a few outliers can greatly skew the result.

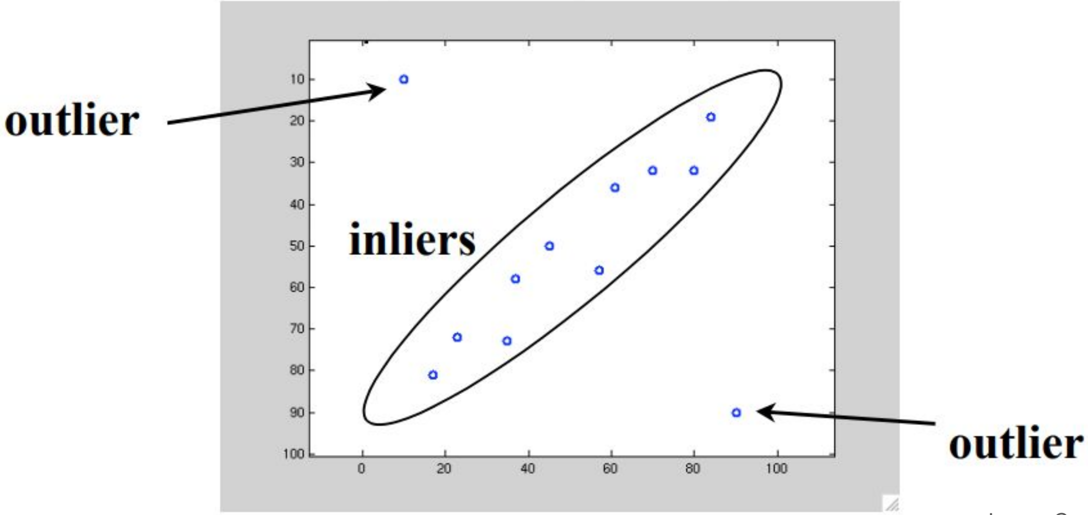
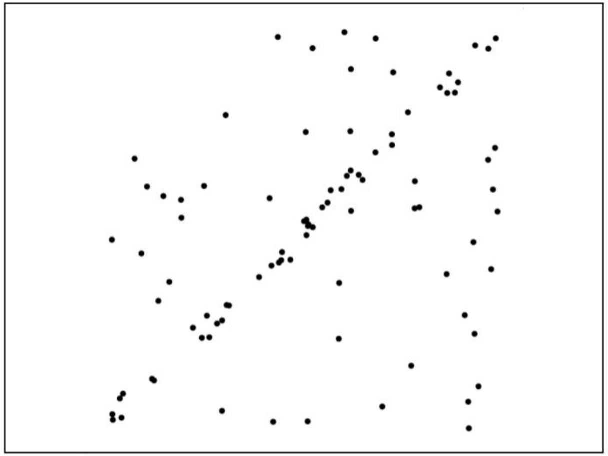
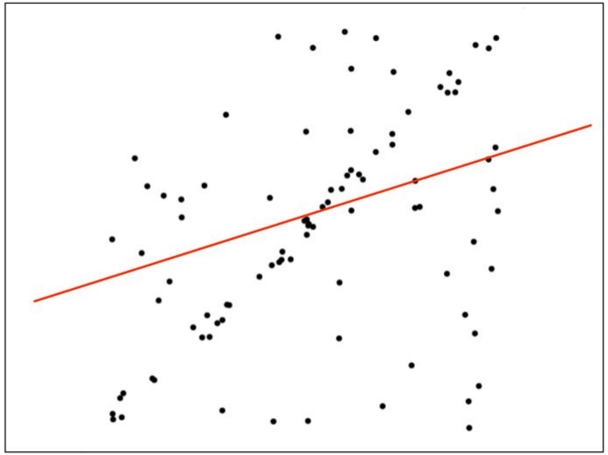


Image Credits: Robert Collins, CSE486, Penn State

Least Square Estimation: Drawbacks

- Least squares estimation is sensitive to outliers, so that a few outliers can greatly skew the result.





Least Squares Fit

Slide Credit:
http://cs.gmu.edu/~kosecka/cs682/lect---fitting.pdf

Least Square Estimation: Drawbacks

- Multiple structures can also skew the results. (the fit procedure implicitly assumes there is only one instance of the model in the data)

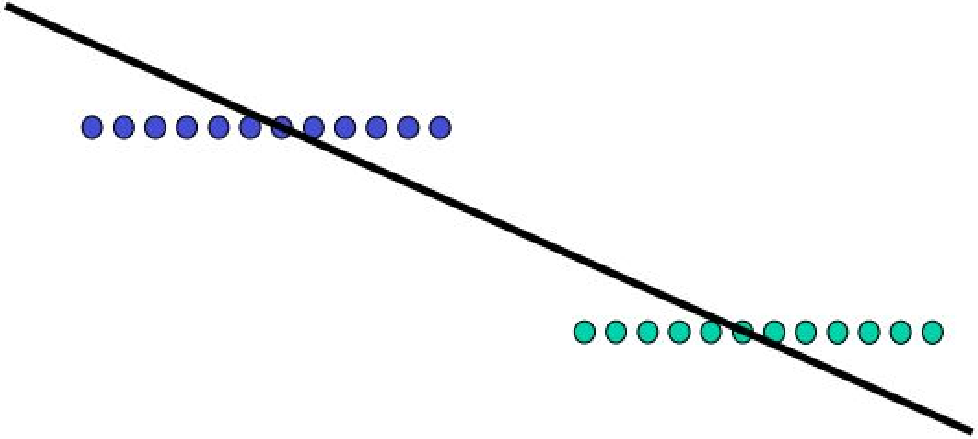


Image Credits: Robert Collins, CSE486, Penn State

Robust Estimation

- View estimation as a two-stage process:
  - Classify data points as outliers or inliers
  - Fit model to inliers while ignoring outliers

- Example technique: **RANSAC (RANdom SAmple Consensus)**

  - M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381--395.

RANSAC Procedure

- Assume:
  - The parameters can be estimated from *n* data items.
  - There are M data items in total.
  - The probability of a randomly selected data item being part of a good model is p_g
  - The probability that the algorithm will exit without finding a good fit if one exists is p_fail

- Algorithm:
  1. select *n* data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

More info: https://en.wikipedia.org/wiki/Random_sample_consensus

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

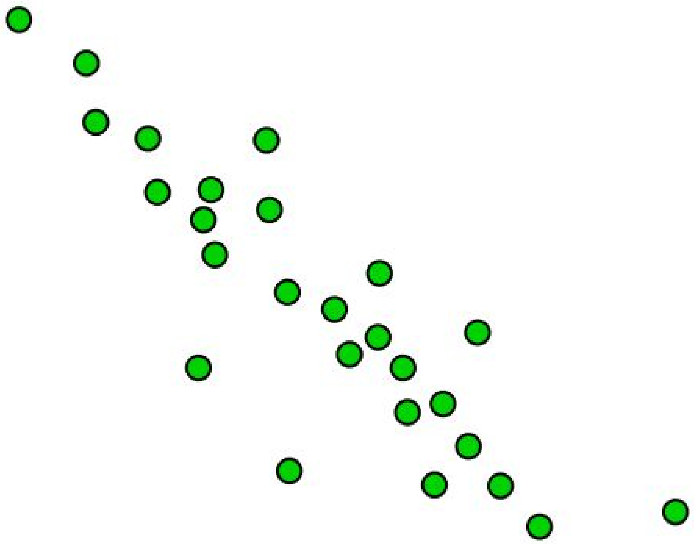


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

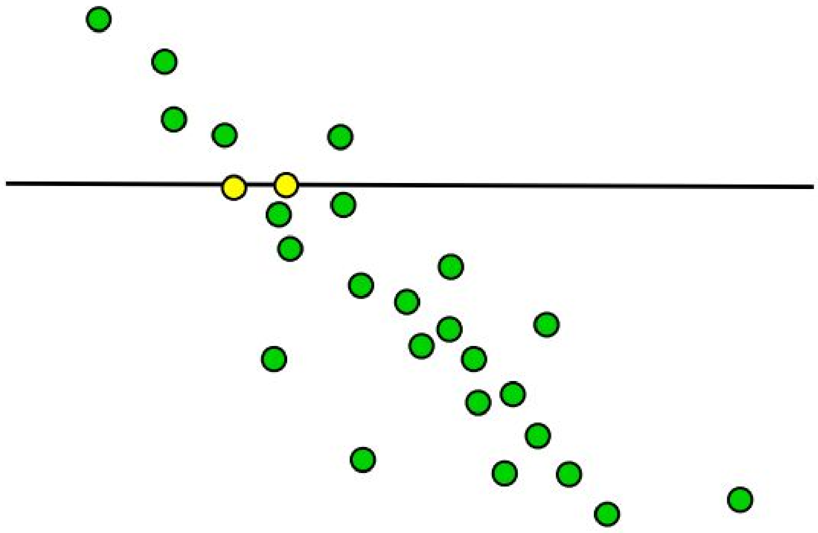


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
    1. select *n*=2 data items at random
    2. estimate parameters of the model
    3. find how many data items (of M) fit the model within a user given tolerance
    4. if number of inliers are big enough, accept fit and exit with success
    5. repeat 1-4 N times
    6. fail if you get here

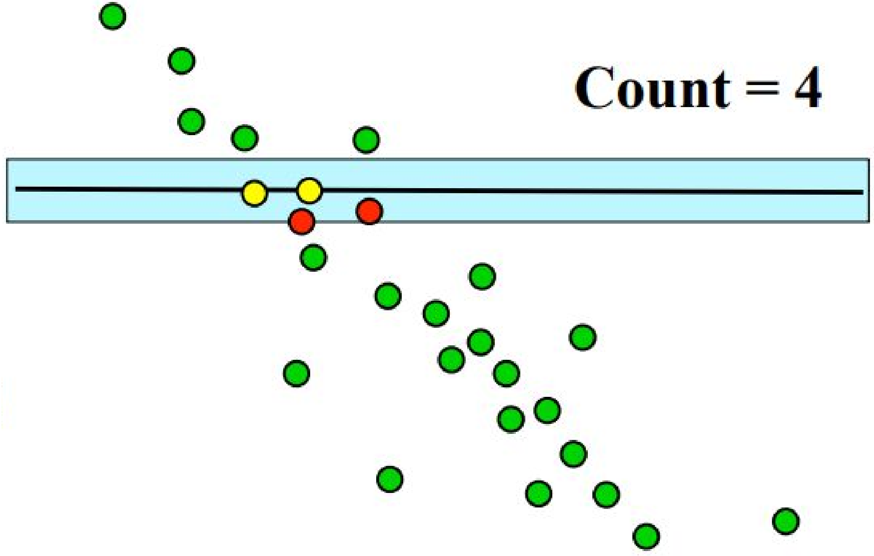


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

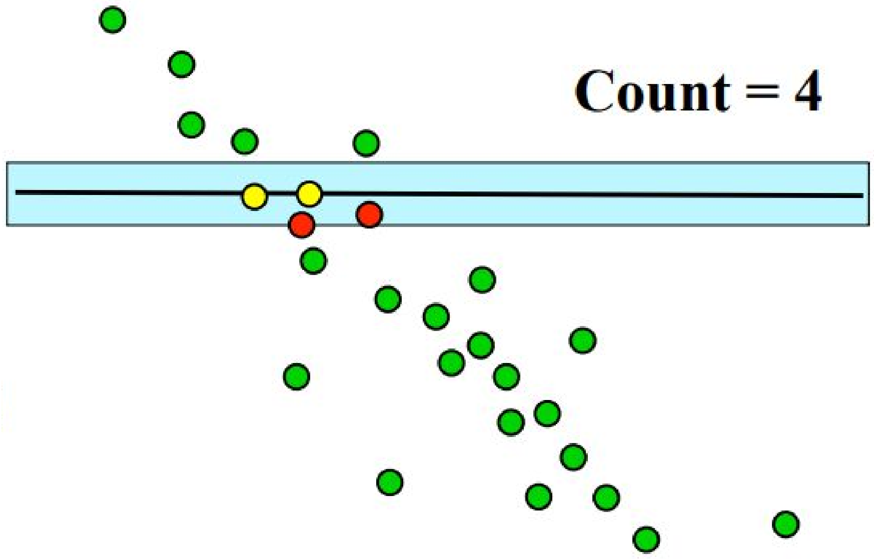


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
    1. select *n*=2 data items at random
    2. estimate parameters of the model
    3. find how many data items (of M) fit the model within a user given tolerance
    4. if number of inliers are big enough, accept fit and exit with success
    5. repeat 1-4 N times
    6. fail if you get here

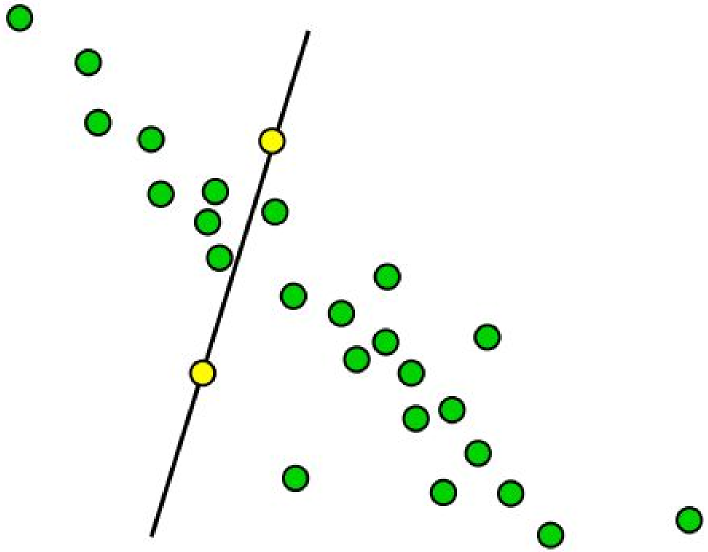


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

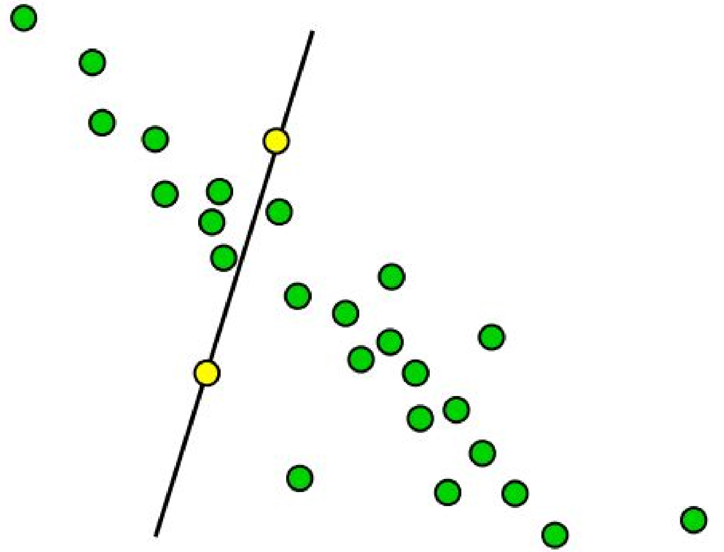


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
    1. select n=2 data items at random
    2. estimate parameters of the model
    3. find how many data items (of M) fit the model within a user given tolerance
    4. if number of inliers are big enough, accept fit and exit with success
    5. repeat 1-4 N times
    6. fail if you get here

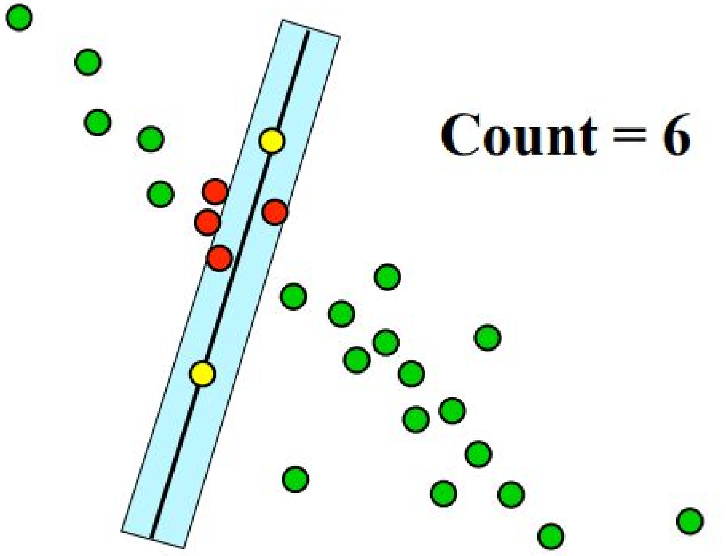


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

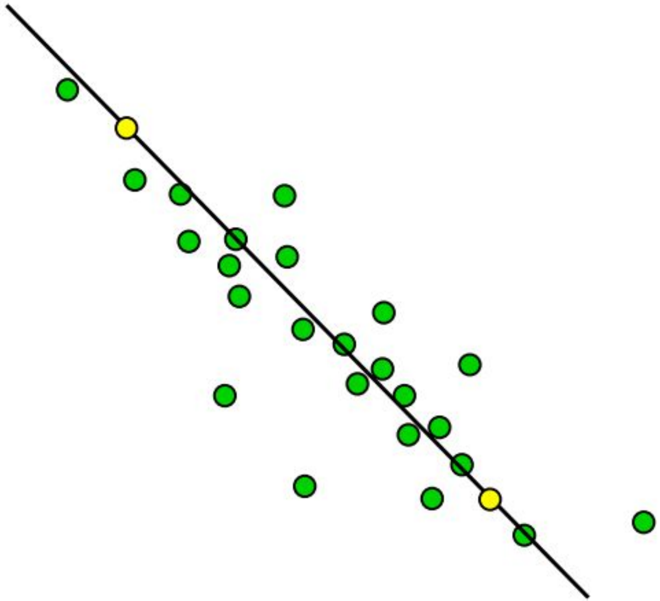


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
    1. select n=2 data items at random
    2. estimate parameters of the model
    3. find how many data items (of M) fit the model within a user given tolerance
    4. if number of inliers are big enough, accept fit and exit with success
    5. repeat 1-4 N times
    6. fail if you get here

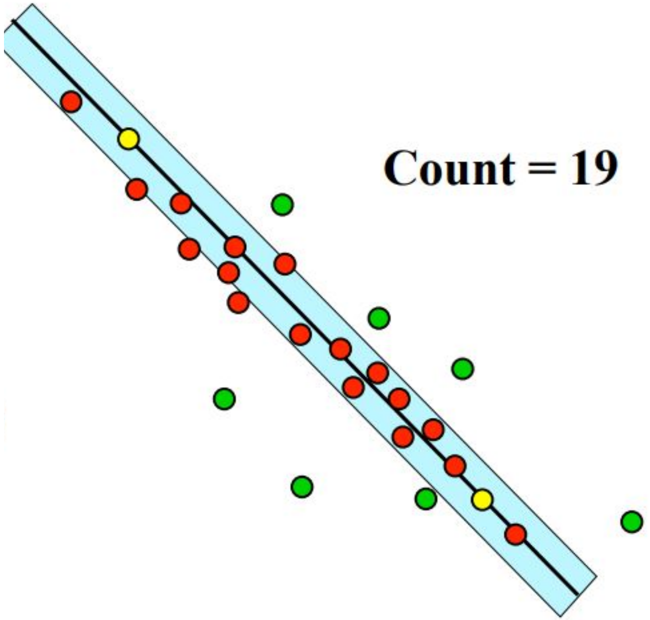


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

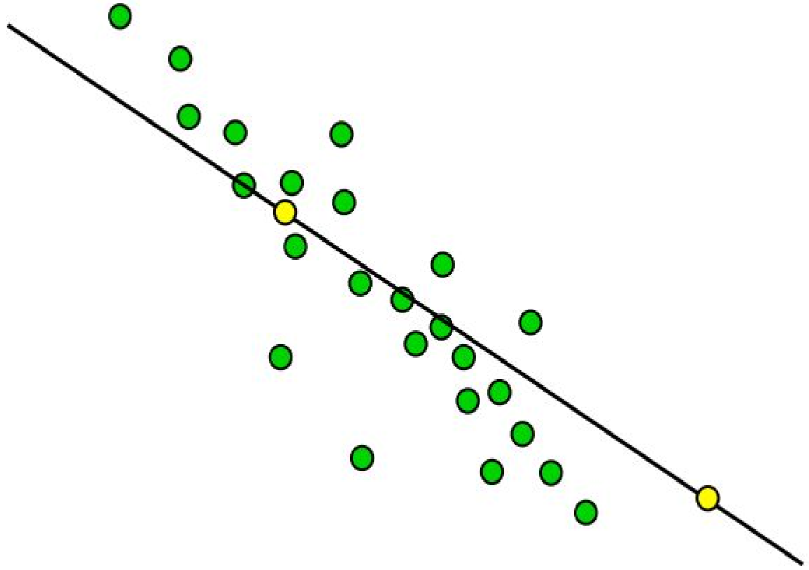


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
    1. select n=2 data items at random
    2. estimate parameters of the model
    3. find how many data items (of M) fit the model within a user given tolerance
    4. if number of inliers are big enough, accept fit and exit with success
    5. repeat 1-4 N times
    6. fail if you get here

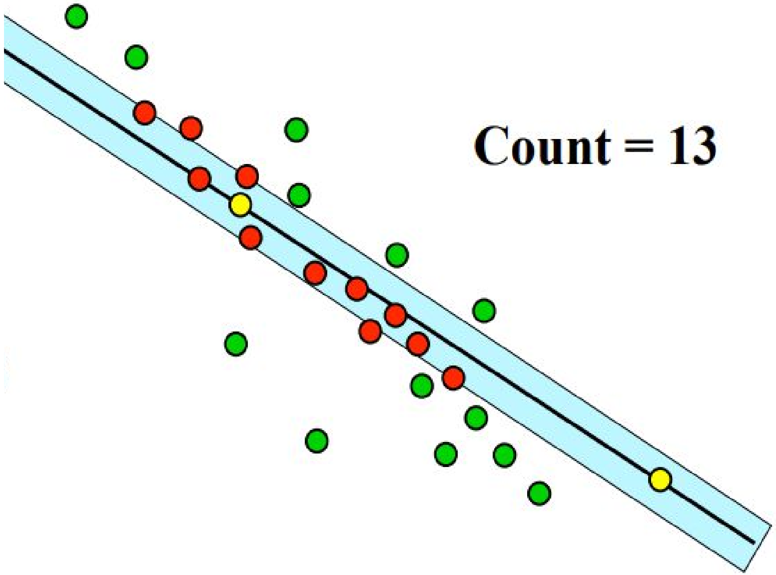


Image Credits: Robert Collins, CSE486, Penn State

RANSAC Procedure

- Algorithm:
  1. select n=2 data items at random
  2. estimate parameters of the model
  3. find how many data items (of M) fit the model within a user given tolerance
  4. if number of inliers are big enough, accept fit and exit with success
  5. repeat 1-4 N times
  6. fail if you get here

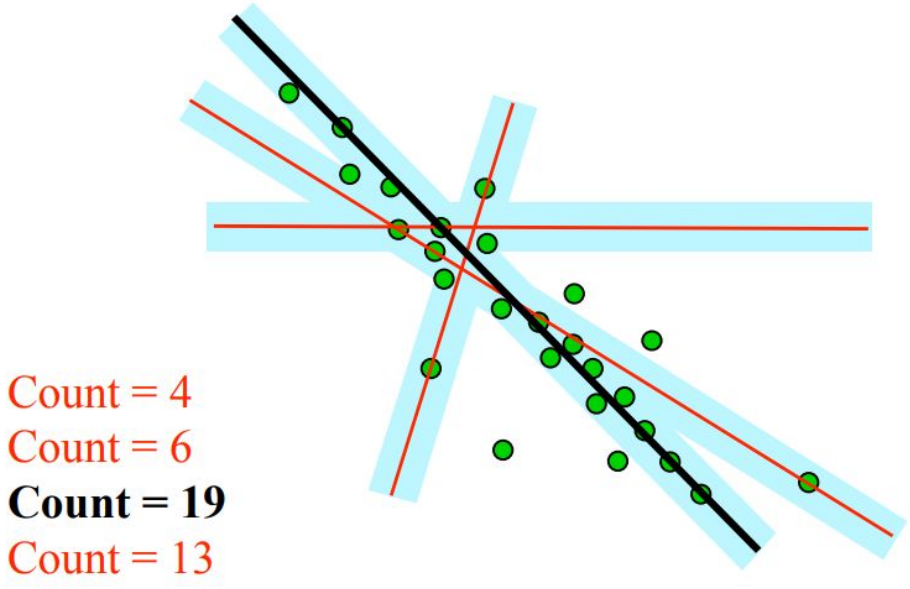


Image Credits: Robert Collins, CSE486, Penn State

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$1 - \left(1 - (\underbrace{1 - e}_{\text{Probability that choosing one point yields an inlier}})^{s}\right)^{N} = p$$

Image Credits: Robert Collins, CSE486, Penn State

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$1 - \left(1 - \underbrace{(1 - e)^s}_{\text{Probability of choosing s inliers in a row (sample only contains inliers)}}\right)^N = p$$

Image Credits: Robert Collins, CSE486, Penn State

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$1 - (\underbrace{1 - (1 - e)^{s}}_{\text{Probability that one or more points in the sample were outliers (sample is contaminated).}})^{N} = p$$

Image Credits: Robert Collins, CSE486, Penn State

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$1 - \underbrace{\left(1 - (1 - e)^{s}\right)^{N}}_{\text{Probability that N samples were contaminated.}} = p$$

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$\underbrace{1 - (1 - (1 - e)^{s})^{N}}_{\text{Probability that at least one sample was not contaminated (at least one sample of } s \text{ points is composed of only inliers}} = p$$

RANSAC: How many samples to choose?

- Suppose:
  - *e*: probability that the point is an outlier
  - *s*: number of points in a sample
  - *N*: number of times sampling to be done / iterations
  - *p:* desired probability that we get a good sample

$$N = \frac{log(1-p)}{log(1-(1-e)^s)}$$

RANSAC: How many samples to choose?

- Example:
  - desired probability that we get a good sample, $p = 0.99$

$$N = \frac{log(1-p)}{log(1-(1-e)^s)}$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

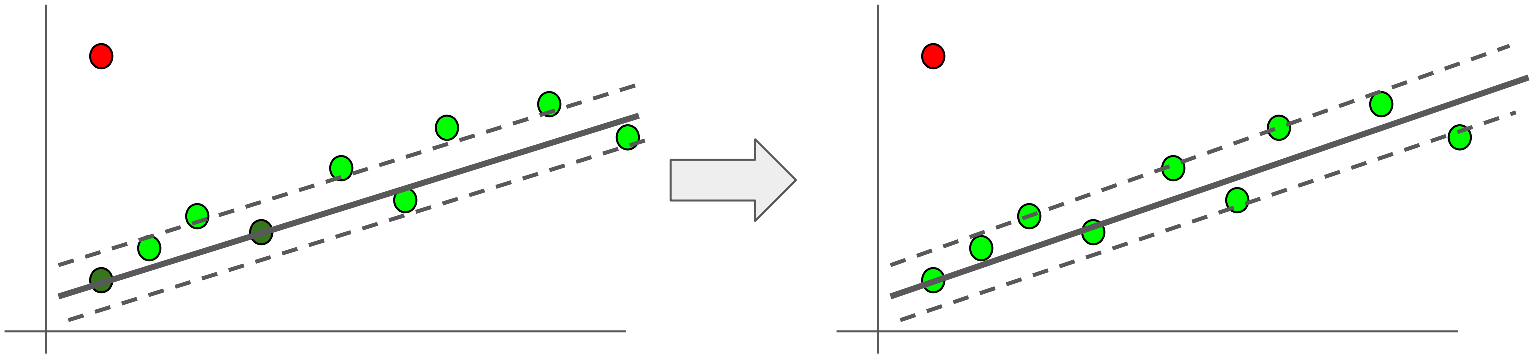
RANSAC: How many samples to choose?

- We don’t always need to exhaustively sample subsets of points, we just need to randomly sample N subsets. Typically, we don’t even have to sample N sets!
- **Early termination**: terminate when inlier ratio reaches expected ratio of inliers

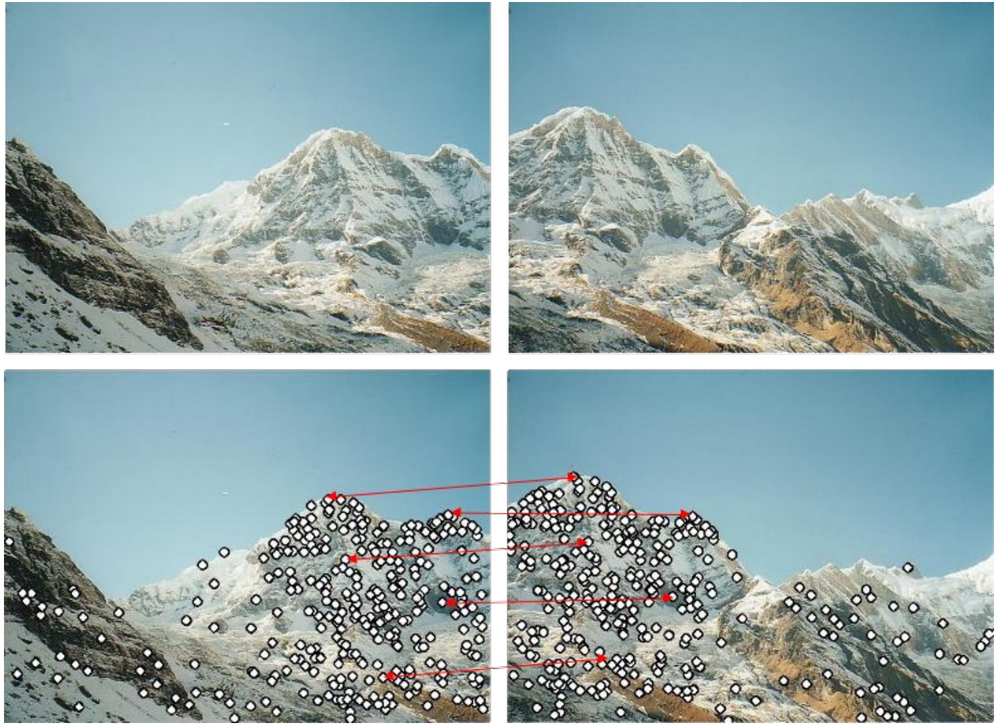
$$T = (1 - e) \times \text{total number of data points}$$

After RANSAC

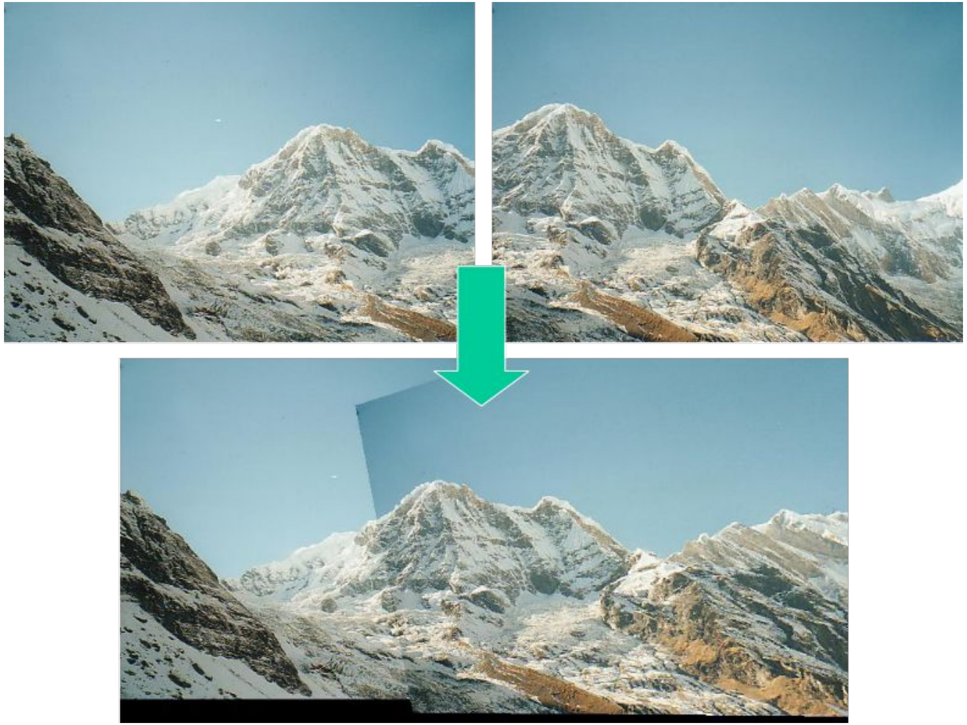
- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers with greatest support
- Improve this initial estimate with Least Squares estimation over all inliers (i.e., standard minimization)
- Find inliers wrt that L.S. line, and compute L.S. one more time



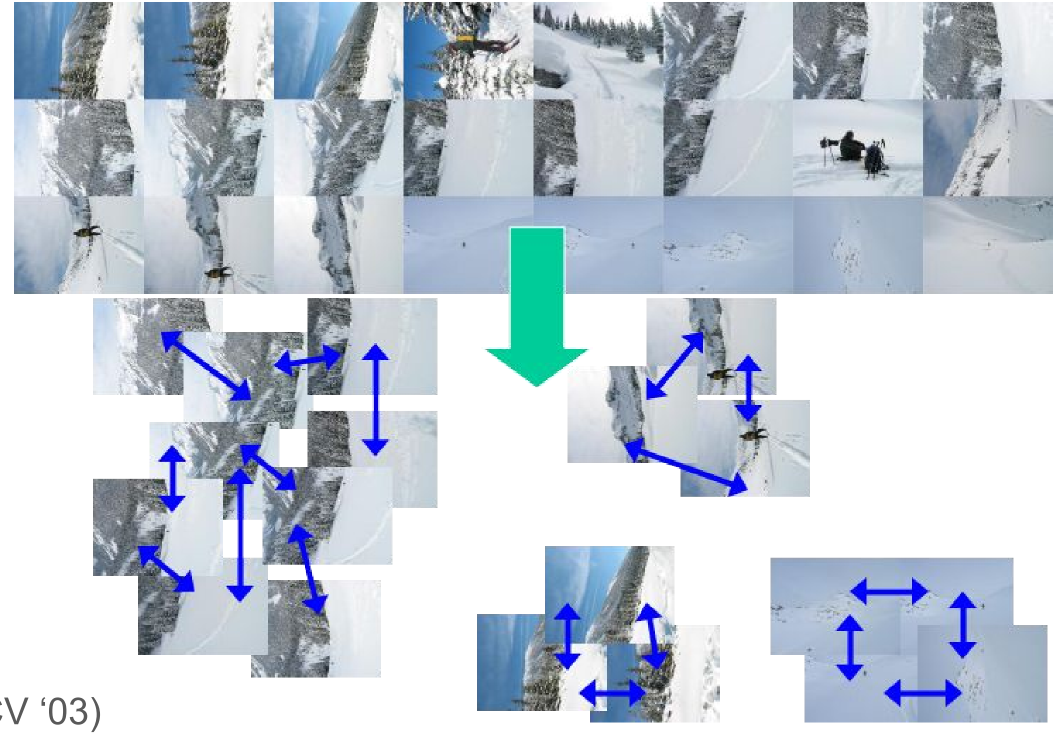
RANSAC Applications: Automatic Image Stitching



RANSAC Applications: Automatic Image Stitching

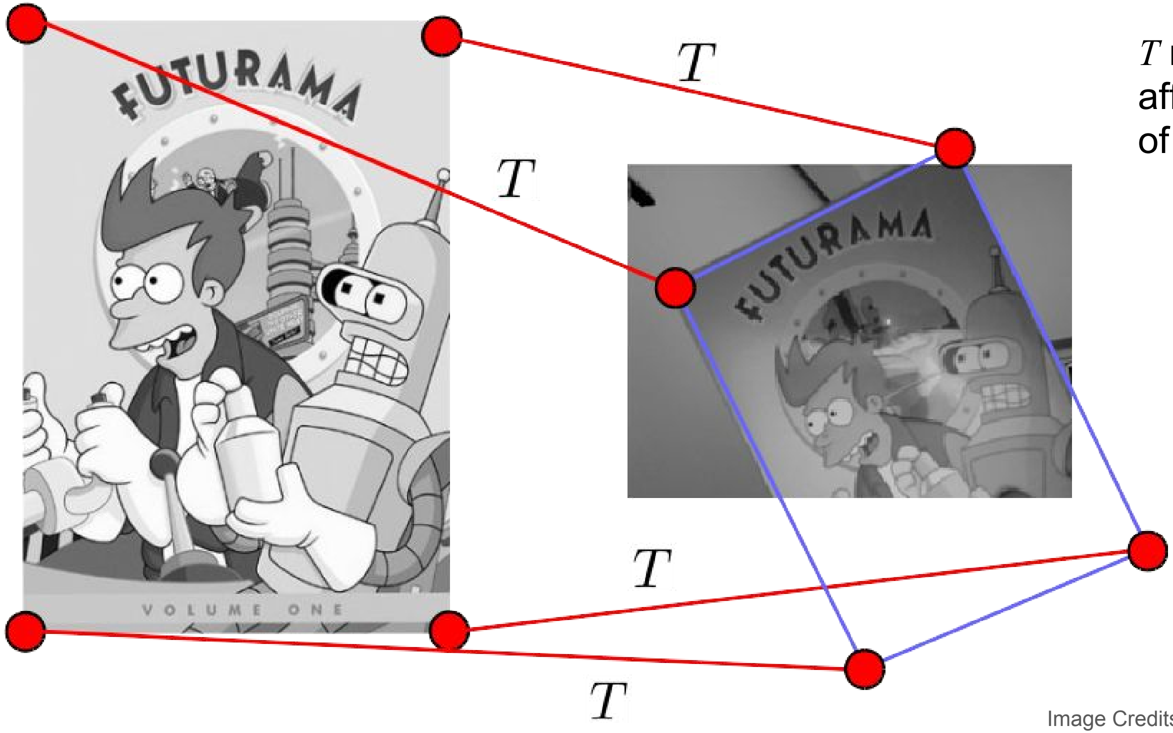


RANSAC Applications: Finding the Panoramas



(Brown & Lowe, ICCV '03)

RANSAC Applications: Finding the Object Boundary

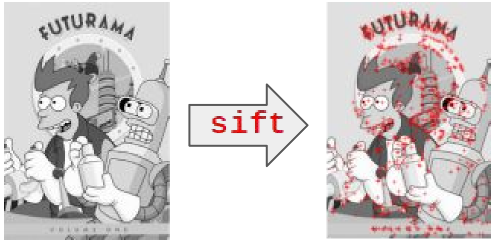


T represents the affine transformation of the points

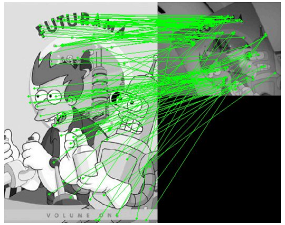
Image Credits: Noah Snavely, Cornell University

RANSAC Applications: Finding the Object Boundary

1. Detect features in the template and search images



2. Match features: find "similar-looking" features in the two images



3. Find a transformation T that explains the movement of the matched features

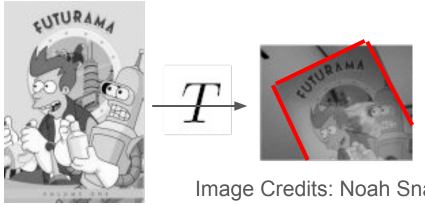


Image Credits: Noah Snavely, Cornell University

Probabilistic Robotics

Real-world data is noisy and uncertain (and often limited if data acquisition is expensive/difficult)

Key idea: Explicit representation of uncertainty (using the calculus of probability theory)

- Perception = state estimation
- Control = utility optimization

Discrete Random Variables

- *X* denotes a random variable
- *X* can take on a countable number of values in {x1, x2, …, xn}.
- *P(X=xi)*, is the probability that the random variable X takes on value *xi*
- *P(.)* is called probability mass function.
- *Example*: Let *X* represent the sum of two dice, then the probability distribution of *X* is as follows:

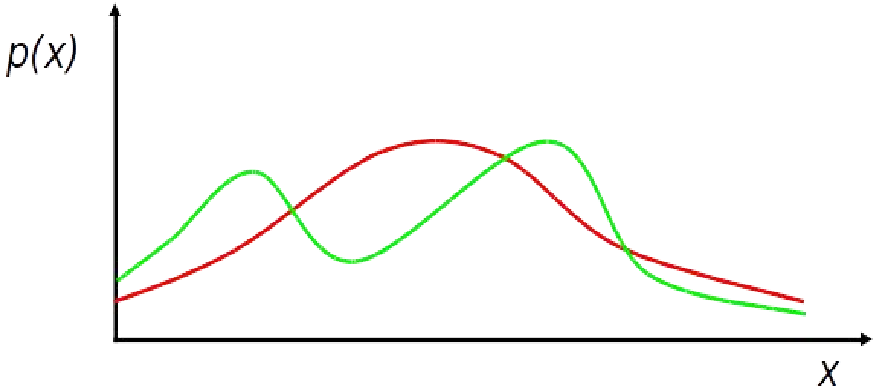
X	2	3	4	5	6	7	8	9	10	11	12
P(*X*)	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

Continuous Random Variables

- *X* takes on values in the *continuum*
- *p(X = x), or p(x),* is a probability density function

$$\Pr(x \in (a,b)) = \int_a^b p(x)dx$$

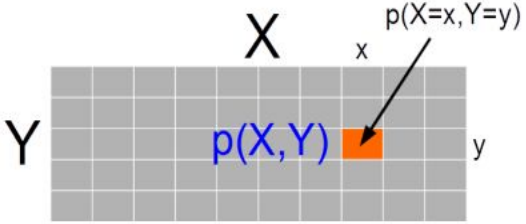
- *Example:*
  - time it takes to get to school
  - distance traveled between classes



Joint Probability Distribution

- *Joint probability distribution p(X, Y)* models probability of co-occurrence of two r.v. *X, Y*
- For discrete r.v., the joint PMF *p(X, Y)* is like a table (that sums to 1)

$$\sum_x \sum_y p(X = x, Y = y) = 1$$



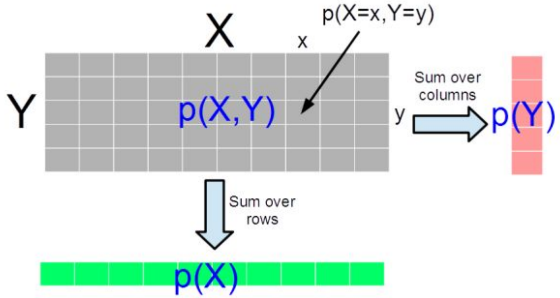
- For continuous r.v.:

$$\int_x \int_y p(X = x, Y = y) dxdy = 1$$

Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Marginal Probability Distribution

- Intuitively, the probability distribution of one r.v. regardless of the value the other r.v. takes
- For discrete r.v.'s: $p(X) = \sum_y p(X, Y = y), \quad p(Y) = \sum_x p(X = x, Y)$
- For discrete r.v. it is the sum of the PMF table along the rows/columns:

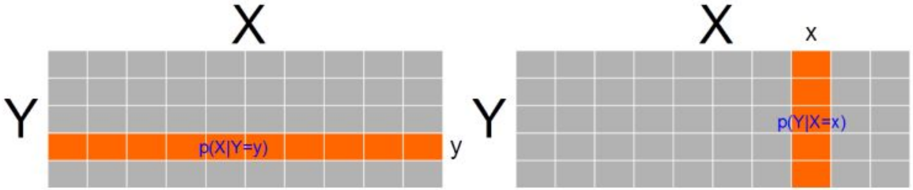


- Note: Marginalization is also called **"integrating out"**

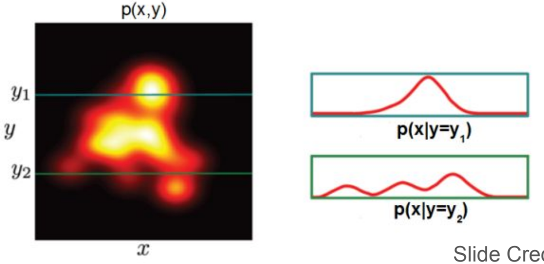
Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Conditional Probability Distribution

- Probability distribution of one r.v. given the value of the other r.v.
- Conditional probability p(X|Y = y) or p(Y |X = x): like taking a slice of p(X, Y) -
- For a discrete distribution:



- For a continuous distribution:

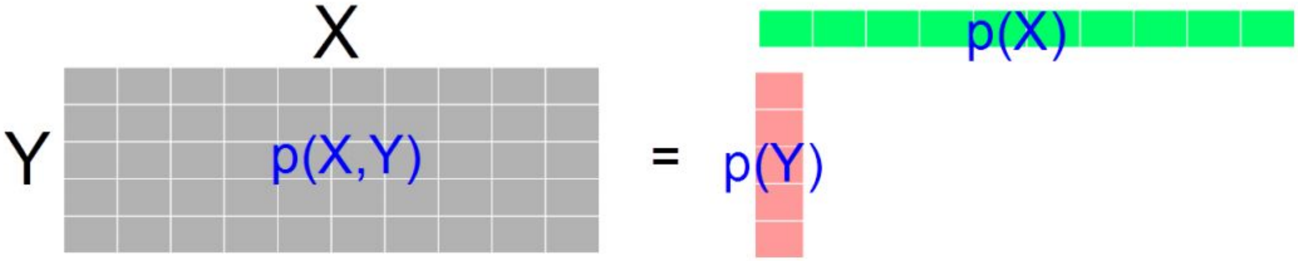


Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)
Image Credits: Noah Snavely, Cornell University

Independence

- X and Y are independent when knowing one tells nothing about the other

$$
\begin{aligned}
p(X|Y=y) &= p(X) \\
p(Y|X=x) &= p(Y) \\
p(X,Y) &= p(X)p(Y)
\end{aligned}
$$



Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Law of Total Probability

Discrete case

$$\sum_x P(x) = 1$$

$$P(x) = \sum_y P(x, y)$$

$$P(x) = \sum_y P(x \mid y) P(y)$$

Continuous case

$$\int p(x)\, dx = 1$$

$$p(x) = \int p(x, y)\, dy$$

$$p(x) = \int p(x \mid y)\, p(y)\, dy$$

Bayes Formula

$$P(x,y) = P(x|y)P(y) = P(y|x)P(x)$$

$$P(x|y) = \frac{P(y|x)\ P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

Conditional Independence

$$P(x, y \mid z) = P(x \mid z) P(y \mid z)$$

equivalent to:

$$P(x \mid z) = P(x \mid z, y)$$

and,

$$P(y \mid z) = P(y \mid z, x)$$

Expectation

- **Expectation** or **mean** μ of an r.v. with PMF/PDF $p(X)$

$$\mathbb{E}[X] = \sum_x xp(x) \quad \text{(for discrete distributions)}$$

$$\mathbb{E}[X] = \int_x xp(x)dx \quad \text{(for continuous distributions)}$$

- **Note:** The definition applies to **functions of r.v.** too (e.g., $\mathbb{E}[f(X)]$)
- **Note:** Expectations are always w.r.t. the underlying probability distribution of the random variable involved, so sometimes we'll write this explicitly as $\mathbb{E}_{p()}[.]$, unless it is clear from the context
- **Linearity of expectation**

$$\mathbb{E}[\alpha f(X) + \beta g(Y)] = \alpha\mathbb{E}[f(X)] + \beta\mathbb{E}[g(Y)]$$

(a very useful property, true even if X and Y are not independent)

- **Rule of iterated/total expectation**

$$\mathbb{E}_{p(X)}[X] = \mathbb{E}_{p(Y)}[\mathbb{E}_{p(X|Y)}[X|Y]]$$

Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Variance and Covariance

- **Variance** σ^2 (or "spread" around mean μ) of an r.v. with PMF/PDF $p(X)$

$$\text{var}[X] = \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2] - \mu^2$$

- **Standard deviation:** $\text{std}[X] = \sqrt{\text{var}[X]} = \sigma$
- For two scalar r.v.'s x and y, the **covariance** is defined by

$$\text{cov}[x, y] = \mathbb{E}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]$$

- For **vector** r.v. $\boldsymbol{x}$ and $\boldsymbol{y}$, the **covariance matrix** is defined as

$$\text{cov}[\boldsymbol{x}, \boldsymbol{y}] = \mathbb{E}\left[\{\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]\}\{\boldsymbol{y}^T - \mathbb{E}[\boldsymbol{y}^T]\}\right] = \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^T] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{y}^\top]$$

- Cov. of components of a vector r.v. $\boldsymbol{x}$: $\text{cov}[\boldsymbol{x}] = \text{cov}[\boldsymbol{x}, \boldsymbol{x}]$
- **Note:** The definitions apply to functions of r.v. too (e.g., $\text{var}[f(X)]$)
- **Note:** Variance of sum of independent r.v.'s: $\text{var}[X+Y] = \text{var}[X] + \text{var}[Y]$

Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Transformation of Random Variables

Suppose $\boldsymbol{y} = f(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} + \mathbf{b}$ be a linear function of an r.v. $\boldsymbol{x}$

Suppose $\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu}$ and $\text{cov}[\boldsymbol{x}] = \boldsymbol{\Sigma}$

- Expectation of $\boldsymbol{y}$

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\mathbf{A}\boldsymbol{x} + \mathbf{b}] = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$$

- Covariance of $\boldsymbol{y}$

$$\text{cov}[\boldsymbol{y}] = \text{cov}[\mathbf{A}\boldsymbol{x} + \mathbf{b}] = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T$$

Likewise if $y = f(\boldsymbol{x}) = \boldsymbol{a}^T\boldsymbol{x} + b$ is a scalar-valued linear function of an r.v. $\boldsymbol{x}$:

- $\mathbb{E}[y] = \mathbb{E}[\boldsymbol{a}^T\boldsymbol{x} + b] = \boldsymbol{a}^T\boldsymbol{\mu} + b$
- $\text{var}[y] = \text{var}[\boldsymbol{a}^T\boldsymbol{x} + b] = \boldsymbol{a}^T\boldsymbol{\Sigma}\boldsymbol{a}$

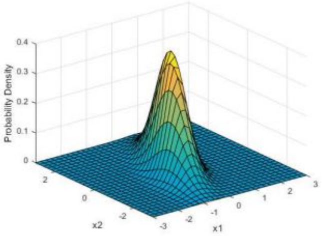
Another very useful property worth remembering

Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Multivariate Gaussian Distribution

- Distribution over a multivariate r.v. vector $\boldsymbol{x} \in \mathbb{R}^D$ of real numbers
- Defined by a **mean vector** $\boldsymbol{\mu} \in \mathbb{R}^D$ and a $D \times D$ **covariance matrix** $\boldsymbol{\Sigma}$

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

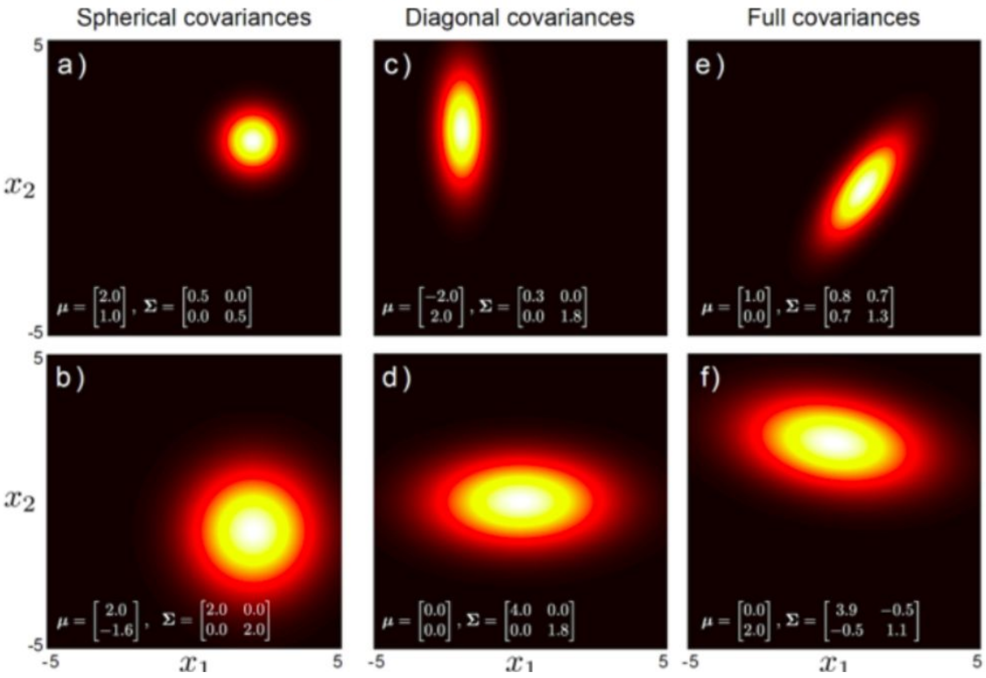


- The covariance matrix $\boldsymbol{\Sigma}$ must be symmetric and positive definite
  - All eigenvalues are positive
  - $\boldsymbol{z}^\top \boldsymbol{\Sigma} \boldsymbol{z} > 0$ for any real vector $\boldsymbol{z}$
- Often we parameterize a multivariate Gaussian using the inverse of the covariance matrix, i.e., the **precision matrix** $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$

Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Multivariate Gaussian: The Covariance Matrix

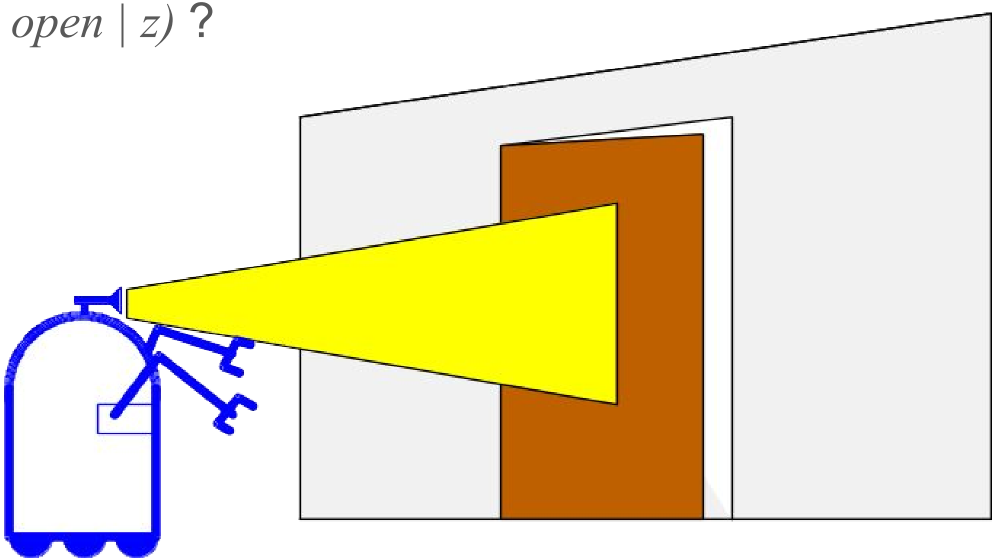
The covariance matrix can be spherical, diagonal, or full



Slide Credit: Prof. Piyush Rai (Probability Refresher Tutorial)

Simple Example of State Estimation

- Consider an environment where the state of door $x = \{open, close\}$ needs to be estimated. The robot obtains a measurement z from its environment.
- What is $P(x = open \mid z)$?



Causal vs. Diagnostic Reasoning

- $P(x = open \mid z)$ is diagnostic
- $P(z \mid x = open)$ is causal
- Often causal knowledge is easier to obtain
- Bayes rule allows us to use causal knowledge:

Posterior

Priori

$$P(x = open|z) = \frac{P(z|x = open)P(x = open)}{P(z)}$$

Example

- Priori: $P(x = open) = P(x = close) = 0.5$
- $P(z \mid x = open) = 0.6$; $P(z \mid x = close) = 0.3$

- Using Bayes rule:

$$P(x = open|z) = \frac{P(z|x = open)P(x = open)}{P(z)}$$

$$\implies \quad P(x = open|z) = \frac{P(z|x = open)P(x = open)}{P(z|x = open)P(x = open) + P(z|x = open)P(x = close)}$$

$$\implies \quad P(x = open|z) = \frac{0.6 \times 0.5}{0.6 \times 0.5 + 0.3 \times 0.5}$$

$$\implies \quad P(x = open|z) = 0.667$$

Note: Observation from the environment makes the state of the system more certain

Bayesian Filters: Framework

- ***Given:***
  - Stream of observations z and action data u
  - Sensor model $P(z|x)$
  - Action model $P(x \mid u, x')$
  - Prior probability of the system state $P(x)$
- **Wanted:**
  - Estimation of the state x of a dynamical system
  - The posterior of the state is also called **belief:**

$$bel(x_t) = P(x_t | u_1, z_2, \ldots, u_{t-1}, z_t)$$

References

- M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981.
- Piyush Rai, Probability Refresher Slides (Course: CS698X, 2017-18/II)
- Sebastian Thrun, Wolfram Burgard and Dieter Fox. Probabilistic Robotics. MIT press, 2005.