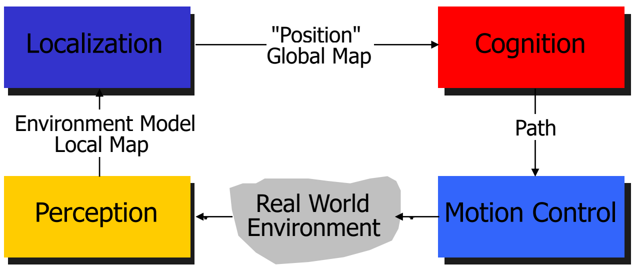
Perception

- Sensors
- Uncertainty
- Features, models of environments



1

BibaBot, BlueBotics SA, Switzerland

IMU
Inertial Measurement Unit

Emergency Stop Button

Wheel Encoders

Omnidirectional Camera

Pan-Tilt Camera

Sonar Sensors

Laser Range Scanner

Bumper

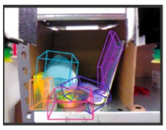
3

Robotic Navigation

- Stanford Stanley Grand Challenge
- Outdoors unstructured env., single vehicle
- Urban Challenge
- Outdoors structured env., mixed traffic, traffic rules

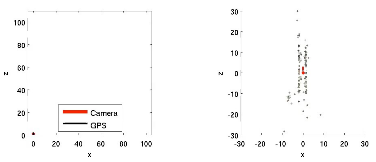
Robotic Manipulation

- Ability to detect objects and their pose
- Challenges – clutter
- Previously unseen objects

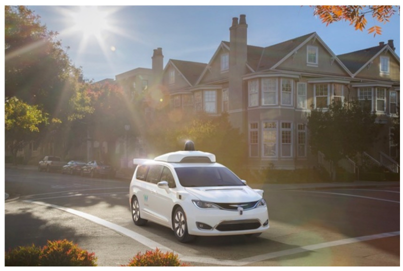
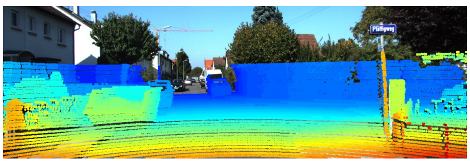



4

Localization



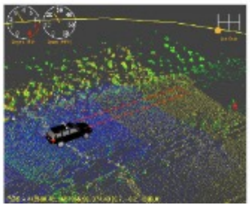
Mapping

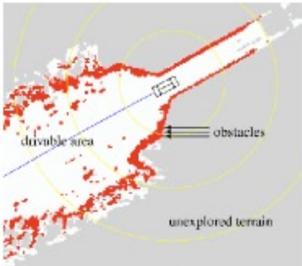

Semantic Understanding

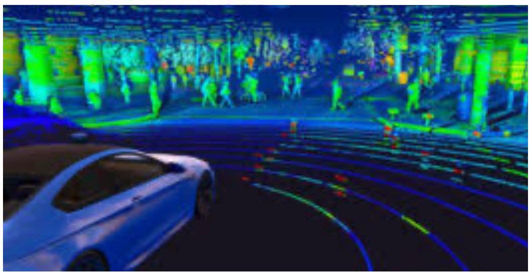
5

- Terrain mapping using lasers

- Determining obstacle course, scene understanding

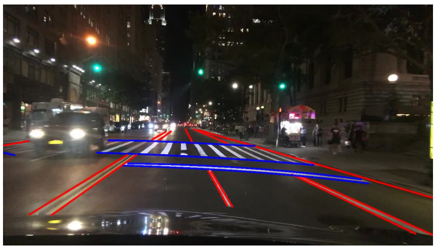




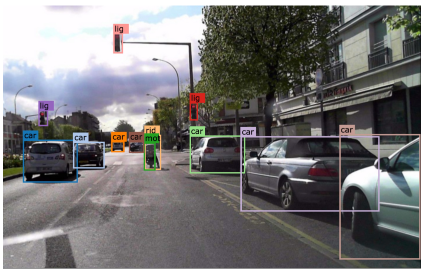
6

Robust Visual, Multi-modal Perception

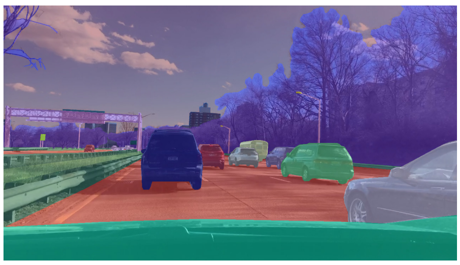
Drivable Areas



Lane Markings



Car detections



Semantic Segmentation

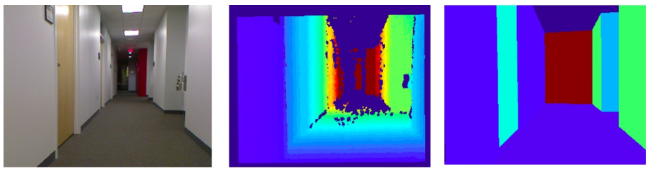
7

source: UC Berkeley Deep Drive

7

Mapping and Localization

- Exploit the single view parsing of indoor environment satisfying Manhattan constraints
- Assign each pixel in the image to one of the floor or walls
- Single view parses can be temporally inconsistent
- Local structure of the scene changes little between consecutive views: the prior wall layout is used as one of the constraints to infer the layout of the current frame.



8

Mapping and Localization

- The basic ingredients: estimation of visual motion and
- 3D structure
- Difficulties: data association, correspondences, invariance
- Choice of features, choice of models, context
- Sparse features, large number of views

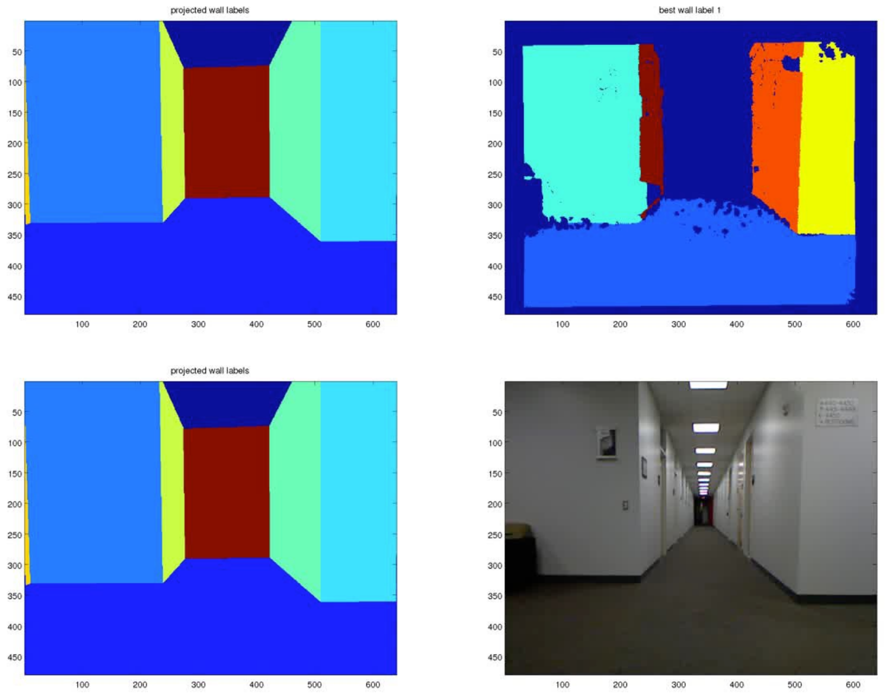
- How can learning alleviate some of the difficulties ?
- How to build representations for more sophisticated tasks ?
- How can the use of semantic information help ?

9

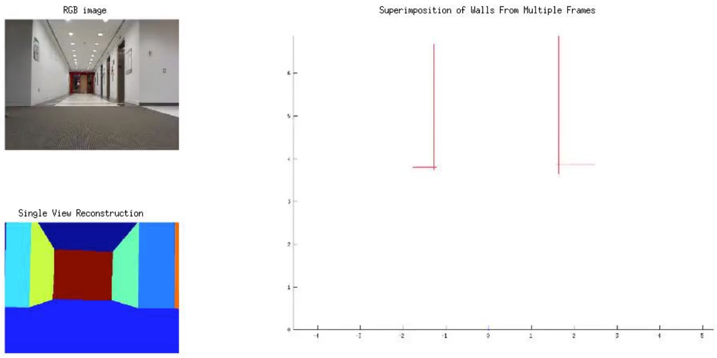
Graph SLAM and Loop Closure

- Manhattan constraints lead to robust and consistent rotation estimates
- Need to global alignment to account to translation drift
- Graph SLAM optimization on (x, y): the camera has fixed height and the estimated rotations are accurate.
- Keyframe based loop closure detection: a frame with visible junction or T-junction is set to be a keyframe. Two keyframes are matched if the GIST score is less than 0.025 and the distance between to keyframe is less than 5m.

11

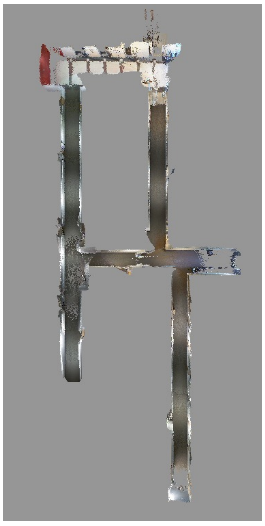


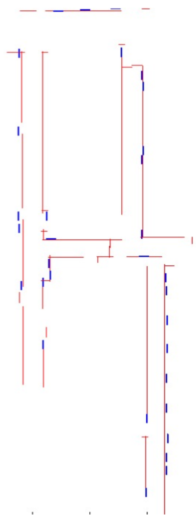
12



13

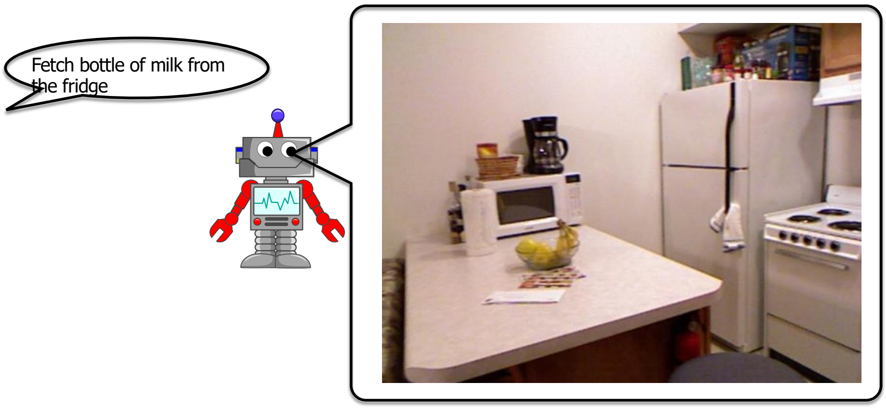
B. GMU sequences





14

Semantic Scene Understandning

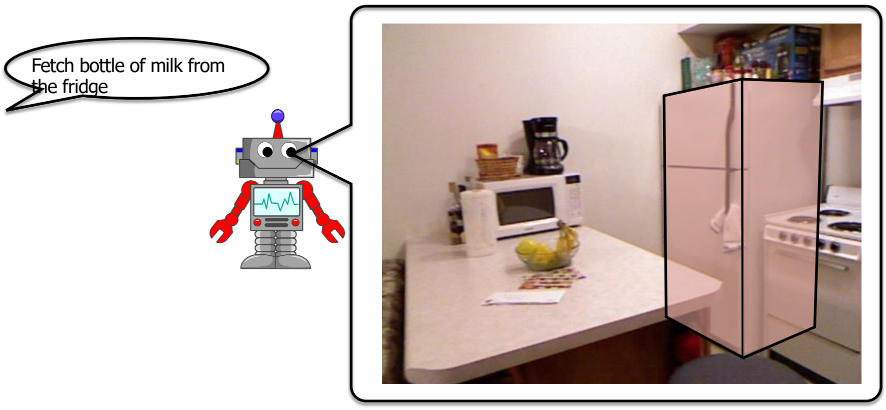


- Need to characterize the entities in the image in term of their semantic and geometric relationship

15

15

Scene Understanding



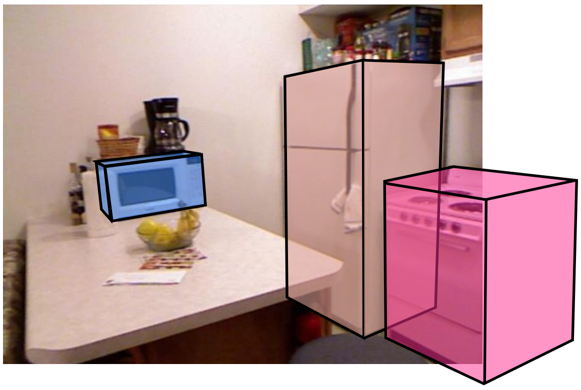
Detecting Fridge and its 3D orientation

16

16

3D Pose Estimation

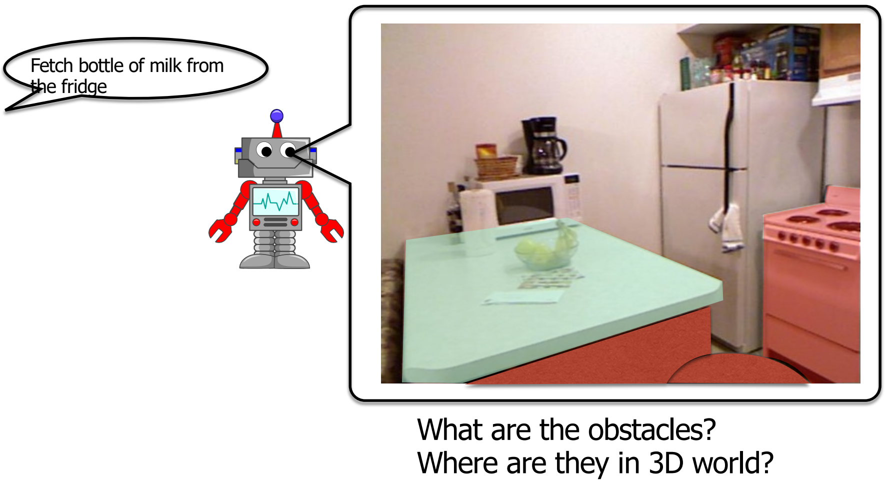
- Definition: Estimating rotation and translation for each object



17

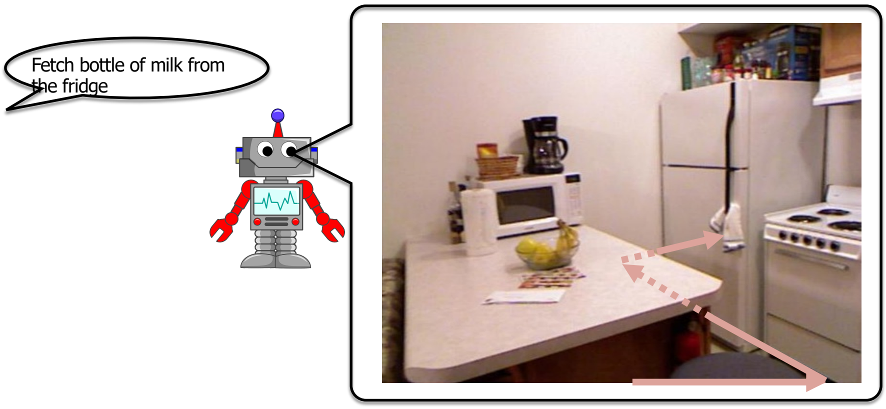
17

Scene Understanding



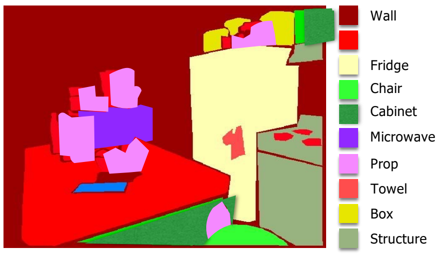
18

Scene Understanding

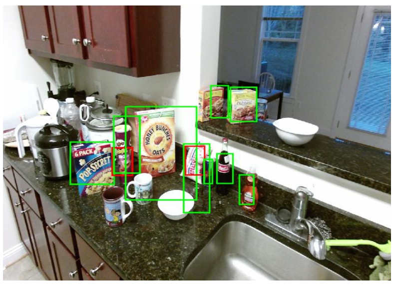


Plan a safe path

Semantic Scene Understanding

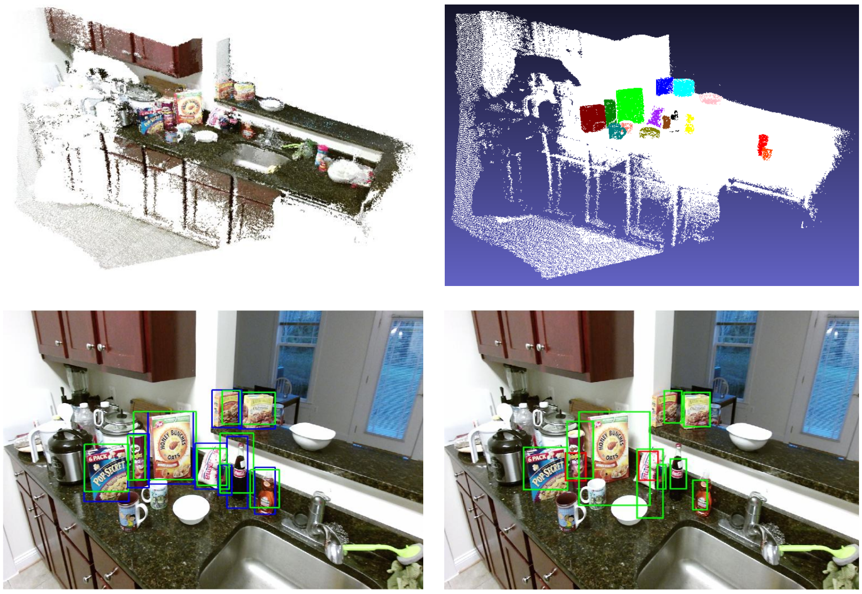


Semantic Labelling



Object Detection, 3D Pose

Mapping and Localization



22

4.1.1

Classification of Sensors

- Proprioceptive sensors
  - measure values internally to the system (robot),
  - e.g. motor speed, wheel load, heading of the robot, battery status
- Exteroceptive sensors
  - information from the robots environment
  - distances to objects, intensity of the ambient light, unique features.
- Passive sensors
  - energy coming for the environment (cameras)
- Active sensors
  - emit their proper energy and measure the reaction
  - better performance, but some influence on environment

25

Role of Perception in Robotics

- Where am I relative to the world?
  - sensors: vision, stereo, range sensors, acoustics
  - problems: scene modeling/classification/recognition
  - integration: localization/mapping algorithms (e.g. SLAM)
  - pose estimation
- What is around me?
  - sensors: vision, stereo, range sensors, acoustics, sounds, smell
  - problems: object recognition, 3D reconstruction, semantic understanding
  - integration: collision avoidance/navigation, learning

Jana Kosecka

26

Role of Perception in Robotics

- How can I safely interact with environment (including people!)?
  - sensors: vision, range, haptics (force+tactile)
  - problems: 3D structure/range estimation, tracking, materials, size, weight,
  - integration: navigation, manipulation, control, learning
- How can I solve "new" problems (generalization)?
  - sensors: vision, range, haptics, undefined new sensor
  - problems: categorization by function/shape/context/??
  - integrate: inference, navigation, manipulation, control, learning

Jana Kosecka

27

Visual Perception Topics

Techniques

- Single view geometry
- Feature detection and matching
- 3D reconstruction

Applications in Robotics:

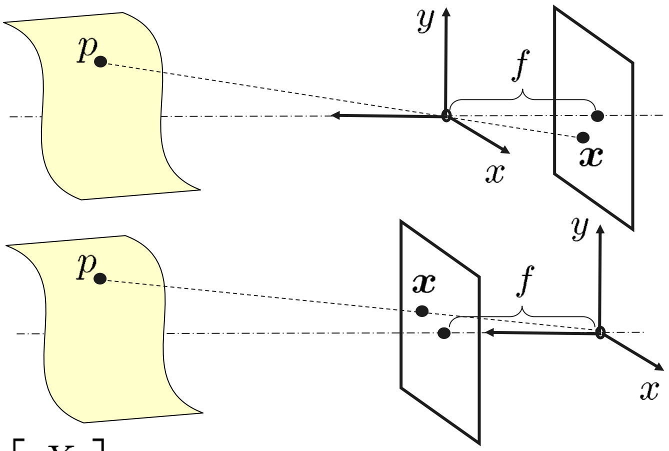
- range sensing, Obstacle detection, environment interaction
- Mapping, registration, localization, recognition
- Manipulation

30

Image Formation

Pinhole

Frontal pinhole



$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}$$

J. Kosecka, GMU

31

Pinhole Camera Model

- Image coordinates are nonlinear function of world coordinates
- Relationship between coordinates in the camera frame and sensor plane

2-D coordinates $$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

Homogeneous coordinates

$$\boldsymbol{x} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}, \quad \boldsymbol{X} \rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

32

Image Coordinates

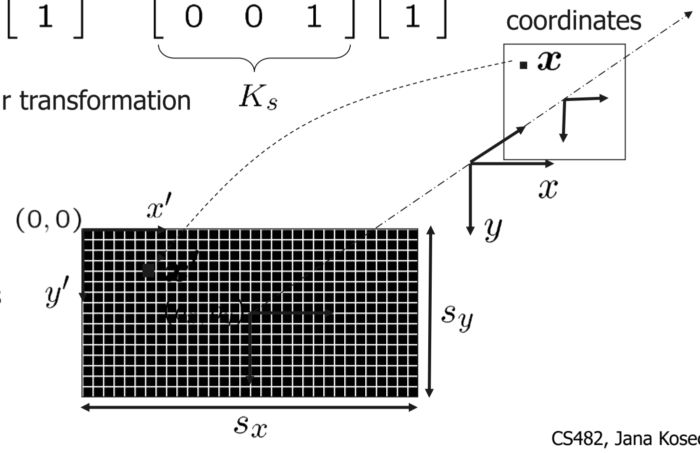
- Relationship between coordinates in the sensor plane and image

$$\boldsymbol{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}}_{K_s} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Linear transformation K_s

metric coordinates

pixel coordinates

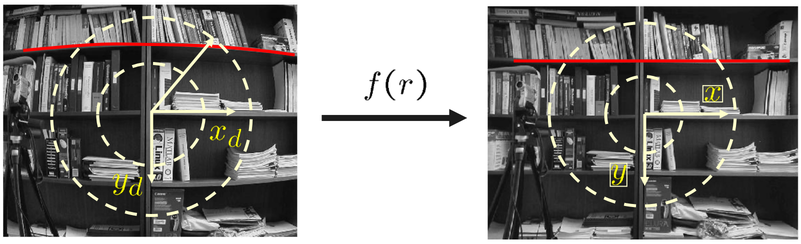


CS482, Jana Kosecka

33

Camera parameters – Radial Distortion

Nonlinear transformation along the radial direction



$$\begin{aligned} \boldsymbol{x} &= c + f(r)(\boldsymbol{x}_d - c), \quad r = \|\boldsymbol{x}_d - c\| \\ f(r) &= 1 + a_1 r + a_2 r^2 + a_3 r^3 + a_4 r^4 + \cdots \end{aligned}$$

Distortion correction: make lines straight

Jana Kosecka, CS 685 34

34

Calibration Matrix and Camera Model

- Relationship between coordinates in the world frame and image
- Intrinsic parameters

Pinhole camera   Pixel coordinates

$$\lambda \boldsymbol{x} = K_f \Pi_0 \boldsymbol{X} \qquad \boldsymbol{x}' = K_s \boldsymbol{x}$$

- Adding transformation between camera coordinate systems and world coordinate system
- Extrinsic Parameters

$$\lambda \boldsymbol{x}' = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\lambda \boldsymbol{x} = K_f \Pi_0 g \boldsymbol{X} = \Pi \boldsymbol{X}$$

35

Image of a Point

Homogeneous coordinates of a 3-D point p

$$\boldsymbol{X} = [X, Y, Z, W]^T \in \mathbb{R}^4, \quad (W = 1)$$

Homogeneous coordinates of its 2-D image

$$\boldsymbol{x} = [x, y, z]^T \in \mathbb{R}^3, \quad (z = 1)$$

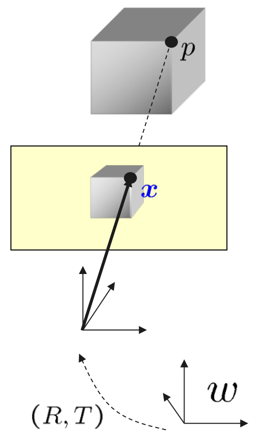
Projection of a 3-D point to an image plane

$$\lambda \boldsymbol{x} = \Pi \boldsymbol{X}$$

$$\lambda \in \mathbb{R}, \ \Pi = [R, T] \in \mathbb{R}^{3\times 4}$$

$$\lambda \boldsymbol{x}' = \Pi \boldsymbol{X}$$

$$\lambda \in \mathbb{R}, \ \Pi = [KR, KT] \in \mathbb{R}^{3\times 4}$$



Jana Kosecka, CS 685

36

36

Image of a Line

Homogeneous representation of a 3-D line L

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X_o \\ Y_o \\ Z_o \\ 1 \end{bmatrix} + \mu \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ 0 \end{bmatrix}, \quad \mu \in \mathbb{R}$$

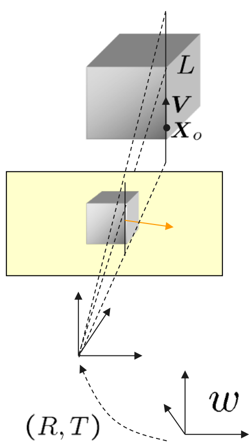
Homogeneous representation of its 2-D image

$$\boldsymbol{l} = [a, b, c]^T \in \mathbb{R}^3$$

Projection of a 3-D line to an image plane

$$\boldsymbol{l}^T \boldsymbol{x} = \boldsymbol{l}^T \Pi \boldsymbol{X} = 0$$

$$\Pi = [KR, KT] \in \mathbb{R}^{3\times 4}$$

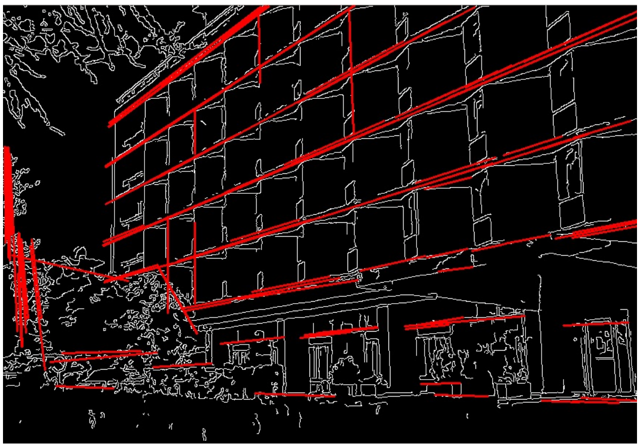


Jana Kosecka, CS 685

37

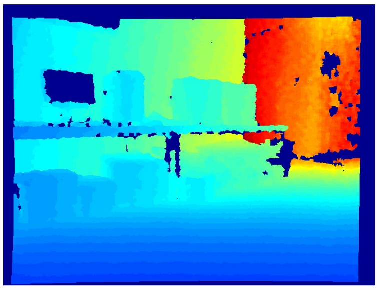
37

Fitting



38

Plane Fitting



Jana Kosecka, CS 685

39

39

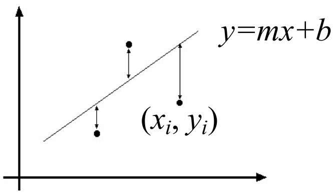
Fitting: Overview

- If we know which points belong to the line, how do we find the "optimal" line parameters?
  - Least squares
- What if there are outliers?
  - Robust fitting, RANSAC
- What if there are many lines?
  - Voting methods: RANSAC, Hough transform
- What if we're not even sure it's a line?
  - Model selection

40

Least squares line fitting

- Data: $(x_1, y_1), \ldots, (x_n, y_n)$
- Line equation: $y_i = m x_i + b$
- Find (m, b) to minimize



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

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad B = \begin{bmatrix} m \\ b \end{bmatrix}$$

$$E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)$$

$$\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0$$

$$X^T XB = X^T Y$$

Normal equations: least squares solution to $XB=Y$

41

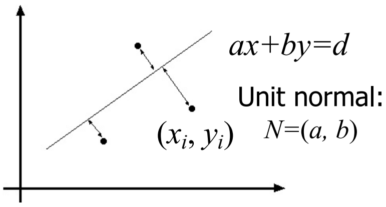
Problem with "vertical" least squares

- Not rotation-invariant
- Fails completely for vertical lines

42

Total least squares

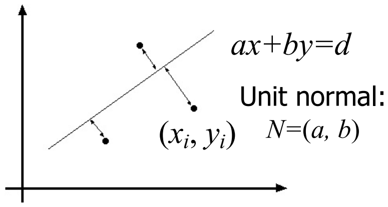
- Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$: $|ax_i + by_i - d|$



43

Total least squares

- Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$: $|ax_i + by_i - d|$
- Find (a, b, d) to minimize the sum of squared perpendicular distances

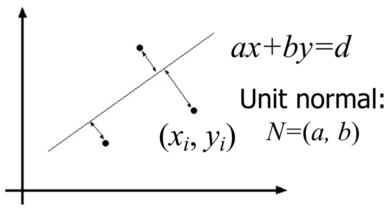


$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

44

Total least squares

- Distance between point (x_i, y_i) and line $ax+by=d$ $(a^2+b^2=1)$: $|ax_i + by_i - d|$
- Find (a, b, d) to minimize the sum of squared perpendicular distances



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

$$\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0 \qquad d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} y_i = a\bar{x} + b\bar{y}$$

$$E = \sum_{i=1}^{n} (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)$$

$$\frac{dE}{dN} = 2(U^T U)N = 0$$

Solution to $(U^T U)N = 0$, subject to $\|N\|^2 = 1$: eigenvector of $U^T U$ associated with the smallest eigenvalue (least squares solution to *homogeneous linear system* $UN = 0$)

45

Total least squares

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{bmatrix}$$

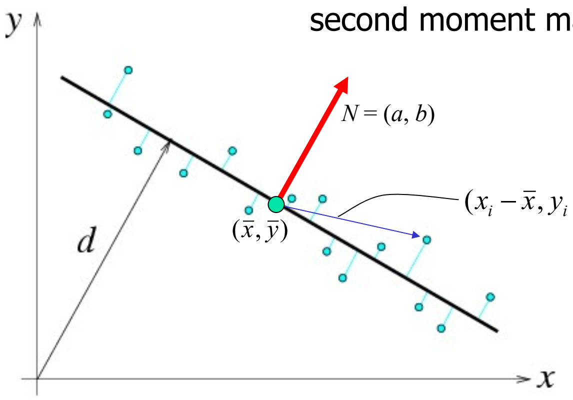
second moment matrix

46

Total least squares

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2 \end{bmatrix}$$

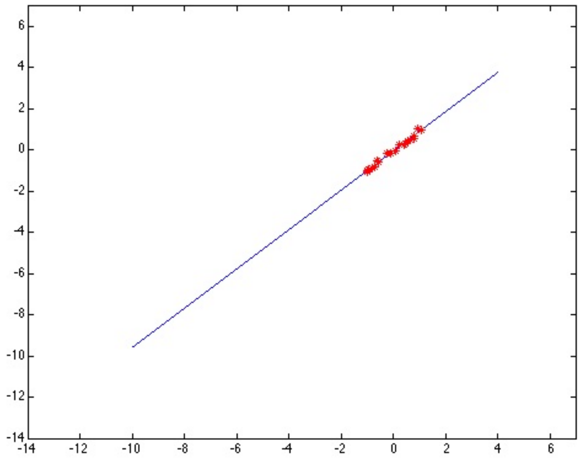
second moment matrix



47

Least squares: Robustness to noise

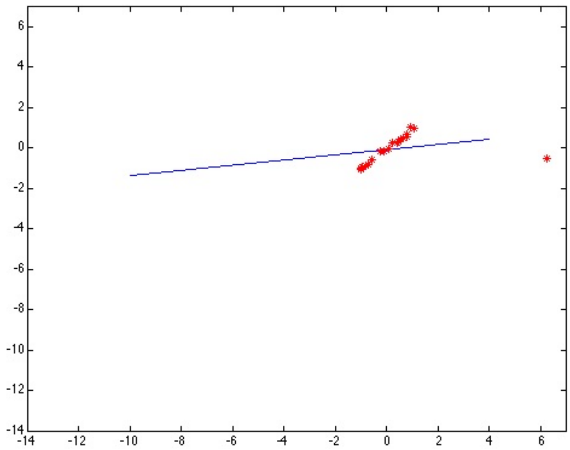
- Least squares fit to the red points:



50

Least squares: Robustness to noise

- Least squares fit with an outlier:



Problem: squared error heavily penalizes outliers

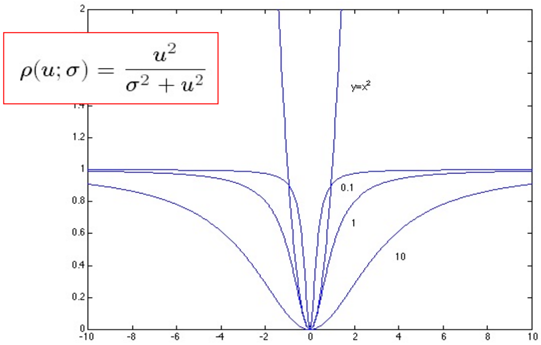
51

Robust estimators

- General approach: find model parameters θ that minimize

$$\sum_i \rho\left(r_i\left(x_i,\theta\right);\sigma\right)$$

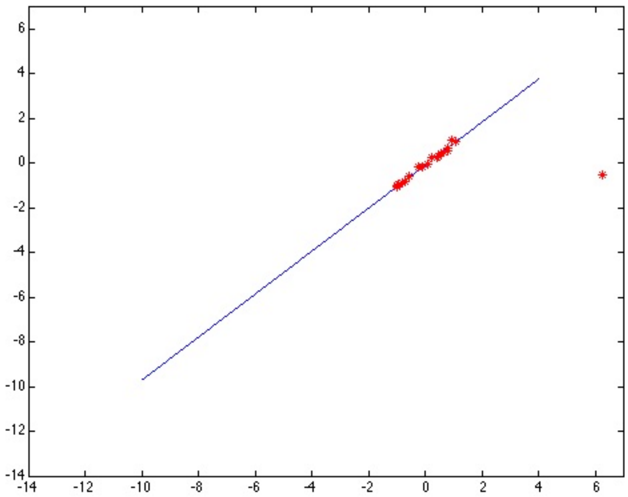
$r_i(x_i, \theta)$ – residual of ith point w.r.t. model parameters θ
ρ – robust function with scale parameter σ



The robust function ρ behaves like squared distance for small values of the residual u but saturates for larger values of u

52

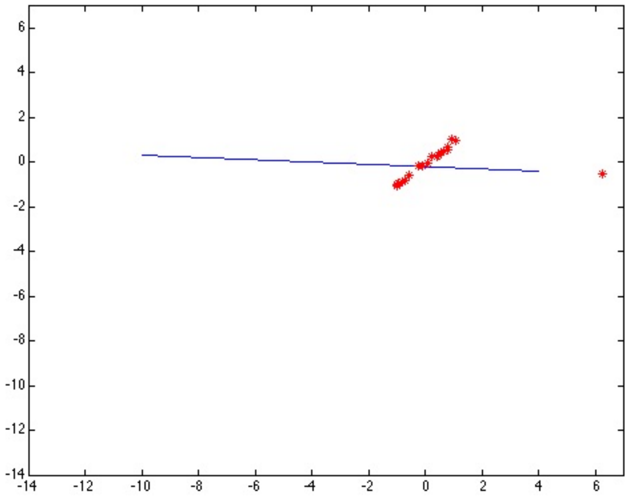
Choosing the scale: Just right



The effect of the outlier is minimized

53

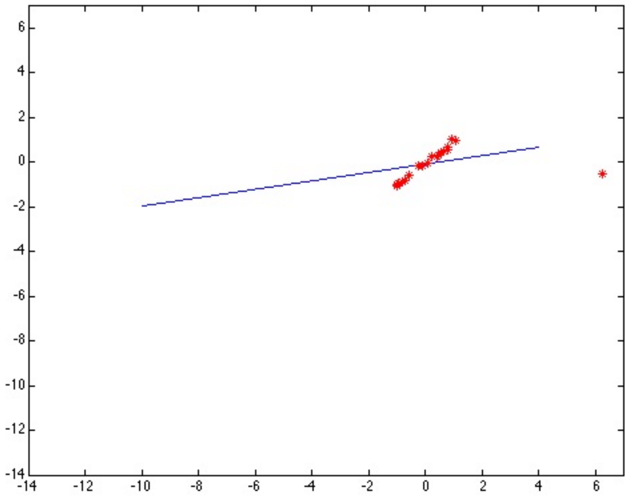
Choosing the scale: Too small



The error value is almost the same for every point and the fit is very poor

54

Choosing the scale: Too large



Behaves much the same as least squares

55

Robust estimation: Details

- Robust fitting is a nonlinear optimization problem that must be solved iteratively
- Least squares solution can be used for initialization
- Adaptive choice of scale: approx. 1.5 times median residual (F&P, Sec. 15.5.1)

56

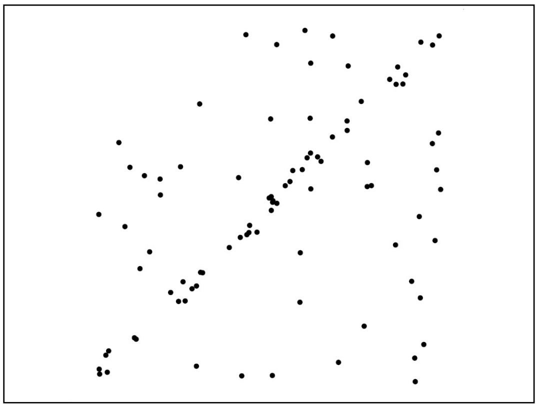
RANSAC

- Robust fitting can deal with a few outliers – what if we have very many?
- Random sample consensus (RANSAC): Very general framework for model fitting in the presence of outliers
- Outline
  - Choose a small subset of points uniformly at random
  - Fit a model to that subset
  - Find all remaining points that are "close" to the model and reject the rest as outliers
  - Do this many times and choose the best model

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

57

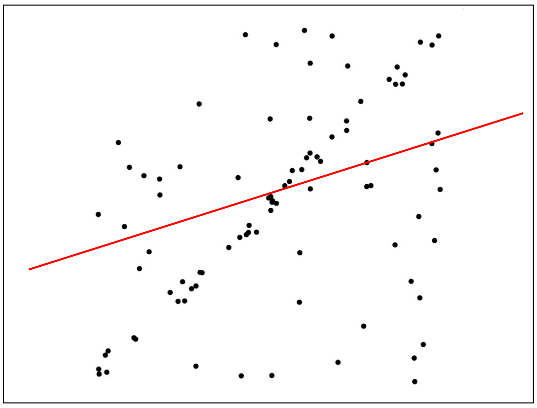
RANSAC for line fitting example



Source: R. Raguram

58

RANSAC for line fitting example

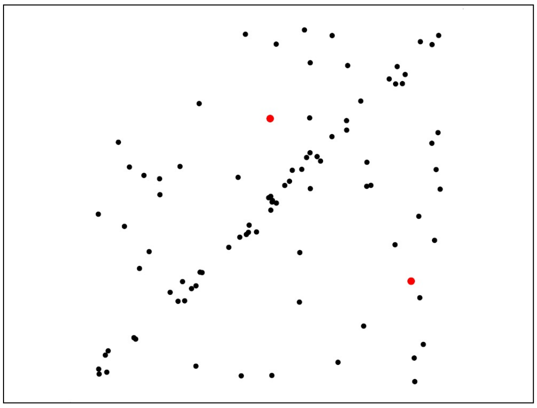


Least-squares fit

Source: R. Raguram

59

RANSAC for line fitting example

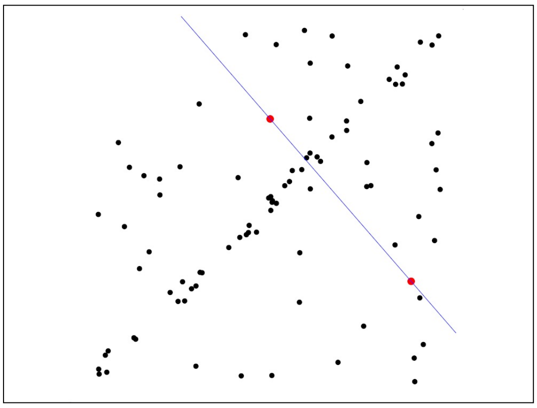


1. Randomly select minimal subset of points

Source: R. Raguram

60

RANSAC for line fitting example

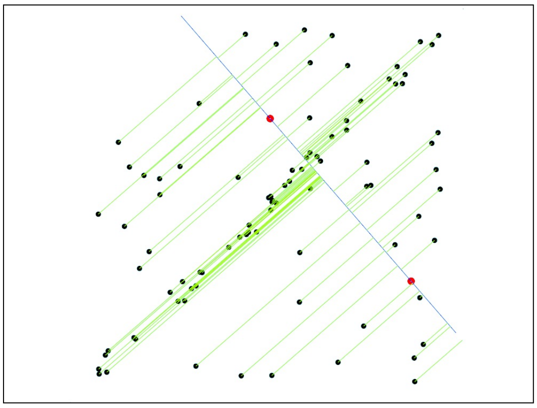


1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram

61

RANSAC for line fitting example

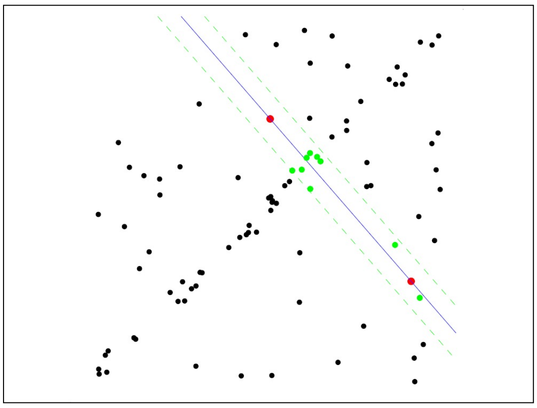


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

62

RANSAC for line fitting example

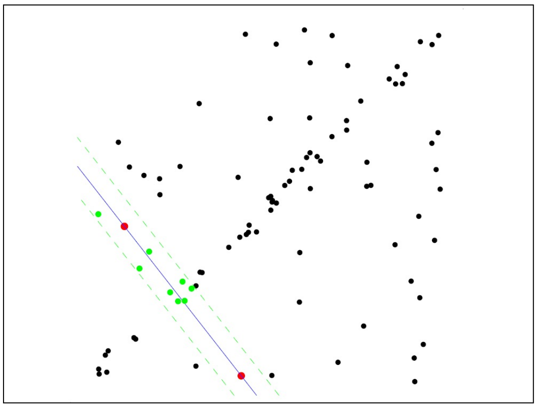


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

Source: R. Raguram

63

RANSAC for line fitting example

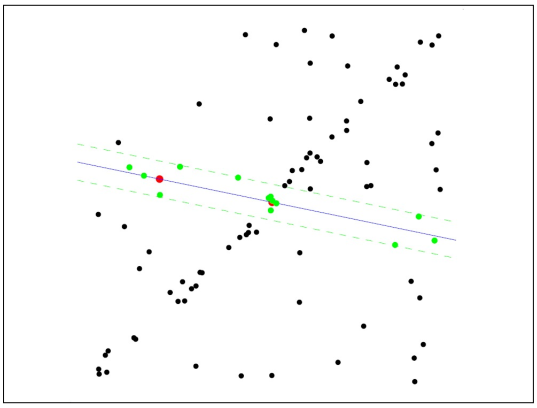


1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

64

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

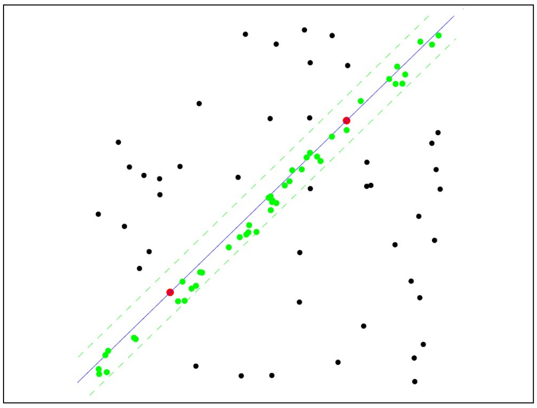
65

Source: R. Raguram

65

RANSAC for line fitting example

Uncontaminated sample



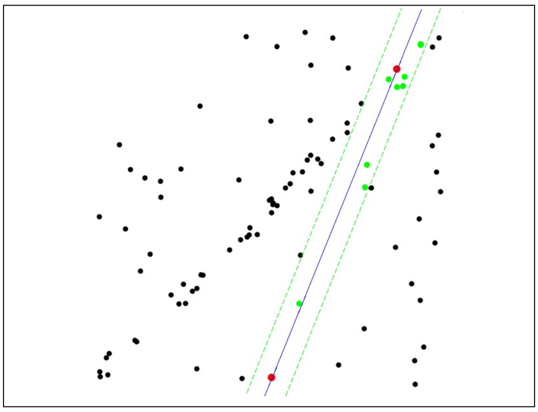
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

66

66

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

67

RANSAC for line fitting

- Repeat ***N*** times:
- Draw ***s*** points uniformly at random
- Fit line to these ***s*** points
- Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than ***t***)
- If there are ***d*** or more inliers, accept the line and refit using all inliers

68

Choosing the parameters

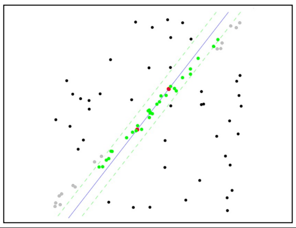
- Initial number of points *s*
  - Typically minimum number needed to fit the model
- Distance threshold *t*
  - Choose *t* so probability for inlier is *p* (e.g. 0.95)
  - Zero-mean Gaussian noise with std. dev. σ: $t^2=3.84\sigma^2$
- Number of samples *N*
  - Choose *N* so that, with probability *p*, at least one random sample is free from outliers (e.g. *p*=0.99) (outlier ratio: *e*)

Source: M. Pollefeys

69

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - Doesn't work well for low inlier ratios (too many iterations, or can fail completely)
  - Can't always get a good initialization of the model based on the minimum number of samples



74

Previously

- Camera model
- Stereo matching, triangulation

- Features
- Least Squares Fitting
- RANSAC

- Today
- Camera pose estimation in the world coordinate frame
- Relative Pose Estimation

75

Calibration with a Rig

Use the fact that both 3-D and 2-D coordinates of feature points on a pre-fabricated object (e.g., a cube) are known.

Calibration with a Rig

- Given 3-D coordinates on known object

$\lambda \mathbf{x}' = [KR, KT]\mathbf{X}$ → $\lambda \mathbf{x}' = \Pi \mathbf{X}$

$$\lambda \begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix} = \begin{bmatrix} \pi_1^T \\ \pi_2^T \\ \pi_3^T \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{bmatrix}$$

- Eliminate unknown scales

$$x^i(\pi_3^T \mathbf{X}) = \pi_1^T \mathbf{X},$$
$$y^i(\pi_3^T \mathbf{X}) = \pi_2^T \mathbf{X}$$

- Recover projection matrix $\Pi = [KR, KT] = [R', T']$

$$\min \|M\Pi^s\|^2 \quad \text{subject to} \quad \|\Pi^s\|^2 = 1$$

$$\Pi^s = [\pi_{11}, \pi_{21}, \pi_{31}, \pi_{12}, \pi_{22}, \pi_{32}, \pi_{13}, \pi_{23}, \pi_{33}, \pi_{14}, \pi_{24}, \pi_{34}]^T$$

- Factor the into $R \in SO(3)$ and using QR decomposition
- Solve for translation $T = K^{-1}T'$

More details

- Direct calibration by recovering and decomposing the projection matrix

$$\lambda \begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix} = \begin{bmatrix} \pi_1^T \\ \pi_2^T \\ \pi_3^T \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{bmatrix} \rightarrow Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} & \pi_{14} \\ \pi_{21} & \pi_{22} & \pi_{23} & \pi_{24} \\ \pi_{31} & \pi_{32} & \pi_{33} & \pi_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x_i = \frac{\pi_{11}X_i + \pi_{12}Y_i + \pi_{13}Z_i + \pi_{14}}{\pi_{31}X_i + \pi_{32}Y_i + \pi_{33}Z_i + \pi_{34}} \quad y_i = \frac{\pi_{21}X_i + \pi_{22}Y_i + \pi_{23}Z_i + \pi_{24}}{\pi_{31}X_i + \pi_{32}Y_i + \pi_{33}Z_i + \pi_{34}}$$

$$x_i(\pi_{31}X_i + \pi_{32}Y_i + \pi_{33}Z_i + \pi_{34}) = \pi_{11}X_i + \pi_{12}Y_i + \pi_{13}Z_i + \pi_{14}$$
$$y_i(\pi_{31}X_i + \pi_{32}Y_i + \pi_{33}Z_i + \pi_{34}) = \pi_{21}X_i + \pi_{22}Y_i + \pi_{23}Z_i + \pi_{24}$$

$$x^i(\pi_3^T \mathbf{X}) = \pi_1^T \mathbf{X}, \quad y^i(\pi_3^T \mathbf{X}) = \pi_2^T \mathbf{X}$$

2 constraints per point

$$[X_i, Y_i, Z_i, 1, 0, 0, 0, 0, -x_iX_i, -x_iY_i, -x_iZ_i, -x_i]\Pi_s = 0$$
$$[0, 0, 0, 0, X_i, Y_i, Z_i, 1, -y_iX_i, -y_iY_i, -y_iZ_i, -y_i]\Pi_s = 0$$

$$\Pi_s = [\pi_{11}, \pi_{12}, \pi_{13}, \pi_{14}, \pi_{21}, \pi_{22}, \pi_{23}, \pi_{24}, \pi_{31}, \pi_{32}, \pi_{33}, \pi_{34}]^T$$

78

78

More details

- Recover projection matrix $\Pi = [KR, KT] = [R', T']$

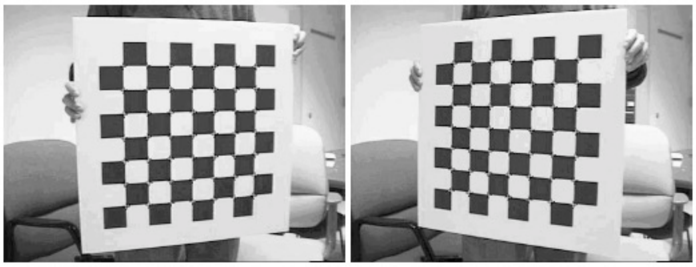
  $\min \|M\Pi^s\|^2$ subject to $\|\Pi^s\|^2 = 1$

  $\Pi^s = [\pi_{11}, \pi_{21}, \pi_{31}, \pi_{12}, \pi_{22}, \pi_{32}, \pi_{13}, \pi_{23}, \pi_{33}, \pi_{14}, \pi_{24}, \pi_{34}]^T$
- Collect the constraints from all N points into matrix M (2N x 12)
- Solution eigenvector associated with the smallest eigenvalue $M^T M$
- Unstack the solution and decompose into rotation and translation
- Factor the R' into $R \in SO(3)$ and K using QR decomposition
- Solve for translation $T = K^{-1}T'$

79

79

Calibration with a planar pattern



$$H \doteq K[r_1, r_2, T] \quad \in \mathbb{R}^{3\times 3} \qquad \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K[r_1, r_2, T] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},$$

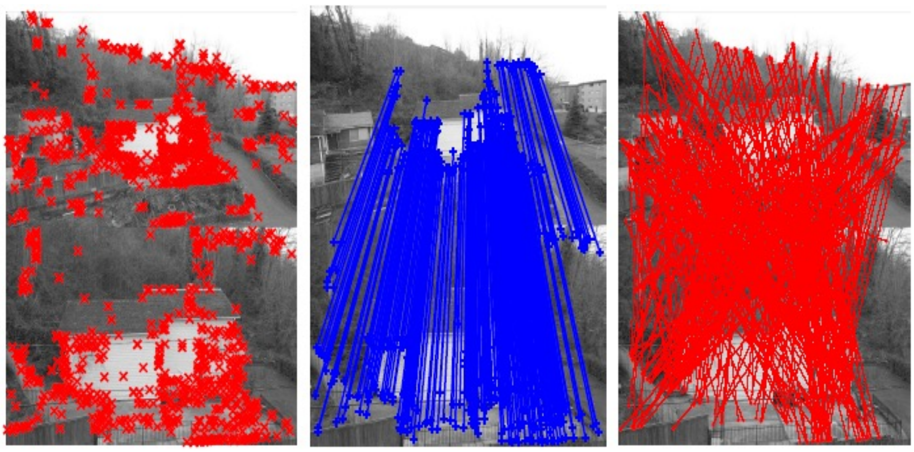
To eliminate unknown depth, multiply both sides by $\widehat{x'}$

$$\widehat{x'} H [X, Y, 1]^T = 0.$$

80

80

Robust technique



(a) correspondences. (b) identified inliers. (c) identified outliers.

88

88