ECE 484: Principles of Safe Autonomy (Fall 2025)
Lecture 7
Perception: Visual Odometry

Professor: Huan Zhang
https://publish.illinois.edu/safe-autonomy/
https://huan-zhang.com
huanz@illinois.edu

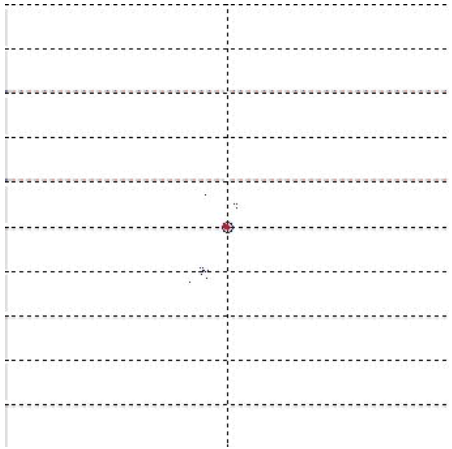
Problem: Visual Odometry (VO)

Incrementally estimate pose of the vehicle from onboard camera images

input

Image sequence (or video stream)
from one or more cameras attached to a moving vehicle

output



$R_0, R_1, R_2 \ldots$
t_0, t_1, t_2

Why VO ?

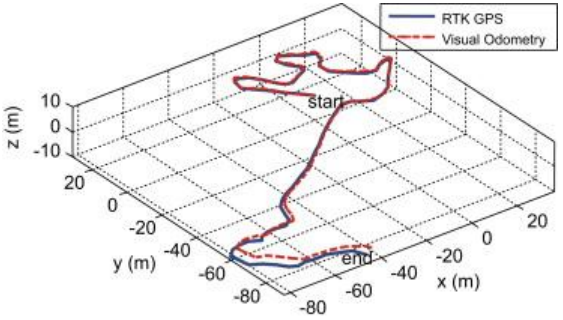
Unlike wheel odometry, VO is not affected by wheel slip in uneven terrain or other adverse conditions.

More accurate trajectory estimates compared to wheel odometry (relative position error 0.1% – 2%)

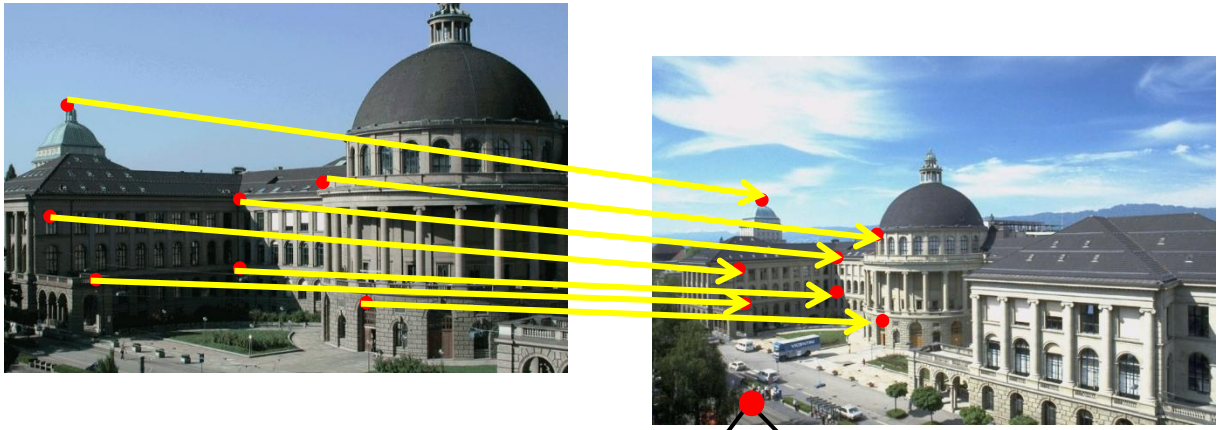
VO can be used as a complement to

- wheel odometry, GPS, IMUs, laser odometry

In GPS-denied environments, such as underwater and aerial

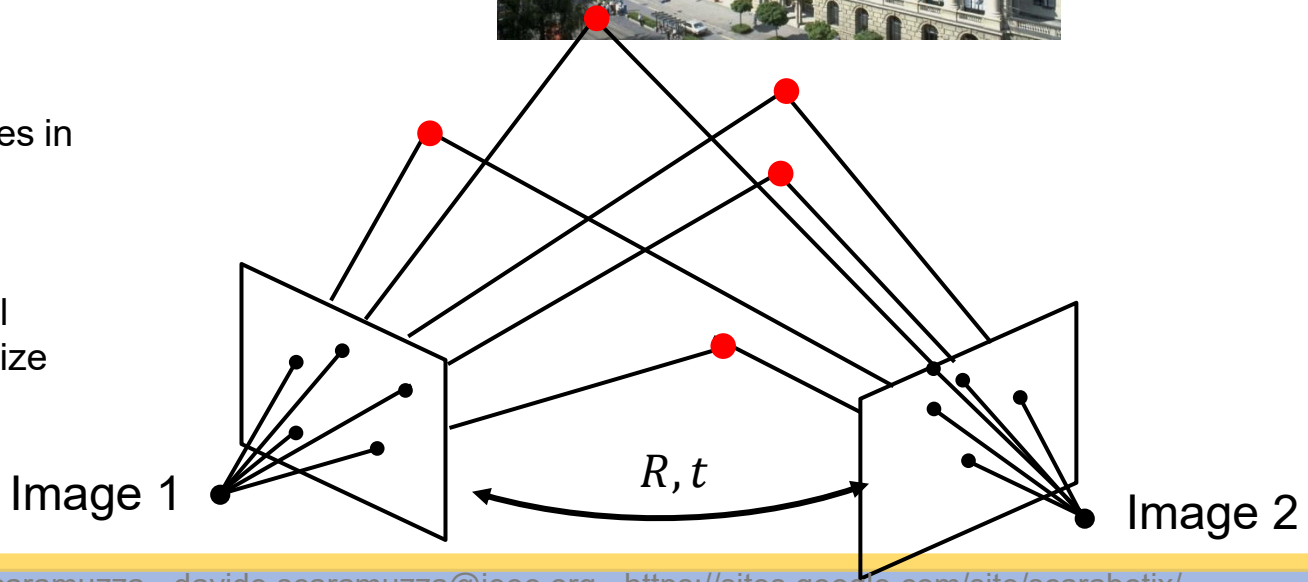


VO Principle



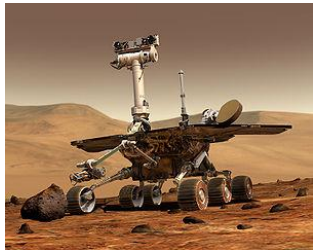
VO Solution Outline

1. Detect and match features in two successive images
2. Estimate motion R_k, t_k (Epipolar geometry)
3. RANSAC outlier removal
4. Repeat Steps 1-3, optimize



Brief history of VO

- 1996: The term VO was coined by Srinivasan to define motion orientation in honey bees.
- 1980: First known stereo VO real-time implementation on a robot by Moraveck PhD thesis (NASA/JPL) for Mars rovers using a sliding camera. Moravec invented a predecessor of Harris detector, known as Moravec detector
- 1980 to 2000: The VO research was dominated by NASA/JPL in preparation of 2004 Mars mission (see papers from Matthies, Olson, etc. From JPL)
- 2004: VO used on a robot on another planet: Mars rovers Spirit and Opportunity
- 2004. VO was revived in the academic environment by Nister «Visual Odometry» paper.
  The term VO became popular.



Recall what a calibrated camera gives us

Camera to pixel

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix} = \begin{bmatrix}f_x & 0 & o_x & 0\\ 0 & f_y & o_y & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x_c\\ y_c\\ z_c\\ 1\end{bmatrix}$$

World to camera

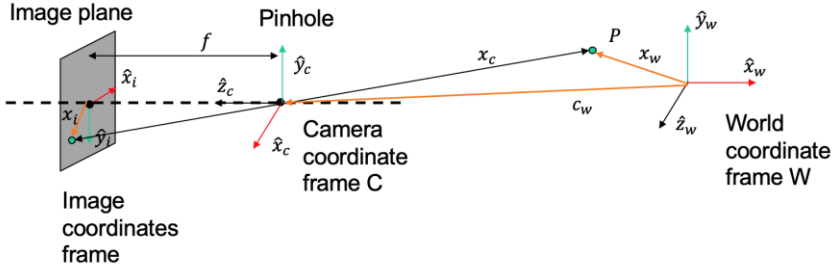
$$\begin{bmatrix}x_c\\ y_c\\ z_c\\ 1\end{bmatrix} = \begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\ r_{21} & r_{22} & r_{23} & t_y\\ r_{31} & r_{32} & r_{33} & t_z\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_w\\ y_w\\ z_w\\ 1\end{bmatrix}$$

$$\tilde{u} = M_{int}\ \tilde{x}_w$$

$$\tilde{x}_c = M_{ext}\ \tilde{x}_w$$

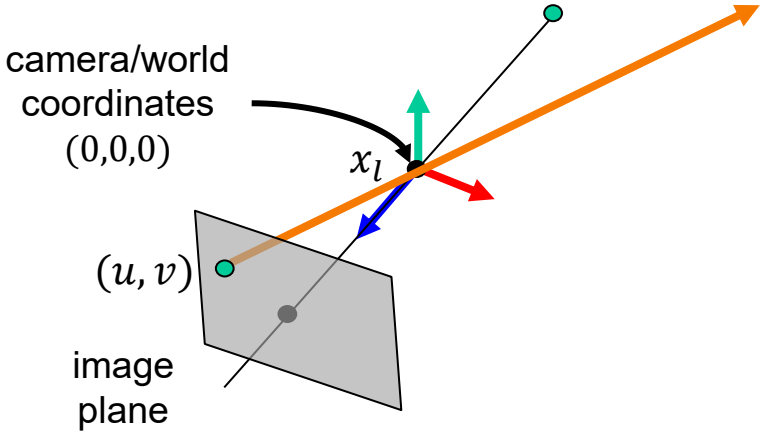
$$\tilde{u} = M_{int}\, M_{ext}\, \tilde{x}_w = P\, \tilde{x}_w$$

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{bmatrix} = \begin{bmatrix}p_{11} & p_{12} & p_{13} & p_{14}\\ p_{21} & p_{22} & p_{23} & p_{24}\\ p_{31} & p_{32} & p_{33} & p_{34}\end{bmatrix}\begin{bmatrix}x_w\\ y_w\\ z_w\\ 1\end{bmatrix}$$



P: **Projection matrix**

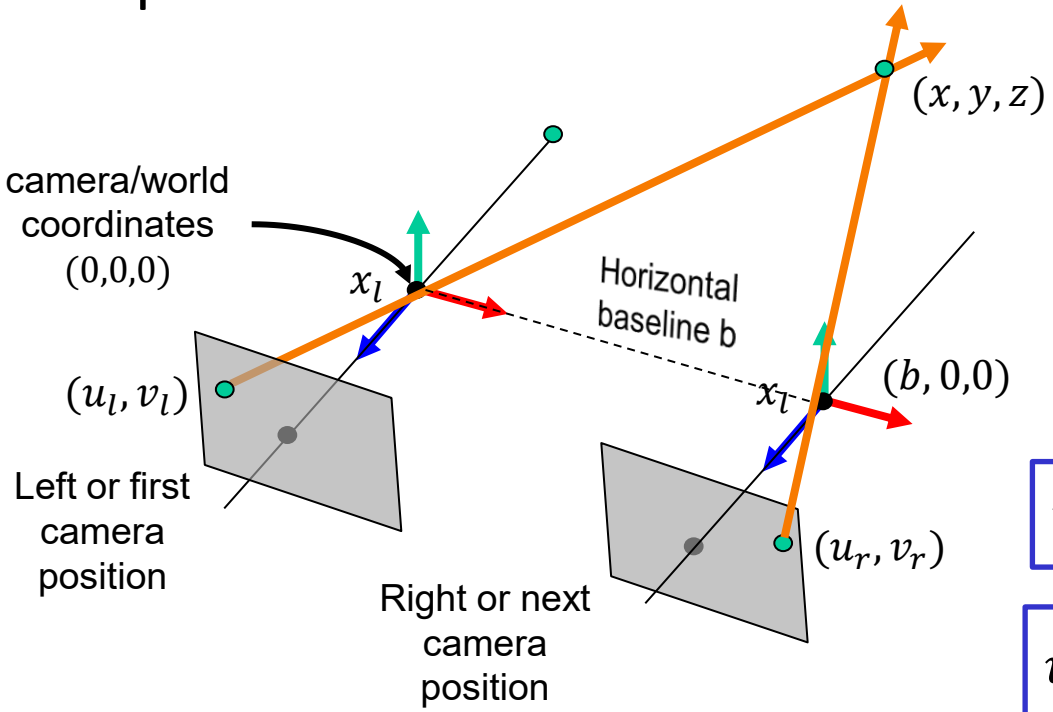
Backward projection from 2D to 3D



3D to 2D $u = f_x \frac{x_c}{z_c} + o_x$ and $v = f_y \frac{y_c}{z_c} + o_y$

2D to 3D $x = \frac{z}{f_x}(u - o_x)$, $y = \frac{z}{f_y}(v - o_y)$; $z > 0$

Simple Stereo Problem: Reconstructing scene from two images

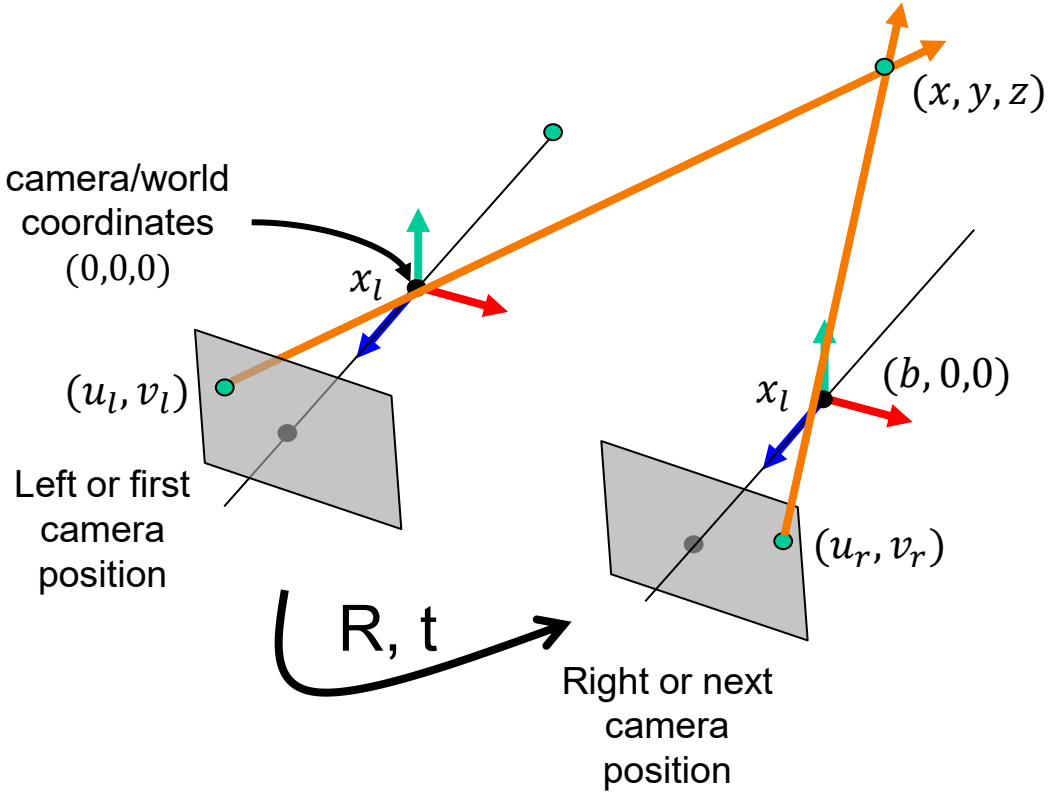


$$u_l = f_x \frac{x}{z} + o_x \text{ and } v_l = f_y \frac{y}{z} + o_y$$

$$u_r = f_x \frac{x-\boldsymbol{b}}{z} + o_x \text{ and } v = f_y \frac{y}{z} + o_y$$

From these 4 equations we can find (x, y, z): $z = \frac{b f_x}{(u_l - u_r)}$

Uncalibrated Stereo Problem

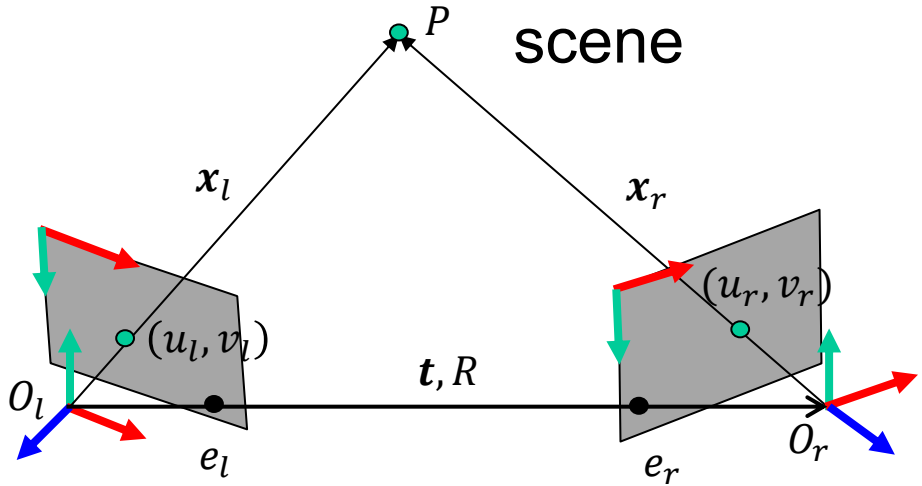


The two cameras' intrinsic parameters are known but the **rotation (R)** and the **translation (t)** between the two camera poses is unknown

This relation between the two poses is described by **epipolar geometry**

Relation captured by Fundamental matrix

Epipolar Geometry: Epipoles



$$x_l = \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} \quad x_r = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}$$

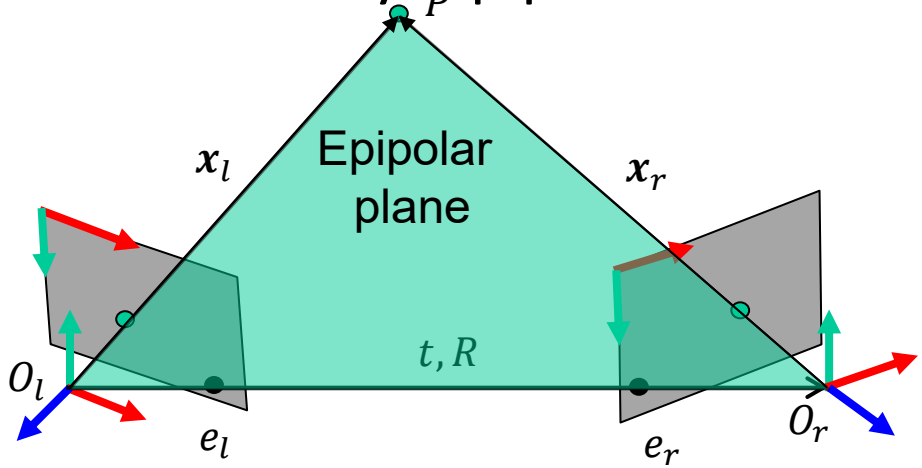
$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

The intrinsic parameters (f_x, f_y, o_x, o_y) are given for each camera

Problem: Compute extrinsic parameters t, R relating the two camera positions, given only coordinates on the image planes $(u_l, v_l), (u_r, v_r)$

Epipole: Image point of origin/pinhole of one camera position as viewed by the other camera position: e_l and e_r

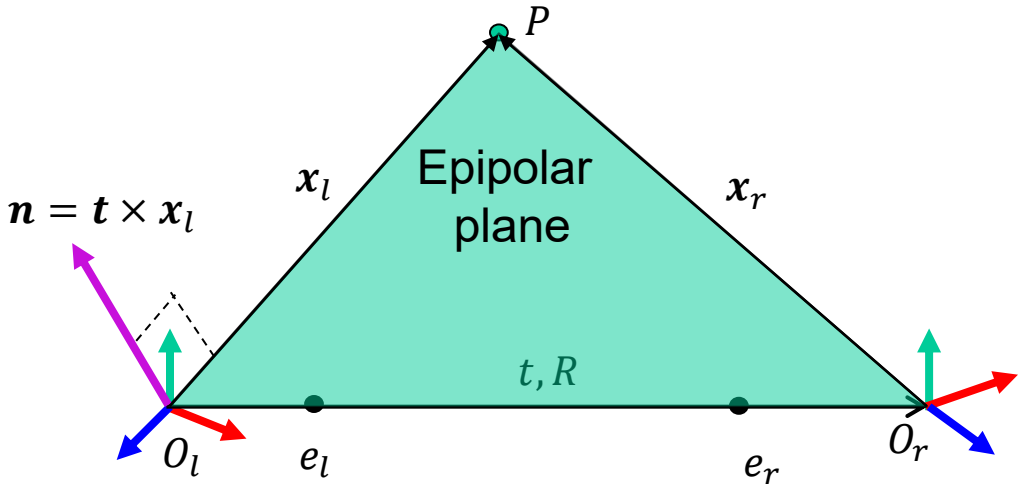
Epipolar Geometry: Epipolar Plane



Epipolar plane of scene point P: The plane formed by camera origins (O_l and O_r), epipoles (e_l and e_r), and P.

Each scene point P has a unique **epipolar plane**

Epipolar Constraints



Recall $a \times b = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{bmatrix}$

Vector normal to the epipolar plane: $\boldsymbol{n = t \times x_l}$

Dot product of x_l and n (perpendicular vectors) is zero $\boldsymbol{x_l . n = x_l . (t \times x_l) = 0}$ **"Epipolar constraint"**

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} t_y z_l - t_z y_l \\ t_z x_l - t_x z_l \\ t_x y_l - t_y x_l \end{bmatrix} = 0 \text{ rewritten as } [x_l \quad y_l \quad z_l] \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

T_X

Epipolar Constraints

Dot product of $\boldsymbol{x}_l$ and n (perpendicular vectors) is zero $\boldsymbol{x}_l . n = \boldsymbol{x}_l . (t \times \boldsymbol{x}_l) = 0$

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} t_y z_l - t_z y_l \\ t_z x_l - t_x z_l \\ t_x y_l - t_y x_l \end{bmatrix} = 0 \qquad [x_l \quad y_l \quad z_l] \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = 0$$

Relating 3D coordinates of P in the left camera with that of the right $\boldsymbol{x}_l = R\boldsymbol{x}_r + \boldsymbol{t}$

Substituting $\begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ in the second $\boldsymbol{x}_l$

$$[x_l \quad y_l \quad z_l] \left(\begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} + \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \right) = 0$$

$$[x_l \quad y_l \quad z_l] \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

Essential matrix

$$E = T_X R = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

E is a product of a skew symmetric matrix T_X and an orthonormal matrix R, and therefore, from E we can get t R by **Singular value decompositions (SVD)** (read: https://www.robots.ox.ac.uk/~vgg/hzbook/hzbook1/HZepipolar.pdf)

How to find the Essential Matrix?

$$\boldsymbol{x}_l^T E \boldsymbol{x}_r = 0$$

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0$$

Unfortunately, we do not have the 3D coordinates of the same scene point x_l and x_r

But we know the corresponding points in image coordinates, (u_l, v_l) **and** (u_r, v_r)

Incorporating image coordinates

$$u_l = f_x^{(l)} \frac{x_l}{z_l} + o_x^{(l)} \qquad v_l = f_y^{(l)} \frac{y_l}{z_l} + o_y^{(l)}$$

$$z_l u_l = f_x^{(l)} x_l + z_l o_x^{(l)} \qquad z_l v_l = f_y^{(l)} y_l + z_l o_y^{(l)}$$

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & 0 & o_x^{(l)} \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

Known intrinsic matrix K_l

Incorporating image coordinates in Epipolar constraints

Left camera

$$z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(l)} & 0 & o_x^{(l)} \\ 0 & f_y^{(l)} & o_y^{(l)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}$$

$$\boldsymbol{x}_l^T = [u_l \quad v_l \quad 1] z_l {K_l^{-1}}^T$$

Right camera

$$z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_x^{(r)} & 0 & o_x^{(r)} \\ 0 & f_y^{(r)} & o_y^{(r)} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}$$

$$\boldsymbol{x}_r = K_r^{-1} z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix}$$

$$[u_l \quad v_l \quad 1] z_l {K_l^{-1}}^T \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

Assume z_l and z_r are not zero, the rest of the products must be 0

$$[u_l \quad v_l \quad 1] {K_l^{-1}}^T \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} K_r^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

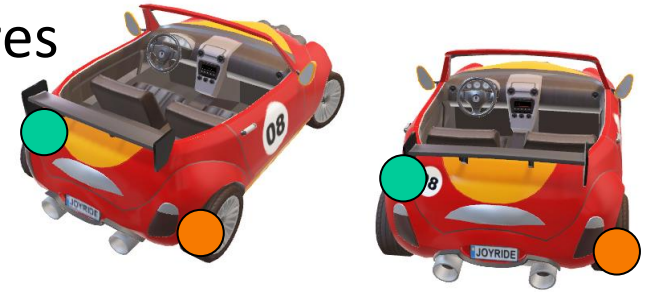
$$E = K_l^T F K_r$$

$$[u_l \quad v_l \quad 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

$F = {K_l^{-1}}^T E K_r^{-1}$ Fundamental matrix

Finding Fundamental Matrix: Matching features

Find a set of corresponding features in the two images (e.g. using SIFT)



- $(u_l^{(1)}, v_l^{(1)})$ $(u_r^{(1)}, v_r^{(1)})$
- …
- $(u_l^{(m)}, v_l^{(m)})$ $(u_r^{(m)}, v_r^{(m)})$

Plugging into Epipolar Constraints

$$\begin{bmatrix} u_l^{(i)} & v_l^{(i)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r^{(i)} \\ v_r^{(i)} \\ 1 \end{bmatrix} = 0$$

known unknown known

One linear equation for each matched feature i

Stacking equations for all the features and the elements of F as a vector $\boldsymbol{f}$ we get

$$A\,\boldsymbol{f} = \mathbf{0}$$

Missing scale

$$[u_l \quad v_l \quad 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0 = [u_l \quad v_l \quad 1]\begin{bmatrix} kf_{11} & kf_{12} & kf_{13} \\ kf_{21} & kf_{22} & kf_{23} \\ kf_{31} & kf_{32} & kf_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

The fundamental matrix works on homogeneous coordinates

Fundamental matrix F and $\mathrm{k}F$ describe the same epipolar geometry

F is defined only up to a scale factor

Set the Fundamental Matrix to some arbitrary scale $\left|\left|\mathrm{f}\right|\right| = 1$

Solving for F up to Scale

$$\begin{bmatrix} u_l^{(i)} & v_l^{(i)} & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_r^{(i)} \\ v_r^{(i)} \\ 1 \end{bmatrix} = 0$$

known unknown known

One linear equation for each matched feature i

Stacking equations for all the features and the elements of F as a vector $\boldsymbol{f}$ we get

$$A\,\boldsymbol{f} = \mathbf{0}$$

We want $A\boldsymbol{f}$ as close to 0 as possible and $\left|\left|\boldsymbol{f}\right|\right|^2 = 1$:

Constrained linear least squares problem $\min_{\boldsymbol{f}} \left|\left|A\boldsymbol{f}\right|\right|^2$ such that $\left|\left|\boldsymbol{f}\right|\right|^2 = 1$

From $\boldsymbol{f}$ rearrange to get F then compute $E = K_l^T F K_r$ and extract R and t from $E = T_X R$ using singular value decomposition (SVD)

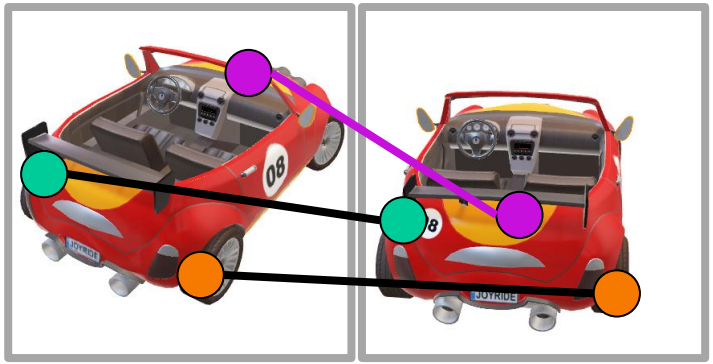
References

- Shree K. Nayer, First Principles of Computer Vision: Camera Calibration and Simple Stereo, https://www.youtube.com/watch?v=hUVyDabn1Mg

- Scaramuzza, D., Fraundorfer, F., Visual Odometry: Part I - The First 30 Years and Fundamentals, IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.

- Fraundorfer, F., Scaramuzza, D., Visual Odometry: Part II - Matching, Robustness, and Applications, IEEE Robotics and Automation Magazine, Volume 19, issue 1, 2012.

Problem of outliers in matching



All the matching pairs of features may not give a valid fundamental matrix

Issue: Deal with inliers (give valid F) and outliers (give invalid F)

RANSAC algorithm

If the number of outliers is less than 50%, RANSAC can work!

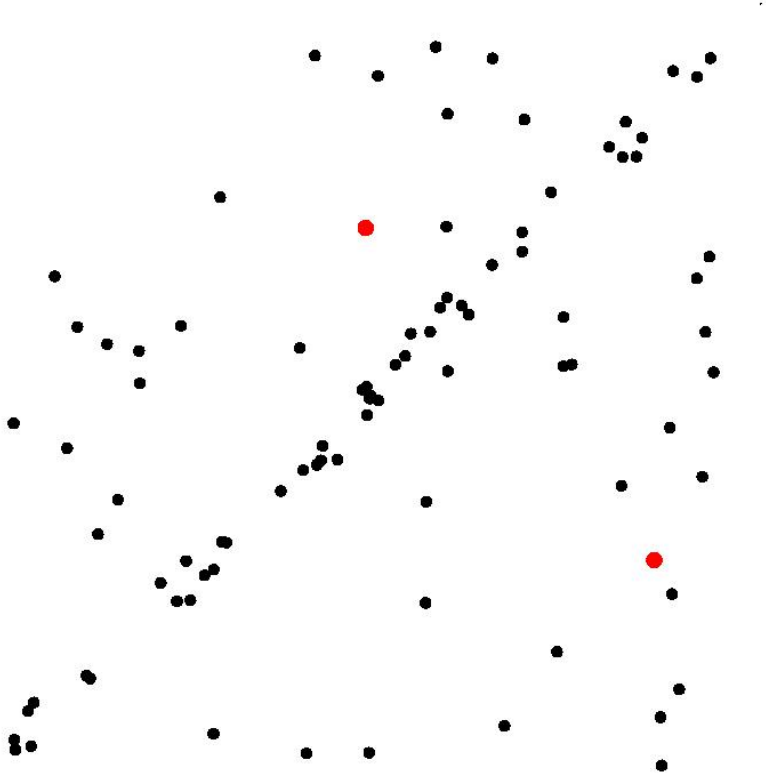
RANdom Sample Consensus

General RANSAC Algorithm

1. Randomly choose s samples. Typically, s is the minimum samples to fit the model
2. Fit the model to the randomly chosen samples
3. Count of the number M of data (inliers) that fit the model within a measure of error ε
4. Repeat Steps 1-3 N times
5. Choose the model that has the largest number M of inliers

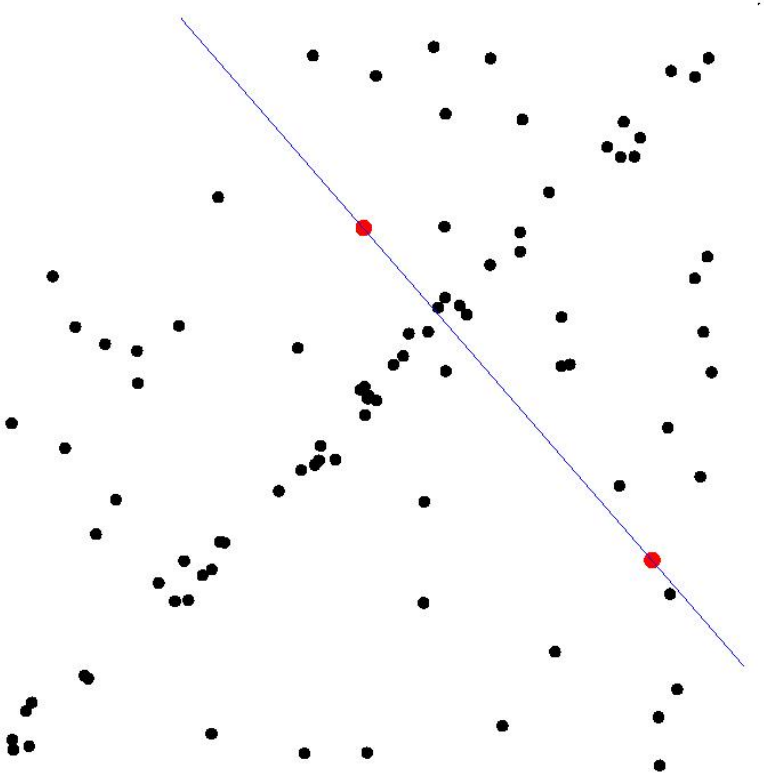
RANSAC Example: Line Extraction

RANSAC Example: Line Extraction



- Select sample of 2 points at random

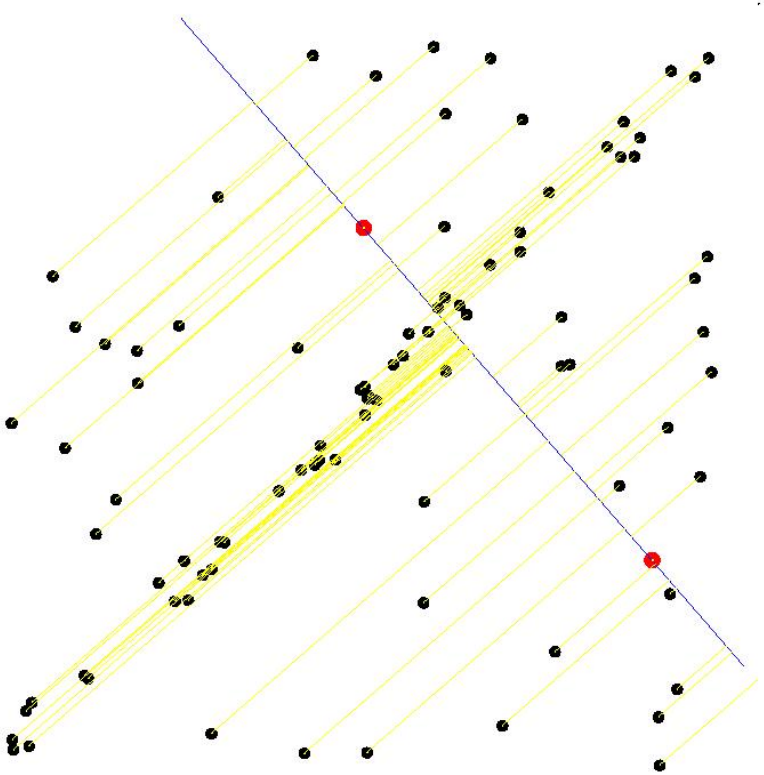
RANSAC Example: Line Extraction



- Select sample of 2 points at random

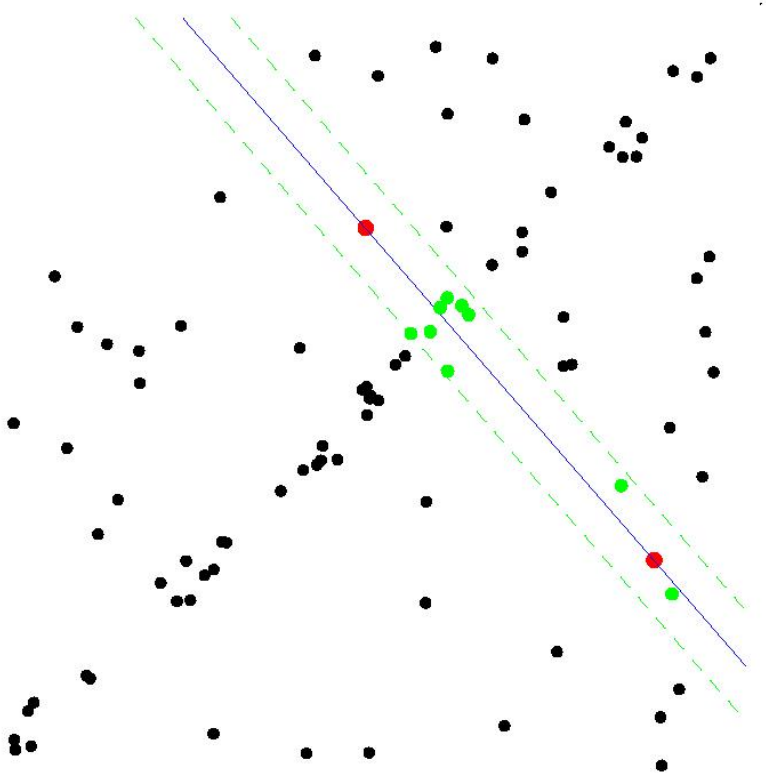
- Calculate model parameters that fit the data in the sample

RANSAC Example: Line Extraction



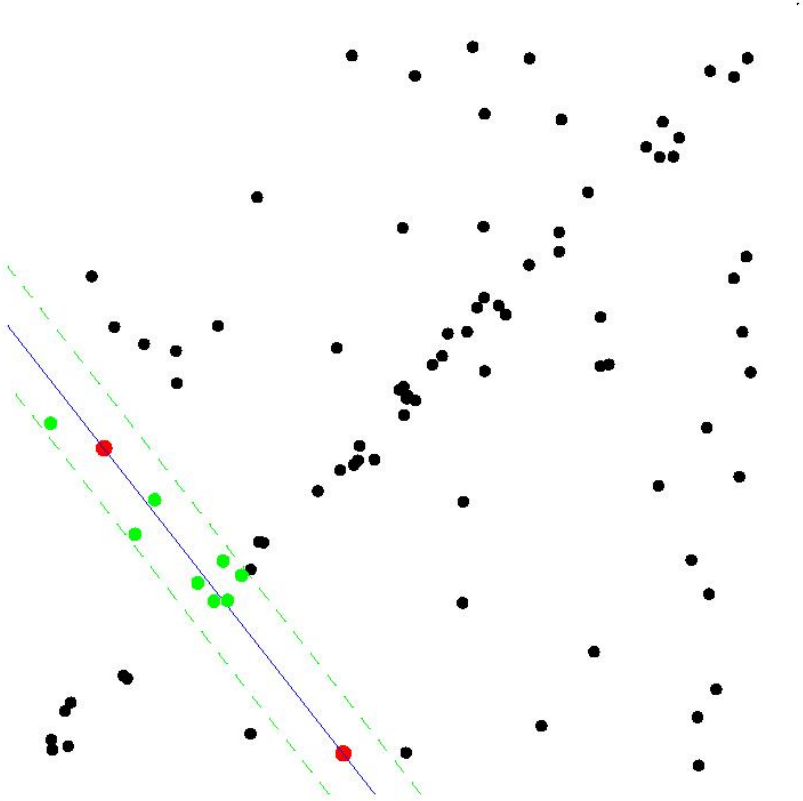
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point

RANSAC Example: Line Extraction



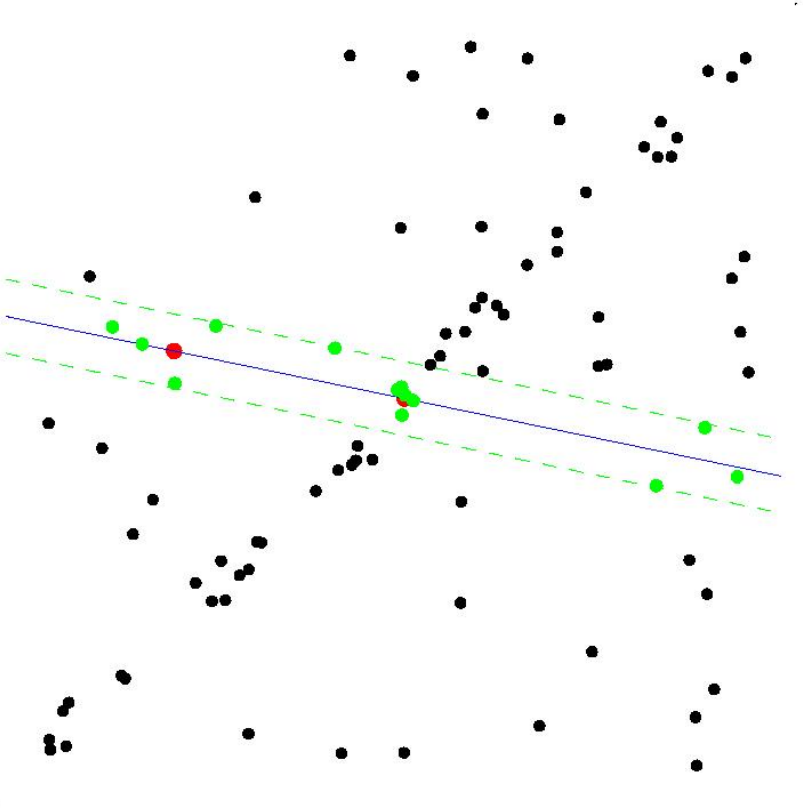
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis

RANSAC Example: Line Extraction



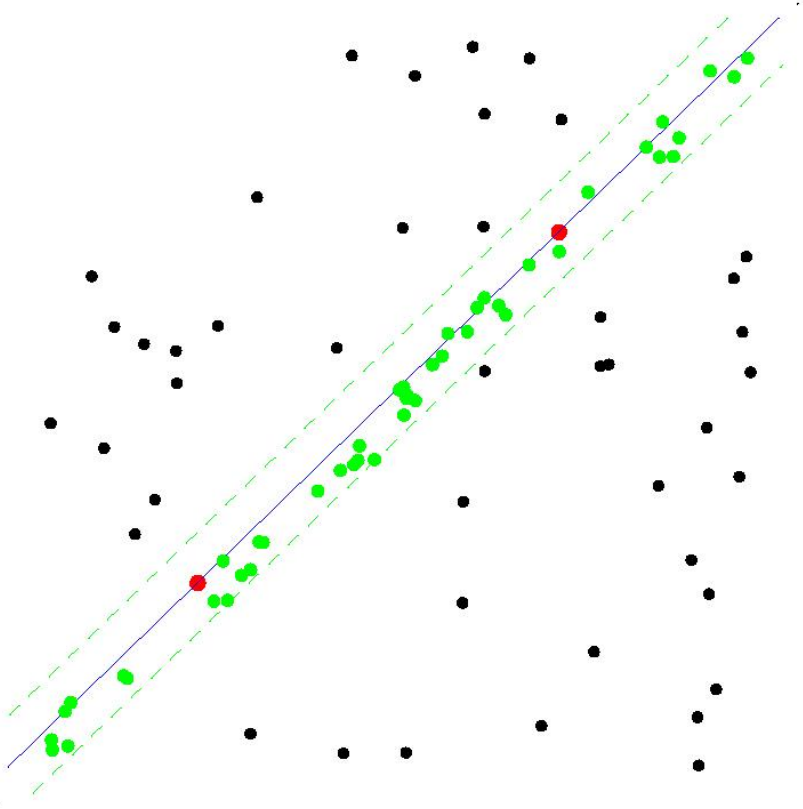
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis
- Repeat sampling

RANSAC Example: Line Extraction



- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis
- Repeat sampling

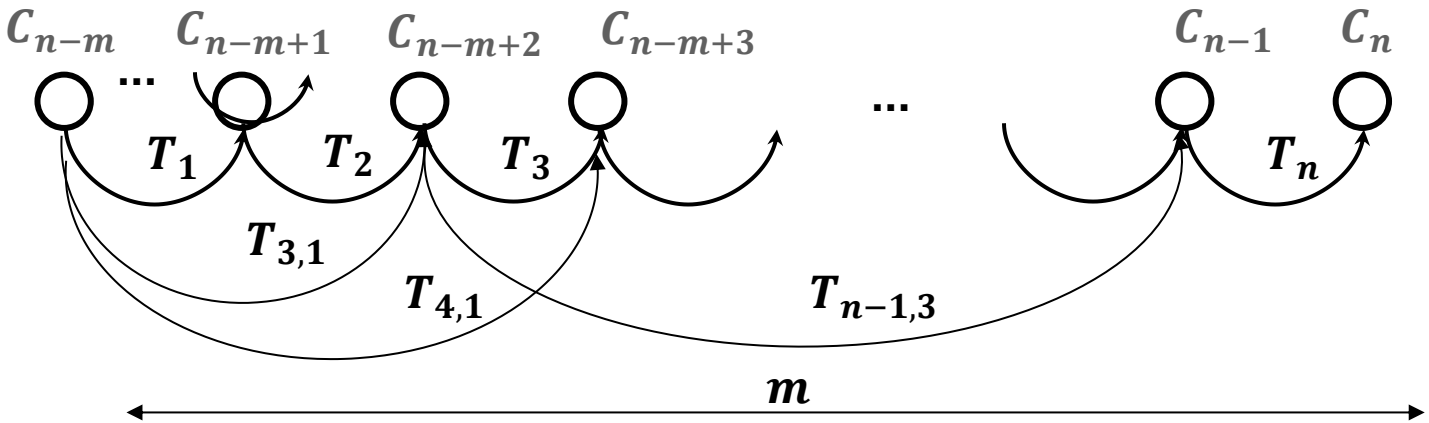
RANSAC Example: Line Extraction



ALL-INLIER SAMPLE

Windowed Camera-Pose Optimization

cameras poses *C*



- So far we assumed that the transformations $\boldsymbol{T_k} = [\boldsymbol{R_k}\,;\boldsymbol{t_k}]$ are between consecutive frames
- Transformations can be computed also between non-adjacent frames $T_{i,j}$ and can be used as additional constraints to improve cameras poses by minimizing $\sum_{i,j} \left|\left|C_i - T_{i,j}C_j\right|\right|^2$
- For efficiency, only the last m keyframes are used

VO or Structure from Motion (SFM)

SFM is more general than VO and tackles the problem of 3D reconstruction of both the structure and camera poses from unordered image sets

The final structure and camera poses are refined with an offline optimization (i.e., bundle adjustment), whose computation time grows with the number of images

VO focuses on estimating the 3D motion of the camera sequentially (as a new frame arrives) and in real time. Bundle adjustment is options.

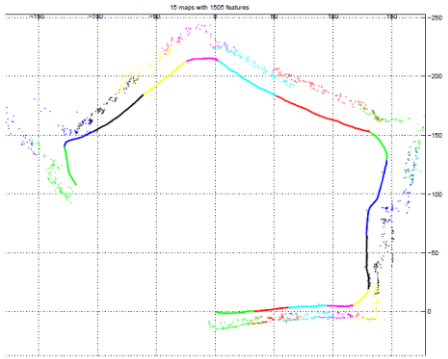
Video example
http://youtu.be/kxtQqYLRaSQ

Reconstruction from 3 million images from Flickr.com
Cluster of 250 computers, 24 hours of computation!
Paper: “Building Rome in a Day”, ICCV’09

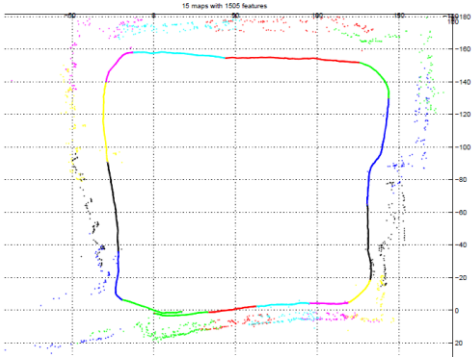
VO vs. Visual SLAM (1/2)

- The goal of SLAM in general is to obtain a global, consistent estimate of the robot path. This is done through identifying loop closures. When a loop closure is detected, this information is used to reduce the drift in both the map and camera path (global bundle adjustment).

- Conversely, VO aims at recovering the path incrementally, pose after pose, and potentially optimizing only over the last m poses path (windowed bundle adjustment)



Before loop closing



After loop closing

Image courtesy of Clemente et al. RSS'07

VO vs. Visual SLAM (2/2)

VO only aims to the local consistency of the trajectory

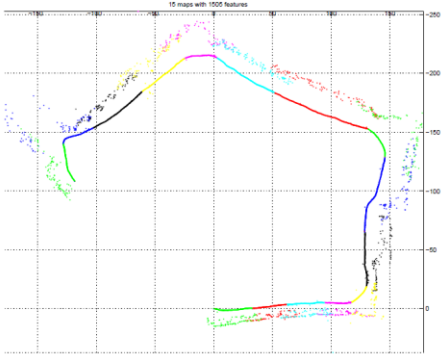
SLAM aims to the global consistency of the trajectory and of the map

VO can be used as a building block of SLAM

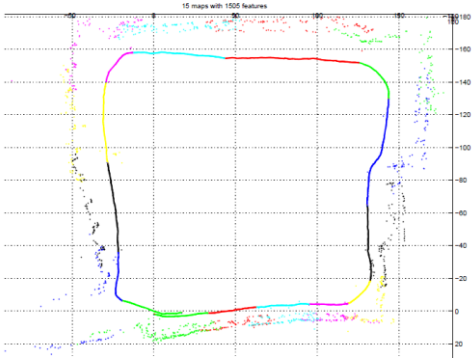
VO is SLAM before closing the loop!

The choice between VO and V-SLAM depends on the tradeoff between performance and consistency, and simplicity in implementation.

VO trades off consistency for real-time performance, without the need to keep track of all the previous history of the camera.



Visual odometry



Visual SLAM

Image courtesy of Clemente et al. RSS'07

Summary

- VO technique for incrementally finding the changes in camera pose from successive images
  - Assumes calibrated camera (intrinsics are known)
- Essential matrix E relates the 3D coordinates of the scene points and cane be decomposed to find the changes in translation and the orientation of the camera
  - Can be calculated from the fundamental matrix and the camera calibration
- Fundamental matrix F relates the pixel coordinates up to scale
  - Can be found using matched features and by solving an eigenvalue problem (constrained least square problem)
- RANSAC algorithm can remove outliers (e.g. in matching features for finding F)