Master "Automática y Robótica"

Técnicas Avanzadas de Vision:

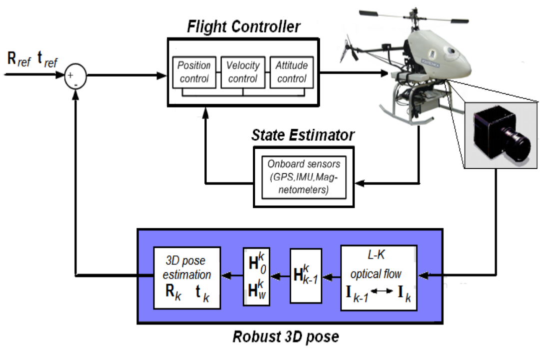
Visual Odometry

by
Pascual Campoy
Computer Vision Group
www.vision4uav.eu

Centro de Automática y Robótica
Universidad Politécnica de Madrid

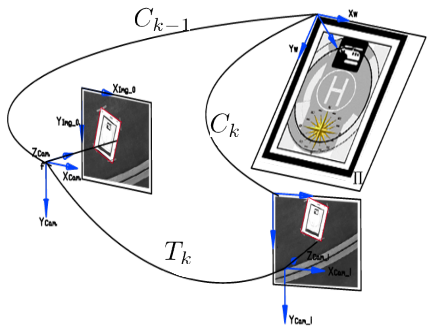
Visual Odometry: Objective

Estimate the egomotion using on-board cameras



Visual Odometry: **working principle**

Estimates incrementally the pose of the vehicle
by examination of the on-board image changes



Visual Odometry: **Sources**

- "Visual Odometry: Part I - The First 30 Years and Fundamentals"
  Scaramuzza, D., Fraundorfer, F.
  IEEE Robotics and Automation Magazine, Volume 18, issue 4, 2011.
- "Visual Odometry: Part II - Matching, Robustness, and Applications"
  Fraundorfer, F., Scaramuzza, D.
  IEEE Robotics and Automation Magazine, Volume 19, issue 2, 2012.
- "3_D Vision and Recognition"
  Kostas Daniilidis and Jan-Olof Eklundh
  Handbook of Robotoics, Siciliano, Khatib (Eds.), Springer 2008
- "Simultaneous Localization and Mapping"
  Sebastian Thrun, John J. Leonard
  Handbook of Robotoics, Siciliano, Khatib (Eds.), Springer 2008
- "On-board visual control algorithms for Unmanned Aerial Vehicles"
  Ivan F. Mondragón
  European PhD thesis at U.P.M. Nov. 2011.

Brief history of VO

- 1996: The term VO was coined by Srinivasan to define motion orientation
- 1980: First known stereo VO real-time implementation on a robot by Moraveck, PhD thesis (NASA/JPL) for Mars rovers
- 1980 to 2000: The VO research was dominated by NASA/JPL in preparation of 2004 Mars mission (papers by Matthies, Olson, ...)
- 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.

When V.O. for positioning?

Alternatives:

- Odometry:
  - Actuators (wheels) odometry
    - displacement measurement
  - Inertial Measurement Units (IMUs)
    - Aceleration measurement
- Global positioning:
  - GPS -Gyroscope - Magnetometer
  - 3D vision - Laser

Adventages:

- More accurate vs. wheel odometry or IMU (relative position error 0.1% – 2%)
- Necessary when global positioning is not available
- Useful for sensor fusion

Visual Odometry: **Steps**

1. Image acquisition and correction
2. Feature detection and description
3. Feature matching
4. Robust matching for pose estimation
5. Pose optimization

Visual Odometry: **Steps**

1. Image acquisition and correction
   1. Acquisition using either single cameras, stereo cameras, or omnidirectional cameras.
   2. Correction: preprocessing techniques for lens distortion removal, noise removal, etc.
2. Feature detection and description
3. Feature matching
4. Robust matching for pose estimation
5. Pose optimization

Visual Odometry: **Steps**

1. Image acquisition and correction
2. Feature detection and description
    1. Feature detection: corner detectors (Moravec, Forstner, Harris, Shi-Tomasi, FAST) or blob detectors (SIFT, SURF, CENSUR)
    2. Feature description: local appearance or invariant descriptors (SIFT, SURF, BRIEF, ORB, BRISK, FAST)
3. Feature matching
4. Robust matching for pose estimation
5. Pose optimization

Visual Odometry: **Steps**

1. Image acquisition and correction
2. Feature detection and description
3. Feature matching

    Local tracking (LK, KLT)

    vs.

    Global matching
4. Robust matching for pose estimation
5. Pose optimization

Table of contents

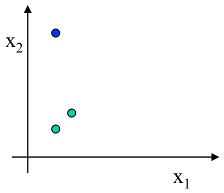
3. Global feature matching
4. Robust matching for pose estimation
5. Pose optimization

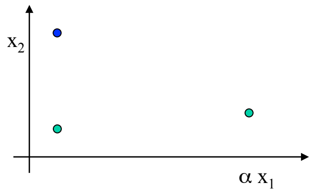
Table of contents

3. Global feature matching
   - Similarity measurement
   - Mutual consistency
   - Motion consistency
4. Robust featuring
5. Pose estimation

Feature matching: Global feature matching

- Similarity
  - Distance in the feature space $\sqrt{\sum_{x,y}(f(x,y)-t(x,y))^2}$
  - Normalized cross correlation $\frac{1}{n}\sum_{x,y}\frac{(f(x,y)-\overline{f})(t(x,y)-\overline{t})}{\sigma_f\sigma_t}$



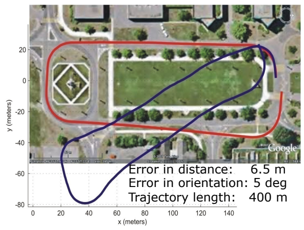


- Mutual consistency:
  only pairs where one point selects each other as the closest
- Motion consistency:
  only pairs where one point is accordingly where it should, taking into account the motion model

Table of contents

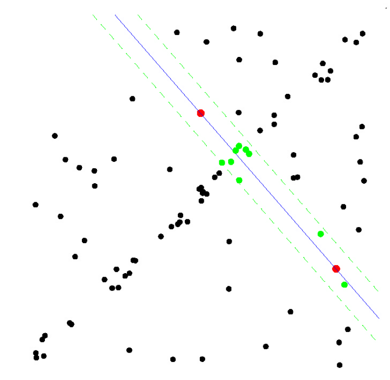
Robust matching

- Problem: false matched points (i.e. outliers) result in errors in pose estimation
  (caused in image acquisition (noise, blur, ..), feature detetor/ descriptor or matching)
- Solution: remove outliers don't fitting predominant model.
- RANSAC is the standard
  - it stands for random sample consensus
  - first by Fishler & Bolles, 1981



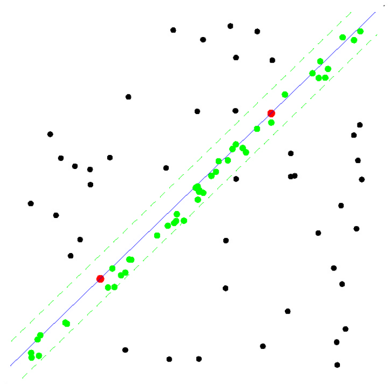
Source: Scaramuzza

RANSAC: working principle



1. Randomly choose *s* samples
   Typically *s* = minimum sample size that lets fit a model
2. Fit a model (e.g., line) to those samples
3. Count the number of inliers that approx. fit the model
   (distance to model <d)

RANSAC: working principle



1. Randomly choose *s* samples
   Typically *s* = minimum sample size that lets fit a model
2. Fit a model (e.g., line) to those samples
3. Count the number of inliers that approx. fit the model
   (distance to model <d)
4. Repeat *N* times
5. Choose the model that has the largest set of inliers

RNSAC: number of iterations

The number of iterations necessary to guarantee a correct solution is:

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

s is the number of points to obtain a model
ε is the rate of outliers in the data
p is the probability of success

Example: p=99.9%, s=2, ε =25% ➔ N= 8.35

Features:

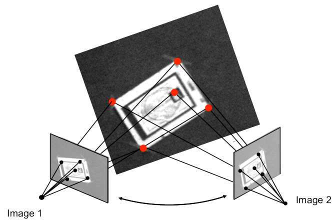
- RANSAC is non deterministic, whose solution tends to be stable when N grows
- N is usually multiply by a factor of 10
- Advanced implementations estimate ε after every iteration

RANSAC for Visual Odometry

1. Randomly choose s samples
2. Fit the motion model

Obtain

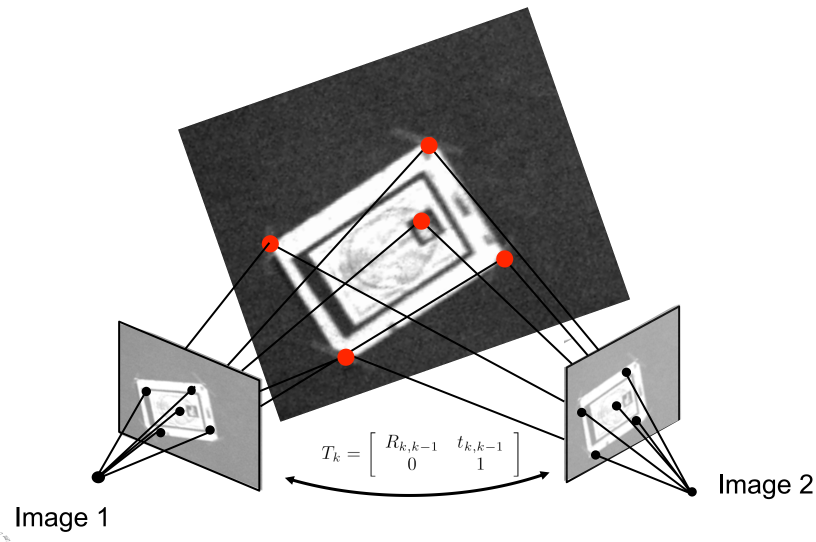
$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$



it can be calculated by minimizing the following points correspondences: 2D-2D, 3D-3D or 3D-2D

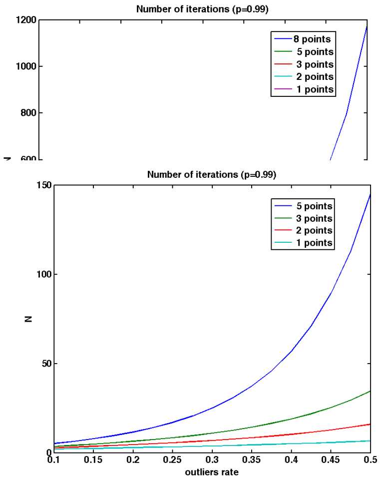
1. Count the number of inliers that approx. fit the model (distance to model <d)
2. Repeat N times
3. Choose the model that has the largest set of inliers

RANSAC for V.O.: motion model



RANSAC for V.O.: nr. of points

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

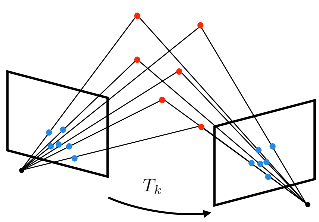


- For a 6 DOF uncalibrated/calibrated camera:
  8 points non coplanar points algorithm by Longguet-Higgins' (1981)
- For a 6 DOF calibrated camera:
  5 points are enough Krupta (1913), efficient implementation by Nister (2003)
- If 2angles are known:
  3 points are enough by Fraundorfer et alt. (2010), 2 angles estimation by far point by Narodisky et alt.(2011)
- If 3 angles are known:
  2 points are enough by Kneip at alt. (2011)
- For planar motion
  2 points are enough by Ortin et alt. (2001)
- For wheeled vehicles of 2DOF
  1 point is enough by Scaramuzza et alt. (2011)

Motion from Image Feature Correspondences: 2D-2D

- The minimal-case solution involves 5-point correspondences
- The solution is found by determining the transformation that minimizes the reprojection error of the triangulated points in each image

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg\min_{X^i, C_k} \sum_{i,k} \| p_k^i - g(X^i, C_k) \|^2$$



$p_2^T \, E \, p_1 = 0$ Epipolar constraint

$E = [t]_\times R$ Essential matrix

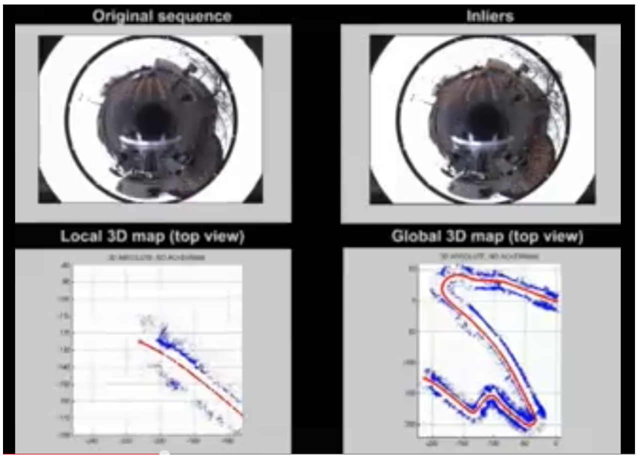
$$p_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad p_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

RANSAC for V.O.: nr of points

Is it really better to use minimal sets in RANSAC?

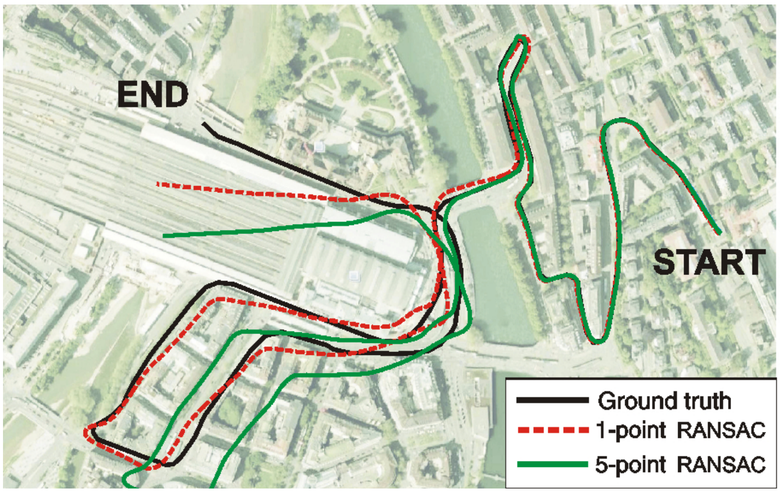
- If one is concerned with certain speed requirements, YES
- However, might not be a good choice if the image correspondences are very noisy: in this case, the motion estimated from a minimal set wil be inaccurate and will exhibit fewer inliers when tested on all other points
- Therefore, when the computational time is not a real concern and one deals with very noisy features, **using a non-minimal set may be better than using a minimal set**

RANSAC for V.O.: results for 1 point



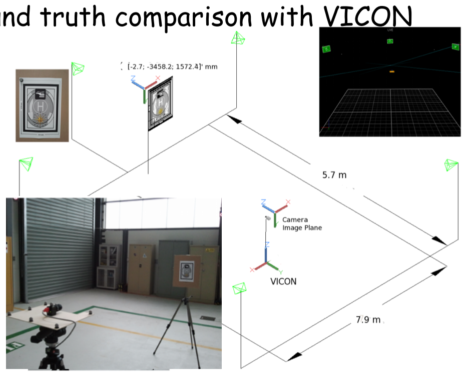
This video can be seen at
http://youtu.be/t7uKWZtUjCE

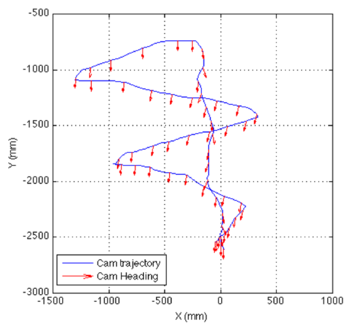
RANSAC for V.O.: results for 1 point



RANSAC for V.O.: results for 5 points

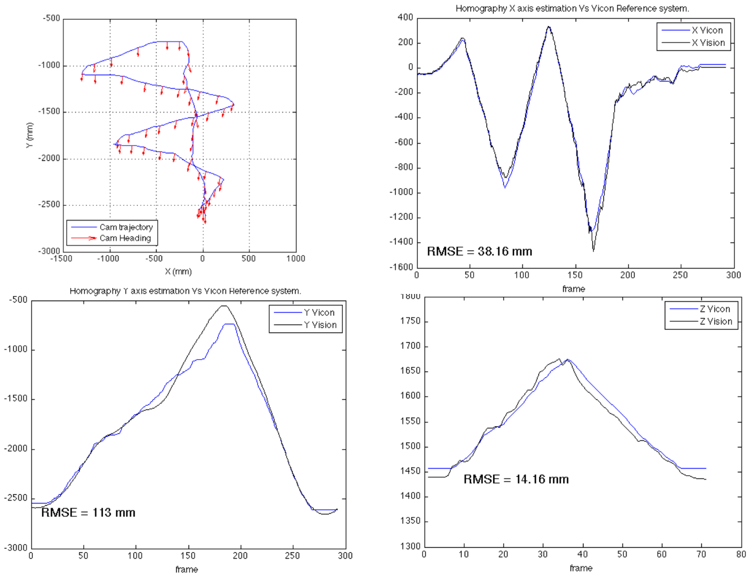
Ground truth comparison with VICON



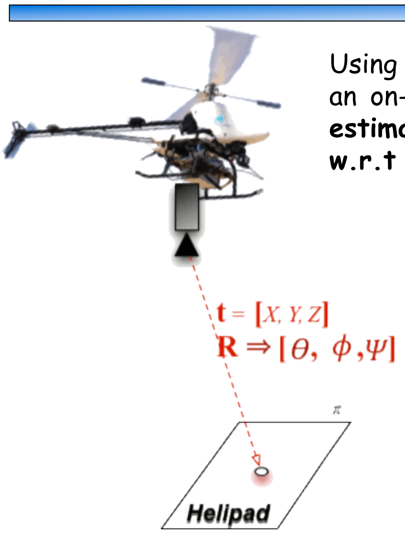


img1 img50 img100 img150 img200 img250

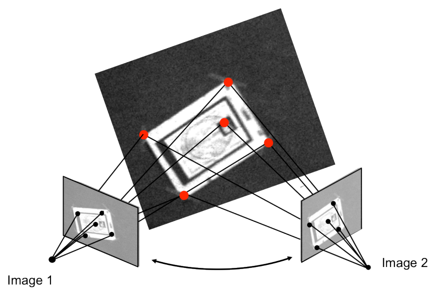
RANSAC for V.O.: results for 5 points



RANSAC for V.O.: results for 5 points

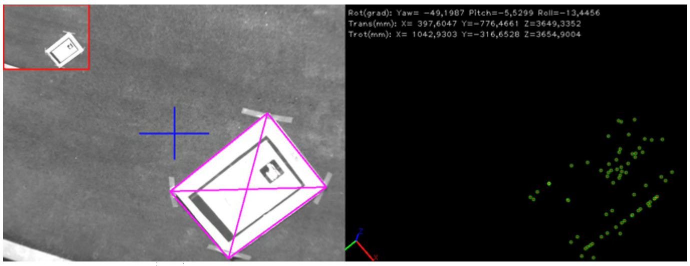


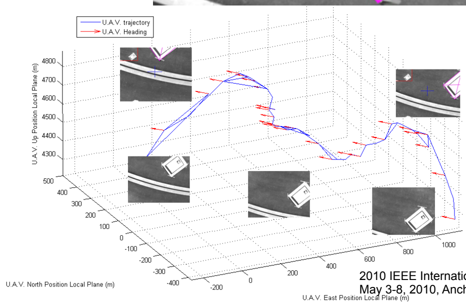
Using only **visual information** provided by an on-board camera and a know landmark, **estimate** camera-aircraft **relative position w.r.t a helipad**



RANSAC for V.O.: results for 5 points

Hover at 4.5m

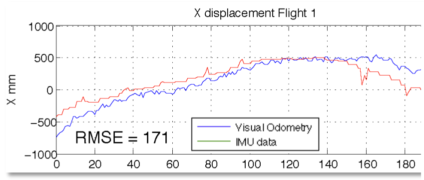


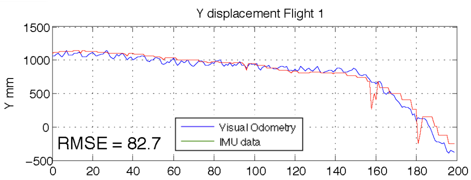


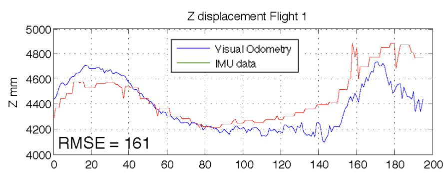
2010 IEEE International Conference on Robotics and Automation Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

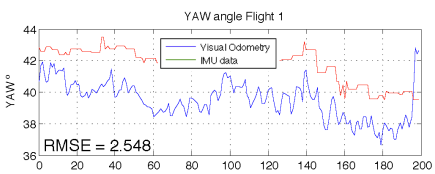
36

RANSAC for V.O.: results for 5 points





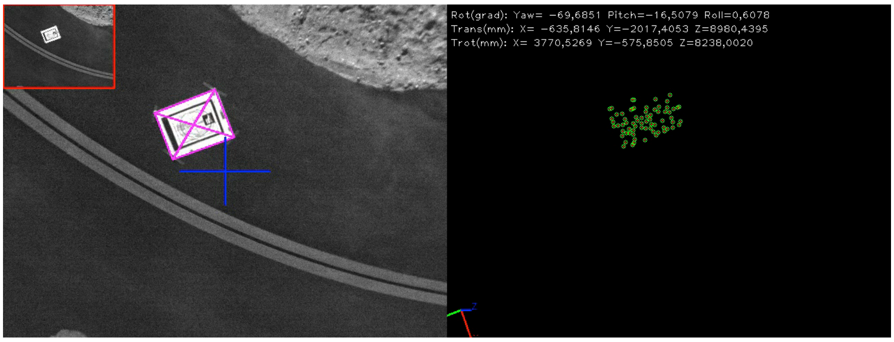




37

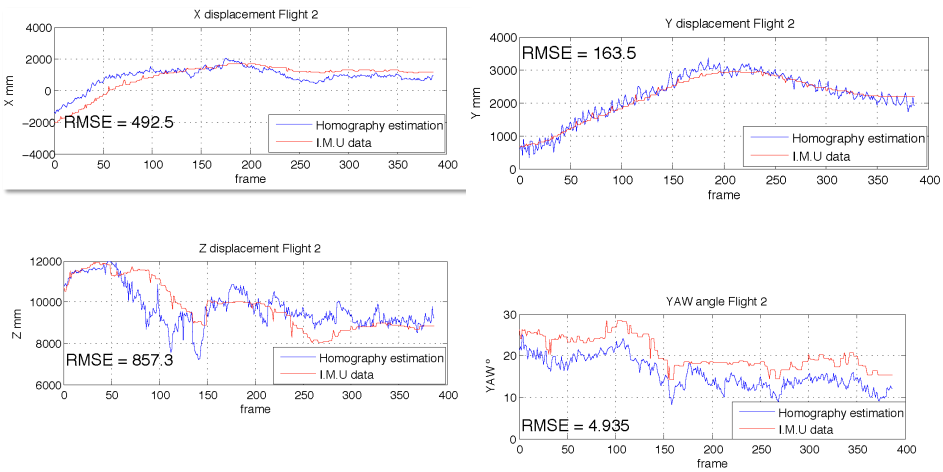
RANSAC for V.O.: results for 5 points

Hover at 10m



38

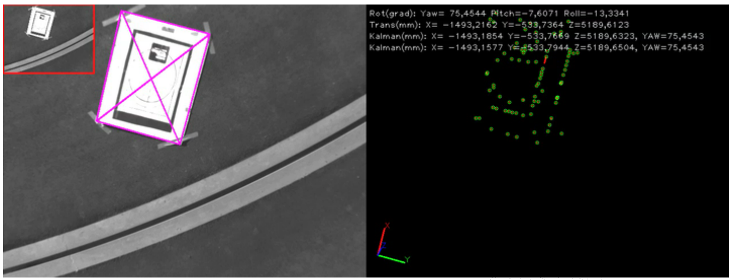
RANSAC for V.O.: results for 5 points

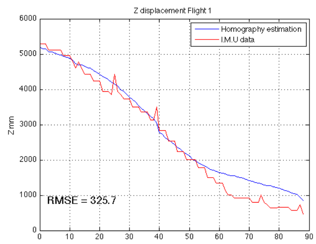


39

RANSAC for V.O.: results for 5 points

Manual Landing





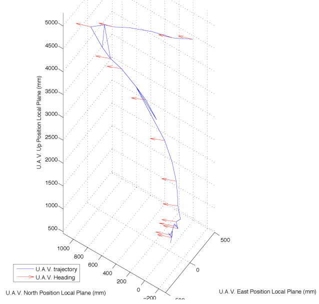
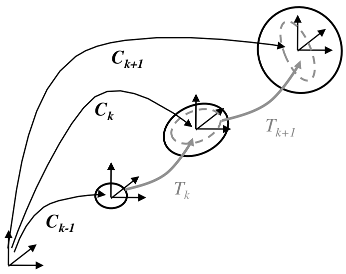


Table of contents

3. Global feature matching
4. Robust matching
5. Pose optimization

Error Propagation

- The uncertainty of the camera pose is a combination of the uncertainty at (black-solid ellipse) and the uncertainty of the transformation (gray dashed ellipse)



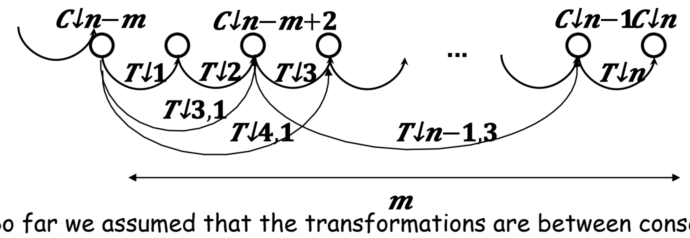
- The combined covariance is

$$\begin{aligned}\Sigma_k &= J \begin{bmatrix} \Sigma_{k-1} & 0 \\ 0 & \Sigma_{k,k-1} \end{bmatrix} J^\top \\ &= J_{\vec{C}_{k-1}} \Sigma_{k-1} J_{\vec{C}_{k-1}}{}^\top + J_{\vec{T}_{k,k-1}} \Sigma_{k,k-1} J_{\vec{T}_{k,k-1}}{}^\top\end{aligned}$$

- The camera-pose uncertainty is always increasing when concatenating transformations. Thus, it is important to keep the uncertainties of the individual transformations small

Source Scaramuzza

Windowed Camera-Pose Optimization



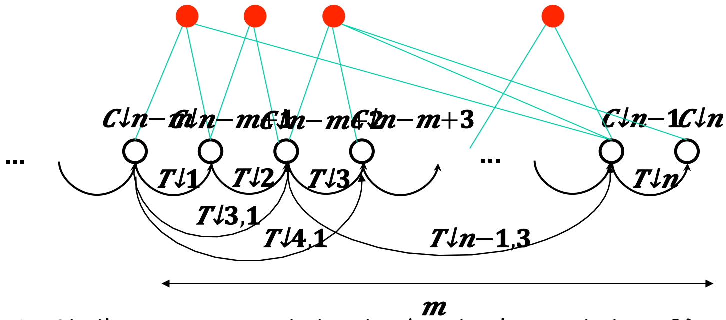
- So far we assumed that the transformations are between consecutive frames
- Transformations can be computed also between non-adjacent frames and can be used as additional constraints to improve cameras poses by minimizing the following

$$\sum_{e_{ij}} \|C_i - T_{e_{ij}} C_j\|^2$$

- For efficiency, only the last keyframes are used
- Levenberg-Marquadt can be used

Source Scaramuzza

Windowed Bundle Adjustment (BA)



- Similar to pose-optimization but it also optimizes 3D points

$$\arg\min_{X^i, C_k} \sum_{i,k} \| p_k^i - g(X^i, C_k) \|^2$$

- In order to not get stuck in local minima, the initialization should be close the minimum
- Levenberg-Marquadt can be used

Source Scaramuzza

When apply V.O. ?

Is any of these scenes good for VO? Why?

Source Scaramuzza

- Sufficient illumination in the environment
- Dominance of static scene over moving objects
- Enough texture to allow apparent motion to be extracted
- Sufficient scene overlap between consecutive frames

Other Applications: Mosaics

questions ?

more info: www.vision4uav.es