Vision-based navigation
Simon Lacroix

To cite this version:
Simon Lacroix. Vision-based navigation. Doctoral. Spring School on Location-Based Services, École Nationale d’Aviation Civile, Toulouse (France), France. 2015. <cel-01276880>

HAL Id: cel-01276880
https://hal.archives-ouvertes.fr/cel-01276880
Submitted on 20 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Vision-based navigation

Simon Lacroix
Robotics and InteractionS (RIS)
LAAS/CNRS, Toulouse

http://homepages.laas.fr/simon/
Where do I come from?

Robotics at LAAS/CNRS, Toulouse, France

• Research topics
  – Perception, planning and decision-making, control
  – Plus: control architecture, interactions, ambient intelligence systems, learning

  A keyword: autonomy

• Research domains
  – Cognitive and interactive Robotics
  – Aerial and Terrestrial Field Robotics
  – Human and anthropomorphic motion
  – Bio-informatics, Molecular motion

• Considered applications: Planetary exploration, Service and personal robotics, virtual worlds and animation, biochemistry, embedded systems, transport, driver assistance, defense, civil safety
Open source software tools: [www.openrobots.org](http://www.openrobots.org)
What am I working on? Field robotics

• Environment perception and modeling

• Localization and SLAM

• Autonomous rover navigation

• Multi-Robot cooperation
Robotics

“From automatic control to autonomous control”

In laboratories

Industrial
Robotics

“From automatic control to autonomous control”

Robots everywhere
“From automatic control to autonomous control”

• Automatic control :
  – Well defined task ("regulate variable", "follow trajectory"…)
  – “Direct” link between (simple) perception and action

• Autonomous control :
  – More general task ("reach position", "monitor area"…)
  – Calls for decisional processes

⇒ “perception / Decision / Action” loop

Plus :
  – Processes integration
  – Learning
  – Interaction with humans
  – Interactions with other robots
  – …
“From automatic control to autonomous control”

E.g. for a drone:

– Regulate heading / speed / altitude

– Follow a list ordered waypoints

– Follow a geometric trajectory

– Follow a road

– Follow a target

– Survey an area while avoiding threats and obstacles

“Decision”: notion of deliberation, planning, prediction and evaluation of the outcomes of an action
Anatomy of a robot

- Stereovision cameras
- Processing units
- Motor control electronics
- IMU
- Odometers

Decision

Perception

Action

Chassis
Anatomy of a robot

Decision

Perception

Action
Illustration: autonomous navigation

navigation |ˌnaveɪˈɡeɪʃn|
noun
1 the process or activity of accurately ascertaining one’s position and planning and following a route.

**Perception**
- detect obstacles, traversable areas, localize the robot

**Decision**
- avoid obstacles, find trajectories, itineraries

**Action**
- ensure the execution of the planned motions
An elementary decision: AGV obstacle avoidance

Simple instance of a perception / decision / action loop:
• Gathering data on the environment
• Structuring the data into a *model*
• *Planning* the trajectory to find the “optimal” one
• Executing the trajectory
Perception in robotics

Perception:
« Acquisition and representation of information on the environment and the robot itself »

Proprioceptive and exteroceptive sensors:

**Proprioceptive** |

* adjective Physiology
* relating to stimuli that are produced and perceived within an organism, esp. those connected with the position and movement of the body. Compare with **exteroceptive** and **interoceptive**.

**Exteroceptive** |

* adjective Physiology
* relating to stimuli that are external to an organism. Compare with **interoceptive**.
Why is localization so important?
Why is localization so important "in robotics"?

Localization is required to:

• Ensure the spatial consistency of the built environment models

• Ensure the achievement of the missions, most often defined in localization terms ("goto [goal]", "explore / monitor [area]", ...)

• Ensure the proper execution of paths / trajectories

• Ensure the lowest level (locomotion) controls
But… what localization?

Essential questions to answer:

1. With which precision? From cm to meters
2. In which frame? Absolute vs. local
3. At which frequency? From kHz to “sometimes”
4. Integrity of the solution?
5. Disponibility of the solution?

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models

- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, …)
Outline

• Autonomous robots
• On the importance of localization
• Localization using dead-reckoning approaches
Odometry

- Odometry: estimation of \((x,y,\theta)\) by integration of elementary motions

Wheel rotation encoders
**Odometry**

**Linear wheel speeds:**

\[ v_g = r_g \frac{dq_g}{dt} \quad v_d = r_d \frac{dq_d}{dt} \]

**Linear speed:**

\[ \dot{x} = \frac{1}{2} (v_g + v_d) \cos(\theta_k + \frac{d\theta}{2}) \]
\[ \dot{y} = \frac{1}{2} (v_g + v_d) \sin(\theta_k + \frac{d\theta}{2}) \]

**Angular speed:**

\[ \dot{\theta} = \frac{v_g - v_d}{E} \]
Odometry

- Exemple: linear motion

- Measured distance:
  \[ d_1 = \hat{d}_1 + \tilde{d}_1 \]

- Error model:
  \[
  \begin{align*}
  E(\tilde{d}_1) &= \mu_1 \\
  \text{Var}(\tilde{d}_1) &= \sigma^2_1
  \end{align*}
  \]

  For instance:
  \[
  \begin{align*}
  \mu_1 &= 0 \\
  \sigma_1 &= 0.1 \cdot \hat{d}_1
  \end{align*}
  \]
Gaussian error model:

\[ \bar{d}_1 \sim N(\mu_1, \sigma_1^2) \]

\[ p(\bar{d}_1) = \frac{1}{\sqrt{2\pi} \sigma_1} e^{-\frac{1}{2} \left( \frac{\bar{d}_1 - \mu_1}{\sigma_1} \right)^2} \]

\( Pr \left\{ | \bar{d}_1 - \mu_1 | \leq \sigma_1 \right\} \approx 0.68 \)
\( Pr \left\{ | \bar{d}_1 - \mu_1 | \leq 2\sigma_1 \right\} \approx 0.95 \)
\( Pr \left\{ | \bar{d}_1 - \mu_1 | \leq 3\sigma_1 \right\} \approx 0.997 \)

Odometry error model:

\[ d_1 = \hat{d}_1 + \bar{d}_1 \]
\[ \bar{d}_1 \sim N(\mu_1, \sigma_1^2) \]
\[ \mu_1 = 0 \]
\[ \sigma_1 = 0.1 \cdot \bar{d}_1 \]

(this is a model)
Odometry

- Example: move 0.9m

\[ \begin{align*}
    x_{WR_1} &= \hat{d}_1 \\
    &= 0.9m \\
    \sigma_{x_{WR_1}} &= 0.09m
\end{align*} \]

- Robot moves again 0.85m:

\[ \hat{d}_2 = 0.85m \]

\[ \begin{align*}
    x_{WR_2} &= \hat{d}_1 + \tilde{d}_1 + \hat{d}_2 + \tilde{d}_2 \\
    E(\tilde{d}_1 + \tilde{d}_2) &= \mu_1 + \mu_2 = 0 \\
    \text{Var}(\tilde{d}_1 + \tilde{d}_2) &= \sigma_1^2 + \sigma_2^2 \\
    &= 0.1^2 (\tilde{d}_1^2 + \tilde{d}_2^2)
\end{align*} \]
• Example: move 0.9m

\[
\begin{align*}
W & \rightarrow R_1 \\
\hat{x}_{WR_1} &= \hat{d}_1 \\
&= 0.9m \\
\sigma_{x_{WR_1}} &= 0.09m \\
\end{align*}
\]

• Robot moves again 0.85m:

\[
\begin{align*}
W & \rightarrow R_2 \\
\hat{d}_2 &= 0.85m \\
\end{align*}
\]

\[
\begin{align*}
\hat{x}_{WR_2} &= \hat{d}_1 + \hat{d}_1 + \hat{d}_2 + \hat{d}_2 \\
E(\hat{d}_1 + \hat{d}_2) &= \mu_1 + \mu_2 = 0 \\
\text{Var}(\hat{d}_1 + \hat{d}_2) &= \sigma_1^2 + \sigma_2^2 \\
&= 0.1^2 (\hat{d}_1^2 + \hat{d}_2^2) \\
\end{align*}
\]

• New estimation:

\[
\begin{align*}
W & \rightarrow R_2 \\
\hat{x}_{WR_2} &= 1.75m \\
\sigma_{x_{WR_2}} &\approx 0.12m \\
\end{align*}
\]

Grows!
Odometry

Now angular rotations come into play

Monte-Carlo simulation:
Odometry: illustration

- With an indoor robot
Localization solutions in robotics

- Odometry
- Similarly, motion / accelerations sensors (inertial navigation)

  Inherently drifts over time and distances, subject to slippages and skids

  Develop solutions relying on the robot exteroceptive sensors
A few words on vision

- **Cameras**: low cost, light and power-saving
- **Perceive data**
  - In a volume
  - Very far
  - Very precisely

- **Stereovision**
  - 2 cameras provide depth

- **Images carry a vast amount of information**

- **A vast know-how exists in the computer vision community**
Camera geometric model

• Pinhole model

Perspective projection

\[(u,v) = f(x,y,z)\]

\[u = k_u f x / z + u_0\]
\[v = k_v f y / z + v_0\]

\[p = I_c P\]

\[I_c : \text{camera intrinsic matrix}\]
Camera geometric model

(But the real world is more complex)
Camera calibration

- **Principle**
  1. Acquisition of images of a known calibration pattern
  2. Extraction of the pattern features
  3. Association between extracted features and pattern features
  4. Use of a minimization technique to estimate the projection parameters

Classic calibration patterns

Well known techniques, available softwares on line (*e.g.* Matlab calibration toolbox, openCV library)
• Triangulation

\[ \rho = B \sin \alpha \left( \frac{1}{\tan \beta} + \frac{1}{\tan \alpha} \right) \]

3D data from vision
3D data from vision

- Triangulation

The precision depends on the baseline
• Autumn 2010: Kinect

3D data from vision
• Stereovision = triangulation

2 angles, 1 distance: \[ d = \frac{b}{\tan(\alpha) + \tan(\beta)} \]
Dense stereovision

0. Image pair acquisition

1. Image rectification

2. Pixel matching process

3. 3D reconstruction
Single camera stereovision
(« Structure from Motion »)

- One moving camera

Additional difficulty: estimation of the motion parameters

(« structure and motion from motion », « SLAM »)
Localization solutions in robotics

- Odometry
- Similarly, motion / accelerations sensors (inertial navigation)

Inherently drifts over time and distances, subject to slippages and skids

→ Develop solutions relying on the robot exteroceptive sensors
Visual odometry: principle
Visual odometry: results

- Applied on the Mars Exploration Rovers

![Graph showing 50% slip](image)

50 % slip
Visual odometry: benefits

• Contrary to 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%)

• In GPS-denied environments (e.g. underwater, planetary or indoor), VO has utmost importance
Visual odometry: implied functions

1. Dense stereovision

2. Feature detection

3. Feature matching

4. Motion estimation
Visual odometry: implied functions

1. Dense stereovision

2. Feature detection

3. Feature matching

4. Motion estimation

Image pair sequence
Harris interest points: sharp peaks of the autocorrelation function

Auto-correlation matrix:
\[
G(x, \bar{\sigma}) \otimes \begin{pmatrix}
I_u(x, s\sigma)^2 & I_u(x, s\sigma)I_v(x, s\sigma) \\
I_u(x, s\sigma)I_v(x, s\sigma) & I_v(x, s\sigma)^2
\end{pmatrix}
\]
\[
I_u(x, s\sigma) = s G_u(x, s\sigma) * I(x),
\]
\[
I_v(x, s\sigma) = s G_v(x, s\sigma) * I(x)
\]

Principal curvatures defined by the two eigen values of the matrix \( \lambda_1, \lambda_2 \)
(s: scale of the detection)
Interest points

- Local gray value invariants [Harris 88][Schmid 97]: local description vector invariant to rotation

Find matches with the similarity measure of descriptors
Harris interest points
Interest points

- Numerous interest points definition available
  - Scale invariant interest points (“SIFT”, [Lowe99])
  - Speeded-up robust features (“SURF”, [Bay 2006])
  - Binary Robust Independent Elementary Features (“BRIEF”, [Calonder 2010])
  - …

The choice depends on the considered problem context
Visual odometry: implied functions

1. Dense stereovision
2. Feature detection
3. Feature matching
4. Motion estimation
Interest points matching: principle

- If a generated match is correct,
  - Similar principal curvatures
  - *Applied to matching candidate selection*
Interest points matching: principle

- If a generated match is correct,
  - In the vicinity of the match, neighbor matches must exist
  - *Local group matching: consideration a local region of image*
Some matching results

Consecutive frames

Small overlap
Visual odometry: implied functions

- Image pair sequence
- Feature detection
- Feature matching (tracking)
- Dense stereovision
- Motion estimation
Motion estimation: problem statement

- Given two set of matched 3D points

- Find the 3D transformation $T_k$ that minimizes the point distances

\[
T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg \min_{T_k} \sum_i ||\tilde{X}_k^i - T_k\tilde{X}_{k-1}^i||
\]
Motion estimation: least square minimization

- Given two set of matched 3D points \( \{p_i\}, \{p'_i\} \)
- The coordinates of matching points are linked by:
  \[
  p'_i = R.p_i + T + N_i
  \]
  Where \( N_i \) denotes some noise
- Least square estimation: find the 3D transform \((R,T)\) that minimizes:
  \[
  \Sigma^2 = \sum_{i=1}^{N} \left\| p'_i - (R.p_i + T) \right\|^2
  \]
Motion estimation: least square minimization

• Assuming a zero-mean noise on the points coordinates, the barycenter of the transformed first set and the second set should be equal:

\[ p' = p'' \]

where \( p = \frac{1}{N} \sum_{i=1}^{N} p_i \); \( p' = \frac{1}{N} \sum_{i=1}^{N} p'_i \); \( p'' = \frac{1}{N} \sum_{i=1}^{N} \hat{R}.p_i + \hat{T} \)

• Changing the coordinates: \( q_i = p_i - p \); \( q'_i = p'_i - p' \)
then we have \( \Sigma^2 = \sum_{i=1}^{N} \|q'_i - R.q_i\|^2 \)

1. Find \( R \) that minimizes \( \Sigma^2 \) (least square minimization)
2. Estimate \( T \) as \( \hat{T} = p' - \hat{R}.p \)
• Matched points are usually contaminated by wrong data associations (= “outliers”)
• Possible causes of outliers are
  - image noise
  - occlusions
  - blur,
  - changes in view point and illumination for which the feature detector or descriptor does not account for
• For the camera motion to be estimated accurately, outliers must be removed
• This is the task of robust estimation

• “RANSAC” (“Random Sample Consensus”) [Fishler & Bolles, 1981] has been established as the standard method for motion estimation in the presence of outliers
Results / Robust estimation

- Error at the loop closure: 6.5 m
- Error in orientation: 5 deg
- Trajectory length: 400 m

Before removing the outliers
After removing the outliers

[D. Scaramuzza @ ETHZ]
Localization solutions in robotics

• Odometry
• Similarly, motion / accelerations sensors (inertial navigation)
  Inherently drifts over time and distances, subject to slippages and skids
  → Develop solutions relying on the robot exteroceptive sensors

• Visual odometry: akin to dead reckoning
  Inherently drifts over time and distances
  → Develop solutions relying on the robot exteroceptive sensors that memorizes stable environment features (SLAM)
SLAM
Simultaneous Localisation And Mapping
Principle of SLAM

- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- **Estimation**: refinement of the landmark and robot positions
Principle of SLAM
• Telemetry (« measuring distances »)

Laser (« Light Amplification by Stimulated Emission of Radiation »)
Time of flight of phase shift measures
⇒ LIDAR (« Light Detection And Ranging »)

In the plane:

(parenthesis: 2D Lidars)
SLAM: illustration

Occupancy grid: on a flat plane, with odometry and “2D” LRF (S. Thrun)
SLAM : illustration

Occupancy grid: on a flat plane, with odometry and “2D” LRF (S. Thrun)
SLAM : illustration

Occupancy grid: on a flat plane, with odometry, a “2D” LRF and SLAM (S. Thrun)
SLAM: illustration

Occupancy grid: on a flat plane, with odometry, a “2D” LRF and SLAM (S. Thrun)
SLAM must reads

There are tons of papers on SLAM...

At least, read those:

Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms
Hugh Durrant-Whyte, Fellow, IEEE, and Tim Bailey

Simultaneous Localisation and Mapping (SLAM): Part II State of the Art
Tim Bailey and Hugh Durrant-Whyte

Basics on estimation

“Estimation is the process by which we infer the value of a quantity of interest, $x$, by processing data that is in some way dependent on $x$."

$$Z^k = \{z_1, z_2 \cdots z_k\}$$

Data $\rightarrow$ Estimation Engine $\rightarrow$ Estimate

Prior Beliefs $\rightarrow p(x_0)$
Basics on estimation

Maximum A Posteriori estimation:

Sensor model $p(z|x)$: “Given $x$, the probability of the sensor measurement being within 1m is….”

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)}$$

$\propto p(z|x) \times p(x)$

Given an observation $z$, a likelihood function $p(z|x)$ and a prior distribution on $x$, $p(x)$, the maximum a posteriori estimator - MAP finds the value of $x$ which maximises the posterior distribution $p(x|z)$

$$\hat{x}_{map} = \underset{x}{\arg\max} p(z|x)p(x)$$ (1)
Examples of MAP estimation
With normal prior and likelihood

\[ \hat{x}_2 = \hat{x}_1 + K_2 \left( z_2 - \hat{x}_1 \right) \]

\[ K_2 = \frac{\sigma^2}{\sigma_1^2 + \sigma_{z_2}^2} \]

\[ \frac{1}{\sigma_2^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_{z_2}^2} \]
Basics on estimation

Recursive Bayesian estimation

\[ Z^k = \{z_1, z_2, \ldots, z_k\} \]

Sequence of data (measurements)

We want conditional mean (mmse) of x given \( Z^k \)

Can we iteratively calculate this – \textit{i.e.} every time a new measurement comes in, update our estimate?

Key idea: “one man’s posterior is another’s prior”
Basics on estimation

Kalman filter

Recursive process: estimate the system state from uncertain observations (measures) and an uncertain system model.

\[ z = H_i x + w \]

\[ x(k + 1) = Fx(k) + \ldots \]

Actual underlying state: \( x \)
Basics on estimation

Kalman filter: things to know

• A recursive process
• Asynchronous
• Prediction / update structure
• Prediction increases covariances
• Update decreases covariances
• Essential importance of correlations
Mapping and Localisation

• With perfect localisation:

  Given the robot position \( x_v \), and a sensor model \( p(Z^k|M, x_v) \)

  compute the map \( M: p(M|x_v, Z^k) \)

  \( \rightarrow \) Mapping problem « solved » (proper management of the sensors uncertainties)

• With a perfect known map

  Given a feature map \( M \), and a sensor model \( p(Z^k|M, x_v) \)

  compute the robot position \( x_v: p(x_v|M, Z^k) \)

  \( \rightarrow \) Localisation problem « solved » (proper matching of sensor data and the map)
• Without perfect localisation, nor perfect map?
  ⇒ Simultaneous localisation and mapping (SLAM)

Given the robot controls $u_k$ and the sensor readings $z_k$, compute the map $M$ and the robot position $x_v$: $p(M, x_v | u_k, z_k)$
SLAM: outline

• Simultaneous localisation and mapping
  • Problem presentation
  • Basics on estimation
  • EKF SLAM
SLAM: EKF-based solution
SLAM: EKF-based solution

Importance of the correlations

\[ P = \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix} \]
SLAM: EKF-based solution

Importance of the correlations

\[ P = \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix} \]
SLAM: EKF-based solution

Importance of the correlations

\[ P = \begin{bmatrix} P_{vv} & P_{vm} \\ P_{vm}^T & P_{mm} \end{bmatrix} \]
Kalman filter for SLAM

Classic implementation (e.g. vehicle position tracking)

System state: \( x(k) \), with variance \( P_x \)

System model:
\[
x(k + 1) = f(x(k), u(k + 1)) + v(k + 1)
\]

Observation model:
\[
z(k) = h(x(k)) + w(k)
\]

Prediction (static features):

SLAM implementation

System state:
\( x(k) = [x_p, m_1, ..., m_N] \), with \( x_p = [\phi, \theta, \psi, t_x, t_y, t_z] \)
and \( m_i = [x_i, y_i, z_i] \)

\[
P(k) = \begin{bmatrix} P_{pp}(k) & P_{pm}(k) \\ P_{pm}(k) & P_{mm}(k) \end{bmatrix}
\]

System model:
\[
x(k + 1) = f(x(k), u(k + 1)) + v(k + 1)
\]

Observation model:
\[
z(k) = h(x(k)) + w(k)
\]

\[
\begin{bmatrix}
x_v(k + 1) \\
x_{f,1}(k) \\
\vdots \\
x_{f,n}(k)
\end{bmatrix} = \begin{bmatrix}
x_v(k) \oplus u(k) \\
x_{f,1}(k) \\
\vdots \\
x_{f,n}(k)
\end{bmatrix}
\]
Kalman filter for SLAM

When exploring new areas, uncertainty grows
Kalman filter for SLAM

When revisiting known areas, uncertainty decreases and the whole map is re-estimated
SLAM: outline

- Simultaneous localisation and mapping
  - Problem presentation
  - Basics on estimation
  - EKF SLAM
  - Main issues
EKF-SLAM Issues

1. Algorithmic complexity: $O(n^2)$
2. Non-linearities yield inconsistency

Inconsistent!

Standard data association cannot close a 250m loop

Computational complexity was not a problem here!
3. Data associations

Within the estimation framework:

- Measurements
- Predicted features

Within the estimation framework?

- Measurements
- Predicted features
SLAM: other estimation approaches

Stochastic approaches:

- Kalman / information
- Ensemblist
- Particles

Global minimization approaches (e.g. bundle adjustment, scan matching)
SLAM: outline

• Simultaneous localisation and mapping
  • Problem presentation
  • Basics on estimation
  • EKF SLAM
  • Main issues

• Vision-based SLAM
Beyond estimation

Functions required by any SLAM implementation:

- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions
- A perception process
- Control, signal processing...
- A perception process
- An estimation process
Use vision!

• Cameras: low cost, light and power-saving

• Perceive data
  – In a volume
  – Very far
  – Very precisely

• Loss of depth is (almost) not anymore a difficulty

• Stereovision
  – 2 cameras provide depth

• Images carry a vast amount of information

• A vast know-how exists in the computer vision community
SLAM : what kind of landmarks ?

A good landmark:
- Should be *discriminant* (easy to associate)
- Should be invariant wrt. the viewpoint
- Its position (or part of it) should be observable — with an associated *error model*

Some point features have these properties:
- Harris features
- SIFT features
Stereovision SLAM

- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions

→ Vision : interest points
→ Stereovision
→ Visual motion estimation
→ Interest points matching
→ Extended Kalman filter
Landmark observation : stereovision

Dense stereovision

or

IP matching applied on stereo frames (even easier)
Visual odometry: principle
Stereovision SLAM

- Landmark detection → Vision : interest points ← OK
- Relative observations (measures)
  - Of the landmark positions → Stereovision ← OK
  - Of the robot motions → Visual motion estimation ← OK
- Observation associations → Interest points matching ← OK
- Refinement of the landmark and robot positions → Extended Kalman filter ← OK
On a ground rover

- 110 stereo pairs processed, 60m loop

landmark uncertainty ellipses (x5)
On a ground rover

- 110 stereo pairs processed, 60m loop

<table>
<thead>
<tr>
<th></th>
<th>Frame 1/100 Reference</th>
<th>Reference Std. Dev.</th>
<th>VME result</th>
<th>VME Abs. error</th>
<th>SLAM result</th>
<th>SLAM Std. Dev.</th>
<th>SLAM Abs. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Theta$</td>
<td>0.52°</td>
<td>0.31°</td>
<td>2.75°</td>
<td>2.23°</td>
<td>0.88°</td>
<td>0.98°</td>
<td>0.36°</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>0.36°</td>
<td>0.25°</td>
<td>-0.11°</td>
<td>0.47°</td>
<td>0.72°</td>
<td>0.74°</td>
<td>0.36°</td>
</tr>
<tr>
<td>$\psi$</td>
<td>-0.14°</td>
<td>0.16°</td>
<td>1.89°</td>
<td>2.03°</td>
<td>1.24°</td>
<td>1.84°</td>
<td>1.38°</td>
</tr>
<tr>
<td>$t_x$</td>
<td>-0.012 m</td>
<td>0.010 m</td>
<td>0.057 m</td>
<td>0.069 m</td>
<td>-0.077 m</td>
<td>0.069 m</td>
<td>0.065 m</td>
</tr>
<tr>
<td>$t_y$</td>
<td>-0.243 m</td>
<td>0.019 m</td>
<td>-1.018 m</td>
<td>0.775 m</td>
<td>-0.284 m</td>
<td>0.064 m</td>
<td>0.041 m</td>
</tr>
<tr>
<td>$t_z$</td>
<td>0.019 m</td>
<td>0.015 m</td>
<td>0.144 m</td>
<td>0.125 m</td>
<td>0.018 m</td>
<td>0.019 m</td>
<td>0.001 m</td>
</tr>
</tbody>
</table>
In indoor environments

- About 30 m long trajectory, 1300 stereo image pairs
In indoor environments

- About 30 m long trajectory, 1300 stereo image pairs

Two rotation angles (\(\text{Phi}, \text{Theta}\)) and \textbf{Elevation} must be zero

![Graphs showing \(\text{Phi}\), \(\text{Theta}\), and \text{Elevation}](image-url)
With a blimp
SLAM: outline

- Simultaneous localisation and mapping
  - Problem presentation
  - Basics on estimation
  - EKF SLAM
  - Main issues

- Vision-based SLAM
  - SLAM with stereovision
  - SLAM with monocular imagery
SLAM with monocular vision

Micro UAVs

Smartphones, wearable devices
SLAM with monocular vision

Prediction: any information on the robot motion

Landmarks:
- interest points = 3D (x,y,z) points

Application of the « usual » SLAM steps:
- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Estimation: refinement of the landmark and robot positions

But why is it particularly difficult?

Partial observations!
(« bearings-only »)
Bearing-only SLAM

Generic SLAM

- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions

Stereovision SLAM

→ Vision: interest points
   → Stereovision
   → Visual motion estimation
   → Interest points matching
   → Extended Kalman filter
Bearing-only SLAM

Generic SLAM
- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions

Monocular SLAM
- Vision: interest points
- « Multi-view stereovision »
- INS, Motion model...
- Interest points matching
- Extended Kalman filter
(Particle filtering)

0. Initialization
1. Prediction
2. Observations + weighting (resampling)

1. Prediction
2. Observations

(time)
Bearing-only SLAM: landmark initialisation

« Initialisation filter » ≈ particle filter

One interest point matched in the image
Bearing-only SLAM: landmark initialisation

« Initialisation filter » ≈ particle filter

Issues with this approach:
- Complexity
- Some landmarks remain non-observable
- Numerous observations are lost
Landmark initialisation

« Initialisation », 2\textsuperscript{nd} approach: inverse depth parametrization

Solution established in 2006 [J. Montiel / A. Davison]:

- *Direct* initialization of a point the first time it is perceived
- Allows to consider points located at the infinite

\[ p_{3D} = \vec{a} + \frac{\vec{u}}{\rho} \]

- Every detected landmark is directly and \textit{immediately} observable

One interest point in the image

One landmark in the MAP
A closer look at the motion prediction

Constant velocity model

Robot state: \( \mathcal{R} = (p \ q \ v \ w)^T \)

Prediction: \( p^+ = p + v \cdot dt \quad q^+ = q \cdot w \cdot dt \quad v^+ = v \quad w^+ = w \)

Allows to focus the data association step:

1. Start with a descriptor of the landmark (e.g. appearance at first observation)
2. Apply affine transformation to predict the current appearance
3. Search the landmark inside the observation uncertainty ellipse
A closer look at the motion prediction

Constant velocity model

Robot state: \( \mathcal{R} = (p \ q \ v \ w)^T \)

Prediction: \( p^+ = p + v.dt \quad q^+ = q \times w.dt \quad v^+ = v \quad w^+ = w \)

- simple (no additional sensor required)
- not precise
  - bad linearization points
  - large search areas
  - difficulty to track very high motion dynamics
- no scale factor estimate possible
A closer look at the motion prediction

Constant acceleration model

Robot state: \( \mathcal{R} = (p \ q \ v \ w \ v_a \ w_a)^T \)

Prediction:

\[
\begin{align*}
\mathbf{v}^+ &= \mathbf{v} + \mathbf{v}_a \cdot dt \\
\mathbf{w}^+ &= \mathbf{w} + \mathbf{w}_a \cdot dt \\
\mathbf{v}_a^+ &= \mathbf{v}_a \\
\mathbf{w}_a^+ &= \mathbf{w}_a
\end{align*}
\]

- simple (no additional sensor required)
- not very precise
  - bad linearization points
  - large search areas
- no scale factor estimate possible
Using an inertial measurement unit

Robot state: $\mathcal{R} = (p \ q \ v \ a_b \ w_b \ g)^T$

Prediction:

\[
\begin{align*}
    p^+ &= p + v \cdot dt \\
    q^+ &= q \otimes \exp(2q((w_m - w_b) \cdot dt)) \\
    v^+ &= v + (q2R(q) \cdot (a_m - a_b) + g) \cdot dt \\
    a_b^+ &= a_b \\
    w_b^+ &= w_b \\
    g^+ &= g
\end{align*}
\]

- additional sensor required
- precise
  - good linearization points
  - small search areas
  - possibility to track very high motion dynamics
- scale factor estimate possible
A closer look at the motion prediction

Using an inertial measurement unit: results
A closer look at the motion prediction

Using an inertial measurement unit: results

Estimated trajectory of the 6 position parameters
A closer look at the motion prediction

Using an inertial measurement unit: results
A closer look at the motion prediction

Using an inertial measurement unit: results
A closer look at the motion prediction

Using an inertial measurement unit: summary

• An additional proprioceptive sensor can help if:
  • high frequency
  • not too noisy
• Can also be directly used in the prediction step if:
  • complete
  • never faulty
SLAM: outline

- Basics of simultaneous localisation and mapping
- SLAM with monocular vision
  - Solutions to the landmark initialisation problem
  - Importance of the motion prediction, introduction of an IMU
  - What about loop-closing?
A closer look at data association

Data association relying on geometric information can become tricky…

Blue: predicted (mapped) features
Green: observed features

2 nearly consecutive cases
→ Better use landmark matching techniques

Large loop-closing case
Overall summary

• Autonomous robots, on the importance of localization
• Localization using dead-reckoning approaches

• SLAM
  • Problem presentation
  • Basics on estimation
  • EKF SLAM
  • Main issues

• Vision-based SLAM
  • SLAM with stereovision
  • SLAM with monocular imagery
    - Solutions to the landmark initialisation problem
    - Illustrations
    - Importance of the motion prediction, introduction of an IMU
    - (Visual loop-closing detection)