Use a Single Camera for Simultaneous Localization And Mapping with Mobile Objects Tracking in Dynamic Environments

Davide Migliore, Daniele Marzorati, Roberto Rigamonti, Matteo Matteucci, Domenico G. Sorrenti

Workshop on Safe navigation in open and dynamic environments. Application to autonomous vehicles

Davide Migliore
migliore@elet.polimi.it
Department of Electronics and Information, Politecnico di Milano, Italy
What is SLAM?
The question:

"Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?"

H. Durrant-Whyte “Simultaneous Localization and Mapping” - RAS Magazine 2006
What is SLAM?

The question:

"Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?"

H. Durrant-Whyte "Simultaneous Localization and Mapping" RAS Magazine 2006
What is SLAM?

The question:

“Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?”

H. Durrant-Whyte “Simultaneous Localization and Mapping” - RAS Magazine 2006

\[ p(x_k, M|z_0, z_1, \ldots, z_k, u_1, u_2, \ldots, u_k). \]

\[ p(x_k, M|z_k, u_k) \propto p(z_k|x_k, M) \int p(x_k|x_{k-1}, u_k) p(x_{k-1}, M|z_{k-1}, u_{k-1}) dx_{k-1} \]

One of the most important results obtained by the Robotics community

- Implemented in a number of different domains
- Using both parametric (EKF, UKF, EIF...) and non-parametric (FastSLAM...) approaches
- Solved using different kind of sensors (Laser, Sonar, Cameras...)
- Considering 3DoF or 6DoF
Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible
Spostamento ignoto
Why is this challenging?

- Rumore --> incertezza
- Non modellizzabile con Gauss
- EKF - unfeasible
- Spostamento ignoto
Why is this challenging?

- Rumore --> incertezza
- Non modellizzabile con Gauss
- EKF - unfeasible
- Spostamento ignoto
Why is this challenging?

- Rumore --> incertezza
- Non modellizzabile con Gauss
- EKF - unfeasible
- Spostamento ignoto
Why is this challenging?

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible
Spostamento ignoto
Why is this challenging?

- Rumore --> incertezza
- Non modellizzabile con Gauss
- EKF - unfeasible
- Spostamento ignoto
Why is this challenging?

Solutions:
- S. Soatto et al “Structure from motion casually integrated over time” - IEEE PAMI 2002
- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti “On the Use of Inverse Scaling in Monocular SLAM” - ICRA 2009 (Friday - 11:10 - room 401)
Why is this challenging?

Solutions:
- S. Soatto et al “Structure from motion casually integrated over time” - IEEE PAMI 2002
- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti “On the Use of Inverse Scaling in Monocular SLAM” - ICRA 2009 (Friday - 11:10 - room 401)
Inverse Scaling Parametrization
Inverse Scaling Parametrization

Idea:
Inverse Scaling Parametrization

Idea:

- Camera Center
- Image Plane
- Viewing Ray
- f
- X1
- X2
- X
Inverse Scaling Parametrization

Idea:

Camera Center

Image Plane

Viewing Ray

Friday, 25 September 2009
Inverse Scaling Parametrization

Idea:

\[
\mathbf{x} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \alpha_1 \mathbf{x}_1 = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 \mathbf{x}_2 = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix}
\]
Inverse Scaling Parametrization

Idea:

\[ \mathbf{x} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \alpha_1 \mathbf{x}_1 = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 \mathbf{x}_2 = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix} \]

\[ \mathbf{x} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X' \\ Y' \\ Z' \\ 1/\alpha_1 \end{pmatrix} \equiv \begin{pmatrix} X'' \\ Y'' \\ Z'' \\ 1/\alpha_2 \end{pmatrix}. \]
Inverse Scaling Parametrization

Idea:

\[
\begin{align*}
X &= \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \alpha_1 X_1 = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 X_2 = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix}, \\
X &= \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X' \\ Y' \\ Z' \\ 1/\alpha_1 \end{pmatrix} \equiv \begin{pmatrix} X'' \\ Y'' \\ Z'' \\ 1/\alpha_2 \end{pmatrix}.
\end{align*}
\]

Undelayed Initialization

\[
X = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} u \\ v \\ f \\ 1/\alpha \end{pmatrix} \equiv \begin{pmatrix} \omega \end{pmatrix},
\]

- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti et al.
“Monocular SLAM with Inverse Scaling Parametrization” - BMVC 2008
MonoSLAM with Inverse Scaling
MonoSLAM with Inverse Scaling

Extended Kalman Filter

Video Frame

\[ x_k = \begin{bmatrix} x^W_{C_k} & v^C_k & x^W_{F_1 k} & \ldots & x^W_{F_n k} & \ldots & x^W_{F_N k} \end{bmatrix}^T \]
MonoSLAM with Inverse Scaling

- Extended Kalman Filter

\[ x_k = \begin{bmatrix} x_{C_k}^W & v_k^C \end{bmatrix} \]
Extended Kalman Filter

\[ x_k = \begin{bmatrix} x_{C_k}^W & v_{C_k}^C & x_{F_1k}^W & \ldots & x_{F_{n,k}}^W & \ldots & x_{F_{N,k}}^W \end{bmatrix}^T \]
MonoSLAM with Inverse Scaling

Extended Kalman Filter

\[
\hat{x}_k = \left[ \begin{array}{c} x^W_{C_k} \\ v^C_k \\ x^W_{F_{1,k}} \\ \vdots \\ x^W_{F_{n,k}} \\ \vdots \\ x^W_{F_{N,k}} \end{array} \right]^T
\]

\[
\hat{x}_k = \left[ \begin{array}{c} x^W_{C_{k-1}} \\ v^C_k \\ x^W_{F_{1,k-1}} \\ \vdots \\ x^W_{F_{N,k-1}} \end{array} \right]
\]

\[
\hat{P}_k = J_1 P_{k-1} J_1^T + J_2 Q J_2^T
\]

\[
J_1 = \left[ \begin{array}{cccc} J_x & J_y & \cdots & J_{F_n} \end{array} \right], \quad J_2 = \left[ \begin{array}{c} J_{a_k} \end{array} \right]
\]
MonoSLAM with Inverse Scaling

Extended Kalman Filter

\[ \hat{x}_k = \left[ \begin{array}{c} x_{C_k}^W \oplus x_{C_{k-1}}^W \\ v_k^C \\ x_{F_1 k}^W \\ \vdots \\ x_{F_n k}^W \oplus x_{F_{n-1} k}^W \\ \end{array} \right]^T \]

\[ \hat{x}_k = \left[ \begin{array}{c} x_{C_k}^W \\ v_k^C \\ x_{F_1 k-1}^W \\ \vdots \\ x_{F_{n-1} k-1}^W \end{array} \right] \]

\[ \hat{P}_k = J_1 P_{k-1} J_1^T + J_2 Q J_2^T \]

\[ J_1 = \left[ \begin{array}{c} J_x \\ J_v \\ \vdots \\ J_{F_n} \end{array} \right], \quad J_2 = \left[ \begin{array}{c} J_{a_k} \end{array} \right] \]
MonoSLAM with Inverse Scaling

- Extended Kalman Filter

\[ x_k = \begin{bmatrix} x^W_{C_k} & v^C_{k} & x^W_{F_1 k} & \ldots & x^W_{F_n k} & \ldots & x^W_{F_N k} \end{bmatrix}^T \]

\[ h^C_{k,n} = M \left( R^C_{W} \left( \begin{bmatrix} x^W_{F_n} \\ y^W_{F_n} \\ z^W_{F_n} \end{bmatrix} - \omega^W_{F_n} x^C_{k} \right) \right) h_{k,n} = \begin{bmatrix} h^C_{x,n} \\ h^C_{y,n} \\ h^C_{z,n} \end{bmatrix}. \]

\[ S = H_k \hat{P}_k H_k^T + W_k R_k W_k^T \]

\[ K = \hat{P}_k H_k^T S^{-1} \]

\[ P_k = \hat{P}_k - KSK^T \]

\[ x_k = \hat{x}_k + K (z_k - h_k) \]
MonoSLAM with Inverse Scaling

Extended Kalman Filter

\[
x_k = \begin{bmatrix} x^W_{C_k} & v^C_k & x^W_{F_1} & \ldots & x^W_{F_n} & \ldots & x^W_{F_N} \end{bmatrix}^T
\]

\[
h^C_{k,n} = M C_k \left( \begin{bmatrix} x^W_{F_n} \\ y^W_{F_n} \\ z^W_{F_n} \end{bmatrix} - \omega^W_{F_n} x^C_{C_k} \right)
\]

\[
S = H_k \hat{P}_k H_k^T + W_k R_k W_k^T
\]

\[
K = \hat{P}_k H_k^T S^{-1}
\]

\[
P_k = \hat{P}_k - K S K^T
\]

\[
x_k = \hat{x}_k + K (z_k - h_k)
\]
Experimental Results
Experimental Results

- Real Dataset

- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti “On the Use of Inverse Scaling in Monocular SLAM” - ICRA 2009 (Friday - 11:10 - room 401)
SLAM in Dynamic Environments
Open issue:

- Consistency of the estimates in scene containing moving objects
Open issue:
- Consistency of the estimates in scene containing moving objects
SLAM in Dynamic Environments

Open issue:
- Consistency of the estimates in scene containing moving objects
Open issue:
- Consistency of the estimates in scene containing moving objects
SLAM in Dynamic Environments

SLAM

\[
p(x_k, M|Z_k, U_k) \propto p(z_k|x_k, M) \int p(x_k|x_{k-1}, u_k) p(x_{k-1}, M|Z_{k-1}, U_{k-1}) \, dx_{k-1}
\]

Posterior at time \( k \)

Update

Posterior at time \( k-1 \)

Prediction
SLAM in Dynamic Environments

SLAM and DATMO

\[ p(x_k, M | Z_k, U_k) \propto p(z_k | x_k, M) \int p(x_k | x_{k-1}, u_k) p(x_{k-1}, M | Z_{k-1}, U_{k-1}) dx_{k-1} \]

Posterior at time \( k \)  
Update

\[ p(x_k, M, O_k | Z_k, U_k) \propto p(z_k | O_k, x_k) \int p(O_k | O_{k-1}) p(O_{k-1} | Z_{k-1}, U_{k-1}) dO_{k-1} \]

Update

\[ p(z_k | x_k, M) \int p(x_k | x_{k-1}, u_k) p(x_{k-1}, M | Z_{k-1}, U_{k-1}) dx_{k-1} \]

Update

Posterior at time \( k-1 \)
Prediction

MonoSLAM with BOT

- Extended Kalman Filter
MonoSLAM with BOT

- Extended Kalman Filter

![Diagram of Extended Kalman Filter]

- Feature Detection (FD)
- Data Association (DA)
- Feature Initialization
- Prediction
- Update

Video Frame

Friday, 25 September 2009
**MonoSLAM with BOT**

- Extended Kalman Filter

---

**Extended Kalman Filter Diagram:**

- **FD (Feature Detection)**
  - **Feature Initialization**
  - **Data Association**
  - **Classification**
  - **Prediction**
  - **Update**

- **SLAM Filter**
  - **Data Association**
  - **Prediction**
  - **Update**

- **BOT Filter**

---

**Video Frame**
Bearing Only Tracking

- Feature Initialization
- Prediction
- Update

BOT Filter
Bearing Only Tracking

- Again Inverse Scaling Parametrization
  - EKF BOT (Robocentric):
    - **State**
      \[
      x_k = \begin{bmatrix}
        x_{Ck}^F \\
        F_k \\
        v_{F_k}^F
      \end{bmatrix}
      \]
    - **Motion Model**
      \[
      x_{k+1} = \begin{bmatrix}
        x_{C_{k+1}}^F \\
        F_{k+1} \\
        v_{F_{k+1}}^F
      \end{bmatrix} = \begin{bmatrix}
        x_{C_k}^F \oplus x_{C_k}^F \oplus (v_{F_k}^F F_{k+1} \Delta t) \\
        x_{F_k}^F \oplus v_{F_k}^F
      \end{bmatrix}
      \]
    - **Measurement Model**
      \[
      h_k = \begin{bmatrix}
        h_{k_x} \\
        h_{k_y} \\
        h_{k_z}
      \end{bmatrix} = M x_{C_k}^F \\
      M = \begin{bmatrix}
        f_{cx} & 0 & cc_{cx} \\
        0 & f_{cy} & cc_{cy} \\
        0 & 0 & 1
      \end{bmatrix}
      \]
      \[
      h_k = \begin{bmatrix}
        h_{k_x} / h_{k_z} \\
        h_{k_y} / h_{k_z}
      \end{bmatrix}
      \]
Bearing Only Tracker Observability
Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)

\[ F_k = \begin{bmatrix} R & R_{C_{k+1}} \\ T_{C_{k+1}} & R_{C_{k+1}} \end{bmatrix}, \]
Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)
  \[ \mathbf{F}_k = \begin{bmatrix} \mathbf{R}_{C_{k+1}}^{C_k} & \mathbf{R}_{C_{k+1}}^{C_k} \mathbf{dt} \\ \mathbf{0} & \mathbf{R}_{C_{k+1}}^{C_k} \end{bmatrix}, \]

- Non linear measurement model... but
  \[ \begin{bmatrix} h_{kx} \\ h_{ky} \\ h_{kz} \end{bmatrix} = \mathbf{Ix}_{F_k}^{C_k} \quad \mathbf{h}_k = \begin{bmatrix} h_{ku} \\ h_{kv} \end{bmatrix} \begin{bmatrix} h_{kx} / h_{kz} \\ h_{ky} / h_{kz} \end{bmatrix} \quad 0 = h_{kx} - h_{ku} h_{kz}. \]
  \[ 0 = h_{ky} - h_{kv} h_{kz}. \]
  \[ \mathbf{H}_k = \begin{bmatrix} 1 & 0 & -h_{ku} & 0 & 0 \\ 0 & 1 & -h_{kv} & 0 & 0 \end{bmatrix}. \]
Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)

\[
F_k = \begin{bmatrix}
RT_{C_{k+1}}^{C_k} & R_{C_{k+1}}^{C_k} dt \\
0 & R_{C_{k+1}}^{C_k}
\end{bmatrix},
\]

- Non linear measurement model... but

\[
\begin{bmatrix}
h_{kx} \\
h_{ky} \\
h_{kz}
\end{bmatrix} = \mathbf{I} x_{F_k}^{C_k} \quad h_k = \begin{bmatrix}
h_{ku} \\
h_{kv}
\end{bmatrix} \begin{bmatrix}
h_{kx}/h_{kz} \\
h_{ky}/h_{kz}
\end{bmatrix} \quad 0 = h_{kx} - h_{ku} h_{kz}, \quad 0 = h_{ky} - h_{kv} h_{kz}.
\]

\[
H_k = \begin{bmatrix}
1 & 0 & -h_{ku} & 0 & 0 \\
0 & 1 & -h_{kv} & 0 & 0
\end{bmatrix}.
\]

- How to check the observability

\[
\begin{cases}
  z_0 = H_0 X_0 \\
  z_1 = H_1 F_1 X_0 \\
  z_2 = H_2 F_2 F_1 X_0 \\
  \vdots \\
  z_k = H_k F_k \ldots F_2 F_1 X_0
\end{cases}
\]

\[
O_k = \begin{bmatrix}
H_0 \\
H_1 F_1 \\
H_2 F_2 F_1 \\
\vdots \\
H_k F_k \ldots F_2 F_1
\end{bmatrix}
\]

- Under the assumption of random camera movements (hand shacking) and noisy measurements the observability is possible
Bearing Only Tracker Observability
Bearing Only Tracker Observability

- Simulated Results (camera position is given)
Bearing Only Tracker Observability

- Simulated Results (camera position is given)
Moving Features Classification

- We need to identify moving features
  - We need at least two viewing rays!
We need to identify moving features
- We need at least two viewing rays!

- The idea:
  - Save a first viewing ray and move
  - Save a second viewing ray (after a while) and move
  - If the intersections of the third viewing ray with the others are distinct, then the feature is moving
Moving Features Classification

- We need to identify moving features
  - We need at least two viewing rays!

- The idea:
  - Save a first viewing ray and move
  - Save a second viewing ray (after a while) and move
  - If the intersections of the third viewing ray with the others are distinct, then the feature is moving
Uncertain Projective Geometry
Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
Uncertain Projective Geometry

Problem: we have to take into account the uncertainties
Solution: use the Uncertain Geometry Reasoning

- Construction mechanism:
  - $O(\cdot)$ (for 3D lines)
  - $\Pi(\cdot)$ (for 3D points and 3D planes)

\[
\Pi(X) = \frac{\partial X \wedge \partial Y}{\partial Y} = \begin{pmatrix}
W_1 & 0 & 0 & X_1 \\
0 & W_1 & 0 & -Y_1 \\
0 & 0 & W_1 & -Z_1 \\
0 & -Z_1 & Y_1 & 0 \\
Z_1 & 0 & -X_1 & 0 \\
-Y_1 & X_1 & 0 & 0 \\
\end{pmatrix}
\]

\[
O(L) = \frac{\partial X \wedge \partial L}{\partial X} = \begin{pmatrix}
0 & L_3 & -L_2 & -L_4 \\
-L_3 & 0 & L_1 & -L_5 \\
L_2 & -L_1 & 0 & -L_6 \\
L_4 & L_5 & L_6 & 0 \\
\end{pmatrix}
\]

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $O(\cdot)$ (for 3D lines)
    - $\Pi(\cdot)$ (for 3D points and 3D planes)
Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $O(\cdot)$ (for 3D lines)
    - $\Pi(\cdot)$ (for 3D points and 3D planes)
  - A bilinear expression to compute relationship and propagate their uncertainties

\[
\begin{align*}
  z &= f(x, y) = O(x)y = \Pi(y)x \quad (x \text{ line, } y \text{ point and } f \text{ intersection}) \\
  (z, \Sigma_{zz}) &= (O(x)y, O(x)\Sigma_{yy}O^T(x) + \Pi(y)\Sigma_{xx}\Pi^T(y)) .
\end{align*}
\]
Uncertain Projective Geometry

Problem: we have to take into account the uncertainties

Solution: use the Uncertain Geometry Reasoning

- Construction mechanism:
  - $O(\cdot)$ (for 3D lines)
  - $\Pi(\cdot)$ (for 3D points and 3D planes)

- A bilinear expression to compute relationship and propagate their uncertainties

\[
    z = f(x, y) = O(x)y = \Pi(y)x \quad (x \text{ line}, y \text{ point and } f \text{ intersection})
\]

\[
    (z, \Sigma_{zz}) = (O(x)y, O(x)\Sigma_{yy}O^T(x) + \Pi(y)\Sigma_{xx}\Pi^T(y))
\]

- We can express the previous test as a probabilistic one to verify the intersection between point and line

\[
    R(x, y) \iff d = O(x)y = \Pi(y)x = 0 \quad \Sigma_d = O(x)\Sigma_xO(x)^T + \Pi(y)\Sigma_y\Pi(y)^T
\]

  • Statistical Test (Chi-square test):

\[
    T = d^T\Sigma_d^{-1}d
\]
Experimental Results
Real Dataset

- **MonoSLAM**
- **MonoSLAMBOT**
Experimental Results

Real Dataset
Conclusions
Conclusions

Principal results achieved:
- The Inverse Scaling Parametrization (ISP) for Bearing Only Tracking
- Integration of SLAM and Moving Feature Tracking
Conclusions

- Principal results achieved:
  - The Inverse Scaling Parametrisation (ISP) for Bearing Only Tracking
  - Integration of SLAM and Moving Feature Tracking

- Ongoing Works (airwiki.elet.polimi.it):
  - Accurate Stability Analysis of BOT Filter
  - Large Maps (CI-SLAMBOT - submitted to IROS 2009)
  - Improve Data association
  - StereoSLAM, BiCamSLAM, OmniSLAM
  - Sensor Fusion: IMU, GPS, Sick Lasers...
  - Model based objects tracking
  - Test with http://www.rawseeds.org/ dataset
Conclusions

- Principal results achieved:
  - The Inverse Scaling Parametrization (ISP) for Bearing Only Tracking
  - Integration of SLAM and Moving Feature Tracking

- Ongoing Works (airwiki.elet.polimi.it):
  - Accurate Stability Analysis of BOT Filter
  - Large Maps (CI-SLAMBOT - submitted to IROS 2009)
  - Improve Data association
  - StereoSLAM, BiCamSLAM, OmniSLAM
  - Sensor Fusion: IMU, GPS, Sick Lasers...
  - Model based objects tracking
  - Test with http://www.rawseeds.org/ dataset
  - Autonomous vehicle navigation
Thanks for your attention

Questions?!?