Tracking of Image Primitives and Correspondence Problem in Navigation and 3D Reconstruction

Darius Burschka

Some slides based on MASK book
Moving camera in an unknown environment (SLAM)

Estimate from changes in the image position
- camera motion
- structure of the environment
Given an image point in left image, what is the (corresponding) point in the right image, which is the projection of the same 3-D point
Matching - Correspondence

Lambertian assumption

\[ I_1(x_1) = R(p) = I_2(x_2) \]

Rigid body motion

\[ x_2 = h(x_1) = \frac{1}{\lambda_2(X)}(R\lambda_1(X)x_1 + T) \]

Correspondence

\[ I_1(x_1) = I_2(h(x_1)) \]

Some slides based on MASK book
Local Deformation Models

- Translational model
  \[ h(x) = x + d \]
  \[ I_1(x_1) = I_2(h(x_1)) \]

- Affine model
  \[ h(x) = Ax + d \]
  \[ I_1(x_1) = I_2(h(x_1)) \]

- Transformation of the intensity values and occlusions
  \[ I_1(x_1) = f_o(X, g)I_2(h(x_1)) \]
Feature Tracking and Optical Flow

- Translational model

\[ I_1(x_1) = I_2(x_1 + \Delta x) \]

- Small baseline

\[ I(x(t), t) = I(x(t) + u dt, t + dt) \]

- Approx. by first two terms of Taylor series

\[ \nabla I(x(t), t)^T u + I_t(x(t), t) = 0 \]

- Brightness constancy constraint

Some slides based on MASK book
Aperture Problem

- Normal flow

\[ \mathbf{u}_n \doteq \frac{\nabla I^T \mathbf{u}}{\| \nabla I \|} \cdot \frac{\nabla I}{\| \nabla I \|} = -\frac{I_t}{\| \nabla I \|} \cdot \frac{\nabla I}{\| \nabla I \|} \]
Some slides based on MASK book

Optical Flow

- Integrate around over image patch

\[ E_b(u) = \sum_{W(x,y)} [\nabla I^T(x, y, t)u(x, y) + I_t(x, y, t)]^2 \]

- Solve

\[ \nabla E_b(u) = 2 \sum_{W(x,y)} \nabla I(\nabla I^T u + I_t) \]

\[ = 2 \sum_{W(x,y)} \left( \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} u + \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \right) \]

\[ \begin{bmatrix} \sum I_x^2 \\ \sum I_x I_y \end{bmatrix} u + \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} = 0 \]

\[ Gu + b = 0 \]
Optical Flow, Feature Tracking

\[ u = -G^{-1}b \]

\[ G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \]

Conceptually:

- rank(G) = 0  blank wall problem
- rank(G) = 1  aperture problem
- rank(G) = 2  enough texture – good feature candidates

In reality:  choice of threshold is involved

Some slides based on MASK book
Feature Tracking

Some slides based on MASK book
Some slides based on MASK book
Point Feature Extraction

\[ G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \]

- Compute eigenvalues of \( G \)
- If smallest eigenvalue \( \sigma \) of \( G \) is bigger than \( \tau \) - mark pixel as candidate feature point

- Alternatively feature quality function (Harris Corner Detector)

\[ C(G) = \text{det}(G) + k \cdot \text{trace}^2(G) \]

Some slides based on MASK book
Harris Corner Detector - Example

Some slides based on MASK book
Some slides based on MASK book
Adaptive and Generic Accelerated Segment Test (AGAST)

• Improvements compared to FAST:
  • full exploration of the configuration space by backward-induction (no learning)
  • binary decision tree (not ternary)
  • computation of the actual probability and processing costs (no greedy algorithm)
  • automatic scene adaption by tree switching (at no cost)
  • various corner pattern sizes (not just one)
Fusion Camera IMU

- IMU-camera setup
  - 15 Hz camera (120° aperture angle) and 120 Hz IMU

- Experiments
  - 8 runs: moving in front of a checkerboard, using bundle adjustment for pose estimation

- Implemented approaches for comparison
  - UKF-based
  - grey-box approach
Step 5: Fusion of IMU and Camera
Problem Description

- Efficient combination of IMU and camera measurements
  - IMU based feature propagation
  - Kalman filter based fusion
- Difficulties in Kalman filter based fusion
  - measurement and system model are not consistent $\rightarrow$ pseudo-measurement model
  - lack of scale in visual odometry $\rightarrow$ keyframe based scale estimation by state augmentation
  - loss of features over time $\rightarrow$ high frame-rates
Feature Propagation

- Two motion prediction concepts
  - 2D feature propagation by motion derivatives
  - IMU-based feature prediction
- Combination of both:
  - translation propagation by feature velocity (2D)
  - rotation propagation by gyroscopes
Feature Propagation

• Two motion prediction concepts
  • 2D feature propagation by motion derivatives
  • IMU-based feature prediction

• Combination of both:
  • translation propagation by feature velocity (2D)
  • rotation propagation by gyroscopes
Data Fusion Example

- Step 1 in the time update phase is merely our prediction based upon the linear state update equation that we have:

\[
\tilde{x}_{k+1}^- = A\tilde{x}_k + Bu_k
\]

- Step 2 of the time update phase comes from projecting our covariance matrix forward where we merely add the process noise variance \( Q \) due to the normal sum distribution property where \( \sigma_3^2 = \sigma_1^2 + \sigma_2^2 \)

\[
P_k = \frac{1}{N} \sum_{i=1}^{N} (\tilde{x}_i - \tilde{\mu}_i)^T (\tilde{x}_i - \tilde{\mu}_i)
\]

\[
P_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (\tilde{x}_{i+1} - \tilde{\mu}_{i+1}) (\tilde{x}_{i+1} - \tilde{\mu}_{i+1})^T + Q
\]

\[
P_{k+1} = \frac{1}{N} \sum_{i=1}^{N} [A(\tilde{x}_i - \tilde{\mu}_i)^T A(\tilde{x}_i - \tilde{\mu}_i)]^T + Q
\]

\[
P_{k+1} = \frac{1}{N} \sum_{i=1}^{N} A(\tilde{x}_i - \tilde{\mu}_i)(\tilde{x}_i - \tilde{\mu}_i)^T A^T + Q
\]

\[
P_{k+1} = AP_k A^T + Q
\]
Data Fusion Example

- OK, let’s say we use code from Team 1 and Team 2 to obtain two different measurements $Z = [z_1, z_2]^T$ for the range $r$ to a beacon.
- Let us further assume that the variance in each of these sensor measurements is $R_1$ and $R_2$, respectively.
- Q: How should we fuse these measurements in order to obtain the “best” possible resulting estimate for $r$?
- We’ll define “best” from a least-squares perspective...
- We have 2 measurements that are equal to $r$ plus some additive zero-mean Gaussian noise $v_1$ and $v_2$.

\[
\begin{align*}
    z_1 &= r + N(0, R_1) = r + v_1 \\
    z_2 &= r + N(0, R_2) = r + v_2
\end{align*}
\]
A Least-Squares Approach

\[ z_1 = r + N(0, R_1) = r + v_1 \]
\[ z_2 = r + N(0, R_2) = r + v_2 \]

- We want to fuse these measurements to obtain a new estimate for the range \( \hat{r} \).
- Using a weighted least-squares approach, the resulting sum of squares error will be
  \[ e = \sum_{i=1}^{n} w_i (\hat{r} - z_i)^2 \]
- Minimizing this error with respect to \( \hat{r} \) yields
  \[ \frac{\partial e}{\partial \hat{r}} = \sum_{i=1}^{n} w_i (\hat{r} - z_i) = 2 \sum_{i=1}^{n} w_i (\hat{r} - z_i) = 0 \]
A Least-Squares Approach

- This can be rewritten as

\[ \hat{r} = z_1 + \frac{R_1}{R_1 + R_2} (z_2 - z_1) \]

or if we think of this as adding a new measurement to our current estimate of the state we would get

\[ \hat{r}_{k+1} = \hat{r}_k^- + \frac{P_{k+1}^-}{P_{k+1}^- + R} (z_{k+1} - \hat{r}_k^-) \quad \Rightarrow \quad \hat{r}_{k+1} = \hat{r}_k^- + K_{k+1} (z_{k+1} - \hat{r}_k^-) \]

- For merging Gaussian distributions, the update rule is

\[ \frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \sigma_2^2} \quad \Rightarrow \quad \sigma_3^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \]

which if we write in our measurement update equation form we get

\[ P_{k+1}^- = \frac{P_{k+1}^- R_{k+1}^-}{P_{k+1}^- + R_{k+1}} \equiv P_{k+1}^- - K_{k+1}^- P_{k+1}^- \]
The Measurement Update Phase

- These are the measurement update equations for the discrete Kalman filter

\[
\begin{align*}
\text{Time Update} & \\
1. \text{Project the state forward} & \quad \bar{x}_k = A\tilde{x} + Bu_k \\
2. \text{Project the covariance forward} & \quad \bar{P}_k = AP_kA^T + Q
\end{align*}
\]

\[
\begin{align*}
\text{Measurement Update} & \\
1. \text{Compute Kalman Gain} & \quad K_k = P_kH(HP_kH^T + R)^{-1} \\
2. \text{Update state estimate with measurement } z_k & \quad \hat{x}_k = \bar{x}_k + K_k(z_k - H\bar{x}_k) \\
3. \text{Update error covariance} & \quad \hat{P}_k = (I - K_kH)P_k
\end{align*}
\]
Feature Propagation

- Two motion prediction concepts
  - 2D feature propagation by motion derivatives
  - IMU-based feature prediction
- Combination of both:
  - translation propagation by feature velocity (2D)
  - rotation propagation by gyroscopes
Region based Similarity Metric

- Sum of squared differences

\[ SSD(h) = \sum_{\tilde{x} \in W(x)} \|I_1(\tilde{x}) - I_2(h(\tilde{x}))\|^2 \]

- Normalize cross-correlation

\[ NCC(h) = \frac{\sum_{W(x)} (I_1(\tilde{x}) - \bar{I}_1)(I_2(h(\tilde{x})) - \bar{I}_2))}{\sqrt{\sum_{W(x)} (I_1(\tilde{x}) - \bar{I}_1)^2 \sum_{W(x)} (I_2(h(\tilde{x})) - \bar{I}_2)^2}} \]

- Sum of absolute differences

\[ SAD(h) = \sum_{\tilde{x} \in W(x)} |I_1(\tilde{x}) - I_2(h(\tilde{x}))| \]
Algorithmic Concept

Outline:

- Feature Tracking
- Pose Estimation
- Error Propagation
- Sensor Registration

Preprocessing and Navigation

Optical Flow Estimation
Obstacle Detection

Image Differences
Visual Homing

Landmarks (Cues) Extraction
Visual Compass

Map Management

Obstacle Avoidance

Optical Flow Error Propagation
Pose Estimation

Location Estimation
Map Error Propagation

Motion Estimation

IMU-Camera Registration

IMU Calibration

Strapdown Calculation
General Formulation

Given two views of the scene, recover the unknown camera displacement and 3D scene structure.
Pinhole Camera Model

- 3D points \( \mathbf{X} = [X, Y, Z, W]^T \in \mathbb{R}^4, \quad (W = 1) \)
- Image points \( \mathbf{x} = [x, y, z]^T \in \mathbb{R}^3, \quad (z = 1) \)
- Perspective Projection \( \lambda \mathbf{x} = \mathbf{X} \)
  \[
  \lambda = Z \quad x = \frac{X}{Z} \quad y = \frac{Y}{Z}
  \]
- Rigid Body Motion \( \mathbf{\Pi} = [R, T] \in \mathbb{R}^{3 \times 4} \)
- Rigid Body Motion + Projective projection
  \( \lambda \mathbf{x} = \mathbf{\Pi} \mathbf{X} = [R, T] \mathbf{X} \)
$\lambda_1 x_1 = X$

$\lambda_2 x_2 = R \lambda_1 x_1 + T$

$\lambda_2 x_2 = R \lambda_1 x_1 + T$
Euclidean transformation

\[ \lambda_2 x_2 = R\lambda_1 x_1 + T \]

Find such Rotation and Translation and Depth that the reprojection error is minimized

\[ \sum_{j=1}^{n} \|x_1^j - \pi(R_1, T_1, X)\|^2 + \|x_2^j - \pi(R_2, T_2, X)\|^2 \]

Two views \~\ 200 points
6 unknowns – Motion 3 Rotation, 3 Translation
- Structure 200x3 coordinates
- (-) universal scale

Difficult optimization problem
Epipolar Geometry

\[ \lambda_2 x_2 = R \lambda_1 x_1 + T \]

- Algebraic Elimination of Depth \([\text{Longuet-Higgins '81}]:\]
  \[ x_2^T \hat{T} R x_1 = 0 \]
- Essential matrix
  \[ E = \hat{T} R \]
Epipolar Geometry

- Epipolar lines $l_1, l_2$
- Epipoles $e_1, e_2$

\[ x_2^T Ex_1 = 0 \]
\[ E = \hat{T}R \]

\[ l_1 \sim E^T x_2 \]
\[ l_i^T x_i = 0 \]
\[ l_2 \sim Ex_1 \]

\[ Ee_1 = 0 \]
\[ l_i^T e_i = 0 \]
\[ e_2 E^T = 0 \]
Characterization of the Essential Matrix

\[ x_2^T \hat{T} R x_1 = 0 \]

- Essential matrix \( E = \hat{T} R \) \quad Special 3x3 matrix

\[
\begin{bmatrix}
   e_1 & e_2 & e_2 \\
   e_4 & e_5 & e_6 \\
   e_7 & e_8 & e_9
\end{bmatrix}
\]

**Theorem 1a (Essential Matrix Characterization)**
A non-zero matrix \( E \) is an essential matrix iff its SVD: \( E = U \Sigma V^T \) satisfies: \( \Sigma = diag([\sigma_1, \sigma_2, \sigma_3]) \) with \( \sigma_1 = \sigma_2 \neq 0 \) and \( \sigma_3 = 0 \) and \( U, V \in SO(3) \)
Estimating the Essential Matrix

- Estimate Essential matrix \( E = \hat{T}R \)
- Decompose Essential matrix into \( R, T \)
  \[ x_2^T \hat{T} R x_1 = 0 \]
- Given \( n \) pairs of image correspondences:
- Find such Rotation and Translation that the epipolar error is minimized
  \[ \min_E \sum_{j=1}^{n} x_2^j E x_1^j \]
- Space of all Essential Matrices is 5 dimensional
  - 3 Degrees of Freedom – Rotation
  - 2 Degrees of Freedom – Translation (up to scale !)
Pose Recovery from the Essential Matrix

Essential matrix

\[ E = \hat{T}R \]

**Theorem 1a (Pose Recovery)**
There are two relative poses \((R, T)\) with \(T \in \mathbb{R}^3\) and \(R \in SO(3)\) corresponding to a non-zero matrix essential matrix.

\[ E = U\Sigma V^T \]

\[
(\hat{T}_1, R_1) = (UR_Z(\frac{\pi}{2})\Sigma U^T, UR_Z^T(\frac{\pi}{2})V^T)
\]

\[
(\hat{T}_2, R_2) = (UR_Z(-\frac{\pi}{2})\Sigma U^T, UR_Z^T(-\frac{\pi}{2})V^T)
\]

\[ \Sigma = \text{diag}([1, 1, 0]) \]

\[ R_z(\frac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

• Twisted pair ambiguity \((R_2, T_2) = (e^{\hat{u}\pi}R_1, -T_1)\)
Estimating Essential Matrix

\[ x_2^T \hat{T} R x_1 = 0 \]

- Denote \( a = x_1 \otimes x_2 \)
  
\[
a = [x_1 x_2, x_1 y_2, x_1 z_2, y_1 x_2, y_1 y_2, y_1 z_2, z_1 x_2, z_1 y_2, z_1 z_2]^T
\]

\[ E^s = [e_1, e_4, e_7, e_2, e_5, e_8, e_3, e_6, e_9]^T \]

- Rewrite \( a^T E^s = 0 \)

- Collect constraints from all points
  
\[ \chi E^s = 0 \]

\[ \min_E \sum_{j=1}^{n} x_2^j E x_1^j \quad \rightarrow \quad \min_{E^s} \| \chi E^s \|^2 \]
Estimating Essential Matrix

\[ \min_E \sum_{j=1}^{n} (x_2^j E x_1^j)^2 \quad \Rightarrow \quad \min_{E^s} \| \chi E^s \|_2^2 \]

Solution
- Eigenvector associated with the smallest eigenvalue of \( \chi^T \chi \)
- if \( \text{rank}(\chi^T \chi) < 8 \) degenerate configuration

**Theorem 2a** (Project to Essential Manifold)
If the SVD of a matrix \( F \in \mathbb{R}^{3 \times 3} \) is given by \( F = U \text{diag}(\sigma_1, \sigma_2, \sigma_3) V^T \)
then the essential matrix \( E \) which minimizes the Frobenius distance \( \| E - F \|_F^2 \) is given by \( E = U \text{diag}(\sigma, \sigma, 0) V^T \)
with \( \sigma = \frac{\sigma_1 + \sigma_2}{2} \)
Example- Two views

Point Feature Matching
Example – Epipolar Geometry

Camera Pose
and
Sparse Structure Recovery
**Epipolar Geometry – Planar Case**

- Plane in first camera coordinate frame

\[ aX + bY + cZ + d = 0 \]

\[ \frac{1}{d}N^T x = 1 \]

\[ \lambda_2x_2 = R\lambda_1x_1 + T \]

\[ \lambda_2x_2 = (R + \frac{1}{d}TN^T)\lambda_1x_1 \]

\[ x_2 \sim Hx_1 \]

\[
H = (R + \frac{1}{d}TN^T)
\]

**Planar homography**

Linear mapping relating two corresponding planar points in two views
Decomposition of $H$

- Algebraic elimination of depth $\hat{x}_2 H x_1 = 0$
- $H_L$ can be estimated linearly: $H_L = \lambda H$
- Normalization of $H = H_L / \sigma_3$
- Decomposition of $H$ into 4 solutions $H = (R + \frac{1}{d}TN^T)$

<table>
<thead>
<tr>
<th>$R_1 = W_1 U_1^T$</th>
<th>$R_3 = R_1$</th>
<th>$R_2 = W_2 U_2^T$</th>
<th>$R_4 = R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_1 = \tilde{v}_2 u_1$</td>
<td>$N_3 = -N_1$</td>
<td>$N_2 = \tilde{v}_2 u_2$</td>
<td>$N_4 = -N_2$</td>
</tr>
<tr>
<td>$\frac{1}{d}T_1 = (H - R_1)N_1$</td>
<td>$\frac{1}{d}T_3 = -\frac{1}{d}T_1$</td>
<td>$\frac{1}{d}T_2 = (H - R_2)N_2$</td>
<td>$\frac{1}{d}T_4 = -\frac{1}{d}T_2$</td>
</tr>
</tbody>
</table>

$$H^T H = V \Sigma V^T \quad V = [v_1, v_2, v_3] \quad \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2)$$

$$u_1 = \frac{\sqrt{1 - \sigma_3^2 v_1 + \sqrt{\sigma_1^2 - 1 v_3}}}{\sqrt{\sigma_1^2 - \sigma_3^2}} \quad u_2 = \frac{\sqrt{1 - \sigma_3^2 v_1 - \sqrt{\sigma_1^2 - 1 v_3}}}{\sqrt{\sigma_1^2 - \sigma_3^2}}$$

$$U_1 = [v_2, u_1, \tilde{v}_2 u_1], \quad W_1 = [Hv_2, Hu_1, Hv_2 Hu_1];$$

$$U_2 = [v_2, u_2, \tilde{v}_2 u_2], \quad W_2 = [Hv_2, Hu_2, Hv_2 Hu_2].$$
Motion and pose recovery for planar scene

- Given at least 4 point correspondences: $\tilde{x}_2^j H x_1^j = 0$
- Compute an approximation of the homography matrix $H_i^s$
- As nullspace of $X$
  $$\chi H_i^s = 0$$  the rows of $X$ are $a = x_1^j \otimes \tilde{x}_2^j$
- Normalize the homography matrix
  $$H = H_L / \sigma_3$$
- Decompose the homography matrix
  $$H^T H = V \Sigma V^T$$
- Select two physically possible solutions imposing positive depth constraint
$Z_\infty$ – Algorithm at Work

Simple sensors, low processing power

Obstacle avoidance
„Simple“ Image Acquisition

60 images taken with a standard low cost digital camera

http://www6.in.tum.de/burschka/
Estimation of the 6 Degrees of Freedom

Estimation of 3 rotational angles  Estimation of a translation vector
3D Reconstruction from the Images (Stereo SGM by H.Hirschmüller)

http://www6.in.tum.de/burschka/
Review

Feature selection → Feature correspondence → Camera Calibration → Euclidean Reconstruction → Sparse Structure and camera motion

Landing
Augmented Reality
Vision Based Control
Review

- Feature selection
- Feature selection
- Feature correspondence
- Projective Reconstruction
- Camera Self-Calibration
- Euclidean Reconstruction

Partial Scene Knowledge
Partial Motion Knowledge
Partial Calibration Knowledge
Examples
Feature Selection

- Compute Image Gradient \( \nabla I^T = [I_x, I_y] \)
- Compute Feature Quality function \( C(x) \) measure for each pixel
  \[
  C(x) = \det(G) + k \cdot \text{trace}^2(G)
  \]
  \[
  G = \begin{bmatrix}
  \sum I_x^2 & \sum I_xI_y \\
  \sum I_xI_y & \sum I_y^2
  \end{bmatrix}
  \]
- Search for local maxima
Feature Tracking

• Translational motion model

\[ E(d) = \min_d \sum_{W(x)} [I_2(\bar{x} + d) - I_1(\bar{x})]^2 \]

• Closed form solution

\[ d = -G^{-1}b \]

\[ G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \]

\[ b = \begin{bmatrix} \sum_{W(x)} I_x I_t \\ \sum_{W(x)} I_y I_t \end{bmatrix} \]

1. Build an image pyramid
2. Start from coarsest level
3. Estimate the displacement at the coarsest level
4. Iterate until finest level
Coarse to fine feature tracking

2

1. compute $d_k = -Gb$
2. warp the window $W(x)$ in the second image by $2d_k$
3. update the displacement $d \leftarrow d + 2d_k$
4. go to finer level $k \leftarrow k - 1$
5. At the finest level repeat for several iterations
Tracked Features
Wide baseline matching

Point features detected by Harris Corner detector
Wide baseline Feature Matching

1. Select the features in two views
2. For each feature in the first view
3. Find the feature in the second view that maximizes
4. Normalized cross-correlation measure

\[
NC\!C(d, x) = \frac{\sum_{\tilde{x} \in W(x)} (I_1(\tilde{x}) - \bar{I}_1)(I_2(\tilde{x}+d) - \bar{I}_2)}{\sqrt{\sum_{\tilde{x} \in W(x)} (I_1(\tilde{x}) - \bar{I}_1)^2 \sum_{\tilde{x} \in W(x)} (I_2(\tilde{x}+d) - \bar{I}_2)^2}}
\]

Select the candidate with the similarity above selected threshold.
Inliers

\[ d_j \leq \tau_d \]

Outliers

\[ d_j > \tau_d \]
Epipolar Geometry

- Epipolar geometry in two views
- Refined epipolar geometry using nonlinear estimation of F