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)

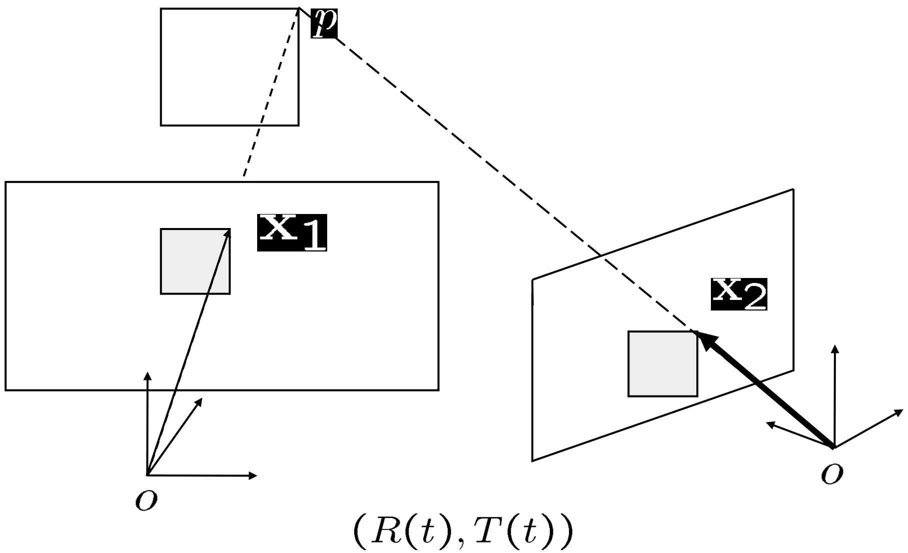
Estimate from changes in the image position
- camera motion
- structure of the environment

Image Primitives and Correspondence

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

Some slides based on MASK book

Matching - Correspondence



Lambertian assumption $$I_1(\mathbf{x}_1) = \mathcal{R}(p) = I_2(\mathbf{x}_2)$$

Rigid body motion $$\mathbf{x}_2 = h(\mathbf{x}_1) = \frac{1}{\lambda_2(\mathbf{X})}(R\lambda_1(\mathbf{X})\mathbf{x}_1 + T)$$

Correspondence $$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

Local Deformation Models

- Translational model

$$h(\mathbf{x}) = \mathbf{x} + d \qquad\qquad I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

- Affine model

$$h(\mathbf{x}) = A\mathbf{x} + d \qquad\qquad I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_1))$$

- Transformation of the intensity values and occlusions

$$I_1(\mathbf{x}_1) = f_o(\mathbf{X}, g) I_2(h(\mathbf{x}_1) \cdot \qquad)$$

Feature Tracking and Optical Flow

- Translational model

$$I_1(\mathbf{x}_1) = I_2(\mathbf{x}_1 + \Delta\mathbf{x})$$

- Small baseline

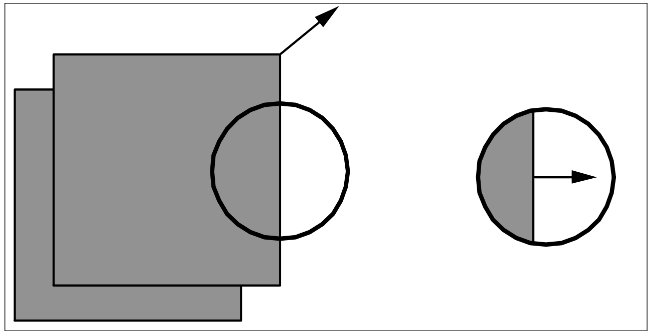
$$I(\mathbf{x}(t), t) = I(\mathbf{x}(t) + \mathbf{u}dt, t + dt)$$

- Approx. by first two terms of Taylor series

$$\nabla I(\mathbf{x}(t), t)^T \mathbf{u} + I_t(\mathbf{x}(t), t) = 0$$

- Brightness constancy constraint

Aperture Problem



- Normal flow

$$\mathbf{u}_n \doteq \frac{\nabla I^T \mathbf{u}}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|} = -\frac{I_t}{\|\nabla I\|} \cdot \frac{\nabla I}{\|\nabla I\|}$$

Optical Flow

- Integrate around over image patch

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

- Solve

$$\begin{aligned}\nabla E_b(\mathbf{u}) &= 2 \sum_{W(x,y)} \nabla I(\nabla I^T \mathbf{u} + I_t) \\ &= 2 \sum_{W(x,y)} \left(\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \right)\end{aligned}$$

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} = 0$$

$$G\mathbf{u} + \mathbf{b} = 0$$

Optical Flow, Feature Tracking

$$\mathbf{u} = -G^{-1}\mathbf{b}$$

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

Conceptually:

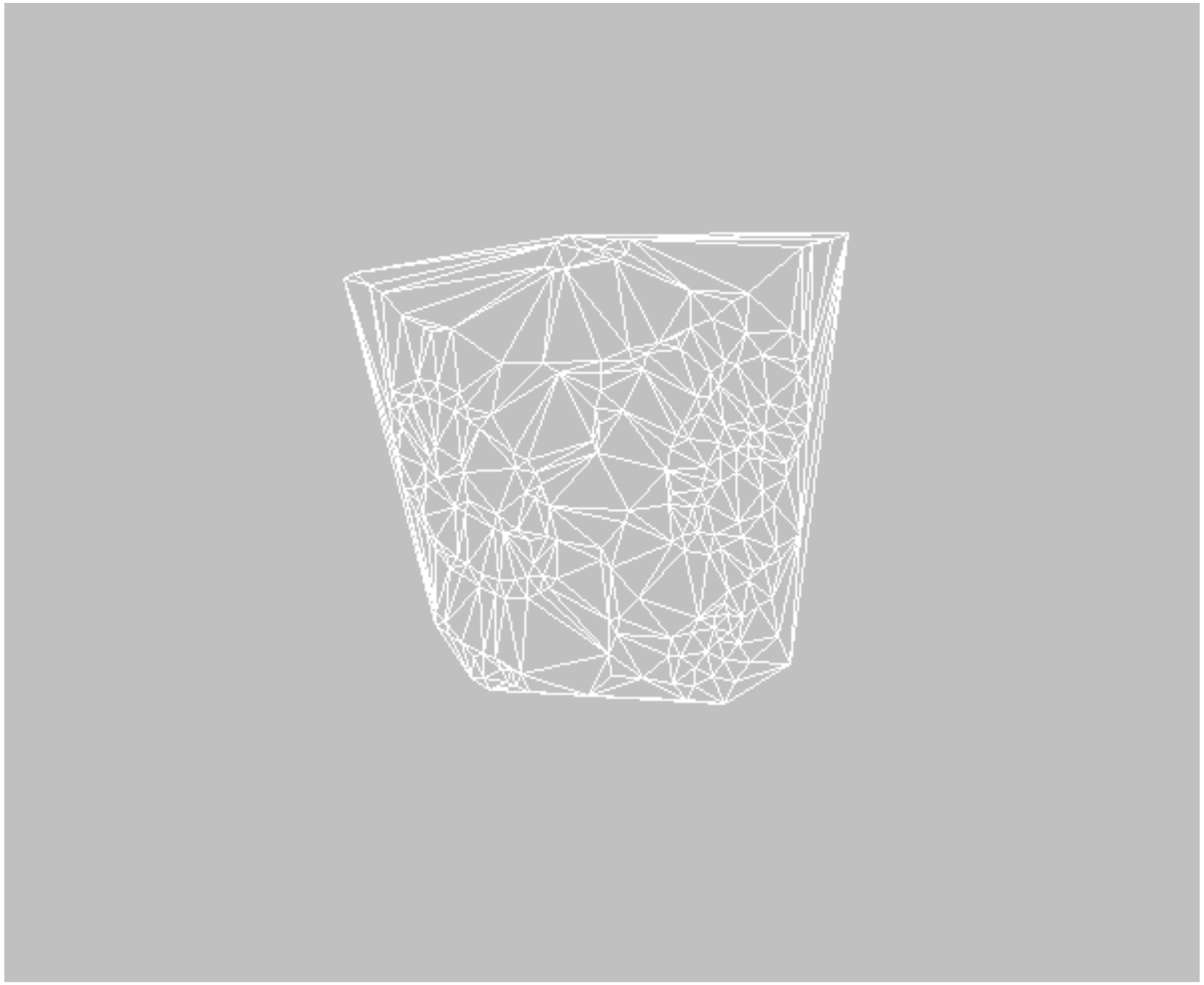
rank(G) = 0 blank wall problem
rank(G) = 1 aperture problem
rank(G) = 2 enough texture – good feature candidates

In reality: choice of threshold is involved

Feature Tracking

Some slides based on MASK book

3D Reconstruction - Preview



Some slides based on MASK book

Point Feature Extraction

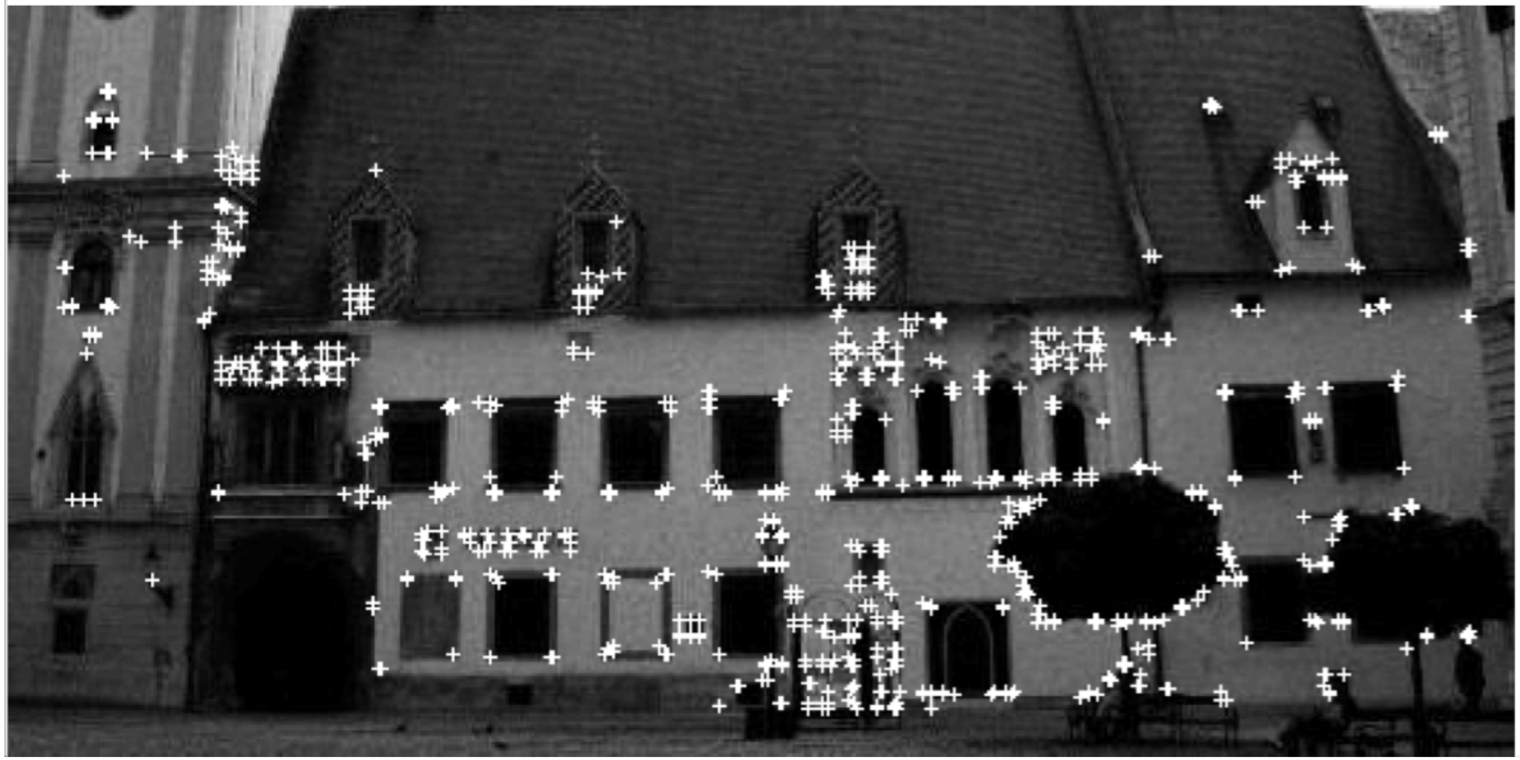
$$G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- Compute eigenvalues of G
- If smalest eigenvalue σ of G is bigger than τ - mark pixel as candidate feature point

- Alternatively feature quality function (Harris Corner Detector)

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

Harris Corner Detector - Example



Some slides based on MASK book

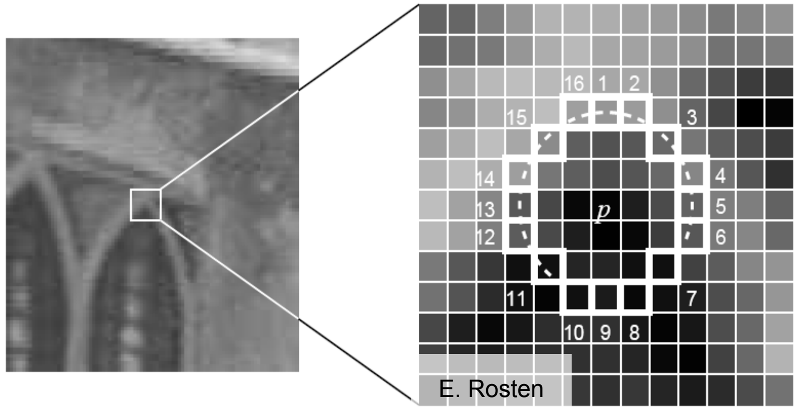
Wide Baseline Matching

Some slides based on MASK book

Mair, Hager, Burschka, Suppa, Hirzinger

Adaptive and Generic Accelerated Segment Test (AGAST)

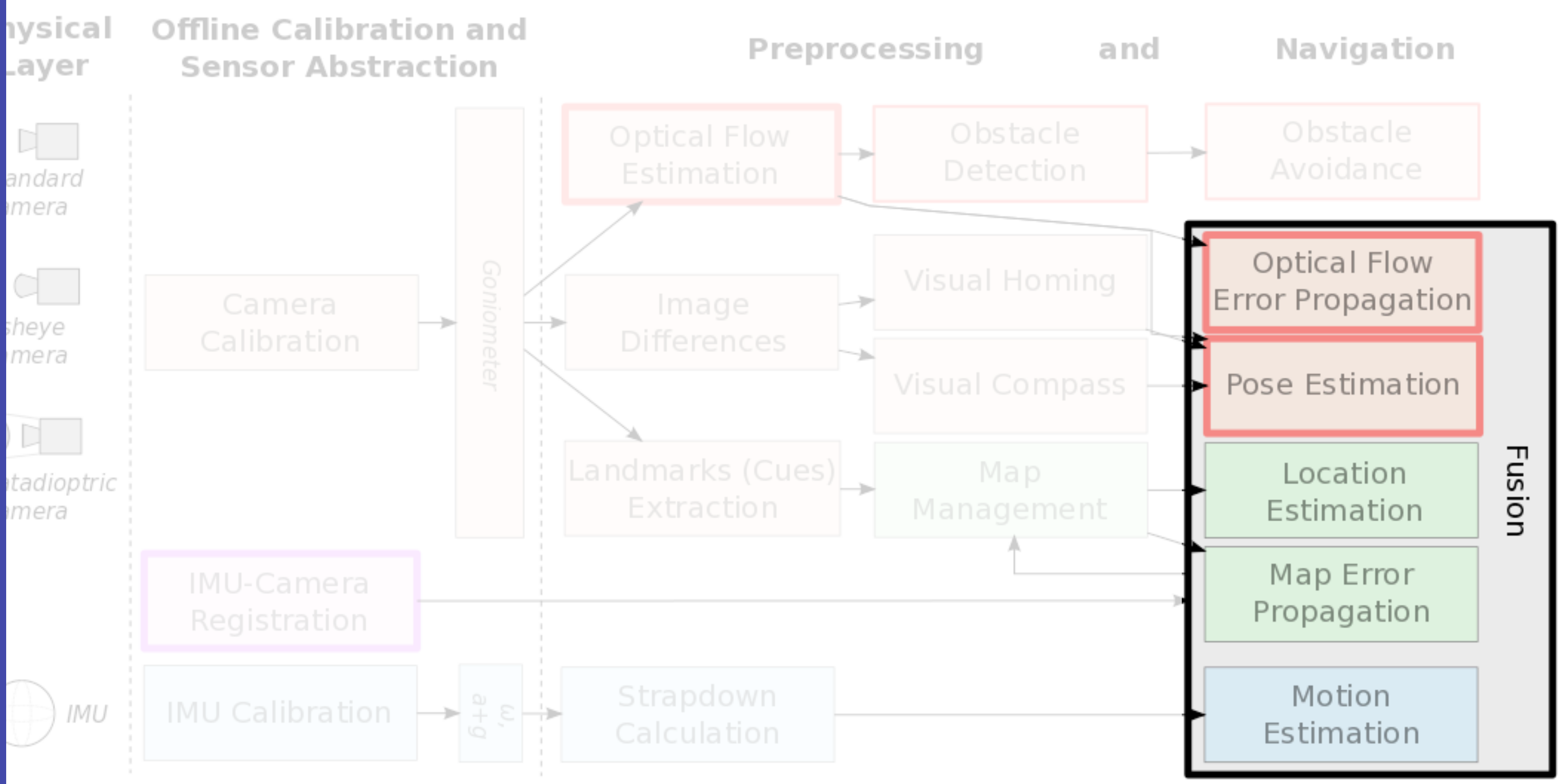
- Improvements compared to FAST:
  - full exploration of the configuration space by backward-induction (no learning)
  - binary decision tree (not ternary)
  - computation of the actual probability and processing costs (no greedy algorithm)
  - automatic scene adaption by tree switching (at no cost)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



Fusion Camera IMU

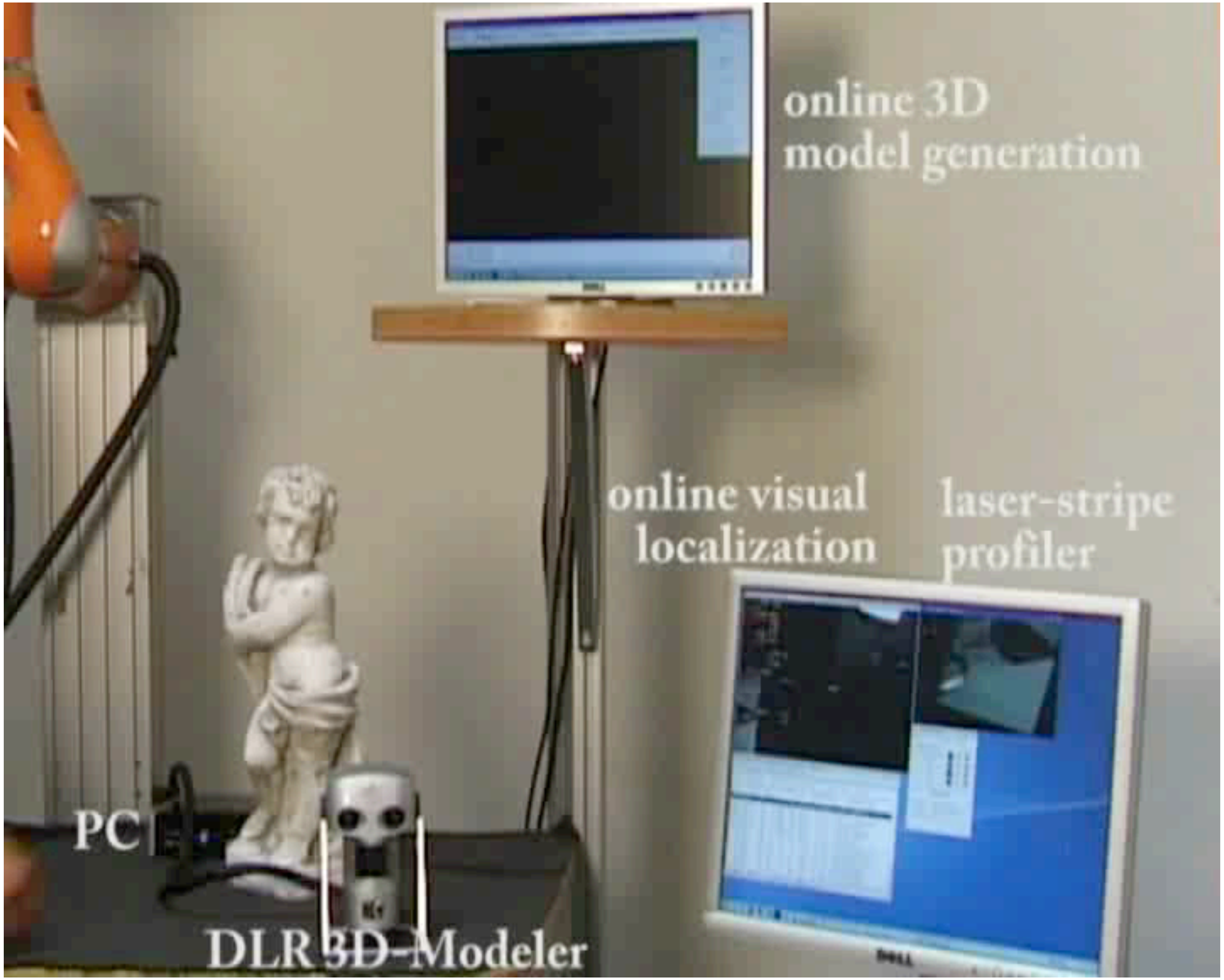
- IMU-camera setup
  - 15 Hz camera (120° aperture angle) and 120 Hz IMU
- Experiments
  - 8 runs: moving in front of a checkerboard, using bundle adjustment for pose estimation
- Implemented approaches for comparison
  - UKF-based
  - grey-box approach

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 → pseudo-measurement model
  - lack of scale in visual odometry → keyframe based scale estimation by state augmentation
  - loss of features over time → high frame-rates

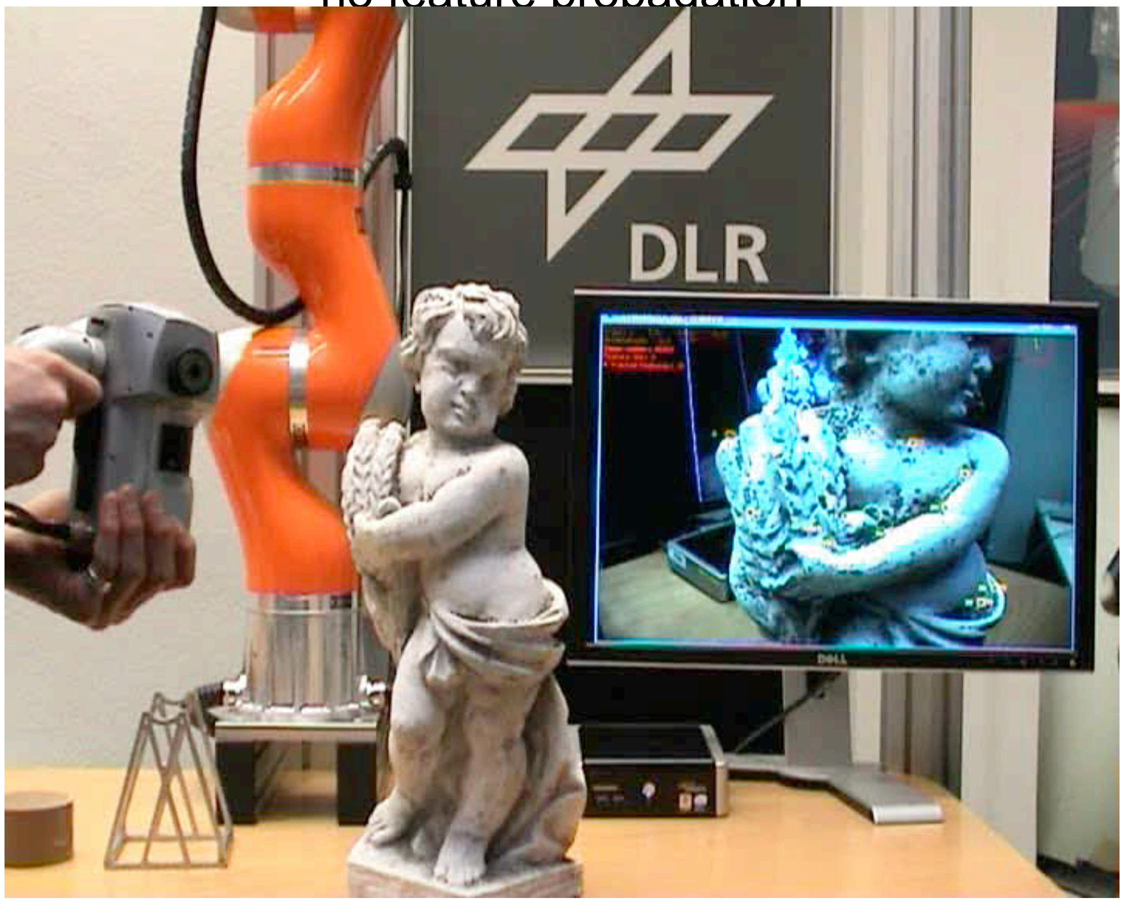

online 3D
model generation
online visual
localization
laser-stripe
profiler
PC
DLR 3D-Modeler

Strobl, Mair, Bodenmüller, Kielhofer, Sepp, Suppa, Burschka, Hirzinger
IROS, IEEE/RSJ, 2009, *Best Paper Finalist*

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

no feature propagation

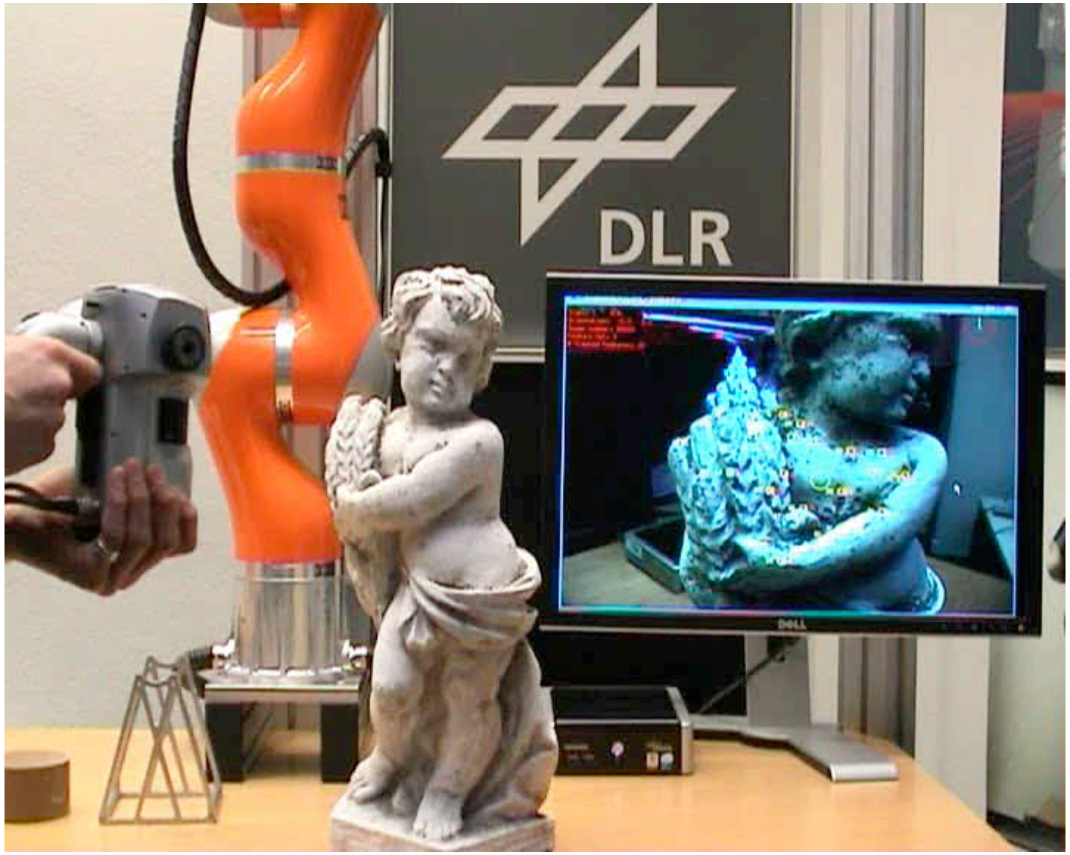


Strobl, Mair, Bodenmüller, Kielhofer, Sepp, Suppa, Burschka, Hirzinger
IROS, IEEE/RSJ, 2009, *Best Paper Finalist*

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

linear feature propagation

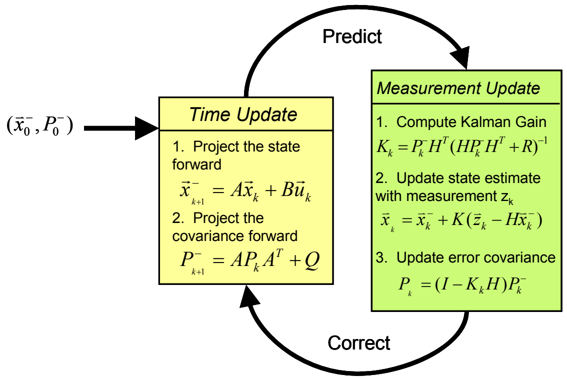


Data Fusion Example

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

$$\vec{x}^{-}_{k+1} = A\vec{x}_k + B\vec{u}_k$$

- 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 $\sigma_3^2 = \sigma_1^2 + \sigma_2^2$



$$P_k = \frac{1}{N}\sum_{i=1}^{N}(\vec{x}_i - \vec{\mu}_i)^T(\vec{x}_i - \vec{\mu}_i)$$

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

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

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

$$P_{k+1} = AP_kA^T + Q$$

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 lest-squares perspective…
- We have 2 measurements that are equal to r plus some additive zero-mean Gaussian noise v_1 and v_2

$$z_1 = r + N(0, R_1) = r + v_1$$
$$z_2 = r + N(0, R_2) = r + v_2$$

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

$$\frac{\partial e}{\partial \hat{r}} = \frac{\partial}{\partial \hat{r}} \sum_{i=1}^{n} w_i (\hat{r} - z_i)^2 = 2 \sum_{i=1}^{n} w_i (\hat{r} - z_i) = 0$$

A Least-Squares Approach

- This can be rewritten as

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

Kalman Gain

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

$$\hat{r}_{k+1} = \hat{r}^-_{k+1} + \frac{P^-_{k+1}}{P^-_{k+1} + R}(z_{k+1} - \hat{r}^-_{k+1}) \quad \Rightarrow \hat{r}_{k+1} = \hat{r}^-_{k+1} + K_{k+1}(z_{k+1} - \hat{r}^-_{k+1})$$

- For merging Gaussian distributions, the update rule is

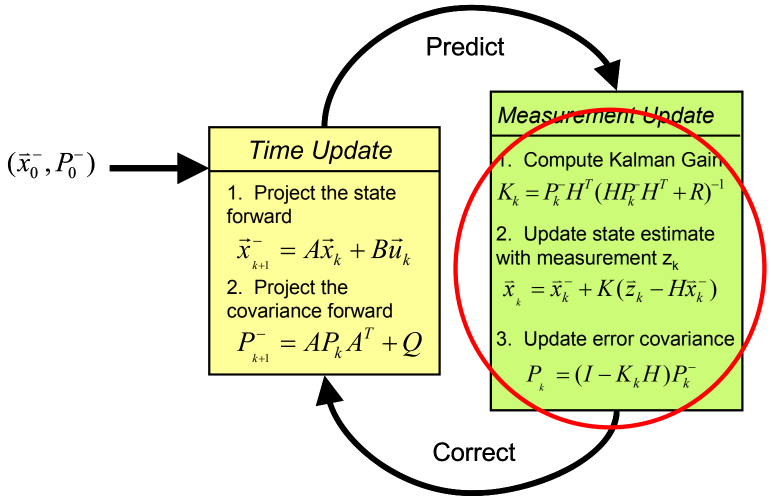
$$\frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \sigma_2^2} \quad \Rightarrow \quad \sigma_3^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

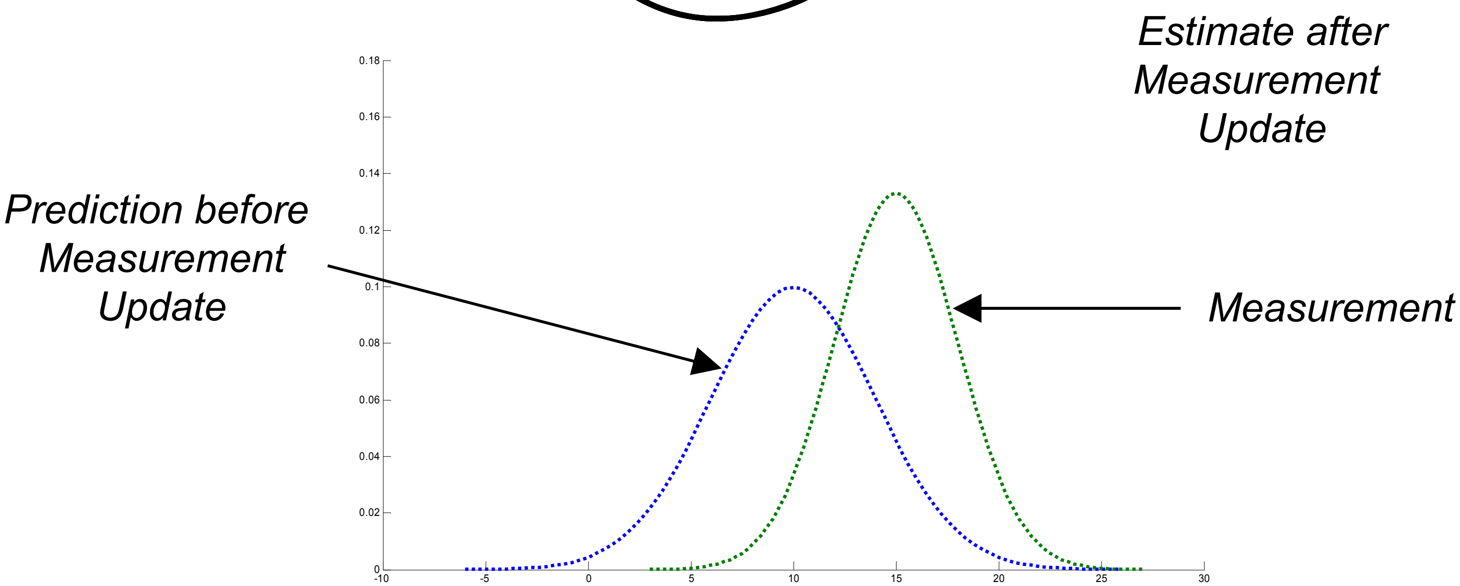
which if we write in our measurement update equation form we get

$$P_{k+1} = \frac{P^-_{k+1} R_{k+1}}{P^-_{k+1} + R_{k+1}} \equiv P^-_{k+1} - K_{k+1} P^-_{k+1}$$

The Measurement Update Phase

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



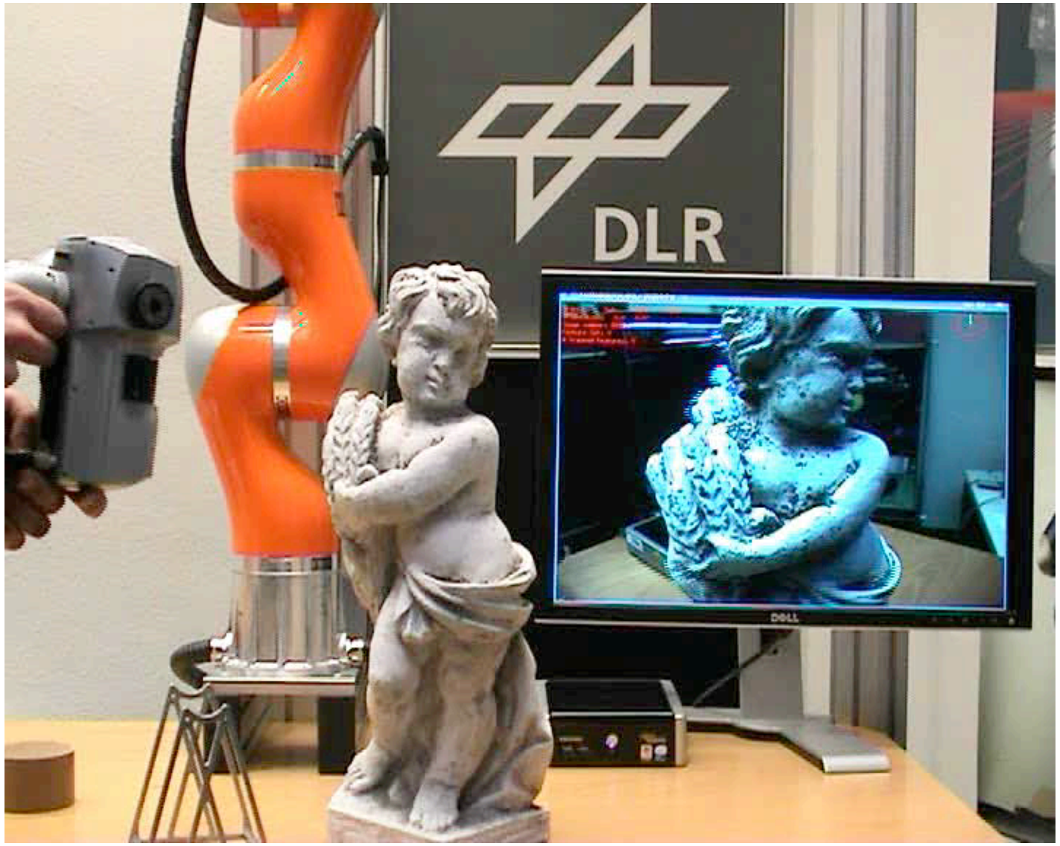


Strobl, Mair, Bodenmüller, Kielhofer, Sepp, Suppa, Burschka, Hirzinger
IROS, IEEE/RSJ, 2009, *Best Paper Finalist*

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

linear + gyros based prop.



Region based Similarity Metric

- Sum of squared differences

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

- Normalize cross-correlation

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

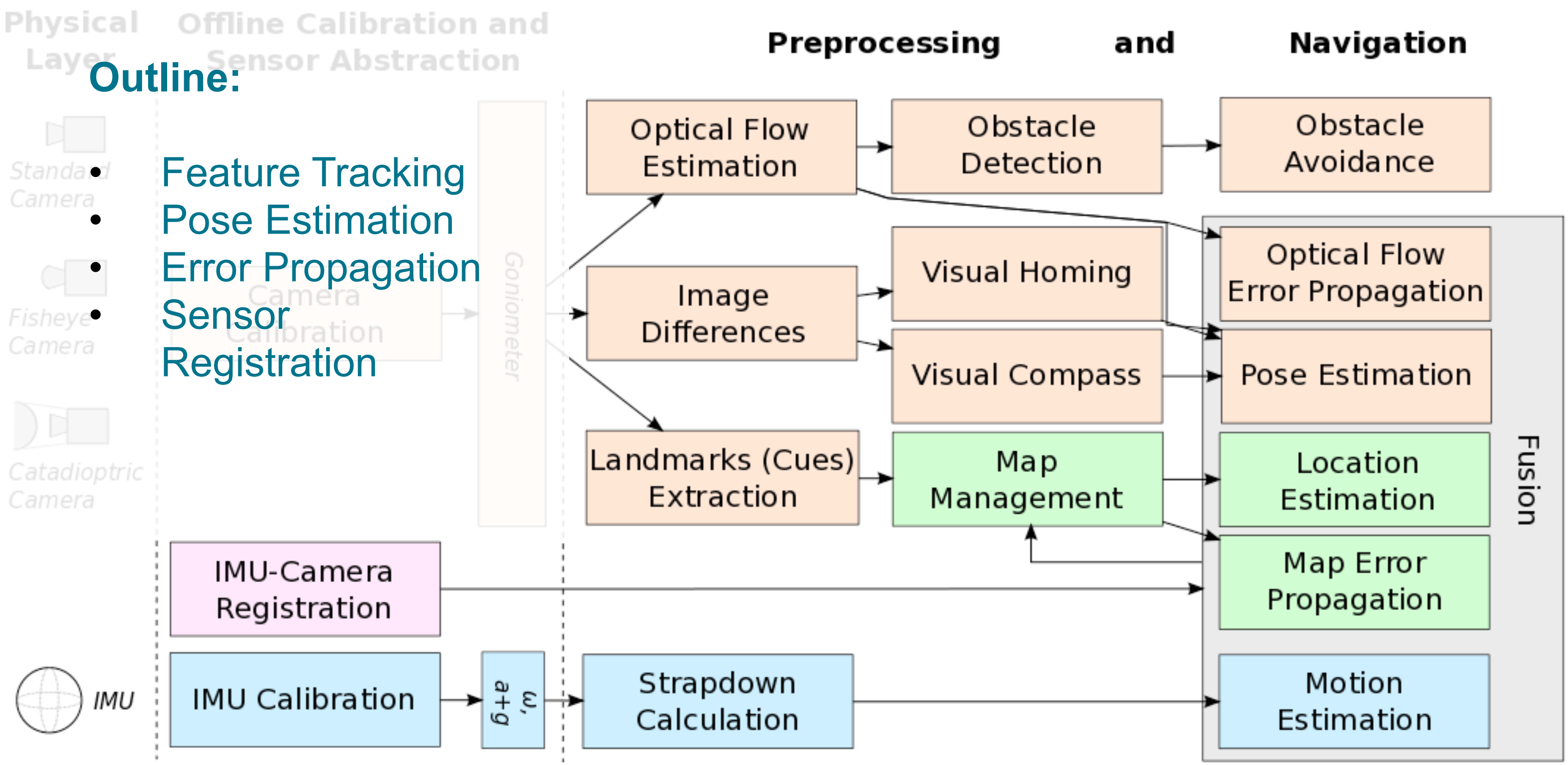
- Sum of absolute differences

$$SAD(h) = \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} |I_1(\tilde{\mathbf{x}}) - I_2(h(\tilde{\mathbf{x}}))|$$

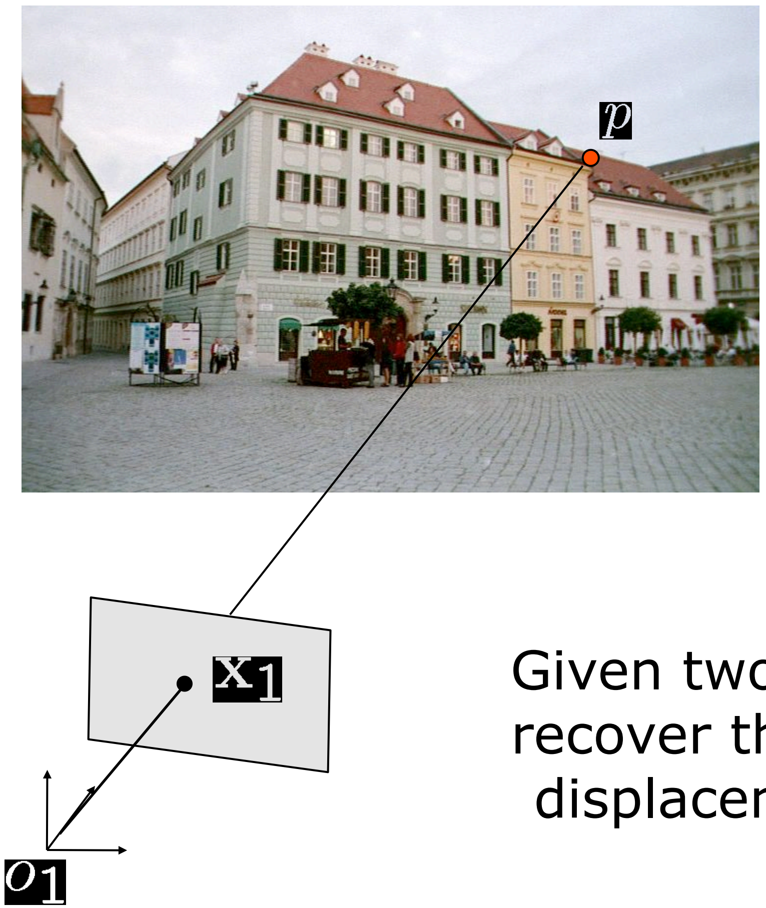
Algorithmic Concept

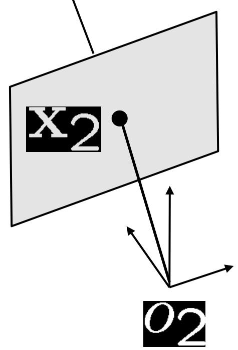
Outline:

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



General Formulation





Given two views of the scene recover the unknown camera displacement and 3D scene structure

Pinhole Camera Model

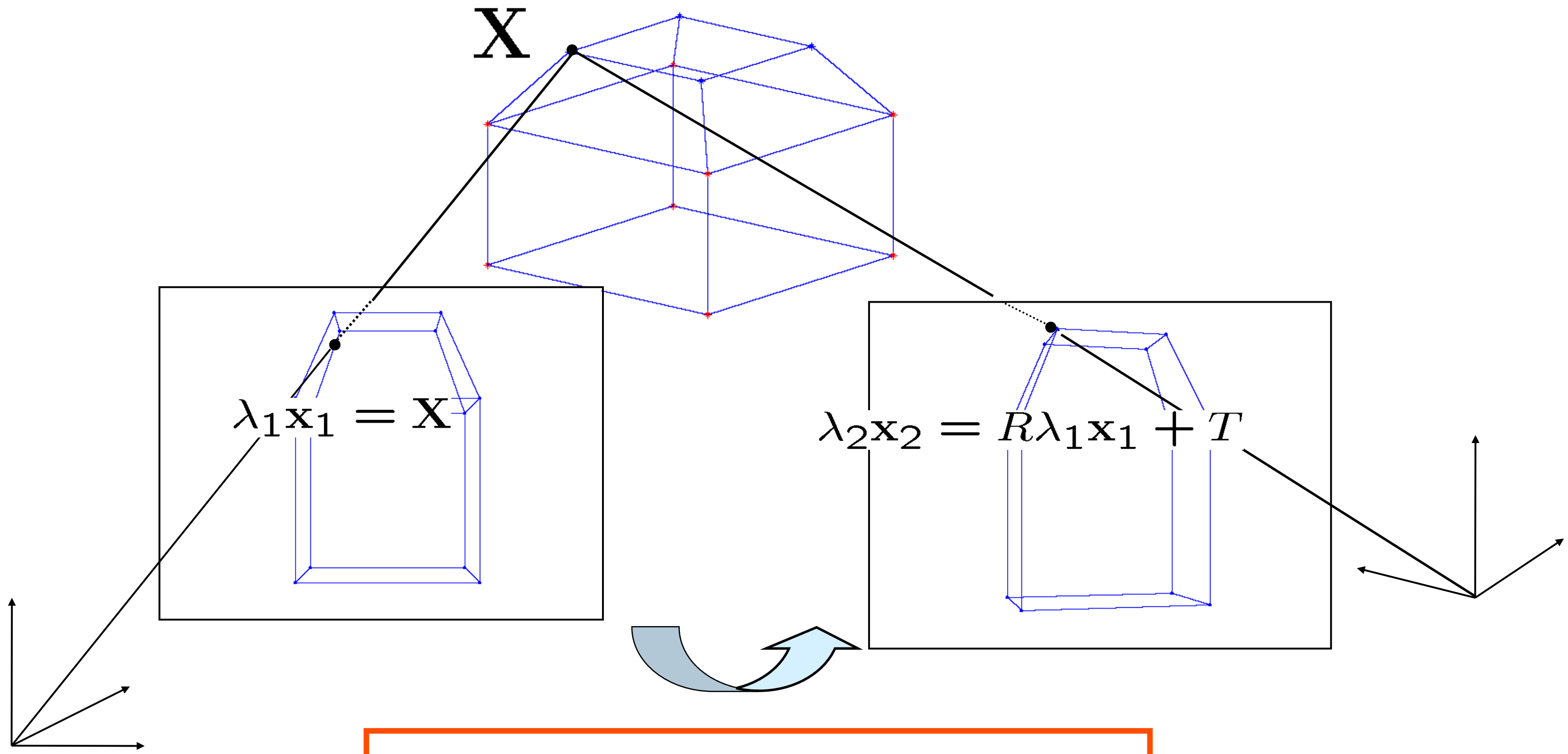
- 3D points $\mathbf{X} = [X, Y, Z, W]^T \in \Re^4, \quad (W = 1)$
- Image points $\mathbf{x} = [x, y, z]^T \in \Re^3, \quad (z = 1)$
- Perspective Projection $\lambda \mathbf{x} = \mathbf{X}$

$$\lambda = Z \quad x = \frac{X}{Z} \quad y = \frac{Y}{Z}$$

- Rigid Body Motion $\Pi = [R, T] \in \Re^{3\times 4}$
- Rigid Body Motion + Projective projection

$$\lambda \mathbf{x} = \Pi \mathbf{X} = [R, T]\mathbf{X}$$

Rigid Body Motion – Two views



$$\lambda_2 \mathbf{x}_2 = R\lambda_1 \mathbf{x}_1 + T$$

3D Structure and Motion Recovery

Euclidean transformation

$$\lambda_2 \mathbf{x}_2 = R\lambda_1 \mathbf{x}_1 + T$$

measurements unknowns

$$\sum_{j=1}^{n} \|\mathbf{x}_1^j - \pi(R_1, T_1, \mathbf{X})\|^2 + \|\mathbf{x}_2^j - \pi(R_2, T_2, \mathbf{X})\|^2$$

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

Two views ~ 200 points

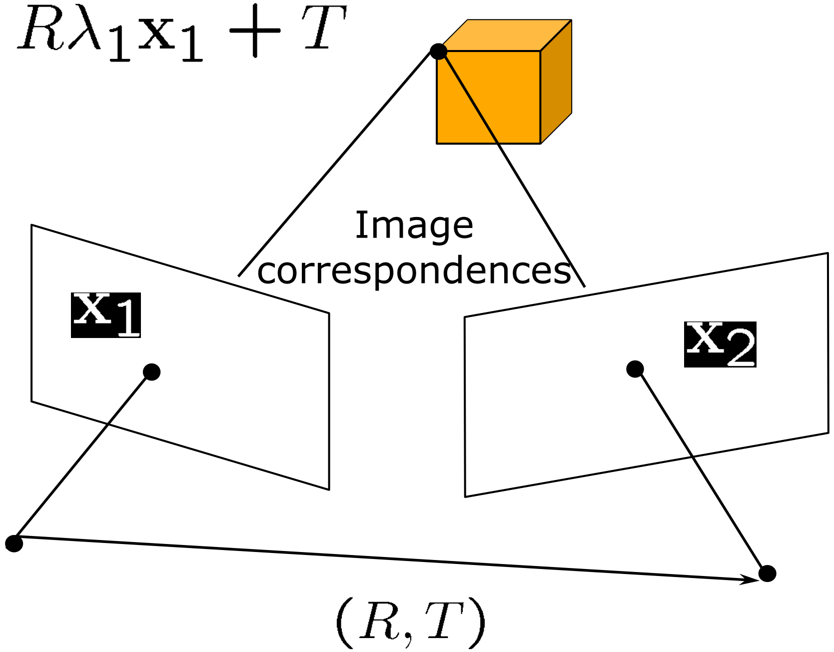
6 unknowns – Motion 3 Rotation, 3 Translation

- Structure 200x3 coordinates
- (-) universal scale

Difficult optimization problem

Epipolar Geometry

$$\lambda_2 \mathbf{x}_2 = R \lambda_1 \mathbf{x}_1 + T$$



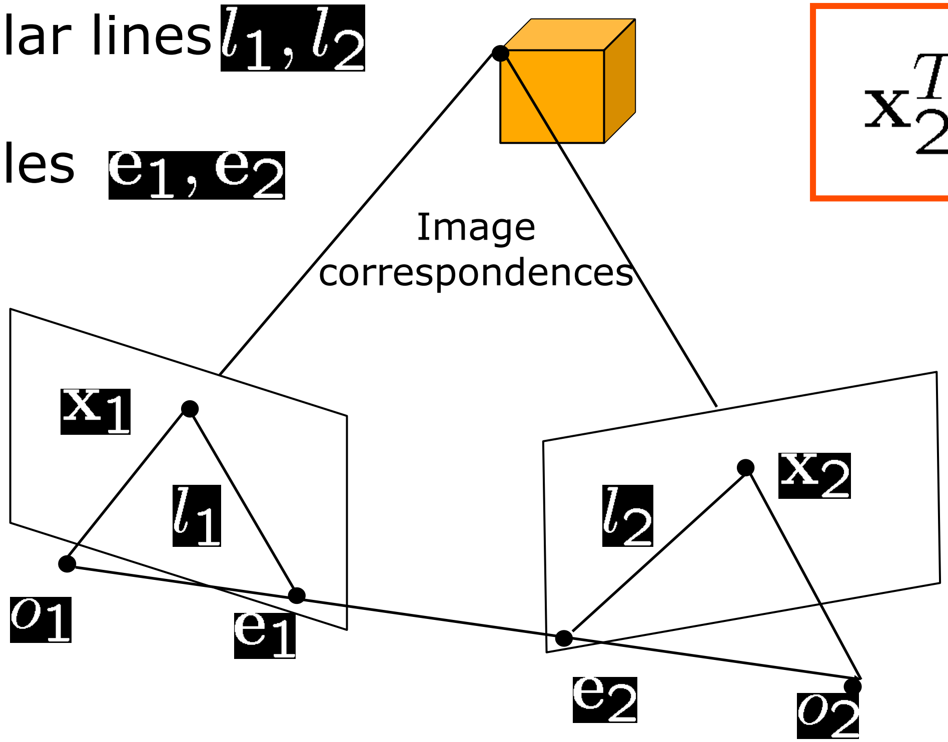
- Algebraic Elimination of Depth [Longuet-Higgins '81]:

$$\mathbf{x}_2^T \underbrace{\hat{T} R}_{E} \mathbf{x}_1 = 0$$

- Essential matrix $E = \hat{T} R$

Epipolar Geometry

- Epipolar lines l_1, l_2
- Epipoles $\mathbf{e}_1, \mathbf{e}_2$



$$\mathbf{x}_2^T E \mathbf{x}_1 = 0$$

$$E = \hat{T} R$$

$$l_1 \sim E^T \mathbf{x}_2 \qquad l_i^T \mathbf{x}_i = 0 \qquad l_2 \sim E \mathbf{x}_1$$

$$E \mathbf{e}_1 = 0 \qquad l_i^T \mathbf{e}_i = 0 \qquad \mathbf{e}_2 E^T = 0$$

Characterization of the Essential Matrix

$$\mathbf{x}_2^T \hat{T} R \mathbf{x}_1 = 0$$

- Essential matrix $E = \hat{T} R$ Special 3x3 matrix

$$\mathbf{x}_2^T \begin{bmatrix} e_1 & e_2 & e_2 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \mathbf{x}_1 = 0$$

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

$$\mathbf{x}_2^T \hat{T} R \mathbf{x}_1 = 0$$

- Given n pairs of image correspondences:
- Find such Rotation and Translation that the epipolar error is minimized

$$min_E \sum_{j=1}^{n} \mathbf{x}_2^{jT} E \mathbf{x}_1^j$$

- 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 $T \in \mathcal{R}^3$ and $R \in SO(3)$ corresponding to a non-zero matrix essential matrix.

$$E = U\Sigma V^T$$

$$\begin{aligned}(\hat{T}_1, R_1) &= (UR_Z(+\tfrac{\pi}{2})\Sigma U^T, UR_Z^T(+\tfrac{\pi}{2})V^T)\\ (\hat{T}_2, R_2) &= (UR_Z(-\tfrac{\pi}{2})\Sigma U^T, UR_Z^T(-\tfrac{\pi}{2})V^T)\end{aligned}$$

$$\Sigma = diag([1,1,0]) \qquad R_z(+\tfrac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Estimating Essential Matrix

$$\mathbf{x}_2^T \hat{T} R \mathbf{x}_1 = 0$$

- Denote $\mathbf{a} = \mathbf{x}_1 \otimes \mathbf{x}_2$

$$\mathbf{a} = [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2]^T$$

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

- Rewrite $\mathbf{a}^T E^s = 0$

- Collect constraints from all points

$$\chi E^s = 0$$

$$min_E \sum_{j=1}^n \mathbf{x}_2^{jT} E \mathbf{x}_1^j \quad \Rightarrow \quad min_{E^s} \|\chi E^s\|^2$$

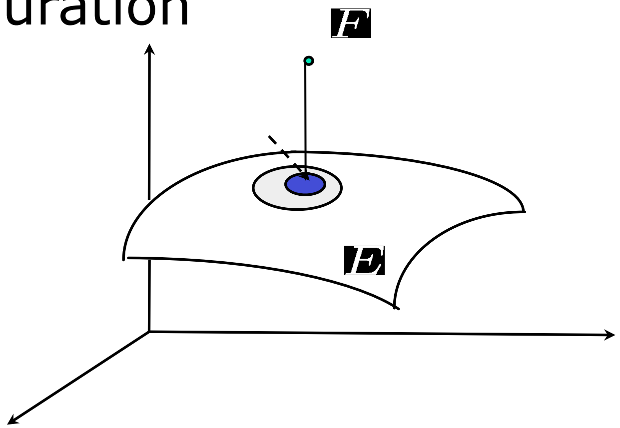
Estimating Essential Matrix

$$min_E \sum_{j=1}^{n} (\mathbf{x}_2^{jT} E \mathbf{x}_1^j)^2 \quad \Rightarrow \quad min_{E^s} \|\chi E^s\|^2$$

Solution

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

Projection on to Essential Space

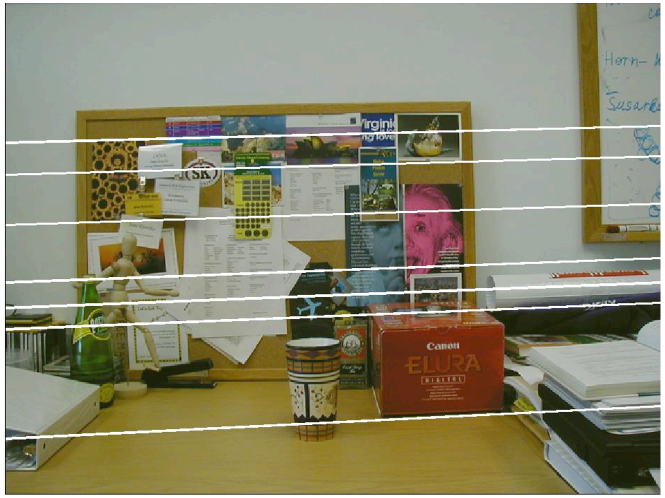


Theorem 2a (Project to Essential Manifold)
If the SVD of a matrix $F \in \mathcal{R}^{3\times 3}$ is given by $F = Udiag(\sigma_1, \sigma_2, \sigma_3)V^T$ then the essential matrix E which minimizes the Frobenius distance $\|E - F\|_f^2$ is given by $E = Udiag(\sigma, \sigma, 0)V^T$ with $\sigma = \frac{\sigma_1 + \sigma_2}{2}$

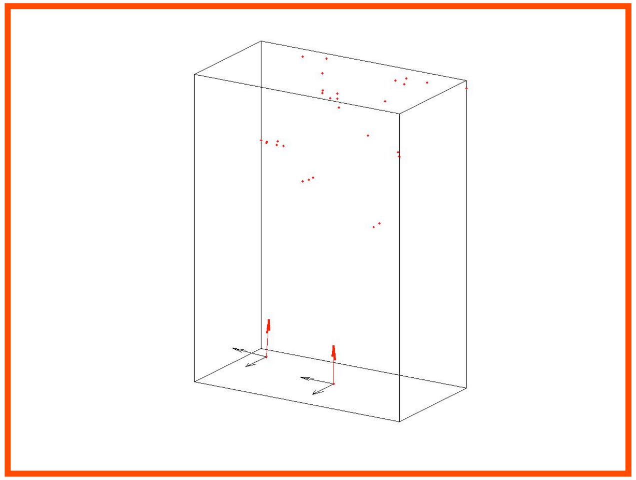
Example- Two views

Point Feature Matching

Example – Epipolar Geometry



Camera Pose
and
Sparse Structure Recovery



Epipolar Geometry – Planar Case

- Plane in first camera coordinate frame

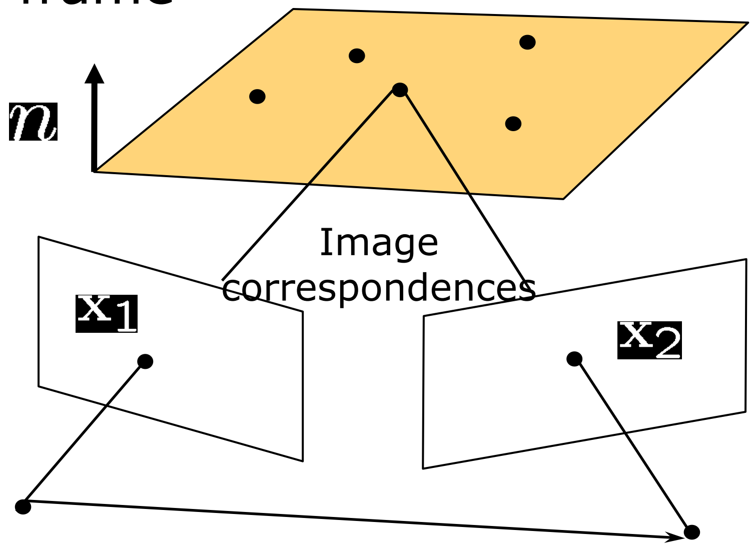
$$aX + bY + cZ + d = 0$$

$$\frac{1}{d} N^T \mathbf{X} = 1$$

$$\lambda_2 \mathbf{x}_2 = R \lambda_1 \mathbf{x}_1 + T$$

$$\lambda_2 \mathbf{x}_2 = (R + \frac{1}{d} T N^T) \lambda_1 \mathbf{x}_1$$

$$\mathbf{x}_2 \sim H \mathbf{x}_1$$



Planar homography

$$H = (R + \frac{1}{d} T N^T)$$

Linear mapping relating two corresponding planar points in two views

Decomposition of H

- Algebraic elimination of depth $\widehat{\mathbf{x}_2} H \mathbf{x}_1 = 0$
- H_L can be estimated linearly $H_L = \lambda H$
- Normalization of $H = H_L / \sigma_3$
- Decomposition of H into 4 solutions $H = (R + \frac{1}{d} T N^T)$

$R_1 = W_1 U_1^T$ $N_1 = \widehat{v_2} u_1$ $\frac{1}{d} T_1 = (H - R_1) N_1$	$R_3 = R_1$ $N_3 = -N_1$ $\frac{1}{d} T_3 = -\frac{1}{d} T_1$	$R_2 = W_2 U_2^T$ $N_2 = \widehat{v_2} u_2$ $\frac{1}{d} T_2 = (H - R_2) N_2$	$R_4 = R_2$ $N_4 = -N_2$ $\frac{1}{d} T_4 = -\frac{1}{d} T_2$

- $$H^T H = V \Sigma V^T \quad V = [v_1, v_2, v_3] \quad \Sigma = diag(\sigma_1^2, \sigma_2^2, \sigma_3^2)$$
  $$u_1 \doteq \frac{\sqrt{1-\sigma_3^2} v_1 + \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \qquad u_2 \doteq \frac{\sqrt{1-\sigma_3^2} v_1 - \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}}$$
  $$U_1 = [v_2, u_1, \widehat{v_2} u_1], \qquad W_1 = [H v_2, H u_1, \widehat{H v_2} H u_1];$$
  $$U_2 = [v_2, u_2, \widehat{v_2} u_2], \qquad W_2 = [H v_2, H u_2, \widehat{H v_2} H u_2].$$

Motion and pose recovery for planar scene

- Given at least 4 point correspondences $\widehat{\mathbf{x}_2^j} H \mathbf{x}_1^j = 0$
- Compute an approximation of the homography matrix H_l^s
- As nullspace of χ

$$\chi H_l^s = 0 \qquad \text{the rows of } \chi \text{ are } \mathbf{a} = \mathbf{x}_1^j \otimes \widehat{\mathbf{x}_2^j}$$

- Normalize the homography matrix

$$H = H_L / \sigma_3$$

- Decompose the homography matrix

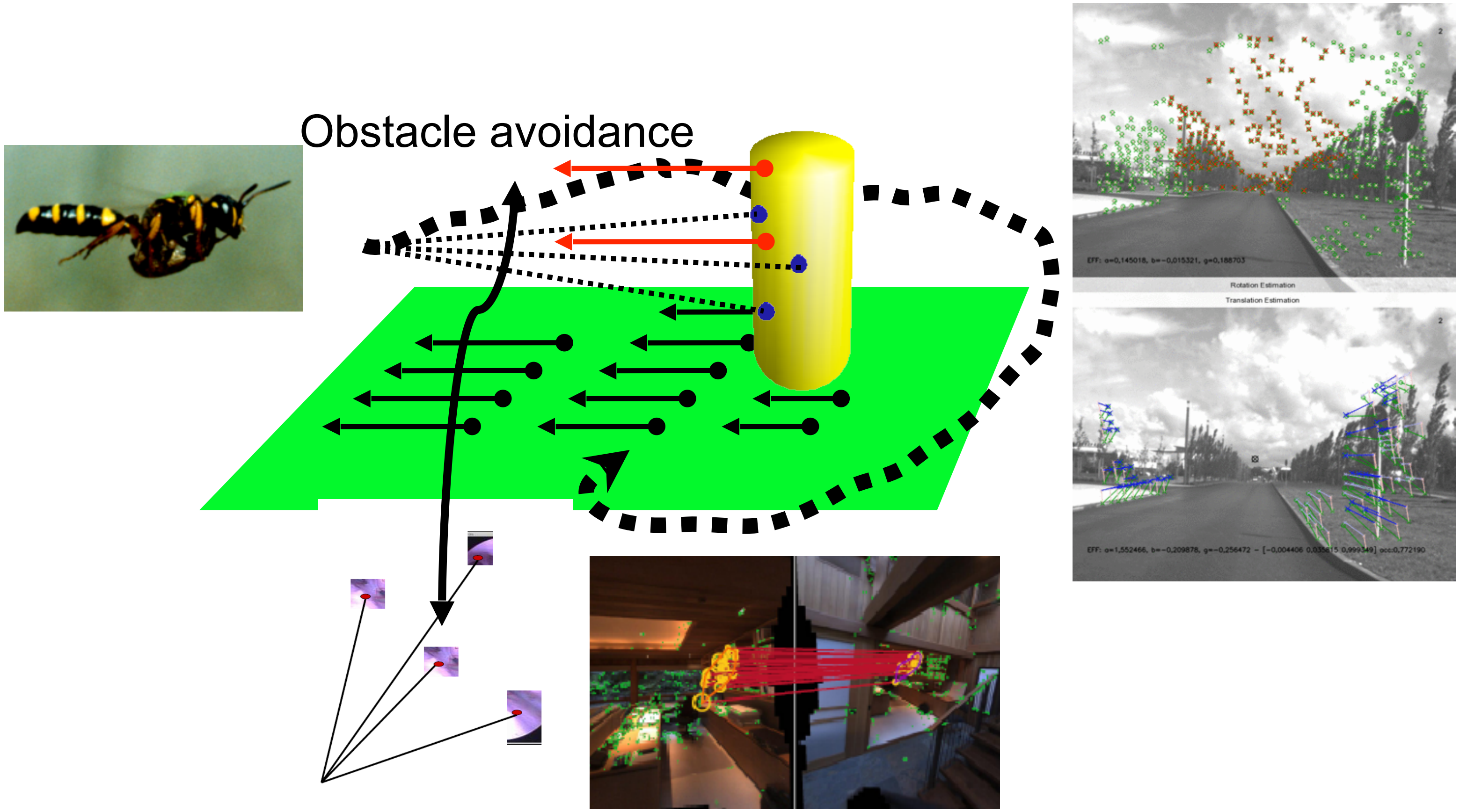
$$H^T H = V \Sigma V^T$$

- Select two physically possible solutions imposing positive depth constraint

Z_∞ – Algorithm at Work

Mair, Burschka

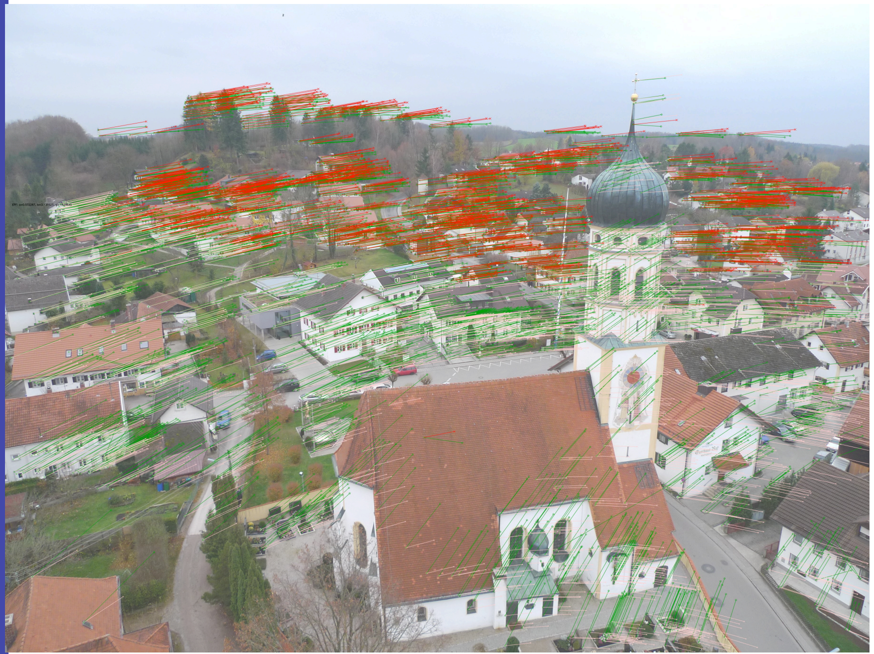
Simple sensors, low processing power



„Simple“ Image Acquisition

60 images taken with a standard low cost digital camera

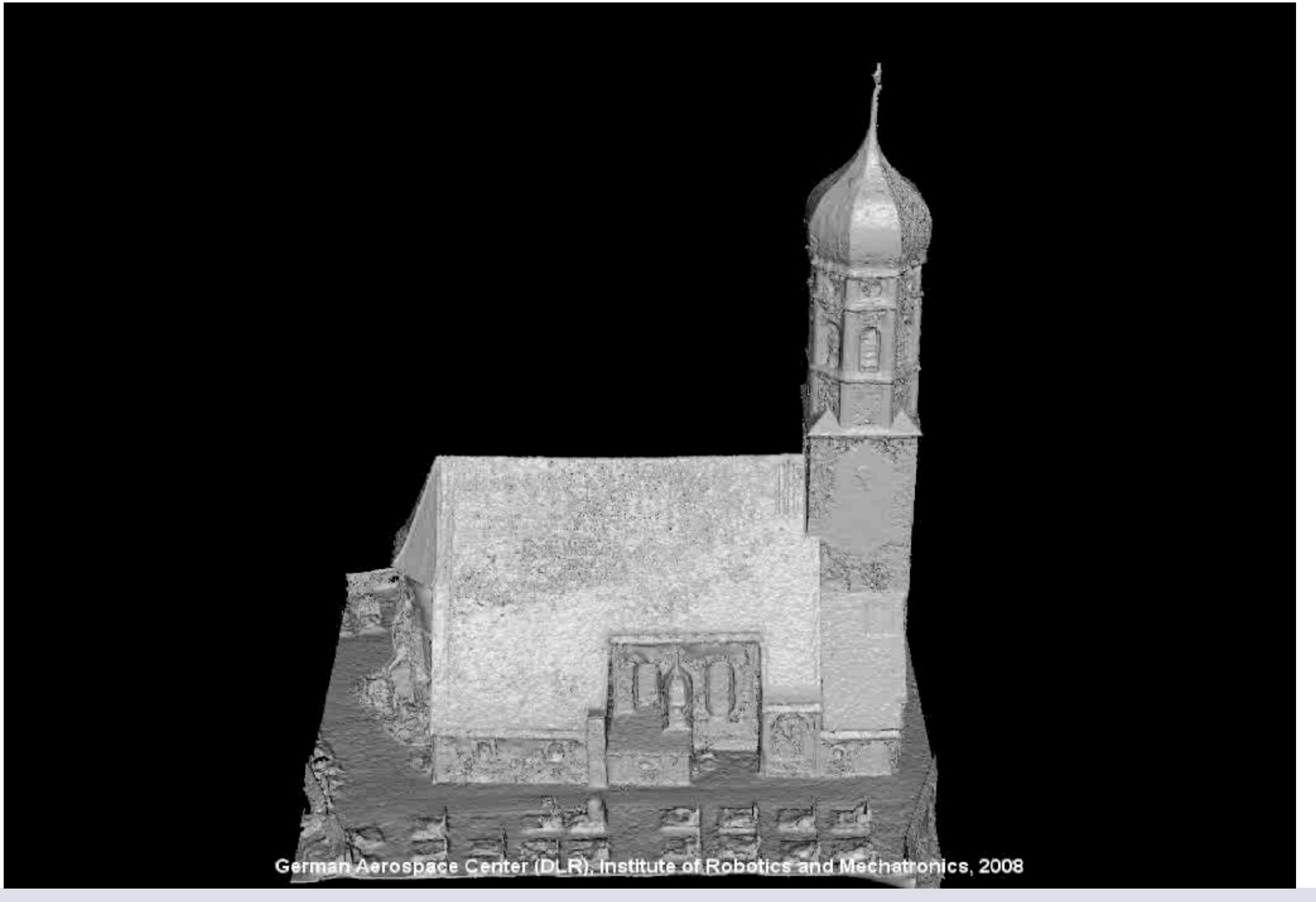
Estimation of the 6 Degrees of Freedom



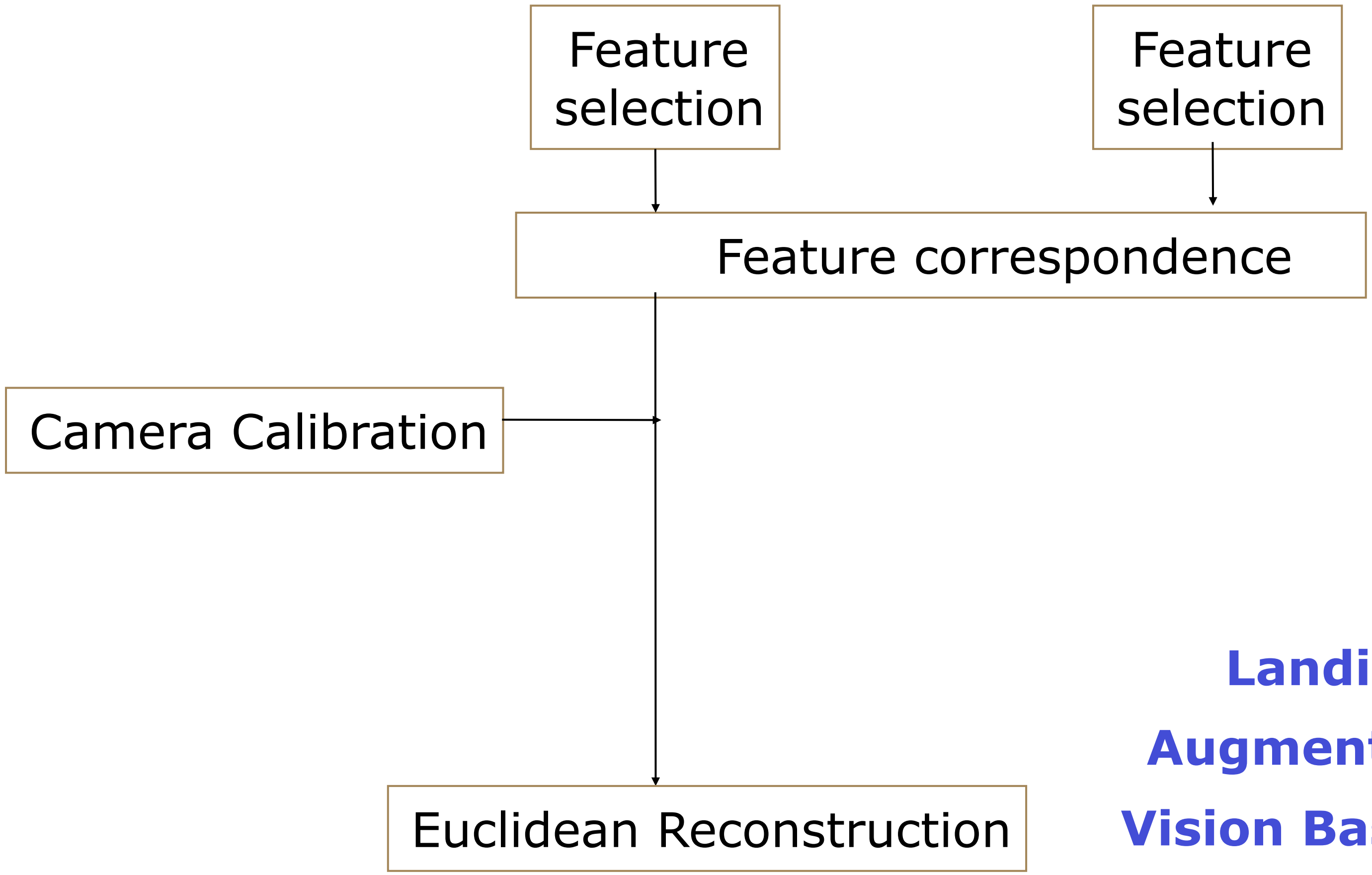
Estimation of 3 rotational angles

Estimation of a translation vector

3D Reconstruction from the Images (Stereo SGM by H.Hirschmüller)



Review



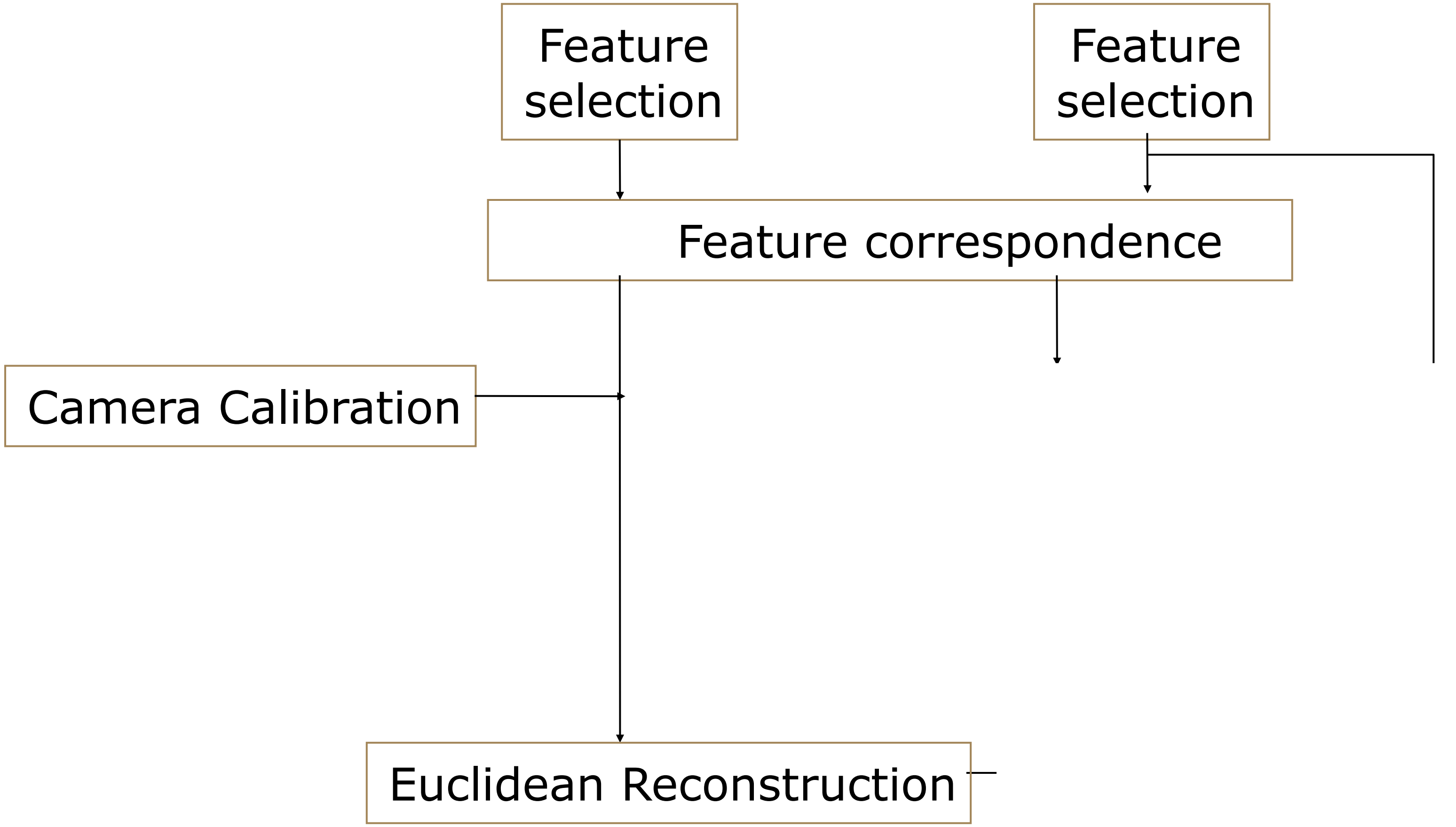
Landing

Augmented Reality

Vision Based Control

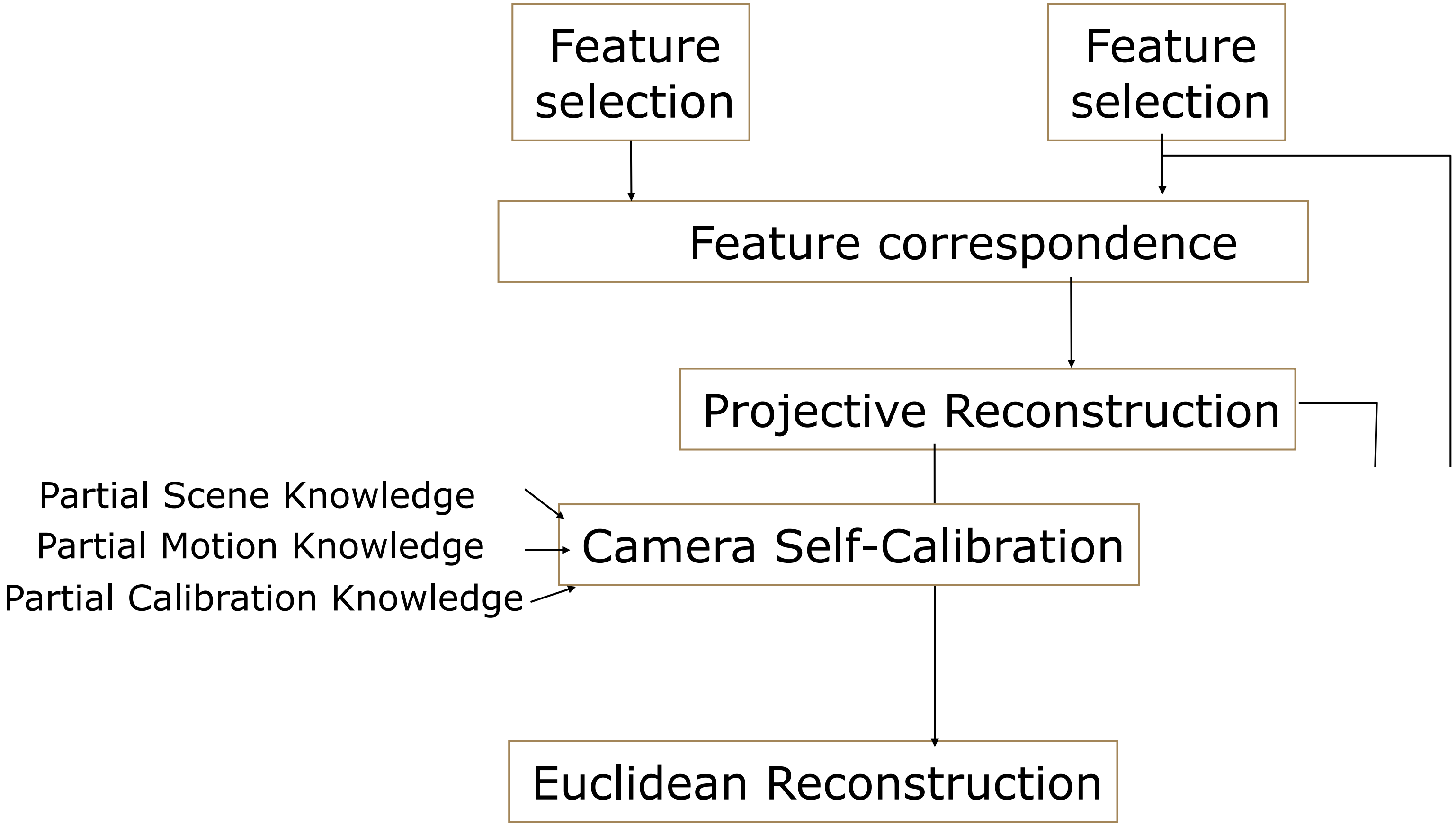
Sparse Structure and camera motion

Review

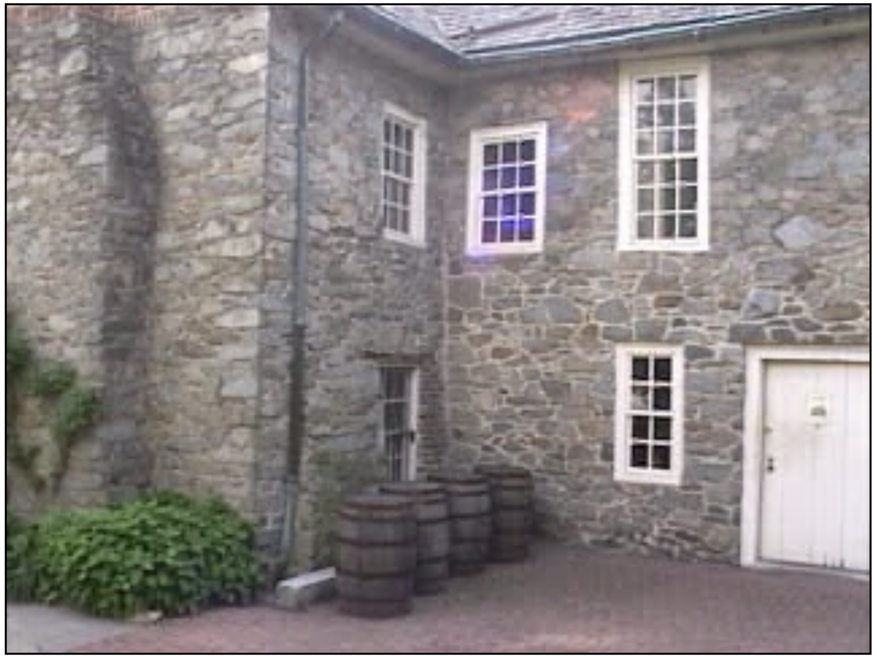


Sparse Structure and motion

Review


Feature selection
Feature selection
Feature correspondence
Projective Reconstruction
Partial Scene Knowledge
Partial Motion Knowledge
Partial Calibration Knowledge
Camera Self-Calibration
Euclidean Reconstruction

Examples

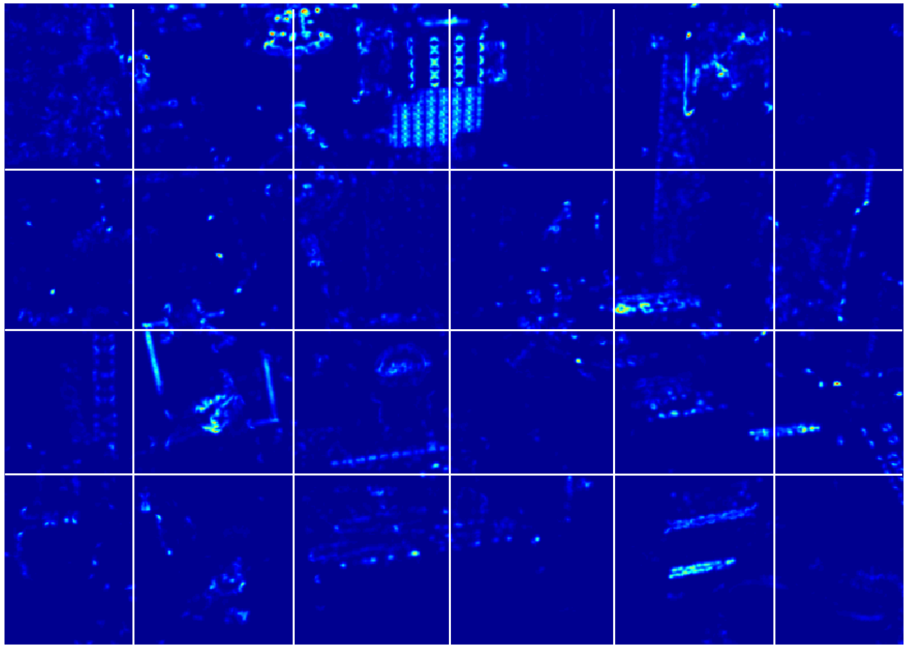


Feature Selection

- Compute Image Gradient $\nabla I^T = [I_x, I_y]$
- Compute Feature Quality $C(\mathbf{x})$ measure for each pixel

$$C(\mathbf{x}) = det(G) + k.\text{trace}^2(G) \qquad G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- Search for local maxima



Feature Quality Function

Local maxima of feature quality function

Feature Tracking

- Translational motion model

$$E(\mathbf{d}) = \min_{\mathbf{d}} \sum_{W(\mathbf{x})} [I_2(\tilde{\mathbf{x}} + \mathbf{d}) - I_1(\tilde{\mathbf{x}})]^2$$

- Closed form solution

$$\mathbf{d} = -G^{-1}\mathbf{b}$$

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

$$\mathbf{b} \doteq \begin{bmatrix} \sum_{W(\mathbf{x})} I_x I_t \\ \sum_{W(\mathbf{x})} I_y I_t \end{bmatrix}$$

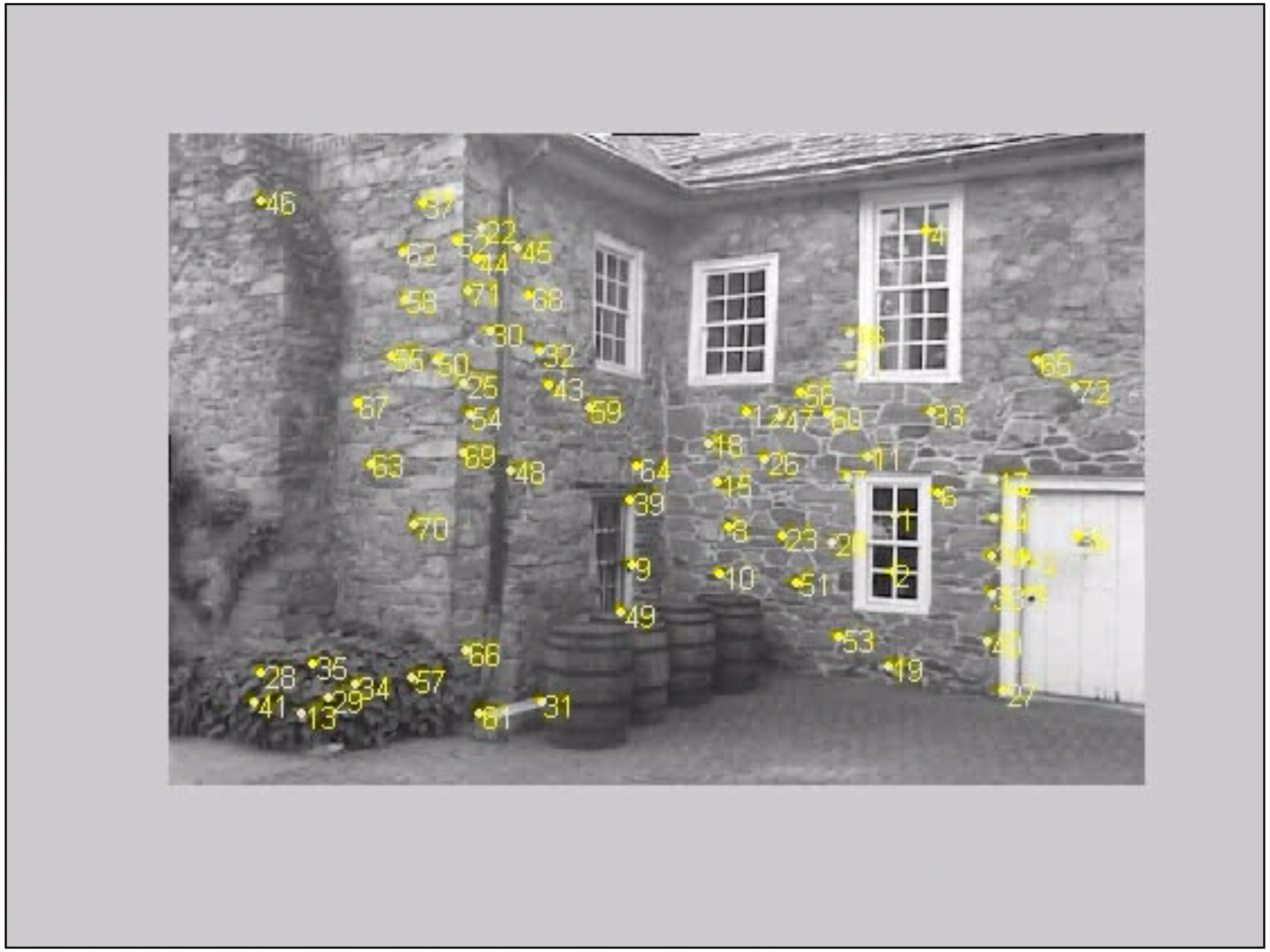
1. Build an image pyramid
2. Start from coarsest level
3. Estimate the displacement at the coarsest level
4. Iterate until finest level

Coarse to fine feature tracking

2 1 0

1. compute $\mathrm{d}_k = -G\mathrm{b}$
2. warp the window $W(\mathrm{x})$ in the second image by $2\mathrm{d}_k$
3. update the displacement $\mathrm{d} \leftarrow \mathrm{d} + 2\mathrm{d}_k$
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

$$NCC(\mathbf{d},\mathbf{x}) = \frac{\sum_{\tilde{\mathbf{x}}\in W(\mathbf{x})}(I_1(\tilde{\mathbf{x}})-\bar{I}_1)(I_2(\tilde{\mathbf{x}}+\mathbf{d})-\bar{I}_2))}{\sqrt{\sum_{\tilde{\mathbf{x}}\in W(\mathbf{x})}(I_1(\tilde{\mathbf{x}})-\bar{I}_1)^2\sum_{\tilde{\mathbf{x}}\in W(\mathbf{x})}(I_2(\tilde{\mathbf{x}}+\mathbf{d})-\bar{I}_2)^2}}$$

Select the candidate with the similarity above selected threshold

RANSAC in action

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