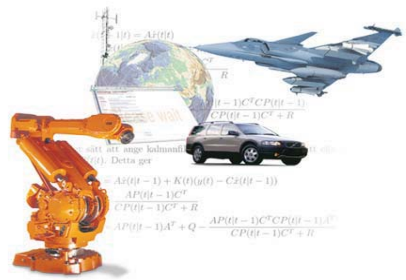
Lecture 3 – Interest Points and Visual Odometry



Thomas Schön,
Division of Automatic Control,
Department of Electrical Engineering,
Linköping University.

Email: schon@isy.liu.se

Camera – A device that provides 2D projections of the 3D world $x = \dfrac{X}{Z},\ y = \dfrac{X}{Z}$

Content – Lecture 3

1. Summary of Lecture 2
2. Correspondence Problem and Interest Points
3. Harris Detector
4. Data Association
5. A First Estimation Problem – Visual Odometry
6. Mars Exploration Rovers

Summary – Lecture 2 (Geometric Camera Models)

The **goal** here is to obtain a transformation from a 3D position in the world frame to the corresponding pixel coordinate.

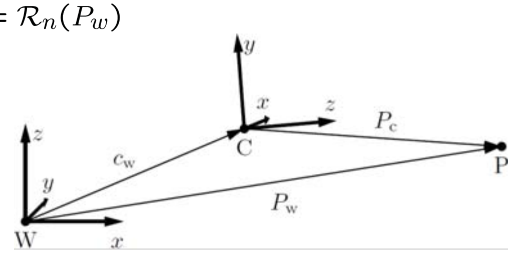
We consider a digital camera to consist of two functional parts

1. An optical system (objective, lens system)
2. An image sensor (CCD/CMOS)

Extrinsic Camera Parameters $P_c = \mathcal{R}_n(P_w)$

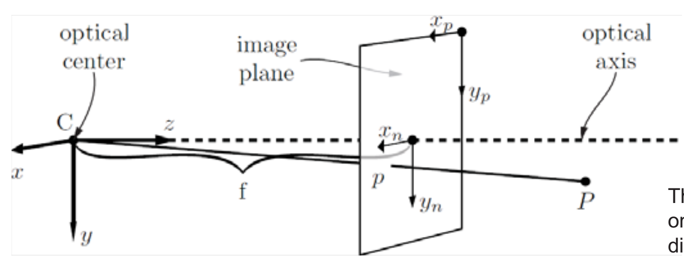
$$P_c = R_{cw}(P_w - c_w)$$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{cw} & w_c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$



Summary – Lecture 2 (Geometric Camera Models)

Normalized Pinhole Model $p_n = \mathcal{P}_n(P_c)$



The camera is a bearings-only sensor providing a direction, **not** a distance.

$$\lambda \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} x_c \\ y_c \end{pmatrix}$$

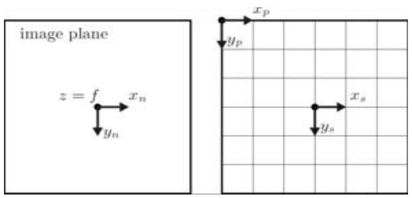
$$\lambda \underbrace{\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix}}_{p_n} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\Pi_0} \underbrace{\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}}_{P_c}$$

Summary – Lecture 2 (Geometric Camera Models)

Lens Distortion $p_d = \mathcal{D}(p_n)$

$$\begin{pmatrix}x_d\\ y_d\end{pmatrix} = \underbrace{(1+a_1r^2+a_2r^4+a_3r^6)\begin{pmatrix}x_n\\ y_n\end{pmatrix}}_{\text{Radial distortion}} + \underbrace{\begin{pmatrix}2a_4x_ny_n + a_5(r^2+2x_n^2)\\ a_4(r^2+2y_n^2)+2a_5x_ny_n\end{pmatrix}}_{\text{Tangential distortion}}$$

Intrinsic Camera Parameters $p_p = \mathcal{K}(p_d)$



$$\underbrace{\begin{pmatrix}x_p\\ y_p\\ 1\end{pmatrix}}_{p_p} = \underbrace{\begin{pmatrix}fs_x & fs_\theta & o_x\\ 0 & fs_y & o_y\\ 0 & 0 & 1\end{pmatrix}}_{K}\underbrace{\begin{pmatrix}x_d\\ y_d\\ 1\end{pmatrix}}_{p_d}$$

Summary – Lecture 2 (Geometric Camera Models)

Summary – Geometric Camera Models

$$\boxed{p_p = \mathcal{P}(P_w) = (\mathcal{K}\circ\mathcal{D}\circ\mathcal{P}_n\circ\mathcal{R})(P_w)}$$

Camera Calibration

Measurements = the corners extracted from the images of the checkerboard pattern $p_p^{ij}, \quad P_w^{ij},$

corner index $i = 1,\dots,M$

$j = 1,\dots,N$ image index

Log – likelihood:

$$\hat{\theta}_{\text{ML}} = \arg\min_{\theta} \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{2}\|p_p^{ij} - \mathcal{P}(P_w^{ij},\theta)\|_{R^{-1}}^2$$

Resulting Measurement Equation

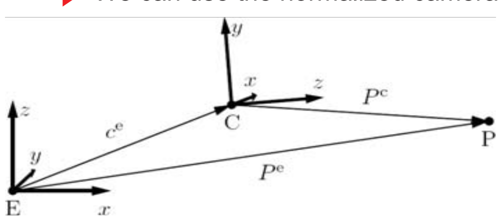
$I_d(u_d, v_d)$ is the distorted image from the camera.

1. Compensate for radial distortion $\rightarrow I(u,v)$
2. Compensate for the intrinsic camera parameters

$$\begin{pmatrix}x\\ y\\ 1\end{pmatrix} = \begin{pmatrix}fs_x & 0 & o_x\\ 0 & fs_y & o_y\\ 0 & 0 & 1\end{pmatrix}^{-1}\begin{pmatrix}u\\ v\\ 1\end{pmatrix}$$

→ We can use the normalized camera model,

$$\boxed{y_t = \frac{1}{Z}\begin{pmatrix}X\\ Y\end{pmatrix} + e_t}$$



Map

Camera pos. (state)

$$\begin{pmatrix}X\\ Y\\ Z\end{pmatrix} = R^{ce}(P^e - c^e)$$

Camera ori. (state)

Hence, this is just a standard nonlinear measurement equation!!

The Correspondence Problem

Given a point in the 3-D world the *correspondence problem* amounts to finding its 2-D projection in two different images.

That is, find pairs of pixels (one in each image) that corresponds to the same point in the 3-D world.

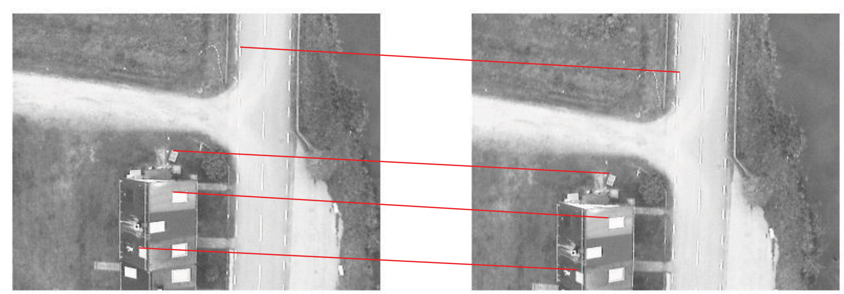


Image 1 Image 2

Interest Point (aka features)

An interest point is a point in the image which has a clear, preferably mathematically well-founded, definition and a well-defined position in the image.

Further attractive properties of interest points include:

- The local image structure around the interest point is rich in terms of local information contents.
- It is stable under perturbations in the image, such as scale changes, rotations and/or translations as well as illumination variations.

(Harris, Förstner, Shi Tomasi, SUSAN, FAST, blobs, SIFT, SURF, GLOH, LESH, …)

Image Gradients

$$\nabla I(x_p, y_p) = \begin{pmatrix} I_x(x_p, y_p) \\ I_y(x_p, y_p) \end{pmatrix},$$
$$I_x(x_p, y_p) = \frac{\partial I}{\partial x}(x_p, y_p),$$
$$I_y(x_p, y_p) = \frac{\partial I}{\partial y}(x_p, y_p).$$

Original image $I(x_p, y_p)$

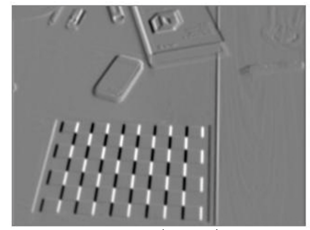


Image gradient $I_x(x_p, y_p)$

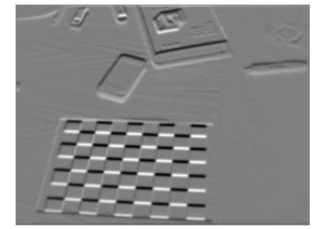
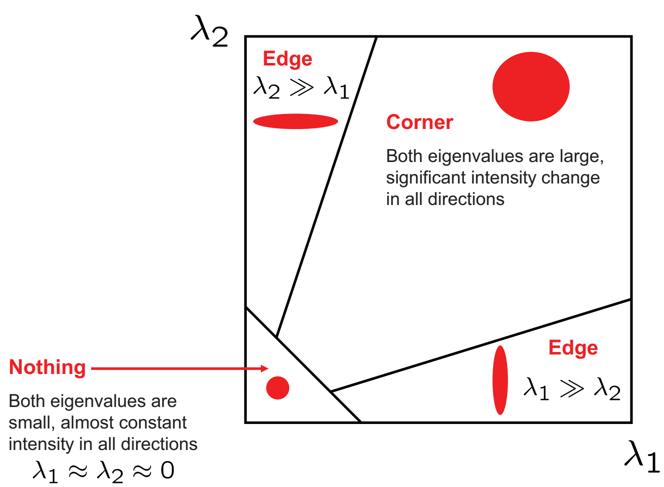
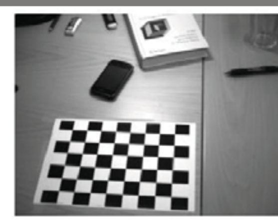


Image gradient $I_y(x_p, y_p)$

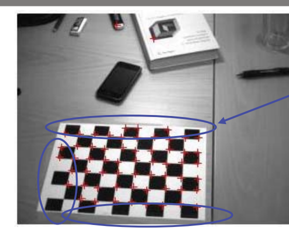
Eigenvalues of H



Harris Detector



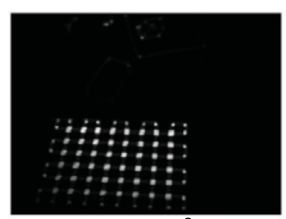
Original image



Extracted Harris corners

Several corners are missing!

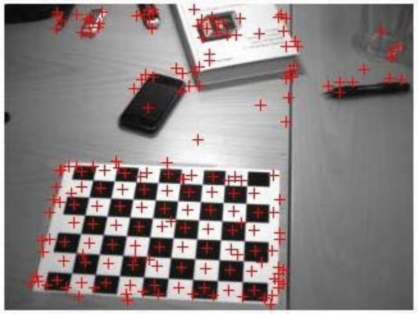
Use a lower threshold



$s = \det(H) + k\,\mathrm{tr}^2(H)$

Extracted Harris corners, lower threshold

SIFT and SURF



Extracted SIFT

D. G. Lowe, **Distinctive image features from scale-invariant keypoints**, *International Journal of Computer Vision,* 60(2): 91–110, 2004.

http://www.cs.ubc.ca/~lowe/keypoints/

http://vision.ucla.edu/~vedaldi/code/sift/sift.html

SURF features

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. **SURF: Speeded Up Robust Features**, *Computer Vision and Image Understanding*, 110(3):346-359, 2008.

http://users.student.lth.se/p04pst/surfmex.html

New interest point detectors and descriptors appear constantly, we view them as tools.

Visual Odometry Using Interest Points

Conceptual solution:

1. Initialization
2. Acquire an image and correct for possible lens distortion
3. Predict the positions of the previously detected map entries
4. Data association, match the predicted interest points to the corresponding descriptors obtained in the new image
5. Remove outliers
6. Search for new interest points in areas where there are no interest points
7. Update the state estimates
8. Repeat from 2

Obviously there are many different choices available here, but the overall structure is given above.

1. Initialization

Assume that there is a few entries present in the map, for instance the four corners of a black rectangle. This is one way of introducing a the scale!

An alternative way to obtain the scale is by using additional sensors, again, this is the topic of lecture 5. For now, only camera.

The positions of the four entries are given in the world coordinate system.

Davison, A. J., I. D. Reid, N. D. Molton and O. Strasse, **MonoSLAM: real-time single camera SLAM**, *IEEE Transactions on pattern analysis and machine intelligence*, 29(6):1-16, Jun. 2007.

3. Prediction Interest Point Position

The geometric camera model,

$$p_p = \mathcal{P}(P_w) = (\mathcal{K} \circ \mathcal{D} \circ \mathcal{P}_n \circ \mathcal{R})\,(P_w)$$

Recall that the lens distortion is compensated for, $\mathcal{D} = I$

$$\lambda \underbrace{\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix}}_{p_p} = \underbrace{\begin{pmatrix} f_x s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R_{cw} & w_c \\ 0 & 1 \end{pmatrix}}_{\Pi} \underbrace{\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}}_{P_w}$$

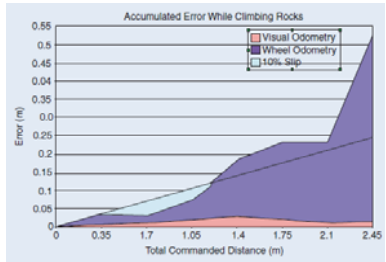
$$\lambda p_p = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} P_w$$

The prediction of the interest point position in pixels is given by

$$x_p = \frac{\pi_1 P_w}{\pi_3 P_w}, \qquad y_p = \frac{\pi_2 P_w}{\pi_3 P_w}$$

Mars Exploration Rovers

You now have sufficient knowledge to implement the visual odometry currently used on Mars!



marsrovers.jpl.nasa.gov

Cheng, B. Y., M. W. Maimone and L. Matthies, **Visual Odometry on the Mars Exploration Rovers,** *IEEE Robotics and Automation Magazine*, 13(2):54-62, Jun. 2006.

Normalized Cross-Correlation (Functional Analysis)

Define the two normalized vectors according to

$$S(x_p, y_p) = \frac{s(x_p, y_p) - \bar{s}}{\|s(x_p, y_p) - \bar{s}\|_2}$$
$$P(x_p, y_p) = \frac{p(x_p, y_p) - \bar{p}}{\|p(x_p, y_p) - \bar{p}\|_2}$$

The normalized cross-correlation is then given by the scalar (inner) product between the two normalized vectors,

$$R(x_p, y_p) = \langle S(x_p, y_p), P(x_p, y_p) \rangle$$