Invariance, Keypoints, and RANSAC[1]

Lecture 22

See Related Material in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

[1] See last slide for copyright information.

Keypoint, Descriptor, Feature

A keypoint and a descriptor define a feature

Examples: SIFT, SURF, and ORB

BRIEF and FAST are needed for defining ORB

Features are tracked over time: KLT, particle filter, or Kalman filter are possible tools

Left: DoG scale space keypoints
Right: Keypoints with disks of influence (radius = scale where keypoint has been detected

Agenda

1. Invariance, Features, and Sets of Features
2. Keypoints and 3D Flow Vectors
3. Random Sample Consensus

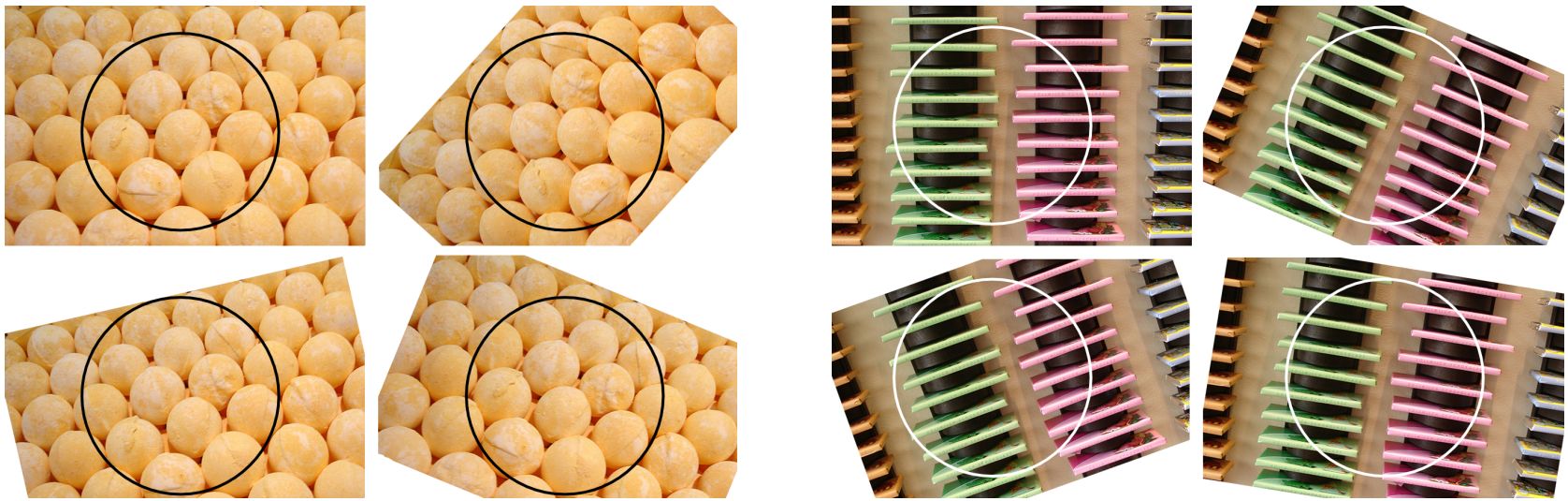
Invariance

Images are taken under varying illumination, different viewing angles, at different times, under different weather conditions, and so forth

When taking an aerial shot from an airplane, we do have a random rotation of shown objects, and *isotropy* (rotation invariance) is of interest

In outdoor scene analysis, we often request types of invariance with respect to some operations, such as illumination changes or recording images at different distances to the object of interest

Illustration



Recorded scene itself may support invariance (e.g. isotropy by the scene on the left)

A Procedure $\mathcal{X}$

We have input images I of scenes $S \in \mathcal{S}$ and a camera (i.e. an imaging process) C

For images $I = C(S)$, a defined analysis procedure $\mathcal{X}$ maps image I into some (say) vectorial output $R(I) = \mathbf{r}$, the *result*

For example, this can be a list of detected features

Altogether:

$$R(I) = R(C(S)) = \mathbf{r}$$

Invariance w.r.t. Changes in the Scene

We have a change in the recorded scene S due to object moves, lighting changes, a move of the recording camera, and so forth

This defines a new scene $S_{new} = N(S)$, with $I_{new} = C(S_{new})$

Procedure $\mathcal{X}$ is invariant to the change N (in an ideal way) if we obtain with

$$R(I_{new}) = R(C(N(S))) = \mathbf{r}$$

still the same result $\mathbf{r}$ for I_{new} as we had for I before

Example. Change N defined (only) by a variation in lighting within a defined range of possible changes

Then $\mathcal{X}$ *invariant to illumination changes* in this particular range

Invariance w.r.t. Used Camera

Assume a modification M in the imaging procedure I (e.g., the use of a different camera, or just of a different lens), $C_{mod} = M(C)$, with $I_{mod} = C_{mod}(S)$

Procedure $\mathcal{X}$ is invariant to the modification M if we obtain with

$$R(I_{mod}) = R(M(C(I))) = \mathbf{r}$$

still the same result $\mathbf{r}$ for I_{mod} as we had for I before

Agenda

1. Invariance, Features, and Sets of Features
2. Keypoints and 3D Flow Vectors
3. Random Sample Consensus

Keypoint

A *keypoint* (or *interest point*) is defined by some particular image intensities "around" it, such as a corner

A keypoint can be used for deriving a *descriptor*

Not every keypoint detector has its particular way for defining a descriptor

A descriptor is a finite vector which summarizes properties for the keypoint

A descriptor can be used for classifying the keypoint

Keypoint and descriptor together define a *feature*

Four Examples

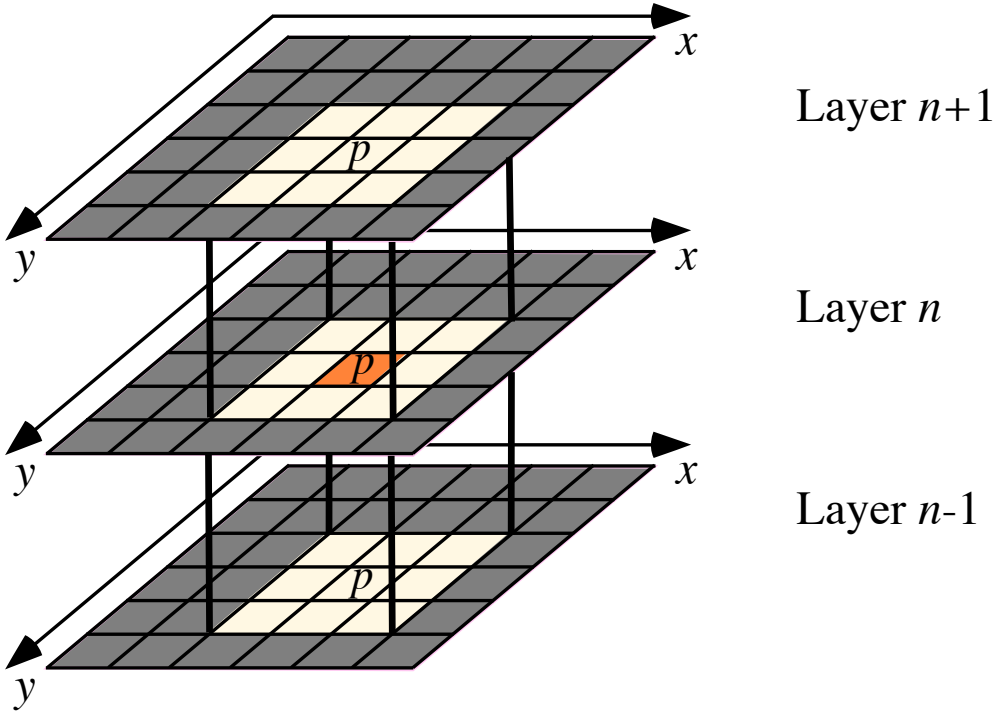
Four of the keypoint detectors in `OpenCV`: FAST, ORB, SIFT, SURF

Keypoints Defined in LoG or DoG Scale Space

Recall: DoG for scale σ and scaling factor $a > 1$ for combining two subsequent layers in Gaussian scale space:

$$D_{\sigma,a}(x, y) = L(x, y, \sigma) - L(x, y, a\sigma)$$

Use initial scale $\sigma > 0$; apply scaling factors a^n, $n = 0, 1, 2, \ldots$ for generating a finite number of Layers n in DoG scale space



3D Data Array and Disk of Influence

Layers D_{σ,a^n}, $n = 0, \ldots, m$, define a 3D data array

Each array position (x, y, n) in this 3D array has 17 or 26 adjacent array positions

Array position (x, y, n) and those 17 or 26 adjacent positions define the *3D neighborhood* of (x, y, n)

A *keypoint* is detected at $p = (x, y)$ if there is a Layer n, $0 \leq n \leq m$, such that $D_{\sigma,a^n}(x, y)$ defines a local minimum or local maximum within the 3D neighborhood of (x, y, n)

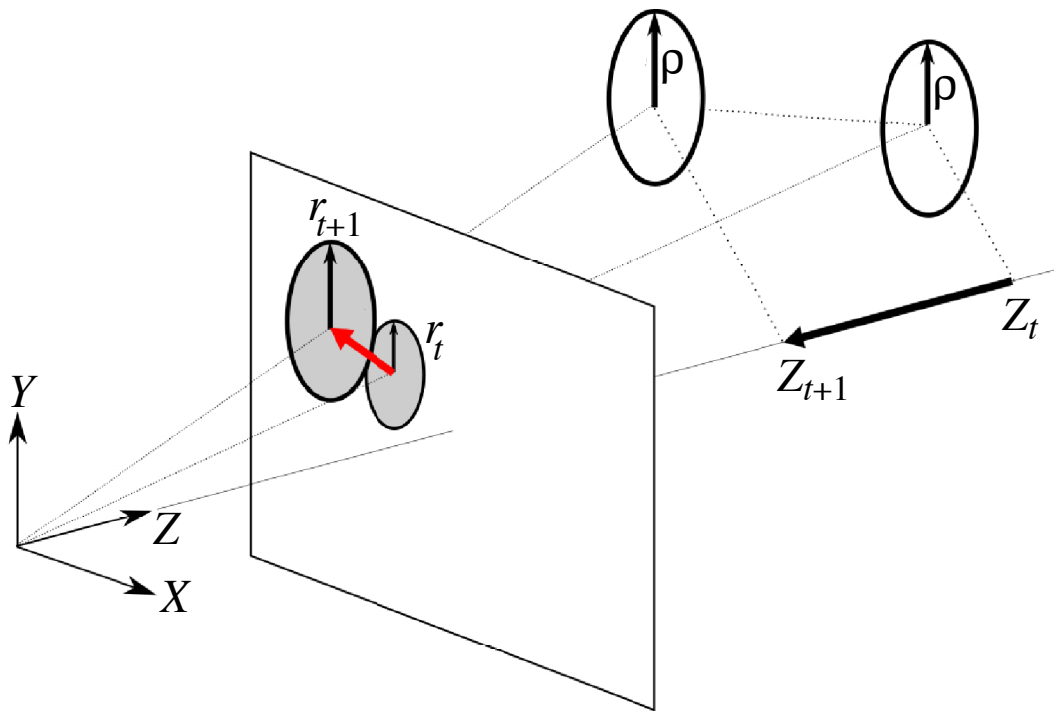
With a detected keypoint in the original image I at a pixel location $p = (x, y)$, we also have the scale $\sigma \cdot a^n$ where it has been detected

This scale defines the radius of the *disk of influence* for this keypoint p

3D Flow Vectors

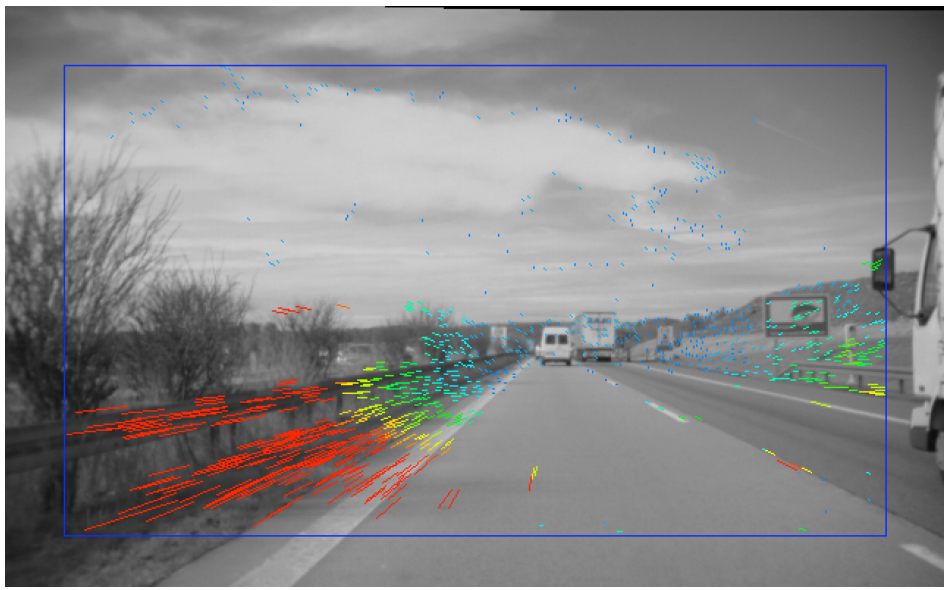
Given: Sequence of frames $I(.,.,t)$, and detected keypoints in scale space, together with radius of disks of influence

Also given: a way to solve the keypoint-correspondence problem between keypoints in frames $I(.,.,t)$ and $I(.,.,t+1)$ (some keypoints may not have a corresponding keypoint in the next frame)



Moving Disks

Slide before: Illustration of a disk in 3D space moving towards the image plane



2D projections of detected 3D flow vectors; the color key used represents different directions and magnitudes of motion in 3D space

3D Flow Calculation: Basic Idea

Consider keypoint $p_t = (x_t, y_t)$ in frame $I(.,.,t)$ with radius $r_t > 0$ of its disk of influence

Moves into keypoint $p_{t+1} = (x_{t+1}, y_{t+1})$ in frame $I(.,.,t+1)$ with radius $r_{t+1} > 0$ of its disk of influence

Increase in radius, from r_t to r_{t+1}, is inversely proportional to the speed that the center point of this local "circular situation" is moving towards the camera

If center point moves away then projected radius increases

Formulas

Recall: focal length f of the camera; $P = (X, Y, Z)$ projects into a point $p = (x, y, f)$ with $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$

Disk moves parallel to the XY-plane of the XYZ-camera coordinate system; radius parallel to Y-axis from Y_c to Y_e, from center point P_c to end point P_e

P_c projects into $p_c = (x_c, y_c, f)$ and P_e into $p_e = (x_e, y_e, f)$

Disk at time t at distance Z_t projected into image $I(.,.,t)$ as disk with radius r_t and area

$$\mathcal{A}_t = \pi r_t^2 = \pi (y_c - y_e)^2 = f\frac{\pi}{Z_t^2}(Y_c - Y_e)^2 = \pi f\frac{\rho^2}{Z_t^2}$$

Robust Estimator

Radius ρ of the disk is constant over time

Product $\mathcal{A}_t Z_t^2 = \pi f \rho^2$ does not change over time

$$\frac{Z_{t+1}}{Z_t} = \sqrt{\frac{\mathcal{A}_t}{\mathcal{A}_{t+1}}}$$

This is a robust estimator for the ratio of distance values

Keypoints at Subpixel Accuracy

Keypoints detected in scale space in a layer defined by scale $\sigma \cdot a^n$

Are at pixel locations (i.e. with integer coordinates)

1. Interpolate a 2D second-order polynomial $g(x, y)$ to the detected keypoint and its four 4-adjacent neighbors
2. Take the derivatives of $g(x, y)$ in x- and y-direction
3. Solve the resulting equational system for a subpixel-accurate minimum or maximum

Corresponding Keypoints

Given: Sets of keypoints in subsequent frames

Compare detected sets of keypoints in two subsequent frames of an image sequence using locations and descriptors

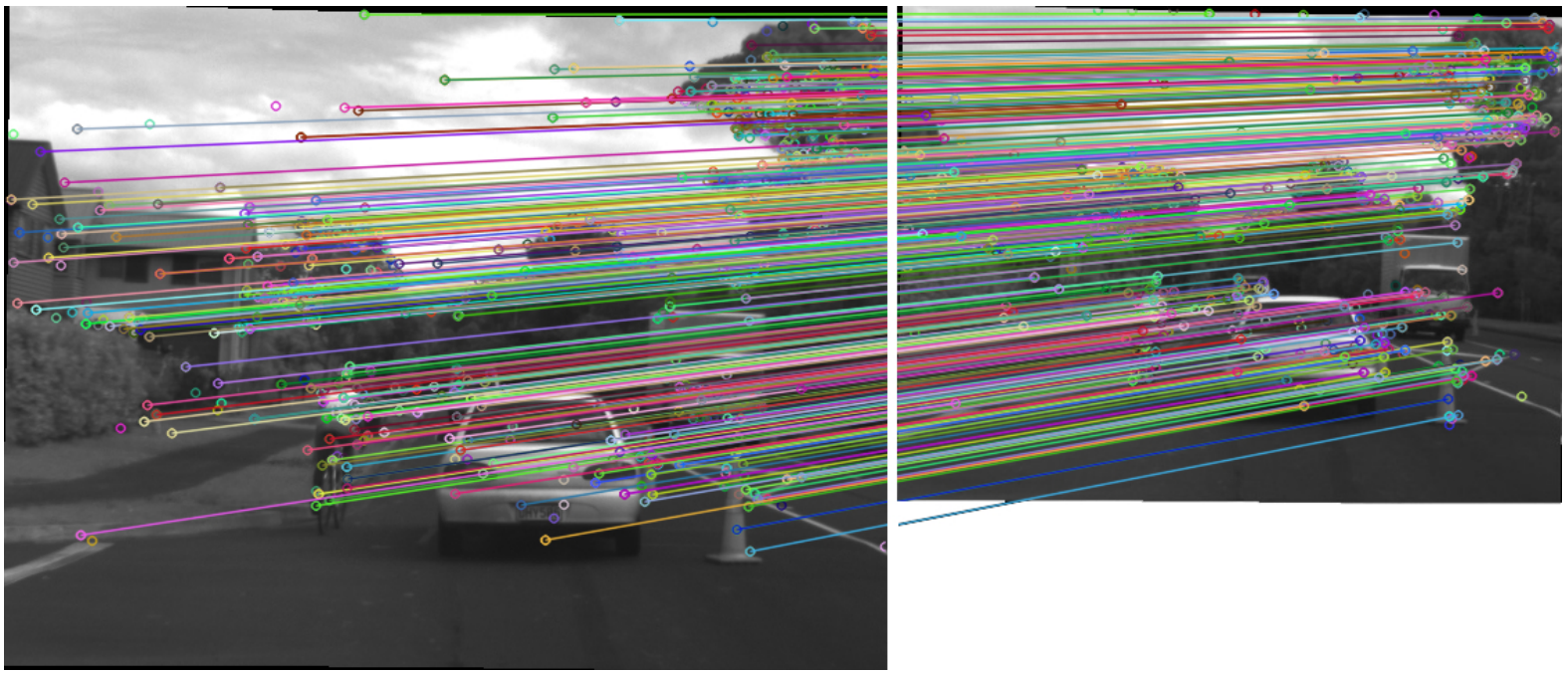
Find corresponding keypoints

There will be *outliers* which have no corresponding keypoint

Inliers have corresponding points

Correspondence e.g. also a (global) set problem (not only a point-by-point problem): There is a global pattern of keypoints, and we want to match this global pattern with another global pattern of keypoints

SIFT Example



Sets of SIFT keypoints in original and demagnified image (not a time sequence)

RANSAC-match of corresponding keypoints

Agenda

1. Invariance, Features, and Sets of Features
2. Keypoints and 3D Flow Vectors
3. Random Sample Consensus

Random Sample Consensus

RANSAC is an iterative estimation technique of parameters of an assumed mathematical model

Given is a set of data, called *inliers*, which follow the model, but there is also additional data, called *outliers*, which do not follow the model

For applying RANSAC, the probability of selecting inliers needs to be reasonably high

Example 1: Inliers and outliers as a noisy representation of a straight line $y = ax + b$

Task: estimate a and b (an alternative to Hough transform)

Example 2: Sets of keypoints in two different images; the model is a geometric transform (example of a *matching problem*), e.g. an affine transform

Test

Need a test for evaluating whether data *satisfy* or *fit* the parameterized model

Here: keypoint p in one image I is mapped by parameterized affine transform onto q in other image J

Test: If there is a keypoint r in J at distance $d_2(q, r) \leq \varepsilon$ then we say that p satisfies the given parameterized affine transform

Tolerance threshold $\varepsilon > 0$ determines whether data fit the model

RANSAC Algorithm

Initialization: select random subset S of given data as inliers; fit the model by estimating model parameters

Test: Test the parameterized model for *all* the other data

Define consensus set: All the data which satisfy the model go into a *consensus set*

Cardinality check: Compare the cardinality of the consensus set against the cardinality of all data

Stop criterion: Percentage is reasonably high

Refined model: Otherwise, estimate updated model parameters based on consensus set

Continue with refined model: if cardinality of newly established consensus set does not increase then go back to initialization step and select another random subset S

RANSAC for Feature Matching

Feature defined by keypoint and descriptor

Initial set S can be three randomly selected keypoints in image I

Search for three keypoints with reasonably matching descriptors in image J

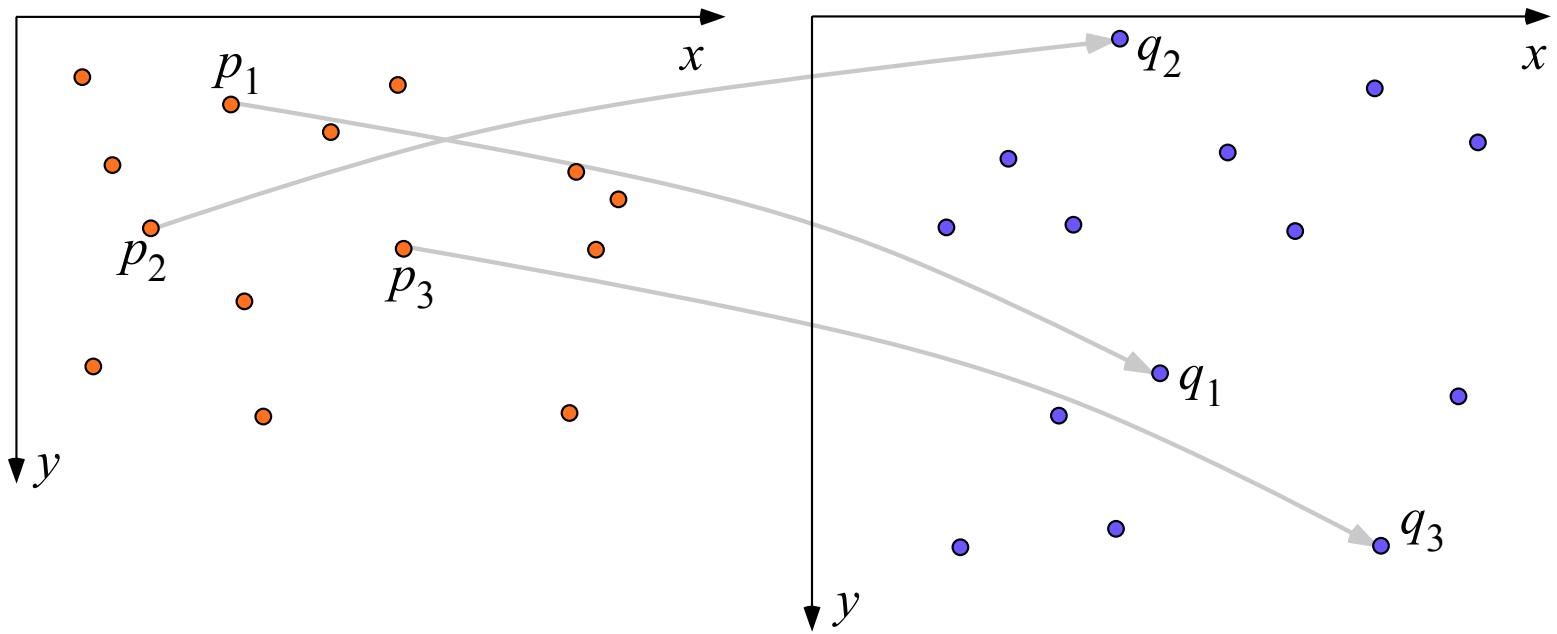
Estimate affine transform: Point $p = (x, y, 1)$ in homogeneous coordinates in image I is mapped into $q = (u, v, 1)$ in image J

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{aligned} u &= r_{11}x + r_{12}y + t_1 \\ v &= r_{21}x + r_{22}y + t_2 \end{aligned}$$

Six unknowns, solvable for non-collinear points p_1, p_2, and p_3 in I

Three Pairs



Definition of an affine transform by three pairs of points in images I (on the left) and J

Consensus Set

For calculated affine transform $A(p) = q$, we apply A now to *all* the keypoints p in I

We obtain points $A(p)$ in J

Point p goes into the consensus set if there is a keypoint q in J in a Euclidean distance less then $\varepsilon > 0$ to $A(p)$ with a "reasonable" match of descriptors, defining the expected image q_p of p in J

Obviously, initially used points p_1, p_2, and p_3 pass this test

Update

In general, consensus set now with more than just 3 points in I

Update the affine transform by calculating (using linear least-squares) the optimum transform for all the established pairs p in I and q_p in J

Defines a refined affine transform

Value ε cannot be "very small"; this would not allow to move away from the initial transform (for a better match between both sets of keypoints)

Estimating a plane in a 3D Cloud of Points

Assume a stereo camera system in a quadcopter

Needs to understand a plane at the time of landing

At time t we record a left and right image, I_L and I_R, respectively

Apply a keypoint detector (e.g. a time-efficient detector such as FAST), providing sets M_L and M_R of keypoints

For each keypoint p in M_L aim at detectin a matching keypoint in M_R (if it exists), defined by a disparity d

Feature descriptors can help to identify a match

Having a projection matrix $\mathbf{P}$ for the left camera, we can map p based on d and $\mathbf{P}$ into a point P in 3D space

3D points P are define set S; fit a plane into the set S using RANSAC

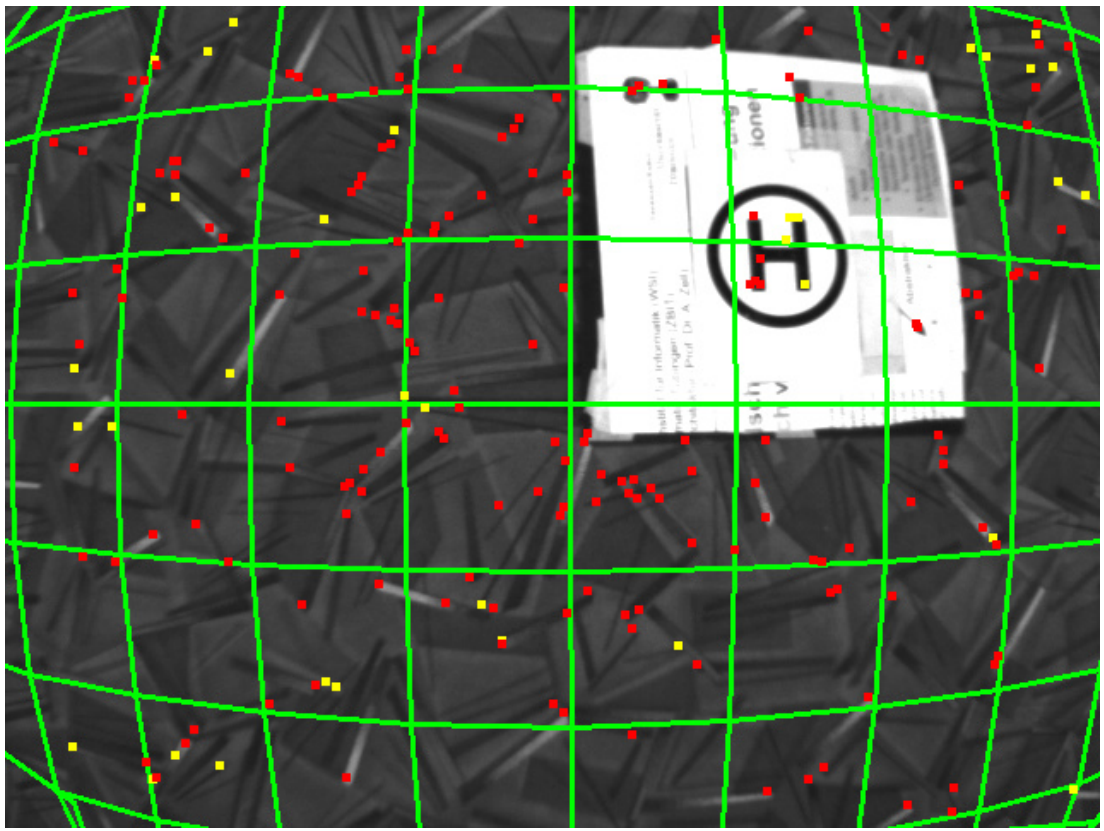
Pseudocode

1: $M_L = \texttt{matchFeatures}(I_L)$ {keypoints in left image};
2: $M_R = \texttt{matchFeatures}(I_R)$ {keypoints in right image};
3: $S = []$ {empty list; will store the 3D points};
4: **for** p in M_L **do**
5: $\quad d = \texttt{findDisparity}(p, M_R)$;
6: $\quad P = (p.x, p.y, d) \cdot \mathbf{P}$ {project detected point pair into 3D by multiplying with projection matrix};
7: $\quad \texttt{append}(S, P)$ {add the projected 3D point P to the set of 3D points};
8: **end for**
9: $\Pi_1 = \texttt{ransacFitPlane}(S)$;
10: $\Pi_2 = \texttt{ransacRefinePlaneModel}(S, \Pi_1)$;

Fitting a plane into sparsely detected 3D points.

Illustration

Estimated plane can be used by the quadcopter for control while landing on a planar surface



Sparse stereo matcher based on FAST; yellow points are outliers
Back-projected plane is curved due to lens distortion

Copyright Information

This slide show was prepared by Reinhard Klette
with kind permission from Springer Science+Business Media B.V.

The slide show can be used freely for presentations.
However, *all the material* is copyrighted.

In case of citation: just cite the book, that's fine.