

COMP3204 - Computer Vision Notes

Bradley Mason
University of Southampton

November 16, 2016

Contents

L1 - Building Machines That See	3
L7 - Local Features and Matching	4

Building Machines That See

A CCD is better than CMOS for sensitivity and noise.

Local Features and Matching

Local Features and Matching Basics

Local feature points are used for:

- Image alignment
- Camera pose estimation and camera calibration
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation

A good example is panoramas, we need to **match** and **align** the images. You detect feature points in both images, find the corresponding pairs and then use those pairs to align the images.

We can use two distinct types of matching problem.

In stereo vision (vision with two cameras for 3D reconstruction) there are two important concepts related to matching:

- Narrow-baseline stereo
- Wide-baseline stereo

In **Narrow Baseline Stereo** you have two images very close together. Similar to how human vision works. This is applicable to tracking where the object doesn't move too much between frames.

In **Wide Baseline Stereo** you have two images that are far apart and likely at completely different angles too. This is applicable to generic matching tasks like object recognition.

Robust Local Description

The requirements of a descriptor are very much dependent on the task at hand.

A narrow baseline is for when you need robustness to rotation and lighting is not so important. The descriptiveness

can be reduced as search is over a smaller area.
A wide baseline is for when you need robustness to intensity change and invariance to rotation. You have to be highly descriptive to avoid mismatches but not so distinctive that you can't find any matches. It is also robust to small localisation errors of the interest point, the descriptor shouldn't change too much if we move by a few pixels but to change rapidly once we move further away.

Matching by Correlation

A.K.A Template Matching.

For narrow baseline you may want to find the same point in two images that has shifted slightly. It can search the local area to match the template against the second image. Wide baselines are more tricky because they are not robust to rotation and you can't assume there will be a definable search area so you have to consider the entire second image which is likely to mismatch.

Local Intensity Histograms

You can describe the region around an interest point with a pixel histogram. You can then measure the euclidean distance between these histograms to find the matching interest point.

The problems with this is that it isn't very distinctive (remember the space image shuffled histogram being the same). It is also not rotation invariant if the sampling window is square or rectangular, which can be overcome with a circular window.

It is also not invariant to illumination changes and sensitive to interest point localisation.

You can overcome the localisation sensitivity by applying a weighting so pixels near the edge of the sample area have less effect than those near the interest point. It is very common to use Gaussian weighting centred on the interest point for the weighting.

It is possible to make it illumination invariant by normalising or equalising the pixel patches before constructing the histogram or by using **Local Gradient Histograms**

Local Gradient Histograms

If you find the partial derivative of an image you can then compute the gradient orientations/directions and magnitudes.

$$\theta = \text{atan2}\left(\frac{\delta f}{\delta y}, \frac{\delta f}{\delta x}\right) \quad m = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2}$$

Instead of building histograms of the raw pixels values we can build histograms that encode the gradient magnitude and direction for each pixel in a sampling patch.

Gradient magnitudes and directions are **invariant to brightness** change!

The gradient magnitude and direction histogram is also **more distinctive**.

In order to build these histograms you quantise the directions (0-360) into a number of bins (usually 8). Then for each pixel in the sampling patch you accumulate the gradient magnitude of that pixel in the respective orientation bin.

By default this is not rotation invariant however it can be by finding the dominant orientation and cyclically shifting the histogram so the dominant orientation is the first bin.

The SIFT Feature

SIFT - Scale Invariant Feature Transform is very widely used.

It builds on the idea of a logical histogram by incorporating **spatial binning**, which in essence creates multiple gradient histograms about the interest point and appends them into a longer feature.

Standard SIFT appends a spatial 4x4 grid of histograms with 8 orientations. This creates a 128 dimension feature which is both highly **discriminative** and **robust**!

To construct this you find your interest point and take a sampling patch (commonly proportional to the scale of interest point). You then apply a Gaussian weighting centred around the interest point, this effectively reduces the corner to a 0 weighting changing this to a circular sample.

You then take your 4x4 spatial bins and produce a gradient histogram for each using the gaussian weighting. The orientation is measured relative to the overall dominant orientation of the patch.

Matching SIFT Features

The simplest way is to find the Euclidean distance and find the most similar in the second image. You can use a threshold to reject poor matches but unfortunately this also gives a lot of mismatches.

It is better to take each feature and find the the two closest matches from the second images, only accept the first closest if the ratio between the two top matches is less than some threshold (typically 0.8 works best).

This makes it much more robust.