COMP3204/COMP6223 - Computer Vision

7

Local features and matching

Summary

In the last lecture, we looked at the problem of finding stable, repeatable points in an image. Now we need to look at what can be done in order to extract *local* features from these interest points. A number of techniques have been proposed in the past in order to extract robust local descriptors, starting from simple pixel histograms, through to advanced features like the SIFT descriptor, which encode weighted local image gradients. Once these descriptors have been extracted, they can be used for things such as image matching, and as we'll see in later lectures for *image retrieval* and *image classification*.

Key points

Local feature matching basics

- Basic idea is to find all local features in one image that have correspondences in another image.
    - Applications include image retrieval, 3d reconstruction, building panoramas, tracking, object recognition, etc.
    - In stereo vision (i.e. for 3d reconstruction), there are two concepts:
        - *Narrow-baseline* stereo is where the two images are very similar – the local features have only moved by a few pixels.
            - Typically the images are from similar points in time.
        - *Wide-baseline* stereo is where the difference in views is much bigger.
    - These concepts extend to general matching: the techniques for narrow-baseline stereo are applicable to tracking where the object doesn't move too much between frames; and the techniques for wide-baseline stereo are applicable to generic matching tasks (object recognition, panoramas, etc.).

Robust local description

- The type of descriptor required is dependent on the task.
    - For narrow baseline applications, we can assume that not much will have changed between two images:
        - The corresponding interest points will have only moved by a few pixels
            - Rotation isn't much of an issue because the changes are small
            - Descriptiveness of the descriptor isn't too important, as it only needs to be compared to potential targets a small distance away
        - Lighting is unlikely to have changed much
    - For wider baselines, ideally, we want local descriptors to have the following attributes:
        - Robustness to uniform intensity changes in the pixel values around the interest point
        - Invariance to the orientation of the image/camera/scene (the interest points are typically invariant to rotation, and we'd like the features not to change as the image is rotated).
        - Robustness to the placing of the interest point by a few pixels
            - We'd like the descriptor to not change too much if we move it by a few pixels, but to change more rapidly once we more further away.

        - The descriptors for visually different local regions should be unique and be far apart from each other in feature space (i.e. highly descriptive and highly discriminative).

Matching by correlation (template matching)

- Most obvious way to describe a local (usually rectangular or square) region around an interest point to use the pixel values directly.
    - Basic template matching (correlation or Sum-Squared-Difference) can be used to perform matching.
    - Can work very well when there are small differences, as the search area can be limited.
    - Tends not to work well in wide-baseline scenarios
        - Not rotation and lighting invariant; sensitive to position of interest point.

Local Intensity Histograms

- Instead of comparing the raw pixels, an alternative is to represent the local region by a histogram of the pixel intensities.
    - If the sampling window is circular, the histogram will be (mostly) rotation invariant.
        - But it will be sensitive to small changes in position of the interest point…
        - Clever trick: If pixels in the sampling window are weighted by their distance from the interest point, so that pixels further away have less effect, then this can be overcome.
            - A Gaussian weighting is common: pixels near the centre have high weights, and those further away have much lower ones.
    - Not invariant to illumination changes.
    - But… the histograms won't be very distinctive (the histograms around any given interest points will all look similar).

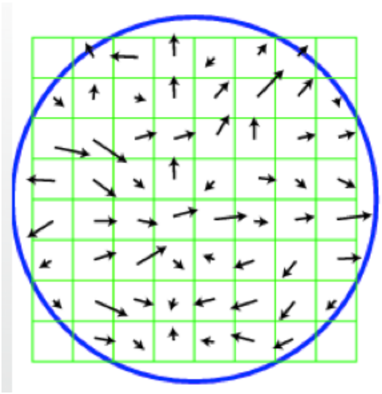
Local Gradient Histograms

- Instead of encoding raw pixel magnitudes in a histogram, we could encode gradient directions within a window.
    - It's easy to compute gradient directions in the horizontal (x) and vertical (y) directions using finite differences, or filters like Sobel.
    - From the x and y gradient maps, you can compute the gradient direction at a given pixel in the image *f*, as

      $$\theta = \mathrm{atan2}(\frac{\delta f}{\delta y}, \frac{\delta f}{\delta x})$$

    - And the magnitude of the gradient as

      $$m = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2}$$

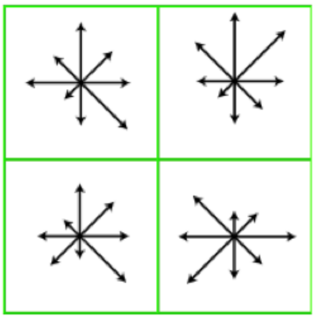


    - An orientation-magnitude histogram can be constructed by quantizing the range of possible orientations (0-360) into a fixed number of bins (often around 8), and summing the corresponding magnitudes for each orientation at each pixel in the window.
        - (Gaussian) weighting can be applied to down-weight the influence of magnitudes far from the interest point, giving invariance to small perturbations in the interest point location.
        - Gradient magnitudes are invariant to uniform intensity changes.
        - But… the histograms are **not rotation invariant**!
            - By robustly computing the "*dominant orientation*" of the window around the interest point and subtracting this from each pixels' orientation before building the histogram will result in a feature that is invariant to rotation.

Written by Jonathon Hare. Send any comments/bugs/typos to jsh2@ecs.soton.ac.uk.

- Robust determination the dominant orientation can be achieved by building the orientation-magnitude histogram and searching for the biggest peak.
    - Often a quadratic will be fitted over the bin with the peak, and the bins to the left and right, in order to get a high quality estimate of the dominant orientation.

The SIFT feature



- The SIFT feature takes the idea of a orientation-magnitude histogram one step further, and instead of building a single histogram, builds a spatial array of orientation-magnitude histograms about the interest point.
    - Pixel contributions are linearly interpolated across the nearest spatial bins to avoid discontinuities
    - Magnitudes are weighted by a Gaussian centred on the interest point.
    - Typical construction is a 4x4 spatial grid, with 8 orientation bins; this leads to a 128 dimensional feature vector.
        - Leads to a very descriptive and discriminative feature.

Matching SIFT features

- SIFT features are often compared using the Euclidean distance.
    - A simple matching strategy between features from a pair of images is to take each feature in turn from the first image and find the closest (in terms in smallest Euclidean distance) feature in the second.
        - A simple threshold can be used to stop very poor matches.
        - Unfortunately, this simple technique tends to result in lots of incorrect matches, and also isn't very robust to big viewpoint changes.
    - A better solution is to take each feature from the first image, and find the two closest features in the second image, and only form a match if the ratio of distances between the closest and second closest matches is less than a threshold (typically set at 0.8, meaning that the distance to the closest feature must be at most 80% of the second closest).

Further reading

- Lowe's seminal SIFT paper has a detailed description of the computation of the SIFT feature, as well as some details on efficient matching: David G. Lowe, **"Distinctive image features from scale-invariant keypoints,"** *International Journal of Computer Vision,* 60, 2 (2004), pp. 91-110. http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

Practical exercises

- If you haven't already tried it, Chapter 5 of the OpenIMAJ tutorial covers finding DoG blobs, extracting SIFT descriptors, and matching them.
- Experiment with using simple gradient histogram features rather than the more complex SIFT features. What do you notice?
    - Hint: you can extract simple orientation-magnitudes by just setting the number of spatial bins in the horizontal and vertical directions to 1. This can be done through the `DoGSIFTEngineOptions` object obtained through the `getOptions()` method of the `DoGSIFTEngine`.

Written by Jonathon Hare. Send any comments/bugs/typos to jsh2@ecs.soton.ac.uk.