COMP3005/Computer Vision

4

Types of image feature and segmentation

Summary

Features that can be extracted from images fall into four main categories: global, grid-based, region-based and local. Global features are extracted from the entire image. Region-based features are extracted from regions of the image called connected components. Detecting connected components requires that you first apply a process called segmentation to partition the image into multiple segments.

Key points

Image feature morphology

- Recap: a feature extractor computes a featurevector describing a *feature* (of an image since we're talking about computer vision).
- Feature morphology refers to the form or shape of a feature; there are many forms of image feature, however they are often grouped into a number of categories:
    - **Global features** are computed from the entire image; there is one feature per image.
    - **Grid or block-based features** are computed for regular (usually rectangular) windows placed across an image; there are as many features as there are blocks. The blocks could be a fixed size (so the number of blocks would change with image size), or there could be a fixed number of blocks, which scale with the image size.
    - **Region-based features** are computed from (typically irregular) regions or **segments** of the image; there might be multiple features per image corresponding to each region.
    - **Local features** are computed at *"interest points"* within an image; there will be as many features as there are interest points.

Global features

- One of the simplest forms of global feature is a histogram, created by accumulating binned pixel values.
    - Pixel values are often considered to be tuples in a specific colour-space.
    - In computer vision, we usually consider **joint colour histograms** – that is we consider all the pixels colour components together.
        - In essence the histograms are multi-dimensional – for example in the case of RGB colour-space, a histogram cell or bin represents a 3D cube of colour-space.
        - The typical feature-space representation of a histogram just unwraps the bins into a 1D vector.
        - Normalisation of the histogram by the number of pixels in the image is important if we want to compare images of different sizes (unless we're using cosine similarity!).
            - Normalisation also makes the histogram represent an empirical probability distribution over the colours in the image.
    - Choice of colour-space is often important when building a histogram feature if you want to achieve certain types of invariance.

Segmentation

- Region-based descriptors (features) are the subject of the next lecture, but to describe a region you first need to detect it!
- Segmentation is the process of partitioning an image into multiple (usually disjoint) segments.

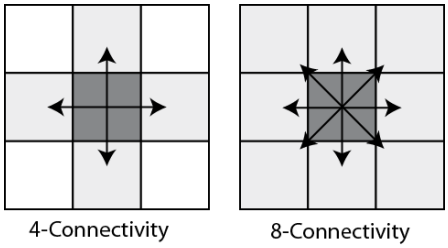
Written by Jonathon Hare, February 11, 2014. Send any comments/bugs/typos to jsh2@ecs.soton.ac.uk.

- A **segment** is a set of pixels that share certain visual characteristics.

- Thresholding is the simplest form of segmentation; it can turn a grey-level image into a binary image.
  - It will really only work well when the problem is constrained:
    - Industrial vision
    - Imaging scenarios where the object has been designed to stand out against the background… QR-Codes are a good example.
  - Basic thresholding requires that you set the threshold that segments the image manually.
  - Otsu's thresholding method automatically computes the threshold on the assumption that that the image to be thresholded contains two classes of pixels (e.g. foreground and background) and thus has a bi-modal histogram. It then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.
  - Local or adaptive thresholding methods compute a different threshold for every pixel, based on the values of the neighbouring pixels.
    - E.g. the *mean adaptive threshold* sets a pixel to the background if its value is less than the mean of its neighbours plus an offset, and sets it to the foreground otherwise.
- The K-Means clustering algorithm is another popular approach for segmentation.
  - In the simplest case you just create a vector from each pixel based on its colour components (in a suitable colour space).
    - This will produce *k* segments based on the colours in the image.
  - Often you want your segments to not only be based on colour, but also on the relative positions of the pixels – for example you might want a lone pixel of one colour to belong to the same segment as the surrounding pixels which all have the same, even if they had a bit of colour variation.
    - One simple approach to doing this is to encode the position of each pixel in the featurevector; for example the RGB value and position could be encoded as follows: [R, G, B, x, y]
      - Because the x and y values are in different units to the pixel values, one has to be careful, otherwise the effect of the location could massively overpower the colour information. If the RGB values are in the range 0…1, then the x and y values could be normalised to the same range by dividing by the height and width of the image respectively. Each element of the vector could then be weighted separately to achieve the desired result.
- Many more advanced segmentation algorithms exist. In the lecture we saw a demo of a graph-based segmentation approach that has become very popular in the computer vision research field. Other techniques include methods pose the segmentation problem as one of pixel classification, and incorporate supervised machine learning techniques to train a classifier to perform segmentation.

Connected components

- A **connected component** is a type of segment in which all the pixels are reachable to each other through a path of spatially adjacent pixels (pixels that are next to each other in the image) that in the set.
  - On a regular pixel grid there are two common ways of considering whether a pair of pixels are spatially adjacent:
    - In **4-connectivity** two pixels are spatially adjacent if one is immediately above, below, left or right of the other pixel.
    - In **8-connectivity** two pixels are spatially adjacent if one is immediately above, above-left, above-right, below, below-left, below-right, left or right of the other pixel.



- The process of **connected-component labelling** takes a binary image and produces a set of connected components.
    - Connected component labelling is useful for breaking an image that had been segmented through thresholding into multiple discrete segments.
    - A number of different algorithms proposed for labelling efficiently. Once such approach is the following two-pass algorithm:

```
1. On the first pass:
        a. Iterate through each element of the data by column, then by row
           (Raster Scanning)
              i. If the element is not the background
                    1. Get the neighbouring elements of the current
                       element
                    2. If there are no neighbours, uniquely label the
                       current element and continue
                    3. Otherwise, find the neighbour with the smallest
                       label and assign it to the current element
                    4. Store the equivalence between neighbouring labels
2. On the second pass:
        a. Iterate through each element of the data by column, then by row
              i. If the element is not the background
                    1. Relabel the element with the lowest equivalent
                       label
```

Further reading

- Colour histograms: http://en.wikipedia.org/wiki/Color_histogram
- Details of Otsu's Thresholding method:
    - http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4310076
    - http://en.wikipedia.org/wiki/Otsu%27s_method
- Adaptive thresholding: http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm
- More segmentation techniques and further details:
    - Mark's book p. 429-431
    - http://en.wikipedia.org/wiki/Image_segmentation
    - The paper describing Felzenszwalb and Huttenlocher's graph-based segmentation technique shown in the lecture: http://cs.brown.edu/~pff/papers/seg-ijcv.pdf
- Connected-component labeling: http://en.wikipedia.org/wiki/Connected-component_labeling

Practical exercises

- Chapter 4 of the OpenIMAJ tutorial covers global colour histogram features
- The `org.openimaj.image.processing.threshold` package contains a number of thresholding implementations for you to play with.
- If you haven't already looked, chapter 3 of the OpenIMAJ tutorial includes information on segmentation and connected components.