Computer Vision Assignment :
Finding bookshelves and recognising books

Romain Perrone

1 Question 1

(a) Given images of bookshelves (such as those shown below), develop a system to locate the shelves on which the books are standing. Your solution must consist of a series of computer vision techniques and the technical details of the techniques used must be provided.

Based on these images, we can make the following assumptions:

1. The shelves always appear at an angle between -30 degree and +30 degree
2. The books are vertically aligned and cannot be laid horizontally.
3. The image is encoded in RGB.

We will use the following techniques to locate the shelves on which the books are standing :

(a) Transform the RGB image into grayscale

(b) Use Otsu's method for thresholding

(c) Remove noise using morphological transformation (opening) to remove small holes

(d) Apply Connected component analysis to this binary image

(e) Use Hough Line to find lines associated with those contours

(f) Filter the lines with regard to their slope value

The technical details of these techniques are provided below:

1.1 Transform the RGB image into grayscale

We will convert RGB values to grayscale by forming a weighted sum of the R, G, B components: $grayscale = 0.298 * R + 0.587 * G + 0.114 * B$

and then compute this grayscale value for every pixel in the image.

1.2 Use Otsu's method for thresholding

Otsu's method tries to minimise intra-class intensity variance, or equivalently to maximise inter-class variance to find an optimal threshold. Otsu's method is useful as it allows us to apply a different threshold to different images, depending on the lighting settings. It is more robust than a predefined threshold value and performs well as long as the image histogram has a bimodal repartition.

Once the threshold value is found: (a) pixels whose intensity $I < T$ are set to 0 (b) the rest of the pixels are set to 1. Thresholding can be fined-tuned by adjusting the number of bins (quantisation) that are used in the histogram.

1.3 Remove noise using morphological transformation (opening) to remove small holes

The binary image we obtained in the previous step might contain some noise. We can remove it using morphological transformation (opening) which is a combination of an erosion (-) followed by a dilation (+)

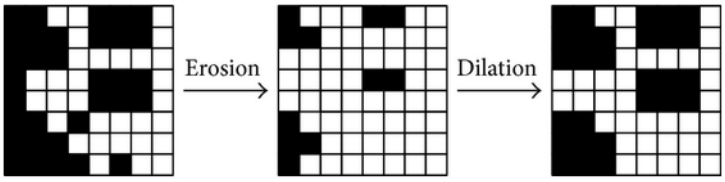

Figure 1: Example of a dilation followed by an erosion

These morphological transformation are defined as follow:
* Erosion: for each black pixel (i,j), turn the following pixels to black
(i-1, j-1), (i-1, j), (i-1,j+1), (i, j-1), (i,j+1), (i+1,j-1), (i+1,j), (i+1, j+1).

* Dilation: for each white pixel (i,j), turn the following pixels to white
(i-1, j-1), (i-1, j), (i-1,j+1), (i, j-1), (i,j+1), (i+1,j-1), (i+1,j), (i+1, j+1).

1.4 Apply Connected component analysis to this binary image

We now need to determine the contours of this binary image. We will use the Connected component algorithm to do that. The binary image is parsed row by row, and newly found pixels are labeled. If a neighbour pixel already has a label, the newly labeled pixel will have the same value. We keep repeating this process until the last row is reached, and finally re-evaluate the image merging connected component together.

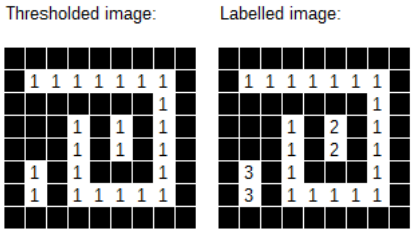

Figure 2: Connected components

1.5 Use Hough Line to find lines associated with those contours

Finally, we can use Hough Line to find the shelves. In Hough space a point (x, y) is identified with the following parameters $(r, theta)$ (polar coordinates).

We can establish the following relationship $r = x * cos(theta) + y * sin(theta)$, so a line in the cartesian coordinate system $y = a * x + b$ will be mapped as a sine wave in Hough space.

For each pixel, the algorithm determines if there is enough evidence of a straight line at that pixel. If so it computes the parameters $(r, theta)$ for that line and adds it to an accumulator. Finally it selects the bins with the highest values, by looking at the local maxima. These are the lines that are the most represented in our image.

1.6 Filter the lines with regard to their slope value

Going back to the cartesian equation, we compare the slope of each line to our $[-30; 30]$ degree interval. Lines whose slope does not fall within that interval are dismissed. Finally, we compute

the standard deviation of the slope of the remaining lines to evaluate our selection, and dismiss outliers from the model using standard linear regression.

2 Question 2

(b) Describe Template Matching and the Scale Invariant Feature Transform (SIFT) and discuss the issues of applying both of these techniques to the problem of recognising individual books, assuming that you have already located the bookshelves. Include a discussion of which of these techniques is more appropriate. You must provide technical details of the both of the techniques.

2.1 Template Matching

Template matching is a popular technique for image recognition. It compares two images using a similarity metric. Because images are not continuous functions, we use cross correlation instead of convolution. The similarity metric is defined as follows : $\sum_{m,n} f(i+m, j+n).t(m,n)$, where $f(i,j)$ corresponds to the pixel (i,j) of the input image and $t(m,n)$ to the pixel (m,n) of the template image.

For every possible position in the image, we evaluate this matching criterion. We can transform the RGB image into grayscale, or alternatively sum up the similarity values for each channel.

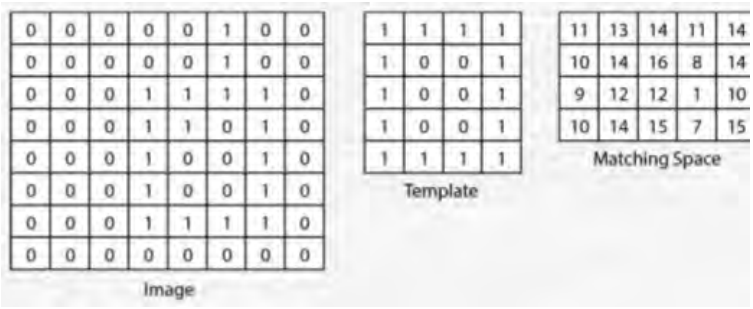

Figure 3: Template matching: similarity matrix obtained using a 4x4 template matrix

Finally we look for local maximas. This is where the similarity between the template and the input image is maximal, and it gives us the position of the object within the original image. We can also derive a probability from the cross-correlation function by dividing the result by the size of the template image. This gives us a probability of present in the [0;1] interval.

A sound approach for template matching would be to use a pyramid of scale (the size of the

image is reduced and a gaussian filter is applied at each iteration). Going through low size/resolution image first would significantly reduce the computational expense that comes from an extensive search. This would allow us to discard whole sections of the image early on to focus on more probable areas for research.

Potential issues associated with this techniques are, but not limited to:

(a) Orientation issues: a book slightly inclined might not be recognised.
(b) Lighting issues: if the input and template image were taken under different lighting settings, template matching will fail.
(c) Template matching requires very close matches: noise and sampling differences might adversely affect book recognition.

2.2 Scale Invariant Feature Transform (SIFT)

SIFT (Scale Invariant Feature Transform) was devised by David Lowe in 1999. To this day, it is one of the most effective and robust recognition methods. SIFT identifies specific features within an image. These features are scale invariant and do not depend on the brightness of the scene or on the rotation of the image.

SIFT find these feature points by computing a difference of Gaussian. This is the equivalent of a pass-band filter, where high and low frequencies are removed. This preserves spatial information within that range, but suppresses details (that are associated to higher frequencies)

The difference of gaussian is applied to a serie of smoothed and resampled images. Extrema are labeled as keypoints. Low contrast points and edges are discarded.

We then compute gradient magnitude and orientation for each pixels around these keypoints and determine the most frequent orientation. After rotating these vectors by keypoint orientation, we divide them into subregions and create 8-bins histograms. The vectors are normalised and distributed into bins, with weight applied by gradient, location and orientation. We finally obtain a feature vector for each keypoint.

To perform SIFT recognition we first compute the features of the input image as described above. The features are then matched to keypoint features from the training data using euclidian distance. Images that have similar features will be close in the feature graph. With some additional processing (Hough transform and Bayesian probability), we derive a probability of presence for our object within the input image.

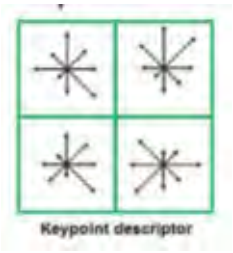

Figure 4: SIFT Keypoint descriptor

2.3 Discussion

SIFT is an excellent candidate for book recognition. The orientation of the books may change based on where the photo is taken or if someone uses the bookshelves. Even though template matching is a strong technique, SIFT offers many benefits (scale invariance, independent to changes in rotation/brightness) despite a higher computational cost. Note the variability in the lighting setting for the second bookshelf. We might expect SIFT to give substantially better results than template matching in this case.