

## CS4243 Recess Review Questions

### Question 1 Point Processing

Matching of equations to functions:

1.  $x = p + 128$  corresponds to c).  
Since the image is simply brighter and the eyes of the child turn gray instead of black
2.  $x = p - 128$  corresponds to a).  
Since the head of the child is completely black after the transformation (was previously  $< 128$ ).
3.  $x = p/2$  corresponds to b).  
Since no values are absolute black (0) just but overall darker.
4.  $x = 2p$  corresponds to d).  
By process of elimination.

### Question 2 Linear Filtering

- (a) The  $7 \times 7$  box kernel was applied to image b), as it shows rectangular artifacts on the blurred regions, where the original image has a high contrast.

Conversely the  $7 \times 7$  gaussian kernel as applied to image a), as it does not show these artifacts and is simply a smoother version of the original image.

- (b) A separable filter kernel  $k$  is one which can be split into two smaller kernels  $f, g$  as follows:

$$k = f * g \quad (1)$$

A box filter of size 7 can be written as:

$$k = \frac{1}{49} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} = \frac{1}{7} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} * \frac{1}{7} [1 \quad \dots \quad 1] = f * g \quad (2)$$

## Question 3 Corner Detection

(a) Suitability of the functions for finding corners:

(a)  $R = \min(\lambda_1, \lambda_2)$

If the eigenvalues  $\lambda_1, \lambda_2$  are both large, their minimum will be as well. This corresponds to large stretching in both dimensions of  $H$ , corresponding to a large change in the image patch in both directions. In all other cases  $R$  is small, corresponding to a small change in the image patch in at least one dimension.

(b)  $R = \det(H) - \kappa \cdot (\text{tr}(H))^2 = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$

If only one eigenvalue is large. Assume w.l.o.g.  $\lambda_1 \gg \lambda_2 > 0$ :

$$R = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 \leq \lambda_1 \lambda_2 - \kappa(\lambda_1^2 + 2\lambda_1 \lambda_2) \approx -\kappa \lambda_1^2 < 0 \quad (3)$$

However if  $\lambda_1 = \lambda_2 \gg 0$ :

$$R = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)^2 = \lambda_1^2 - 4\kappa \lambda_1^2 \underset{\kappa < 1/4}{\approx} \lambda_1^2 > 0 \quad (4)$$

(c)  $R = \frac{\det(H)}{\text{tr}(H) + \epsilon} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2 + \epsilon}$

If only one eigenvalue is large. Assume w.l.o.g.  $\lambda_1 \gg \lambda_2 > 0$ :

$$R = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2 + \epsilon} < \frac{\lambda_1 \lambda_2}{2\lambda_1} = \lambda_2 \quad (5)$$

so  $R \in O(\lambda_2)$  and therefore small.

However if  $\lambda_1 = \lambda_2 \gg 0$ :

$$R = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2 + \epsilon} = \frac{\lambda_1^2}{2\lambda_1 + \epsilon} \approx \lambda_1/2 \quad (6)$$

So  $R$  is large as well.

(b) the second or third function is preferable over the first one, as it represents the same notion of corneriness; but does not require the expensive computation of the eigenvalues  $\lambda_1, \lambda_2$  of  $H$ .

(c) If  $\lambda_2 > k \cdot \lambda_1 \geq 0$  with  $0 \leq k \leq 1$

When gradually increasing  $k$  the constraint  $\lambda_2 > k \cdot \lambda_1$  gradually becomes more constricting until  $\lambda_2 > \lambda_1$  for  $k = 1$ . This forces both eigenvalues to be of similar magnitude, giving a large corneriness score by the metrics above. Therefore the number of corners found in an image if the constraint is fulfilled by every  $H$  matrix an image patch, grows with  $k$ .

## Question 4 System Design

- (a) To create a fabric fault detector, one needs to create a feature description which can detect large changes in texture. For this a texton representation is suitable. Here each image is processed by filters chosen from a filter bank. If the response is large everywhere but small in just one spot across multiple filters, this can be considered as a fault in the fabric.

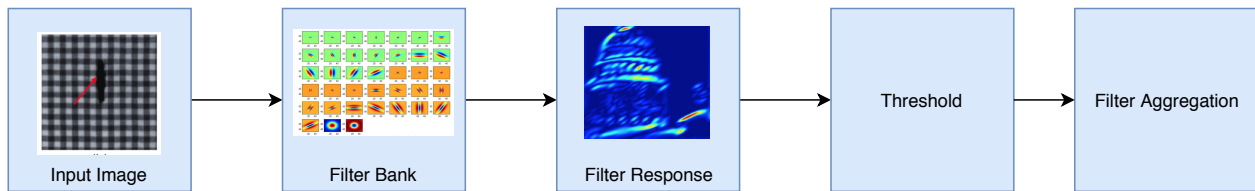


Figure 1: Fabric fault detector pipeline

- (a) **Input image:** Image is cropped to a fixed size.
- (b) **Filter Bank:** Each filter of the filter bank is convolved with the image.
- (c) **Filter Response:** For each of the filters, a response map is created, which is a normalized value, of how much the image in the given window matches the filter kernel.
- (d) **Threshold:** A threshold is applied on every filter response pixel value  $p$  in the form of  $p' = \min(p, \tau_i)$ , where  $\tau_i$  is the maximal response allowed for each filter kernel  $\kappa_i$ .
- (e) **Filter Aggregation:** The responses for each of the filters is collected after applying the threshold and if a certain set of filter responses show a low enough response in local image patches, the image is marked as a faulty fabric.

When training this algorithm, the goal is to identify the set of filter kernels in the aggregation step, which when giving a low response correspond to a faulty fabric.

- (b) For detecting each of the faults in the given images, the Threshold and aggregation sections of the classifier needs to be fitted to make sure they are classified as faulty. Therefore these images are used as training examples.