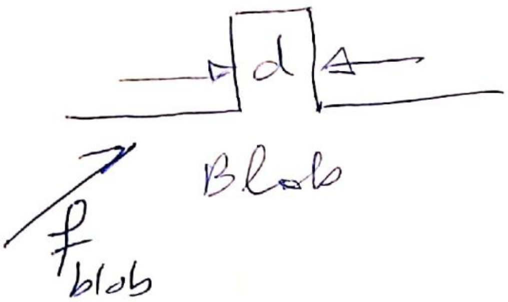
Question 3:

(a) Explain using equations: Why and How the Laplacian of Gaussian is used to detect blobs in images and how the Gaussian width "σ" determines the scale of the detected blob **(2)**

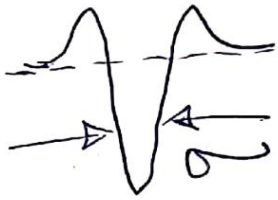
(b) Explain using a figure how many blob-scales can be detected in the standard SIFT algorithm? **(1)**

(a)



g'' Laplacian

σ scale of (g'')



$f_{(d)} \circledast g''_{(\sigma)} \Rightarrow$ Max. (extreme) Value

2

when scales ($d \approx \sigma$) $\Rightarrow$

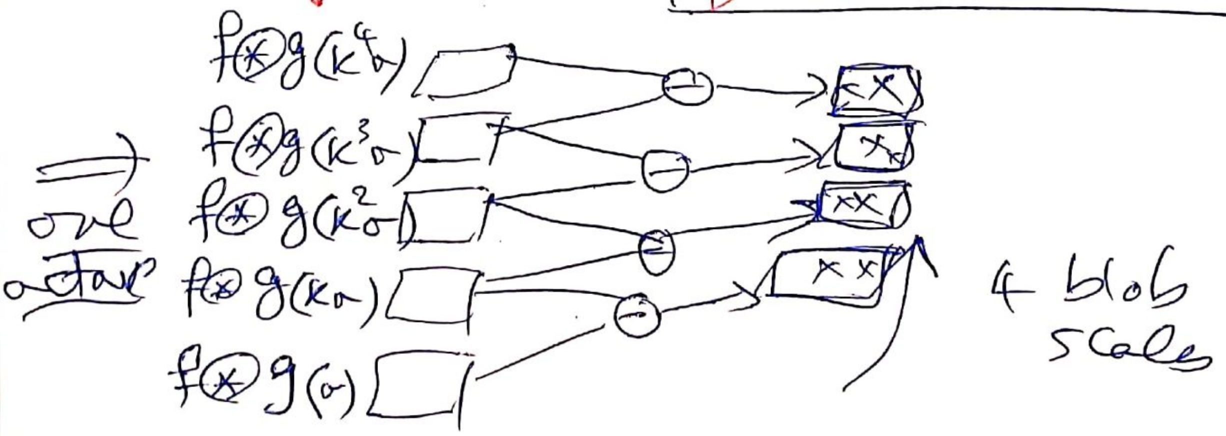
So conv. between g'' (Laplacian) $\circledast$ f (blob) $\longrightarrow$ extreme value ~~of~~ → the location of blob

(b) 4 octaves, each octave detects 4 sift (blob) scales

then, resize image to half rows, columns and repeat → octave 2, 3, 4

so, we get 16 scales

1



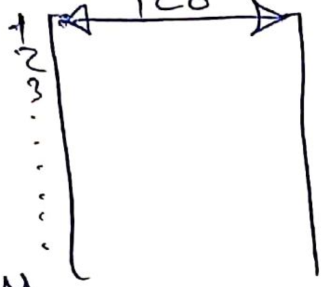
CamScanner

(c) Using 10,000 images, explain the *steps required to generate a SIFT codebook (dictionary)* with **256** centers. Assume each image will produce 200 SIFT descriptors (Blobs), show the size of the array used before the K-means and the size of the dictionary (number of rows and number of columns) **(2)**

(d) Explain how to extract a **256**-dimension (bag-of-words) feature vector for a new image using the dictionary in (c). Then show the extracted vector assuming the new image has 200 SIFT-interest points, where 100 of them match dictionary center #5 , 50 match center #11 and 50 match center #255. **(2)**

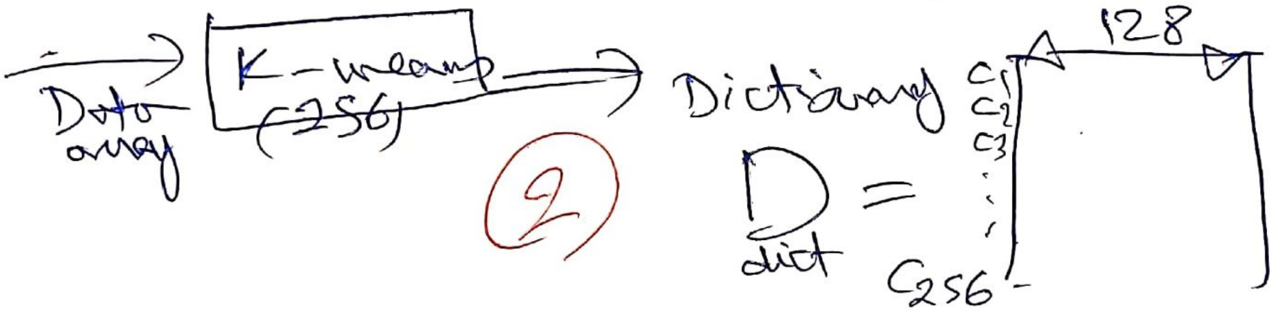
(c) (1) apply (SIFT) on each image (10,000 * 200) ≃ 2M (SIFT) descriptors each <128>

D_{data} (array) =



N=2M

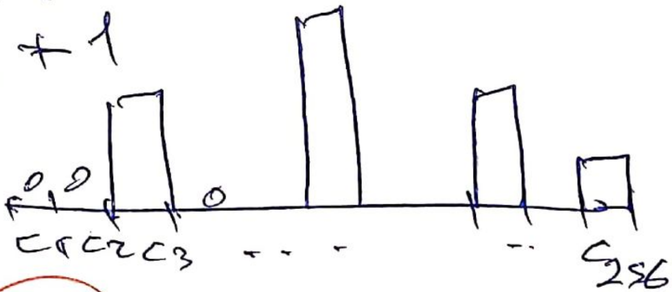
(2) Apply K-means with K=256 clusters



2

(d) (i) New image F → has ≃ 200 interest points (blobs)
for each one, search in dict for nearest center & increment its histogram count by +1

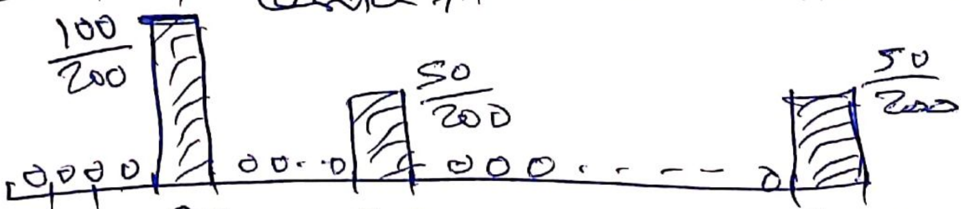
then, Normalize by dividing all counts by 200.



2

(ii) (100) Center #5, (50) center 11 and (50) center 255

histogram feature vector



Question 4:

(a) A quantized Image has the following 5 pixel values shown in the table below with their number of occurrences in the image:

Pixel Value	Number of occurrences
0	5
1	10
2	170
3	10
4	5

(i) Design a Huffman code using the given image data showing the binary codes for all the 5 possible pixel values **(2)**

(ii) Find the total number of bits required to store this image using the designed Huffman code and the total number of bits required if we are using a fixed-length encoder. **(1)**

(b) For the image given by the table above:

(i) Plot the **Normalized Histogram** as well as the **cumulative distribution** of the image **(1.5)**

(ii) Apply the **Histogram Equalization Algorithm** on this image and find the new pixel values and Plot the histogram for the equalized image **(2.5)**

(a) Huffman:

(i)

(1)
S_2 0.85
S_1 0.05
S_3 0.05
S_0 0.025
S_4 0.025

(2)
S_2 0.85
S_1 0.05
S_3 0.05
S_{04} 0.05

(3)
S_2 0.85
$S_3(S_{04})$: 0.1
S_1 0.05

(4)
S_2: 0.85
$S_1(S_3(S_{04}))$: 0.15

Code assign:

$S_2 \leftarrow 0$

$S_1(S_3(S_{04})) \Rightarrow 1$

S_1: 10 — 2 bits

$S_3(S_{04}) \Rightarrow 11$

S_3: 110 — 3 bits

S_{04}: 111

S_0: 1110 — 4 bits

S_4: 1111 — 4 bits

2

(ii) Σ bits with Huffman = (170*1) + (5*4) + (5*4) + (10*3) + (10*2) = 260 bits

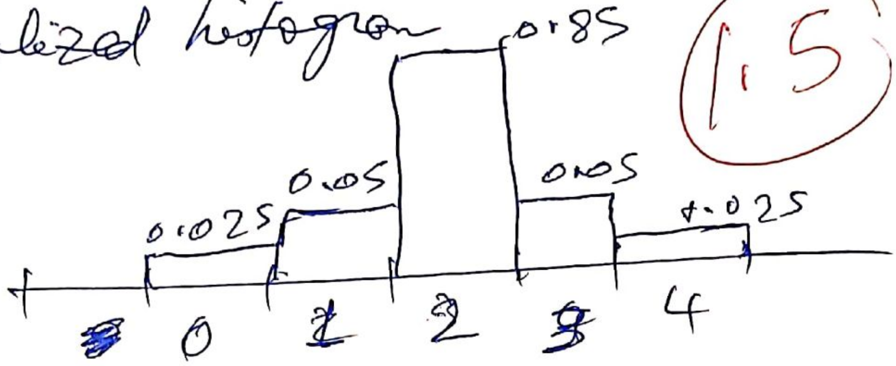
vs fixed length coding → 5 symbols → 3 bits

3*200 = 600 bits

1

(b) Histogram equalization

(i) Normalized histogram



1.5

(ii) new pixel values

$(L * \sum p)$

$L \Rightarrow 4$

$S_0 = 4 * 0.025 = 0.1 \rightarrow 0$

$S_1 = 4 * (0.025 + 0.05) = 0.3 \rightarrow 0$

$S_2 = 4 * (0.025 + 0.05 + 0.85)$
$= 3.7 \simeq 4$

$S_3 \rightarrow 3.9 \simeq 4$

$S_4 \rightarrow 4 \Rightarrow 4$

1.5

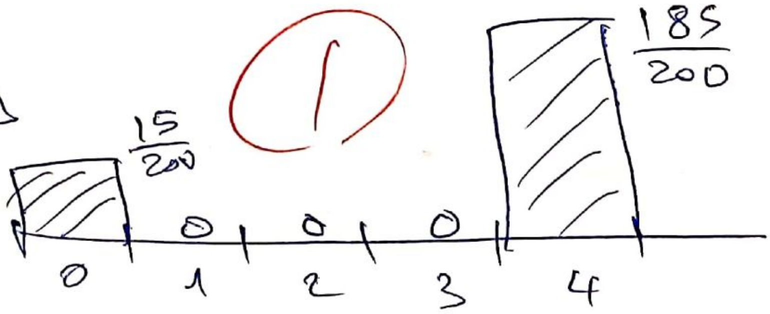
so, new count of pixels:

$r_0 \Rightarrow 5 + 10 \Rightarrow 15$ pixels

$r_1, r_2, r_3 \Rightarrow$ zero pixels

$r_4 \Rightarrow 5 + 10 + 170 = 185$ pixels

New histogram (Normalized)



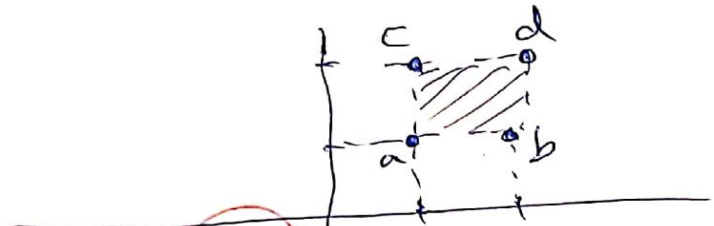
1

(c) An image has a Box with pixel Coordinates: $\{x_i, y_i\} = \{(10,10), (20,10), (10,20), (20,20)\}$. The Homography transform below is applied to the image:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

i. Draw the Box before and after the transform. Explain what happened to the shape of the Box? **(2)**

ii. Is this is a translation, Euclidean, Affine or projective transform? **(1)**



$a = (10, 10, 1)$
$b = (20, 10, 1)$
$c = (10, 20, 1)$
$d = (20, 20, 1)$

1

$$a' = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 31 \\ 22 \\ 1 \end{bmatrix}$$

$$b' = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 20 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 51 \\ 32 \\ 1 \end{bmatrix}$$

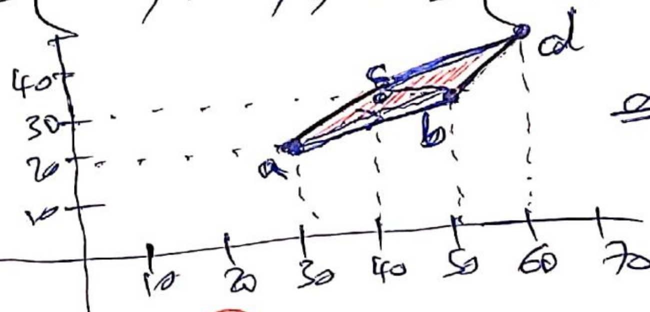
$$c' = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \\ 1 \end{bmatrix} = \begin{bmatrix} 41 \\ 32 \\ 1 \end{bmatrix}, \quad d' = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 20 \\ 20 \\ 1 \end{bmatrix} = \begin{bmatrix} 61 \\ 42 \\ 1 \end{bmatrix}$$

1

new box coordinate

$a' = (31, 22)$, $b' = (51, 32)$, $c' = (41, 32)$

$d' = (61, 42)$



distorted

This is an Affine transform → 6 coeffs are allowed

1

CamScanner