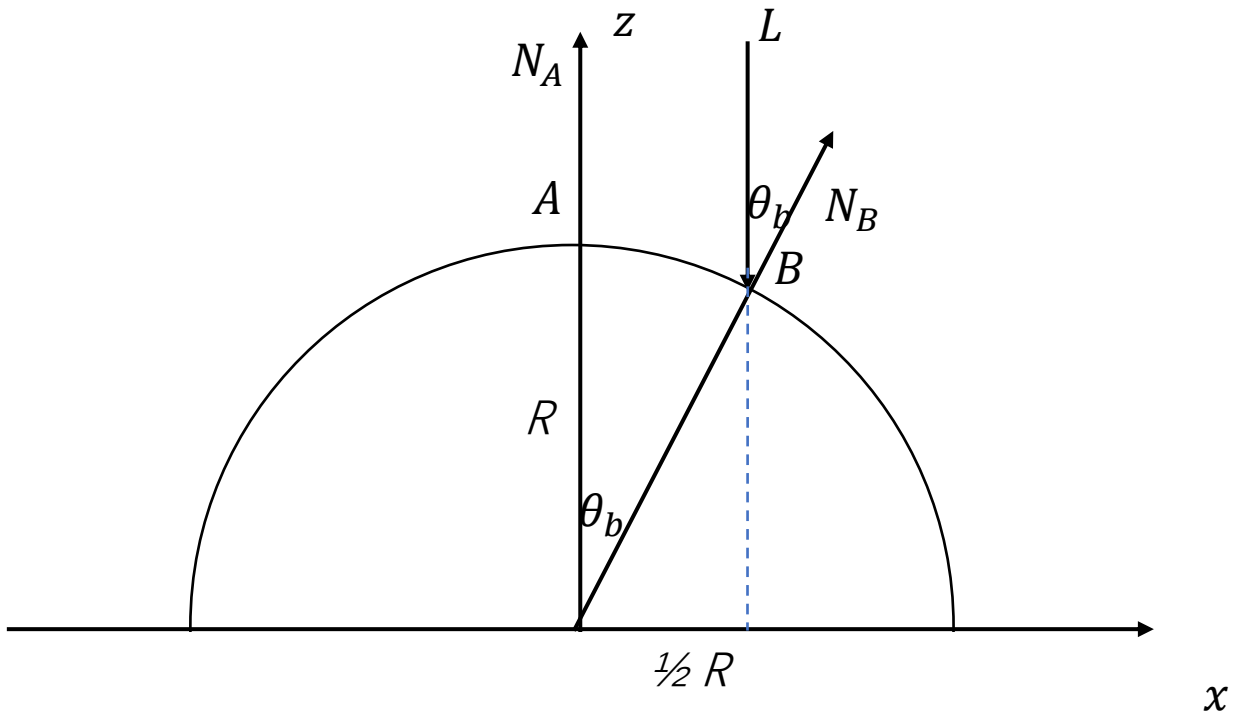


1. a

$\theta_a = 0$, N_A lies in the same direction as L

$\theta_b = 30^o$, See the illustration of the x-z plane, $\sin\theta_b = \frac{\frac{1}{2}R}{R} = \frac{1}{2}$, therefore $\theta_b = 30^o$

$\theta_c = 60^o$, similarly as above, in the y-z plane.

- 1.b, no, we do not know the material of the sphere

- 1.c, since it is Lambertian, $I = |L|\rho \cos\theta$

  we know

$$I_A = |L|\rho_a \cos\theta_A = |L|\rho_a = 100$$

  where ρ_a is the albedo, we do not know the albedo for B and C and therefore cannot determine I_b and I_c

- 1.d, yes, the sphere works like a perfect mirror, the camera is pointing downwards from the z direction, therefore we can see light reflected from A, but light reflected from B, C do not go to the camera, and therefore $I_b = I_c = 0$.

1.e, Yes, now we have constant albedo.

we know

$$I_A = |L|\rho \cos\theta_A = |L|\rho = 100$$

therefore

$$I_b = |L|\rho \cos\theta_b = 100\cos 30^{\circ} = 50\sqrt{2}$$
$$I_c = |L|\rho \cos\theta_c = 100\cos 60^{\circ} = 50$$

- 1.f
  - 4 bit means 0-15 and integer only
  - Compress to 4bit with round up. For example, using flooring ⌊ ⌋ for round up. (Other way of round up is also correct, if specified.)
    - $I'_A = \left\lfloor \frac{I_A}{16} \right\rfloor = 6$
    - $I_B = 5$
    - $I_c = 3$
  - Decompress to 0-255, the easiest way is to multiply by 16, if you do linear stretch, from [0-15] to [0-255] that is also correct
    - $I''_A = 16 * I'_A = 96$
    - I''_B=80
    - I''_C=48

- 1.g Human cannot see it, if the material is not florescence, all reflected color is still 1100nm, which are not visible by human eyes.
- 1.h Now the reflected light is 400nm, which is visible by human, the color is blue to violet.

- 1.i

$$\begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 1.j

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

- 1.k, we can, we have three points that are linear independent

- 2.a

$$f_{xx} = h * h * f = (h * h) * f$$

$$(h * h) = [-1,1] * [-1,1] = [-1, 2, -1]$$

depending on where you specify the center of the kernel, the number may vary.

- 2.b,

0.25

2. C

h: high pass, g: low pass, g*h: band pass

- 2.d do it yourself :P, we have same question in exercise.
- 2.e it is a 2D Laplacian of Gaussian (or difference of Gaussian) filter, which detects blobs. Corners is also correct.
- 2.f:

  $U = [-1, 2, -1] * [-1, 2, -1]^T$

2. g do it yourself :P, we have same question in exercise.

2.h if goes eigenvalues are significantly larger than zero, it indicates corner.

2. i, do it yourself :P, we have same question in exercise.

- 2.j
  - Scaling, rotation, translation, global illumination changes and etc.
- 2. k
  - No, P is an edge, which is not deterministic for optical flow.
- 2. l
  - The filter center is not specified, and therefore any specified center is correct. If the filter is centered at upper left corner, then

0.5	1	0.5	0
1.5	4.25	4	1.5
2.25	4.75	5	3
2	2	2.5	3

0.5	1	0.5	0
1.5	2	2	1.5
2	2.5	3	3
2	2	2.5	3

  - Averaging median
  - The median is the average of the 2nd and 3rd smallest value, if you specified the median as the 2nd or the 3rd value for simplicity, it is also acceptable.

- 3. a we uniformly sample image patches/window throughout the image. The sampling is in x and y direction with a unit step.
- 3. b for simplicity if not considering handling boundaries, then simply

  200*200 for step=5 pixels

  100*100 for step= 10

  50*50 for step = 20

  if considering the boundaries:

  181 * 181

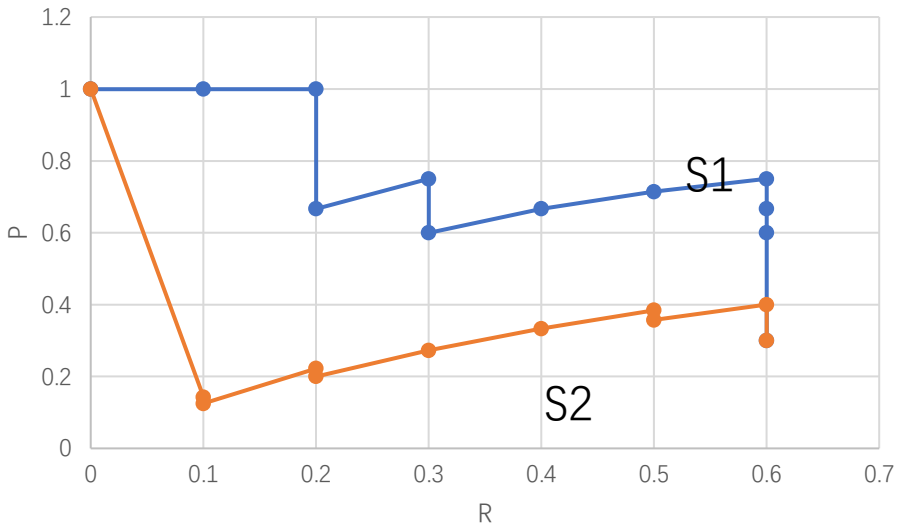
  91*91

  46*46

- 3. c
  - pros:
    - less windows, more efficient,
    - No repeated detection of the same object
    - Any other reasonable answers
  - Cons
    - May likely to mis-detect
    - Any other reasonable answers
- 3.d
  - Selective search
  - Key point detection then sliding windows
  - Salient region detection then sliding windows
  - Any other reasonable answers

- 3.e
  - S1: precision 5/20 = 0.25, Recall 5/10 = 0.5
  - S2: precision 5/20 = 0.25, Recall 5/10 = 0.5

- 3. f



Clarification:
The required concept is Precision@K, if you plot the average precision curve, it is also correct

- 3.g S1 is better
- 3.h convolutional layer only has parameters/weights for the convolutional kernels, while fully connected layer has weights for all the connections.

  Any other similar or reasonable answers

3.i 7*7*100*25 if not considering the shift. Considering the shift is also correct

3.J 7*7*100*100 if not considering the shift. Considering the shift is also correct

- 3.k

12	12
12	12

3*3 max pooling, stride = 1

11	1
3	12

2*2 max pooling, stride = 2