1. **SIFT 1**

   When computing a SIFT keypoint, one has to compute an orientation histogram. Consider a scale space point (x, y, σ) such that the orientation histogram is non-zero in only one bin. Could this point be a keypoint?

   (a) Yes, because its orientation histogram has a well defined maximum.

   (b) No, because such a point would not be a local extremum. **(answer – although see below)**

   (c) Yes, if the Difference of Gaussians function is sufficiently different from 0 at that point. (Use a threshold.)

   Answer: The answer is (b) for the following reason: If the gradient orientations in a neighborhood all fall within one bin, then these vectors are essentially parallel since the bins are only 10 degrees wide. When the DOG gradient vectors are parallel at a point, the DOG function cannot have a local extremum at that point.

   The answer (c) was chosen by 45% of you, but it is incorrect. The condition that the DOG is greater than some threshold is not the condition defining a keypoint. I decided to give 0.5 for this answer since the correct answer (b) was subtle.

   Updated. Strictly speaking, (b) is not correct because we cannot assume that the vectors are exactly parallel and so it could happen that the point gives an extremum. Indeed this the reason why Lowe suggests using a condition based on the Hessian – see Assignment 2. **So, since none of the three choices is strictly speaking correct, I have now decided to give the point to everyone.**

2. **SIFT 2**

   Given an $N \times N$ image, suppose we compute SIFT feature points locations (x, y, σ) using a DOG pyramid using $m = 4$ slices per octave. What is the time complexity to construct the pyramid and compute the feature point locations?

   (a) $O(N^2 \log_2 N)$

   (b) $O(N^2)$ **(answer – although see below)**

   (c) $O(N^4)$

   (d) $O(N^4 \log_2 N)$

   Answer: The correct answer is (b). While there are $O(\log_2 N)$ layers, one does not have N^2 pixels at each layer. Rather the number of pixels decreases by a factor of $2^2 = 4$ for each octave.

   Updated: The correct answer above is based on what one does *in practice*, namely a Gaussian pyramid is computed with a sequence of blur → subsample → blur → subsample, etc, where the blur is always with the same small Gaussian. (See lecture 9 slide 24). The reason that this is done in practice is that it is very efficient, namely the time required is $O(N^2)$ specifically one uses $N^2 + (N/2)^2 + (N/4)^2 + \ldots$ operations – which is less than $2N^2$.

Unfortunately, I did not explicitly say in the question that I meant the minimum time required to build a Gaussian pyramid, or that I meant what is the time required for the typical way it is computed. So this opens up choices in the answer that correspond to methods for computing the Gaussian pyramid that are correct, but are not used in practice because they are inefficient. **For this reason, I have regraded the question and given the point to everyone.**

3. **Histogram tracking**

   In the context of histogram tracking, we can define the weighted histogram in the neighborhood of a region of interest by:

   $$hist(u; \mathbf{y}) = \frac{1}{m} \sum_{\mathbf{x} \in ROI(\mathbf{y})} \delta(u - bin(I(\mathbf{x}))).$$

   If we want $hist(u; \mathbf{y})$ to be a probability function of $\mathbf{u}$, then what must m be in the above expression?

   (a) the number of pixels in the region of interest **(answer)**

   (b) the number of bins in the histogram

   (c) the number of RGB values in each bin

   (d) the number of distinct RGB values in the region of interest

   Answer: The answer is (a). This is the case of a uniform weighting function with weights $\frac{1}{m}$.

4. **Perspective**

   What is the perspective projection of a scene point $(X, Y, Z) = (15, 20, 30)$ onto the $Y = 4$ plane? Assume the projection is toward the origin.

   (a) (3, 4, 6) **(answer)**

   (b) (6, 4, 12)

   (c) (12, 4, 24)

   (d) (2, 3, 4)

5. Consider a 3D scene that consists of a ground plane surface that is tiled with regular hexagons (see below). Also suppose that two 3D cubes are placed at random positions and orientations on this ground plane.

   What is the maximum number of vanishing points in an image of this 3D scene?

   (a) 6

   (b) 9 **(answer)**

   (c) 12

   (d) 8 **(answer – assuming cubes are lying flat on ground)**

   (e) 5

   (f) 3

(g) 2

(h) 1

Answers (2): There are three sets of parallel lines on each cube, and three pairs (a pair is also a set) of parallel lines on the tiles. Therefore there are nine sets of parallel lines in the scene. Each set of parallel lines defines a vanishing point. So this gives an answer of 9.

Update: However, if the two cubes are laying flat on the surface, then the vertical lines of the two cubes are parallel. We shouldn't double count these two vanishing points. So, this would give an answer 8.