Assignment #3 (15%)
Both written and programming parts due: Wednesday, April 4
Total mark: 30

On the cover page of your written submission, write and sign the following statement: “I have read and understood the policy concerning collaboration on homework and lab assignments”. Without such a signed statement, your work will not be marked. A sample cover page can be found on the course webpage. You need to explain your work.

Problem 1 (2 marks) CMY as primary colors?
Can cyan, magenta, and yellow phosphors be used to build a shadow mask CRT monitor? If yes, how would it compare with a CRT made up with red, green, and blue phosphors?

[Solution] Yes, it can be done. But the color gamut will be smaller.

Problem 2 (2 marks) Sampled signals in frequency domain
Provide a detailed explanation why the Fourier transform of a sub-sampled signal, with a sampling frequency of \( f \), is composed of replicas of the Fourier transform of the original signal and that the distance between adjacent replicas is \( f \).

[Solution] A sub-sampled signal can be obtained as a product between the original signal and a Dirac comb (impulse train or sampling) function having a frequency \( f \). By Convolution Theorem, the Fourier transform (FT) of a product is a convolution of FT of the two terms. The FT of a comb is also a comb. Convolving this comb with the FT of the original signal, we obtain the replicas separated by the sampling frequency \( f \).

Problem 3 (2 marks) “Spinning of the wheel”
In movies and on TV, you can often see the wheels of a car appear to be spinning in the wrong direction. What causes this artifact? Can anything be done to fix this problem?

[Solution] This results from under-sampling in time. Note that the camera can only capture the scene at discrete time instants. For example, suppose that the wheel can complete one rotation in 1 second but the camera samples at every 0.9 second, then a red mark on the edge of the wheel would appear to be rotating in the wrong direction. One can obviously fix this problem by increase the sampling frequency.

Problem 4 (2 marks) Texture mapping without distortion
What kinds of surfaces can be texture mapped (assuming that the texture is a planar image) without any distortion? Try to provide as general an answer as possible.
When the surface is developable … a surface is developable if it has constant zero Gaussian curvature, e.g., a plane, cylinder, cone, etc.

**Problem 5 (2 marks) Radiosity**

Explain in what ways the radiosity method covered in class is designed to model global illumination of a scene composed of perfect diffuse reflectors.

**[Solution]** The main point here to make is that the radiosity being computed does not depend on the receiving patch; see course slides on global illuminations.

**Problem 6 (2 marks) Mean vs. median filtering**

In mean filtering of an image, we assign to each pixel \((i, j)\) an average of a set of neighboring pixels of \((i, j)\). In median filtering, we take the median instead of the mean. Please compare the two approaches in terms of quality of results and efficiency.

**[Solution]** Mean filtering blurs the image (e.g., it can destroy features) but it is linear and can be efficiently executed via convolution. Mean filtering does a better job at feature preservation but it is more expensive to carry out.

**Problem 7 (3 marks) Triangular coverage on summed area table**

Describe how to efficiently compute the sum of intensities covered by an arbitrary triangle over a texture map, using a pre-computed summed area table. You can assume that the texture map stores one intensity value per texel and that the vertices of the triangle locate at texel centers. Your calculation should be based purely on values stored in the summed area table and no intersection tests between triangle edges and grid lines should be computed. Hint: Try to cover only cases that are really distinct and use figures to assist the description of your solution. Make your algorithm as efficient as possible.

**[Solution]** Consider the tightest bounding rectangle for the given triangle and there are really just three cases in terms of how many corners of the rectangle coincides with one of the vertices of the triangle.
One easy mistake to make is to confuse the area of a shape with the sum covered by that shape. For example, in the last case, the area of the triangle is a half of the area of the rectangle, but one cannot say the same with texture coverage.

Using the summed area table, one can compute the sum covered by any axis-aligned rectangle in constant time. For a triangle however, I am not aware of a way to do the same in general and in exact terms. One possible solution is to approximate the coverage of a triangle using subtraction of rectangular coverages; these are illustrated. To speed up computations, we always choose one corner of such rectangles as the midpoint of an edge, so that shifts can be performed.