Prof. M. Gross (Computer Graphics Lab)
Prof. M. Pollefeys (Computer Vision Group)

P. Kaufmann, S. Martin, M. Germann,
G.-M. Baschera, F. Fraundorfer, G. H. Lee,
A. Taneja, P. Singh

Final Exam

17 August 2010

First and Last name: ______________________________

ETH number: ______________________________

Signature: ______________________________

General Remarks

- At first, please check that your exam questionnaire is complete (there are 16 pages total).
- Remove all material from your desk which is not allowed by examination regulations.
- Fill in your first and last name and your ETH number and sign the exam. Place your student ID in front of you.
- You have 3 hours for the exam. There are 7 questions, where you can earn a total of 180 points. You don't have to score all points to obtain a very good grade.
- Start each question on a separate sheet. Put your name and ETH number on top of each sheet. Only write on the question sheet where explicitly stated.
- You can answer questions in English or in German. Do not use a pencil or red color pen.
- You may provide at most one valid answer per question. Invalid solutions must be canceled out clearly.

	Topic	Max. Points	Points Achieved	Visum
1	Filters	20		
2	Fourier Transform	25		
3	Optical Flow	20		
4	PCA and Data Compression	25		
5	Colors and Shading	30		
6	Transformations and Projections	30		
7	Anti-Aliasing and Texture Filtering	30		
Total		180		

Grade: ....

Question 1: Filters (20 pts.)

$$h_1 = \begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix}$$

Figure 1: 5×5 filter kernel h_1.

a) Figure 1 shows a 5×5 image filter kernel h_1. Name this filter kernel and describe the effect of convolving this filter kernel with an image. **2 pts.**

b) The filter is also a 5×5 matrix. What is the rank (linear algebra) of this matrix? Explain your answer. **4 pts.**

c) The naïve way of convolving a $N \times N$ image with the filter kernel h_1 requires $25N^2$ multiplications (ignoring border issues). Suggest a method to reduce the number of multiplications. What is the number of multiplications needed after the reduction (ignoring border issues)? **6 pts.**

$$A = \begin{bmatrix} 0.8147 & 0.5469 & 0.8003 & 0.0357 & 0.6555 & 0.8235 & 0.7655 \\ 0.9058 & 0.9575 & 0.1419 & 0.8491 & 0.1712 & 0.6948 & 0.7952 \\ 0.1270 & 0.9649 & 0.4218 & \mathbf{0.9340} & \mathbf{0.7060} & \mathbf{0.3171} & 0.1869 \\ 0.9134 & 0.1576 & 0.9157 & \mathbf{0.6787} & \mathbf{0.0318} & \mathbf{0.9502} & 0.4898 \\ 0.6324 & 0.1706 & 0.7922 & \mathbf{0.7577} & \mathbf{0.2769} & \mathbf{0.0344} & 0.4456 \\ 0.0975 & 0.2572 & 0.4595 & 0.7431 & 0.0462 & 0.4387 & 0.6463 \\ 0.2785 & 0.4854 & 0.6557 & 0.3922 & 0.0971 & 0.3816 & 0.7094 \end{bmatrix}$$

Figure 2: 7×7 image A.

d) Figure 2 shows a 7×7 image A. Design a 3×3 correlation filter kernel for the detection of the 3×3 image region in bold. **2 pts.**

e) Figure 3 gives the output of the correlation of image A (assuming zero padding for the borders) with the filter kernel h_1 shown in Figure 1. Figure 4 shows another filter kernel h_2 which has similar properties as h_1. Compute the convolution of h_2 with image A assuming zero padding for the borders. Justify your answer. **6 pts.**

$$A \circ h_1 = \begin{bmatrix} 0.3683 & 0.4474 & 0.3923 & 0.3424 & 0.3804 & 0.4328 & 0.3517 \\ 0.4679 & 0.5972 & 0.5571 & 0.5202 & 0.5153 & 0.5191 & 0.4042 \\ 0.4302 & 0.5909 & 0.6240 & 0.6082 & 0.5419 & 0.4802 & 0.3449 \\ 0.3821 & 0.5248 & 0.6167 & 0.5974 & 0.4863 & 0.4253 & 0.3171 \\ 0.3105 & 0.4418 & 0.5630 & 0.5434 & 0.4090 & 0.3630 & 0.3086 \\ 0.2240 & 0.3630 & 0.4772 & 0.4522 & 0.3393 & 0.3327 & 0.3157 \\ 0.1568 & 0.2725 & 0.3400 & 0.3005 & 0.2318 & 0.2567 & 0.2560 \end{bmatrix}$$

Figure 3: Output of $A \circ h_1$.

$$h_2 = \begin{bmatrix} 0.0060 & 0.0266 & 0.0438 & 0.0266 & 0.0060 \\ 0.0266 & 0.1192 & 0.1966 & 0.1192 & 0.0266 \\ 0.0438 & 0.1966 & 0.3242 & 0.1966 & 0.0438 \\ 0.0266 & 0.1192 & 0.1966 & 0.1192 & 0.0266 \\ 0.0060 & 0.0266 & 0.0438 & 0.0266 & 0.0060 \end{bmatrix}$$

Figure 4: Another 5×5 filter kernel h_2 with similar properties as h_1.

Question 2: Fourier Transform (25 pts.)

The fourier transform of an infinite continuous signal g is defined as

$$F(g)(u) = \int_{\mathbb{R}} g(x)\, e^{-j2\pi ux} dx$$

if g is a bidimensional continuous signal, the fourier transform is defined as

$$F(g)(u,v) = \int\int_{\mathbb{R}^2} g(x,y)\, e^{-j2\pi(ux+vy)} dx dy$$

Similarly the fourier transform of 2D discrete and finite signals (i.e. images) can be defined by using zero padding and summation instead of integration as:

$$F(g)(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x,y)\, e^{-j2\pi\left(\frac{ux+vy}{N}\right)}$$

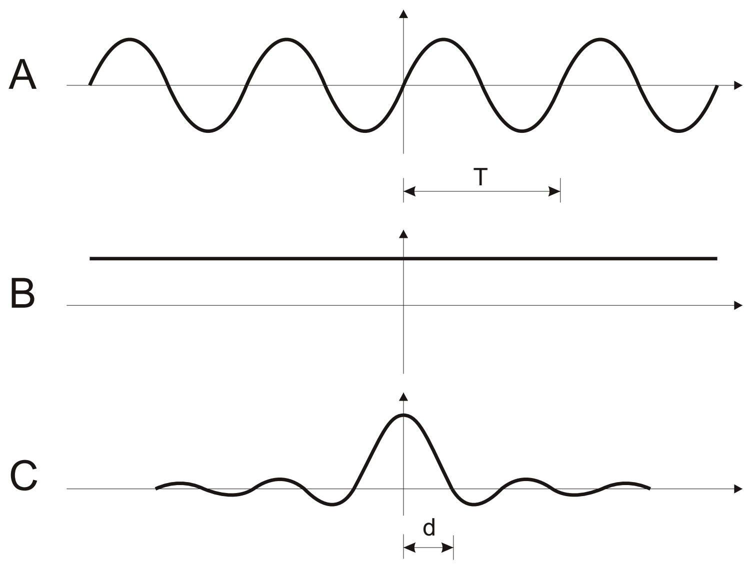


Figure 5: A, B and C are three continuous signals defined as $y = \sin(x)$, $y = c$ and $y = \mathrm{sinc}(x) = \frac{\sin(x)}{x}$ respectively.

a) Sketch the result of the convolution (in spatial or frequency domain) between the two signals A and B depicted in Figure 5. Justify your answer using the Fourier transform. **3 pts.**

b) Sketch the result of the convolution (in spatial or frequency domain) between the two signals A and C depicted in Figure 5. Does the result depend on d and T? Justify your answer using the Fourier transform. **4 pts.**

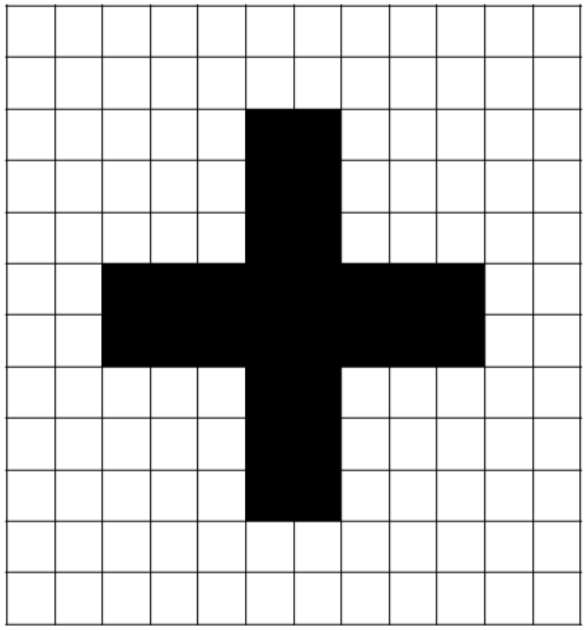
c) A digital image $f(x,y)$ is of size 1000×1000. If $F(u,v)$ denotes the Discrete Fourier Transform of $f(x,y)$, compute the Discrete Fourier Transform of the following functions in terms of $F(u,v)$: **4 pts.**

1. $f1(x,y) = 2f(x,y)$
2. $f2(x,y) = f(x+100,y)$
3. $f3(x,y) = f(x,y) + 100$

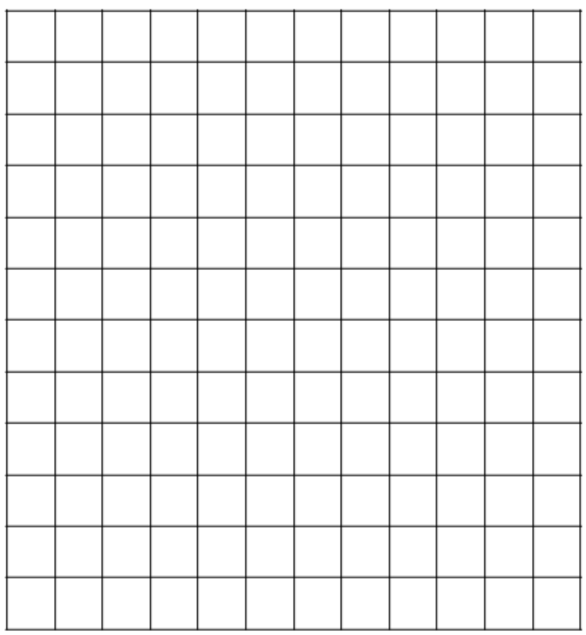
d) Consider a 1-D ideal low pass filter f_{lp} with cutoff frequency D_{lp} and a 1-D ideal high pass filter f_{hp} with cutoff frequency D_{hp}. Using these filters, under what conditions can one obtain a band pass and a band reject filter. Justify your answer. **6 pts.**

e) Refer the image I shown in Figure 6(a). The image is 12×12 pixels in size and contains a plus shaped object. The background and foreground pixels have value 1 and 0 respectively. The image is filtered using the mask $[-1, 0, 1]$. **8 pts.**

1. Show the resulting image in Figure 6(b). (Ignore the border issues)
2. If $F(u, v)$ be the discrete Fourier Transform of the resulting image, what is the value of $F(0, 0)$.
3. Let I_1 be the output of part 1 and I_2 be the result of convolving the image with the filter $[-1, 0, 1]$. Compute I_2 in terms of I_1. Justify your answer. (Ignore the border issues)



(a) Input Image I



(b) Output Filtered Image

Figure 6: (b) shows the output of applying the filter $[-1, 0, 1]$ to the image in (a)

Question 3: Optical Flow (20 pts.)

a) In Lucas-Kanade optical flow, the optical flow is computed for an image patch instead of a single pixel. Explain why! **4 pts.**

b) What issues arise when computing optical flow of an image patch which contains a grayvalue edge compared to a grayvalue corner. Is there a specific type of texture that is especially good for optical flow tracking? **4 pts.**

c) What do you do when an image contains regions with homogeneous texture (e.g. regions with uniform color). How can optical flow be computed in this case. Describe 2 possible methods.: **4 pts.**

d) For optical flow computation image derivatives in x and y need to be computed. Write down the necessary filter masks to do this by filtering. **2 pts.**

e) Explain how optical flow information can be used for video compression. **2 pts.**

f) What image artifacts can appear in video compression if you compute frame $n+1$ by applying forward propagation to frame n using the optical flow computed from image n to $n+1$. **2 pts.**

g) When is video compression using optical flow ineffective? **2 pts.**

Question 4: PCA and Data Compression (25 pts.)

For implementing a PCA based data driven compression, a training set of grayscale images containing faces is provided. $\mathcal{I}_o$ is a particular image containing a face that does not belong to the training set. All images are 100×100 in size and require 10^4 floating points for storage.

a) Describe how the Eigenfaces can be obtained from the training set using PCA. **3 pts.**

b) The image $\mathcal{I}_o$ is to be compressed to 1% of the original file size by using Eigenfaces obtained through PCA. How many Eigenfaces need to be used in the compression? Which Eigenfaces should be used and why should these be used in preference over other Eigenfaces? **4 pts.**

c) Using these Eigenfaces, how can you compress the image $\mathcal{I}_o$? Specify clearly what you consider as the compressed version of the image. **4 pts.**

d) How can you decompress the compressed version obtained in part (c)? **2 pts.**

e) How many Eigenfaces are required for storing an image like $\mathcal{I}_o$ without loss ? **2 pts.**

f) Using the Eigenfaces how can you generate a quick, even if inaccurate, preview of images like $\mathcal{I}_o$ that gradually improves to 100% accuracy if provided enough time to read the entire file? You can create your own format for storing the image. **5 pts.**

g) What problem could you face if you try to use this quick preview method for an arbitrary 100×100 grayscale image like a photo of a chair? **2 pts.**

h) What are the advantages of building image pyramids like Gaussian Pyramids? **3 pts.**

Question 5: Colors and Shading (30 pts.)

a) **The CMY Color Model**

i) Describe the purpose of the CMY color model in one sentence. **1 pt.**

ii) For practical purposes the CMY model can be modified by adding a black channel K. What value should K take if the black channel should be used maximally when printing a color (C,M,Y)? **1 pt.**

iii) Due to chemical degradation processes, the colors of a painting have started to fade out after several decades. Measurements have given the following changes: In the CMY color model, C lost 10%, M 20% and Y 0%. You now have taken a photo of this painting and would like to restore the original colors in the photo. What transformation do you have to apply to your RGB image in order to retrieve the original image? **4 pts.**

b) **Color Reproduction**

Assume you are given a projector with three primaries $\mathbf{A} = (2, 1, 7)^T$, $\mathbf{B} = (3, 1.5, 0.5)^T$ and $\mathbf{C} = (1.6, 2, 0.4)^T$ in XYZ coordinates.

i) Compute their corresponding CIE chart xy-coordinates $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ and mark their location in the chart below. Further, give also their dominant wavelength values for the theoretical white point $\mathbf{w} = (1/3, 1/3)^T$ (it suffices to determine them geometrically). **4 pts.**

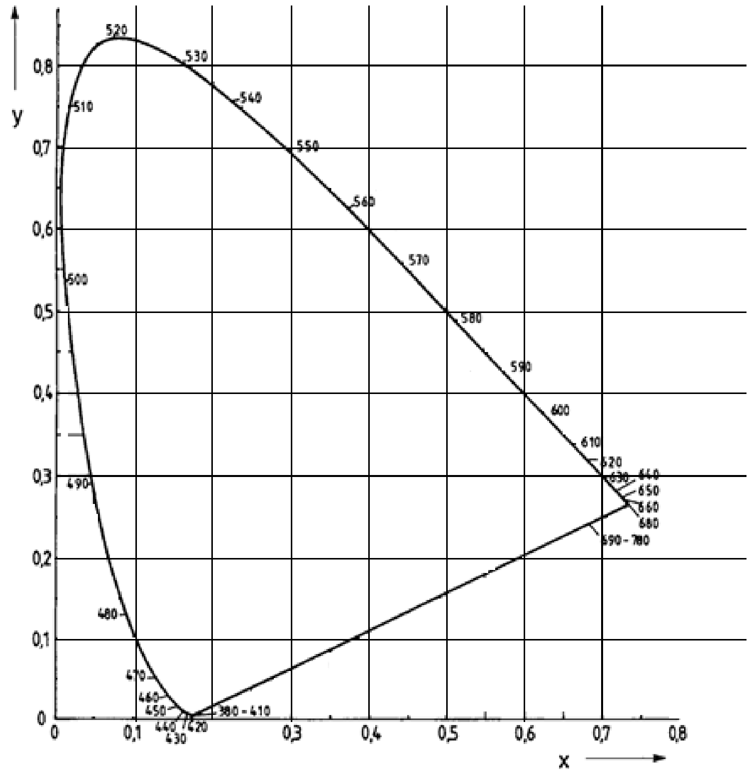
ii) A second projector should now be installed which projects onto the same screen. You have the choice between a first projector with primary xy-coordinates $\mathbf{d_1} = (0.25, 0.1)^T$, $\mathbf{e_1} = (0.5, 0.4)^T$ and $\mathbf{f_1} = (0.1, 0.4)^T$, and a second one with primaries $\mathbf{d_2} = (0.1, 0.2)^T$, $\mathbf{e_2} = (0.5, 0.3)^T$ and $\mathbf{f_2} = (0.2, 0.6)^T$. Which one would you choose in order to get as close as possible to real life colors? Explain your answer. Furthermore, mark the reproducible colors of the new projector system into the chart below. **3 pts.**

The two projector system should now be used to transmit a secret image to an agent in the audience. For this sake, the agent's visual system has been augmented such that he now has an additional fourth sensor to the three cones of the human visual system. The secret grayscale image is given by the scalar values $g(\mathbf{x})$, while the original image perceived by a normal human is described by 3-vectors $\mathbf{o}(\mathbf{x})$ given in color coordinates of the first projector. $\mathbf{x}$ denotes the pixel position.

iii) Why does the additional cue allow the agent to perceive information not visible for a normal non-augmented human? **2 pts.**

iv) The encoding of the grayscale image should be realized by having both projectors representing the same image and weighting the contribution of each by the grayscale values $g(\mathbf{x})$. How does this restrict the colors that can be used in the image $\mathbf{o}(\mathbf{x})$? **2 pts.**

v) Give the matrix $\mathbf{T}$ that linearly transforms a color vector given in primary coordinates of the first projector into corresponding coordinates of the second. Assume that the primaries are just given symbolically as $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ for the first, and $\mathbf{D}$, $\mathbf{E}$ and $\mathbf{F}$ for the second projector. **2 pts.**



c) **BRDF and Phong Illumination Model**

i) An anisotropic BRDF has four parameters. The inclination angles θ_i and θ_r, as well as the azimuth angles ϕ_i and ϕ_r. How do they relate to the three parameters of the isotropic BRDF model? **2 pts.**

ii) Make a sketch, illustrating the relationship between the anisotropic BRDF parameters θ_i, ϕ_i, θ_r, ϕ_r and the Phong Illumination Model variables $\mathbf{N}$,$\mathbf{L}$,$\mathbf{V}$. **2 pts.**

iii) You are given the four images (a)-(d), taken with different parameters n and ratios k_d/k_s of the Phong Illumination model. Place them at the correct location in the graph of Figure 7. **2 pts.**

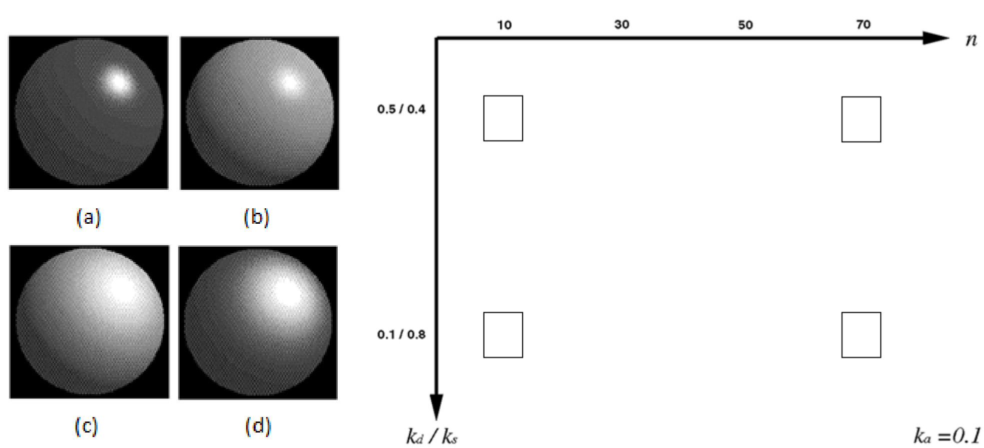


Figure 7: Task c) iii)

d) **Shading Models**

i) For each of the three shading models in Table 1, describe per which graphics primitive the lighting evaluation happens and which attribute is interpolated. **2 pts.**

	flat	Gouraud	Phong
lighting evaluation per			
interpolation			

Table 1: Task d) i)

ii) Can a highlight be generated inside a triangle using Phong shading? Is it possible with flat or Gouraud shading? Explain your answer. **3 pts.**

Question 6: Transformations and Projections (30 pts.)

a) **Quaternions**

Translations and rotations in 3D can be represented by 4×4 matrices using homogeneous coordinates or by quaternions.

i) Given the following transformation matrix $\mathbf{A}$, what are the corresponding quaternions $\mathbf{r}$ and $\mathbf{t}$ ($\mathbf{p}' = \mathbf{r}\mathbf{p}\bar{\mathbf{r}} + \mathbf{t}$), representing the same transformation? **4 pts.**

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ii) Transform the point $\mathbf{p} = (0, 1, 0)^T$ using these quaternions (the method used has to be identifiable by means of your solution)! **3 pts.**

b) **Perspectively Corrected Interpolation**

Values can be interpolated between vertices. If we assign a color to each of the vertices of a triangle in screen space, then the color values can be linearly interpolated between them.

i) In which part of the rendering pipeline are these interpolations done? **1 pt.**

ii) Given are the values c_i at each of the vertices $i = \{1, 2, 3\}$ of the triangle. The corresponding vertices have 2D coordinates $(0, 0)^T$, $(0, 1)^T$ and $(1, 1)^T$. Write down the formula for the linearly interpolated value c at a position $(u, v)^T$ (i.e., using barycentric coordinates). **5 pts.**

iii) The above interpolations in screen space are not correct when the 3D scene is rendered using perspective projections. Why? What would be the correct interpolation method? (just describe with words) **3 pts.**

c) **Non-flat Display**

For a planetarium a bent display is developed to show visitors projections of stars and planets. The surface of the display is a segment of a sphere with radius $R = 1$. The center of the sphere is the position of the viewer $(0, 0, 0)^T$ and the screen spans horizontally and vertically an angle of $\alpha < 180$. An illustration is shown in Figure 8 left.

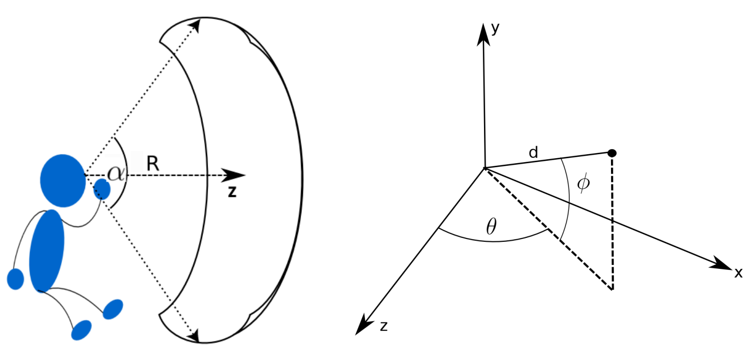


Figure 8: Planetarium and spherical coordinates

The individual "pixels" of the display are accessed by spherical coordinates θ and ϕ (see Figure 8 right):

$$x = d \sin(\theta) \cos(\phi)$$

$$y = d\sin(\phi)$$
$$z = d\cos(\theta)\cos(\phi)$$

$\theta = 0, \phi = 0$ represents the center of the screen. It is $-\alpha/2 \leq \theta \leq \alpha/2$ and $-\alpha/2 \leq \phi \leq \alpha/2$.

When simply displaying an image on this screen, rendered using a standard perspective camera model for flat displays, the virtual world would appear warped because of the bent surface of the screen. Therefore we have to develop a renderer that takes this into account.

i) Write down the formulas for projecting a 3D point $\mathbf{p} = (x, y, z)^T$ onto the spherical coordinates θ, ϕ. You can assume that all objects are entirely visible. **4 pts.**

ii) We want to do this warping entirely in a vertex shader. Write down the proper code (no pseudo-code) of a function that does the warping for this shader. It gets as input the vertex coordinates in world space and should return the spherical coordinates. You can assume that the model-view matrix is set correctly. **4 pts.**

```
function sphericalDisplayProjection(
     in mat4 modelview_matrix, in vec3 v_world,
     out float theta, out float phi, out float d)
{



}
```

iii) A vertex shader with the above function correctly maps the vertices into the screen space. However, if our scene contains large primitives, such as two triangles with textures of the night sky spanning the entire screen, the linear interpolation in angular coordinates yields a problem. What happens and why? **3 pts.**

iv) To overcome the problems of interpolation, we could also do the following: Assume we use a standard vertex shader to project the scene with the night sky onto a 2D plane. To display this image correctly on the new bent display we write a fragment shader that does the warping. However, with this approach we run into problems if α is close to 180 degrees. What happens and why? **3 pts.**

Question 7: Anti-Aliasing and Texture Filtering (30 pts.)

a) **Temporal Aliasing**

The three spoke wheel in Figure 9 revolves exactly ten times a second.

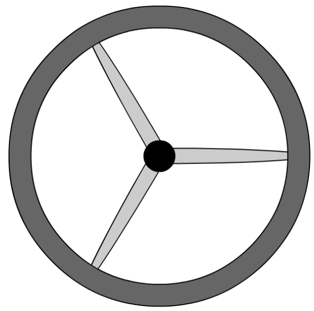


Figure 9: Three spoke wheel

i) When creating an animation of the wheel, what is the minimum number of images per second you need to generate so the wheel appears to rotate at the correct speed? **2 pts.**

ii) Assume your display device is not able to achieve the frame rate computed in the previous question. When creating an animation of the wheel for this particular device, what can you do to avoid temporal aliasing? **1 pt.**

b) **Reducing Aliasing**

i) When discretizing a continuous signal, there are two distinct strategies to reduce aliasing. What are they? **1 pt.**

ii) For each of the two strategies mentioned above, give an example of where and how they are typically used in computer graphics. **1 pt.**

c) **Low-Pass Filters**

i) In the continuous case, which kind of low-pass filter would you apply to an arbitrary signal to make sure the discretization of the filtered signal does not result in any aliasing at all? What does the kernel of this filter look like? **2 pts.**

ii) Why is this filter not commonly used in practical applications? Which low-pass filter would you use instead and why? **2 pts.**

d) **B-Spline Filters**

i) A B-Spline filter of high order is a good approximation of a Gaussian filter. What advantages does the B-Spline filter have over the Gaussian filter? **1 pt.**

ii) Sketch the kernels of the B-spline filters of order 1 to 3, i.e. $g_i(x)$, $1 \leq i \leq 3$. Indicate the support of each kernel (the range of x where $g_i(x)$ is non-zero) and compute the value $g_i(0)$ for each kernel. **4 pts.**

iii) Show that all B-Spline filters are normalized by construction. In other words, show that $\int_{-\infty}^{\infty} g_n(x)\, dx = 1$ for all $n \geq 1$. Hint: Proof by induction. **4 pts.**

e) **Anisotropic Texture Filtering**

i) Explain the basic idea behind anisotropic texture filtering. In which particular cases does anisotropic filtering perform much better than mip-mapping with trilinear filtering? **2 pts.**

ii) A tilted, textured plane at distance d is parameterized as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} + \begin{pmatrix} \cos(\alpha) \\ 0 \\ \sin(\alpha) \end{pmatrix} s + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} t$$

in view coordinates. The texture coordinates are $(s,t)^T$. The field of view (FOV) of the camera is 90 degrees in horizontal and vertical direction and the camera looks in the $+z$ direction (see Figure 10). A circular pixel at position $(u,v)^T$ in screen coordinates ($-1 \leq u, v \leq 1$) and pixel radius r (also in screen coordinates) will map to an ellipse on the parameterized plane. Find the equation to compute the extent of the ellipse in s direction, for an arbitrary u and $v = 0$. **6 pts.**

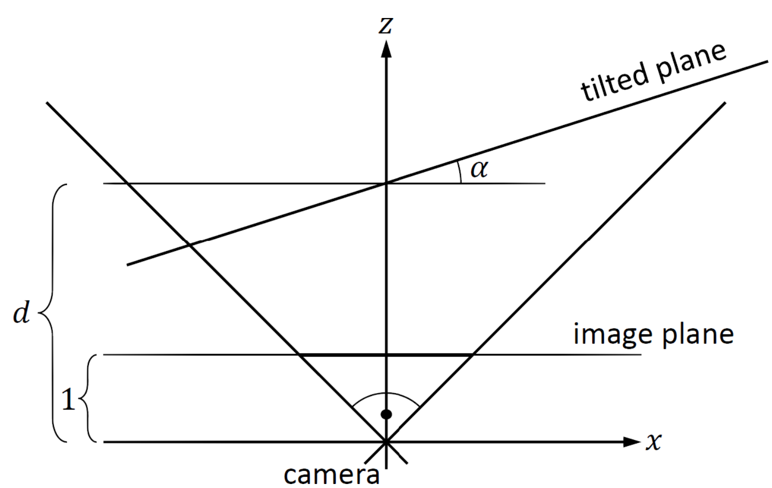


Figure 10: Tilted plane in front of a camera

iii) Given the two orthogonal axes `vec2 a` and `vec2 b` of an ellipse at position `p` in texture space, write a fragment shader function that performs anisotropic filtering by *approximating* the average of the color values of texture `txt` inside the ellipse. Use the following template: **4 pts.**

```
vec3 performAnisoFiltering(vec2 a, vec2 b, vec2 p, sampler2D txt)
{
    // Contains 12 random 2D vectors with length <= 1:
    vec2 filterKernel[12];
    filterKernel[0] = vec2(0.326212, 0.40581);
    filterKernel[1] = vec2(-0.840144, 0.07358);
    // ...
    // (other values omitted)

    vec3 c = vec3(0, 0, 0);










    return c;
}
```