ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visual Computing
AS 2016

Prof. M. Gross (Computer Graphics Lab)
Prof. O. Hilliges
(Advanced Interactive Technologies Lab)

Y. Aksoy, F. Pece, J. Song
A. Milliez, V. Megaro, R. Roveri
C. Öztireli

Final Exam

08 February 2017

First and Last name: ____________________

ETH number: ____________________

Signature: ____________________

General Remarks

- At first, please check that your exam questionnaire is complete (there are 14 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 12 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.
- Please do not use a pencil or red color pen to write your answers.
- You may provide at most one valid answer per question. Invalid solutions must be canceled out clearly.

	Topic	Max. Points	Points Achieved	Visum
1	Principal Component Analysis	10		
2	Filtering	14		
3	Optical Flow	20		
4	Sampling	16		
5	Morphological Operators and Texture	13		
6	Image Features	17		
7	OpenGL and Rendering	32		
8	Light and Colors	24		
9	Transformations	22		
10	Physics Simulation	12		
Total		180		

Grade: ....

Question 1: PCA (10 pts.)

For implementing a PCA based data driven compression, a training set of greyscale images (i.e., single-channel images) containing faces is provided. I_0 is a particular image containing a face that *does not* belong to the training set. All images are $100x100$ pixels in size.

a) Explain why PCA can be used for compression. How does it modify the data? **2 pts.**

b) Describe how the Eigenfaces can be obtained from the training set using PCA. **2 pts.**

c) Which Eigenfaces should be used, and how can you use them to compress the image I_0? Assuming you need to implement your own file format for this compression algorithm, what would you store in the compressed file to minimize usage space (i.e., specify clearly what you consider as the compressed version of the image)? **3 pts.**

d) How can you decompress the compressed version of the image I_0 obtained in part (c)? **2 pts.**

e) Assume you have built a compressing matrix P using the first 20 Eigenfaces. You are now given the task to detect a face in an unseen image, with arbitrary shape. How would you implement this? Outline the steps involved. **1 pt.**

Question 2: Filtering (14 pts.)

a) Given a low-pass filter, we can get the high-pass filtered image by first filtering it using the low-pass filter and subtracting it from the original image. Alternatively, following the same idea, we can generate a high-pass filter kernel from a low-pass filter kernel.

Compute a high-pass filter kernel using the low-pass filter kernel given below. **3 pts.**

0.03	0.11	0.03
0.11	0.44	0.11
0.03	0.11	0.03

b) Briefly explain why the low-pass filter kernel below can not directly be used to generate a high-pass filter kernel as we did in Part (a). Explain how we can modify the method above to generate a high-pass filter kernel using the kernel below. **2 pts.**

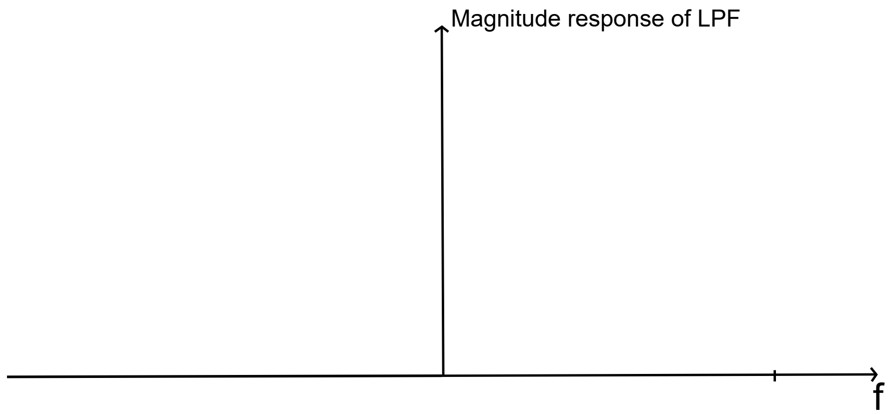
1	2	1
2	5	2
1	2	1

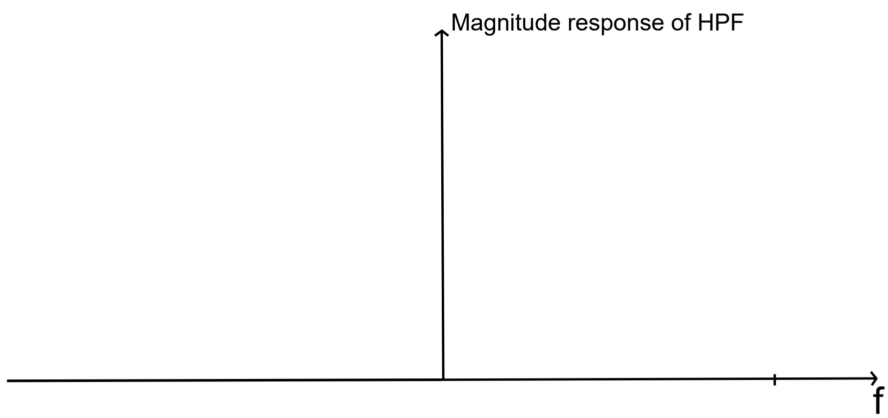
c) Using a high-pass filter and a low-pass filter, we can get a band-pass filtered image. Explain how this can be accomplished without explicitly computing a band-pass filter kernel. **3 pts.**

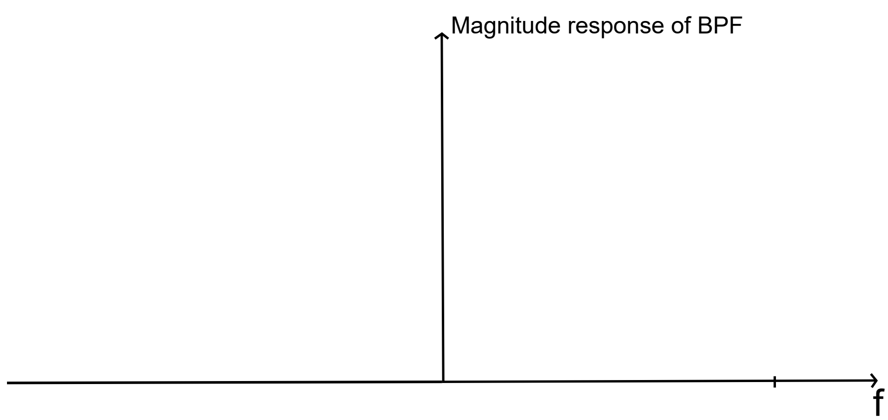
d) How can one generate a band-pass filter kernel using a low-pass and a high-pass filter kernel? **2 pts.**

e) Assume that we generated a band-pass filter from two one-dimensional filters, a low-pass and a high-pass. Roughly draw in the space below the magnitude responses you expect.

4 pts.







Question 3: Optical flow (20 pts.)

a) Let $I(x, y, t)$ be a video sequence taken by a rigidly moving camera observing a rigid scene. Assume that between two consecutive frames, there is an affine change in the intensities, such that:

$$I(x + u, y + v, t + 1) = aI(x, y, t) + b$$

where $u(x, y)$ and $v(x, y)$ represent the optical flow (motion parameters) and $a(x, y)$ and $b(x, y)$ represent photometric parameters. Propose a linear system of equations for estimating (u, v, a, b) from the image brightness I. Derive all the equations that allow you to reach a solution (hint: you may get your inspiration from the Lucas-Kanade method). What is the minimum size of a window around each pixel that allows one to solve the problem? **9 pts.**

b) The algorithm outlined at step (a) can be used to estimate the optical flow of an image sequence. However, it makes three assumptions that, when not fulfilled, can cause the algorithm to fail. What are these three conditions (hint: what are the Lucas-Kanade algorithm conditions)? When does the "aperture problem" arise in computing optical flow? **3 pts.**

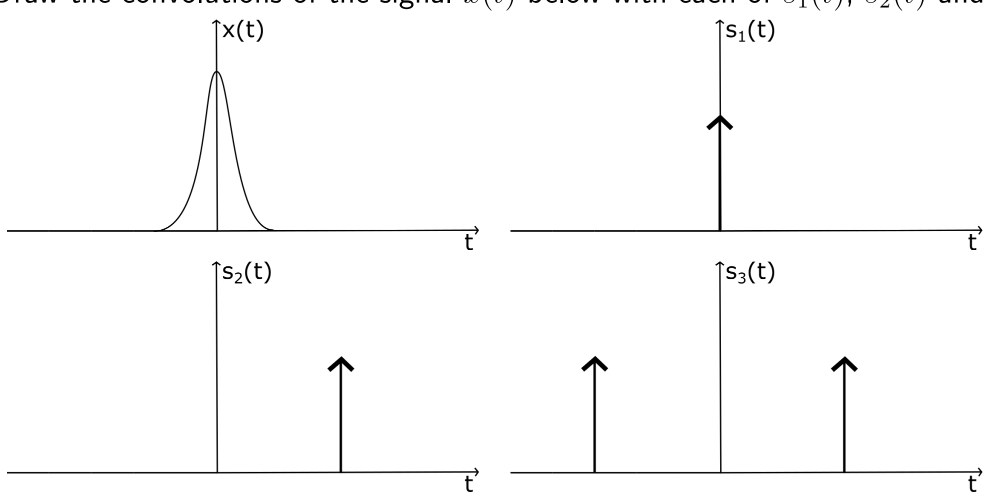
c) State the main steps of an algorithm for optical flow computation when using iterative refinement. **3 pts.**

d) The original Lukas-Kanade algorithm works well for small motion. How can it be improved when the motion is large? State the main steps of an algorithm for estimating the optical flow on larger motion displacements. **2 pts.**

e) Explain how optical flow can be used for video compression (give a coarse outline in no more than three lines, assuming only flow and standard image compression is used in the process). When is video compression using optical flow ineffective? What other application can you think of for optical flow? **3 pts.**

Question 4: Sampling (16 pts.)

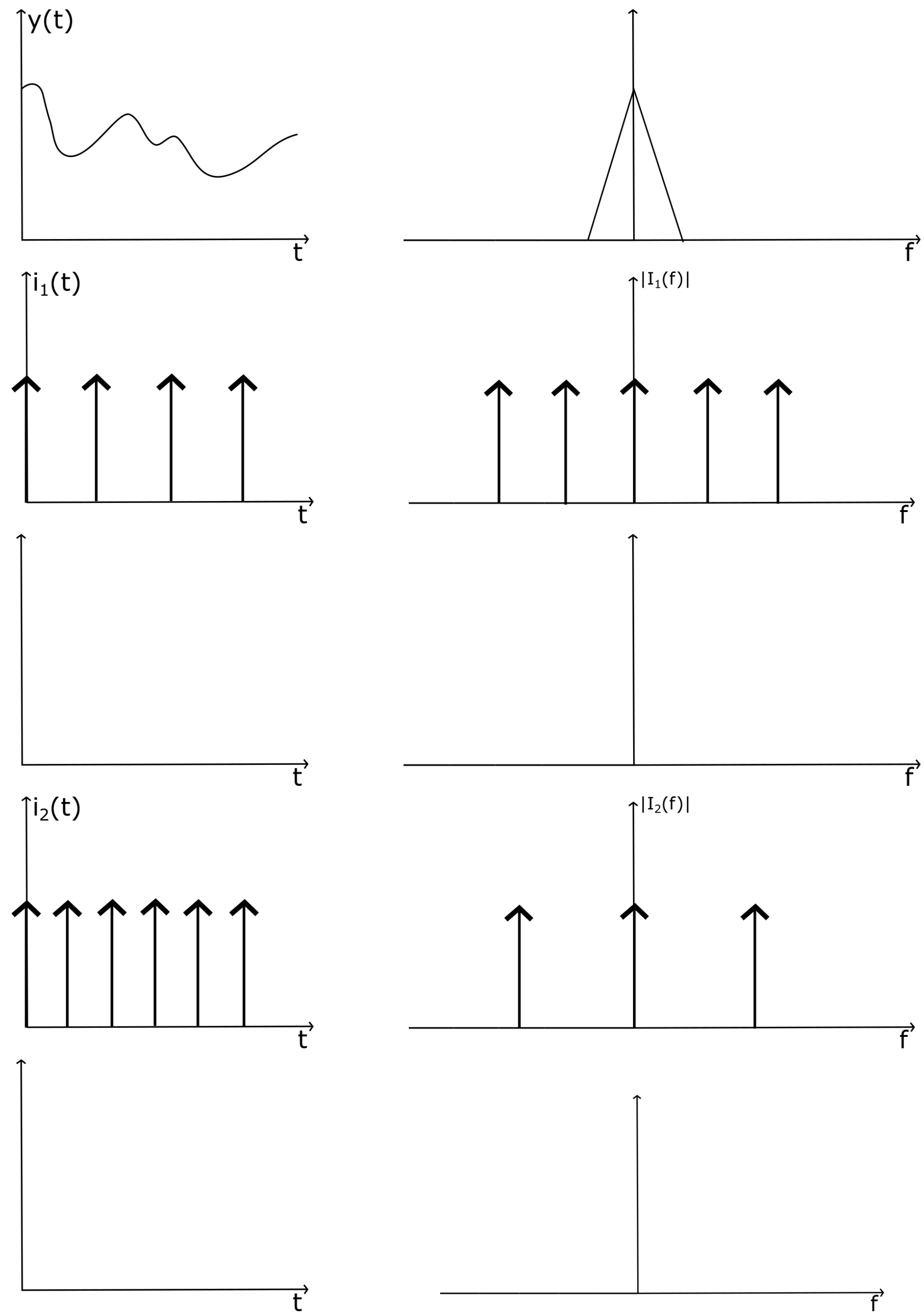
a) Draw the convolutions of the signal $x(t)$ below with each of $s_1(t)$, $s_2(t)$ and $s_3(t)$. **5 pts.**



b) What is the convolution property of Fourier transform? **5 pts.**

c) Given the two impulse trains and their Fourier transforms, draw the sampled signals and their corresponding Fourier transforms in the corresponding spaces provided below. **6 pts.**

Time domain Frequency domain



Question 5: Morphological operators and Texture (13 pts.)

a) Define opening and closing of an image I by a structuring element S. Give the set formulation for any basic operators you may use.

3 pts.

b) A pair of images is shown on figure 1. Image A was processed by a morphological operation to produce image B. A structuring element S of size 3 x 3 was used in this transformation. Choose the operation used from the list provided. **2 pts.**

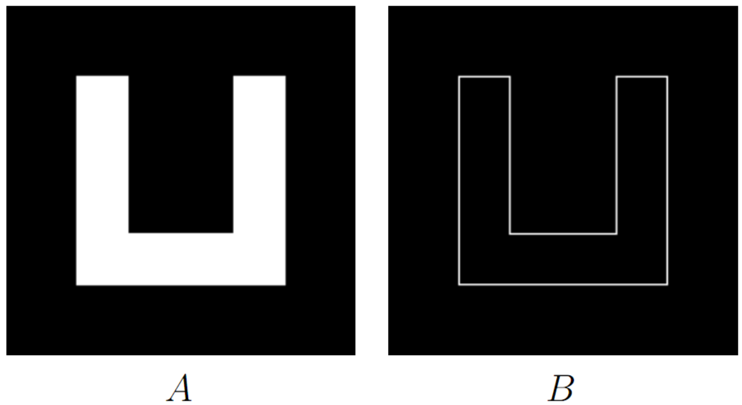


Figure 1

(a) $B = (A \ominus S) \cap A^c$
(b) $B = (A \oplus S)^c \cup A$
(c) $B = (A \oplus S) \cap A^c$
(d) $B = (A \ominus S)^c \cup A^c$

c) One way of representing textures is by using oriented pyramids.

i) Briefly give an outline of this method. **1 pt.**

ii) Why do we use a pyramid of the image instead of different sized filters in order to get the scale information? **1 pt.**

d) Why is it that histograms are not a good choice for characterizing textures? Can you think of a more adapted tool to characterize textures based on the same principle as histograms? **2 pts.**

e) Texture synthesis by using image-based methods can be fairly simple and efficient. One of the methods seen in class is the Chaos Mosaic method.

i) State the three main steps of this method. **2 pts.**

ii) For which kind of textures does this method work the best? When does it fail? Give an example of each kind of texture. **2 pts.**

Question 6: Image Features (17 pts.)

a) Given a filter shown below, What is the effect of convolving an image with this filter for $k = 0$ and $k = 2$ **2 pts.**

$$G = \begin{bmatrix} -k/8 & -k/8 & -k/8 \\ -k/8 & k+1 & -k/8 \\ -k/8 & -k/8 & -k/8 \end{bmatrix}.$$

b) $G(x, y, \sigma)$ defines a 2D image filter. Show that the filter is separable. **2 pts.**

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^3 + x^2 + xy^2 + y^2}{2\sigma^2 x + 2\sigma^2}\right)$$

c) Starting from the definitions of correlation and convolution, prove that the outputs from correlation and convolution are the same for the filter $G(x, y, \sigma)$ on the image $I(x, y)$. **3 pts.**

d) Name the filter $G(x, y, \sigma)$ and describe its effect on an image. **2 pts.**

e) Harris corners are selected from analyzing the eigenvalues λ_1 and λ_2 of the "Structure Matrix"' M from the local displacement sensitivity function

$$S(\Delta x, \Delta y) = (\Delta x, \Delta y) M (\Delta x, \Delta y)^T$$

Explain the observations of (i) $\lambda_1 >> \lambda_2$ and (ii) λ_1 and λ_2 are both large. **2 pts.**

f) Explain why Harris corners are not scale invariant but rotation invariant. **2 pts.**

g) Suggest a way to make Harris corners scale invariant. **1 pt.**

h) Briefly illustrate how to fit a line feature by using hough transform. What is the disadvantage of using $y = mx + c$ as the line equation? What is the solution? **3 pts.**

Question 7: OpenGL and Rendering (32 pts.)

a) **General Questions**

i) Two objects, A and B, are rendered in OpenGL. A is placed in front of B, and is not fully opaque. However, B is not visible through A.
Give two possible explanations for this apparent bug. **2 pts.**

ii) When using a shadow map, the shadow cast by an object onto a surface exhibits jaggy contours. Explain why. **2 pts.**

iii) Why does texture mip-mapping reduce aliasing? **2 pts.**

iv) Explain the difference between bump mapping and displacement mapping. **2 pts.**

b) **Vertex Shader**

We consider the following vertex shader.

```
#version 440

uniform mat4 projectionMatrix;
uniform mat4 modelviewMatrix;
uniform mat3 normalMatrix;

in vec3 inPosition;
in vec3 inNormal;

out vec3 outPosition;
out vec3 outNormal;

void main()
{
  outPosition = inPosition;
  outNormal = normalMatrix * inNormal;

  gl_Position
    = projectionMatrix * modelviewMatrix * vec4(inPosition, 1.0f);
}
```

i) What is the purpose of `projectionMatrix` and `modelviewMatrix`? **4 pts.**

ii) The input normal is transformed using `normalMatrix`, unlike the input position. Why is that? **2 pts.**

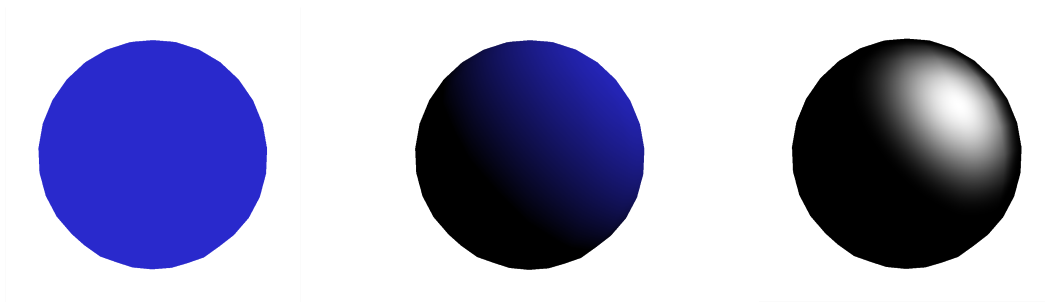
iii) The computed `gl_Position` is a vector of 4 components, what do they represent? **3 pts.**

c) **Fragment Shader**

```
#version 440

in vec3 outPosition;
in vec3 outNormal;

uniform vec3 eyePosition;
uniform vec3 lightDir;

void main()
{
  vec3 normal = normalize(outNormal);

  vec3 baseColor = vec3(0.15f, 0.15f, 1.0f);

  // COMPUTE COMPONENT 1
  vec3 component1 = dot(lightDir, normal) * baseColor * 0.9f;

  // COMPUTE COMPONENT 2
  vec3 component2 = baseColor;

  // COMPUTE COMPONENT 3
  vec3 R = 2.0f * dot(lightDir, normal) * normal - lightDir;
  R = normalize(R);

  vec3 V = eyePosition-outPosition;
  V = normalize(V);

  float value = max(0.0f, dot(V, R));
  value = pow(value, 2.0f);
  vec3 component3 = vec3(value);

  // COMPUTE FRAGMENT COLOR
  vec3 color = component1 + component2 + component3;
  gl_FragColor = vec4(color, 1.0f);
}
```

i) This fragment shader implements the Phong shading model. Give the names of components 1, 2 and 3. **3 pts.**

ii) Identify those three components in Figure 2. (We are asking for their names, not their number in the previous question.) **3 pts.**

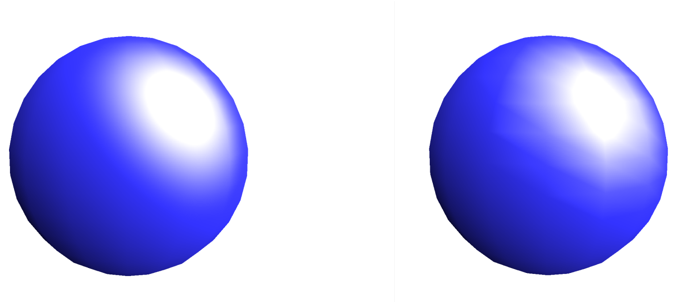


(a) (b) (c)

Figure 2: The three components of the Phong shading model.

d) **Alternative shading algorithm**

Figure 3 shows in `(a)` a sphere rendered with the vertex and fragment shaders presented above. In `(b)` we see a slightly different shading algorithm.



(a) (b)

Figure 3: A sphere lit with Phong shading (left) and with an alternative algorithm (right).

i) What is the name of the shading algorithm used in `(b)`? **1 pt.**

ii) Why does the Phong shading algorithm look smoother than the one seen in `(b)`? **2 pts.**

iii) How would you adapt the shaders used to render `(a)` in order to render `(b)`? We want explanations, not code. **4 pts.**

iv) Give an advantage of using `(b)` over Phong shading. **2 pts.**

Question 8: Light and Color (24 pts.)

a) **Color Spaces**

i) What is the main advantage of the Lab color space over the RGB color space? Describe briefly how the CIE chart can be transformed to obtain the Lab space. **2 pts.**

ii) Where are the CMY and HSV color spaces usually applied? Provide one example for each color space. **2 pts.**

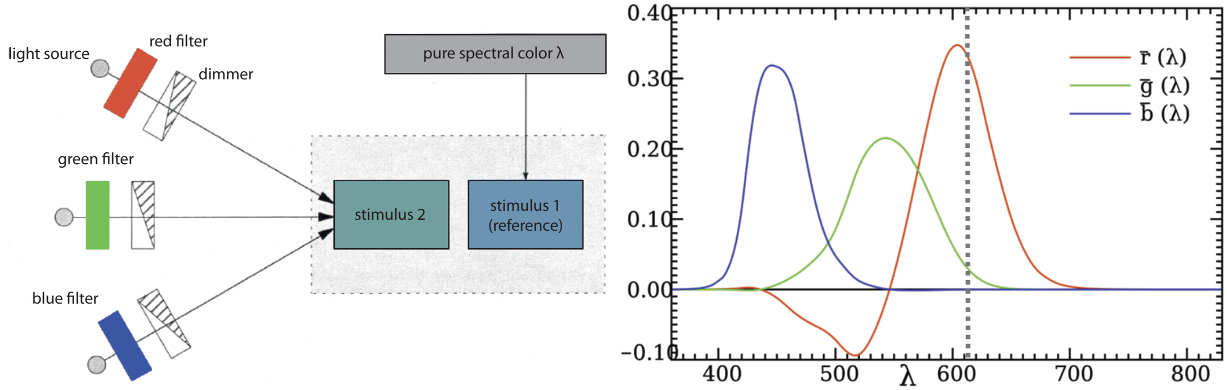


Figure 4: Left: CIE 1931 Experiment Setup. Right: Experiment Results.

b) **CIE Experiment**

i) In 1931 CIE performed an experiment. Figure 4 (left) shows a diagram of the setup. With the help of this diagram explain briefly how the experiment was conducted and the motivation behind it. **4 pts.**

ii) In Figure 4 (right) the results of the CIE experiment are shown. Explain what a vertical line (like the dotted one shown in the plot) describes. **2 pts.**

iii) What is a main difference between the color matching functions of the CIE RGB color space and the ones of the CIE XYZ color space? **2 pts.**

c) **CIE Color Spaces**

Consider the 3 following primaries given in the CIE xyY color space:

	x	y	Y
c_1	0.2	0.6	12
c_2	0.2	0.05	10
c_3	0.6	0.3	7

i) Explain briefly the perceptual meaning of the xy and Y axes in the CIE xyY color space. **2 pts.**

ii) Figure 5 is an empty CIE xy Chromaticity diagram. Plot c_1, c_2 and c_3 on the diagram. **1 pt.**

iii) Draw a line on the diagram where colors sharing the same dominant wavelength as c_1 lie. **1 pt.**

iv) What does the position of c_1 on this line characterize? **1 pt.**

v) What is the name of the connection between 770nm and 380nm? What do the points on that line represent? **2 pts.**

vi) Provide the transformation formulas from CIE XYZ to CIE xyY and from CIE xyY to CIE XYZ. **2 pts.**

vii) Compute the sum of the three primaries c_1, c_2 and c_3 in the XYZ color space. Plot the resulting color on the CIE xy Chromaticity diagram (Figure 5). **3 pts.**

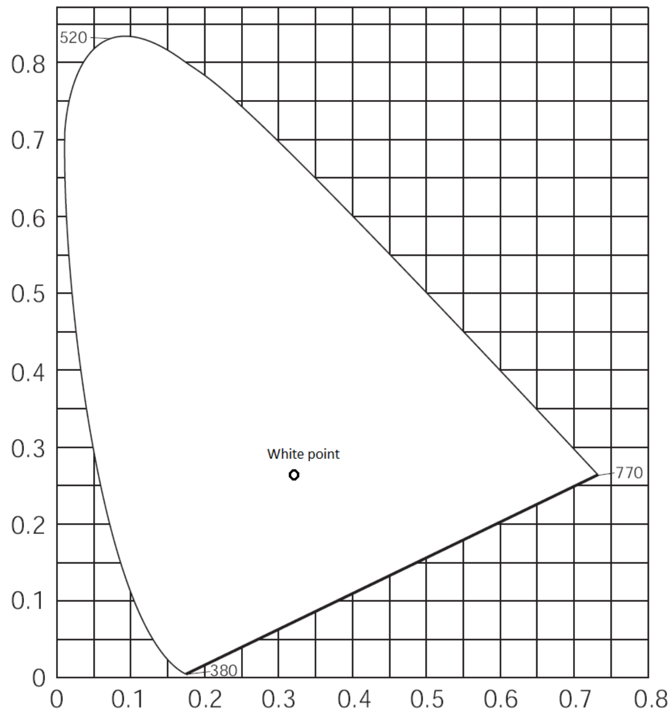


Figure 5: CIE xy Chromaticity diagram

Question 9: Transformations (22 pts.)

a) **Multiple Choice**

Every multiple choice question can have any number of correct solutions. It can go from zero (all false) to the number of choices (all correct). The correct answers can just be crossed, without the need of explanation. For every correct answer that is not crossed and for any crossed false answer, one point is removed. The minimal number of points per question is zero. Below we show an example of multiple choice answer:

The Apollo 11 landed on the moon:

☒ before 1980.
☒ in 1969.
☐ in 1492.
☒ with commander Neil Armstrong.

i) All 3D rotation matrices R have the property that $R^{-1} = R^T$.
☐ True
☐ False **1 pt.**

ii) All 3D matrices R that have the property $R^{-1} = R^T$ are 3D rotation matrices.
☐ True
☐ False **1 pt.**

iii) Talking about rotation matrices, mark all correct answers:

- ☐ Any linear combination R_{comb} of two 3D rotation matrices R_1 and R_2 in the form of $R_{comb} = \alpha R_1 + \beta R_2$, is a rotation matrix.
- ☐ Assume a 3D rotation matrix R with rotation axis $\mathbf{u}$. The rotation axis and its negative, $-\mathbf{u}$, are the only real unitary vectors that are not affected by the transformation R.
- ☐ Assume two 3D rotation matrices R_1 and R_2 with rotation axes $\mathbf{u}_1$ and $\mathbf{u}_2$. The resulting matrix R_{res} of the matrix multiplication $R_{res} = R_2 \cdot R_1$ has a rotation axis $\mathbf{u}_{res}$ which can be computed using the vector cross product $\mathbf{u}_{res} = \mathbf{u}_2 \times \mathbf{u}_1$.

3 pts.

b) **Matrices and Rotations**

i) The following matrices describe 3D transformations in homogeneous coordinates. For each, describe that transformation.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3 pts.

ii) We describe a rotation r by its rotation axis $\mathbf{u}$ and a scalar rotation angle θ. Now assume that $\pi < \theta < 2\pi$ and that $\theta = \pi + \alpha$ for a scalar value α. Find a new rotation axis $\mathbf{u}'$ and a rotation angle θ' such that they describe a rotation r' which is equivalent to r, but with $0 < \theta' < \pi$. Express $\mathbf{u}'$ and θ' as function of $\mathbf{u}$, θ and α. Explain why your solution is correct. **3 pts.**

c) **Quaternions**

We are going to perform a transformation $T : \mathbb{R}^{3\times 1} \mapsto \mathbb{R}^{3\times 1}$ to a point $\mathbf{p}$ using quaternions. The transformation is a rotation around the rotation axis $\mathbf{u} = (1, 1, 0)^T$ with an angle of $\theta = \frac{\pi}{2}$. The point $\mathbf{p}$ is expressed in euclidean coordinates as $\mathbf{p} = (5, 0, 0)^T$. Follow the steps below to find p', the rotated point corresponding to $\mathbf{p}' = T(\mathbf{p})$.

i) Describe $\mathbf{p}$ in quaternion form. **1 pt.**

ii) Write down the general formula to compute a rotation quaternion $\mathbf{q}_{gen}$ from a rotation axis $\mathbf{u}_{gen}$ and an angle θ_{gen}. Make sure the formula is correct even if $\mathbf{u}_{gen}$ is not a unitary vector. **1 pt.**

iii) Describe $\mathbf{q}$, a rotation quaternion representing the transformation T. Make sure to normalize $\mathbf{u}$ before anything else. **1 pt.**

iv) Write down the general formula to compute $\mathbf{p}'_{gen}$, a transformed point, result of applying a quaternion $\mathbf{q}_{gen}$ to a point $\mathbf{p}_{gen}$. You can assume that $\mathbf{q}_{gen}$ is a unit quaternion. **1 pt.**

v) Now apply the formula that you just wrote in the previous question to compute the rotated point $\mathbf{p}'$. Part of the operation was already done for you and you can use the following quaternion product equality $\mathbf{q} \cdot \mathbf{p} = \frac{1}{2}(-5 + 5\sqrt{2}i - 5k)$ without the need to derive it. **4 pts.**

d) **Rotations++**

i) You and your friend during a Visual Computing class break are casually talking about rotations. Your friend tells you that the matrix representation for rotations is not really compact. You of course already know that, and you reply that is outrageous to use 9 parameters (the ones of a 3 by 3 matrix), when the minimal number of parameters to uniquely describe a rotation is:

☐ 2

☐ 3 **1 pt.**

ii) You both then notice that quaternions need 4 parameters (a scalar s plus a 3-dimensional vector $(x, y, z)^T$) to be uniquely described. This implies that not all quaternions are rotation quaternions. Which property/properties must a quaternion have in order to be a rotation quaternion?
Hint: We are not asking the formula that was asked in question 9.c.ii, but intrinsic property/properties that quaternion can have in general. **1 pt.**

iii) You and your friend then decide to find a way to use only the minimal number of parameters to describe a rotation and describe the other/others as a function of them. To achieve that, if you think that the minimal number parameters for a rotation is 2, describe two functions $s = f_1(y, z)$ and $x = f_2(y, z)$. If instead you believe that the minimal number of parameters for a rotation is 3, describe a single function $s = f(x, y, z)$. **1 pt.**

Question 10: Physics Simulation (12 pts.)

a) **Rigid-Body Physics**

i) Whenever you want to perform a rigid-body simulation, you need to store and keep track of every rigid-body state. Write down the four quantities that describe a rigid body state at any point in time t. Then write also the time-derivatives of those quantities. At this point you should simply list their names and you do NOT need to explain how to compute them. **4 pts.**

ii) In simulation, whenever you want to advance in time, you need to integrate the state quantities over time. The easiest way to integrate over time is to use explicit Euler:

$$f(t + \Delta t) = f(t) + \Delta t \cdot \dot{f}(t) \tag{1}$$

Compute the four state quantities of a rigid-body at time $t + \Delta t$ by integrating over time using the formula above. Now you need to explain how to compute them in terms of: mass m, moment of inertia $I(t)$, force $F(t)$ and torque $\tau(t)$. This are the only quantities that can be used in order to express the state.

4 pts.

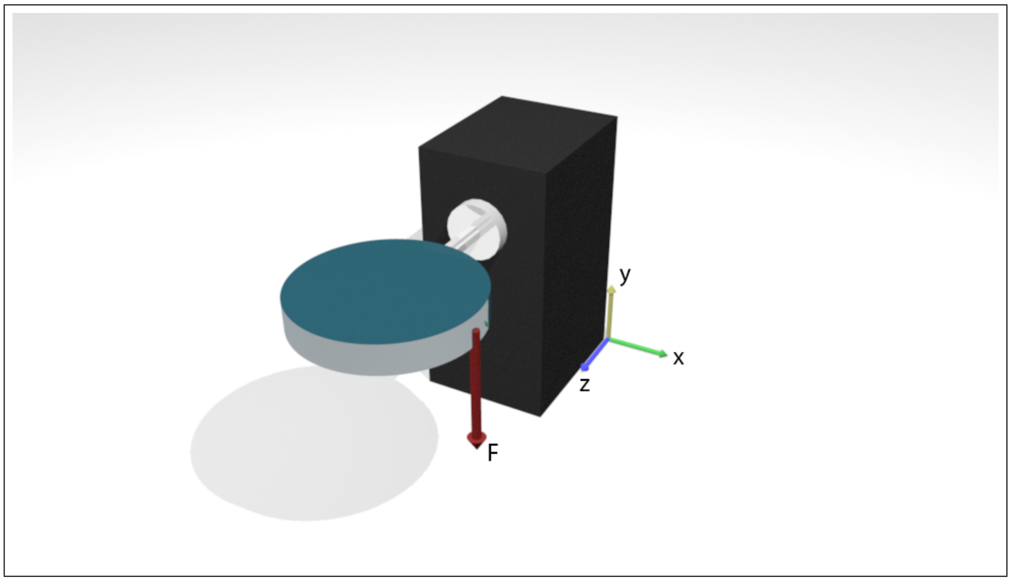


Figure 6: Scene that shows a motor (*black*), its actuated shaft (*shiny light-grey*) and an attached disk (*shiny light-blue*). The *red* arrow indicates the applied force F.

b) **Rigid-Body Physics++**

Figure 6 shows a motor with its shaft and a disk attached to it. For this exercise you assume that everything is rigid (i.e. no deformation involved), that the motor is perfectly attached to the (fixed) ground and the disk is perfectly attached to shaft. Moreover the shaft can only rotate with respect to the motor, around his rotation axis $\mathbf{u} = (0, 0, 1)^T$. Long story short: the disk will only be able to rotate and not translate.
The disk has a mass $m = 0.1kg$, a radius of $r = 3cm$ and its center of mass CoM is placed at $CoM = (-0.025, 0.075, 0.011)$. We then apply a force $F = (0, -1, 0)^T$ to the disk at the global point location $r = (0.005, 0.075, 0.011)$. You can ignore gravity due to symmetry.

i) Compute the torque τ_F that the force F induces to the disk as 3-dimensional vector. Hint: the torque will be aligned to the rotation axis $\mathbf{u}$ but with a different (possibly negative) magnitude. Below the formula of the cross product.

$$\begin{pmatrix} \mathbf{u}_x \\ \mathbf{u}_y \\ \mathbf{u}_z \end{pmatrix} \times \begin{pmatrix} \mathbf{v}_x \\ \mathbf{v}_y \\ \mathbf{v}_z \end{pmatrix} = \begin{pmatrix} \mathbf{u}_y\mathbf{v}_z - \mathbf{u}_z\mathbf{v}_y \\ \mathbf{u}_z\mathbf{v}_x - \mathbf{u}_x\mathbf{v}_z \\ \mathbf{u}_x\mathbf{v}_y - \mathbf{u}_y\mathbf{v}_x \end{pmatrix} \tag{2}$$

3 pts.

ii) What torque τ_{still} should the motor apply to the disk to withstand the torque generated by the force F, thus making the disk stay still? **1 pt.**