ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visual Computing

AS 2017

Prof. O. Hilliges
(Advanced Interactive Technologies Lab)
Prof. S. Coros (Computational Robotics Lab)

A. Cohen, J. Song, B. Hepp
V. Megaro, R. Roveri, J. Bern, L. Helminger
C. Öztireli

Final Exam

?? February 2018

First and Last name: ______________________________

ETH number: ______________________________

Signature: ______________________________

General Remarks

- At first, please check that your exam questionnaire is complete (there are 26 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 10 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	15		
2	Filtering	20		
3	Fourier Transform	15		
4	Optical Flow	25		
5	Miscellaneous	15		
6	OpenGL and Rendering	30		
7	Light, Color and Ray Tracing	24		
8	Transformations	22		
9	Rigid Bodies	12		
Total		180		

Grade: ....

Question 1: PCA (15 pts.)

a) Given a raw data matrix $\mathbf{D}$, what are the pre-processing steps you have to perform before applying Principal Component Analysis (PCA)? **2 pts.**

ANSWER

Center the data. **1 pts.**

Normalize the data. **1 pts.**

ANSWER

b) Given a properly pre-processed data matrix **D**. How do you compute the PCA? **2 pts.**

ANSWER

Compute the covariance matrix Σ of $\mathbf{D}$. **1 pts.**

Perform an Eigen Decomposition of Σ. **1 pts.**

The resulting eigen vectors give directions that describe the variation

in the data independent of each other.

The corresponding eigenvalue describes how large the variation along this direction is. **1 pts.**

Note: **0.5 pts** *if only eigenvector or only eigenvalue is mentioned.*

ANSWER

c) When computing a PCA, the resulting components are orthogonal to each other and the corresponding eigenvalues are real. Explain why this is always the case. **2 pts.**

ANSWER

The covariance matrix is symmetric and real by definition (for real data). **1 pts.**
The eigen decomposition of a symmetric real matrix has orthogonal eigenvectors and real eigenvalues. **1 pts.**

ANSWER

d) How can PCA be used to compress images of faces? Explain what computation has to be done with the training set of face images and how a single image can be compressed and decompressed. What are Eigenfaces? **6 pts.**

ANSWER

A PCA is performed on the training images (each image is converted to a row vector and the resulting data matrix is centered and normalized). **2 pts.**
An image can then be compressed by subtracting the mean image and projecting the centered image onto the N eigenvectors with the largest eigenvalues (N depends on the desired compression ratio).
1.5 pts.
To decompress an image the N coefficients can be multiplied with the corresponding eigenvectors. The mean image has to be added.
1.5 pts.
Note: **1.5 pts.** *only if mean of image is mentioned. Otherwise* **1 pts.**.
Eigenfaces are the components of the PCA on the face dataset. **1 pts.**

ANSWER

e) Given a set of images with 50×50 pixels in size, what is the size of the components from a PCA on this data? **1 pts.**

ANSWER

The components of a PCA have a size $50 \cdot 50 = 2500$. **1 pts.**

ANSWER

f) When compressing images of 50×50 pixels in size with PCA, is it possible to achieve a loss-less compression? **2 pts.**

ANSWER

To ensure a loss-less compression of arbitrary data all
$50 \times 50 = 2500$ *coefficients need to be kept.* **1 pts.**.
Thus the compressed size is equal to the non-compressed size.
Additionally the mean-vector needs to be stored and the Eigen-images
need to be known. Thus the "compressed" images would need
more storage than the original images. **1 pts.**.

ANSWER

Question 2: Filtering (20 pts.)

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

Alternatively, following the same idea, we can generate a low-pass filter kernel from a high-pass filter kernel.

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

-0.07	-0.12	-0.07
-0.12	0.76	-0.12
-0.07	-0.12	-0.07

ANSWER

Impulse kernel (image)		*High-pass kernel*	*Low-pass kernel*
$\| > m4ex\| > m4ex\| > m4ex\|$*height0*	*0*	*0*	
0	*1*	*0*	
0	*0*	*0*	

—

—

¿ m4ex

—

¿ m4ex

—

¿ m4ex

—

-0.07 -0.12 -0.07
-0.12 0.76 -0.12
-0.07 -0.12 -0.07

$=$

—

¿ m5ex

—

¿ m5ex

—

¿ *m5ex*

—

0.07	0.12	0.07
0.11	0.24	0.11
0.07	0.12	0.07

ANSWER

b) Both filter kernels shown below can be used for horizontal edge detection. Which of them is a better approximation of the gradient along the horizontal direction? Briefly explain your answer. **2 pts.**

$$G1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

$$G2 = \begin{bmatrix} -1/6 & 0 & 1/6 \\ -1/6 & 0 & 1/6 \\ -1/6 & 0 & 1/6 \end{bmatrix}.$$

ANSWER

The kernel filter $G2$ is the better approximation as it is properly normalized.

$G1$ gives an approximation with 6 times higher values.

ANSWER

c) Given below is a low-pass filter kernel L and a high-pass filter kernel H.

Compute the band-pass filter kernel resulting from applying the low-pass and the high-pass filter.

Note: Perform a convolution with zero-padding and

an output size of $3x3$.

Make use of symmetries to reduce the computation.

5 pts.

$$L = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

$$H = \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}.$$

ANSWER

Compute the convolution of L with H using "same" zero padding.

$$L \otimes H = \begin{bmatrix} -4 & 0 & -4 \\ 0 & 76 & 0 \\ -4 & 0 & -4 \end{bmatrix}.$$

Give 2 points if convolution was written down correctly (i.e. they realized

to convolve L and H).

Give 3 points if final matrix size correct but results wrong.

Give 4 points if only one number is wrong.

ANSWER

d) Write down the definition of correlation and convolution operations

in two dimensions.

Explain their difference.

Can a convolution always be represented by a correlation?

4 pts.

ANSWER

Correlation:

I'(x,y) $= \sum_{j=-k}^{k} \sum_{i=-k}^{k} G(i,j) I(x+i, y+i)$

1 pt.

Convolution:

I'(x,y) $= \sum_{j=-k}^{k} \sum_{i=-k}^{k} G(i,j) I(x-i, y-i)$

$= \sum_{j=-k}^{k} \sum_{i=-k}^{k} G(-i,-j) I(x+i, y+i)$

1 pt.

The difference is that a convolution is "running" through the image and the kernel in opposite directions.

1 pt.

Convolution and correlation can be replaced with each other by mirroring the filter kernel.

1 pt.

ANSWER

e) What is the minimum number of multiplications needed to compute the convolution between an image of size $N \times N$ pixels and the 3×3 filter (K) shown below? (You can neglect the border issues and justify your answer.) **3 pt.**

$$K = \begin{bmatrix} 0.08 & 0.12 & 0.08 \\ 0.12 & 0.20 & 0.12 \\ 0.08 & 0.12 & 0.08 \end{bmatrix}.$$

ANSWER

Since the filter is separable, **1 pt** *it is* $6N^2$ **2 pt**.

($9N^2$ *or* $9(N-2)^2$ *1pt)* ($6N^2$ *or* $6(N-2)^2$ *3pt)* ($3N^2$ *or* $3(N-2)^2$*(buffer solution) 3pt)*

ANSWER

f) Areas with high gradient correspond to image edges. Give a sample of filter kernel(name) for edge detection. If the image contains high frequency noise the filter kernels used to compute the gradient could falsely detect the noise as image edges. With which filter could we prefilter the image to overcome this. How can we apply this extra filter without the need to filter the whole image with an additional filter **3 pt**

ANSWER

Possible filter kernels Prewitt, Sobel... **1 pt**. *Gauss filter the image.* **1 pt**. *Convolution is commutative so we can compute a derivative of Gaussian filter.* **1 pt**.

ANSWER

Question 3: Fourier transform (15 pts.)

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

$$F(g)(u) = \int_{\mathbb{R}} g(x)\, e^{-j2\pi ux} dx \tag{1}$$

The inverse fourier transform is defined as

$$g(x) = \int_{\mathbb{R}} e^{j2\pi ux} F(g)(u)\, du \tag{2}$$

The fourier transform of the dirac delta function is given as

$$F(\delta(x - x_0))(u) = e^{-j2\pi ux_0} \tag{3}$$

Euler's formula states

$$e^{jx} = \cos(x) + j\sin(x) \tag{4}$$

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

$$F(g)(u, v) = \int\int_{\mathbb{R}^2} g(x, y)\, e^{-j2\pi(ux+vy)} dxdy \tag{5}$$

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:

$$F(g)(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y)\, e^{-j2\pi\left(\frac{ux+vy}{N}\right)} \tag{6}$$

i) Derive the Fourier Transform of the infinite continuous signal $f_1(x) = \cos(2\pi k_0 x)$ using Equation 1(Your solution must not contain any integrals). **4 pts.**

ii) Given a function $f_2(x)$ whose Fourier Transform is F_2 and a function $f_3(x)$ which is the product of $f_1(x)$ and $f_2(x)$, compute the Fourier Transform of $f_3(x)$ in terms of F_2. **2 pt.**

ANSWER

ANSWER

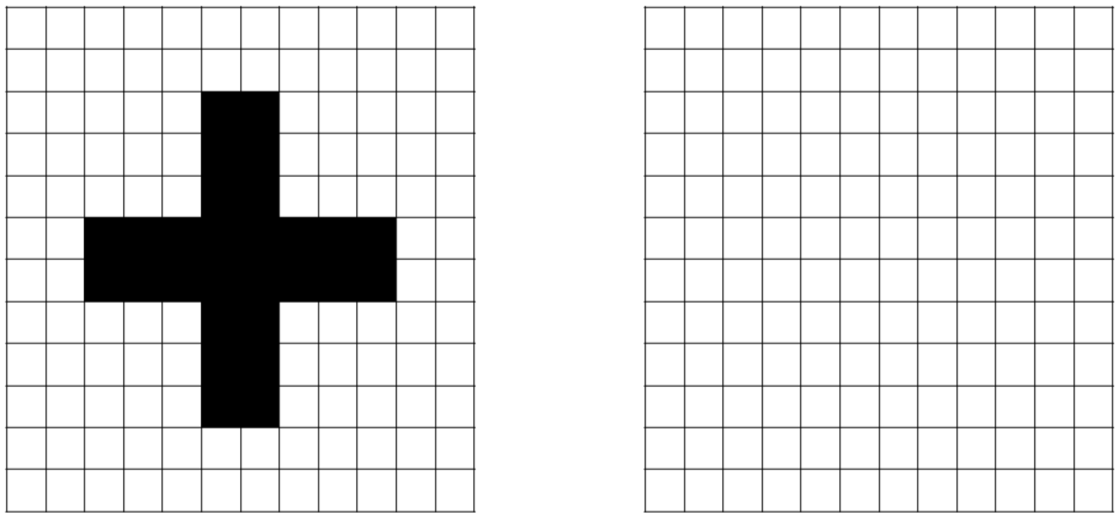
b) You are given a generic normalized low pass filter kernel. Which filter would be obtained by subtracting 1 from its central element? Explain your answer. **4 pts.** ANSWER

Let f be a generic low pass filter and a be a generic image, the result of convolving the new filter $(f - \delta)$ with a is:

$$(f - \delta) * a = f * a - \delta * a = -(a - f * a)$$

To derive this expression, we need to use the linearity properties of the convolution and the fact that the convolution with a delta does not alter the original signal. In the end, it results in a high pass filter which inverts the sign of the image (i.e., its phase is delayed by 180 degrees). (high pass filter 1pt) (incomplete 2pts) (some errors 3pts) (inverted high pass filter + procedure 4pts)

ANSWER



(a) Input Image I (b) Output Image I_1

Figure 1: Convolution with kernel $[-1, 0, 1]$

c) Refer the image I shown in Figure 1a. The image is 16×16 pixels in size and contains a plus shaped object. The background(white) and foreground(black) pixels have values 1 and 0 respectively. Let K be the kernel $[-1, 0, 1]$.

   i) Show the image I_1 obtained by computing the convolution between the image I and the kernel K in Figure 1b. (Ignore the border issues) **4 pts.**

   ii) Let I_2 be the image obtained by the corelation of image I and the kernel K. Compute I_2 in terms of I_1. **1 pt.**

ANSWER

*-1*Figure* **??**

2. $I_2 = -1 * I_1$ *1pt*

ANSWER

1. /include/OutputGrid.PNG

Figure 2: Solution

Question 4: Optical flow (25 pts.)

a) The Lucas-Kanade algorithm can be used to estimate the optical flow of an image sequence. Nevertheless, it doesn't work for all cases. State the 3 conditions that have to be fulfilled so that this method works **3 pts.** ANSWER ANSWER ANSWER ANSWER

1. *Brightness constancy: the intensity of the objects in the scene do not change in time*
2. *Small motion: objects move very slowly from frame to frame, which means that corresponding points from 2 consecutive images are not far apart*
3. *Spatial coherence: all points in a neighborhood have the same motion*

ANSWER

b) 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, i.e. the brightness constancy constraint becomes:

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

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 Lucas-Kanade method. What is the minimum size of a window around each pixel that allows one to solve the problem? **10 pts.**

ANSWER

First apply a first order Taylor expansion:

$$I(x, y, t) + I_x u + I_y v + I_t = aI + b$$

$$I_x u + I_y v + I_t = (a - 1)I + b$$

$$I_x u + I_y v + (1 - a)I - b = -I_t$$

$$\begin{bmatrix} I_x & I_y & I & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 - a \\ -b \end{bmatrix} = -I_t$$

From this equation, we can solve for the parameters (u, v, a, b) in a least squares sense by assuming that such parameters are constant on a neighborhood θ around each pixel. This leads to the following linear system of equations:

$$\sum \begin{bmatrix} I_x^2 & I_x I_y & I_x I & I_x \\ I_x I_y & I_y^2 & I_y I & Iy \\ I_x I & I_y I & I^2 & I \\ I_x & I_y & I & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 - a \\ b \end{bmatrix} = \sum \begin{bmatrix} I_t I_x \\ I_t I_y \\ I_t I \\ I_t \end{bmatrix}$$

Since there are 4 unknowns, we need at least 4 pixels, i.e. a 2x2 window.

ANSWER

c) You have captured a video at 25 frames per second of a car moving at 18 kilometers per hour. The side of the car is parallel to the image plane and the car is moving straight. The car is 2.4 meters long, but in your video it is 192 pixels long. Assume that your optical flow algorithm breaks down for pixel displacements that are larger than 1 pixel. By using image pyramids in a coarse-to-fine approach you can still use your algorithm. Explain this method, how you would proceed and how many pyramid levels you would need (the original image counts as one level and in each level the image size is half the size of the level before).

5 pts. ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

Explain how the coarse-to-fine approach works. 18 km/h equals 5 meters per second, which equals 20 cm per frame, i.e. 12 of the length of the car. 12 of 192 pixels is 16 pixels. Going from 16 to 8 to 4 to 2 to 1 leads to 5 levels.

(16 pixels: 1)

(5 levels: 1)

(Explanation of the coarse-to-fine approach: 2)

ANSWER

d) What is the aperture problem in computing optical flow? **1 pts.** ANSWER ANSWER

The aperture problem refers to the fact that when flow is computed for a point that lies along a linear feature, it is not possible to determine the exact location of the *corresponding point in the second image. Thus, it is only possible to determine the flow that is normal to the linear feature.*

ANSWER

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

- *Estimate velocity at each pixel using one iteration of Lucas and Kanade estimation*
- *Warp one image towards the other using the estimated flow field*
- *Refine estimate by repeating the process*

ANSWER

f) Explain how you would use optical flow for image stabilization. What other application can you think of for optical flow? **3 pts.**

ANSWER

Estimate the flow between the frames, and warp the image using the same flow over all the pixels so that the flow is close 0. (2 points) Name another application (e.g. video compression, slow motion, etc) (1 point)

ANSWER

Question 5: Miscellaneous (15 pts)

a) What are the three properties a feature detector should have in order to be robust? Is the Harris Corner detector robust to all three of them? **3 pts.**

1pt per item.

- *Invariant to brightness change*
- *Invariant to shift and rotation*
- *Invariant to scale (this is where Harris Corner detector fails)*

b) How would you exploit temporal redundancy for video compression? Describe the three types of frames used in this method. **4 pts.**

1pt: The idea is to predict the current frame based on previously coded frames: Temporal redundancy reduction 3pts: 3 types of frames:

- *I-frame: intra-coded frame, coded independently of all other frames.*
- *P-frame: coded frame, coded based on previously coded frame (based on previous I and P frames)*
- *B-frame: Bi-directionally predicted frame, coded based on both previous and future coded frames.*

c) When may temporal redundancy reduction be ineffective? **2 pts.**

When there are many scene changes (1pt) or/and high motion (1pt)

d) Which technique can be used when temporal redundancy reduction fails? Briefly describe the main 2 steps of the practical approach for this technique. **4 pts.**

The technique is motion-compensated prediction (1pt.) Practical approach: Block-matching Motion Estimation. Partition each frame into blocks and describe motion of each block by finding the best matching block in the reference frame.(3pts)

e) Name the two disadvantages of the block-matching algorithm. **2 pts.**

1pt per disadvantage. 1) It assumes translational motion model which breaks down for more complex motion. 2) often produces blocking artifacts.

Question 6: OpenGL and Rendering (30 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.** ANSWER ANSWER

- *blending not enabled*
- *objects drawn in wrong order.*

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

ii) Given the following CPP code:

```
glEnable(GL_BLEND);
glBlendEquation(GL_FUNC_ADD);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

An object A with color $c_A = (1.0, 0.0, 0.0, 1.0)$ was already rendered. A second object B with color $c_B = (0.0, 1.0, 0.0, 0.6)$ is now placed in front of A. What is the RGB value of the overlapping region. **2 pts.** ANSWER ANSWER ANSWER

final = (0.4, 0.6, 0.0)

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

iii) Why does z-fighting happen? **1 pts.** ANSWER ANSWER ANSWER ANSWER ANSWER

Because two objects have the same value in the depth buffer. / floating point unprecision

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

iv) Name 3 stages of the graphics pipeline and order them according to the step they occur. **1 pts.** ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

Vertex shader, rasterization, Fragment Shader

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

v) Name two implicit and two explicit representations in geometry. In addition, name two pros and two cons of implicit representations. **3 pts.** ANSWER ANSWER ANSWER

Implicit: algebraic surfaces, constructive solid geometry, level set methods, blobby surfaces, fractals

Explicit: triangle meshes, polygon meshes, subdivision surfaces, NURBS, point clouds

Implicit: Pros: description can be very compact (e.g., a polynomial), easy to determine if a point is inside/outside (just plug it in!), other queries may also be easy (e.g., distance to surface), for simple shapes, exact description/no sampling error, easy to handle changes in topology (e.g., fluid). Cons: expensive to find all points in the shape (e.g., for drawing) very difficult to model complex shapes

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.**

2pts for each.

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

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

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

Because if the modelviewMatrix contains shear/scale, the normal doesn't remain orthogonal.

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

c) **Fragment Shader**

i) Consider the following fragment shader. Complete the missing lines of code to implement the Phong reflection model, where I_a is the ambient color, I_p is the color for the diffuse and specular components, k_a, k_d and k_s are the material parameters for the ambient, diffuse and specular component, respectively, and α is the exponent of the specular component. The geometry vectors are shown in Figure 2. **8 pts.**

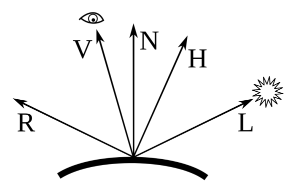


Figure 3: Geometry vectors for the reflection model.

```
#version 440

in vec3 outNormal;
in vec3 outPosition;

uniform vec3 eyePosition;
uniform vec3 lightDir;

out vec4 gl_FragColor;

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

        vec3 Ia = vec3(1.0, 0.0, 0.0);
        vec3 Ip = vec3(1.0, 1.0, 1.0);

        float ka = 0.2;
        float kd = 0.2;
        float ks = 0.6;
        float alpha = 2.0;


        // Ambient
        vec3 ambient = . . . . . . . . . . . . . . . . . . . . . . . . .


        // Diffuse
        vec3 diffuse = . . . . . . . . . . . . . . . . . . . . . . . . .


        // Specular
        vec3 R = . . . . . . . . . . . . . . . . . . . . . . . . .

        vec3 V = . . . . . . . . . . . . . . . . . . . . . . . . .

        R = normalize(R);
        V = normalize(V);

        vec3 specular = . . . . . . . . . . . . . . . . . . . . . . . . .


        vec3 color = . . . . . . . . . . . . . . . . . . . . . . . . .


        gl_FragColor = vec4(color, 1.0);
}
```

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

```
#version 440

in vec3 outNormal;
in vec3 outPosition;
in vec3 eyePosition;

out vec4 gl_FragColor;

void main()
{
```

```
    vec3 normal = normalize(outNormal);

    vec3 lightDir = vec3(1.0, 1.0, 1.0);
    vec3 Ia = vec3(1.0, 0.0, 0.0);
    vec3 Ip = vec3(1.0, 1.0, 1.0);

    float ka = 0.2;
    float kd = 0.2;
    float ks = 0.6;
    float alpha = 2.0;


    // Ambient
    vec3 ambient = Ia;


    // Diffuse
    vec3 diffuse = dot(lightDir, normal) * Ip;


    // Specular
    vec3 R = 2.0 * dot(lightDir, normal) * normal - lightDir;
    vec3 V = eyePosition-outPosition;


    R = normalize(R);
    V = normalize(V);


    vec3 specular = vec3(pow(max(0.0, dot(V, R)), alpha)) * Ip;

    //vec3 halfDir = normalize(lightDir + eyePosition);
    //float specAngle = max(dot(halfDir, normal), 0.0);
    //vec3 specular = vec3(pow(specAngle, alpha));


    vec3 color = kd*diffuse + ka*ambient + ks*specular;

    gl_FragColor = vec4(color, 1.0);
}
```

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

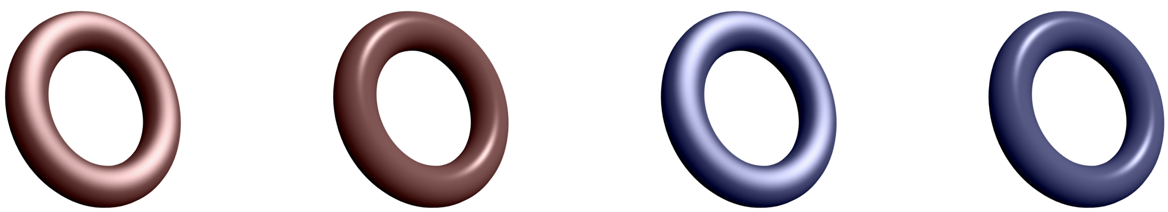
ii) Given are two possible values for α ($\alpha = 2.0$ and $\alpha = 20.0$), and two possible vectors I_a ($I_a = (0.0, 0.0, 1.0)$ and $I_a = (1.0, 0.0, 0.0)$). For each subfigure (a,b,c,d) in Figure 3 obtained with the above fragment shader, decide which values for α and I_a were used. **2 pts.**

a) $\alpha =$ ...... $I_a =$ ....

b) $\alpha =$ ...... $I_a =$ ....

c) $\alpha =$ ...... $I_a =$ ....

d) $\alpha =$ ...... $I_a =$ ....



(a) (b) (c) (d)

Figure 4: Different configurations of the phong shading model.

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

a) $\alpha = 2$ $I_a = (1.0, 0.0, 0.0)$
b) $\alpha = 20$ $I_a = (1.0, 0.0, 0.0)$
c) $\alpha = 2$ $I_a = (0.0, 0.0, 1.0)$
d) $\alpha = 20$ $I_a = (0.0, 0.0, 1.0)$

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

iii) The Blinn-Phong reflection model is a modification to the Phong reflection model. The Blinn-Phong model is the same as the Phong model, except for the specular component which is computed as $(H \cdot N)^{\alpha'}$, where H is the normalized halfway vector between light and viewing direction, as shown in Figure 2. In this exercise, we assume that $\alpha' = \alpha$. Complete the following lines of code to compute the specular component of the Blinn-Phong reflection model. **5 pts.**

```
vec3 H = . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
vec3 specular = . . . . . . . . . . . . . . . . . . . . . . . . .
```

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

```
vec3 halfDir = normalize(lightDir + eyePosition);
float specAngle = max(dot(halfDir, normal), 0.0);
vec3 specular = vec3(pow(specAngle, alpha));
```

ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

Question 7: Light, Color and Ray Tracing (24 pts.)

a) Why, in the YIQ color space, less bandwidth is required for Q than for I? Where is the YIQ color applied? **3 pts.**

ANSWER

The YIQ system is intended to take advantage of human color-response characteristics. The eye is more sensitive to changes in the orange-blue (I) range than in the purple-green range (Q). Therefore less bandwidth is required for Q than for I. YIQ is the color space used by the NTSC color TV system.

ANSWER

b) Explain a main difference between the RGB color space and the CMY color space. **2 pts.**

ANSWER

RGB is additive, while CMY is subtractive.

ANSWER

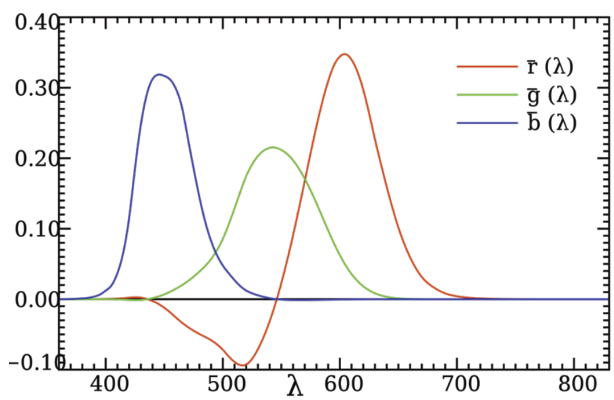


Figure 5: Color Matching Functions results from the CIE Experiment.

c) In 1931 CIE performed an experiment, where the observers had to trim the brightness of each of three primary beams in order to match a reference color. Figure 4 shows the resulting color matching functions, describing how the primaries were mixed. Why does one of them have some negative values? **2 pts.**

ANSWER

Not all test colors could be matched using the original experiment technique. When this was the case, a variable amount of one of the primaries could be added to the test color, and a match with the remaining two primaries was carried out with the variable color spot. For these cases, the amount of the primary added to the test color was considered to be a negative value. In this way, the entire range of human color perception could be covered.

ANSWER

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

	x	y	Y
c_1	0.3	0.2	20
c_2	0.5	0.25	10
c_3	0.35	0.5	10

Figure 5 is an empty CIE xy Chromaticity diagram.
Plot c1, c2 and c3 on the diagram. **2 pts.**

ANSWER

Simply draw the x and y coordinates of the three points in the CIE plot.

ANSWER

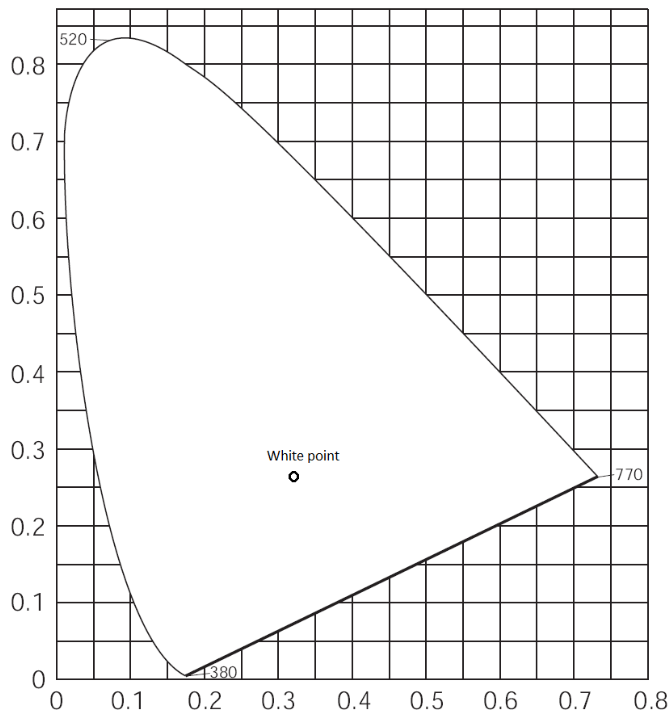


Figure 6: CIE xy Chromaticity diagram

e) Plot an additional primary c4 that can be created by mixing the three primaries c1, c2 and c3. In addition, plot another primary c5 that cannot be created by mixing the three primaries c1, c2 and c3. **2 pts.** ANSWER ANSWER ANSWER ANSWER ANSWER

c4 needs to be inside the triangle formed with c1, c2 and c3, while c5 needs to be outside of it.

ANSWER

f) Compute the sum of the three primaries c1, c2 and c3 in the XYZ color space. Plot the resulting color as c6 on the CIE xy Chromaticity diagram (Figure 5). **6 pts.**

ANSWER

XYZ to xyY :

$x = \frac{X}{X+Y+Z}$

$y = \frac{Y}{X+Y+Z}$

xyY to XYZ :

$X = x\frac{Y}{y}$

$Z = \frac{Y}{y} - x\frac{Y}{y} - Y$

From the previous equations we can compute X and Z for c_1, c_2, c_3, we get :

C_1 : $X_1 = 30$; $Z_1 = 50$

C_2 : $X_2 = 20$; $Z_2 = 10$

C_3 : $X_3 = 7$; $Z_3 = 3$

The addition in XYZ space gives us :

$X_{123} = X_1 + X_2 + X_3 = 57$

$Y_{123} = Y_1 + Y_2 + Y_3 = 40$

$Z_{123} = Z_1 + Z_2 + Z_3 = 63$

From the previous equations we convert these results to xyY and we get

$x = 0.356$

$y = 0.25$

$Y = 40$

ANSWER

g) Name four basic reflection functions, and for each of them describe, in a few words, how the incoming light energy from a given direction is reflected in various directions. **2 pts.**

ANSWER

Ideal specular (perfect mirror), ideal diffuse (uniform reflection in all directions), Glossy specular (majority of light distributed in reflection direction), retro-reflective (reflects light back towards source).

ANSWER

h) Briefly explain the differences between primitive partitioning and space partitioning for acceleration data structures and give an example for each. Which of the two (primitive partitioning or space partitioning) would you use if the objects in a scene move? **3 pts.**

ANSWER

Primitive partitioning (bounding volume hierarchy): partitions node's primitives into disjoint sets (but sets may overlap in space). Space-partitioning (grid, K-D tree) partitions space into disjoint regions (primitives may be contained in multiple regions of space). If the objects move, primitive partitioning is better.

ANSWER

i) What is an advantage and a disadvantage of an octree compared to a K-D tree? **2 pts.**

ANSWER

Easy to build (don not have to choose partition planes), but lower intersection performance than K-D tree (only limited ability to adapt) .

ANSWER

Question 8: Transformations (XX pts.)

a) **Transformations → Drawings**

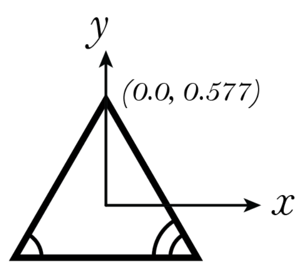
We will be working with the following homogeneous transformation matrices.

Note: This problem is in 2D, so a point $\begin{pmatrix} x \\ y \end{pmatrix}$ in Cartesian space would be written $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ in homogeneous coordinates.

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

$$\mathbf{R} = \begin{pmatrix} \cos 120° & -\sin 120° & 0 \\ \sin 120° & \cos 120° & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Consider the following equilateral triangle, with vertices labeled with arcs.



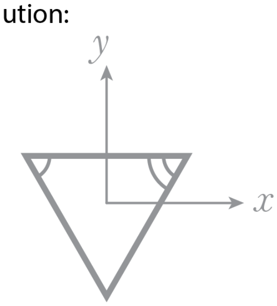
In each of the following questions, you will be given some transformation $\mathbf{A}$.

Please draw the result of applying $\mathbf{A}$ to the original triangle. Drawings do **not** need to be precise, just a rough sketch is perfect.

Note: To be clear, if $\mathbf{p}$ is some point on the triangle written in homogeneous coordinates, then the transformed point would be $\mathbf{Ap}$.

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

Example Solution:



i) $\mathbf{A} = \mathbf{T}$ **1 pts.**

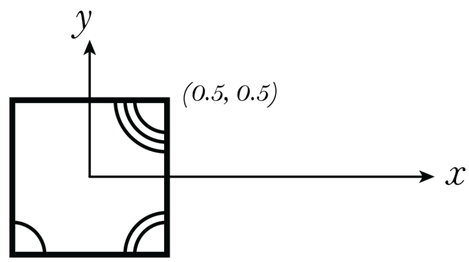
ii) $\mathbf{A} = \mathbf{R}$ **1 pts.**

iii) $\mathbf{A} = \mathbf{TR}$ **1 pts.**

iv) $\mathbf{A} = \mathbf{RT}$ **1 pts.**

b) **Drawings → Transformations**

Now consider the following square, again with vertices labeled with arcs.

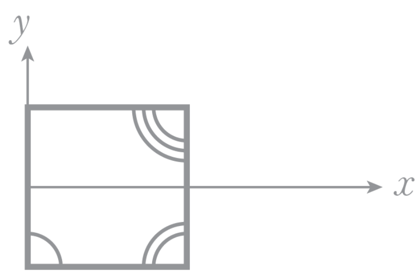


In each of the following questions, you will be given a picture. This picture is the result of applying some transformation $\mathbf{B}$ to the original square.

Note: To be clear, if $\mathbf{q}$ is some point on the square written in homogeneous coordinates, then the transformed point would be $\mathbf{Bq}$.

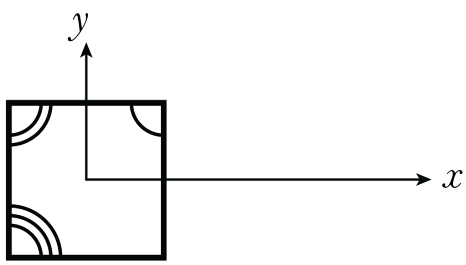
Please write down $\mathbf{B}$ for each picture. Your answers do not need to be in any particular form, the product of a few matrices is fine, as is a single matrix. As long as your answer would produce the given figure, it is perfect.

Example Problem:

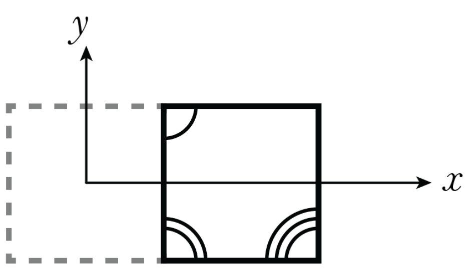


Example Solution: $\mathbf{B} = \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

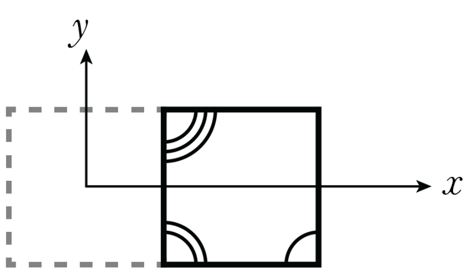
i) What is $\mathbf{B}$? **1 pts.**



ii) What is $\mathbf{B}$? **1 pts.**



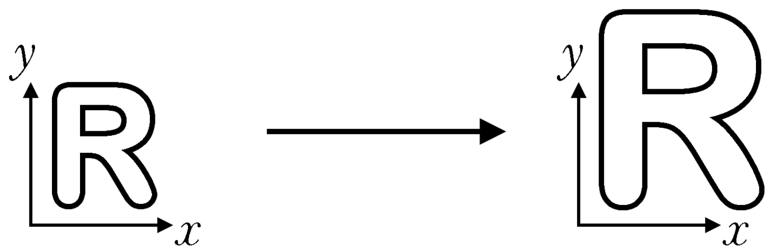
iii) What is $\mathbf{B}$? **1 pts.**



Question 9: Rigid Bodies (XX pts.)

a) **Rigid Body Transforms**

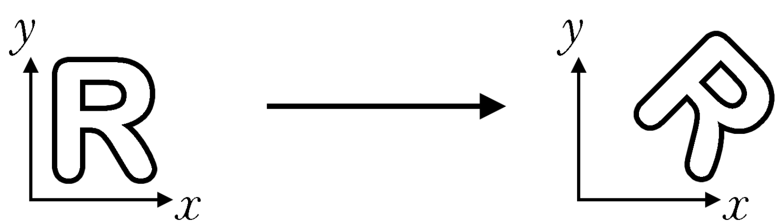
i) Is this a rigid body transformation?



☐ Yes

☐ No **1 pt.**

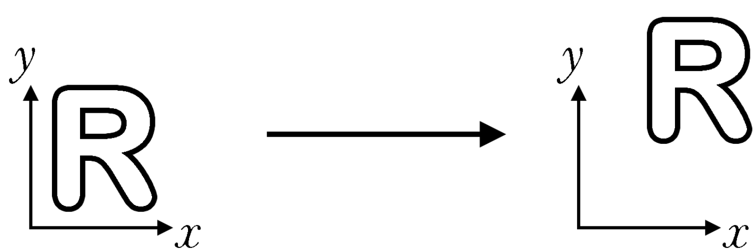
ii) Is this a rigid body transformation?



☐ Yes

☐ No **1 pt.**

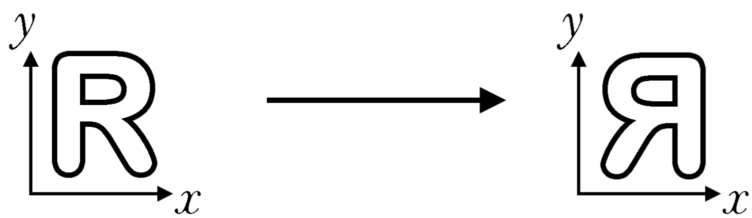
iii) Is this a rigid body transformation?



☐ Yes

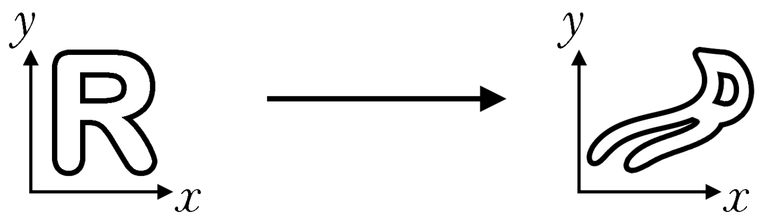
☐ No **1 pt.**

iv) Is this a rigid body transformation?



- ☐ Yes
- ☐ No **1 pt.**

v) Is this a rigid body transformation?



- ☐ Yes
- ☐ No **1 pt.**

b) **Bouncing Balls**

We consider a very simple collision example.

Consider two spheres.
They both have radius $1m$ (one meter).
The spheres are constrained to the x-axis.
They both have the same mass.
We neglect gravity, air resistance, etc.

Sphere A has initial center position $x_0^A = -0.5m$.
Sphere B has initial center position $x_0^B = 0.5m$.
Sphere A has initial velocity $\dot{x}_0^A = 1.0m/s$.
Sphere B has initial velocity $\dot{x}_0^B = -1.0m/s$

Notation: x_k^A means the position of Sphere A at timestep k.

i) Draw a 2D diagram of the initial state of the system. Include arrows for velocities. Don't forget to draw the origin and x-axis. **1 pts.**

ii) Given sphere positions x^A, x^B, write down a condition for whether they are currently colliding (intersecting). **1 pts.**

iii) Evolve the system forward in time one step using forward Euler (explicit Euler) with timestep $h = 1.0s$. What are x_1^A, x_1^B? **1 pts.**

$$\begin{pmatrix} x_{k+1}^A \\ x_{k+1}^B \end{pmatrix} = \begin{pmatrix} x_k^A \\ x_k^B \end{pmatrix} + h \begin{pmatrix} \dot{x}_k^A \\ \dot{x}_k^B \end{pmatrix}$$

iv) The spheres should now be colliding (if not please check your math). Apply the following update rule (simple inelastic collision for spheres of equal mass) with $\varepsilon = 0.25$. What are $\dot{x}_1^A, \dot{x}_1^B$? **1 pts.**

$$(\dot{x}_1^A, \dot{x}_1^B) = (\varepsilon \dot{x}_0^B, \varepsilon \dot{x}_0^A)$$

v) Draw a diagram of the system after this first time step. Include arrows for velocities. Don't forget to draw the origin and x-axis. **1 pts.**

vi) Follow the same steps to compute x_2^A, x_2^B. The spheres should still be colliding (if not please check your math). **1 pts.**

vii) What would happen if we naively applied the inelastic collision update rule **again**, to compute $\dot{x}_2^A, \dot{x}_2^B$? What does this approach mean for the behavior of the spheres in the long run? **1 pts.**