ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visual Computing

AS 2015

Prof. M. Gross (Computer Graphics Lab) — *A. Cohen, P. Liu, Y. Aksoy*
Prof. M. Pollefeys (Computer Vision Group) — *A. Milliez, V. Megaro, C. Öztireli*

Final Exam

10 February 2016

First and Last name: ______________________

ETH number: ______________________

Signature: ______________________

General Remarks

- At first, please check that your exam questionnaire is complete (there are 31 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	Video Compression	10		
2	Edge Detection	9		
3	Principal Component Analysis	10		
4	Optical Flow	20		
5	Filters and Image Features	13		
6	Morphological Operators and Texture	11		
7	Fourier Transform	17		
8	Light and Colors	24		
9	Transformations	22		
10	Animation and Physics	12		
11	Rendering and Geometry	32		
Total		180		

Grade: ....

Question 1: Video compression (10 pts)

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

ANSWER

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.*

ANSWER

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

ANSWER

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

ANSWER

c) 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.**

ANSWER

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)

ANSWER

Question 2: Edge Detection (9 pts)

a) Looking at the first and second derivative of an image, how can you tell that there might be an edge? How could you only select strong edges? **3 pts.**

ANSWER

Maximum of first derivative, zero crossing of second derivative. Only take edges with strong first derivative.

ANSWER

b) Using filtering, image gradients can be computed. Which image filters are needed? How often do you need to filter the image. Give the formulas to compute the gradient magnitude and the gradient orientation. **3 pts.**

ANSWER

The gradient is composed of the derivatives in x and y direction. Each of them can be computed using a respective filter kernel.

$$D_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \qquad D_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Therefore the image needs to be filtered twice. The magnitude and the orientation are computed as

$$|\nabla I| = \sqrt{(D_x * I)^2 + (D_y * I)^2}$$

$$\phi(\nabla I) = \arctan\left(\frac{D_y * I}{D_x * I}\right)$$

ANSWER

c) In the lecture the Canny edge detection algorithm has been presented in 5 steps. Give a list of the 5 steps.

3 pts.

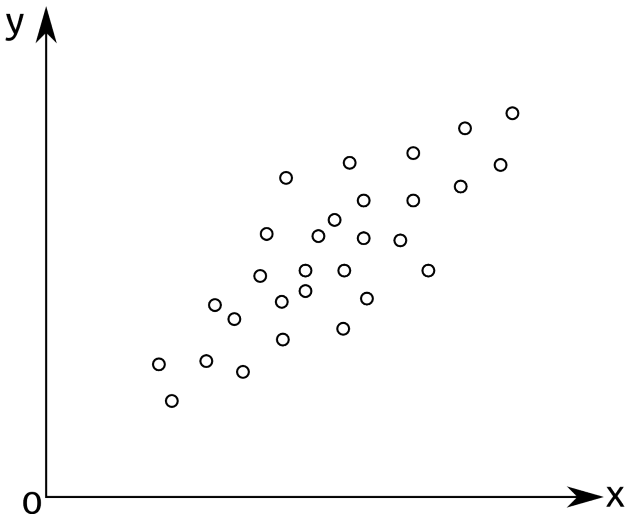
ANSWER

1. *Smooth image with a Gaussian filter*
2. *Compute the gradient magnitude and angle*
3. *Nonmaxima suppression*
4. *Double thresholding*
5. *Reject weak edge pixels not connected with strong edge pixels*

ANSWER

Question 3: Principal Component Analysis (10 pts.)

You are given the following dataset S with $n = 2$ dimensional inputs, meaning that each sample has 2 features: x and y. We run PCA on this dataset.



a) Draw the principal components you expect from PCA on the figure. Clearly indicate the first principal component. **2 pts.**

b) What does the first principal component represent in terms of the distribution of the samples? **1 pt.**

c) Shortly explain the steps of PCA. **5 pts.**

d) Explain how one can use PCA for data compression. **1 pt.**

e) Explain how one can use PCA for pattern recognition. **1 pt.**

Question 4: Optical flow (20 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

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) What is the aperture problem in computing optical flow? **1 pt.**

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

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

ANSWER

- *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

e) 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: Filters and Image Features (13 pts.)

a) The sum of entries of linear low-pass filter is usually equals to 1. What is the effect if the sum is greater than 1? For example, the filter **1 pt.**

$$G = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

ANSWER

If the sum is greater than 1, the overall intensity of the filtered image will be large. Thus, the resulted image will be brighter than usual.

ANSWER

b) Show when the convolution and correlation operations are dual with each other for image filtering in spatial domain. **3 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:

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

1 pt.

If $G(i,j) = G(-i,-j)$, then the correlation and convolution operations are dual with each other. **1 pt.**

ANSWER

c) Name the differences between gradient filter and laplacian filter. **2 pts.**

ANSWER

- *Gradient operator is isotropic, but laplacian operator is not.* **1 pt.**
- *Laplacian operator usually gives more noisy result since it uses 2nd-order derivative.* **1 pt.**

ANSWER

d) Briefly describe 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.**

ANSWER

Define the line equation as $y = mx + c$ or $x\cos(\theta) + y\sin(\theta) = \rho$, parameterize possible (m, b) or (ρ, θ) to satisfy all given (x, y) points. Find the maximum counts of (m, b) or (ρ, θ) pair. Then the line is fitted. **1 pt.**

The disadvantage of using $y = mx + c$ as the line equation is that when the line is (nearly) parallel to y axis, m is approaching to ∞, it is hard for parameterization. **1 pt.**

The solution is to use $x\cos(\theta) + y\sin(\theta) = \rho$ instead. **1 pt.**

ANSWER

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

ANSWER

$(\Delta x, \Delta y)$ *is constant size (blue boxes in the figure), and it can only detects corners that fit well into the window size (right). Corners which are larger than the window size are detected as edges instead (left)* **1 pt**.

Ellipses (eigenvectors) rotate but shapes (eigenvalues) remain the same, therefore Harris corners are rotation invariant. **1 pt**

ANSWER

f) Suggest a way to make Harris corners scale invariant. **2 pts.**

ANSWER

Look for strong responses of DoG filter (Difference of Gaussian) over scale space **1 pt.**, *and only consider local maxima in both position and scale space* **1 pt.**.

ANSWER

Question 6: Morphological operators and Texture (11 pts.)

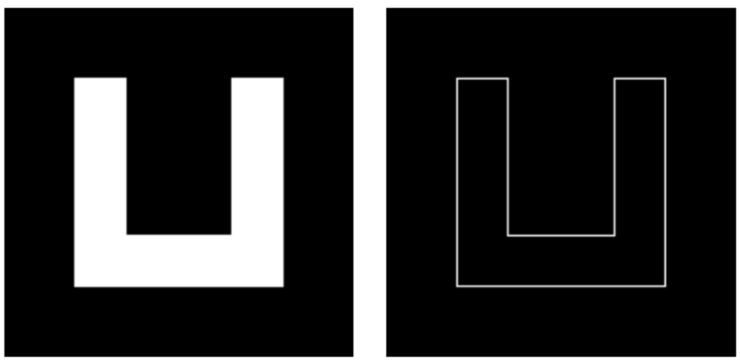
a) Define opening and closing of an image I by a structuring element S. Give the set formulation for any basic operators you may use. **3 pts.**

ANSWER

Erosion: $E = \{y : y = x + s, s \in S\} \subset I$ *(0.5 pts)*
Dilation: $D = \{y : x - s, s \in S\} \cap I \neq \emptyset$ *(0.5 pts)*
Opening $= (I \ominus S) \oplus S$ *(1 pt)*
Closing $= (I \oplus S) \ominus S$ *(1 pt)*

ANSWER

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.**



A B

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) $B = (A \ominus S)^c \cup A^c$

ANSWER

The answer is d

ANSWER

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

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

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

Build a Laplacian pyramid of the image of the texture and then apply a number of oriented filters to each level. This represents image information at a particular scale and orientation.

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

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

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

Because it's a lot cheaper to filter smaller images than big images with bigger filters. It's a question of efficiency.

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

d) 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.**

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

1. *tile input image (0.5 pts)*
2. *pick random block and place them at random locations (0.5 pts)*
3. *smooth edges (0.5 pts)*

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

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.**

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

This method works the best for very random looking textures (like vegetation or hay), as opposed to very structured textures (like a brick wall).

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

Question 7: Fourier transform (17 pts.)

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

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

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

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

Similarly the fourier transform of 2D discrete and finite signals (i.e. $f(x,y)$ can be a digital image of size $M \times N$) can be defined by using zero padding and summation instead of integration as

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{x}{M}u + \frac{y}{N}v\right)}, \tag{3}$$

a) Derive the Fourier transform of $f(x,y) = \sin(2\pi u_0 x + 2\pi v_0 y)$. **4 pts.**

ANSWER

$$\begin{aligned} f(x,y) &= sin(2\pi u_0 x + 2\pi v_0 y) \\ &= \frac{1}{2j}(e^{j2\pi(u_0x+v_0y)} - e^{-j2\pi(u_0x+v_0y)}) \end{aligned}$$

1 pt.

$$\begin{aligned} F(u,v) &= \int\int_{\mathbb{R}^2} f(x,y) e^{-j2\pi(ux+vy)} dxdy \\ &= \frac{1}{2j}\int\int_{\mathbb{R}^2} (e^{j2\pi(u_0x+v_0y)} - e^{-j2\pi(u_0x+v_0y)}) e^{-j2\pi(ux+vy)} dxdy \\ &= \frac{1}{2j}\int\int_{\mathbb{R}^2} (e^{-j2\pi[(u-u_0)x+(v-v_0)y]} - e^{-j2\pi[(u+u_0)x+(v+v_0)y]}) dxdy \textbf{(1 pt.)} \\ &= \frac{1}{2j}(\delta(u-u_0, v-v_0) - \delta(u+u_0, v+v_0)) \textbf{(2 pts.)} \end{aligned}$$

ANSWER

b) Given an image corrupted by this $2D$ sinusoidal signal, briefly state how you can remove it. **2 pts.**

ANSWER

Design a band-pass-filter at (u_0, v_0) *to remove this specific signal.* **2 pts.**

ANSWER

c) 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 **1 pt.***) (incomplete* **2 pts.***) (some errors* **3 pts.***) (inverted high pass filter + procedure* **4 pts.***)*

ANSWER

d) Nyquist theorem states that the sampling frequency must be at least twice the highest frequency of the signal

$$\omega_s > 2\omega_h.$$

Use Fourier transform to show why Niquist sampling theorem should hold. For simplicity, you can use a 1-D function (e.g., $f(t)$) to illustrate your answer instead of 2-D. **4 pts**

ANSWER

Define the continuous function as $f(t)$ and the sampling function $S_{\Delta t}(t)$ with sampling frequency as ω_s. Then the sampled function will be $f(t)S_{\Delta t}(t)$.

The Fourier transform of the sampled function can be derived as

$$\begin{aligned}\tilde{F}(u) &= \bar{F}\left(f(t)S_{\Delta t}(t)\right)\\ &= F(u)\star S_{\Delta t}(\omega)\\ &= \int_{-\infty}^{\infty} F(\tau)S_{\Delta t}(\omega-\tau)d\tau\\ &= \int_{-\infty}^{\infty} F(\tau)\frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}\delta(\omega-\tau-\frac{n}{\Delta T})d\tau\\ &= \frac{1}{\Delta T}\sum_{n=-\infty}^{\infty}F(\omega-n\omega_s)\end{aligned}$$

where $F(\omega)$ is the fourier transform of the original continuous singal $f(t)$. **2 pts.**

If we want to reconstruct the orignal signal $f(t)$ from the Fourier transform of the sampled function, i.e., $\tilde{F}(u)$, $F(\omega)$ cannot overlap with its neighbours, i.e., $F(\omega-\omega_s)$ and $F(\omega+\omega_s)$. Thus, ω_s should be larger than $2\omega_h$, where ω_h is the highest frequency of the original signal $f(t)$. **2 pts.**

ANSWER

e) What is the effect if the Nyquist theorem fails when sampling a signal? Propose a solution to solve this problem (assume the sampling frequency cannot be changed). **3 pts**

ANSWER

Aliasing effect will occur **1 pt.**. *The solution is to filter the signal with a low-pass filter to reduce its high frequencies to acceptable levels before it is sampled.* **2 pts.**

ANSWER

Question 8: Light and Color (24 pts.)

a) Color Spaces

i) Explain two main differences between the RGB and CMY color spaces.

2 pts.

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

The first main difference is that RGB is an additive color space, that is every color that can be represented in RGB is a non negative linear combination of the 3 primaries. CMY on the other hand is a subtractive color space, that is every color that can be represented in CMY is a non positive linear combination of the 3 primaries.

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

ii) Why were the HSL and HSV color systems invented?

2 pts.

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

To facilitate color picking.

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

b) CIE Experiment

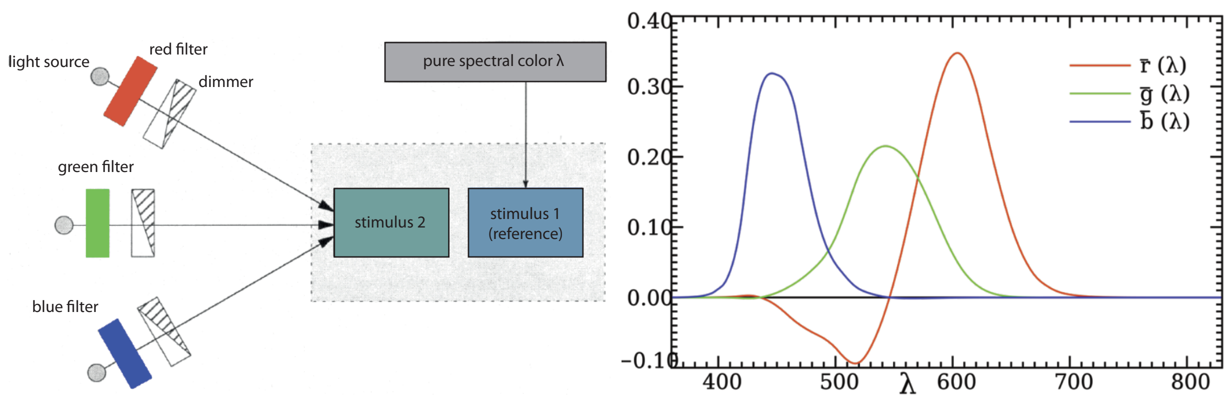


Figure 2: Left: CIE 1931 Experiment Setup. Right: Experiment Results.

i) In 1931 CIE performed an experiment. Figure 2 (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.

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

In 1931 CIE performed an experiment (Figure 2) where they would show to humans a reference (stimulus 1) illuminated by monochromatic light and the subjects task was *to illuminate another patch (stimulus 2) using 3 primaries such that the color in both patches would be identical. In their possession they had 3 light sources with R G and B filters that were pointing towards stimulus 2. Finally through the help of dimmers they were able to control the intensity of the 3 light sources and match the color of reference.*

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

ii) In Figure 2 (right) the results of the CIE experiment are shown. For a point (x, y) on one of the curve, please explain briefly what x and y represent.

2 pts.

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

The horizontal axis shows the wavelength of the monochromatic light that was on stimulus 1. The vertical axis shows for each one of the target wavelengths what was the intensity of the light used from each one of the RGB lights that subjects had to use in order to match the color on stimulus 2.

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

iii) In the range of 450-550 on the x-axis we observe that the red curve has negative values. Explain briefly how and why this is possible given the fact that negative light does not exist.

2 pts.

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

The human subjects were also allowed to illuminate stimulus 1 with red light. The reason is that the monochromatic light on stimulus 1 was outside of the gamut that the 3 primaries on stimulus 2 could reproduce.

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

c) **CIE Color Spaces**

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

	x	y	Y
c_1	0.1	0.8	12
c_2	0.2	0.05	10
c_3	0.7	0.25	26

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

2 pts.

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

xy axes control Chromaticity and Y controls Luminance (brightness).

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

ii) Figure 3 is an empty CIE xy Chromaticity diagram. Plot c_1, c_2 and c_3 on the diagram.

1 pt.

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

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

iii) Draw the isoline of constant saturation passing through c_1 on the diagram.

2 pts.

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

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

iv) What is the name of the connection between 770nm and 380nm? What do the points on that line represent?

2 pts.

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

The purple line. They represent colors that are not spectral.

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

v) Provide the transformation formulas from CIE XYZ to CIE xyY and from CIE xyY to CIE XYZ.

2 pts.

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

XYZ to xyY :

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

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

xyY to XYZ :

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

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

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

vi) 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 3).

3 pts.

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

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

C_1 : $X_1 = 1.5$; $Z_1 = 1.5$

C_2 : $X_2 = 40$; $Z_2 = 150$

C_3 : $X_3 = 72.8$; $Z_3 = 5.2$

The addition in XYZ space gives us :

$X_{123} = X_1 + X_2 + X_3 = 114.3$

$Y_{123} = Y_1 + Y_2 + Y_3 = 48$

$Z_{123} = Z_1 + Z_2 + Z_3 = 156.7$

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

$x = 0.36$

$y = 0.15$

$Y = 48$

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

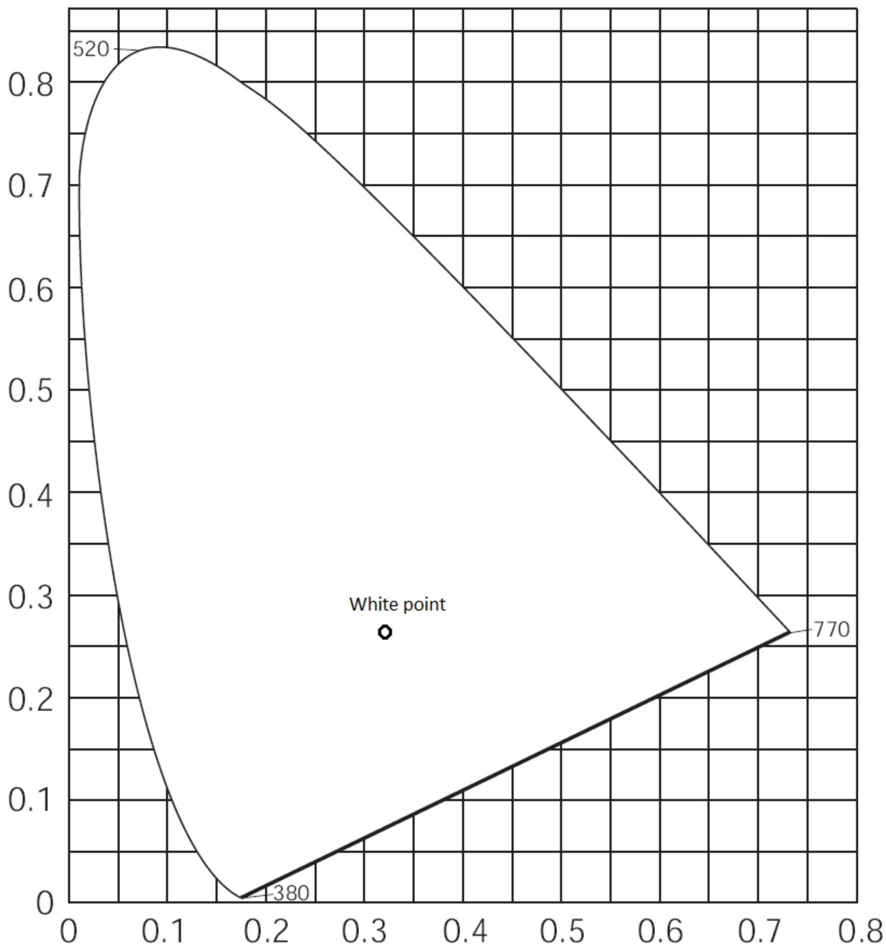


Figure 3: CIE xy Chromaticity diagram

Question 9: Transformations (22 pts.)

a) **Matrices**

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

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

3 pts. ANSWER ANSWER

- *A rotation by $\pi/4$ around $(0,1,0)^T$.*
- *First a rotation by $-\pi/2$ around $(0,0,1)^T$ then a translation by $(1,-1,1)^T$.*
- *A scaling by a factor 4.*

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

ii) Figure (4) shows a 3D scene before (4a) and after (4b) a linear transformation $\mathbf{M}$ is applied to the grey arrow-shaped object. The camera through which the scene in rendered is orthographic and placed at $(1,1,5)^T$. The looking direction of the camera is $(0,0,-1)^T$ while its up direction is $(0,1,0)^T$. By knowing that the object in the scene lies in the $x-y$ plane and does not leave it after the transformation is applied, describe $\mathbf{M}$ in words and find its homogeneous matrix representation. **3 pts.**

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

To go from figure 4a to 4b, first a rotation by $\pi/2$ around $(0,0,1)^T$ then a translation by $(2,2,0)^T$ must be applied. Its matrix representation is: $\begin{pmatrix} 0 & -1 & 0 & 2 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

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

b) **Matrices**

In linear algebra the nullspace of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is vector space defined as $\mathcal{N}(\mathbf{A}) = \{\mathbf{v} \in \mathbb{R}^{n\times 1} | \mathbf{A}\mathbf{v} = \mathbf{0}\}$. Let $\mathbf{R}$ be a 3-dimensional rotation matrix that rotates by $\pi/6$ around $(0,1,0)^T$.

i) Determine the intrinsic dimensionality of the nullspace of $\mathbf{R}$. Shortly explain your answer with two/three sentences. **2 pts.**

ANSWER

A rotation matrix $\mathbf{R}$ is a rigid transformation, thus it preserves a vector length once applied (1 pt). The intrinsic dimensionality of the nullspace of $\mathbf{R}$ is then 0, since the only vector $\mathbf{v}$ that satisfies $\mathbf{R}\mathbf{v} = \mathbf{0}$ is the null vector $\mathbf{0}$ (1 pt).

ANSWER

ii) Let $\mathbf{v_0}$ be a unit vector with positive y-component. Moreover let be $\mathbf{v_0} \in \mathcal{N}(\mathbf{R} - \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Determine $\mathbf{v_0}$ and explain your answer. **3 pts.**

ANSWER

If $\mathbf{v_0} \in \mathcal{N}(\mathbf{R} - \mathbf{I})$ *it means that:*

$$(\mathbf{R} - \mathbf{I})\mathbf{v_0} = \mathbf{0} \tag{4a}$$
$$\mathbf{R}\mathbf{v} - \mathbf{v_0} = \mathbf{0} \tag{4b}$$
$$\mathbf{R}\mathbf{v} = \mathbf{v_0} \tag{4c}$$

(1pt)
The only way to satisfy equation (4c) is when $\mathbf{v_0}$ *is aligned with the rotation axis of* $\mathbf{R}$, *which in this case is* $(0,1,0)^T$ *(1pt). By knowing that* $\mathbf{v_0}$ *is a unit vector and that has a positive y-component we uniquely determine* $\mathbf{v_0}$ *to be* $(0,1,0)^T$ *(1pt).*

ANSWER

c) **Quaternions**

i) Let $\mathbf{p}$ be $(1,1,0)^T$. Describe this point as a quaternion $\mathbf{q_p}$. **1 pt.**

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

$\mathbf{q_p} = i + j.$

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

ii) Let R_1 be the rotation by $120°$ around $(1,1,1)^T$. Describe R_1 as a unit quaternion $\mathbf{q_1}$ and simplify it as much as possible. You can use the following table for sines:

θ	$sin(\theta)$
$30°$	$\frac{1}{2}$
$60°$	$\frac{\sqrt{3}}{2}$
$120°$	$\frac{\sqrt{3}}{2}$

1 pt.

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

$\mathbf{q_1} = \cos(60°) + \sin(60°)\frac{1}{||(1,1,1)^T||}(i + j + k)$
$\mathbf{q_1} = \frac{1}{2} + \frac{\sqrt{3}}{2}\frac{1}{\sqrt{3}}(i + j + k)$
$\mathbf{q_1} = \frac{1}{2}(1 + i + j + k)$

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

iii) Show that a quaternion $\mathbf{q}$ describing a rotation θ around a vector $\mathbf{v}$ is a unit quaternion if $\mathbf{v}$ is a unit vector. **2 pts.**

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

In general a quaternion $\mathbf{q}$ describing a rotation θ around a normalized vector $\mathbf{v}$ has the form:

$\mathbf{q} = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})\frac{1}{||\mathbf{v}||}(v_x i + v_y j + v_z k)$

Therefore its squared norm is:

$||\mathbf{q}||^2 = cos^2(\frac{\theta}{2}) + sin^2(\frac{\theta}{2})\frac{1}{||\mathbf{v}||^2}(v_x^2 + v_y^2 + v_z^2)$

$||\mathbf{q}||^2 = cos^2(\frac{\theta}{2}) + sin^2(\frac{\theta}{2})\frac{1}{||\mathbf{v}||^2}(||\mathbf{v}||^2)$

$||\mathbf{q}||^2 = cos^2(\frac{\theta}{2}) + sin^2(\frac{\theta}{2})$

$||\mathbf{q}||^2 = 1$

Which implies $||\mathbf{q}|| = 1$

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

iv) We define $\mathbf{p}'$ as the transformations of $\mathbf{p}$ by R_1 (see above). Using quaternion operations, compute $\mathbf{p}'$ and give its euclidean coordinates. **4 pts.**

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

The quaternion representation of $\mathbf{p}'$ is:

$\mathbf{p}' = \mathbf{q_1 q_p q_1^{-1}}$.

First we compute $\mathbf{q_1 q_p}$:

$\mathbf{q_1 q_p} = \frac{1}{2}(1 + i + j + k)(i + j)$

$\mathbf{q_1 q_p} = \frac{1}{2}(i + j + ii + ij + ji + jj + ki + kj)$

$\mathbf{q_1 q_p} = \frac{1}{2}(i + j - 1 + k - k - 1 + j - i)$

$\mathbf{q_1 q_p} = \frac{1}{2}(-2 + 2j)$

$\mathbf{q_1 q_p} = -1 + j$

Now we compute $\mathbf{q_1 q_p q_1^{-1}}$:

$\mathbf{q_1 q_p q_1^{-1}} = \frac{1}{2}(-1 + j)(1 - i - j - k)$

$\mathbf{q_1 q_p q_1^{-1}} = \frac{1}{2}(-1 + i + j + k + j - ji - jj - jk)$

$\mathbf{q_1 q_p q_1^{-1}} = \frac{1}{2}(-1 + i + j + k + j + k + 1 - i)$

$\mathbf{q_1 q_p q_1^{-1}} = \frac{1}{2}(2j + 2k)$

$\mathbf{q_1 q_p q_1^{-1}} = j + k$

This implies that $\mathbf{p}'$ *is represented by* $(0,1,1)^T$ *in euclidean coordinates.*

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

v) Let $\mathbf{q}$ be a unit quaternion describing a rotation by θ around a vector $\mathbf{v}$. Then let $\bar{\mathbf{q}}$ be a quaternion such that $\bar{\mathbf{q}} = -\mathbf{q}$. Does $\bar{\mathbf{q}}$ describe a rotation? If yes what is the relation between $\mathbf{q}$ and $\bar{\mathbf{q}}$? Prove your answer and make sure that all the steps of your proof are clear. **3 pts.**

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

In general a quaternion $\mathbf{q}$ *describing a rotation* θ *around a vector* $\mathbf{v}$ *has the form:*
$\mathbf{q} = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})(v_x i + v_y j + v_z k)$

Instead $\bar{\mathbf{q}}$ *has the form:*
$\bar{\mathbf{q}} = -cos(\frac{\theta}{2}) - sin(\frac{\theta}{2})(v_x i + v_y j + v_z k)$

We bring $\bar{\mathbf{q}}$ *into a quaternion form that represents a rotation:*
$\bar{\mathbf{q}} = -cos(\frac{\theta}{2}) - sin(\frac{\theta}{2})(v_x i + v_y j + v_z k)$
$\bar{\mathbf{q}} = cos(\pi - \frac{\theta}{2}) - sin(\pi - \frac{\theta}{2})(v_x i + v_y j + v_z k)$
$\bar{\mathbf{q}} = cos(\pi + \frac{\theta}{2}) + sin(\pi + \frac{\theta}{2})(v_x i + v_y j + v_z k)$
$\bar{\mathbf{q}} = cos(\frac{2\pi+\theta}{2}) + sin(\frac{2\pi+\theta}{2})(v_x i + v_y j + v_z k)$

This shows that $\bar{\mathbf{q}}$ *represents a rotation by* $2\pi+\theta$ *around* $\mathbf{v}$*, which is the same rotation described by* $\mathbf{q}$ *due to the fact that rotations can be taken modulo* 2π*.*

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

Question 10: Physics Simulation (12 pts.)

a) **Rigid-Body Physics**

The kinematic state of a rigid body can be described by the the position $\mathbf{x}(t)$ of the center of mass, the linear momentum $\mathbf{P}(t)$, the rotation $\mathbf{q}(t)$ (quaternion), and the angular momentum $\mathbf{L}(t)$. Write down the *ordinary differential equations* (ODEs) that govern the dynamics of a rigid body with a single force $\mathbf{F}_i(t)$ applied to the point $\mathbf{r}_i(t)$. You can assume that the mass m and moment of inertia tensor $\mathbf{I}(t)$ of the rigid body are known at any time. If you use any other quantities than the ones mentioned here, you have to define them first in terms of previously defined quantities. **4 pts.**

ANSWER

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \frac{\mathbf{P}(t)}{m} \\
\dot{\mathbf{P}}(t) &= \mathbf{F}_i(t) \\
\dot{\mathbf{q}}(t) &= \frac{1}{2}(\mathbf{I}^{-1}(t)\mathbf{L}(t))\mathbf{q}(t) \\
\dot{\mathbf{L}}(t) &= (\mathbf{r}_i(t) - \mathbf{x}(t)) \times \mathbf{F}_i(t)
\end{aligned}$$

ANSWER

b) **2D Physics Simulation**

Figure 5 shows the state of a 2D scene at time $t = 0s$. The scene contains two rigid bodies: a circle placed at $\mathbf{x}(0) = (0, 5)^T$ and a ground floor that you can assume is infinite in width and its surface lies at $y = 0$. The circle has radius $0.1m$, mass of $2kg$, and it is thrown with an initial velocity of $\mathbf{v_0} = (\frac{1}{2}, 0)^T \frac{m}{s}$. The only force acting on the circle is gravity which accelerates it by (approximatively) $10\frac{m}{s^2}$ towards the ground. You can assume that the ground is stationary.

i) Compute one Symplectic Euler simulation step using a time step of $\delta t = \frac{1}{2}s$. Then give the position of the center of mass as well as the linear momentum of the circle at the end of it. Recall that the Symplectic Euler method first updates the linear momentum and then position of a rigid body, by exploiting the Taylor Expansion:

$$\mathbf{P}(t + \delta t) = \mathbf{P}(t) + \delta t \dot{\mathbf{P}}(t)$$
$$\mathbf{x}(t + \delta t) = \mathbf{x}(t) + \delta t \dot{\mathbf{x}}(t + \delta t)$$

6 pts.

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

The linear part of the kinematic state of the circle at $t = 0s$ is:

$$\mathbf{P}(0s) = (1, 0)^T \frac{mkg}{s}$$
$$\mathbf{x}(0s) = (0, 5)^T m$$

By performing the first simulation step we get:

$$\mathbf{P}(\frac{1}{2}s) = \mathbf{P}(0) + \frac{1}{2}s\dot{\mathbf{P}}(0s)$$
$$\mathbf{P}(\frac{1}{2}s) = \mathbf{P}(0) + \frac{1}{2}s(0, -20)^T N$$
$$\mathbf{P}(\frac{1}{2}s) = (1, 0)^T \frac{mkg}{s} - (0, 10)^T \frac{mkg}{s}$$
$$\mathbf{P}(\frac{1}{2}s) = (1, -10)^T \frac{mkg}{s}$$

and

$$\mathbf{x}(\frac{1}{2}s) = \mathbf{x}(0s) + \frac{1}{2}s\frac{P(\frac{1}{2}s)}{2kg}$$
$$\mathbf{x}(\frac{1}{2}s) = (0,5)^T m + \frac{1}{2}s(0.5,-5)^T\frac{m}{s}$$
$$\mathbf{x}(\frac{1}{2}s) = (0,5)^T m + (0.25,-2.5)^T m$$
$$\mathbf{x}(\frac{1}{2}s) = (0.25,2.5)^T m$$

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

ii) What event will happen if you would perform another simulation step keeping the same δt? Describe two ways to handle this event and prevent infeasible simulation results.

2 pts.

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

By performing another simulation step, the circle will hit the ground and a collision will occur.
There are two common ways to update the states of rigid bodies after collisions. Either you adjust the state of the bodies after collisions by updating forces that act on the bodies or by updating momenta of the involved bodies.

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

[b]0.4

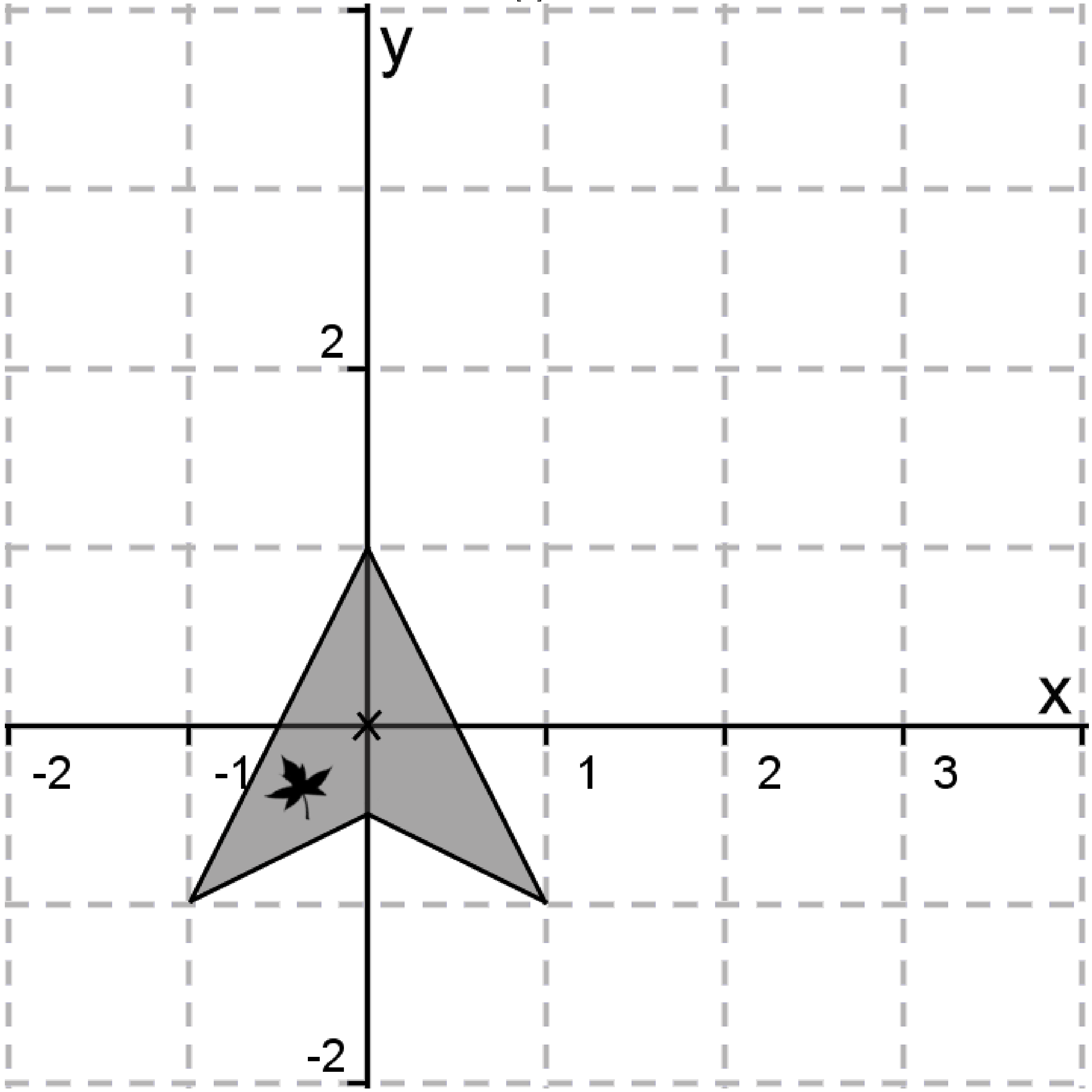
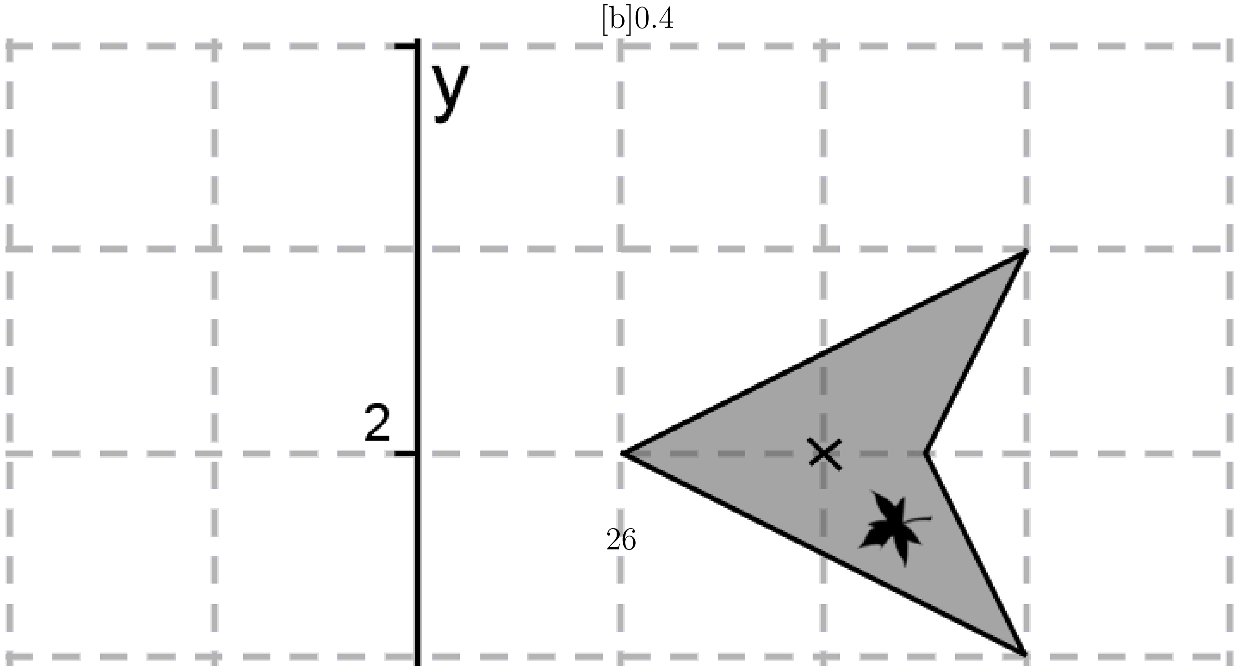


Figure 4: Scene before transformation **M** is applied.

[b]0.4



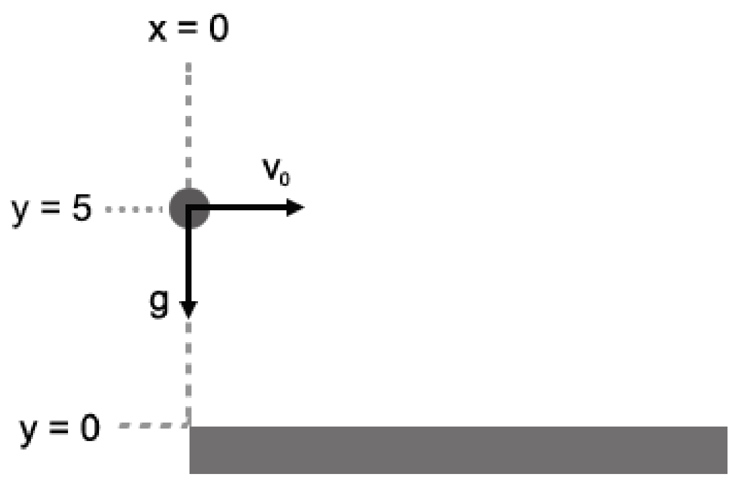


Figure 7: Scene at $t = 0s$. Note: is a representative image and proportions are not correct.

Question 11: OpenGL and Rendering (32 pts.)

a) **General**

i) In the lecture notes, 3D clipping is said to prevent divisions by zero. Explain why. **2 pts.** ANSWERANSWERANSWERANSWERANSWERANSWERANSWERANSWERANSWER

Save computation, memory consistency, stability (division by zero close to the camera).

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

ii) In the OpenGL rendering pipeline, in which order do the following steps happen? Fragment shader, vertex shader, rasterization. **2 pts.** ANSWER ANSWER ANSWER

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

iii) What is the difference between a normal map and a bump map? **2 pts.** ANSWER

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

b) **Goal**

The Chinese year of the monkey has just started, and you want to render this fortune teller scene, predicting the year to come.

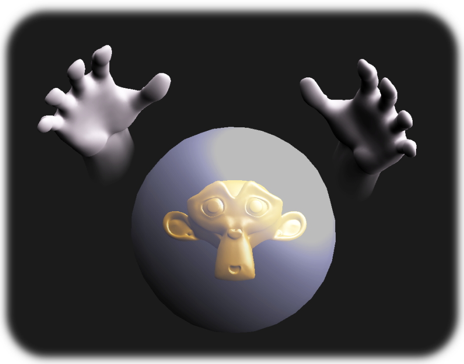


Figure 8: *"I see monkeys in your near future."*

You are going to build this scene step by step. Each subquestion (11c, 11d, 11e, 11f) can be treated independently.

c) **Hands**

You are provided with a file describing hand geometry as shown below.

Such a roughly cut hand is not mysterious enough. You want it to fade out to **transparent** depending on its distance from the camera, as such:

i) How can you implement this fade out using the vertex and fragment shaders? (Exact GLSL syntax is not required. Pseudo code or a clear and concise explanation in English are sufficient). **4 pts.**

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

After projection in the vertex shader, the z coordinate describes the depth in camera space.
Either pass it to the fragment shader, or pass the whole position, or compute the alpha there and pass it.
In the fragment shader, adapt the output's alpha.
2 points to understand that we don't work in world coordinates.
2 points to understand the vertex / fragment shader communication.

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

d) **Monkey**

Provided a file that describes the monkey face geometry, we want to produce the rendering seen on the left. It is a composition of the three components on the right.

i) What is the name of this lighting model? **1 pt.**

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

Phong reflection model (Phong Shading also accepted).

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

ii) What are the names of each of these components? Identify them in the above figure. **3 pts.**

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

1) Diffuse, 2) Specular, 3) Ambient.
0.5pt for each correct name. 0.5pt for each correct correspondence.

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

iii) The vertex shader for the monkey currently looks like this:

```
1  #version 440
2
3  uniform mat4 projection;
4  uniform mat4 modelview;
5
6  in vec3 inPosition;
7  in vec3 inNormal;
8
9  out vec3 outPosition;
10 out vec3 outNormal;
11
12 void main()
13 {
14   outPosition = inPosition;
15   outNormal = inNormal;
16
17   gl_Position = projection * modelview * vec4(inPosition, 1.0f);
18 }
```

What do projection and modelview represent? **2 pts.**

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

Projection transforms points from world space to screen space.
Modelview transforms points from object space to world space.

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

iv) For a more magical effect, you want the monkey to constantly spin around its y axis.
What input do you need to add to the vertex shader?
How will you change the body of the main() function?
(Exact GLSL syntax is not required, only the proper list of operations). **3 pts.**

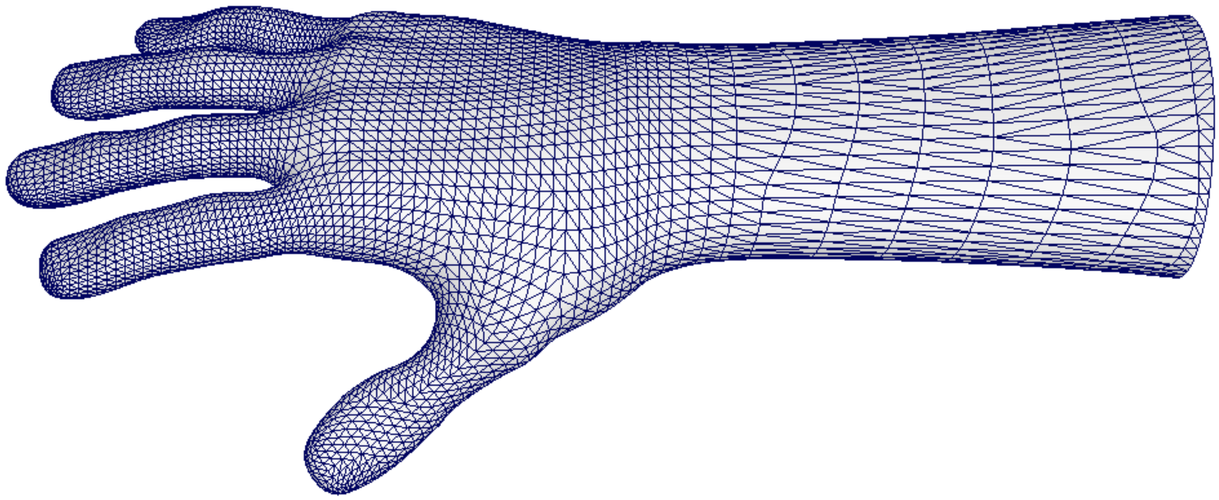
ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER ANSWER

Version 1 (lazy): input a rotation matrix depending on the time. Multiply inPosition by it before line 17.

Version 2 (hardcore): input the elapsed time. Build matrix and multiply inPosition by it.
Both versions accepted.

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

[b]0.4



[b]0.24

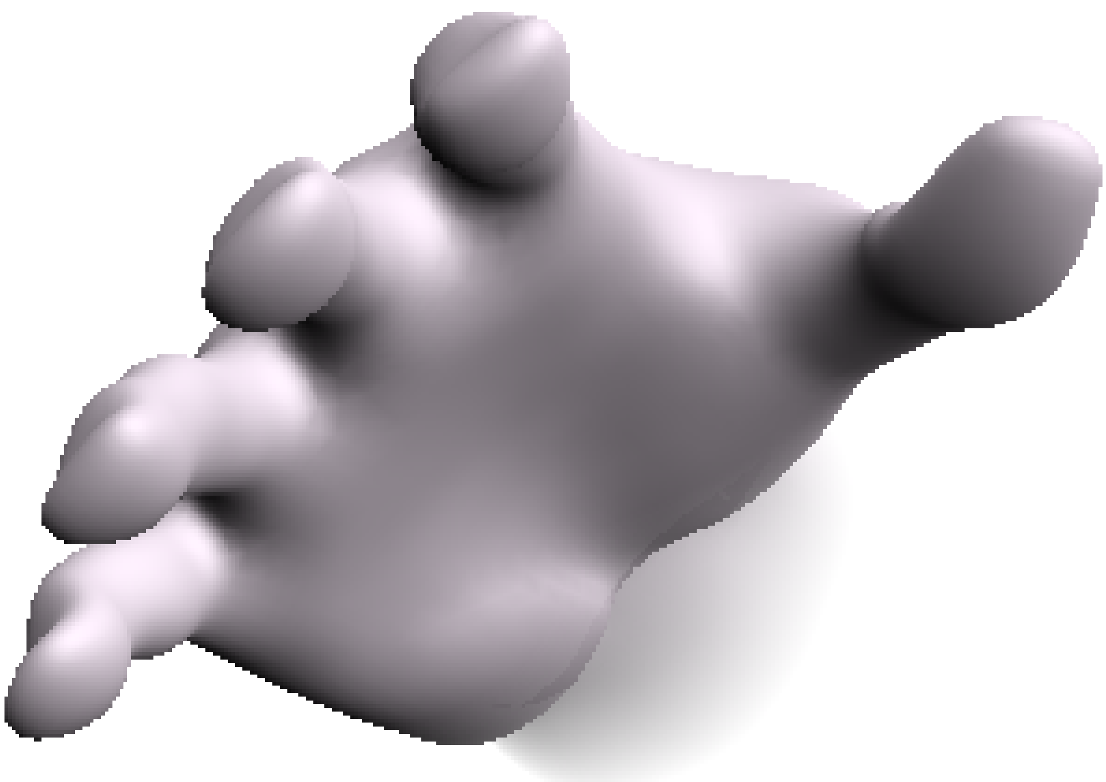


Figure 9: Hand geometry (left) and desired rendering (right).

[b]0.24

[b]0.24

e) **Crystal Ball**

i) The crystal ball is represented by a triangle mesh. Here are two renderings of that mesh. Can you explain the difference between both?
Using knowledge from the lectures and exercise sessions, what can you guess regarding the shader implementation of both rendering algorithms? **4 pts.**

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

2 points to explain per-color vs per fragment color. 2 points to understand the shader communication

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

ii) The monkey face geometry is placed inside the sphere mesh. In the fragment shader for that sphere, you set each fragment's alpha value to be below 1, expecting to see the monkey through the crystal ball. However when you run your application, no monkey is visible. When changing the order in which objects are rendered, the monkey is displayed. Using knowledge from the lectures and exercise sessions, explain what happened. **3 pts.**

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

Drawing the sphere first fills the depth buffer with values smaller than the monkey's depth. OpenGL discards the monkey fragments.

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

f) **Screen Space Effects**

We want to improve the scene using screen space effects, for example by fading the rendered color depending on its position on the screen.
This is done using a two-pass rendering: first we render the scene into a texture, using the shaders and algorithms described in the previous questions. We thus obtain a texture which is the same size as the OpenGL window and contains the rendered scene.

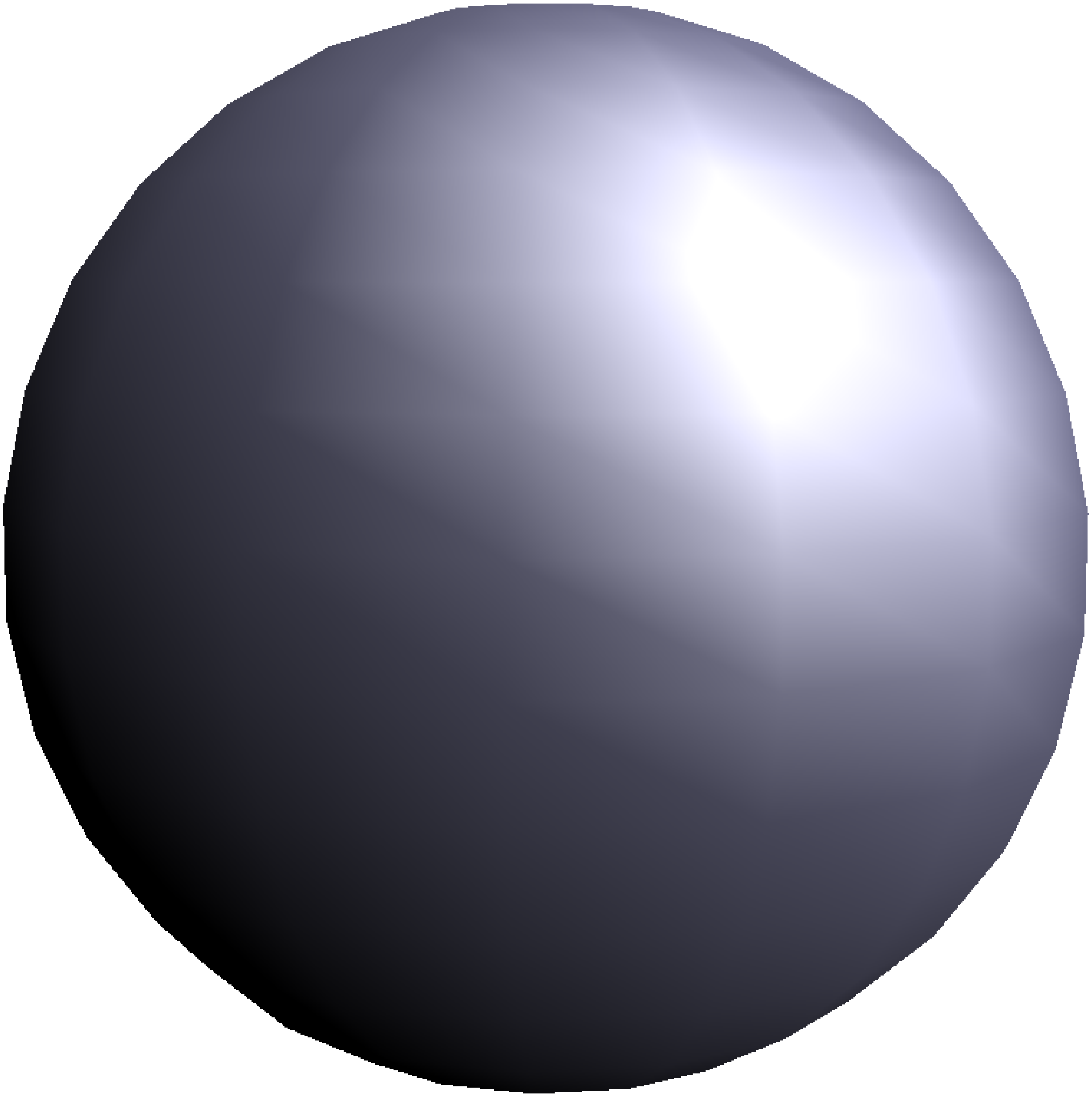
i) In order to display that texture on the screen and implement effects altering the color of each pixel depending on its position on the screen, explain what type of geometry you would use, and what you would write in your vertex and fragment shaders (in concise English). **6 pts.**

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

Display a quad that takes the whole screen. Vertex shader just passes stuff down the line, fragment shader reads the texture as input and does its thing.

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

[b]0.24



[b]0.24