ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

**Visual Computing**
AS 2010

*Prof. M. Gross (Computer Graphics Lab)*
*Prof. M. Pollefeys (Computer Vision Group)*
*G.-M. Baschera, P. Kaufmann, C. Kuster, A. Cohen, G. H. Lee, P. Singh, A. Taneja*

# Final Exam

4 February 2011

First and Last name: ______________________________

ETH number: ______________________________

Signature: ______________________________

# General Remarks

- At first, please check that your exam questionnaire is complete (there are ?? 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 8 questions, where you can earn a total of 180 points. You don't have to score all points to obtain a very good grade.
- Start each question on a separate sheet. Put your name and ETH number on top of each sheet. Only write on the question sheet where explicitly stated.
- You can answer questions in English or in German. Do not use a pencil or red color pen.
- You may provide at most one valid answer per question. Invalid solutions must be canceled out clearly.

| | Topic | Max. Points | Points Achieved | Visum |
|---|---|---|---|---|
| 1 | Filters and Image Features | 19 | | |
| 2 | Fourier Transform and Compression | 17 | | |
| 3 | Morphological Operators and Performance | 9 | | |
| 4 | Unitary Transforms and PCA | 22 | | |
| 5 | Optical Flow | 23 | | |
| 6 | Graphics Pipeline and Shaders | 30 | | |
| 7 | Colors and Reflection Models | 30 | | |
| 8 | Transformations and Projections | 30 | | |
| Total | | 180 | | |

Grade: ..............................................................

## Question 1: Filters and Image Features (19 pts.)

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

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

b) 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 $I(x, y)$. **2 pts.**

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

d) Harris corners are selected from analyzing the eigenvalues $\lambda_1$ and $\lambda_2$ 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) $\lambda_1$ and $\lambda_2$ are both large. **2 pts.**

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

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

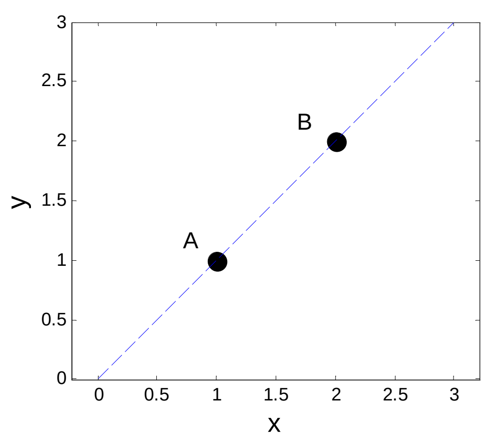



Figure 1: Two points on a line

g) Figure ?? shows two points on a line which could be detected using the Hough transform. Write the general parameterization which avoids the infinite-slope problem that represents this line. **1 pt.**

h) Compute and plot the Hough transform accumulator array for these two points. **4 pts.**

i) *With reference* to the accumulator array plot, deduce the Cartesian equation of the line shown in Figure ??. **1 pt.**

j) Explain why in old cowboy films that show wagon moving, the wheel often seems to be
   i) stationary or **1 pt.**
   ii) moving in the wrong direction (i.e. the wagon moves from left to right and the wheel seems to be turning counterclockwise). **1 pt.**

## Question 2: Fourier Transform and Compression (17 pts.)

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

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

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

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

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:

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

i) Derive the Fourier transform of the infinite continuous signal $f_1(x) = \cos(2\pi k_0 x)$ using Equation ??. **3 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 $F_2$. **1 pt.**

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.**

c) Refer the image $I$ shown in Figure ??. The image is $16 \times 16$ 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 ??. (Ignore the border issues) **4 pts.**

ii) If $F(u,v)$ is the discrete Fourier transform of the resulting image $I_1$, what is the value of $F(0,0)$? **1 pt.**

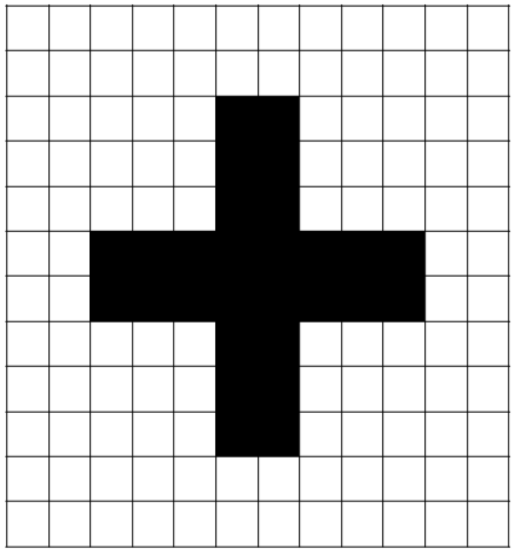
iii) Let $I_2$ be the image obtained by the correlation of image $I$ and the kernel $K$. Compute $I_2$ in terms of $I_1$. **1 pt.**

d) In JPEG compression:

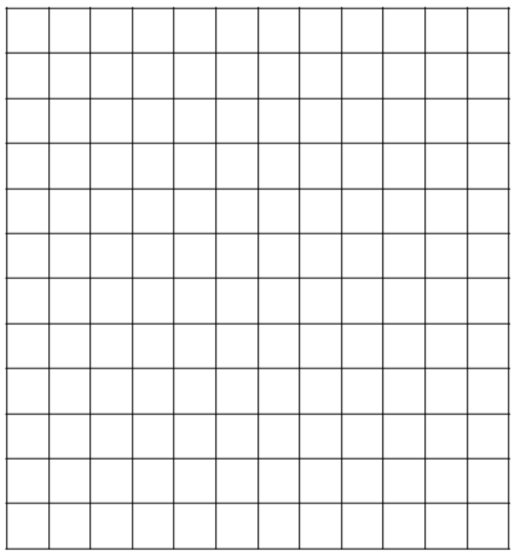
i) Why is the discrete cosine transform used instead of the Fourier transform in JPEG compression? **1 pt.**

ii) Mention one "lossy" step in the JPEG compression. **1 pt.**

iii) DCT transform applied to a $8 \times 8$ block of the image results in the block shown in Figure ??. Draw the order (on Figure ??) in which these coefficients are encoded (or accessed). **1 pt.**



(a) Input Image $I$



(b) Output Image $I_1$

Figure 2: Convolution with kernel $[-1, 0, 1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
| C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 |
| C17 | C18 | C19 | C20 | C21 | C22 | C23 | C24 |
| C25 | C26 | C27 | C28 | C29 | C30 | C31 | C32 |
| C33 | C34 | C35 | C36 | C37 | C38 | C39 | C40 |
| C41 | C42 | C43 | C44 | C45 | C46 | C47 | C48 |
| C49 | C50 | C51 | C52 | C53 | C54 | C55 | C56 |
| C57 | C58 | C59 | C60 | C61 | C62 | C63 | C64 |

Figure 3: DCT coefficients

## Question 3: Morphological Operators and Performance (9 pts.)

a) Define erosion and dilation, by giving their set formulation. **2 pts.**

b) Erode the image on Figure ??. Use the structuring element given in Figure ??. Consider the white pixels as foreground. You may draw your result on the image. **2 pts.**

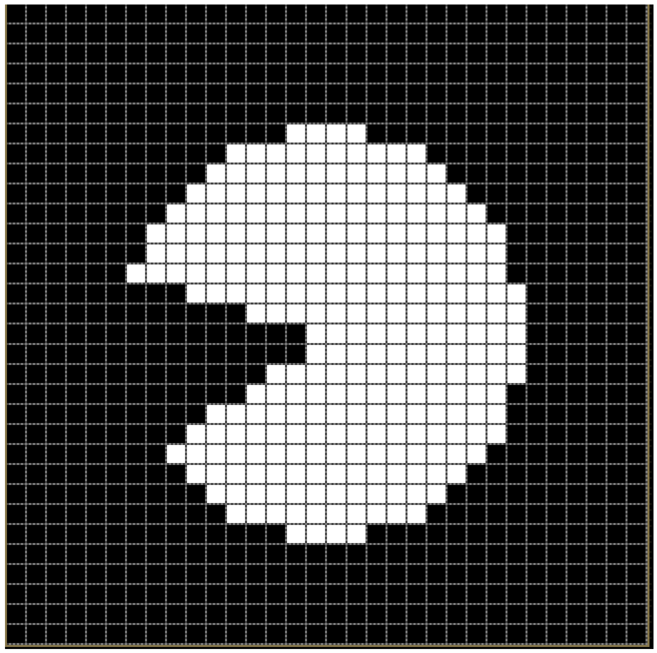


Figure 4:

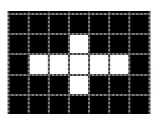


Figure 5: Structuring element

c) Describe a way to perform edge substraction on a binary image by using a morphological approach. **1 pt.**

d) Three binary classifiers are applied to check if a patient has a disease or not. Below are the tables for the resulting performance of these classifiers. Plot the performance of the classifiers on a ROC Curve, and mention which one performs the best. **2 pts.**

| | |
|---|---|
| $TP = 63$ | $FP = 28$ |
| $FN = 37$ | $TN = 72$ |

Table 1: Classifier A

| | |
|---|---|
| $TP = 77$ | $FP = 77$ |
| $FN = 23$ | $TN = 23$ |

Table 2: Classifier B

| | |
|---|---|
| $TP = 24$ | $FP = 88$ |
| $FN = 76$ | $TN = 12$ |

Table 3: Classifier C

e) If a classifier D guesses the output randomly everytime, what will the corresponding ROC curve look like? **1 pt.**

f) If the ROC curve of a classifier E always lies below the ROC curve of classifier D, how can its performance be improved? **1 pt.**

## Question 4: Unitary Transforms and PCA (22 pts.)

The dataset $S$ contains $|S|$ images of faces with size $n \times n$. An image $f \in S$ can be thought of as an $n^2$ length vector. Images are distributed normally with mean $\mu$ and autocorrelation (or covariance) matrix $R_{ff} = E[(f-\mu)(f-\mu)^T]$.

a) What physical property of the dataset do the eigenvalues and eigenvectors of the autocorrelation matrix $R_{ff}$ capture? Comment specifically on the eigenvector corresponding to the largest eigenvalue. **2 pts.**

b) Let $c = A(f-\mu)$ where $A$ is $n^2 \times n^2$.

  i) Give the unitary transform $A$ which packs the most energy in the first $J$ coefficients of $c$. **1 pt.**

  ii) Prove that it does so.
  (Hint: Derive the relation between the autocorrelation matrices $R_{ff}$ and $R_{cc}$. Energy in the coefficient vector $c$ is given by $E = Tr(R_{cc})$ where $Tr$ is the trace operator and gives the sum of the diagonal entries of a matrix.) **5 pts.**

c) In PCA based data-driven compression of faces, we use the mean-face $\mu$ and the eigenfaces. $c = V^H(f-\mu)$ is the compressed version of image $f$ and $d = Vc + \mu$ is the decompressed version.

  i) What is the possible disadvantage of ignoring the mean in this procedure (and not subtracting or adding it) and why? Justifying mathematically, give the condition on mean $\mu$ under which it would not matter whether the mean is ignored or not. **3 pts.**

  ii) The algorithm is trained on dataset $S$ and is then used to compress photographs from the dataset $C$ which is a dataset of chairs. Why doesn't it work well despite *"packing the most energy in its first J coefficients"*? **1 pt.**

  iii) Suppose the image being compressed $f \in C$ is a photograph of a chair. Can there be a choice of $J$ for which the SSD error between $d$ and $f$ is $0$? Prove why or why not. **2 pts.**

d) $g$ is a particular $n \times n$ image which is to be searched in the database $S$. One method to do this search is to find the square (for convenience) of Euclidean distance between $g$ and every $f \in S$ using $d_E(f,g)^2 = \|f-g\|_2^2 = (f-g)^T(f-g)$ and find the $f$ with the minimum distance from $g$.

  i) Describe a method for faster approximation of the Euclidean distance based on PCA. **2 pts.**

  ii) Derive an expression for the expected error in this calculation $E(d_E^2 - d_A^2)$ where $d_E$ is the Euclidean distance and $d_A$ is distance given by your approximation method.
  (Note: Images in the dataset are identicially and independently distributed random variables with distribution $N(\mu, R_{ff})$. Assume $g$ is also from the dataset $S$. Hint: split the full coefficient vector into the part truncated and the part kept) **4 pts.**

  iii) Let $g$ be any $n \times n$ image. One way to see if $g$ is a face is to check if $d_E(g,\mu) < t$ for some threshold $t$. Give another distance function $d_M$ that gives a more accurate test (i.e. works better as a classifier) and explain briefly why it is likely to be better. **2 pts.**

## Question 5: Optical Flow (23 pts.)

a) The Lucas-Kanade algorithm can be used to estimate the optical flow of an image sequence

i) State the 3 assumptions of this method **3 pts.**

ii) The Lucas-Kanade method provides the following result

$$\begin{pmatrix} \sum \mathrm{I}_x^2 & \sum \mathrm{I}_x\mathrm{I}_y \\ \sum \mathrm{I}_x\mathrm{I}_y & \sum \mathrm{I}_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} \sum \mathrm{I}_x\mathrm{I}_t \\ \sum \mathrm{I}_y\mathrm{I}_t \end{pmatrix}$$

Where $(u, v)$ is the displacement of a pixel within an image patch.
Mathematically derive the above result using the 3 basic assumptions of this method.
Indicate the aperture problem. **4 pts.**

b) Some problems may arise while applying the previous method:

i) Sometimes the solution found is not unique. Under what condition does this happen and why? Give an example and a mathematical explanation. **3 pts.**

ii) In some cases the movement of the objects on the images is too big. What can be done to solve this problem and still be able to apply the Lucas-Kanade algorithm? Explain. **3 pts.**

c) One of the many uses of the optical flow is the determination of time-to-contact (TTC). This is, the time it will take for a camera moving forward to collide. This can be known without any information on the velocity of the camera (only the assumption that it is constant) or the distance to be traveled. To simplify this, we will also assume that there is no rotational component on the movement of the camera, so that it only translates forward at a constant velocity.

i) The first step consists of computing the focus of expansion (FOE). The focus of expansion is the point around which all of the flow vectors diverge, since the camera is moving forward. This point is also the point to which the camera is directed. Draw such an optical flow field and give a method to compute the FOE. **3 pts.**

ii) Now you will have to estimate the TTC. To simplify the task, you may use the diagram from Figure **??**.
From the diagram we can observe that the camera is moving along the $z$ axis towards the FOE, with velocity $V = \frac{\delta Z}{\delta t}$. $P = (X, Y, Z)$ is a generic point on the same plane as the FOE. $p$ and $p'$ are the projections of $P$ at time $T$ and $T + 1$. You can assume that the $x$ coordinate of $P$ is the same as the FOE. You may take as a starting clue the result given by similar triangles:

$$\frac{y}{f} = \frac{Y}{Z}$$

You will obtain the TTC in terms of $y$. You must relate this result with the optical flow vector and the FOE in order for your result to be general. **7 pts.**

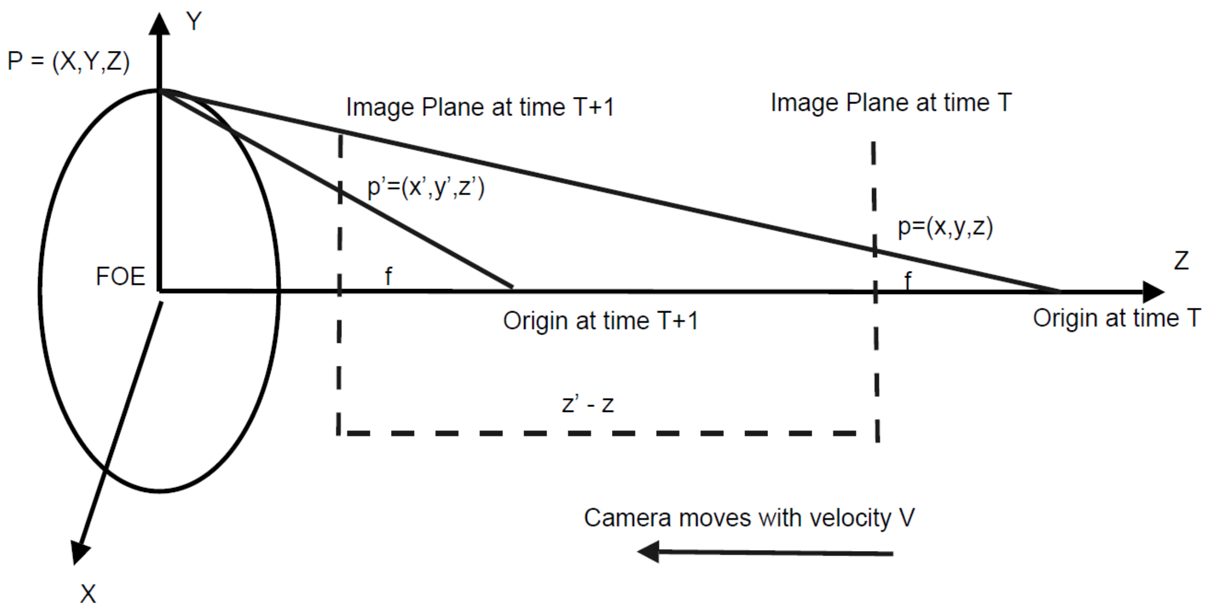



Figure 6: Diagram for the computation of the TTC

## Question 6: Graphics Pipeline and Shaders (30 pts.)

a) **Graphics Pipeline**

The following diagram shows an overview of the graphics pipeline.

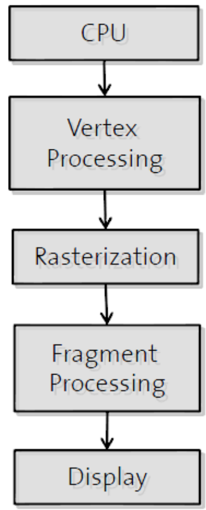



Figure 7: Simplified Graphics Pipeline

i) Referring to the diagram, explain the difference between the fixed function pipeline and the programmable pipeline. **3 pts.**

ii) What is the purpose of the rasterization step? What are the inputs and outputs? **3 pts.**

iii) Name two steps of the graphics pipeline that are not shown in the simplified diagram and indicate where they are performed. **2 pts.**

b) **Shaders I**

Consider the following vertex and fragment shaders:

**Vertex Shader**

```
uniform mat4 modelview_mat;
uniform mat4 proj_mat;
uniform vec3 light_pos;

attribute vec4 position;
attribute vec3 normal;

varying vec3 obj_position;
varying float intensity;

void main() {
   normal = normalize(normal);
   vec3 vec_to_light = normalize(light_pos.xyz - position.xyz);
   intensity = dot(normal, vec_to_light);
```

```
    intensity = clamp(intensity, 0.0, 1.0);

    obj_position = position.xyz;

    gl_Position = proj_mat * modelview_mat * position;
}
```

**Fragment Shader**

```
varying vec3 obj_position;
varying float intensity;

void main() {
    vec4 color;
    int x = mod(trunc(fract(obj_position.x) * 10), 2);
    int y = mod(trunc(fract(obj_position.y) * 10), 2);
    int z = mod(trunc(fract(obj_position.z) * 10), 2);
    if ((x == 0 && y == 0 && z == 0) ||
        (x == 0 && y == 1 && z == 1) ||
        (x == 1 && y == 0 && z == 1) ||
        (x == 1 && y == 1 && z == 0)) {
            color = vec4(0.8, 0.8, 0.8, 1.0);
    } else {
            color = vec4(1.0, 0.0, 0.0, 1.0);
    }

    gl_FragColor = intensity * color;
}
```

GLSL built-in functions:
- fract(x) : returns the fractional part of x, example: fract(15.74) = 0.74
- trunc(x) : returns the integer part of x, example: trunc(15.74) = 15
- mod(x,y) : returns x modulo y
- clamp(x, a, b) : computes min(max(x, a), b)

i) Which of the images in Figure ?? was generated by the given shader pair? **2 pts.**

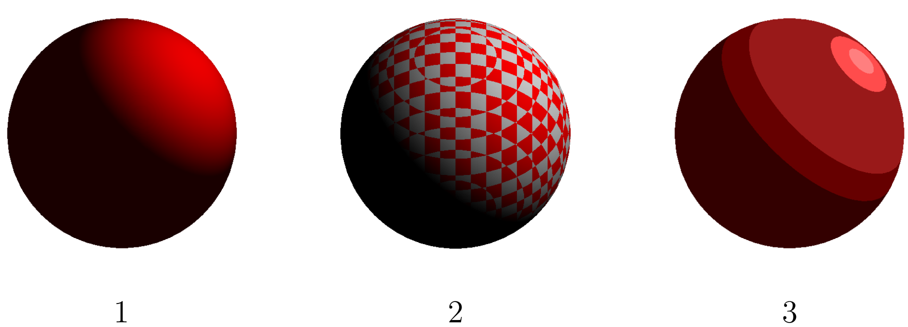



Figure 8: Shader outputs

ii) A usual vertex shader contains the line

```
gl_Position = proj_mat * modelview_mat * position;
```

What is computed here? **2 pts.**

iii) In the fragment shader shown above the pixel color is multiplied by an intensity term. What effect does this achieve? **3 pts.**

iv) The value of the vector 'color' in the fragment shader depends on the 3D position of the fragment. Explain how it is computed. **6 pts.**

c) **Shaders II**

i) Figure ?? illustrates a technique to add details to a surface without using more polygons. What is the name of this technique and how does it work? **4 pts.**

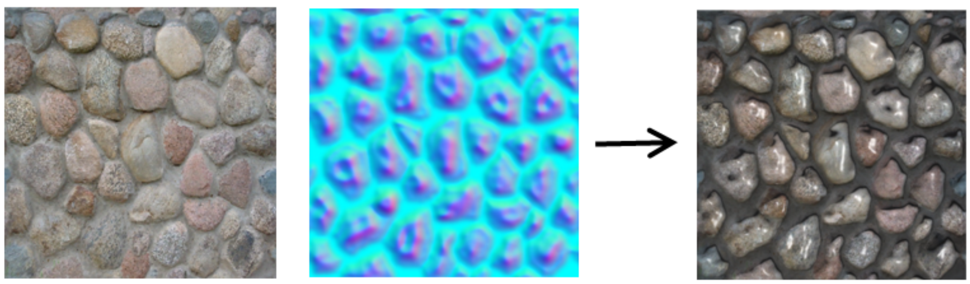


Figure 9: Adding details to a surface

Figure 10: Silhouette

ii) In Figure ?? the same technique is applied to a sphere. Why is the silhouette of the sphere flat and not bumpy as you would expect? **2 pts.**

d) **Texture Mapping**

i) If we read out a cube map at direction $\mathbf{d} = (2, 3, 5)^T$, which texture (face) of the cube map will be accessed, and at which texture coordinate? **3 pts.**

## Question 7: Colors and Reflection Models (30 pts.)

a) **Colors**

It is the year 2150. Due to the omnipresence of monitors, which emit standardized RGB light, the human eye evolved and the three cones ($c_r$, $c_g$, $c_b$) of the eye show the sensitivity spectra ($\bar{r}_r$, $\bar{r}_g$, $\bar{r}_b$) displayed in Figure ??. As you can see, each cone features a hat shaped spectrum with a maximal sensitivity at $\lambda_r$, $\lambda_g$ and $\lambda_b$ respectively. The following questions are with respect to an eye characterized by the illustrated sensitivity spectra.

i) Which wavelengths belong to the spectrum of visible light? **1 pt.**

ii) What can you say about the color perception of monochromatic light from the spectrum 450nm to 500nm? **2 pts.**

iii) What can you say about the color perception of monochromatic light from the spectrum 500nm to 600nm? **2 pts.**

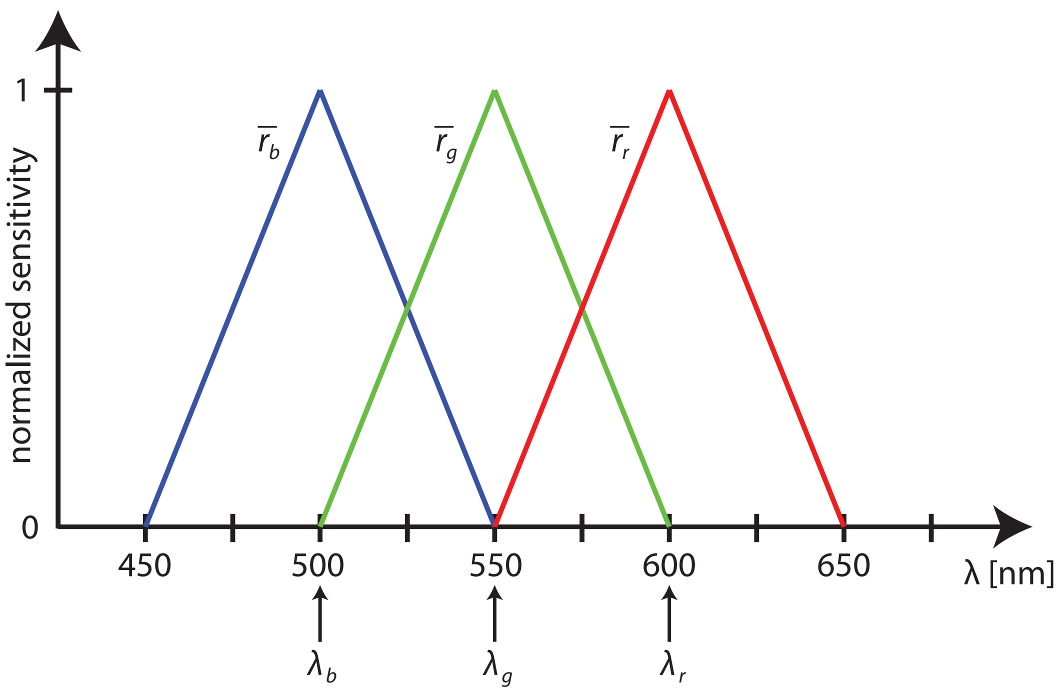



Figure 11: Sensitivity spectra of the three cones $c_r$, $c_g$, $c_b$ of the human eye in 2150.

Assume we use the three primaries $r(\lambda) = \delta(\lambda - \lambda_r)$, $g(\lambda) = \delta(\lambda - \lambda_g)$ and $b(\lambda) = \delta(\lambda - \lambda_b)$ with the corresponding color coordinates R, G and B for a monitor ($\delta()$ denotes the Dirac delta function). This means, R, G and B describe how much the wavelengths $\lambda_r$, $\lambda_g$ and $\lambda_b$ are activated. Similar to the CIE chart of the XYZ colorspace, we project the colors on a plane with normal $\mathbf{n} = (1, 1, 1)^T$. This leaves us with the 2D coordinates r and g:

$$\mathrm{r} = \frac{\mathrm{R}}{\mathrm{R}+\mathrm{G}+\mathrm{B}} \quad \mathrm{g} = \frac{\mathrm{G}}{\mathrm{R}+\mathrm{G}+\mathrm{B}} \quad \mathrm{b} = 1 - \mathrm{r} - \mathrm{g}$$

iv) In the rg-chart (Figure ??) mark the three primaries and indicate the gamut of visible light. **2 pts.**

v) Mark the spectral colors for the wavelengths 475nm, 500nm, 550nm, 575nm and 625nm and the whitepoint $(\mathrm{R}, \mathrm{G}, \mathrm{B})^T = (1, 1, 1)^T$ in the rg-chart (Figure ??). **3 pts.**

vi) Denote the purple line in the rg-chart (Figure ??). **1 pt.**

vii) Mark the colors ($C_1$, $C_2$) described by the following two power spectra ($P_1$, $P_2$) in the chart (Figure ??) and describe the perception of these colors. **6 pts.**

$$P_1(\lambda) = \begin{cases} 2 & \text{if } \lambda = 475 \\ 1 & \text{if } \lambda = 550 \\ 1 & \text{if } \lambda = 600 \\ 0 & \text{else} \end{cases} \qquad P_2(\lambda) = \begin{cases} 1 & \text{if } \lambda = 500 \\ 2 & \text{if } \lambda = 575 \\ 0 & \text{else} \end{cases}$$

viii) Are there any visible colors that can not be reproduced with the given three primaries? Justify your answer! **2 pts.**

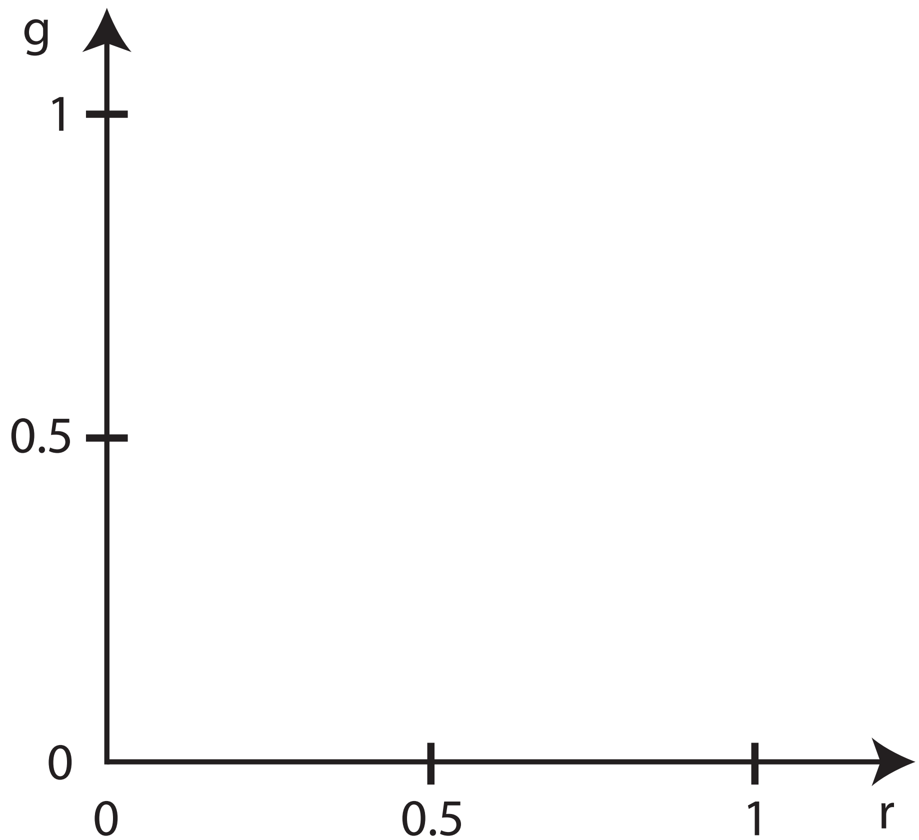



Figure 12: rg-chart

b) **Reflection Models**

i) Name the three components of the Phong reflection model. **2 pts.**

ii) Figure ?? shows the luminous intensity diagram resulting from the evaluation of the Phong reflection model at a surface point with fixed reflection constants. The only thing changing from the left to the right diagram is the position of the light source. In Figure ??, draw an arrow with the direction of the incoming light illuminating the surface point in both charts. **2 pts.**

iii) When we created the luminous intensity diagram in Figure ?? we only used two components of the Phong reflection model. This means, one component of the Phong reflection model is equal zero. Which one? Explain your answer! **4 pts.**

iv) If this missing component would also be used for the evaluation of the reflection model, how would this change the two luminous intensity diagrams? **3 pts.**

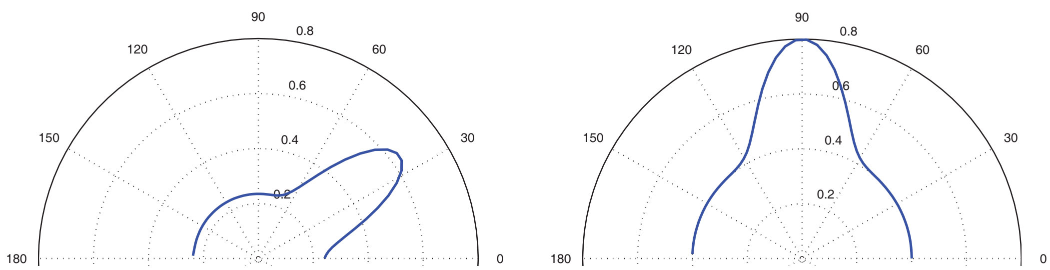



Figure 13: Luminous intensity diagram for two different light positions.

## Question 8: Transformations and Projections (30 pts.)

a) **Transformations**

   i) Object coordinates are denoted by $\mathbf{x}_c = (x_c, y_c, z_c)^T$ and world coordinates by $\mathbf{x} = (x, y, z)^T$. The model matrix $\mathbf{M}$ transforms the object as follows: The point $(0, 0, 0)^T$ in object coordinates ends up at $(1, 2, 3)^T$ in world coordinates. The $x_c$-axis of the object points in the direction $(0, -1, 0)^T$ in world coordinates and its $y_c$-axis points in the direction $(-1, 0, 0)^T$. Determine the model matrix $\mathbf{M}$. **4 pts.**

b) **Transformation of Normals**

   Consider a plane $H$ defined as the set of points $(x, y, z)^T$ satisfying the implicit equation $2x + 3y - z + 4 = 0$.

   i) What is the unit normal vector of this plane? **2 pts.**

   ii) The matrix $\mathbf{A}$ given below transforms points $\mathbf{p}$ in homogeneous coordinates to $\mathbf{p}' = \mathbf{A}\mathbf{p}$. Apply this transformation to the plane $H$. What is the implicit equation of the transformed plane $H'$? **4 pts.**

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

   Hint: the inverse of a $2 \times 2$ matrix can be computed as follows:

$$\mathbf{B} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \rightarrow \quad \mathbf{B}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

c) **Quaternions**

   Consider the quaternion $q = \frac{1}{\sqrt{2}} i + \frac{1}{\sqrt{2}} j$.

   i) Transform the point $\mathbf{p} = (0, 0, 2)^T$ with the quaternion $q$. **4 pts.**

   ii) Describe the transformation represented by $q$. **2 pts.**

   iii) The inverse of $q$ is $q^{-1} = -\frac{1}{\sqrt{2}} i - \frac{1}{\sqrt{2}} j$. Using the quaternion multiplication rules, show that $q$ and $q^{-1}$ do in fact represent the same geometric transformation. **3 pts.**

d) **Projections**

   A projection matrix has the following structure:

$$\mathbf{P} = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

   where $a$, $b$, $c$ and $d$ are unknown at this point.

   i) Given the points $\mathbf{p}_1 = (-3, -2, -2)^T$ and $\mathbf{p}_2 = (9, 6, -6)^T$ in camera coordinates, transform those points to screen coordinates using the projection matrix $\mathbf{P}$ and perform the perspective division. **2 pts.**

ii) Given the fact that the point $\mathbf{p}_1$ in camera coordinates gets projected to the point $(-1,-1)^T$ in screen coordinates with z-buffer value $-1$, and the point $\mathbf{p}_2$ gets projected to point $(1,1)^T$ in screen coordinates with z-buffer value $1$, determine the values of $a$, $b$, $c$ and $d$ in $\mathbf{P}$. **4 pts.**

e) **Shadow Mapping**

i) In what order are the following computations A) - F) performed when rendering an image using shadow mapping? **3 pts.**

**A)** Compare depth map value to distance value

**B)** Transform screen coordinate of the light into shadow map u/v coordinate

**C)** Transform vertex into screen coordinates of the camera

**D)** Read value from depth map

**E)** Transform vertex into screen coordinates of the light

**F)** Transform fragment position into screen coordinates of the light

ii) What kind of rendering artifacts do you expect when the spatial resolution of the depth map is too low? **1 pt.**

iii) What kind of rendering artifacts do you expect when the quantization of depth map values is too coarse? (e.g. when using 8 bit textures instead of 16 bit) **1 pt.**