ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visual Computing
2008

Prof. M. Gross (Computer Graphics Lab)
Prof. J. M. Buhmann (Machine Learning Group)
B. Bickel, P. Kaufmann, R. Oeschger
P. Pletscher, C. Sigg

Final Exam

21 August 2008

First and Last name: _____

ETH number: _____

Signature: _____

General Remarks

- At first, please check that your exam questionnaire is complete (there are 16 pages total).
- Remove all material from your desk which is not allowed by examination regulations.
- Fill in your first and last name and your ETH number and sign the exam. Place your student ID in front of you.
- You have 3 hours for the exam. There are six 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. Do *not* write on the question sheet, unless explicitly stated in the instructions.
- 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	OpenGL, Pipeline, Shading and Colors	30		
2	From Perceptrons to SVMs	30		
3	Fourier Transform and Texture Mapping	28		
4	Motion Blur and Deblurring	30		
5	Transformations and Projections	32		
6	Parametric Density Estimation	30		
Total		180		

Grade: .....

Question 1: OpenGL, Pipeline, Shading and Colors (30 pts.)

a) OpenGL and the Graphics Pipeline

i) What is the OpenGL command *glNormal3fv(arg)* used for and what can you say about the type of the argument *arg* of this function? **1 pt.**

ii) How do vertex arrays combined with index arrays improve the performance of OpenGL applications compared to sending vertices individually? **1 pt.**

iii) Give two examples of buffers present in the graphics pipeline and explain their purpose. **1 pt.**

iv) What is the general reasoning for using a pipeline architecture? What has to be fulfilled such that a pipeline architecture can be applied? **3 pts.**

v) Name the two stages 1 and 2 of the graphics pipeline in Figure 1. **1 pt.**

Figure 1: *Graphics Pipeline*

vi) Mark in the following table for each row if the given task is performed in stage 1, 2 or in another stage of the classical graphics pipeline given in Figure 1. **3.5 pts.**

Task	**1**	**2**	**Other**
Transform & projection			
Stencil test			
Clipping & culling			
Texture filtering			
Input event handler			
Blending			
Lighting			

vii) Explain 'double buffering' and why it is used. **2 pts.**

viii) Definition of *fillrate*:

The theoretical **fillrate** *refers to the number of pixels a video card can render and write to video memory in a second. Fillrates are obtained by multiplying the number of raster operation units by the clock frequency of the graphics processor unit of a video card.*

Given the technical data in the following table for a GeForce GTX 260 GPU. What is the theoretical maximum fillrate of such a graphics card? **3 pts.**

	GeForce GTX 260
Core clock max (MHz)	576
Stream processors	192
Effective memory clock (MHz)	1998
Raster operation units	28

ix) You are running a graphics application on your display with a resolution of 1920x1440 pixels. This application achieves 55% of your theoretical maximum fillrate from the question above. You want to maintain 60 frames per second. Can the fillrate become the performance bottleneck? Motivate your answer. **2 pts.**

x) You are given a graphics application where the fillrate is the performance bottleneck. Your are not allowed to change the hardware or the geometric complexity of your application. How could you try to improve the performance? **2 pts.**

b) Shading and Colors

i) Name the shading model used to render each of the three images in Figure 2 and briefly explain for each model how the color of a pixel in a triangle is computed. **3 pts.**

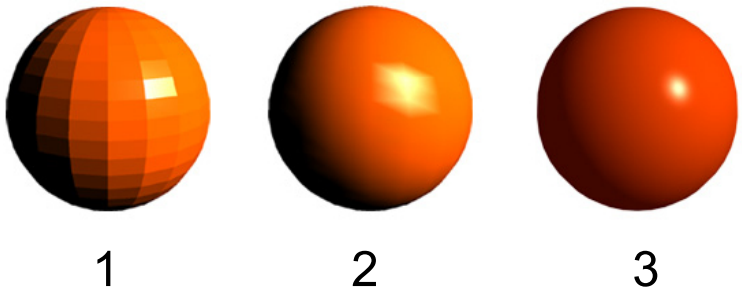


Figure 2: *Different Shading Models*

ii) Name the three kinds of light reflections supported by the OpenGL illumination model. **1.5 pts.**

iii) Why does the transformation matrix from the CIE XYZ color space to an RGB color space contain negative values? **1.5 pts.**

iv) How is the CIE chromaticity diagram related to the XYZ space? State your answer in a mathematical formula for x, y and z. **1.5 pts.**

v) Given the x and y coordinates of a color in the CIE chromaticity diagram. Where are all other colors with the perceptually same hue located? **1 pt.**

vi) How are the two different segments of the CIE chromaticity diagram's boundary called? **1 pt.**

vii) What desirable property not present in the XYZ color space led to the development of the Lab and Luv color spaces? **1 pt.**

Question 2: From Perceptrons to SVMs (30 pts.)

Training and Test Error. Quantifying the performance of a classifier is an important part of a machine-learning application.

a) Give a definition of *training* and *test error* (also called *empirical* and *expected risk*, respectively). Write down formulae for both terms and explain the quantities which appear in them. **2 pts.**

b) Explain how to measure both quantities (exactly or approximately), given a dataset of N labeled samples. **2 pts.**

c) Assume that you have setup a classifier for a certain task. You measure test error and conclude that it is too high. In order to lower the test error you either ask for more training samples, or use a more sophisticated classifier (e.g. a non-linear instead of a linear classifier). Which choice is appropriate if the measured *training* error was low, which choice is appropriate if it was high? Give an explanation for both cases. **2 pts.**

Linear and Second Order Decision Rules. A training sample is a pair $(\mathbf{x}, y)$, where $\mathbf{x} = (x_1, \ldots, x_D)^\top$ is a D-dimensional feature vector and $y \in \{-1, +1\}$ is the associated label, for class $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

d) A linear decision rule that classifies the training samples correctly must satisfy

$$\forall \mathbf{x} \in \mathcal{C}_1 : \sum_{d=1}^{D} w_d x_d + w_0 \overset{!}{\leq} 0 \tag{1}$$

$$\forall \mathbf{x} \in \mathcal{C}_2 : \sum_{d=1}^{D} w_d x_d + w_0 \overset{!}{\geq} 0, \tag{2}$$

where $w_0, \ldots, w_D$ define the location and orientation of the decision boundary. Construct $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ such that the same requirement can be more compactly represented in a single inequality

$$\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} \overset{!}{\geq} 0. \tag{3}$$

2 pts.

e) A *second-order* decision rule also includes weighted pairwise interaction terms

$$w_{d,e} x_d x_e$$

($d \neq e$) between all features. Write down equations for this case, which are analogous to equations (1) and (2). Use the minimum number of additional weights necessary. How many weights do you need in total? Argue using the fact that $\sum_{i=1}^{n} i = \frac{(n+1)n}{2}$. **3 pts.**

f) Rewrite your new second-order equations more compactly, by using vector and matrix multiplications instead of summation symbols, wherever possible. You *don't* have to combine the two equations into a single equation. **2 pts.**

g) In going from equations (1) and (2) to (3), you replaced two inequalities with a single inequality. Why can't you apply the same trick to write the second-order decision rule as a single inequality? An explanation is enough, you don't have to provide a fix. **2 pts.**

Optimization Criteria. Training a classifier involves optimization of a mathematical criterion. This optimization is often iterative, e.g. a gradient-descent procedure. Depending on the criterion and the optimization procedure, we obtain different classifier architectures (e.g. Perceptrons or SVMs).

h) One possible criterion is the number of samples which are misclassified for the current weight vector $\mathbf{w}$. Explain why gradient-descent w.r.t. $\mathbf{w}$ fails to minimize this criterion. **1 pt.**

i) A variant of the Perceptron criterion is

$$J_q(\tilde{\mathbf{w}}) = \sum_{\tilde{\mathbf{x}} \in \mathcal{X}_b} \frac{(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} - b)^2}{\|\tilde{\mathbf{x}}\|^2}, \tag{4}$$

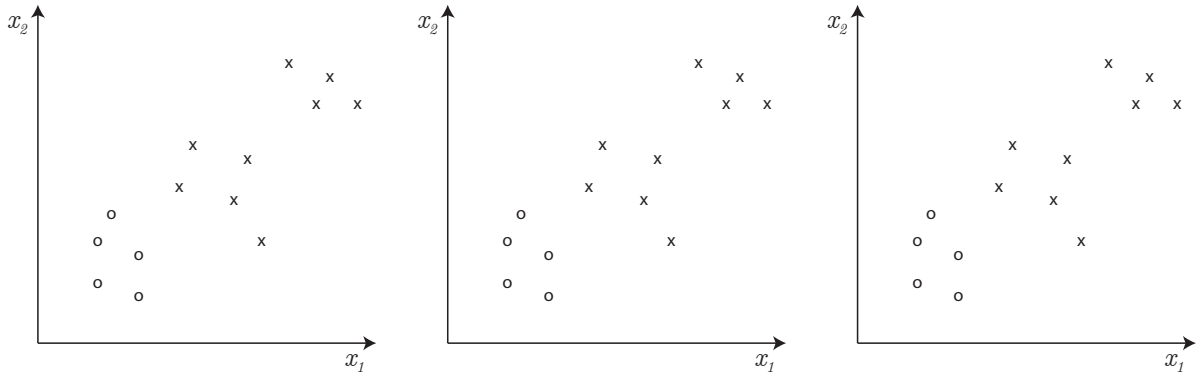
where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{w}}$ are defined as in equation (3), $b > 0$ is an additional constant and $\mathcal{X}_b$ is the set of all samples for which $\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} \not> b$. Compute the negative gradient $-\nabla J_q$ of the criterion w.r.t. $\tilde{\mathbf{w}}$. Finally, provide an iterative update rule for $\tilde{\mathbf{w}}$, based on the negative gradient and a learning rate parameter η. **4 pts.**

j) The sum in equation (4) can be extended over all samples $\mathcal{X}$ (not just those that don't fulfill a certain requirement), which leads to a third criterion

$$J_{\text{ls}}(\tilde{\mathbf{w}}) = \sum_{\tilde{\mathbf{x}} \in \mathcal{X}} \frac{(\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} - b)^2}{\|\tilde{\mathbf{x}}\|^2}. \tag{5}$$

This criterion is called *least-squares for classification*. Explain in words what minimization of J_{ls} means geometrically (you may also draw a sketch, but this is not necessary). Give one advantage and one disadvantage of an algorithm which uses J_{ls} instead of the standard Perceptron criterion from the lecture. **4 pts.**

Different Classifiers - Different Solutions. Here are three sketches of the same hypothetical data sample from two classes in two dimensions:

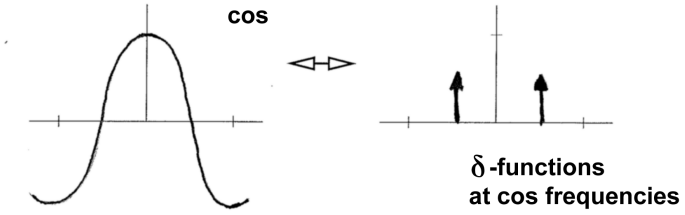


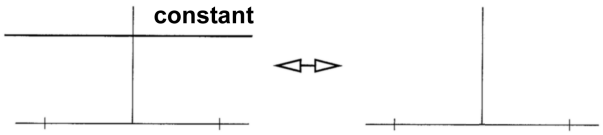
Perceptron *Least-Squares* *Linear SVM*

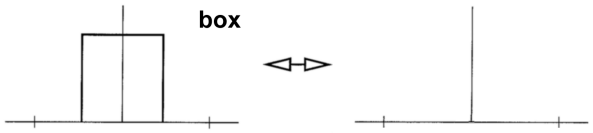
k) Draw three linear decision boundaries, one for the Perceptron, one for the least-squares classifier (from the previous question) and one for a linear SVM. Draw your answer directly onto the question sheet. Each decision boundary should be different from the others, a (qualitatively) correct solution of the respective algorithm and illustrate a property which is unique to that classifier. If a classifier can have multiple solutions for the same data set (e.g. depending on initialization), then draw a typical one. Explain what that unique property of each solution is. **6 pts.**

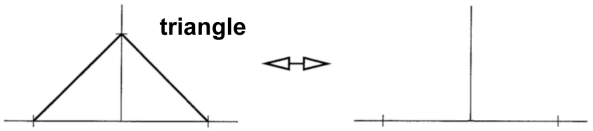
Question 3: Fourier Transform and Texture Mapping (28 pts.)

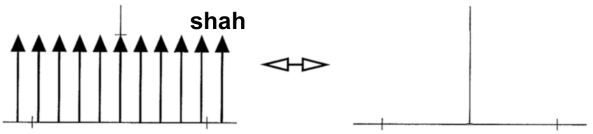
a) Sketch the corresponding fourier transforms of the following basic functions and describe them briefly as shown in the example below. **5 pts.**

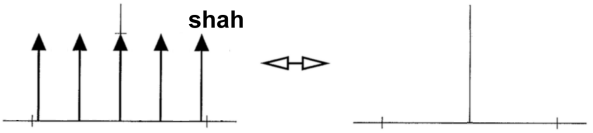


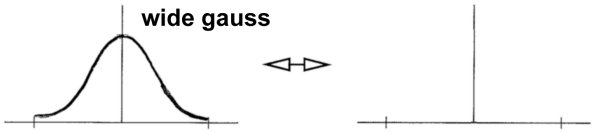


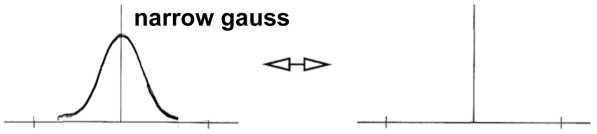




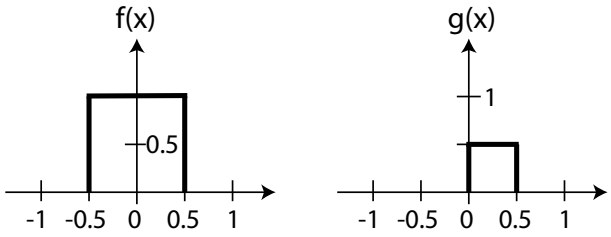








b) Convolve graphically the two functions $f(x)$ (signal) and $g(x)$ (filter) given in the illustration below. **3 pts.**



c) Texture mapping

i) Briefly describe the general concept of texture mapping in two sentences and draw a small sketch to illustrate your answer. **2 pts.**

ii) List three desirable properties of a texture parametrization. **1 pt.**

iii) Texture filtering is an important step in the graphics pipeline. Why is it needed at all? Name two negative effects that could occur without filtering. **3 pts.**

iv) What method allows to add small surface details such as wrinkles without increasing the object's geometric complexity? Briefly describe the method in two sentences and list three limitations of it. **3 pts.**

d) Perspective correct textures.

i) Texture mapping on a triangle can be done by linearly interpolating in screen-space the texture values of the triangles vertices. What is the problem using this simple algorithm? **2 pts.**

ii) Investigate the interpolation of points along a line segment. Consider the line segment

$$\mathbf{p_1} + t(\mathbf{p_2} - \mathbf{p_1}), t \in [0, 1]$$

defined by the two points $\mathbf{p_1}$ and $\mathbf{p_2}$ in world coordinate space. When transformed by a 4×4 matrix $\mathbf{M}$, the homogeneous points $\mathbf{r_1} = \mathbf{M}\mathbf{p_1} = (x_1, y_1, z_1, h_1)^T$ and $\mathbf{r_2} = \mathbf{M}\mathbf{p_2} = (x_2, y_2, z_2, h_2)^T$ are obtained. After homogenization we obtain the coordinates $\mathbf{s_1} = (x_1/h_1, y_1/h_1, z_1/h_1, 1)^T$ and $\mathbf{s_2} = (x_2/h_2, y_2/h_2, z_2/h_2, 1)^T$, respectively.
Transform all points on the line segment given above using the matrix $\mathbf{M}$ by finding a closed form of the function $f(t, h_1, h_2)$ so that

$$\mathbf{s} = \mathbf{s_1} + f(t, h_1, h_2) \cdot (\mathbf{s_2} - \mathbf{s_1}).$$

4 pts.

iii) Figure 3 shows a line segment lying in the x-z plane that corresponds to a single scanline of a triangle. Let $p_3 = p_1 + t(p_2 - p_1)$, for some t satisfying $0 \leq t \leq 1$, be the x-coordinate of an interpolated point on the projection plane. Proof that the reciprocal of the corresponding z-coordinate of the point (x_3, z_3) can be correctly interpolated in a linear manner across the face of a triangle, by showing

$$\frac{1}{z_3} = \frac{1}{z_1}(1-t) + \frac{1}{z_2}t.$$

Hint: Describe the line with the equation $ax + bz = c$ and use the fact that $x = \frac{p \cdot z}{-e}$.

5 pts.

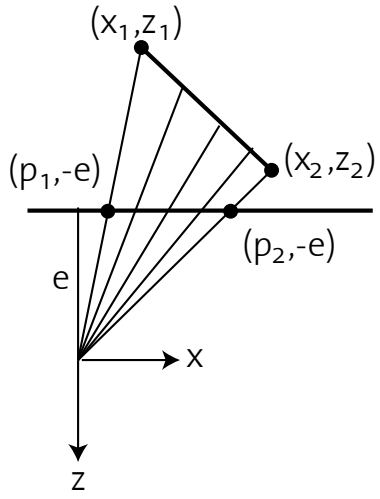


Figure 3: *The line segment corresponding to a single scanline of a triangle is sampled by casting rays.*

Question 4: Motion Blur and Deblurring (30 pts.)

In this question we study the motion blurring process and also discuss deblurring strategies. We are interested in motion blur that is caused by a moving camera.

Continuous Kernels in Image Processing. Let's denote the true image by $f(x, y)$, which is convolved with a kernel $\psi(x, y)$ to produce the observed image $g(x, y)$:

$$g(x, y) = f(x, y) \star \psi(x, y).$$

a) In the lecture, it was shown that the Dirac δ-function is the kernel which fulfills the identity $f(x, y) = g(x, y)$. The 1D Dirac is given by

$$\delta(x) = \lim_{\epsilon \to 0} \frac{1}{\sqrt{2\pi}\epsilon} \exp\left(\frac{-x^2}{2\epsilon^2}\right). \tag{6}$$

Observe that it is the limit of a Gaussian distribution where the variance tends towards zero. Derive the 2D Dirac $\delta(x, y)$ as follows:

1. Write down the formula for a 2D Gaussian distribution with uncorrelated random variables, each having a variance ϵ^2. *Hint: Start with the formula for a 1D Gaussian, i.e. equation* (6) *without the limit.*
2. Simplify your formula as much as possible.
3. Take the limit of $\epsilon \to 0$, as in the 1D case. **3 pts.**

b) Derive the Fourier transform of the two-dimensional Dirac $\delta(x, y)$. The two-dimensional Fourier transform $\hat{f}(u, v)$ of $f(x, y)$ is given by

$$\hat{f}(u, v) = \mathcal{F}[f(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-\imath 2\pi(ux + vy)) dx dy.$$

You may assume that the properties of the 1D Dirac w.r.t. integration also hold in 2D. **3 pts.**

c) Write down a symmetric motion blur kernel $\psi(x, y)$ for the following situation: You take a picture out of a glass elevator moving upwards. During the exposure, the camera sensor moves 80 pixels in vertical direction. **2 pts.**

d) Derive the Fourier transform of the motion kernel. Which frequencies in the image will be preserved, which ones destroyed? Argue based on the Fourier transform. **2 pts.**

Blurring and Deblurring using Linear Algebra. A digital image is discrete, so we can represent it as a matrix. In the following, the input image $\mathbf{F}$, the kernel $\mathbf{\Psi}$ and the output image $\mathbf{G}$ are all matrices. Discrete 2D convolution $\mathbf{G} = \mathbf{F} \star \mathbf{\Psi}$ is defined as

$$G_{i,j} = \sum_{n=-w/2}^{w/2} \sum_{m=-w/2}^{w/2} F_{i-n,j-m} \Psi_{n,m}.$$

w is the kernel window parameter, so $\mathbf{\Psi}$ is of size $(w + 1) \times (w + 1)$. In all of the following, ignore boundary problems where the convolution window extends beyond the input image.

e) Consider the discrete analogue of the 2D Dirac $\delta(x, y)$: Construct the kernel matrix $\mathbf{\Psi}$ that has the identity transform property $\mathbf{G} = \mathbf{F} \star \mathbf{\Psi} = \mathbf{F}$. **2 pts.**

f) Construct the discrete analogue of the motion blurring kernel, as it has been defined in question c). **2 pts.**

In the discrete case, we run into problems if the total window width W is even (which 80 is, which was given in the question), as the kernel can't be symmetric anymore. We counted everything as correct that was only wrong by ± 1 in the normalization or in the size. This is also a problem with the next subquestion.

g) Discrete convolution of $\mathbf{F}$ with the motion blurring kernel can be implemented by standard matrix multiplication. Construct a suitable *filter matrix* $\mathbf{P}$ and derive a linear matrix equation which implements the same motion blurring process as in question f). **2 pts.**

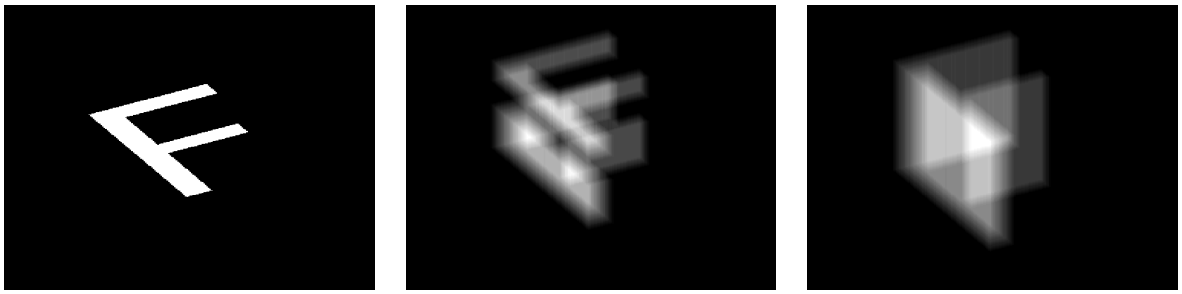
h) How do you reconstruct $\mathbf{F}$ from $\mathbf{G}$ and $\mathbf{P}$? What properties must the matrices fulfill, such that an exact reconstruction is possible? What criterion from linear algebra will give you an indication of how well posed the problem is? **2 pts.**

Flutter Shutter. Assume now that we split the exposure into m equally spaced time intervals. We're given a binary sequence $\mathbf{b}$ of length m (e.g. $m = 5$ and $\mathbf{b} = 01101$). For each time interval k, the camera shutter blocks incoming light if $b_k = 0$, otherwise ($b_k = 1$) it lets the light go through. You again take a picture in the elevator moving upward, but this time the camera sensor moves exactly m pixels during exposure. In the following, you can assume that m is odd.

i) What is the discrete motion blur kernel matrix $\tilde{\mathbf{\Psi}}$ for the flutter shutter? **4 pts.**

j) Construct a suitable filter matrix $\tilde{\mathbf{P}}$ and derive a linear matrix equation which implements the flutter shutter blurring process. **4 pts.**

Which image was convolved by which kernel? Below is an input image and two output images that were generated by convolving the input image with the two kernels from above (standard motion blur and flutter shutter). For the flutter shutter, the sequence $\mathbf{b} = 01101$ was used.



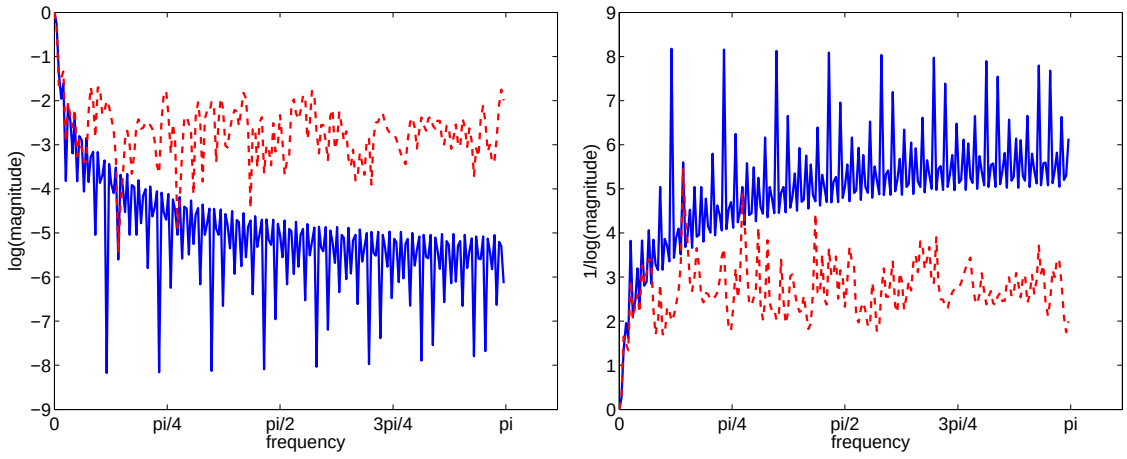
input image output 1 output 2

k) Assign the kernels to the output images and motivate your choice. **2 pts.**

(Continued on the next page.)

Reconstruction with the flutter shutter. Below is a plot that shows the discrete Fourier transform (DFT) magnitudes of the standard motion kernel and the flutter shutter kernel, for a random sequence $\mathbf{b}$.

I) Given that the reconstrution of the input image from the flutter shutter is more accurate, identify the curve that corresponds to the flutter shutter kernel. Explain your answer. **2 pts.**



Question 5: Transformations & Projections (32 pts.)

a) OpenGL

i) OpenGL provides two matrix stacks that define how geometry is transformed. What are they called? Write down the equation describing how those two transformations are applied to a point $\mathbf{p} = (x, y, z, 1)^T$ in object coordinates. **2 pts.**

ii) Explain why OpenGL provides stacks for transformation matrices instead of only storing the current matrices. **1 pt.**

b) Quaternions

i) Find the quaternion $\mathbf{q}$ representing a rotation around the x-axis by $\phi = 90$ degrees. Find the 3×3 matrix $\mathbf{R}$ describing the same transformation. Transform the point $\mathbf{p} = (1, 2, 3)^T$ using the quaternion. **3 pts.**

ii) What are the advantages of using quaternions to work with rotations rather than using 3×3 matrices? **2 pts.**

iii) Given an arbitrary 3×3 transformation matrix $\mathbf{M}$, a normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$ can be transformed using $\mathbf{n}' = (\mathbf{M}^{-1})^T \cdot \mathbf{n}$. How does the normal vector $\mathbf{n}$ have to be transformed if the transformation is a rotation defined by a unit quaternion $\mathbf{q} = c + \mathbf{u}$? **2 pts.**

c) Transformations

i) Consider the 5 transformation matrices $\mathbf{M}_1$ to $\mathbf{M}_5$:

$$\mathbf{M}_1 = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{M}_3 = \begin{pmatrix} 0.5 & 1.0 & 0.7 & 0 \\ 0.3 & 2.3 & 4.2 & 0 \\ 1.2 & 1.3 & 0.2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{M}_4 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad \mathbf{M}_5 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Those can be interpreted as transformations in $\mathbb{R}^4 \rightarrow \mathbb{R}^4$ ($\mathbf{M}_i \cdot (x, y, z, w)^T \mapsto (x', y', z', w')^T$) or as transformations in $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ (by performing the homogeneous division). In the table below, insert one of the letters *L*, *A* or *X* in each cell, indicating for each of the two interpretations:

- *L*: The transformation is linear
- *A*: The transformation is only affine, but not linear
- *X*: The transformation is not affine

3 pts.

	$\mathbb{R}^4 \rightarrow \mathbb{R}^4$	$\mathbb{R}^3 \rightarrow \mathbb{R}^3$
$\mathbf{M}_1$		
$\mathbf{M}_2$		
$\mathbf{M}_3$		
$\mathbf{M}_4$		
$\mathbf{M}_5$		

ii) A house model is defined in object coordinates as shown in Figure 4. It has been transformed to world coordinates using four different transformation matrices ($\mathbf{T}_1$ to $\mathbf{T}_4$). Find the corresponding transformation matrix for each transformed instance of the house in Figure 5. **2 pts.**

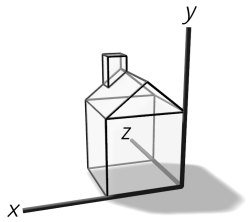


Figure 4: *House model in object coordinate system*

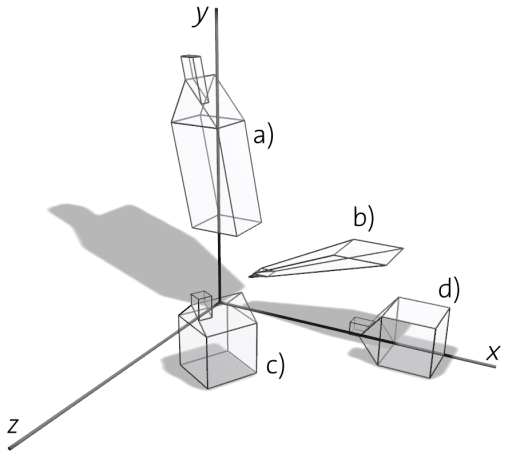


Figure 5: *Four transformed instances of the house model in the world coordinate system*

$$\mathbf{T}_1 = \begin{pmatrix} 0 & -1 & 0 & 5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_2 = \begin{pmatrix} 0.8776 & 0 & -0.4794 & 1.2757 \\ 0 & 1 & 0 & 0 \\ 0.4794 & 0 & 0.8776 & 1.8364 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{T}_3 = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_4 = \begin{pmatrix} 1 & -0.4 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

iii) You want to attach a 3D object to a camera with view matrix $\mathbf{V}$, so that its position and orientation relative to the camera is fixed and can be described by a constant transformation matrix $\mathbf{T}$. Write down an equation for the object-world transformation matrix of the object. **2 pts.**

d) Projections

i) Why do we need to specify the distances to the near and far clipping planes when defining a projection matrix? **1 pt.**

ii) Describe two kinds of visual artifacts that can occur if the distances to the near and far clipping planes are not chosen appropriately for a given scene. **1 pt.**

iii) In an OpenGL application, the projection matrix $\mathbf{P}$ is given as: **4 pts.**

$$\mathbf{P} = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & -3 \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{7}$$

Reconstruct the distances to the near and far clipping planes from $\mathbf{P}$, assuming that camera coordinates are defined in such a way that the camera looks in the $-z$ direction.

iv) An OpenGL application uses the projection matrix $\mathbf{P}$ (see Eq. 7) to render a scene to the screen (see Figure 6, left). You now want to run the application on four screens, each screen representing a quarter of the full image (see Figure 6, right). Compute the new projection matrix $\mathbf{P}'$ for the upper-right screen. Remember that visible 2D device coordinates after projection (x_d, y_d) lie in the range $-1 \leq x_d, y_d \leq 1$. **4 pts.**

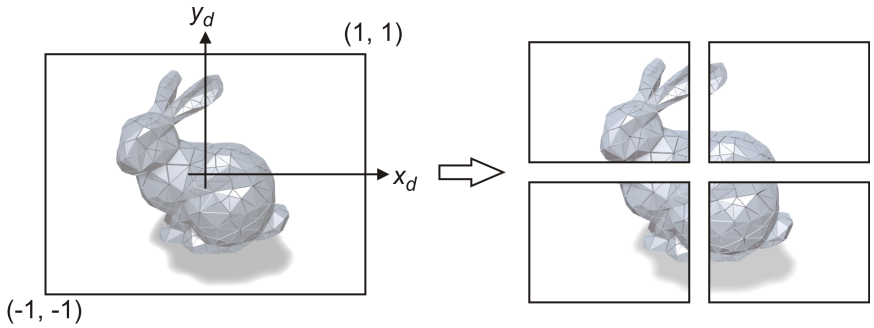


Figure 6: *Single screen rendering (left) and* 2×2 *multi screen rendering (right).*

v) Consider a scene with two cameras. The world-camera matrix of camera i ($i \in \{1, 2\}$) is $\mathbf{V}_i$ and its projection matrix is $\mathbf{P}_i$. A point $\mathbf{p}$ with unknown world coordinates is rendered at 2D normalized device coordinates (u_1, v_1) in camera 1. Given (u_1, v_1), proof mathematically that the possible positions at which the point would get rendered in camera 2 all have to lie on a straight line (see Figure 7). **5 pts.**

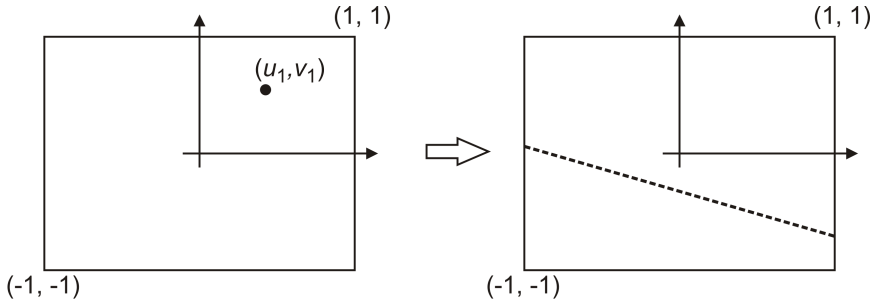


Figure 7: *Projected point in camera 1 (left) and possible locations of the same point in camera 2 (dotted line, right)*

Question 6: Parametric Density Estimation (30 pts.)

We often make the *parametric assumption* when estimating densities from data: we assume that the data distribution is well approximated by a parametric model $p(x; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the model (e.g. $\boldsymbol{\theta} = (\mu, \sigma)$ for the univariate Gaussian). Estimation of the distribution then reduces to estimation of the parameter $\boldsymbol{\theta}$. We consider two approaches: maximum-likelihood (ML) estimation and Bayesian inference.

a) Begin with a short summary of the ML method:

   1. Write down the formula for the likelihood of the data set $\mathcal{X}$. Explain what the i.i.d. assumption about $\mathcal{X}$ is, and why it is helpful to make this assumption in the formula for the likelihood.
   2. State the estimation principle of the ML estimator $\hat{\boldsymbol{\theta}}$ of the parameter $\boldsymbol{\theta}$, i.e. how is $\hat{\boldsymbol{\theta}}$ related to the data likelihood? **4 pts.**

b) Assume that the exponential distribution

$$p(x; \lambda) = \lambda e^{-\lambda x}$$

   with *rate parameter* $\lambda > 0$ is an adequate model for the data, which satisfies $x \geq 0$.

   1. Draw the graph of $p(x; \lambda)$ for $\lambda = 1$ in the interval $x \in [0, 4]$. A qualitative sketch suffices, but give the precise function value for at least two points in the domain of $p(x; \lambda)$.
   2. You are given a data set $\mathcal{X} = \{1, 2, 4\}$ which is exponentially distributed, but you don't know the exact value of the rate parameter. Find a visual representation of the data likelihood of $\mathcal{X}$ for $\lambda = 1$, and draw it into the graph above.
   3. Is the ML estimate $\hat{\lambda}$ for this dataset $\mathcal{X}$ larger or smaller than one? Argue using your visualization of the data likelihood and the functional form of the density. **4 pts.**

c) Derive the formula for the ML estimator $\hat{\lambda}$, given an i.i.d. exponentially distributed data set of size n:

$$X_1, \ldots, X_n \sim \text{Exponential}(\lambda).$$

   Be sure to write down every step of the derivation. Comment your calculations where necessary. **5 pts.**

d) Show that $\hat{\lambda}$ is a biased estimator of the true rate parameter λ:

$$\hat{\lambda} = \frac{n}{n-1}\lambda.$$

   To do this:

   1. Write down the definition of estimator bias as a formula and explain its meaning in one sentence.
   2. Now substitute $\hat{\lambda}$ (which you derived before) into the definition of estimator bias.

   The following facts will be useful:

   - Expectation of a function $g(X)$ of a random variable X: $\mathrm{E}[g(X)] = \int_{-\infty}^{\infty} g(x) p(x)\, dx$
   - Let $X_1, \ldots, X_n \sim \text{Exponential}(\lambda)$ be exponentially distributed and independent random variables, and let $S = \sum_{i=1}^{n} X_i$. Then $p(s) = \text{Gamma}(n, \lambda)$.

- The *gamma distribution* $\mathrm{Gamma}(\alpha, \beta)$ is defined as

$$p(x; \alpha, \beta) = x^{\alpha-1} \frac{\beta^\alpha e^{-\beta x}}{\Gamma(\alpha)}$$

  for $x \geq 0$ and $\alpha, \beta > 0$.
- The *gamma function* obeys $\Gamma(n) = (n-1)\Gamma(n-1)$.

Again, write down every step of your calculation and comment where necessary. **5 pts.**

We now turn to Bayesian estimation of λ given a data set $\mathcal{X}$, by introducing a prior distribution $p(\lambda)$ over the quantity of interest.

e) Bayes' rule relates the prior distribution and data likelihood to the posterior distribution:

$$p(\lambda|\mathcal{X}) = \frac{p(\mathcal{X}|\lambda)\, p(\lambda)}{p(\mathcal{X})}$$

Explain in words what each term in the formula is. **2 pts.**

f) Bayesian estimation of λ is conceptually different from maximum likelihood estimation, which is a frequentist method. State two such differences, one related to the prior and one related to the posterior. **2 pts.**

g) Computing the posterior analytically can be hard in general, but if we choose the right distribution for the prior it can be easy. Show that if the prior is gamma distributed

$$p(\lambda) = \mathrm{Gamma}(\alpha_0, \beta_0)$$

and the data likelihood $p(\mathcal{X}|\lambda)$ is defined as in c), then the posterior is again gamma distributed

$$p(\lambda|\mathcal{X}) = \mathrm{Gamma}(\alpha_n, \beta_n),$$

but with different parameters. To compute the parameters α_n and β_n:

1. Ignore multiplicative and normalization constants (which ensure that densities integrate to one).
2. Show that the posterior is proportional to a gamma distribution.
3. Infer the parameters by comparing your result for the posterior to the definition of the gamma distribution. **5 pts.**

h) Show that for large n (i.e. many data points), the maximum of the posterior converges to the ML estimator $\hat{\lambda}$:

1. Start with the fact that the gamma distribution attains its maximum value at $\frac{\alpha-1}{\beta}$.
2. Plug in the values α_n and β_n that you have obtained for the posterior distribution.
3. Take the limit $n \to \infty$ and compare to $\hat{\lambda}$. **3 pts.**