ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visual Computing
2007

Prof. M. Gross (Computer Graphics Lab)
Prof. J. M. Buhmann (Machine Learning Group)

R. Ziegler, B. Bickel, D. Steinemann
C. Sigg, W. Einhäuser

Final Exam - Summer 2007

01 September 2007

Last name: ____________________

First name: ____________________

ETH number: ____________________

Signature: ____________________

General Remarks

- At first, please fill in your last and first name and your ETH number and sign the exam. Place your student ID on the table for control.
- You have 3 hours for the exam. There are 7 questions. You can earn a total of 180 points. The maximum number of points for each question is listed on the right margin.
- 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.
- You can answer the questions in English or in German.
- Write as legible as possible. Do not use a pencil! You may provide at most one valid answer per question. Invalid solutions must be canceled out clearly.

	Topic	max. points	points	Visum
1	Anti-Aliasing	25		
2	Image Processing	30		
3	Variational Calculus & Diffusion	30		
4	Appearance	13		
5	Classification	30		
6	Projective Texturing and Shadow Maps	17		
7	Graphics Pipeline, Transformations & Projections	35		
Total		180		

Grade: ....

Question 1: Anti-Aliasing (25 pts.)

a) Temporal aliasing. Imagine you have an analog clock (a 12 hour clock face with a minute hand and an hour hand) and you want to produce a time-lapse video. Describe the perceived rotation of the minute and hour hand when sampling at a rate of 0.9 frames per hour. What sampling rate is necessary to prevent temporal aliasing? **2 pts.**

b) Name two aliasing effects that can occur when sampling a static scene and name the stages of the graphics pipeline that could cause those effects. **2 pts.**

c) Explain the occurrence of aliasing. Draw six plots to illustrate your explanation, showing the continuous signal, the sampling function, and the sampled signal in spacial AND temporal domain. Point out explicitly in one sentence what causes aliasing. **4 pts.**

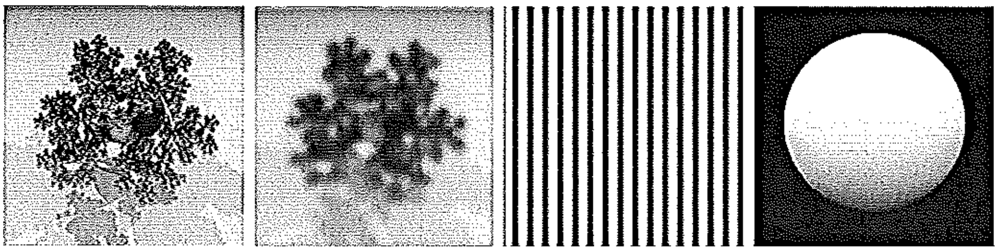
d) Strategies for anti-aliasing. Name two general strategies for anti-aliasing, and explain in terms of signal processing why they work. Given an arbitrary scene, can you guarantee that both strategies completely avoid aliasing? Explain your answer in 1-2 sentences. **3 pts.**

e) Ideal filter vs. useful filter. According to the sampling theory, the multiplication of a signal with a box function in frequency space would set all frequencies higher than a certain threshold to zero while keeping all the lower frequencies. Why is this filter not used in practice and what are its shortcomings? What filter would you use instead? Explain your decision in 2 sentences. **3 pts.**

f) Defocusing the beam of a cathode ray tube (CRT) (this technology is used in old TVs) is sometimes called "the poor man's anti aliasing". Can you very briefly explain why? What bad side effect does this technique has in terms of display capacity/resolution? **1 pts.**

g) Given are four images (a)-(d) in Figure 1. Find the matching fourier spectrum (i)-(iv) in Figure 2 for each of the images.

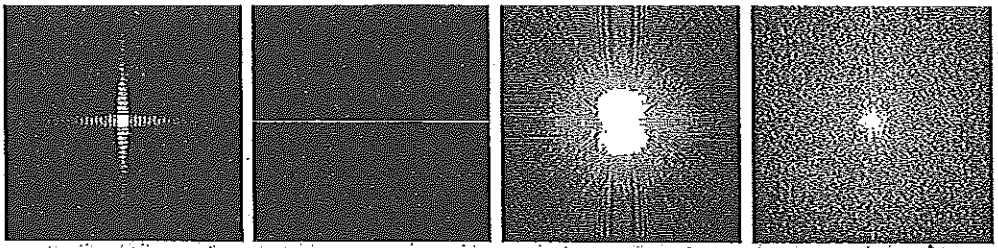
image	spectrum
(a)	( )
(b)	( )
(c)	( )
(d)	( )



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

Figure 1: *Images*

2 pts.



(i) (ii) (iii) (iv)

Figure 2: *Fourier spectra*

h) An image processing package allows the user to design 3×3 convolution filters. Design 3×3 filters to perform the following tasks:

(a) Uniform blurring

(b) Edge detection of horizontal edges

Explain how the filters work, using the following image as an example (evaluate your filter on this example, you may round off calculated values to the nearest integer). Assume circular boundary conditions, which means that the intensity at pixel position (x, y) is $I(x, y) = I(x \mod 6, y \mod 6)$. The upper left corner has position $(0, 0)$.

100	100	100	100	100	100
100	100	0	0	0	100
100	100	0	0	0	100
100	100	0	0	0	100
100	100	100	100	100	100
100	100	100	100	100	100

6 pts.

i) Anisotropic Filtering. Describe briefly what anisotropic filtering means and given an example where it is useful to apply such a filtering. Is anisotropic filtering supported by OpenGL? **2 pts.**

Question 2: Image Processing (30 pts.)

Hints:

$$sinc(x) = \begin{cases} \frac{\sin(x)}{x} & \forall x \neq 0 \\ 1 & x = 0 \end{cases}$$

The Fourier transform of $sinc(x)$ is $f(k) = \begin{cases} 1 & \forall |k| < \frac{1}{2} \\ 0 & otherwise \end{cases}$

The Fourier transform of a Gaussian with standard deviation σ is again a Gaussian with standard deviation $\frac{1}{\sigma}$.

a) **Filter** **total: 12 pts.**

You build an imaging device that scans the input in one-dimensional lines. Let the input be $f(x)$ and the corresponding output of your device $g(x)$. $\mathcal{T}$ is the operator that describes the transformation your device performs, i.e. $g(x) = \mathcal{T}f(x)$.

i) $\mathcal{T}$ is linear and shift-invariant. What does this mean, i.e. which conditions are fulfilled? **3 pts.**

ii) You probe $\mathcal{T}$ with a dot at x=0. The dot has infinitely small size and unit integrated brightness ("a δ-peak"). You measure the output of your system and find it to be $h(x) = sinc(x)$. Now you show your system two different images that are modulated sinusoidally:

- $f_1(x) = \sin(\frac{x}{50})$
- $f_2(x) = \sin(50x)$

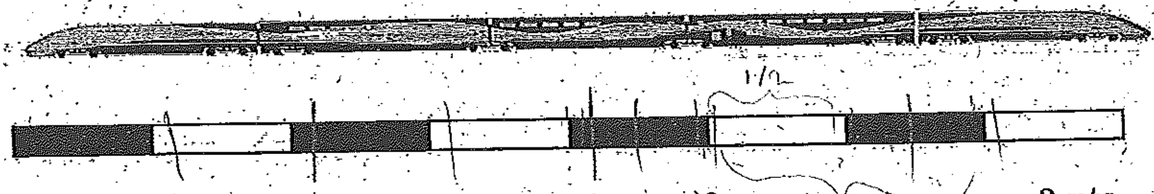
What is the output of the system, i.e. $\mathcal{T}f_1(x)$ and $\mathcal{T}f_2(x)$? **6 pts.**

iii) What happens if you show f_1 and f_2 superimposed, i.e. what is $\mathcal{T}(f_1 + f_2)$?
Hint: You can give an answer, even if you could not solve the previous part. **1 pt.**

iv) In words, what operation does you system (the transformation $\mathcal{T}$) perform? **2 pts.**

b) **Blur** **total: 5 pts.**

Now you build a new imaging device that captures your image with negligible distortion, i.e. g(x)=f(x) for static images to a good approximation. However, your device has a finite shutter time of 10ms and you use it out of the side windows of the new TGV driving at a constant speed of 360 km/h parallel to your scanline. At each pixel of your device, brightness is averaged over one exposure. Consequently your image gets blurred.



i) What is the blurring kernel (use units of meters)? **2 pts.**

ii) How will a dashed line with alternating black and white segments of L=100m be captured by your device? **2 pts.**

iii) How small can L get before your device records a uniform gray line? **1 pt.**

c) Noise **total: 13 pts.**

Your third device is described by the following equation:

$$g(x) = (f * \gamma)(x) + \eta(x) \tag{1}$$

where f is the original image, g the measured image as above; η is noise, which is uncorrelated to the signal; $*$ denotes convolution and γ Gaussian blur:

$$\gamma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{2}$$

i) First assume a case without noise, i.e. $\eta(x) = 0$. You measure $g(x)$. Provide a formula to reconstruct $f(x)$. **4 pts.**

ii) Is the reconstruction of $f(x)$ perfect? Why or why not? **2 pts.**

iii) Compute (the reconstruction of) $f(x)$ explicitly for a constant $g(x) = c$. **1 pt.**

iv) Explain the dependence of $f(x)$ on σ.
Hint: Think of the physical reason for blurring, if σ models the width of individual sensor elements. **1 pt.**

v) Now we add white noise, i.e. noise with a uniform spectrum. Which frequencies are affected most? Argue in terms of signal-to-noise ratio. **3 pts.**

vi) What are potential problems with extremely large or extremely small σ?
Hint: You can provide intuitive arguments or use the results of the previous questions. **2 pts.**

Question 3: Variational Calculus (30 pts.)

a) Data Smoothing **total: 15 pts.**

Hint: You can earn points for intuitive solutions to parts iv) and v), even if you could not solve ii) and/or iii) completely.

You measure noisy one-dimensional data, your measurement is denoted by $g(x)$, where $x \in \left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$. You want to approximate the real data g by a smooth function $f(x)$, which faithfully represents the data. The objective is thus to minimize the quadratic error of f with respect to the measurement g. This objective is tradeoff against a smoothness constraint, which we here enforce by minimizing the integrated squared gradient of f.

i) Write down the functional to be optimized. Denote the weighing between smoothness and faithfulness to the data by the parameter λ (larger λ shall imply more smoothness). **2 pts.**

ii) Which condition has to hold for the extremum of this functional? Write down the equation and its name. **2 pts.**

The data you measure is an oscillator of frequency ω_1, which is corrupted by its odd higher harmonics $\omega_j = (2j-1)\omega_1$

$$g(x) = \sum_{j=1}^{n} a_j \cos(\omega_j x) \tag{3}$$

with $a_1 = 1$ and $a_j < 1$ for $j > 1$. For simplicity we set $\omega_1 = 1$, so

$$g(x) = \sum_{j=1}^{n} a_j \cos((2j-1)x) \tag{4}$$

iii) Show that with this g(x)

$$f(x) = \sum_{j=1}^{n} \mu_j \cos(\omega_j x) \tag{5}$$

is a solution of the differential equation in ii). Express the constants μ_j in terms of a_j, λ and ω_j. **5 pts.**

iv) In the limit of small λ ($\lambda \to 0$), what is μ_j? Why would you expect this result even without a (detailed) calculation? **3 pts.**

v) Discuss qualitatively, how the relation between a_j and μ_j depends on ω_j: Which frequencies are damped particularly? Give an intuitive explanation (e.g., which part of g is "smoothest"?) for this dependence. **3 pts.**

b) Diffusion **total: 15 pts.**

i) What do Fick's law and the continuity equation state? How do you obtain the Diffusion equation? **2 pts.**

ii) Now consider linear, isotropic diffusion in two dimensions. What is the Diffusion equation for this special case? **1 pt.**

iii) For simplicity we now consider linear diffusion in one dimension. Assume $u(x, t)$ obeys the linear, isotropic Diffusion equation with diffusion constant D. As boundary condition, assume that $u(x, 0)$ is given by a one-dimesional image $f(x)$. Show that for $t > 0$, $u(x, t)$ is given by a convolution of f with a Gaussian kernel. How does the standard deviation (width) of the kernel depend on Diffusion constant D and time t? *Hints: The spatial Fourier transform of a spatial derivative* $\frac{d}{dx}$ *corresponds to a multiplication with* ik_x *in the Fourier domain, where* i *is the imaginary unit and* k_x *the spatial frequency in direciton* x. *The temporal derivative remains unaffected under the spatial transform, i.e.* $\mathcal{F}\left[\frac{du}{dt}\right] = \frac{d\mathcal{F}[u]}{dt}$. *The Fourier transform of a Gaussian of width* σ *is a Gaussian of width* $\frac{1}{\sigma}$. *The solution of the differential equation of the form* $\frac{du}{dt} = -Dt$ *has the form* $u(t) = A\exp(-\frac{t}{\tau})$ *with amplitude* A *and time constant* τ. **6 pts.**

iv) What is the main idea underlying non-linear diffusion for image processing? Give an example for a diffusion constant D that implements this idea. *Hint: You can use Perona-Malik diffusion as an example.* **3 pts.**

v) Describe 2 approaches to extend diffusion to color images (e.g., RGB images). What are the advantages and disadvantages of the methods you propose? **3 pts.**

Question 4: Appearance (13 pts.)

a) Can a highlight be produced inside a triangle by using Gouraud Shading? Why? **2 pts.**

b) Explain in 1-2 sentences the difference of an isotropic and an anisotropic BRDF. **2 pts.**

c) Describe in 2-3 sentences how you would achieve a metallic looking surface. **3 pts.**

d) The Phong model gives the possibility to introduce ambient, diffuse and specular components into the illumination of a model. One very typical component is missing for rendering the appearance of skin. Which component is missing? **1 pts.**

e) Which difference can be seen if you compare a rendering with vs. one without this additional component? **1 pts.**

f) Which are the additional parameters of the additional component ? **2 pts.**

g) Imagine you would have to create an application, which is simulating a walk through a museum, where the appearance of the paintings should be represented correctly. Why would a color representation by an RGB triplet probably not be sufficient ? **2 pts.**

Question 5: Classification

a) **Perceptron I** **total: 3 pts.**

Rosenblatt's (1959) Perceptron employs the following neuron (brain cell) model, which was proposed earlier by McCullogh & Pitts (1943): For a given input vector $\mathbf{x} \in \mathbb{R}^n$, synaptic weights $\mathbf{w} \in \mathbb{R}^n$, and threshold $b \in \mathbb{R}$ it outputs 1 whenever $\sum_{i=1}^{n} w_i x_i \geq b$ and 0 otherwise.

i) Assume just two input dimensions, x_1 and x_2, and weights $w_1 = 1$ and $w_2 = -2$; $b = 0.5$. Characterize (sketch) the part of the input space, for which the neuron outputs 1. **2 pts.**

ii) You want to express the behavior of the same neuron by an equation of the form $\sum_{i=1}^{m} w_i x_i \geq 0$, how could you do so? *Hint: $n \neq m$ is allowed.* **1 pt.**

b) **Minsky and the Perceptron** **total: 7 pts.**

In 1969, Marvin Minsky & Seymour Papert criticized the Perceptron of not being able to learn an XOR funtion.

i) Explain what is meant by this statement. **2 pts.**

ii) Can a Perceptron represent other Boolean functions? If so, provide the weights for $AND(x_1, x_2, x_3)$ and $NOT(x_1)$; if not, explain why. **2 pts.**

iii) Can you combine multiple perceptrons to represent XOR? **2 pts.**

iv) Does this invalidate Minsky's criticism? **1 pt.**

c) **Perceptron III** **total: 9 pts.**

Now the domain of your classification problem has two real-valued dimensions (x_1, x_2). You know the following for certain:

- Your problem is linearly separable.
- Datapoints (1,2), (2,1) and (3,1) belong to class 1.
- Datapoint (4,4) belongs to class 2.

Draw one or more figures when answering the following questions (no explicit formulas are necessary in 5c).

i) Characterize all datapoints that are certainly in class 1 and those that are certainly in class 2. **2 pts.**

ii) What is the set of datapoints for which you cannot decide based on the above knowledge alone? **1 pt.**

iii) You are forced to make a decision on all points of the plane. Which principle could you use? Why would you use it? And what is the decision boundary between class 1 and 2 then (characterize geometrically)? *Hint: Consider for example point (3.9,3.9). Where would you intuitively place it?* **3 pts.**

iv) Which datapoint(s) could you remove from your training set, without any effect on the classification? What are the other points of the training set called? **2 pts.**

iv) You get a new training set:
   - Class 1: (6,5), (7,12), (10,25)
   - Class 2: (-10,-3), (-7,-6), (-3,0)

   You start with the decision boundary of part iii) and perform a single iteration of the perceptron learning rule with the new training set. How does the decision boundary change and why? **1 pt.**

d) **Support Vector Machines (SVMs)** **total: 11 pts.**

i) What are the 3 fundamental restrictions of a classical (2-layer) perceptron, and how are they overcome by SVMs? **3 pts.**

ii) Which of the following functions are valid, positive semidefinite, kernels? Explain why $(\mathbf{x}, \mathbf{y} \in \mathbb{R}^2)$. **4 pts.**
   - $k_1(\mathbf{x}, \mathbf{y}) = (x_1 y_2 + x_2 y_1)$
   - $k_2(\mathbf{x}, \mathbf{y}) = -10(x_1 y_1 + x_2 y_2)$
   - $k_3(\mathbf{x}, \mathbf{y}) = \cos(x_1 - 10x_2)\cos(y_1 - 10y_2)$
   - $k_4(\mathbf{x}, \mathbf{y}) = \exp\left(\exp\left((x_1 y_1 + x_2 y_2)^{16}\right)\right) + 15(x_1 y_1 + x_2 y_2) + 10^{(x_1 y_1 + x_2 y_2)}$

iii) A toy example: Your company produces rectangular Lego bricks of different lengths (x) and widths (y). The marketing department decides to sell some of them in circular boxes of radius 1. Your job is to build a classifier that decides - knowing x and y - whether a given brick fits into this box (class 0) or not (class 1). On the shelf you have a two layer perceptron with 3 inputs z_1, z_2, and z_3, weights w_1, w_2, and w_3, bias b, and output c. You can set each z_i to be any algebraic funciton of x and y. Provide a choice for w_i, z_i, and b that solves this task. **4 pts.**

Question 6: Projective Texturing and Shadow Maps(17 pts.)

a) In which case do you apply projective texturing ? Name three different scenarios. **1 pts.**

b) Why should texture coordinates be generalized to 4D coordinates, if the texture is only 2D? **2 pts.**

c) Describe the four necessary steps to obtain the texture coordinate for projective texturing. **2 pts.**

d) Provide the matrices necessary for projective texturing as shown in Fig.3, where the projective texture has a field of view of 90° (visualized as dashed lines). The light is located at position **P**. All coordinates are given relative to the world-coordinate system with origin **W**$= (0, 0)$. The texture coordinates inside the field of view should lie between 0 and 1. **5 pts.**

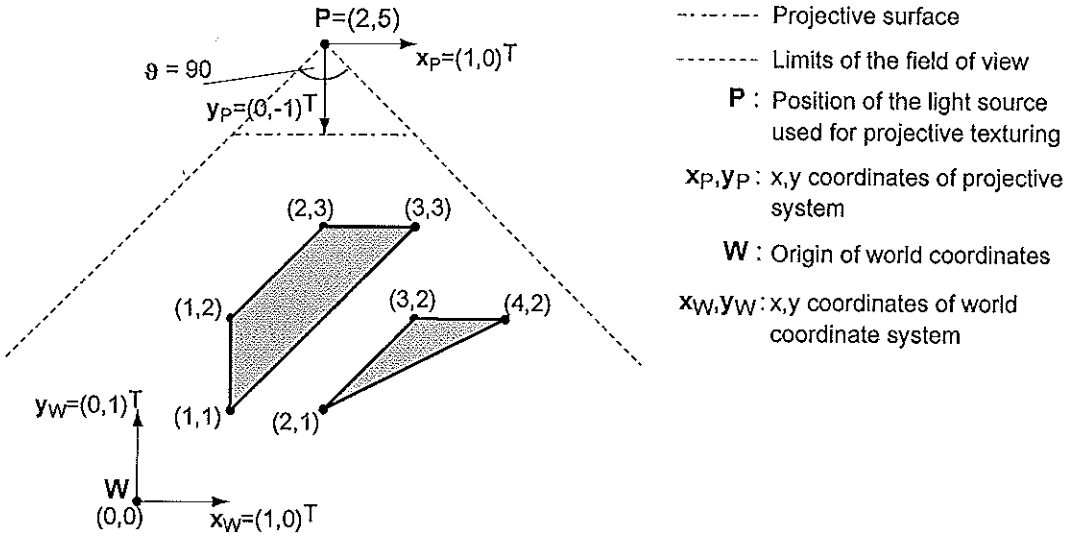


Figure 3: This figure shows a scene with two polygons onto which a projective texture has to be put with the light shining from position **P**.

e) For how many points would OpenGL generate a texture coordinate? **1 pts.**

f) Calculate texture coordinates for two points of the scene. Provide the coordinates of the point as well as its texture coordinate. **2 pts.**

g) Describe shadow mapping in 3-4 sentences. **3 pts.**

h) What can cause problems when using shadow mapping and what can you do to minimize these problems? **2 pts.**

Question 7: Graphics Pipeline, Transformations & Projections (35 pts.)

a) Graphics Pipeline and OpenGL **total: 7 pts.**

i) The graphics pipeline can be roughly divided into vertex processing and fragment processing. Explain both using just a few keywords, and give one specific example from the classical graphics pipeline for each. **2 pts.**

ii) What two types of graphics APIs are there? Describe each type very briefly and give an example API for each. **1 pts.**

iii) What is GLUT and what is it used for? **1 pts.**

iv) Give two examples of geometric primitives used by OpenGL. What image primitives does OpenGL work with? **2 pts.**

v) "OpenGL is a state machine": Briefly explain what this means. **1 pts.**

b) Transformations **total: 15 pts.**

i) Explain in 3 sentences what the model-view matrix is used for, what transformations it contains, and how it is generated. **2 pts.**

ii) Explain homogeneous coordinates. Why are they used? **2 pts.**

iii) A tetrahedron lies in some 3D space. The scene is described by local object coordinates $(x_{obj}, y_{obj}, z_{obj})$ and world coordinates (x, y, z). The origin of the local coordinate system lies at $\mathrm{t}_0 = (1, -4, 2)^T$ and the orientation and scaling of both coordinate systems are the same. Point P is defined as some corner of the tetrahedron. Please refer to Figure 4.

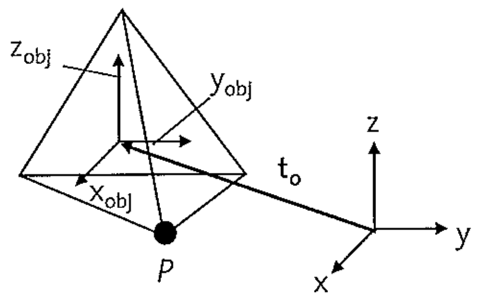


Figure 4: A tetrahedron in 3D. The scene is described by local object coordinates and world coordinates.

Assume that P is given in world coordinates $\mathrm{p}_w = (2, -3, 0)^T$. First, we want to rotate the whole tetrahedron by 90 degrees around it's z-axis (z_{obj}), and then we want to scale it by 2 along its x-axis (x_{obj}). Define appropriate rotation and scaling matrices and use them to compute the transformed point in world coordinates. Use homogeneous coordinates if possible. **7 pts.**

iv) To rotate a point around an axis, a more elegant solution is to use quaternions. Assume now that the point $\mathbf{p}_w = (2, -3, 0)^T$ from question 7.b.iii is rotated by 45 degrees around the local axis $\mathrm{N} = (1, 1, 0)^T$. The axis is defined in the tetrahedron's local coordinate system, while the rotated point should again be given in world coordinates.
Set up the quaternion expression for this rotation. Explain all variables involved and explicitly state their values. You do *not* have to evaluate the final quaternion expression. **4 pts.**

c) **Projections** **total: 13 pts.**

Assume that a point P_c has been transformed into camera coordinates, resulting in the point $\mathbf{p}_c = (4, 6, 8)$. As we eventually want to see it on screen, it must be projected into the camera's projection plane.

i) Assume that the projection plane is located at $z = 2$ in the camera coordinate system C, and that the center of projection lies in the origin of C. Refer to Figure 5 for a schematical explanation. Define the corresponding homogeneous projection matrix and use it to compute the projected point $\mathbf{p}_p$.

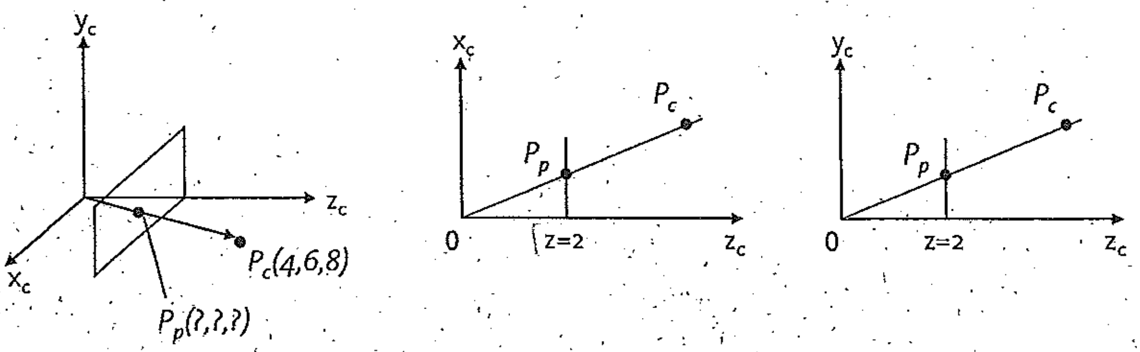


Figure 5: Projecting a point onto the camera's projection plane.

4 pts.

ii) What is the difference between an orthographic and an oblique projection? What is common to these projections? **1 pts.**

iii) State one OpenGL command to define a perspective projection and another command with which the camera position can be set. For each command, state on which matrix stack it operates on. **2 pts.**

iv) After projecting P onto the projection plane, one last transformation is required to obtain the final screen coordinates. What is it called? **1 pts.**

v) Let's assume that the screen (u, v) is defined by coordinates $(0,0)$ to $(500,400)$ and that the corresponding window (x_c, y_c) on the projection plane is defined by $(-10,-10)$ to $(10,10)$, as depicted by Figure 6. The projected point is given as $\mathbf{p}_p = (7,3)^T$. Compute the linear transformation $\mathbf{T}$ in

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix}$$

and use it to compute $\mathbf{p}_{screen}$. Note that the two coordinate systems have different orientation. **5 pts.**

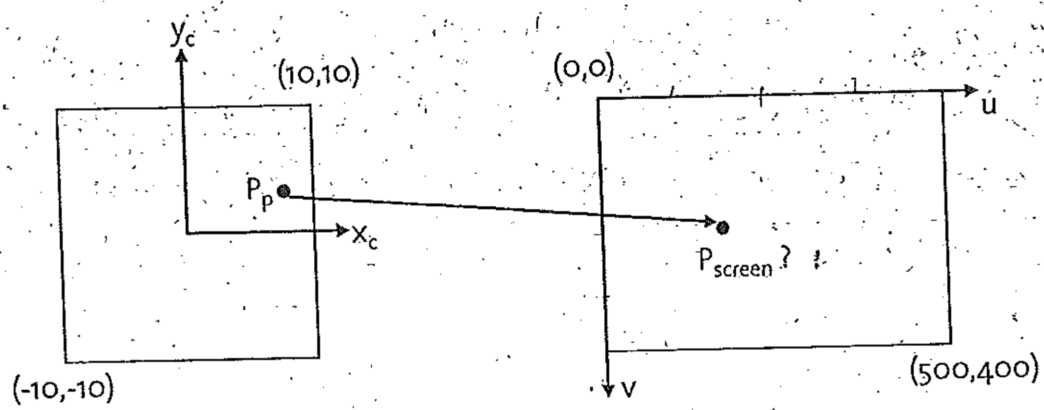


Figure 6: Mapping a point into screen coordinates.