



Klausur Wintersemester 2017/2018, Fragen und Antworten

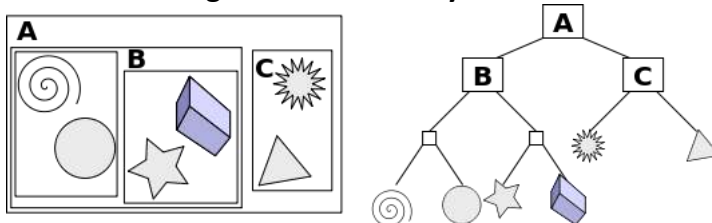
Graphische Datenverarbeitung (Eberhard Karls Universität Tübingen)

Klausurfragen GDV WS 17/18

1) Spatial Acceleration Structures

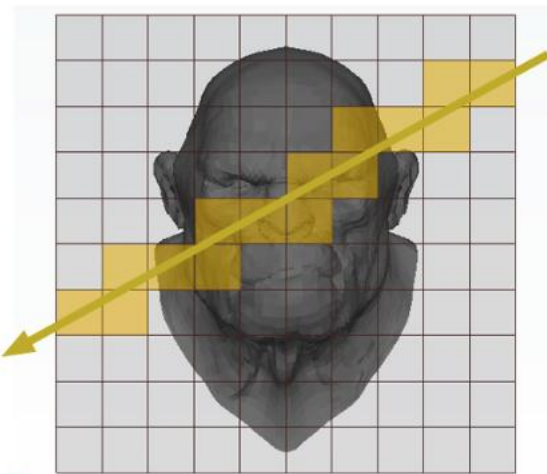
Ray tracing an unorganized polygon soup is rather slow. Describe three different spatial acceleration structures and explain their benefits and drawbacks with regard to traversal complexity, building time and space requirements. (3-4 sentences for each).

a) BVH – Bounding volume hierarchy



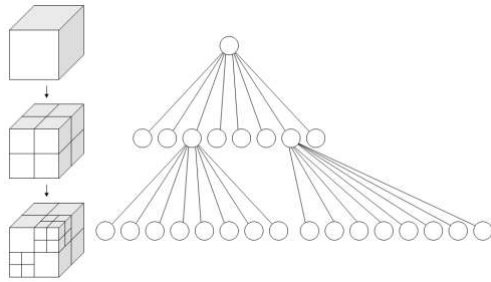
+	-
<ul style="list-style-type: none"> - Adaptiv - Gut zu traversieren → $O(\log n)$ 	<ul style="list-style-type: none"> - Wahl der Bounding Volumes - Einfügen von Objekten

b) Grid



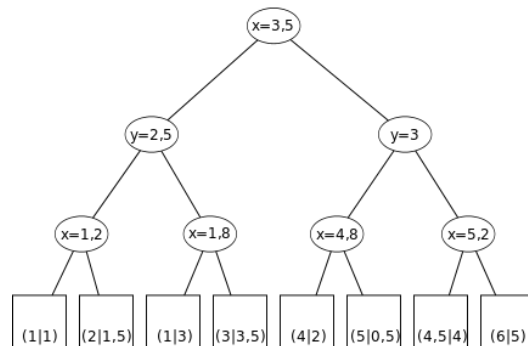
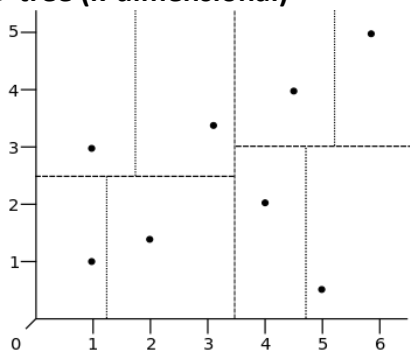
+	-
<ul style="list-style-type: none"> - Triviale Insertion - Hardware freundlich 	<ul style="list-style-type: none"> - Relativ teuer zu traversieren, da viele leere Felder durchquert - Memory consumption ($O(a^k)$) - Auflösung wählen (→ octree)

c) Octree/Quadtree



+	-
<ul style="list-style-type: none"> - Adaptiv - Build $up \in O(n) \rightarrow$ one-time (Einfaches einfügen) 	<ul style="list-style-type: none"> - Traversal $\in O(k \log n)$

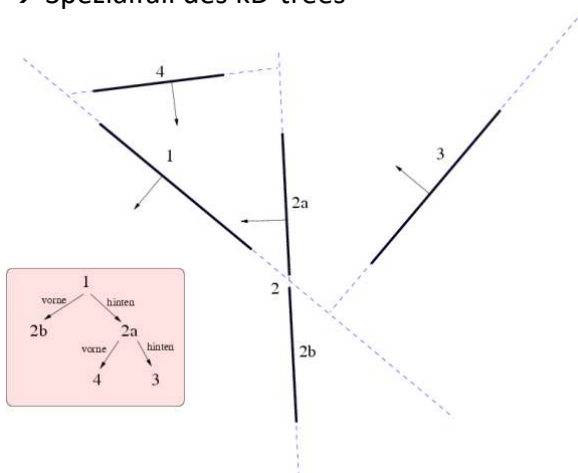
d) kD-tree (k-dimensional)



+	Laufzeiten
<ul style="list-style-type: none"> - Balancierbar mit Median - Traversal günstig 	<ul style="list-style-type: none"> - Construct: $\in O(n(k + \log n))$ - Traversal: $\in O(n^{1-\frac{1}{k}} + a)$ - Memory: $O(k * n)$

e) BSP-tree (Binary Space Partitioning)

→ Spezialfall des kD-trees



-
<ul style="list-style-type: none"> - Wo hört man auf zu teilen?

2) Rendering Equation

a) Complete the formula of Rendering Equation for surface models:

$$L(x, w_o) = L_e(x, w_o) + \int_{\Omega} f(x, w_o, w_i) * L_i(x, w_i) * \cos \theta_i * dw_i$$

Emitted radiance

→ Ausgesendete Strahlung der Oberfläche (ungleich 0 nur für licht-quellen)

BRDF (Bidirectional reflectance distribution function)

→ Reflexionsverhalten von Oberflächen

Cosine shading angle

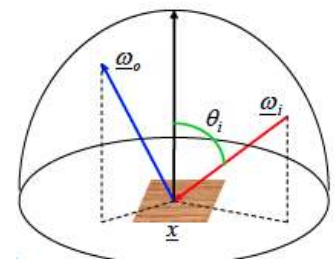
Einfall d. Strahlung von allen Richtungen

Integral over Hemisphere

Jeder mögliche Strahl

Incoming radiance

Einfallende Strahlung



b) How does the rendering equation account for global illumination effects?

- Illuminated Object Surface: Durch Radiosity & Irradiance geht Lichtkraft verloren
- Invariance of radiance in free space (Unverändlichkeit von Radiance im freien Raum)
- Total radiance = emitted radiance + reflected radiance

c) Which assumption is made for rendering equation? How can the rendering equation be simplified because of that assumption?

- Radiosity instead of Radiance
- Use reflectance instead of BRDF
 - 1) constant reflectance
 - 2) Radiosity instead of Radiance
- Final form: $B(x) = B_e(x) + P(x) \int_{y \in S} F(x, y) B(y) dA_y$
 $B_e(x)$ → Radiosity instead of Radiance
 $P(x)$ → Reflectance instead of BRDF

3) BRDF – Bidirectional Reflectance Distribution Function)

Which of these properties does a BRDF need to fulfill to be physically valid?

Conservation of energy	True
Wavelength limited	False
Helmholtz reciprocity	True
Can be negative	False
The unit is $1/m^2$	False
The unit is $1/sr$	True
The diffuse BRDF is constant	True
$\int_{\Omega} f_r(w_i, x, w) \cos \theta_i dw_i \leq 1$	True

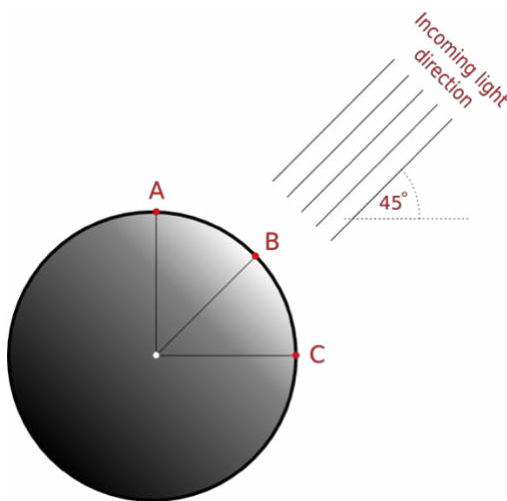
4) Path Tracing

For some of the following scenes, a path tracer will definitely produce black images.
Which ones?

A pinhole camera model is used	Depends
All light sources are point lights	Black
All light sources are directional lights	Black
All light sources are area lights	Depends
All light sources are area lights behind glass	Depends
All surfaces which are not light sources are mirrors	Depends
All surfaces which are not light sources are white	Depends
All surfaces which are not light sources are black	Black

5) Shading with a Directional Light Source

Compute the irradiance of a directional light source on a sphere for three points A, B and C, with the incoming radiosity (from a surface perpendicular to the light direction) being L_s .



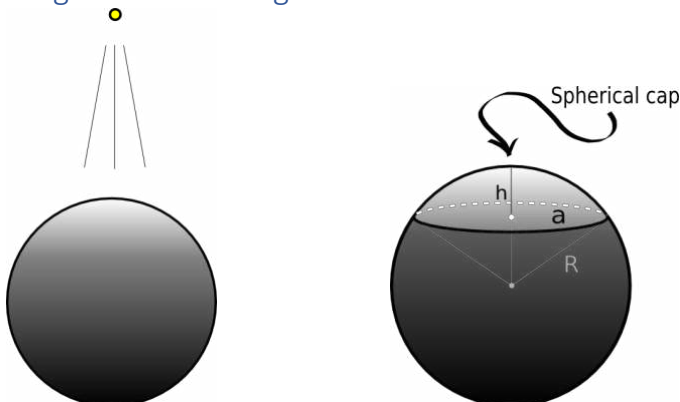
$$E_{A,B,C} = \int_{\Omega} L_s * \cos(w) dw$$

$$E_{A,C} = L_s * \cos \frac{\pi}{4} = L_s * \frac{\sqrt{2}}{2}$$

$$E_B = L_s * 1$$

Da man nur eine Richtung des einkommenden Lichts hat, wird keine Hemisphäre benötigt und das Integral fällt weg.

6) Shading with a Point Light Source



A point light source illuminates a sphere with radius r , as illustrated in the picture. The distance of the light source to the center of the sphere is d . Given the light source ϕ_s compute the total radiant power incident on the sphere

Note:

- Do not compute any integral.
- Pythagoras says: $a^2 + b^2 = c^2$ for a rectangular triangle.
- The height h of a rectangular triangle can then be computed by $h * c = a * b$.
- The area of a spherical cap is given by $S = \pi(a^2 + h^2)$.

$$A_{cap} = \pi(a^2 + h^2)$$

$$A_L = 4\pi r^2 = 4\pi L^2$$

$$\rightarrow L = \sqrt{d^2 - r^2} \rightarrow \text{Pythagoras}$$

$$\rightarrow \sin(\alpha) = \frac{r}{d}, \sin(\alpha) = \frac{a}{L}$$

$$\rightarrow a = \frac{r * L}{d}$$

$$\rightarrow h = L - x, x = \sqrt{L^2 - a^2} \rightarrow L = L_s * \frac{|d| * \sqrt{d - r} * \sqrt{r + d} + d^2}{2d^2}$$

7) Sampling theory / Bayer Pattern

Most affordable color cameras use a color filter array in front of the image sensor so that each sensor pixel captures only one of the three primary colors. The most popular pattern is the tiling by Bayer (1975), which is depicted below.

\rightarrow a \leftarrow

G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Using a sensor with pixel size a and Bayer color filter, what is the highest spatial frequency for each color channel individually that can be captured without aliasing (in every direction)?

$$F_{\text{sample}} = 2/a, \text{ Frequenz} = 1/2a$$

$$\rightarrow \text{Nyquist: Frequenz}_{\text{sample}} > 2 F_{\text{signal}}$$

$$\rightarrow F_{\text{signal}} < F_{\text{sample}}/2 = 1/4a$$

$$T < \frac{1}{2 f_{\max}} \Leftrightarrow 2a < \frac{1}{f_{\max}} \Leftrightarrow 4a < \frac{1}{f_{\max}} \Leftrightarrow \frac{1}{4a} > f_{\max} \rightarrow \text{Für Rot und Blau}$$

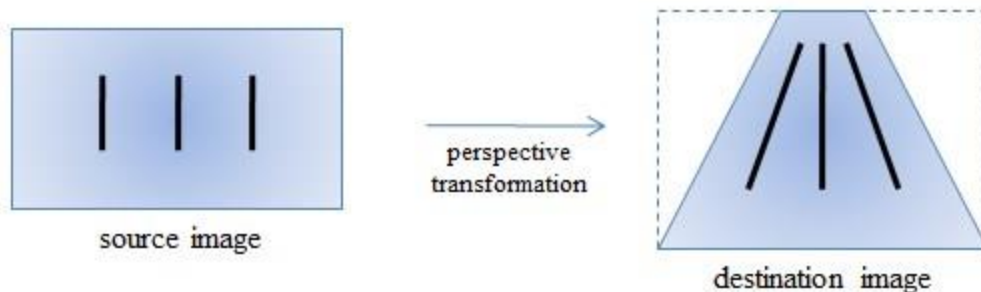
$$\text{Für Grün: } \frac{1}{2a} > f_{\max}$$

8) Affine Transformations

a) *Affine Invariance: Explain what affine invariance means and give a simple example.*

- Kollinearität: Die Bilder von Punkten, die auf einer Geraden liegen, liegen wieder auf einer Geraden.
- Parallelität: Die Bilder zweier paralleler Geraden sind parallel, wenn keine der beiden Geraden auf einen Punkt abgebildet wird.
- Teilverhältnisse: Drei verschiedene Punkte, die auf einer Geraden liegen (kollineare Punkte), werden so abgebildet, dass das Teilverhältnis ihrer Bildpunkte mit dem der Urbildpunkte übereinstimmt (es sei denn: alle 3 werden auf denselben Bildpunkt abgebildet).

b) *Perspective Transformation: Is the perspective projection an affine transformation?*



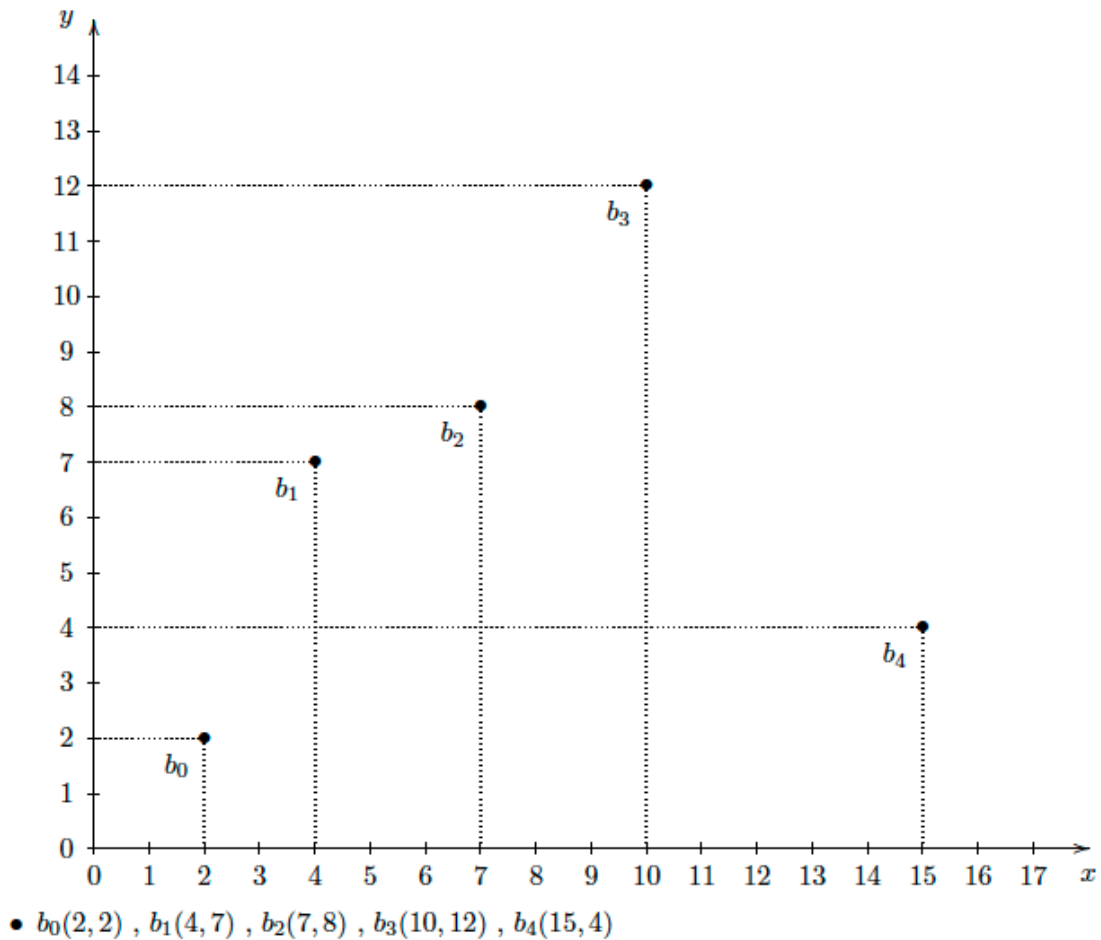
Nein, da Parallelität (& Teilverhältnisse) nicht mehr gelten.

9) Splines

a) *How many times can a spline of degree n be differentiated? Assume spline segments to be polynomials of degree n and the joints between the segments to be as smooth as possible.*

Splines mit Grad n können $n-1$ mal stetig differenziert werden.

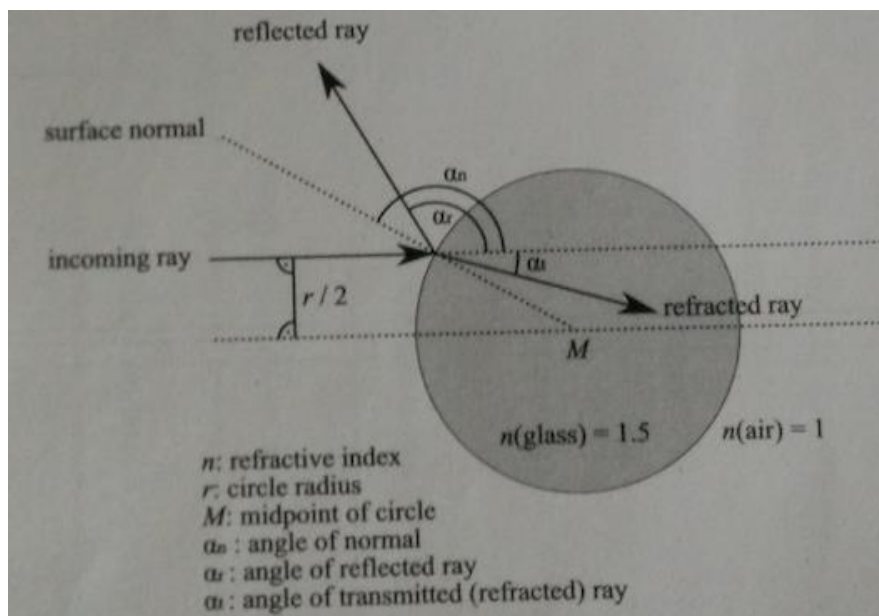
b) *Given five control points b_0, \dots, b_4 of a Bezier curve $P(t)$ with degree 4, $t \in [0, 1]$, as depicted below. Apply the deCasteljau algorithm graphically and numerically to compute the point $P_{0,5} = P(t=0,5)$ on the curve.*



10) Geometry

Given is a 2D sphere (circle) that is being hit by a light ray as depicted in the figure below. Calculate the following angles exactly with respect to the initial ray direction:

- Give the result in symbolic form using $\arcsin(\dots)$ etc.
- The distance of the ray from the sphere diameter (eccentricity) equals to half the sphere radius.
- For the refracted ray, use Snell's law and the given refractive indices of the materials.
- $\sin(30^\circ) = \frac{1}{2}$



a) The angle of the surface normal ray, α_n

$$\alpha_n = 180^\circ - \arcsin\left(\frac{r}{\frac{r}{2}}\right) = 180^\circ - \arcsin\left(\frac{1}{2}\right) = 180^\circ - 30^\circ = 150^\circ$$

b) The angle of the reflected ray, α_r

$$\alpha_r = 180^\circ - 2 \arcsin\left(\frac{r}{\frac{r}{2}}\right) = 180^\circ - 2 * 30^\circ = 120^\circ$$

c) The angle of the refracted ray, α_t

$$n_i(\lambda) \sin \theta_i = n_t(\lambda) \sin \theta_t$$

$$n(\text{air}) * \sin(30^\circ) = n(\text{glass}) * \sin(\alpha_t)$$

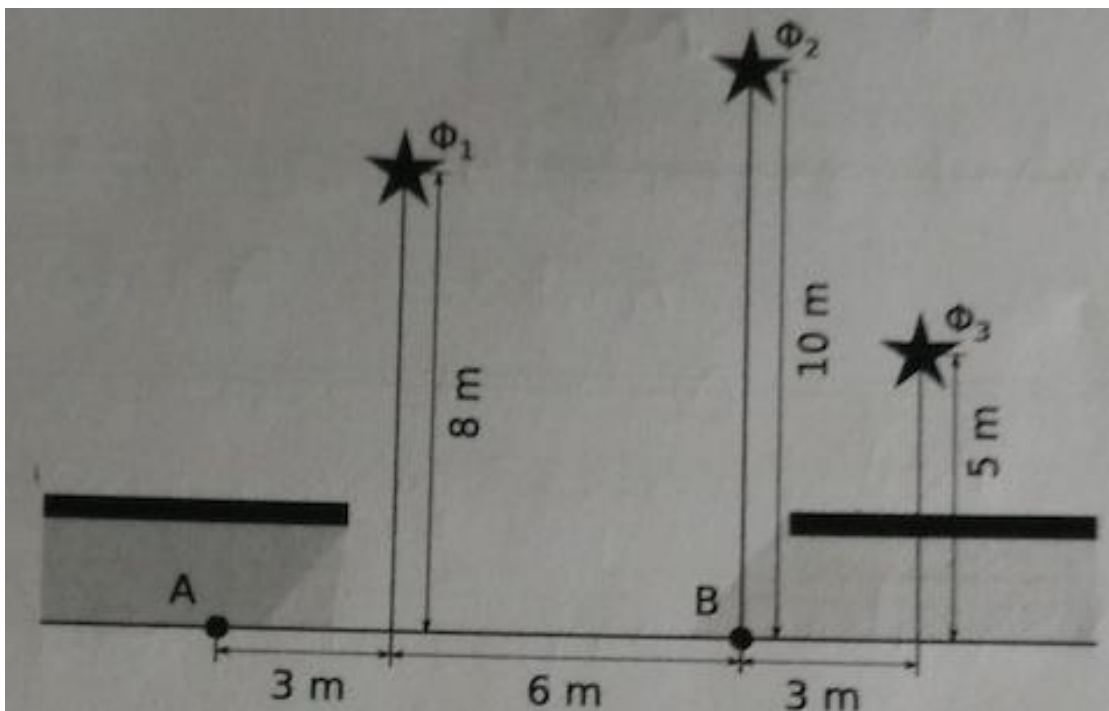
$$\sin(\alpha_{t_2}) = \frac{n(\text{air}) * \sin(30^\circ)}{n(\text{glass})} = \frac{0,5}{1,5}$$

$$\alpha_{t_2} = \arcsin\left(\frac{0,5}{1,5}\right) = 19,47^\circ$$

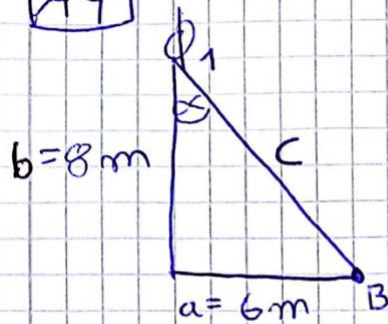
$$\alpha_t = 30^\circ - \alpha_{t_2} = 10,53^\circ$$

11) Radiometry

Given is a 3D scene as sketched below from the side. The scene contains three light sources with their radiant powers ϕ_1, ϕ_2 and ϕ_3 , a ground plane as well as some occluders. Calculate the irradiances at the points A and B. Do not take indirect lightning into account.



A4



$$a^2 + b^2 = c^2$$

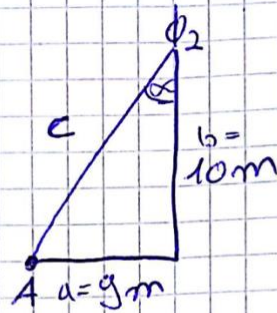
$$36 + 64 = c^2$$

$$\underline{10 = c}$$

$$\sin \alpha = \frac{a}{c}$$

$$\sin \alpha = \frac{6}{10} \quad \arcsin \quad \underline{\underline{\cancel{0.6} \rightarrow 36.87^\circ}}$$

$$\underline{\alpha = 36.87^\circ}$$



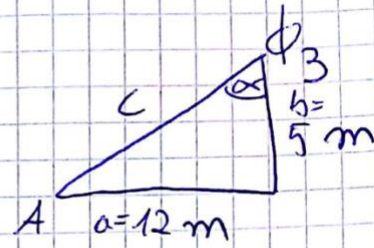
$$a^2 + b^2 = c^2$$

$$81 + 100 = c^2$$

$$13.45 = c$$

$$\sin \alpha = \frac{a}{c}$$

$$\underline{\alpha = 64.16^\circ}$$



$$a^2 + b^2 = c^2$$

$$144 + 25 = c^2$$

$$13 = c$$

$$\sin \alpha = \frac{a}{c}$$

$$\sin \alpha = \frac{12}{13}$$

$$\alpha = 66.93^\circ$$

$$E_A = \phi_2 \cdot \cos(64.16^\circ)$$

$$E_A = \phi_3 \cdot \cos(66.93^\circ)$$

$$E_B = \phi_1 \cdot \cos(36.87^\circ)$$

$$E_B = \phi_2 \cdot \cos(0^\circ)$$

$$E_A = \phi_2 \cdot \cos(64.16^\circ) + \phi_3 \cdot \cos(66.93^\circ)$$

$$E_B = \phi_1 \cdot \cos(36.87^\circ) + \phi_2 \cdot \cos(0^\circ)$$

12) Mip-Mapping

a) Compute the mip-map of the following 4x4 image.

7	11	0	0
9	5	0	12
7	2	5	4
1	10	4	3

Mean-Filter: 2x2 Filter nehmen & Mittelwert berechnen.

- $7+11+9+5 = 32 \rightarrow 32/4 = 8$
- $0+0+0+12 = 12 \rightarrow 12/4 = 3$
- $7+2+1+10 = 20 \rightarrow 20/4 = 5$
- $5+4+4+3 = 16 \rightarrow 16/4 = 4$

Wieder Mean-Filter 1x1:

- $8+3+5+4 = 20 \rightarrow 20/4 = 5 \rightarrow$ Ergebnis ist 5.

b) Compute the memory consumption for a 4096x4096 mip-map with 8 bits per pixel and one color channel. Note:

- A mip-map includes the original texture.
- Give a compact formula rather than the final number.

$$\text{Anzahl Bytes} = \frac{4n^2 * c - 1}{3} \text{ . Da hier aber nur 1 Farbkanal: } \frac{4n^2 * c - 1}{1}$$

$c = 8 \text{ bits, } n = 4096$

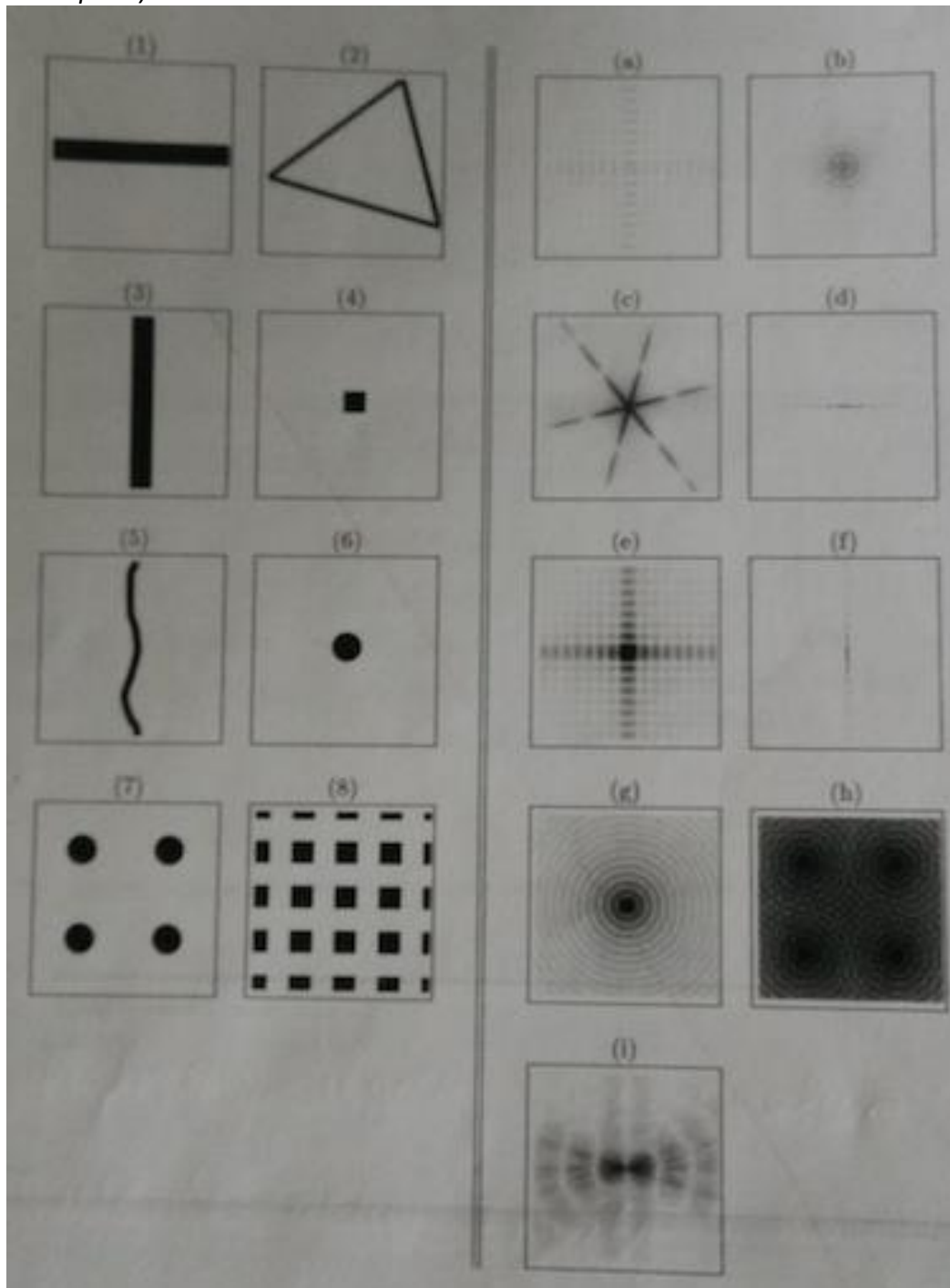
$$\rightarrow \frac{4 * 4096^2 * 8 - 1}{1}$$

c) What are the benefits from using mip-maps?

Wenn das Objekt weiter weg liegt, dann kann die passende Auflösung in Mip-Map ausgesucht werden. \rightarrow Es kann schneller gerendert werden.

13) Fourier Transform

Pair each of the images 1-8 with the corresponding Fourier spectra a-i. (One does not correspond).



1 \rightarrow F, 2 \rightarrow C, 3 \rightarrow D, 4 \rightarrow E, 5 \rightarrow I, 6 \rightarrow G, 7 \rightarrow B, 8 \rightarrow A

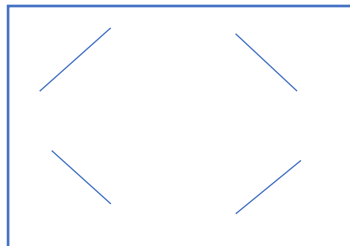
14) Sampling

- a) *Why is a Gauss filter not optimal for prefiltering before sampling to avoid aliasing? Which filter would be better?*

Für das Prefiltering vor Sampling ist der Gaussfilter nicht optimal, da es auch die Kanten blurt. Man braucht einen Filter der die Ecken zerstört. (sinc-Funktion)

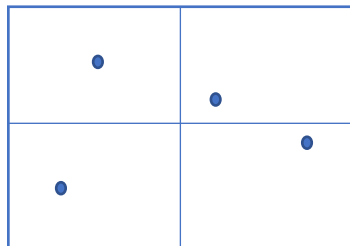
- b) *Name four supersampling sample placement strategies. Explain the sample placement for each. (Abtastmöglichkeiten für Sampling)*

- i) Rotated grid



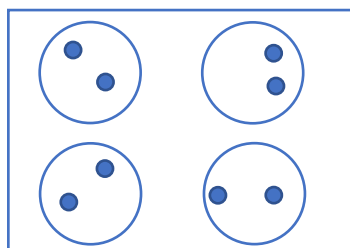
- Normalverteilung
- gedreht und leicht gestreckt
- + Relativ gleichmäßig
- Randbelastet

- ii) Stratified



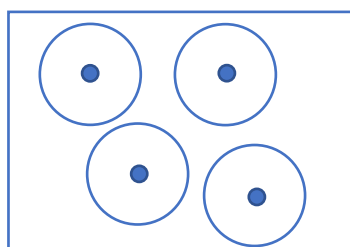
- Fläche in 4 Teile unterteilt mit Zufallspunkten
- + leicht zu berechnen
- Punkte können nah aneinander sein

- iii) Jittered



- Punkte zufällig um normale Verteilung gestreut (mit Max. Abstand)
- + Zufällig, aber gleichmäßig und nicht komplett am Rand
- bei zu großem Radius können Punkte nah bei einander liegen.

- iv) Poisson disc



- Zufällige Punkte mit „Gift-Zone“ außen herum, in die kein anderer Punkt sein darf.
- + Mindestabstand zwischen zwei Punkten gegeben (wie im menschlichen Auge)
- Können trotzdem alle in einer Ecke sein

- c) *How well do the sample placement strategies from b) perform considering noise and aliasing?*

Siehe Vorteil.

Bei Grid: Wenn Objekte ungünstig darin liegen, kommt es zum Aliasing.

15) Lightfields

- a) *Describe the information about the scene that is contained in a Lightfield and two possible parametrizations to store this data. What is dimensionality of the Lightfield?*

Ein Lichtfeld repräsentiert eine statische 3D-Szene mit einer 2D-Matrix, die 2D-Bilder enthält. Jedes 2D-Bild wurde von einer anderen Kamera-Position innerhalb einer Ebene aufgenommen. Für das Rendering werden Schnittpunkte der Kameraebene mit der Bildebene durchgeführt und der nächstgelegene Lichtstrahl verwendet. Durch Verschieben der Projektionsebene kann man auf verschiedene Objekte fokussieren.

- b) *Given a Lightfield, what novel renderings of the captured scene can be produced?*

- Pushbroom (Google Street View) → Zeilenkamera mit nur einer lichtempfindlichen Zeile, zwar keine bessere Pixelgröße, aber wesentlich mehr → bessere räumliche Auflösung. Dafür meist wesentlich höhere Taktraten gegenüber Flächenkameras → mehr Licht muss vorhanden sein
- Geometry assisted (IBR) → Generiert aus vielen 2D-Bildern ein 3D-Modell

- c) *Name 3 disadvantages of Lightfields*

- Kann nur eine statische Szene mit fester Beleuchtung aufnehmen
- Hoher Speicherplatzbedarf → Auflösung gering → Aliasing
- Für Photorealistisches Rendering bräuhete man unendlich viele Bilder

+ Einfache Berechnung

+ Kosten sind unabhängig von Komplexität der Szene, den Materialeigenschaften und anderen optischen Effekten

16) OpenGL

Describe the following terms. Explain for each what it is or what it does, where it is located in the OpenGL rendering pipeline and give an example application.

- a) *Transform Feedback*

(Prozess der Erfassung von Primitiven, die von Vertex Processing erzeugt werden)
Dynamische Länge, da Anzahl der Primitiven aus der Geometrieshaders nicht bekannt ist. Transform-Feedback-Buffer können als Eingang für Vertex Shader verwendet werden.

- b) *Fragment Shader*

(Shader-Stufe, die erzeugtes Fragment in einen Satz von Farben verarbeitet)
Beleuchtung der jeweils getroffenen Pixel.
Wird einmal pro gerenderten (Teil-) Pixel ausgeführt.

c) *Vertex Shader*

(programmierbare Shader-stufe in der Rendering-Pipeline)

Wird einmal pro Vertex ausgeführt.

Sieht nur die Daten eines Vertex + einheitliche Variablen.

Behandelt Verarbeitung einzelner Ecken.

d) *Vertex Array Object*

(Prozess der Einrichtung der notwendigen Objekte für das Rendering mit einem bestimmten Shader-Programm, sowie der Prozess der Verwendung dieser Objekte)

Verknüpft Daten in Array-Puffern mit Eingabestellen (0:1:...) des Vertex-Shaders.

Speichert aus den Array-Buffern.

Ließt:

→ welcher Datentyp

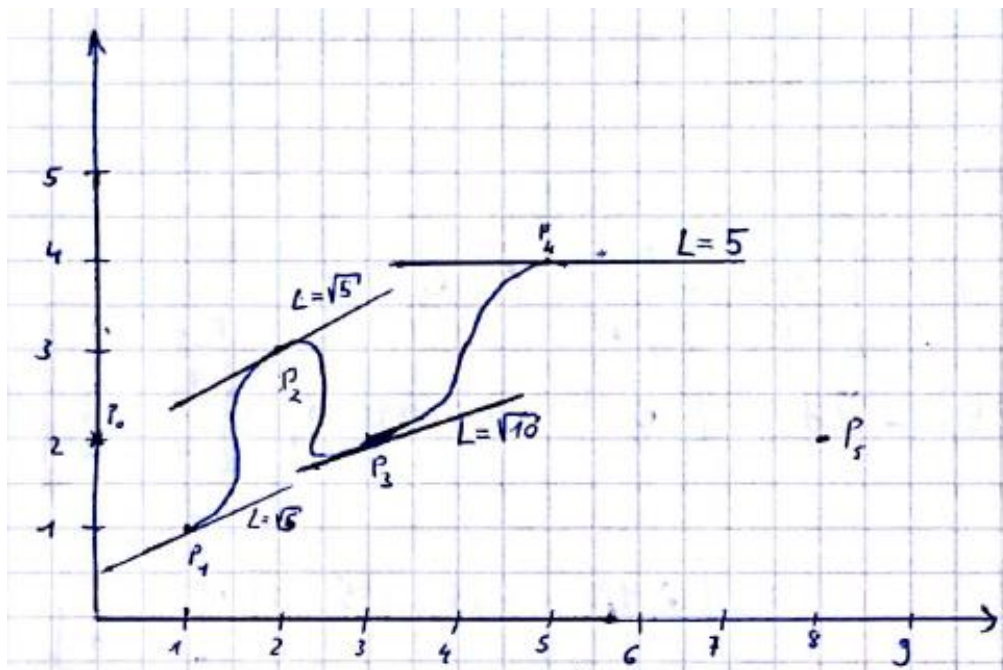
→ Vektorabmessungen

→ Offset

→ Stride

17) Catmull-Rom Splines

a) Draw and calculate the Catmull-Rom Spline tangent vectors of the points p_1, \dots, p_4 .



$P_0(0,2); P_1(1,1); P_2(2,3); P_3(3,2); P_4(5,4); P_5(8,2)$

$$P_1: \begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad L = \sqrt{2^2 + 1^2} = \sqrt{5}$$

$$P_2: p_3 - p_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad L = \sqrt{2^2 + 1^2} = \sqrt{5}$$

$$P_3: p_4 - p_2 = \begin{pmatrix} 5 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad L = \sqrt{3^2 + 1^2} = \sqrt{10}$$

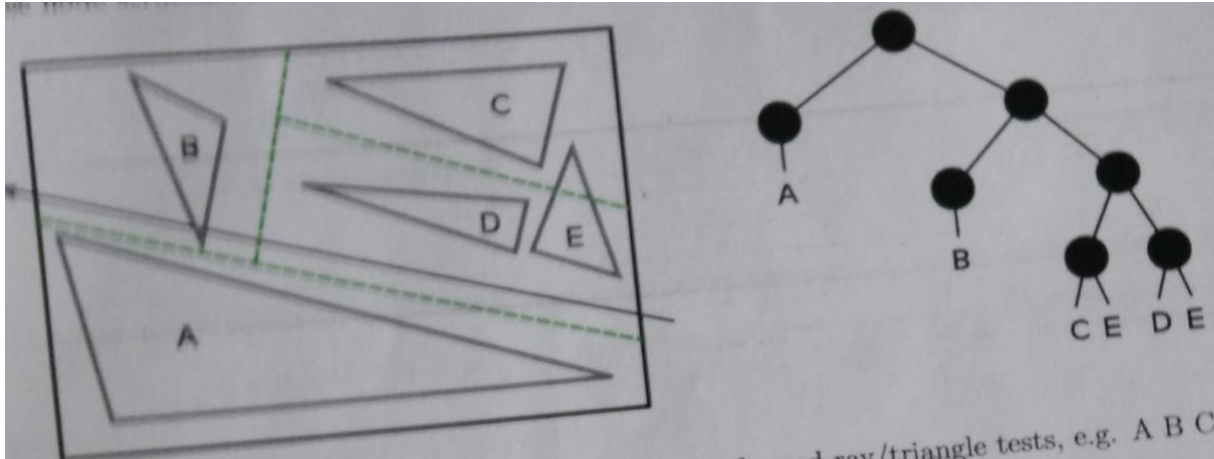
(D.h. 1,5 in x-Richtung und 0,5 in y-Richtung → Tangente an P_3)

$$P_4: p_5 - p_3 = \begin{pmatrix} 8 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \quad L = \sqrt{5^2 + 0^2} = \sqrt{25} = 5$$

- b) Which continuity does a Catmull-Rom Spline (including transitions between spline segments) have?

C_1 -Kontinuität. Für B-Splines: C_2 -Kontinuität.

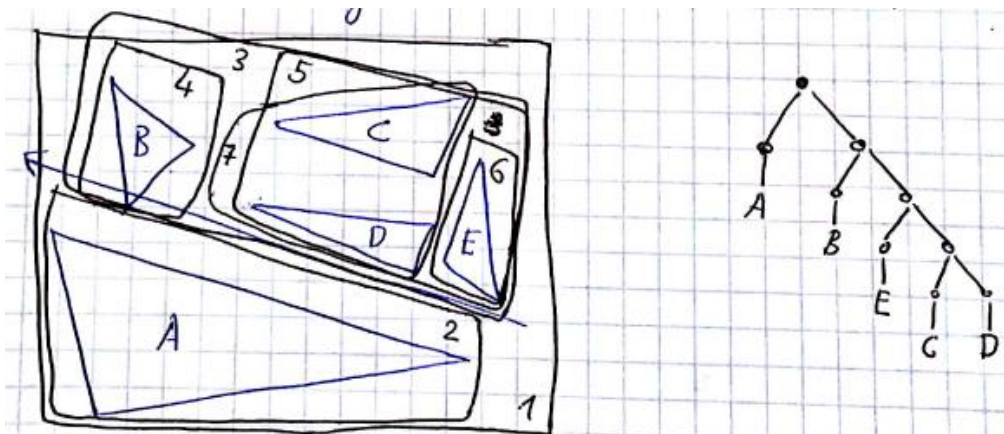
18) Acceleration Structures



- a) Traverse the BSP-tree! Give one possible order of performed ray/triangle tests, e.g. A B C D D. Do not optimize out multiple tests on the same triangle!

Reihenfolge: D, E, B. (A und C nicht, da die Boxen nicht vom Ray getroffen werden.)

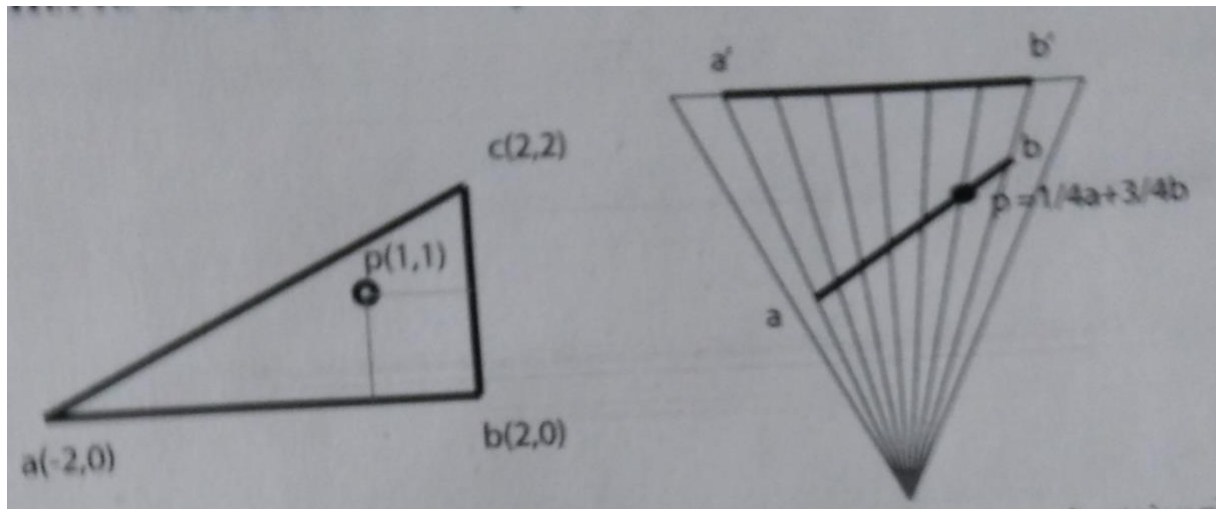
- b) Construct a bounding volume hierarchy (BVH) of your choice with at least four nodes. Don't count triangles as nodes (nodes are dots in the graph above). Draw the bounding volumes inside the picture below and draw a schematic tree for the structure you created.



Es sind mindestens so viele Boxen, wie Objekte vorhanden.

Ziel: So viele Objekte, wie möglich durch Rechtecke ausschließen, so dass geschnittenes Objekt übrig bleibt.

19) Barycentric Coordinates



- a) Compute the barycentric coordinates $\lambda_A, \lambda_B, \lambda_C$ of the point $p(1,1)$ in the triangle shown in the left figure.

$$\lambda_1 = 1 - \lambda_2 - \lambda_3$$

$$\lambda_2 = \frac{b_x P_y - b_y P_x}{b_x c_y - b_y c_x}$$

$$\lambda_3 = \frac{P_x c_y - P_y c_x}{b_x c_y - b_y c_x}$$

$$b = C - A, c = B - A, p = P - A$$

$$b = (2,2) - (-2,0) = (4,2)$$

$$c = (2,0) - (-2,0) = (4,0)$$

$$p = (1,1) - (-2,0) = (3,1)$$

$$\lambda_1 = \frac{4 * 1 - 2 * 3}{4 * 0 - 2 * 4} = \frac{-2}{-8} = \frac{1}{4}$$

$$\lambda_2 = \frac{3 * 0 - 1 * 4}{4 * 0 - 2 * 4} = \frac{-4}{-8} = \frac{1}{2}$$

$$\lambda_3 = 1 - \frac{1}{4} - \frac{1}{2} = \frac{1}{4}$$

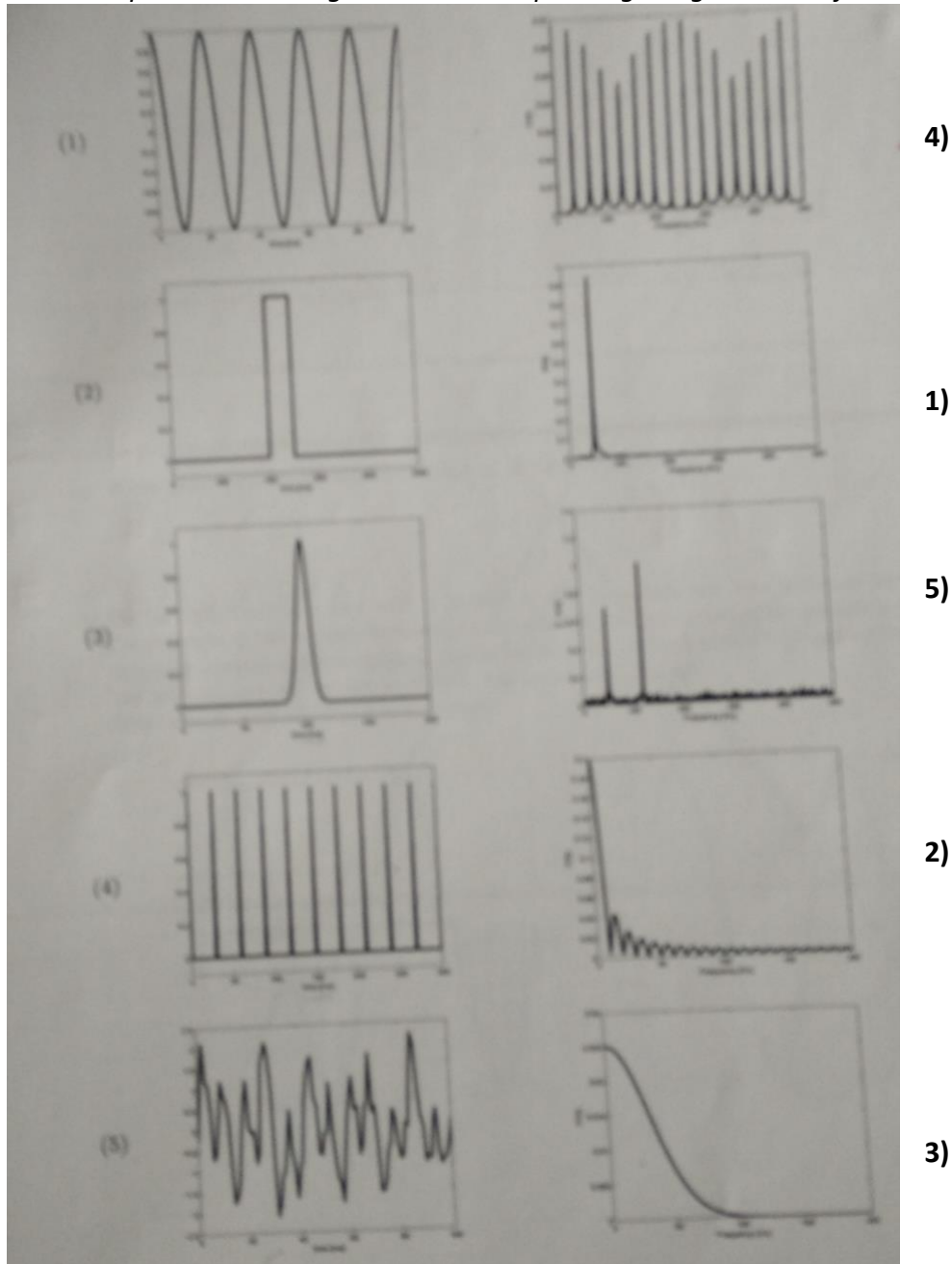
→ Barycentischen Koordinaten von P sind: $(\lambda_1, \lambda_2, \lambda_3) = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$

- b) Explain why barycentric coordinates need to be calculated before applying a perspective projection as seen in the right figure. Compare this to the case where one computes the barycentric coordinates on the projected triangle. (No numbers required).

Man muss die Baryzentischen Koordinaten vor der perspektivischen Projektion berechnen, damit man weiß, ob der Punkt überhaupt im Dreieck liegt. Je weiter weg von der Kamera, desto näher ist z.B. b zu b' .

20) Fourier Transform

Pair each picture on the right to the corresponding image on the left.



21) Image Filter

- a) The original image was convolved with different filters: Gaussian, Bilateral and Sobel. Assign the correct filter to the images.



- b) Describe in short the median filter. Is the median filter appropriate to remove salt and pepper noise?

- Gut gegen Impulsrauschen (Salt and pepper noise)
- Dünne Linien und Ecken werden zerstört – kleine Strukturen gehen verloren
- Kreuzförmige Maske für den Erhalt senkrechter + waagrechter Linien
- Mehrfachwertung des zentralen Wertes
- Bewirkt nicht so hohe Kantenglättung wie Gauss Filter
- Nicht geeignet für Text

Vorgehensweise: Betrachte alle Pixel um den ausgewählten Pixel (z.B. 3x3). Sortiere nach ihrem Grauwert. Bestimme Median. Ersetze alle betrachteten Pixel durch den Pixel, der Median entspricht.

22) Homogeneous Transformations

a) Identity:

7	0	0	0
0	7	0	0
0	0	7	0
0	0	0	7

e) Rotation around the z-axis

0	-1	0	0
1	0	0	0
0	0	1	0
0	0	0	1

b) Translation:

3	0	0	3
0	3	0	3
0	0	3	3
0	0	0	3

f) Non-uniform scaling

1	0	0	0
0	3	0	0
0	0	1	0
0	0	0	1

c) Projective Transformation

3	0	0	0
0	3	0	0
0	0	3	1
0	0	1	0

g) Rotation around the x-axis

2	0	0	0
0	0	-2	0
0	2	0	0
0	0	0	2

d) Uniform scaling

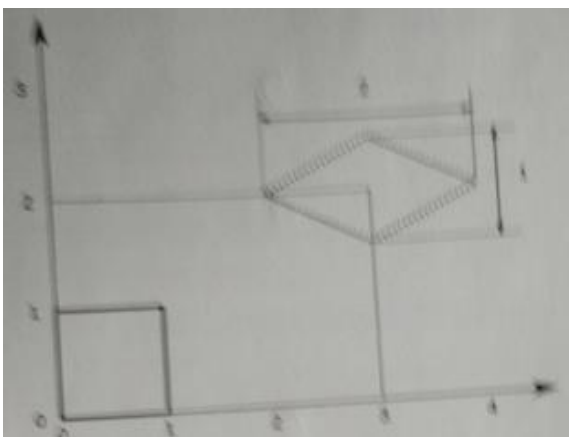
3	0	0	0
0	3	0	0
0	0	3	0
0	0	0	1

h) Rotation and Mirroring

1	0	0	0
0	0	-1	0
0	-1	0	0
0	0	0	1

23) Transformations

Calculate the transformation. → Mittelpunkt vom Startpunkt: (0,5/0,5) → Punkt muss 2,5 in x-Richtung und 1,5 in y-Richtung verschoben werden.



$$\text{Translation: } \begin{pmatrix} 1 & 0 & 2,5 \\ 0 & 1 & 1,5 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 0,5 \\ 0,5 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

$$\text{Rotation: } \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(45^\circ) & -\sin(45^\circ) \\ 0 & \sin(45^\circ) & \cos(45^\circ) \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0,7 & -0,7 \\ 0 & 0,7 & 0,7 \end{pmatrix}$$

Scherung: $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ in x-Richtung um 1




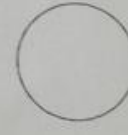


Skalierung: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ in x-Richtung um 1 und in y-Richtung um 2.

Ergebnis:

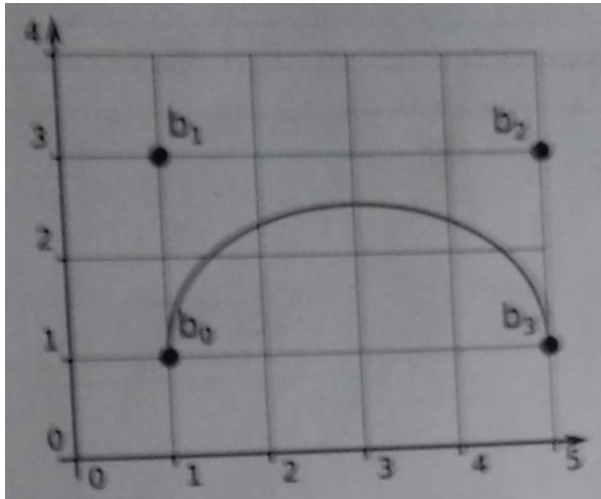
$$\begin{pmatrix} 1 & 0 & 2,5 \\ 0 & 1 & 1,5 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(45^\circ) & -\sin(45^\circ) \\ 0 & \sin(45^\circ) & \cos(45^\circ) \end{pmatrix} * \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Rechnen von hinten nach vorne und immer zuerst Scherung vor Rotation.

24) OpenGL Shading Language

F	<pre> 1 vec3 toLight = normalize(-LightDirection); 2 3 vec3 n = normalize(inData.Normal); 4 5 vec3 color = MatCol * LightCol * max(0.0, dot(n, toLight)); 6 7 outColor = vec4(color, 1.0); </pre>	A: 
D	<pre> 1 vec3 toViewer = normalize(-inData.Position); 2 vec3 n = normalize(inData.Normal); 3 4 if(dot(n, toViewer) < 0.2) 5 outColor = vec4(0.0, 0.0, 0.0, 1.0); 6 else 7 outColor = vec4(1.0, 1.0, 1.0, 1.0); </pre>	B: 
B	<pre> 1 vec3 toViewer = normalize(-inData.Position); 2 vec3 toLight = normalize(-LightDirection); 3 4 vec3 n = normalize(inData.Normal); 5 vec3 h = normalize(toViewer + toLight); 6 7 vec3 d = MatCol * LightCol * max(0.0, dot(n, toLight)); 8 vec3 s = SpecCol * LightCol * max(0.0, pow(dot(h, n), 1000.0)); 9 10 outColor = vec4(d + s, 1.0); </pre>	C: 
E	<pre> 1 vec3 toViewer = normalize(-inData.Position); 2 vec3 toLight = normalize(-LightDirection); 3 4 vec3 n = normalize(inData.Normal); 5 vec3 h = normalize(toViewer + toLight); 6 7 vec3 d = dot(n, toLight) > 0 ? MatCol : AmbientColor; 8 vec3 s = pow(dot(h, n), 20.0) > 0.85 ? SpecCol : vec3(0.0, 0.0, 0.0); 9 10 outColor = vec4(d + s, 1.0); 11 12 if(dot(n, toViewer) < 0.2) 13 outColor = vec4(0.0, 0.0, 0.0, 1.0); </pre>	D: 
A	<pre> 1 vec3 toViewer = normalize(-inData.Position); 2 vec3 toLight = normalize(-LightDirection); 3 4 vec3 n = normalize(inData.Normal); 5 vec3 h = normalize(toViewer + toLight); 6 7 float w1 = dot(n, toViewer); 8 float w2 = 1.0 - w1; 9 vec3 materialColor = w1 * MatCol1 + w2 * MatCol2; 10 11 vec3 d = materialColor * LightCol * max(0.0, dot(n, toLight)); 12 vec3 s = SpecCol * LightCol * max(0.0, pow(dot(h, n), 1000.0)); 13 14 outColor = vec4(d + s, 1.0); </pre>	E: 
C	<pre> 1 vec3 toViewer = normalize(-inData.Position); 2 vec3 toLight = normalize(-LightDirection); 3 4 vec3 n = normalize(inData.Normal); 5 vec3 h = normalize(toViewer + toLight); 6 7 vec3 hTan = normalize(h - n * dot(h, n)); 8 float w1 = abs(dot(hTan, inData.Tangent1)); 9 float w2 = 1.0 - w1; 10 float e = 100.0 * w1 + 10.0 * w2; 11 12 vec3 d = MatCol * LightCol * max(0.0, dot(n, toLight)); 13 vec3 s = SpecCol * LightCol * max(0.0, pow(dot(h, n), e)); 14 15 outColor = vec4(d + s, 1.0); </pre>	F: 

25) Splines



- a) What is the coefficient matrix A for the monomial representation of the Bézier spline shown above? The basis Matrix M for Bézier splines is:

$$\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$[t^3 \ t^2 \ t \ 1] * \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 5 & 3 \\ 5 & 1 \end{pmatrix}$$

$$A = M * G \quad \begin{pmatrix} -8 & 0 \\ 12 & -6 \\ 0 & 6 \\ 1 & 1 \end{pmatrix}$$

$$P(t) = T * A$$

- b) Calculate the position of the point $P(0,5)$ with $t=0,5$ on the spline.

$$b_0(1,1), b_1(1,3), b_2(5,3), b_3(5,1)$$

$$\begin{matrix} b_0 \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \searrow & \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ b_1 \begin{pmatrix} 1 \\ 3 \end{pmatrix} & \searrow & \begin{pmatrix} 2 \\ 2,5 \end{pmatrix} \\ b_2 \begin{pmatrix} 5 \\ 3 \end{pmatrix} & \searrow & \begin{pmatrix} 3 \\ 4 \end{pmatrix} \\ b_3 \begin{pmatrix} 5 \\ 1 \end{pmatrix} & \searrow & \begin{pmatrix} 5 \\ 2,5 \end{pmatrix} \end{matrix} \rightarrow \begin{pmatrix} 3 \\ 2,5 \end{pmatrix}$$

- c) Calculate the tangent vector $P'(0,5)$ with $t=0,5$ on the spline.

$$p'(t) = [3t^2 \ 2t \ t \ 0] \cdot \begin{bmatrix} -8 & 0 \\ 12 & -6 \\ 0 & 6 \\ 1 & 1 \end{bmatrix}$$

$$p'(0,5) = [0,75 \ 1 \ 1 \ 0] \cdot \begin{bmatrix} -8 & 0 \\ 12 & -6 \\ 0 & 6 \\ 1 & 1 \end{bmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} \rightarrow \text{Tangent vector}$$

Radiometric Quantities

- energy	[watt second]	$n \cdot h\lambda$ (Photon Energy)
- radiant power (total flux)	[watt]	Φ
- radiance	[watt/(m ² sr)]	L Strahlung
- irradiance	[watt/m ²]	E Bestrahlungsstärke
- radiosity	[watt/m ²]	B Ausstrahlung
- intensity	[watt/sr]	I