Prof. Dr. Leif Kobbelt
Patrick Schmidt, Joe Jakobi

Basic Techniques in Computer Graphics

Assignment 9

Date Published: December 20th 2022, Date Due: January 10th 2023

- All assignments (programming and theory) have to be completed in teams of 3–4 students. Teams with fewer than 3 or more than 4 students will receive no points.
- Hand in **one solution per team per assignment**.
- Every team must work independently. Teams with identical solutions will receive no points.
- Solutions are due on January 10th 2023 via Moodle. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Make sure you are part of a Moodle group with 3-4 members. See "Group Management" in the Moodle course room.
  - Download the solution template (a zip archive) through the Moodle course room.
  - Unzip the archive and populate the `assignmentXX/MEMBERS.txt` file. The names and student ids listed in this file **must match** your moodle group **exactly**.
  - Complete the solution.
  - Prepare a new zip archive containing your solution. It must contain exactly the files that you changed. **Only change the files you are explicitly asked to change in the task description.** The directory layout must be the same as in the archive you downloaded. (At the very least it must contain the `assignmentXX/MEMBERS.txt`.)
  - One team member uploads the zip archive through Moodle before the deadline, using the group submission feature.
  - Your solution must compile and run correctly **on our lab computers** by only inserting your **assignment.cc** and **shader files** into the Project. If it does not compile on our machines, you will receive no points. If in doubt you can test compilation in the virtual machine provided on our website.
- Instructions for **text assignments**:
  - Prepare your solution as a single pdf file per group. Submissions on paper will not be accepted.
  - If you write your solution by hand, write neatly! Anything we cannot decipher will receive zero points. No exceptions!
  - Add the names and student ID numbers of all team members to every pdf.
  - Unless explicitly asked otherwise, always justify your answer.
  - Be concise!
  - Submit your solution via Moodle, together with your coding submission.

Exercise 1 Ray-Quadric Intersection **[10 Points]**

In the lecture you learned about a general way to find the intersection point(s) of a ray with a quadric surface. Consider the surface in $\mathbb{R}^3$ defined by the quadric

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

(a) Surface Description **[2 Points]**

Describe the surface defined by $\mathbf{Q}$ in a few words (e.g. the type of object, its dimensions, its position and alignment in space).

(b) Ray Intersection **[6 Points]**

We now intersect the above surface with the ray $\mathbf{c}+\lambda\mathbf{r}$, where $\mathbf{c}$ is given as $(0, 0, \sqrt{2})^\mathsf{T}$ and $\mathbf{r} = (r_x, r_y, r_z)^\mathsf{T}$ is a general direction. Derive a closed form solution for the parameter λ at the intersection point(s), i.e. a formula $\lambda(r_x, r_y, r_z) \in \mathbb{R}$. **Explain your derivation** step by step!

(c) Example **[2 Points]**

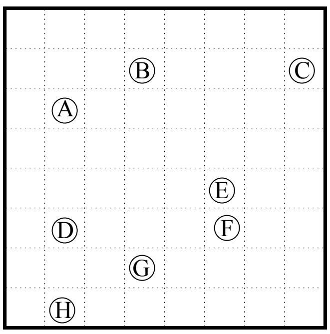
Now also consider $\mathbf{r}$ to be given as $(0, 1, -1)^\mathsf{T}$ and compute the intersection points $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^3$. How many distinct intersection points exist? What is the geometric interpretation of this configuration?

Exercise 2 Spatial Data Structures **[10 Points]**

To speed up algorithms a scene is often partitioned using spatial data structures.

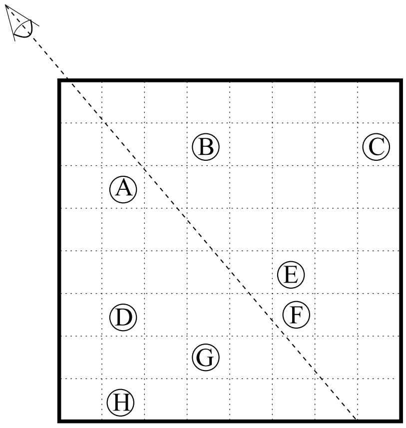
(a) Quad-Tree **[2 Point]**

Construct a quad-tree for the scene shown below. Stop subdivision as soon as every node contains at most one object. Draw the splitting lines into the illustration of the scene and sketch the resulting tree structure.



(b) kD-Tree **[2 Point]**

Construct a kD-tree for the scene shown below. Stop subdivision as soon as every node contains at most one object. Start with a split parallel to the horizontal axis. Draw the splitting lines into the illustration of the scene and sketch the resulting tree structure.

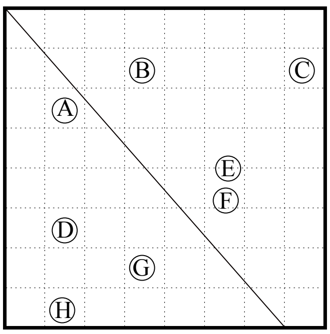


(c) Painter's Algorithm **[2 Point]**

Use the kD-tree constructed in the previous task to determine the order in which the objects are drawn during Painter's Algorithm, given the camera position shown in the illustration.

(d) Binary Space Partition **[2 Point]**

Construct a BSP-Tree for the scene shown below. Stop subdivision as soon as every node contains at most one object. Draw the splitting lines into the illustration of the scene and sketch the resulting tree structure. Make sure your tree is balanced!



(e) Comparison **[2 Point]**

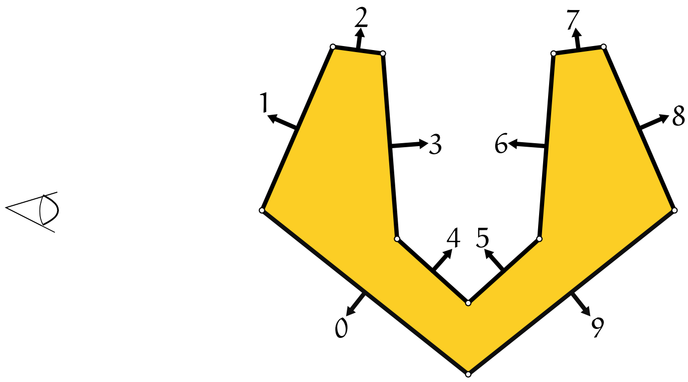
What are the differences between BSP-trees and kD-trees? What benefits and disadvantages do BSP-trees have compared to kD-trees?

Exercise 3 Culling **[10 Points]**

In the lecture, we learned about various culling techniques, i.e., methods to discard triangles early-on if it is clear that they will not be visible.

(a) Backface Culling Example **[2 Points]**

Consider the following scenario in *camera space*. Assume that backface culling is performed by evaluating the dot product of the viewing direction with each face normal. Decide for each triangle (numbered 0 to 9) whether it is culled or not.



Triangle	BF Culled
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

(b) Backface Culling **[2 Points]**

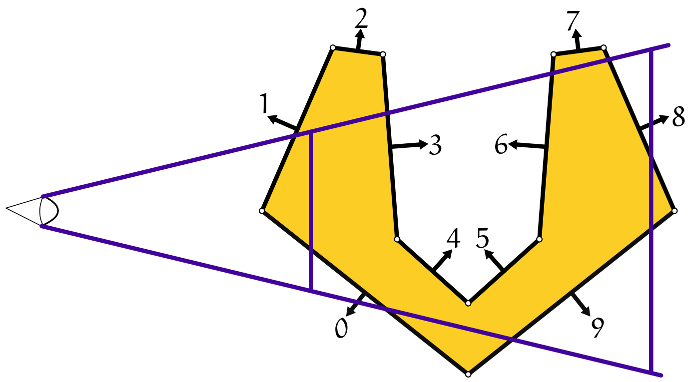
Instead of performing backface culling in camera space, we could perform it in normalized device coordinates, i.e., after applying the frustum transformation. Does this strategy change any of your answers in the above table? If yes, which one(s) and why?

(c) View Frustum Culling and Clipping **[2 Points]**

Briefly explain the terms *view frustum culling* and *clipping*.

(d) View Frustum Culling and Clipping Example **[2 Points]**

Consider the view frustum illustrated in the following sketch. For each triangle (numbered 0 to 9), decide whether it is discarded by *view frustum culling*, and whether it is subject to *clipping*.



Triangle	VF Culled	Clipped
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

(e) Occlusion Culling **[2 Points]**

Briefly explain the concept of *occlusion culling* and describe one possible way of implementing it.

Exercise 4 Color Models **[10 Points]**

In the lecture we learned several things about colors and color spaces.

(a) **[1 Points]**

Explain in one sentence what metamerism is.

(b) **[1 Points]**

Why does it make sense to only represent a color with three base colors?

(c) **[3 Points]**

Name three color models (one hardware oriented, one user oriented and one scientifically founded) and their main usage (or advantage).

(d) **[2 Points]**

Explain the differences between an *additive* and a *subtractive color model*. Name one example for both categories.

(e) **[2 Points]**

Briefly explain the difference between RGB and sRGB. How to transform a color from RGB space to sRGB?

(f) **[1 Points]**

Describe the process of acquiring HDR Images using a *digital single-lens reflex* (DSLR) camera in one sentence.