

Color Code:

Use green highlights if you are 100% sure the answer is correct.

Use red highlights for answers that need to be verified by a 2nd person to turn green.

Unanswered questions

When verifying, add the lecture slide number for source.

CG1 Possible Questions

WiSe 21/22

Lecture 7 Light

1. What part of the electromagnetic spectrum is visible?

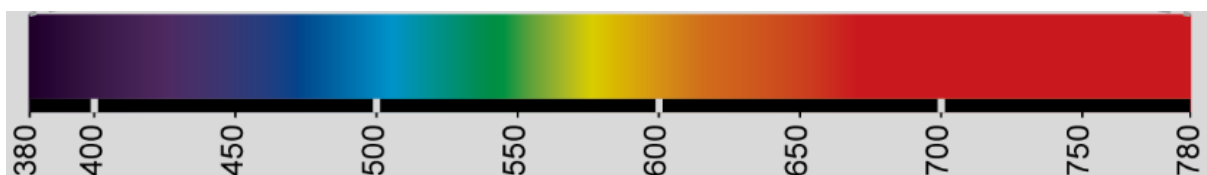
- Electromagnetic waves with a wavelength of ~380 to about ~770nm

2. What is monochromatic light? Is white light monochromatic?

- Monochromatic light is a light that only contains a single color (wavelength)
- White light is not monochromatic since there is no single wavelength that produces white light

3. Order shown colors by frequency/wavelength

- visible light



4. Explain geometric, wave, quantum optics [6]

- Geometric:
 - Basic idea: Model light as **rays** (to simulate properties of most optical instruments)
 - Light travels along straight lines
 - **Reflection and refraction** taken into account
- Wave:
 - Basic idea: Model light as **wave** (waves travel from each point in space in all directions)
 - **Interference and diffraction**
 - Holography
- Quantum:
 - Basic idea: Light as a **Photon** (travels in straight lines with speed c)
 - Photo effect, Compton effect
 - Light frequency: ν , planck's constant h , photon energy: $h \cdot \nu$

5. (Explain the eye...) What are rods and cones.. What are they used for.. [12]

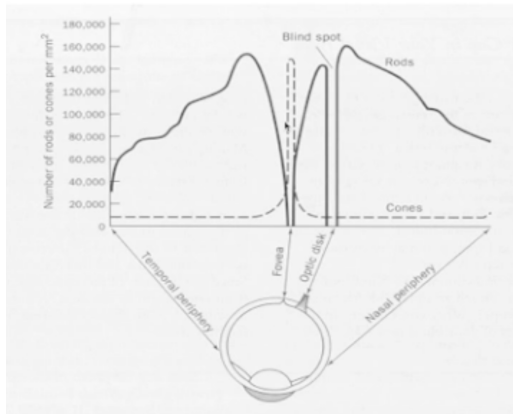
- Rods and Cones are photoreceptors within the retina.
- Rods (highly sensitive): (brightness)
 - 100 - 120 Mio.
 - Sensitivity max. at 498 nm (green)
- Cones: (color at high light conditions)
 - 7 - 8 Mio.
 - 3 types
 - Sensitivity max at.
 - 420 nm (short - blue)
 - 534 nm (middle - green)
 - 564 nm (large - red)

6. Explain the blind spot

- Spot where the optic nerve connects to the eye, the retina lacks rods and cones in this area, called the optic disk.

Where are the most cones located? [13]

- fovea centralis
- (few rods in fovea)

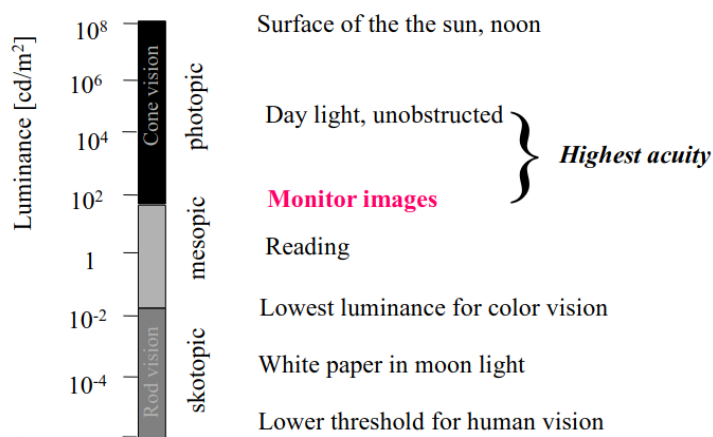


What type of cone is the least frequent one?

- S - Cones (short 420 nm) (Only 10%)

7. Explain scotopic vs photopic vision.

- Night vision (scotopic): rods
- Daylight vision (photopic): cones



8. What is a cause for red-green weakness?

-Looking at the absorbance curves for the cones, M(Green responsible), L(Red responsible). We notice that their maximas are close (534, 564m). Due to certain human mutations these curves can overlap which will result in inability to differentiate red from green color.

9. What is inner/outer color mixing? How and why do you do it?[21-24]

- In the tristimulus system (RGB) colors can be represented as vectors in barycentric coordinates in a triangle. The inner mixing includes all colors inside the triangle spanned by the primary color vectors. The outer color mixing refers to the outside of the triangle.

The color equation $F = rR + gG + bB$ would result in r, g, b psychophysical coefficients given by observers to produce additive color mixing. The colors that cannot be generated by addition (for eg. subtract some red from the equation) is outer color mixing. This would give different coefs, if birds were observers for instance.

10. What is Grassmann's law?

- Among four colors there is a linear dependency
⇒ Any color requires three independent colors for representation (color as three dimensional quantity) (at least mathematical, in reality we cannot cover all visible color with only three colors)

11. What are metamers?

- Metamers are colors that appear identical to the human eye, but have different spectral distributions

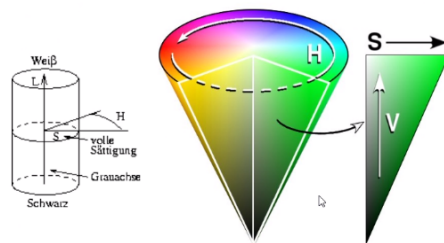
12. Why can't we mix all colors from three given ones?

-

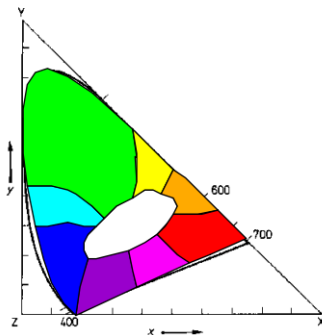
13. List different color models/spaces and explain the differences/what they are used for..

- RGB:
 - Used for: Computer Screens
 - Background black, all color mixing: white
 - additive mixing

- CMY(K):
 - Used for: Painting/Printing
 - Background white, all color mixing: brown (theoretically black but practically brown)??
 - subtractive mixing
- YIQ, YCrCb:
 - Used for: Video Coding
 - Grayscale + R-Y, B-Y (differences red - gray and blue - gray, respectively)
- HLS:
 - Used for: Easy to understand for Humans \Rightarrow Editing
 - H: "angle" (color direction), L: brightness, S: "how much" color



- Cylinder: Relative saturation, Cone: absolute saturation
- CIE XYZ:



14. Why do we need black for CMY(K)?

- Because, when actually printing, you can't get a nice black color through mixing Cyan Magenta and Yellow. A brownish color will come from it -> need black for printing e.g.

15. Convert from RGB to CMY and back

- $RGB \Rightarrow CMY: (C, M, Y) = (1, 1, 1) - (R, G, B)$

- CMY \Rightarrow RGB: $(R, G, B) = (1, 1, 1) - (R, G, B)$

This is possible since CMY is the complement of RGB

16. Why do we only need two colors for YCrCb?

- Chromaticity is represented as the differences R-Y and B-Y (gray to red and gray to blue, respectively) which together form a three-dimensional color space.
- To get Y (Luminance) $Y = 0.3R + 0.59G + 0.11B$

17. Convert from RGB to YCrCb

$$\begin{pmatrix} Y \\ C_R \\ C_B \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 1.402 \\ 1.0 & -0.34414 & -0.71414 \\ 1.0 & 1.772 & 0.0 \end{pmatrix} \begin{pmatrix} Y \\ C_R \\ C_B \end{pmatrix}$$

YIQ \Rightarrow YCrCb

- $Y = 0.3R + 0.59G + 0.11B$
- $I = Cr = 0.6R - 0.28G - 0.32B$
- $Q = Cb = 0.21R - 0.52G + 0.31B$

PS: The factors only reflect human lightness sensitivity

18. What is H, L, S in HLS?

- Hue, Lightness, Saturation

19. What are the parameters used to calculate color in different light models?

- usually, luminance and chromaticity

Which physical quantity do humans perceive as "brightness" of an object?

- Luminance L (Leuchtdichte)

What is the difference between luminance and luma?

- luminance: physical unit, luma: corresponding encoding, nonlinear luminance

Lecture 8 Rasterization

1. List different rasterization algorithms, give definition/description

- **Bresenham:**
 - Which pixel center is closer to the line at the current index/slope
 - You can use this with only integer calcul. (sign checking)
- **Midpoint:**
 - In which half-space is $M = (x+1, y+\frac{1}{2})$ if you consider the implicit line equation

$$g(x, y) = \Delta y * x - \Delta x * y + (\Delta y * y_1 - \Delta y * x_1)$$
 - This introduces a cool incremental computation for circles and other line equations, to choose which pixel to color according to result >0 ?
- **Scanline:**
 - Generate intersections of scan line with all edges of polygon
 - Sort intersections by x coordinates
 - Fill pixels between pairs of consecutive intersections
 -

Explain a Verdeckungsrechnungs algorithm

- **z-Buffer:**
 - Store the color and depth of considered objects in a buffer. (Only consider the nearest and overwrite)

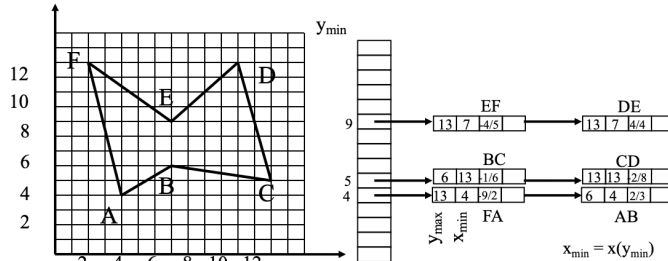
Z-buffer and transparency

- Difficult to realize ([How the Z-Buffer Works - Interactive 3D Graphics](#))

2. Fill out the table for scanline algorithm Slide 20

▪ Edges:

- $e = (x_{min}, y_{min}, x_{max}, y_{max}, D_y/D_x)$
- AET (Active Edge Table)
 - (e_1, \dots, e_k) sorted by x_{cut}
- ET (Edge Table)
 - All edges, sorted with Bucket-Sort by y_{min}



Given a closed polygon with vertices (1, 11), (6, 1), (12, 4), (10, 7), (13, 9), (17, 7) and (14, 14), give an edge table for the polygon.

Name 3 Scanline algorithm polygon traversing strategies

- brute force (snake over the full picture frame)
- edge to edge (snake from edge of polygon to opposing edge of polygon)
- edge following ("tree" from edge to edge of polygon)

3. Why is Bresenham's faster than the others?

- Incremental integer algorithm that requires minimal calculation (integer, plus and minus) per pixel instead of floating point operations.

Rasterize a line from point (101, 5) to point (107, 9). Write all values x, y, E' and whether you switch rows for each step.

- (102,6)(103,6)(104,7)(105,8)(106,8)

$E' = 2\Delta y - \Delta x = 2 \cdot 4 - 6 = 2 > 0 \Rightarrow x:=x+1, y:=y+1$ (102,6)

$E' := E' + 2\Delta y - 2\Delta x = 2 + 8 - 12 = -2 < 0 \Rightarrow x:=x+1$ (103,6)

$E' := E' + 2\Delta y = -2 + 8 = 6 > 0 \Rightarrow x:=x+1, y:=y+1$ (104,7)

$E' := E' + 2\Delta y - 2\Delta x = 6 + 8 - 12 = 2 > 0 \Rightarrow x:=x+1, y:=y+1$ (105,8)

$E' := E' + 2\Delta y - 2\Delta x = 2 + 8 - 12 = -2 < 0 \Rightarrow x:=x+1$ (106,8)

[\(Lecture 8 Slide 8\)](#)

Derive the midpoint algorithm for a parabola.

4. How do you smooth a line?

- Interpolate a grayscale value for each pixel depending on the distance to the midpoint

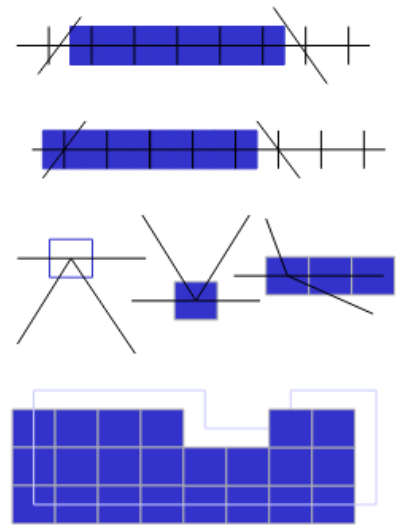
5. Argue Pros and Cons of the algorithms

-

6. What strategies are used for polygon filling in the Scanline algorithm

■ Fill strategy

- Intersection at float value
 - Round up inside, round up outside
- Intersection at integer value
 - Left endpoint of span: inside
 - Right endpoint of span: outside
- Common vertex for neighboring polygons
 - Only y_{\min} -vertex of an edge is considered for parity
- Horizontal edge
 - Pixel of horizontal edges are not considered for parity



With which value does the z-buffer need to be initialized?

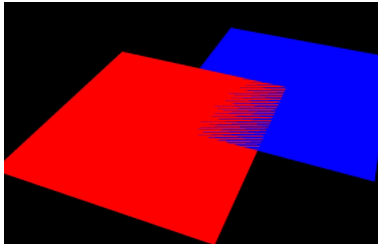
- initialize with 1 (value of far clipping plane)
- (1 is furthest away from camera (far clipping plane), 0 is closest (near clipping plane))

How does the z-buffer algorithm work?

- Initialize with 1 (max z-value, or far-clipping plane)
- For every pixel of every primitive, compute the Z Buffer, (After perspective transformations have been applied)
If z-value (= depth) at pixel is smaller than the current stored value, overwrite value and color.

What is Z-fighting?

- If two planes are at the same/similar coordinates, they fight over depth (due to floating point precision) → The last considered primitive wins
- Causes artifacts



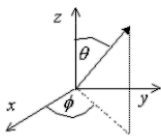
Lecture 9 Texturing

Why do we use texturing?

- Modeling of objects using primitives often too complex for real-time applications
⇒ Textures add visual complexity

List different mapping functions and explain how they work

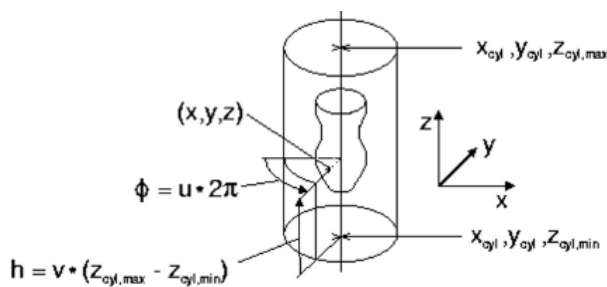
- Sphere (x_s, y_s, z_s represent the center of the sphere)



$$u = \frac{\pi + \arctan 2(y - y_s, x - x_s)}{2\pi}$$

$$v = \frac{\arctan 2(\sqrt{(x - x_s)^2 + (y - y_s)^2}, z - z_s)}{\pi}$$

- Cylinder

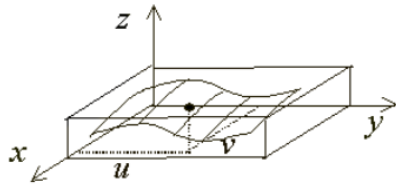


- Box

z-axis and y-axis might need to be switched!

Choose biggest lengths of box (in this case x, then y) for u and v

$$u = \frac{x - x_{box,min}}{x_{box,max} - x_{box,min}} \quad v = \frac{y - y_{box,min}}{y_{box,max} - y_{box,min}}$$



For which class of objects can sphere mapping be applied?

- convex objects (more accurate: any object where it is possible to draw lines between sets of two arbitrary points inside of the object, so that the line completely lies inside the object as well)

Apply box mapping (center at the origin, x-width = 3, y-height = 2, z-depth = 4),

cylinder mapping (center at the origin, height along y, total height = 4, discontinuity at -z) and

sphere mapping (center at the origin, discontinuity at -z) to the point (1, 1, 1).

Box:

- $u = (z - z_{min}) / (z_{max} - z_{min}) = 3/4$
- $v = (x - x_{min}) / (x_{max} - x_{min}) = 5/8$

Sphere :

- $u = (\pi + \arctan(-1,-1)) * 1/2 \pi = 5/8$
- $v = \arctan(\sqrt{2}, 1) * 1/\pi$

Cylinder: (Phi: angle between x, y; $h = z - z_{min}$)

- $u = \text{Phi} / (2 * \pi) = 5/8 \quad (\text{Phi} = \arctan(x, y))$
- $v = (z - z_{min}) / (z_{max} - z_{min}) = 3/4$

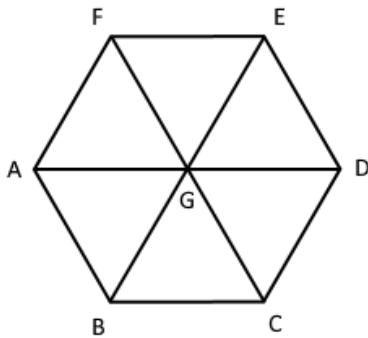
What mapping was used for a given model?

- If cool chess checkerboard, then box mapping.

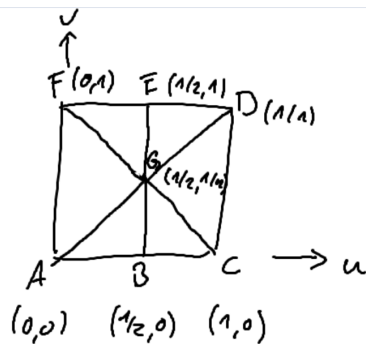
Give two assignments of texture coordinates for the following mesh of a regular hexagon

(see image): One which maps the texture to the hexagon with minimal distortion and one

which uniquely maps all points of the texture to the hexagon, and vice versa.



- minimal distortion: box mapping
 - A: 0, 0.5
 - B: 0.25, 0.07
 - C: 0.75, 0.07
 - D: 1, 0.5
 - E: 0.75, 0.93
 - F: 0.25, 0.93
 - G: 0.5, 0.5
- unique mapping:



Why would we use 3D textures?

- The object appears to be carved out of the solid texture

What are problems that can occur with rasterized textures?

- Magnification of discrete images can cause artifacts
- Mapping flat textures to curved surfaces can cause distortions and discontinuities
-

What is Over/Undersampling?

- Oversampling: "Magnification", sample points are distributed more dense than pixels

- Undersampling: "Minification", sample points are more sparse than pixels

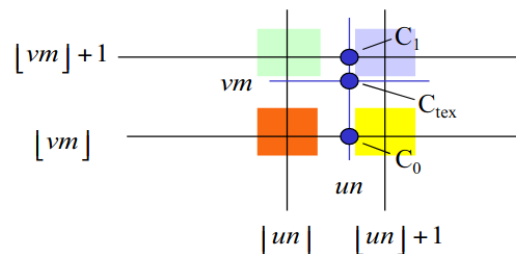
List 3 strategies of how to choose a color when a texture is sampled outside of the usual u,v-coordinate range.

- repeat texture (texture tiling, texture can be mirrored)
- return background color
- nearest texture point

Name 6 methods for sampling a texture when the texture is minified.

- average pixels (Summed area tables(SATs))
- mip-maps
- nearest neighbour
- bilinear interpolation between "hit pixels"
- mip maps with bilinear
- trilinear interpolation (bilinear interpolation between two bilinearly sampled mip map layers)

Bilinear interpolation formula



$$\hat{u} = un - [un]$$

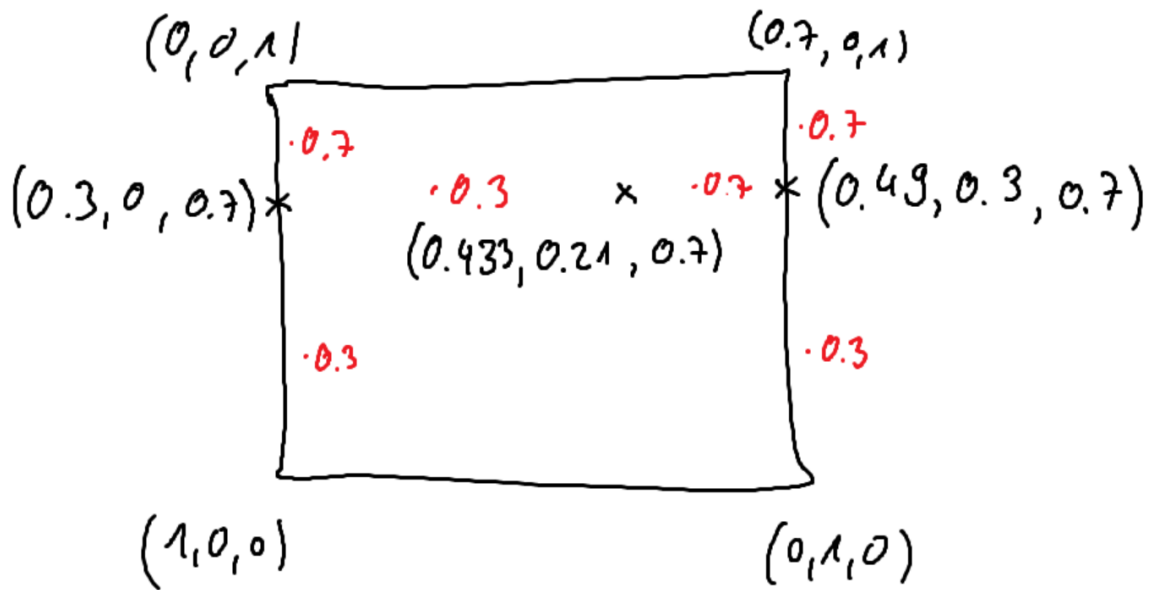
$$\hat{v} = vm - [vm]$$

$$C_0(u, v) = \hat{u} * C [[un] + 1, [vm]] + (1 - \hat{u}) * C [[un], [vm]]$$

$$C_1(u, v) = \hat{u} * C [[un] + 1, [vm] + 1] + (1 - \hat{u}) * C [[un], [vm] + 1]$$

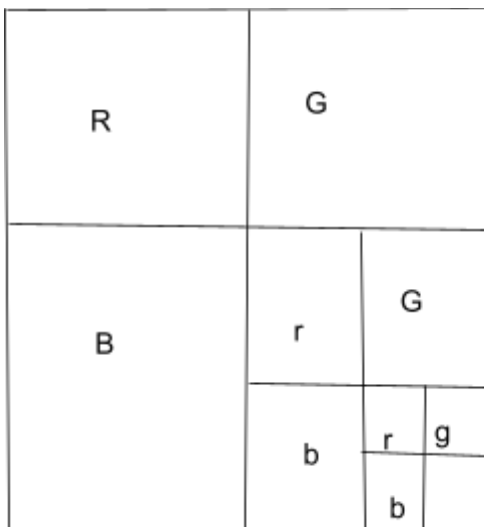
$$C_{tex}(u, v) = \hat{v} * C_1(u, v) + (1 - \hat{v}) * C_0(u, v)$$

Given four colors and their u,v-coordinates: (1, 0, 0) at (0, 0), (0, 1, 0) at (1, 0), (0, 0, 1) at (0, 1) and (0.7, 0, 1) at (1, 1), what is the resulting color after bilinear interpolation at (u, v) = (0.7, 0.7)?



What are the benefits of Mip-Maps? Show in a given diagram where and how the first x levels are stored? From 2009 exam on Freitagsrunde

2-level mipmap



- No need to calculate interpolations during runtime

Calculate the mipmap level for a footprint of size 16x13.

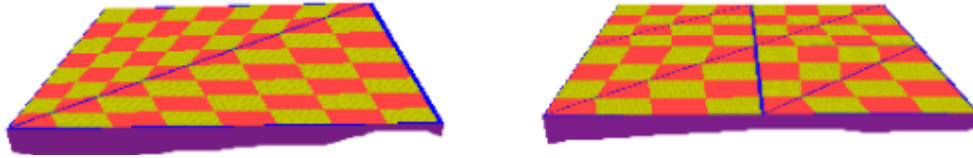
- $l = \max(16, 13) = 16 \rightarrow \log_2(16) = 4$

Why does simple (barycentric) interpolation of texture coordinates lead to wrong results during rasterization?

- Uniform steps in image space are not the same as uniform steps along lines in 3D

Name an alternative solution to perspective correct texture interpolation

- additional primitives (result in t-junktions → looks awful ;-()



How does trilinear interpolation work?

- Interpolate values of two different layers of a MipMap, obtained through bilinear interpolation on each layer

What are summed area tables (SATs)? What are they used for?

Anisotropy of a projection cannot be taken into account when using mip-maps

⇒ SATs approximate footprints using axis-parallel rectangles in (u, v) space

Is the time complexity of SATs proportional to the area of the rectangle?

- No, the mean of an arbitrary rectangle in (u, v)-space can be computed in constant time.
- But $O(n*m)$ in space complexity.

Compute a summed area table for the following square of numbers, starting at the top left corner (this task can also be posed with a partially complete SAT that you have to complete).

Compute the sum and average of the marked subrectangle.

| | | | | | |
|----|----|----|----|----|----|
| 73 | 57 | 44 | 34 | 5 | 37 |
| 59 | 87 | 56 | 32 | 37 | 57 |
| 76 | 9 | 90 | 12 | 43 | 11 |
| 4 | 43 | 78 | 65 | 73 | 17 |
| 98 | 65 | 90 | 26 | 10 | 24 |

| | | | | | |
|----|----|----|----|----|----|
| 45 | 31 | 32 | 80 | 60 | 13 |
|----|----|----|----|----|----|

Sum of all blue (dotted and continu.) - sum of only dotted

659

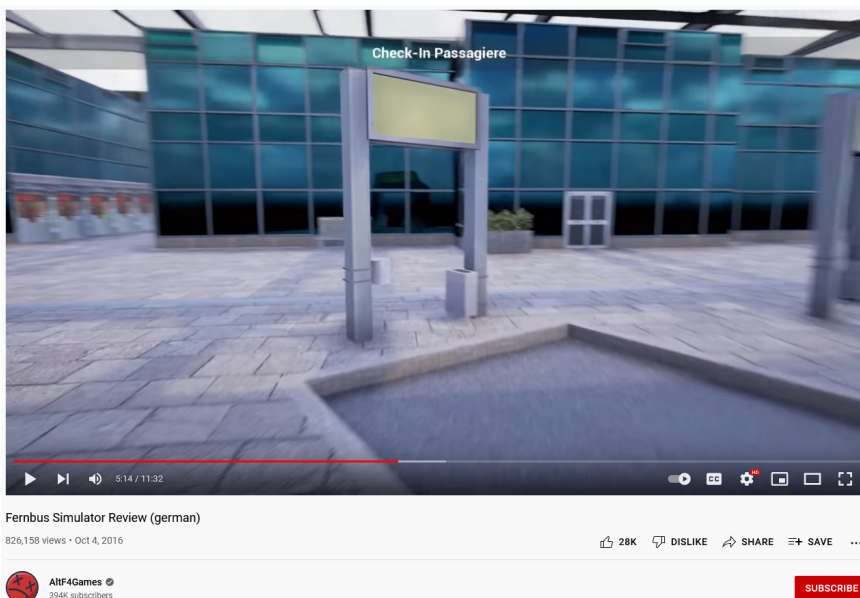
Given the points $(x_1, z_1) = (7, 1)$ and $(x_2, z_2) = (12, 4)$, where does the perspective correct interpolated point for a screen space interpolation factor of $t=0.5$ lie in world space?

p

9. What benefits do we get from light mapping? When can it be applied?

- Advantages: Precalculated light → looks nicer
- Disadvantage: only static environments

Under which conditions can environment mapping be applied? Are they satisfied for the windows in the following picture (taken from <https://youtu.be/9HIslost7IE?t=312>)?



- Small object in big scene

10. Pros and Cons of different mapping techniques

Explain how the four parameterization techniques for environment mapping that were presented in the lecture work (spherical coordinates, box map, sphere map, dual-parabolic map).

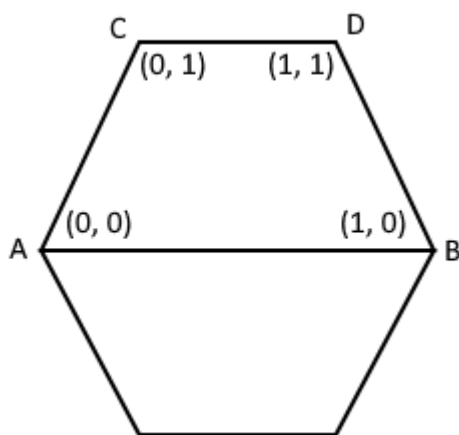
- **Spherical coordinates:** surrounding scene is stored in a texture, sphere mapping

- Disadvantages: Polar regions are tessellated finer, triangles might map weirdly
- **Box map:** Environment map consists of 6 planar textures, EM can be generated by rendering the scene from the center of the object using 3 * 2 orthogonal directions → dynamic, real time
- **Dual parabolic map:**

11. How to generate bump maps?

- High resolution geometry used to generate normals, use simpler version later and shade it using said normals

The following picture shows the top face of a cube with names and texture coordinates for the vertices. Give tangents, bitangents and normals for the faces vertices. Why do we in practice only need to store normals and tangents for any given geometry?

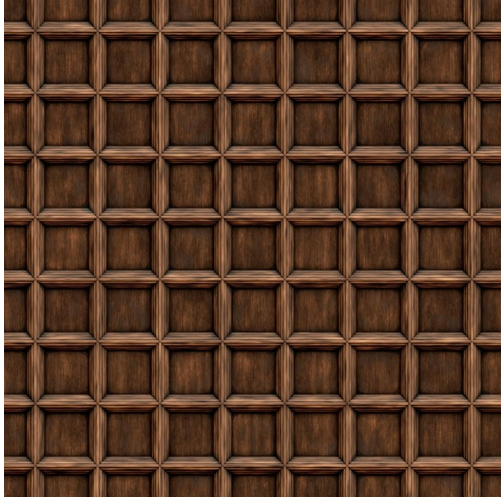


- Tangent: u-direction in world coordinates → (1, 0, 0)
- Bitangent: v-direction in world coordinates → (0, 0, -1)
- Normal: Cross Product(Tangent, Bitangent) → (0, 1, 0)

Assume you use a RGB-texture for the reflectivity of an object in a dynamically lit environment. What condition should the reflectivity map fulfill to give convincing visuals?

- Information about shadow and light should not be included in the texture (Example: photo of object includes shadows of real light and shading → might look weird)

Does the following texture fulfill the condition?



- NO

Lecture 10 Raytracing

What is the basic idea behind raytracing?

A: We simulate the exchange of light between small surfaces.

B: We calculate the color of pixels for one primitive at a time.

→ C: We follow the path of the light through the scene. ←

D: We accurately simulate physical lighting.

Why do we not follow the light rays from the light source to the eye?

- Computationally expensive (many rays never reach the eye, some light rays hit points not visible to observer)

1. What phenomenons can/can not be modeled using raytracing?

- Can:
 - reflection/mirroring
 - refraction
 - shadowing
- Cannot:
 - soft shadows
 - diffuse global illumination

2. Pros and Cons of raytracing?

- Pro
 - Refraction, reflection, shadowing
- Con
 - Slow.

3. How to model phenomenon xy with raytracing?

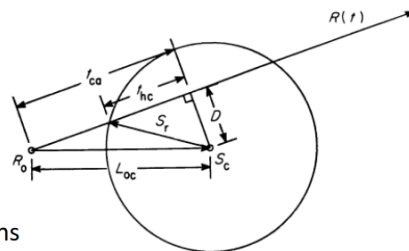
Calculate the intersection point of a cube with size 2, centered at the origin and a ray through the points $O = (0.5, 3, 0.5)$ and $P = (0, 2, 0)$.

Calculate the intersection point of a cylinder with center at $(1, 2, 3)$, radius = 1, height = 2 along y and a ray through the points $O = (1, 2, 5)$ and $P = (2, 2, 2)$.

How to calculate ray intersections?

- Sphere:

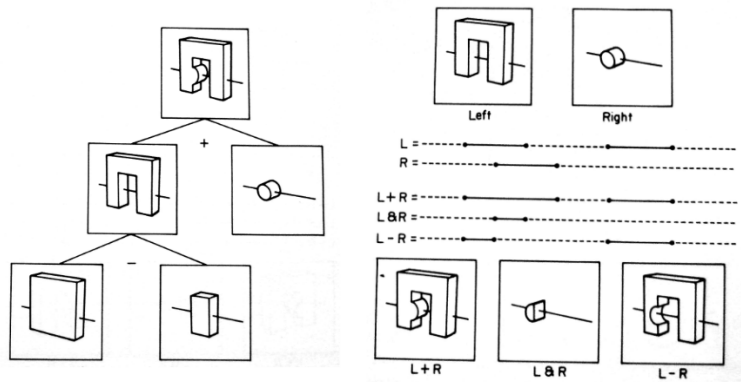
- Compare distance of ray to center with radius
 - $t_{ca} = (S_c - R_o)R_d$
 - $t_{hc}^2 = S_r^2 - D^2$
 - $D^2 = L_{oc}^2 - t_{ca}^2$
 - $\Rightarrow t_{hc}^2 = S_r^2 - L_{oc}^2 + t_{ca}^2$
 - If $t_{hc}^2 < 0$, no intersection
- Compute parameter for intersections
 - $t = t_{ca} \pm \sqrt{t_{hc}^2}$
- Intersection for ray only if $t > 0$!



- Triangle:

- Triangle V_1, V_2, V_3
- Intersection with triangle plane
 - Normal of triangle $N = (V_3 - V_1) \times (V_2 - V_1)$
 - Plane: all P for which $(P - V_1) \cdot N = 0$
 - Intersection: t so that $(R_o + t_i \cdot R_d - V_1) \cdot N = 0$
 - $t = (R_o - V_1) \cdot N / R_d \cdot N$
 - Intersection point: $S = R_o + t_i \cdot R_d$
- Is intersection point inside triangle?
 - Identical orientation of triangles $(S, V_1, V_2), (S, V_2, V_3), (S, V_3, V_1)$
 - Orientation: for example sign of $(V_1 - S) \times (V_2 - S)$

4. How do you use constructive solid geometry?



5. How can you speed up ray tracing?

- less rays: adaptive recursion rate, adaptive anti aliasing
- fewer/faster ray intersections: spatial data structures/bounding volume hierarchies, directional precomputation (light buffer)

6. What is distribution ray tracing?

- Used to generate soft shadows/diffuse lighting
- shoot multiple randomly distributed rays

7. How do you calculate motion blur/shadows/...?

- shoot rays distributed over time

8. What is aliasing? How do you minimize it?

- Hard line edges resulting from per pixel sampling
- Minimize by supersampling

9. What happens when modeling a diffuse picture?

- Looks bad with ray tracing (light bleed)

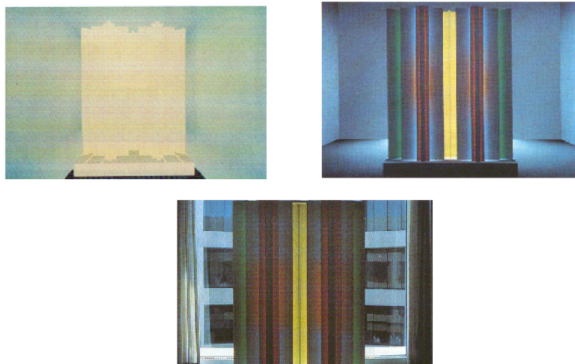
How would you integrate fog into the ray tracing calculation?

- how far is intersection point from origin? → whiten color

Lecture 11 Radiosity

Comparison to Raytracing! What can be modeled.. etc. look at the kind of questions from raytracing!

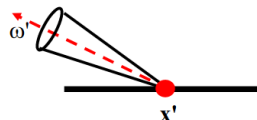
- Can:
 - Better for diffuse lighting
 - Indirect lighting/color bleeding
 - Soft shadows
 - John Ferren Daylight Experiment
- Cannot:
 - reflection



More efficient radiosity. How?

- “shooting” instead of “gathering” → useful approximate solutions emerge faster

Explain the different parts of the rendering equation: $L(x', \omega')$, $E(x', \omega')$, $\rho(x', \omega, \omega')$, $L(x, \omega)$, $G(x, x')$, $V(x, x')$.



$$L(x', \omega') = E(x', \omega') + \int \rho_{x'}(\omega, \omega') L(x, \omega) G(x, x') V(x, x') dA \leftarrow \text{discrete}$$

$$B_{x'} = E_{x'} + \underbrace{\rho_{x'} \int G(x, x') V(x, x') B_x}_{\text{Form factor}} \leftarrow \text{Continuous Equation}$$

- $L(x', \omega')$: Luminance (Leuchtdichte) at point x' in direction ω' , the “current” point.
- $L(x, \omega)$: Luminance of all other points.
- $E(x, \omega)$: Emission of the current point.
- $\rho(x', \omega, \omega')$: Reflectivity of the point, dependent on incidence and outgoing direction.
- $G(x, x')$: Geometric term: how x and x' are oriented in space relative to each other
- $V(x, x')$: Visibility of x' from x (either 0 if not visible or 1 if visible)

What assumptions does Radiosity make in regards to the reflective behavior of the surface elements?

- ρ not dependent on ω, ω' . (directions)

Arrange surface elements in regards to the amount of exchanged light. Check Slide 15!

What's the use of form factors?

- Precalculation of form factors to faster compute lighting in real time

Give the formula for calculating form factors.

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_{ij} dA_j dA_i$$

Which variables do form factors depend on?

What is the Nusselt analogy

- project surface element onto unit hemisphere and this projection onto the unit disk
⇒ Area of projection is equivalent to form factor

Explain methods to approximate the form factors.

- **Hemicube Algorithm:** Construct "Hemicube" instead of sphere, divide faces into pixels, render surface elements into hemicube faces
⇒ Form factor is the sum of pixels covered (advantage: occlusion is automatically accounted for, since we can use z-buffers to render the surface element onto the hemicube faces.)
- **Ray Casting:** Generate 4 - 32 (or more) rays between two surface elements, determine visibility, integrate point to point form factors

Explain/Understand Galerkin

- ????? (???)
- ????????????

Which assumptions does radiosity have?

Define Progressive Refinement, list advantages

- Intermediate radiosity solutions are progressively improved by incorporating more patches from most important (brighter) to least important (darker)
- Idea: "Shooting" radiosity instead of "gathering" ⇒ useful approximate solutions emerge faster
- Advantages:
 - At every step, only form factors of one element are needed (one row of form factors in matrix)
 - Looks convincing after only a few steps if few elements are emitting, since most important patches are considered first
⇒ Solutions can be computed much faster than using Gauss-Seidel
 - Smaller memory requirements, since only one matrix row is needed in each step

What simplifying assumptions are made to get from the continuous rendering equation to the discrete radiosity equation?

Lecture 12 Output

1. Name image formats and whether they are raster formats or vector formats, if they are raster formats, what kind of compression is used (none, lossless, lossy)?

- Raster:
 - JPEG: lossy, discrete cosine transformation (wavelet)
 - GIF
 - PNG: lossless
 - BMP: none?
- Vector:
 - PDF (partly)
 - SVG

2. What are the benefits and drawbacks of raster formats and vector formats? What are common applications?

- vector formats used for 2D Graphic programming e.g. a circle is a circle
- raster formats are for discrete picture elements aka pixels

How much memory does an uncompressed image of size 1920×1080 need if each channel (RGB, no alpha) is stored within one byte?

- 6.2MB

What is dithering? What is it used for?

- Reducing dimension of color space at expense of resolution (useful when only few base colors available)

What is lossless/lossy compression?

- lossless: after decompression exact original data is back
- lossy: losing certain information about the picture

What is the benefit of using color tables? Describe how color tables are generated.

- Benefits: Less memory required
- Generation: (Median Cut Algorithm) Cluster colors into similar color clouds, use mean color of each cloud to represent other colors in the same cloud

Describe how to compress an image using run length encoding. On what kind of image does RLE work best? How can you combine RLE with other compression methods to achieve better results?

- RLE: Exploit local coherence of image colors
- RLE combined with Huffman encoding is more efficient

4. What are the underlying assumptions of JPEG? Why is it a bad idea to use JPEG to encode textures storing geometric information (e.g. normal maps)?

- Few hard edges exist in photographic images → high frequencies are filtered
- Humans are more sensitive to brightness than color → colors not as important

Describe the process of encoding an image using JPEG. Which stages of the process are lossless, which are lossy?

1. converting picture to other color space YCrCb - lossless
2. chroma subsampling - lossy
3. discrete cosine transformation - lossless
4. quantisation of DCT-Coefficients - lossy
5. encoding - lossless

5. For which pictures is JPEG suitable/not suitable?

- suitable for photographs
- not suitable for fonts and "hard edges" (?)

6. What is chroma subsampling and why can we use it?

- separation of color and luminance and then reduction of color representation
- color difference signals for CrCb are averaged over small regions in the picture thus reducing size of image
- We can do that because the human visual system has higher resolution in brightness than in chroma

7. What problem of the earlier JPEG algorithm does wavelet compression fix?

- block size can be adjusted as needed according to image frequencies (harder edges → smaller blocks → sharper edges after compression)

8. JPG is a lossy compression format that reduces high frequency components in the image.

What is the motivation behind removing this kind of information?

- Human visual system is limited for high spatial frequencies