Fachpraktikum Algebra

Shading of Simplicial Surfaces

Lukas Schnelle

March 2023

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

(ii) Every edge is incident to at least one face

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

(ii) Every edge is incident to at least one face

(iii) Every vertex is incident to at least one edge

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

(ii) Every edge is incident to at least one face

(iii) Every vertex is incident to at least one edge

(iv) Every face is incident to the same number of vertices and edges

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

(ii) Every edge is incident to at least one face

(iii) Every vertex is incident to at least one edge

(iv) Every face is incident to the same number of vertices and edges

(v) For every face there is a enumeration of all incident vertices $(v_1, \ldots, v_n)$ and edges $(e_1, \ldots, e_n)$ such that:

$$\forall 1 \leq i < n : v_i, v_{i+1} \text{ incident to } e_i$$

Simplicial Surface

Definition (Polygonal Complexes)

A set consisting of three disjoint sets of vertices, edges and faces with an incidence relation is called a polygonal complex if the following hold:

(i) Every edge has exactly two incident vertices.

(ii) Every edge is incident to at least one face

(iii) Every vertex is incident to at least one edge

(iv) Every face is incident to the same number of vertices and edges

(v) For every face there is a enumeration of all incident vertices $(v_1, \ldots, v_n)$ and edges $(e_1, \ldots, e_n)$ such that:

$$\forall 1 \leq i < n : v_i, v_{i+1} \text{ incident to } e_i$$

In general we allow so called boundary edges, that means that an edge can be incident to only one face.

Simplicial Surface

Definition (Umbrella Path)

Let v a vertex in a polygonal complex, v has n incident vertices and edges, $P := (e_1, f_1, e_2, f_2, \ldots, e_n, f_n)$ with e_i edges, f_i faces.
Then we call P an umbrella path if the following hold: $\forall 1 \leq i, j \leq n$

Simplicial Surface

Definition (Umbrella Path)

Let v a vertex in a polygonal complex, v has n incident vertices and edges, $P := (e_1, f_1, e_2, f_2, \ldots, e_n, f_n)$ with e_i edges, f_i faces.
Then we call P an umbrella path if the following hold: $\forall 1 \leq i, j \leq n$

(i) $i \neq j \implies f_i \neq f_j, e_i \neq e_j$

Simplicial Surface

Definition (Umbrella Path)

Let v a vertex in a polygonal complex, v has n incident vertices and edges, $P := (e_1, f_1, e_2, f_2, \ldots, e_n, f_n)$ with e_i edges, f_i faces.
Then we call P an umbrella path if the following hold: $\forall 1 \leq i, j \leq n$

(i) $i \neq j \implies f_i \neq f_j, e_i \neq e_j$

(ii) f_i is incident to v

(iii) e_i is incident to v

Simplicial Surface

Definition (Umbrella Path)

Let v a vertex in a polygonal complex, v has n incident vertices and edges, $P := (e_1, f_1, e_2, f_2, \ldots, e_n, f_n)$ with e_i edges, f_i faces.
Then we call P an umbrella path if the following hold: $\forall 1 \leq i, j \leq n$

(i) $i \neq j \implies f_i \neq f_j, e_i \neq e_j$

(ii) f_i is incident to v

(iii) e_i is incident to v

(iv) e_i, e_{i+1} are incident to f_i

Simplicial Surface

Definition (Umbrella Path)

Let v a vertex in a polygonal complex, v has n incident vertices and edges, $P := (e_1, f_1, e_2, f_2, \ldots, e_n, f_n)$ with e_i edges, f_i faces.
Then we call P an umbrella path if the following hold: $\forall 1 \leq i, j \leq n$

(i) $i \neq j \implies f_i \neq f_j, e_i \neq e_j$

(ii) f_i is incident to v

(iii) e_i is incident to v

(iv) e_i, e_{i+1} are incident to f_i

If the vertex does not have a incident boundary edge, this sequence can be viewed as a cycle with $e_1 = e_n$. Otherwise it is unique up to reversal.

Simplicial Surface

Definition

A polygonal complex C is called simplicial surface, if all faces are triangular and for all vertices there exists an umbrella path in C.

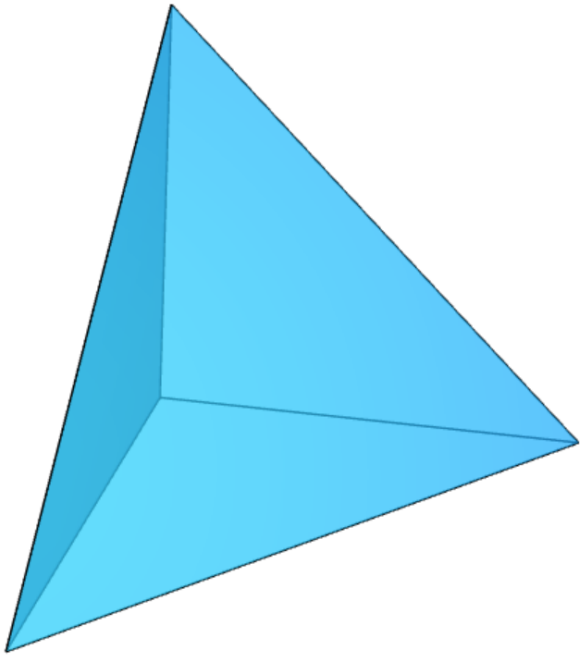
Simplicial Surface

Definition

A polygonal complex C is called simplicial surface, if all faces are triangular and for all vertices there exists an umbrella path in C.

Example

Positive



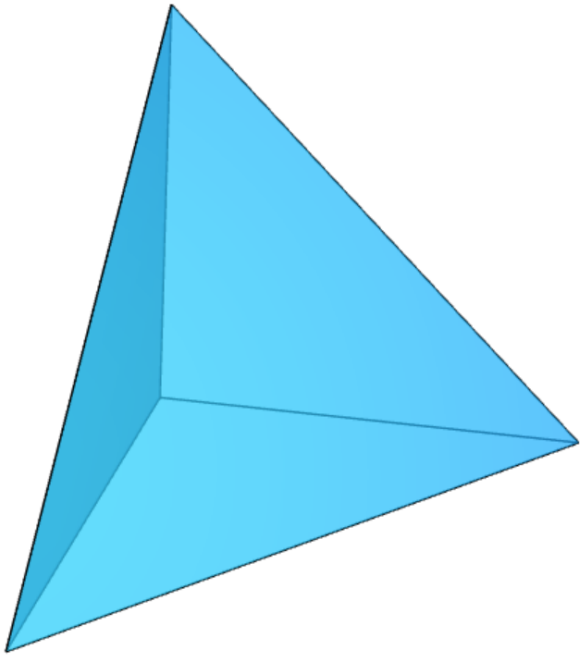
Simplicial Surface

Definition

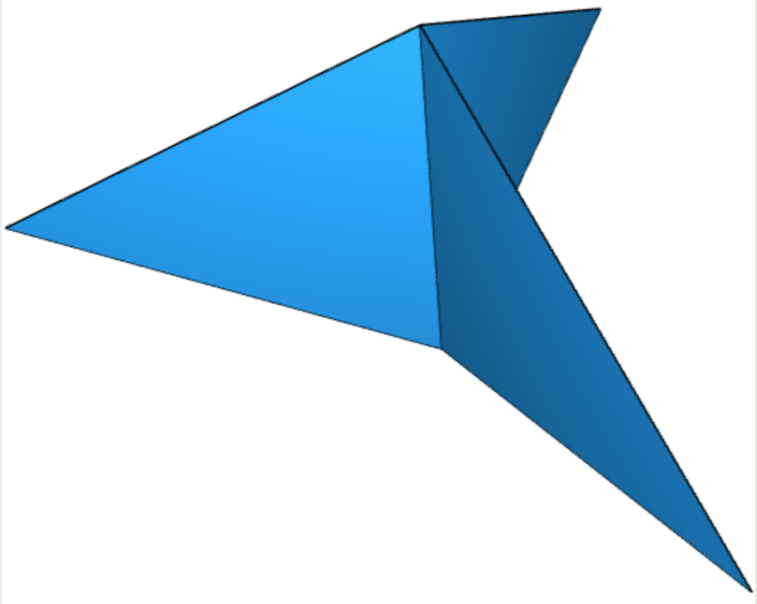
A polygonal complex C is called simplicial surface, if all faces are triangular and for all vertices there exists an umbrella path in C.

Example

Positive



Negative



Shading

Goal of rendering

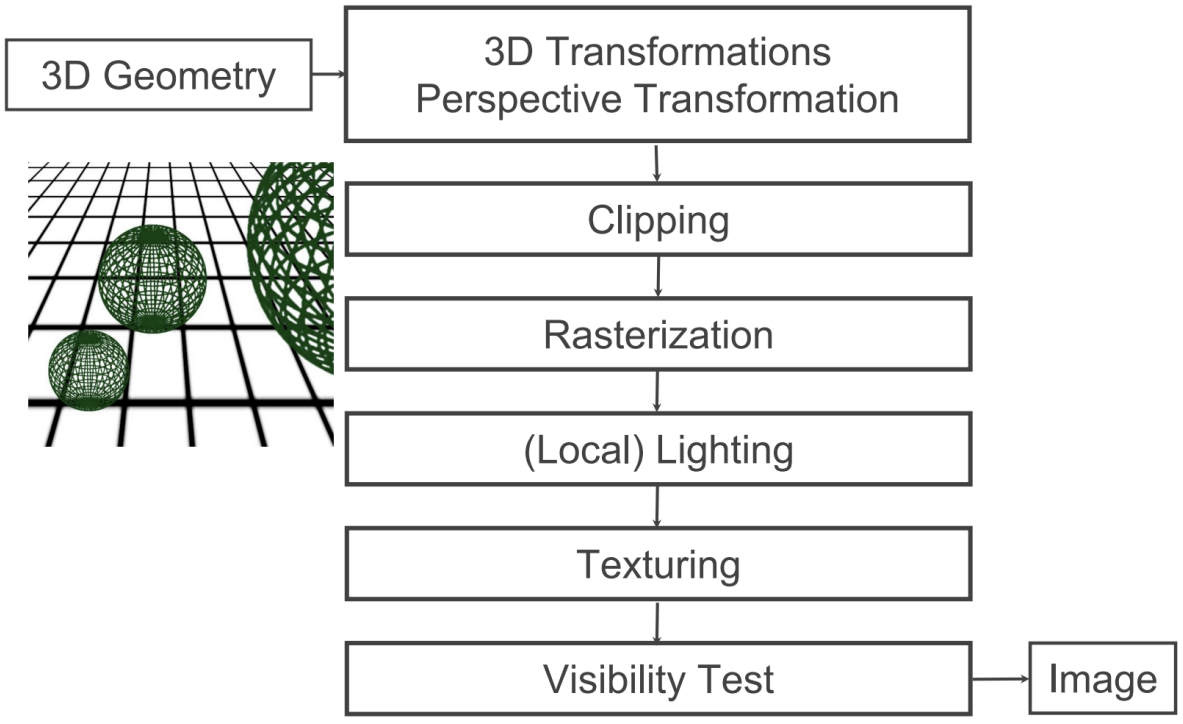
Generate a 2D projection of a 3D geometry

Shading

Goal of rendering

Generate a 2D projection of a 3D geometry

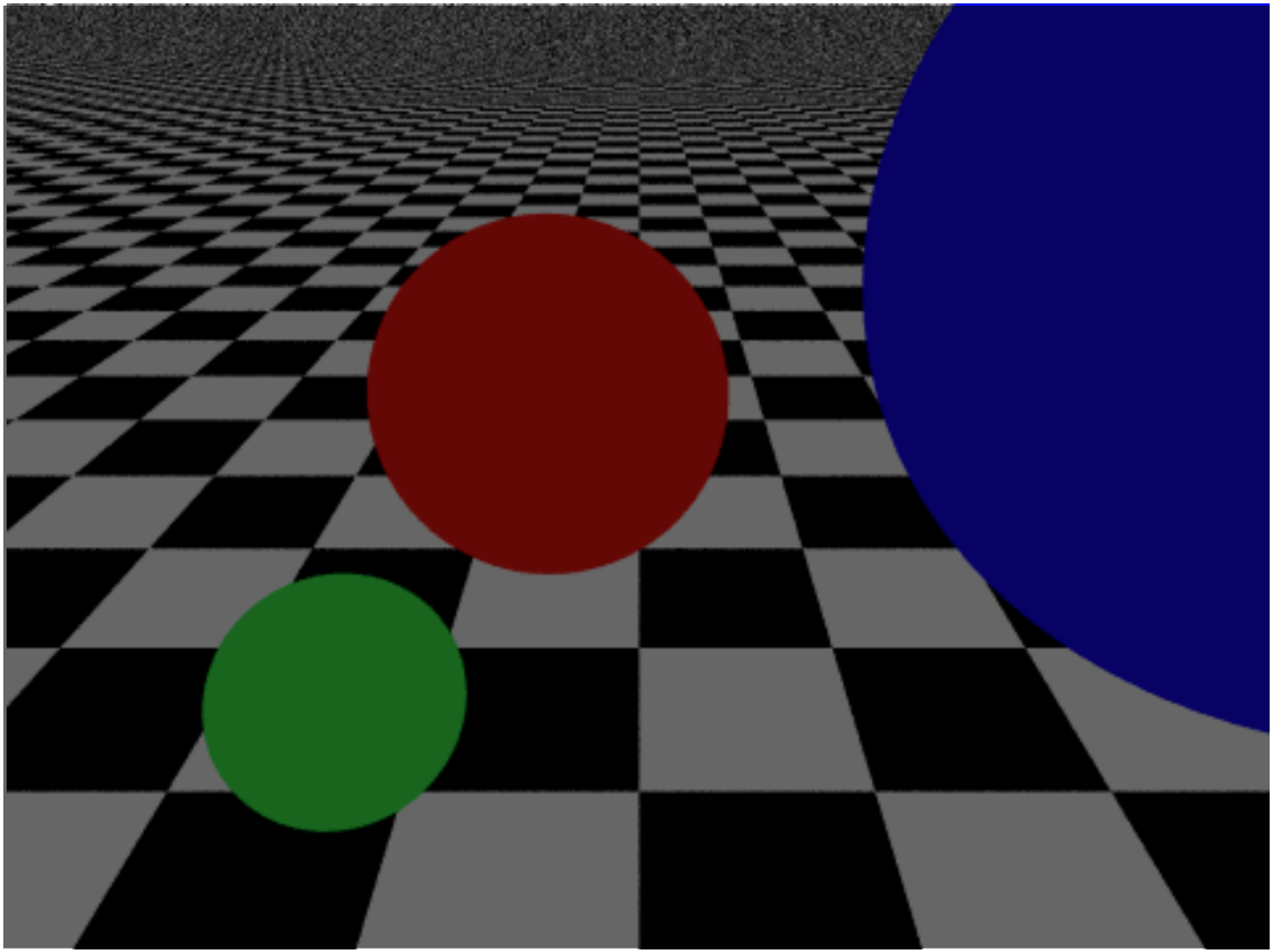
Rendering Pipeline



[2]

Shading

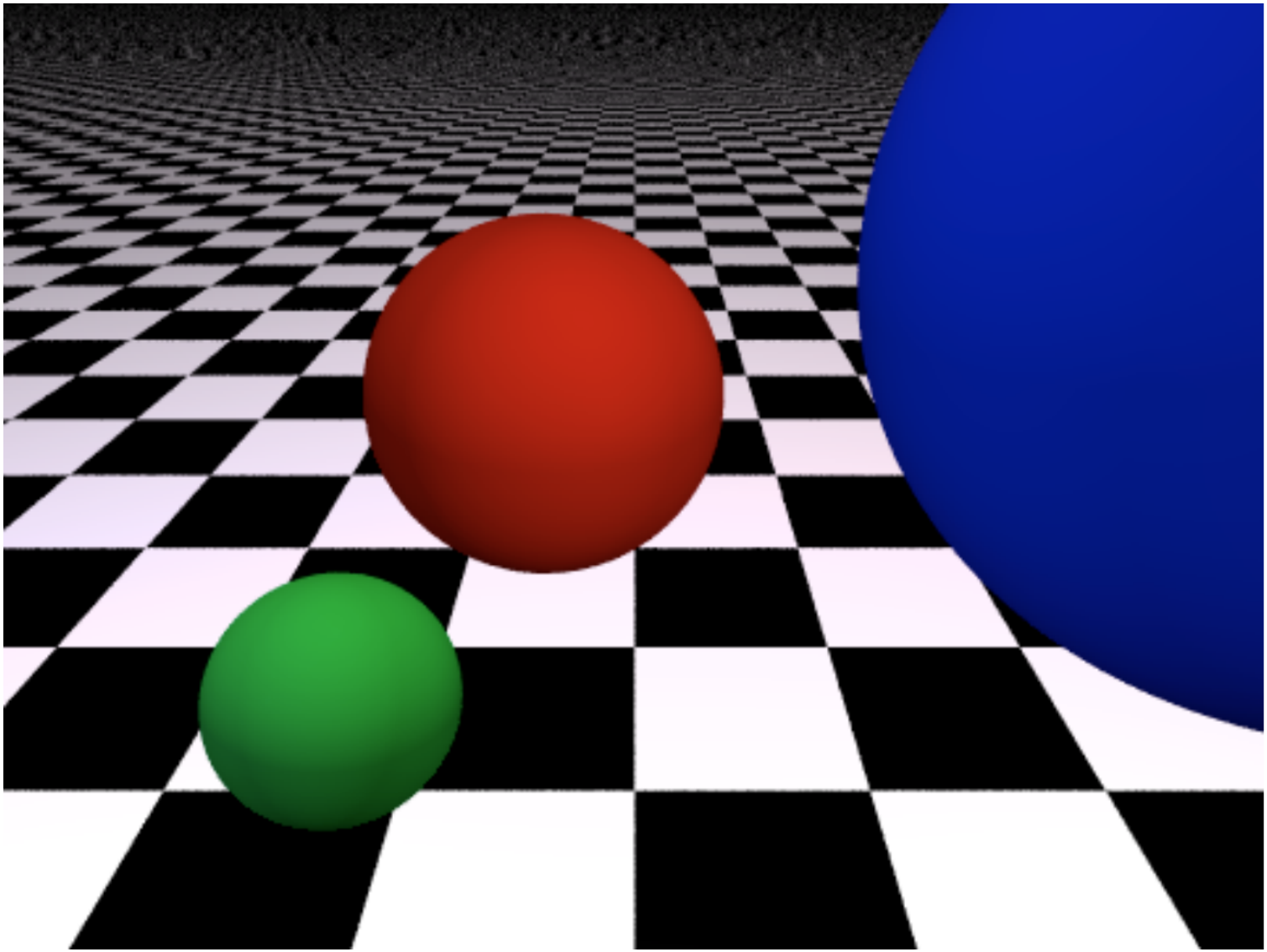
Only Ambient Term



[2]

Shading

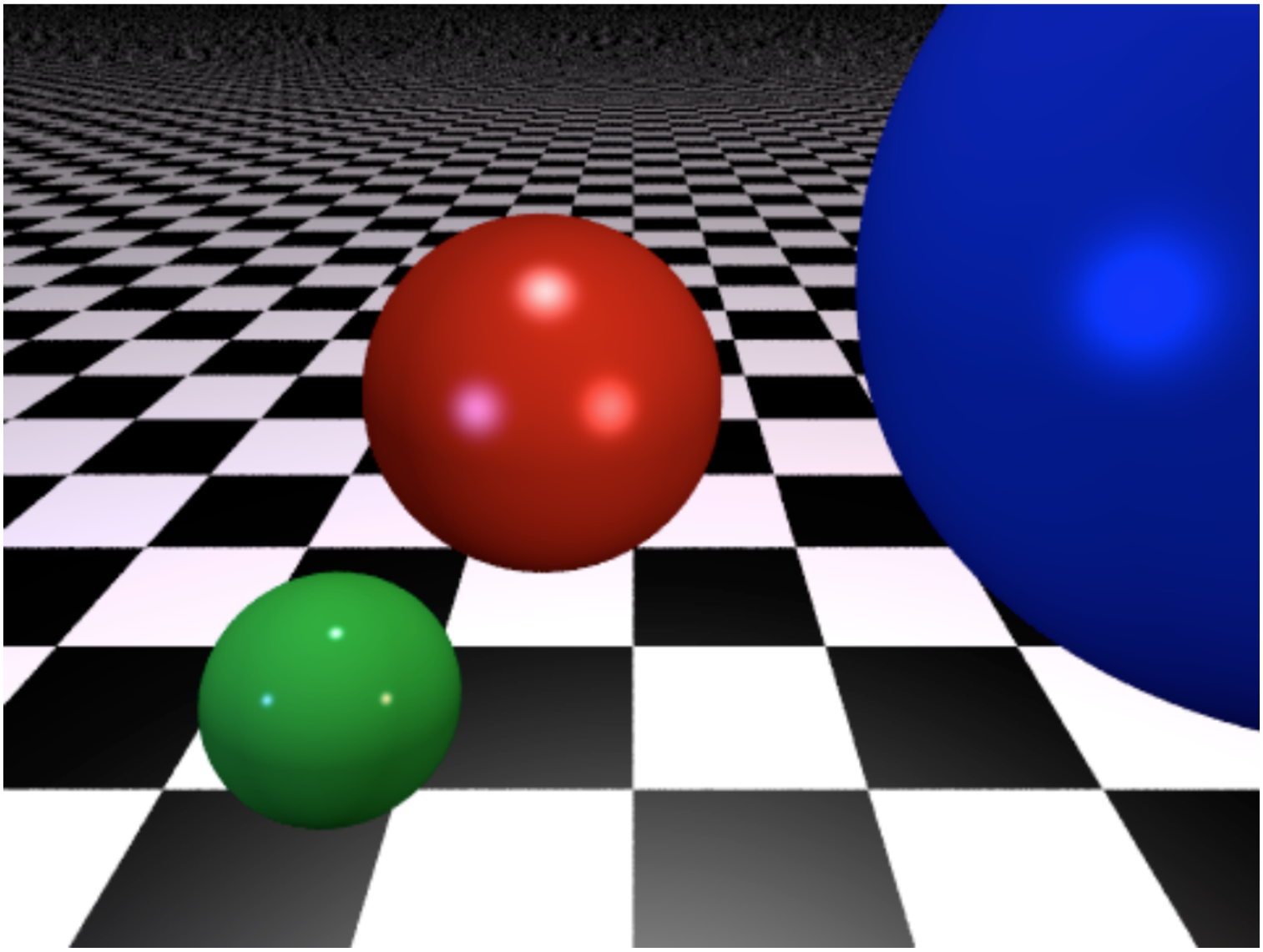
Ambient plus diffuse



[2]

Shading

Ambient plus diffuse plus specular



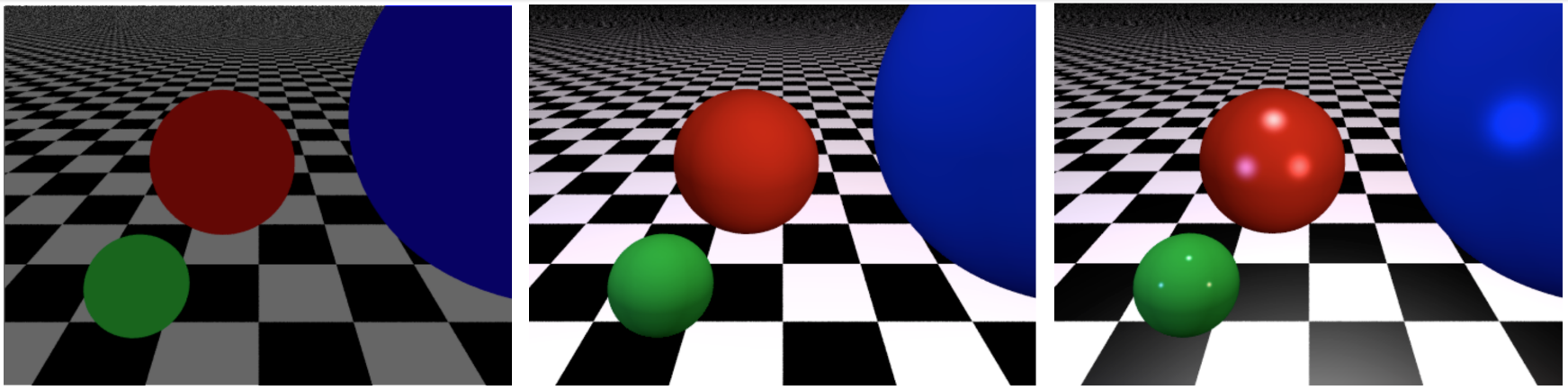
[2]

Shading

(Local) Lighting

Which color does a pixel get?

- Ambient: Independent of location of light and viewer
- Diffuse: Accounts for where the light is
- Specular: Accounts for where the light and the camera are

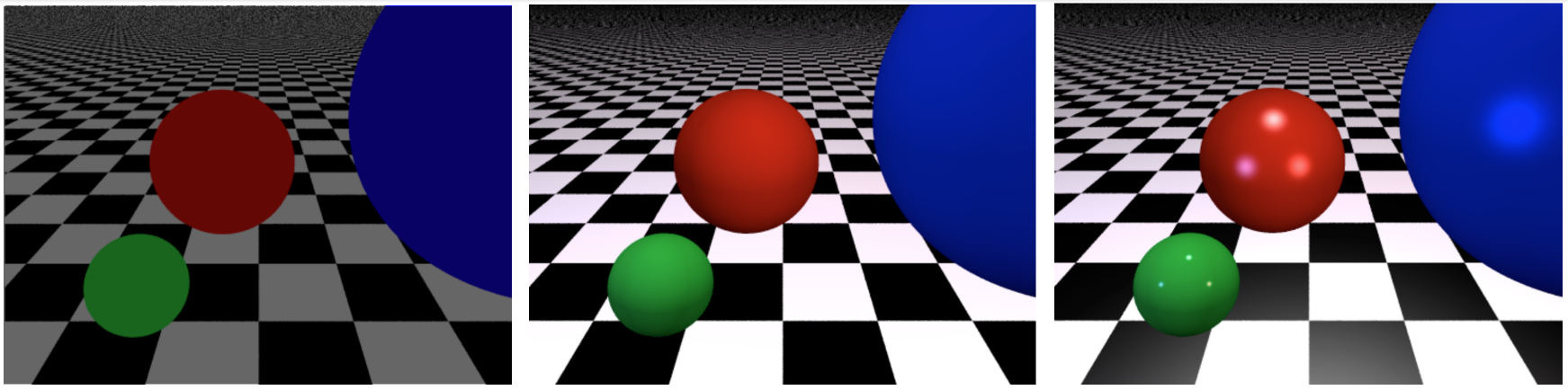


Shading

(Local) Lighting

Which color does a pixel get?

- Ambient: Independent of location of light and viewer
- Diffuse: Accounts for where the light is
- Specular: Accounts for where the light and the camera are



Formula

$$\underbrace{C_a}_{\text{color of polygon}} + \underbrace{C_d(p,n,l)}_{\text{color of light}} + \underbrace{C_{sp}(p,n,l,c)}_{\text{color of light}}$$

Shading

Different models for diffuse and specular term. Decided on Phong model, already in three.js and fairly efficient.

Idea

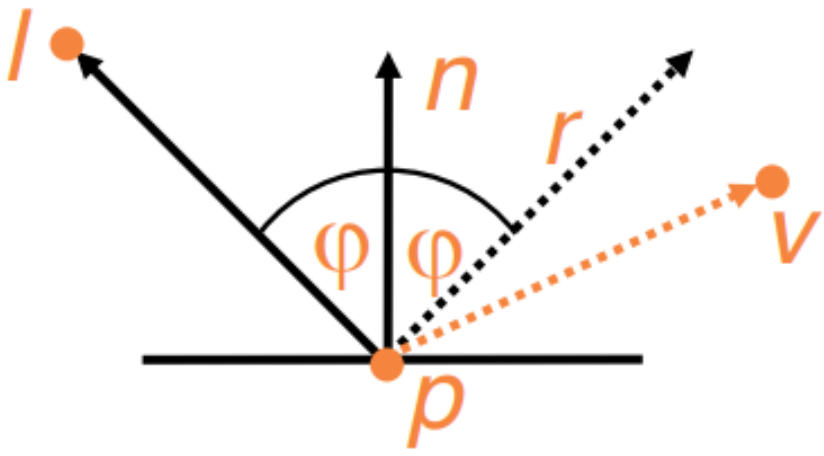
Objects reflect light the most if lightsource and camera are in the same angle w.r.t. the normal of the surface/polygon.

Shading

Different models for diffuse and specular term. Decided on Phong model, already in three.js and fairly efficient.

Idea

Objects reflect light the most if lightsource and camera are in the same angle w.r.t. the normal of the surface/polygon.

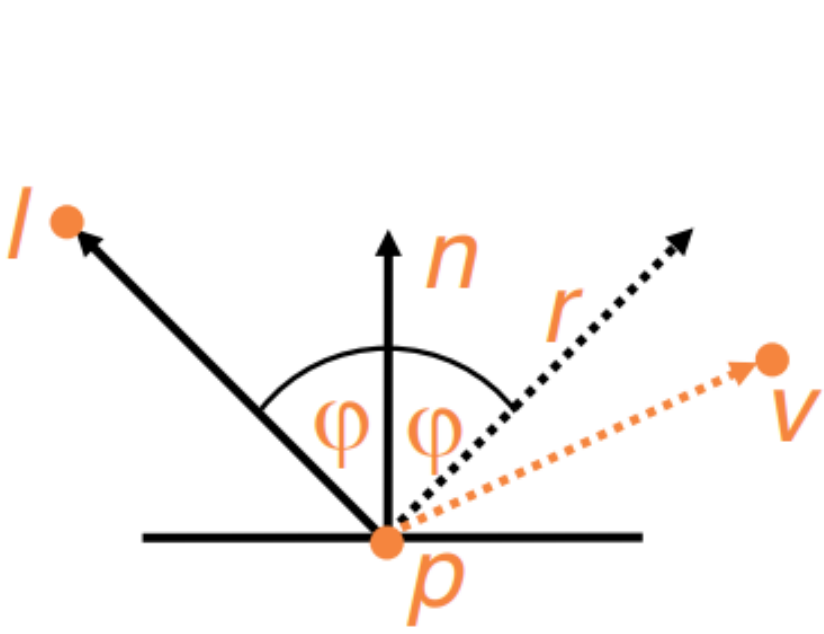


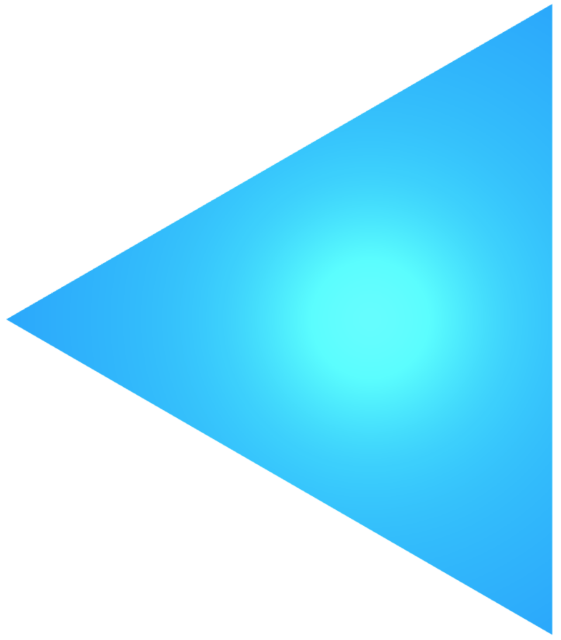
Shading

Different models for diffuse and specular term. Decided on Phong model, already in three.js and fairly efficient.

Idea

Objects reflect light the most if lightsource and camera are in the same angle w.r.t. the normal of the surface/polygon.





[2]

What we call shading here, is in Computer Graphics called local lighting.

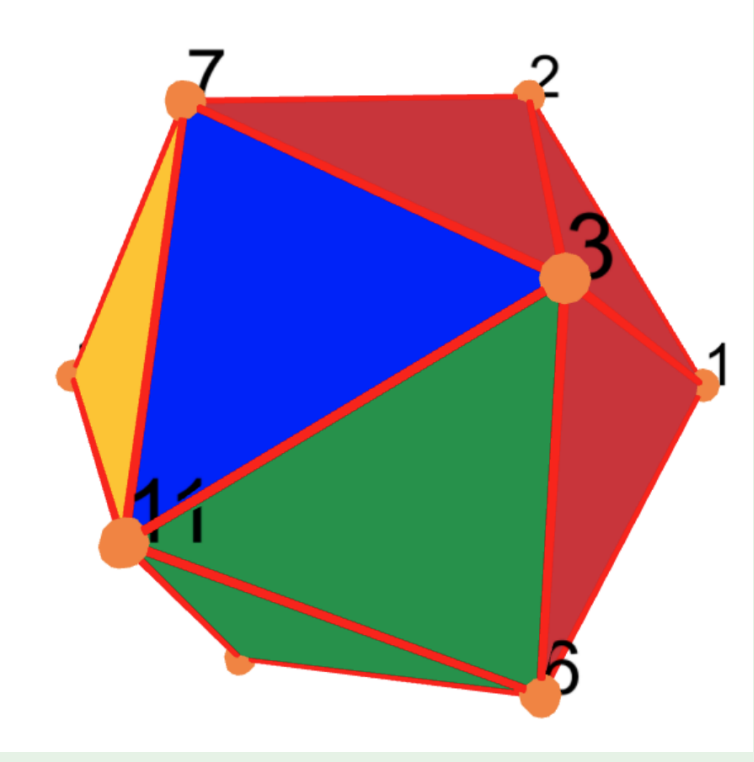
Simplicial Surfaces Package

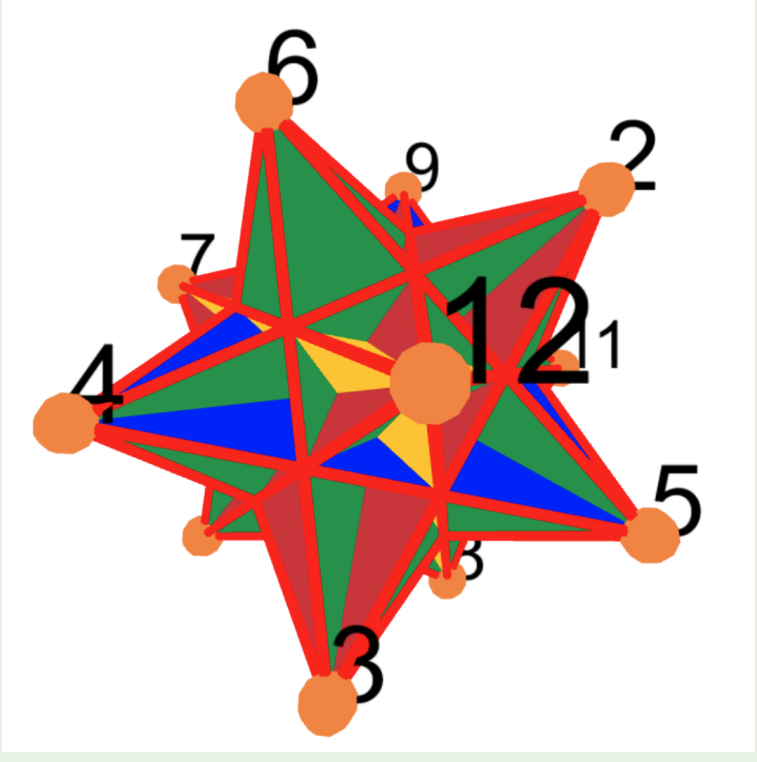
- Package for the GAP Programming language
- Has functionality for displaying surfaces
    - Generates a .html file
    - Uses three.js

Simplicial Surfaces Package

- Package for the GAP Programming language
- Has functionality for displaying surfaces
  - Generates a .html file
  - Uses three.js

Example (Number 2.1 and 2.2 from [1])





Goal

Goal

Implement shading in the visualizations of the Simplicial Surface package.

Goal

Goal

Implement shading in the visualizations of the Simplicial Surface package.

First Approach: Implement directly
$\rightarrow$ Learn how the output is generated
Uses a class called `THREE.Geometry`

Goal

Goal

Implement shading in the visualizations of the Simplicial Surface package.

First Approach: Implement directly
$\rightarrow$ Learn how the output is generated
Uses a class called `THREE.Geometry`
After some work it turns out: class is deprecated.

THREE.Geometry will be removed from core with r125

Discussion geometry

Mugen87 3 Jan '21

The upcoming release `r125` will contain a major, potentially breaking change. The class `THREE.Geometry` will be no longer part of the core but moved to `jsm/deprecated/Geometry.js`. It will only be available as an ES6 module and not as a global script.

Workflow

But: In newer revisions of three.js shading is already implemented.
$\rightarrow$ After some promisiong tests: Decided to rewrite the entire function.

Demo

We need to switch to the browser for this.
For one example we use [1]

Results

ToDos

- Some of the functions are not yet implemented again

Results

ToDos

- Some of the functions are not yet implemented again

Advantages

- New security requirements of javascript and modern browsers: need to load the code from server $\rightarrow$ way smaller file sizes
  (for small examples 9kB vs. 539kB)

Results

ToDos

- Some of the functions are not yet implemented again

Advantages

- New security requirements of javascript and modern browsers: need to load the code from server $\rightarrow$ way smaller file sizes
  (for small examples 9kB vs. 539kB)
- More efficient Animations, faster loading, fewer memory
  (Demo in Browser)

Results

ToDos

- Some of the functions are not yet implemented again

Advantages

- New security requirements of javascript and modern browsers: need to load the code from server $\rightarrow$ way smaller file sizes
  (for small examples 9kB vs. 539kB)
- More efficient Animations, faster loading, fewer memory
  (Demo in Browser)
- Also works for triangular complexes
  $\rightarrow$Does not depend on incidence structure for visualization
  (Demo in Browser)

Future

- More functions in the GUI, e.g.
  - Turning the vertices on and off
  - Changing location of a vertex on the fly
- More options materials
  e.g. Color dependent on the normal of the polygon
  (Demo in browser)
- Intersection planes
  (Demo in browser)

Thank You for your attention

Are there Questions?

[1] Karl-Heinz Brakhage et al. *The icosahedra of edge length 1*. 2019. DOI: 10.48550/ARXIV.1903.08278. URL: https://arxiv.org/abs/1903.08278.

[2] Prof. Leif Kobbelt. *Lecture slides: Basic Techniques in Computer Graphics*. WS 2022/2023.