

Computergraphik

Computergraphik

Übung: Besprechung 1. Übungsblatt

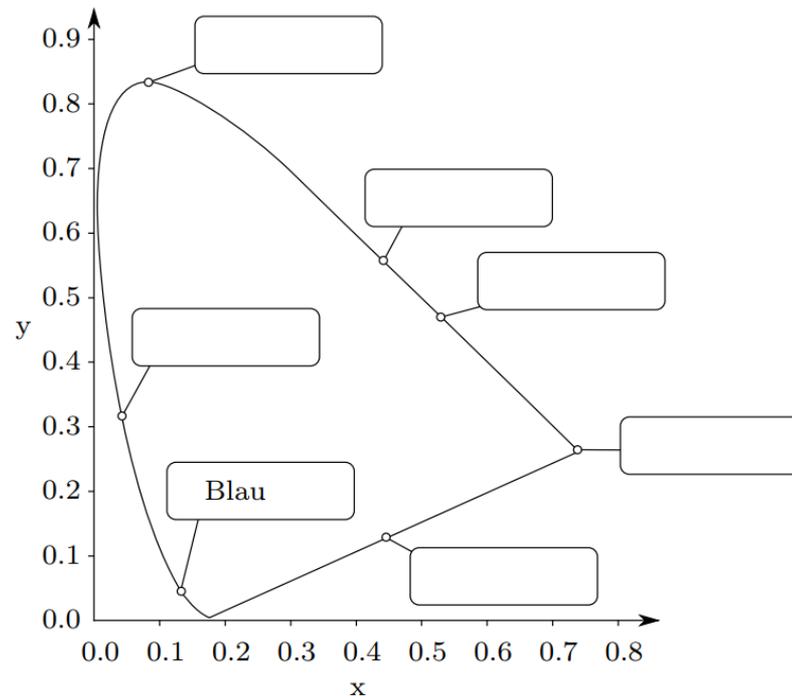
Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie



a) Tragen Sie die Farben

Grün, Rot, Gelb, Orange, Cyan, Magenta

in die entsprechenden Felder im Chromatizitätsdiagramm ein.



b) Welcher der Farbeindrücke aus Aufgabe a) lässt sich nicht durch monochromatisches Licht erzeugen?

- c) Die Farbräume xyY und XYZ sind eng verwandt. Wie ist der mathematische Zusammenhang zwischen der Chromatizität (x, y) und der passenden Farbe (X, Y, Z) ?
- d) Gegeben sind
- 1) der Farbraum **XYZ**,
 - 2) ein physikalisch realisierbarer **RGB**-Farbraum,
 - 3) sowie der Raum aller Farben, die durch **100 monochromatische Leuchtdioden** mit den äquidistanten Wellenlängen $\{380\text{nm}, 384\text{nm}, \dots, 776\text{nm}\}$ darstellbar sind.

Ordnen Sie diese Räume aufsteigend nach der Größe ihres für den Menschen sichtbaren Gamut.

Allgemeine Hinweise



- ▶ An Spezifikationen halten!
- ▶ Keine neuen Annahmen treffen.
- ▶ In `generate_grid/generate_strip` nicht zeichnen!
- ▶ `draw_triangles` soll allgemein funktionieren
- ▶ Richtige Funktionen verwenden (`glDrawElements` wo gefordert...)

Allgemeine Hinweise



- ▶ Es gibt mehrere Versionen von abs:
 - ▶ Integer: `std::abs`
 - ▶ Float: `std::fabs`, `fabsf`
 - ▶ Double: `std::fabs`, `fabs`

- ▶ Schaut euch im Zweifelsfall eine C++-Referenz an!
- ▶ Ebenso: OpenGL-Referenz lesen!

- ▶ Schleifenvariablen sollten Integer sein (nicht float)
 - ▶ Float führt zu numerischen Fehlern

```
const int N = 1023;

std::vector<float> v0(N+1);
for (int i = 0; i <= N; ++i) {
    v0[i] = float(i)/N;
}

std::vector<float> v1;
for (float f = 0; f <= 1.0f; f += 1.0f/N) {
    v1.push_back(f);
}

std::cout << v0.back() << std::endl;
std::cout << v1.back() << std::endl;
```

```
const int N = 1023;

std::vector<float> v0(N+1);
for (int i = 0; i <= N; ++i) {
    v0[i] = float(i)/N;
}

std::vector<float> v1;
for (float f = 0; f <= 1.0f; f += 1.0f/N) {
    v1.push_back(f);
}

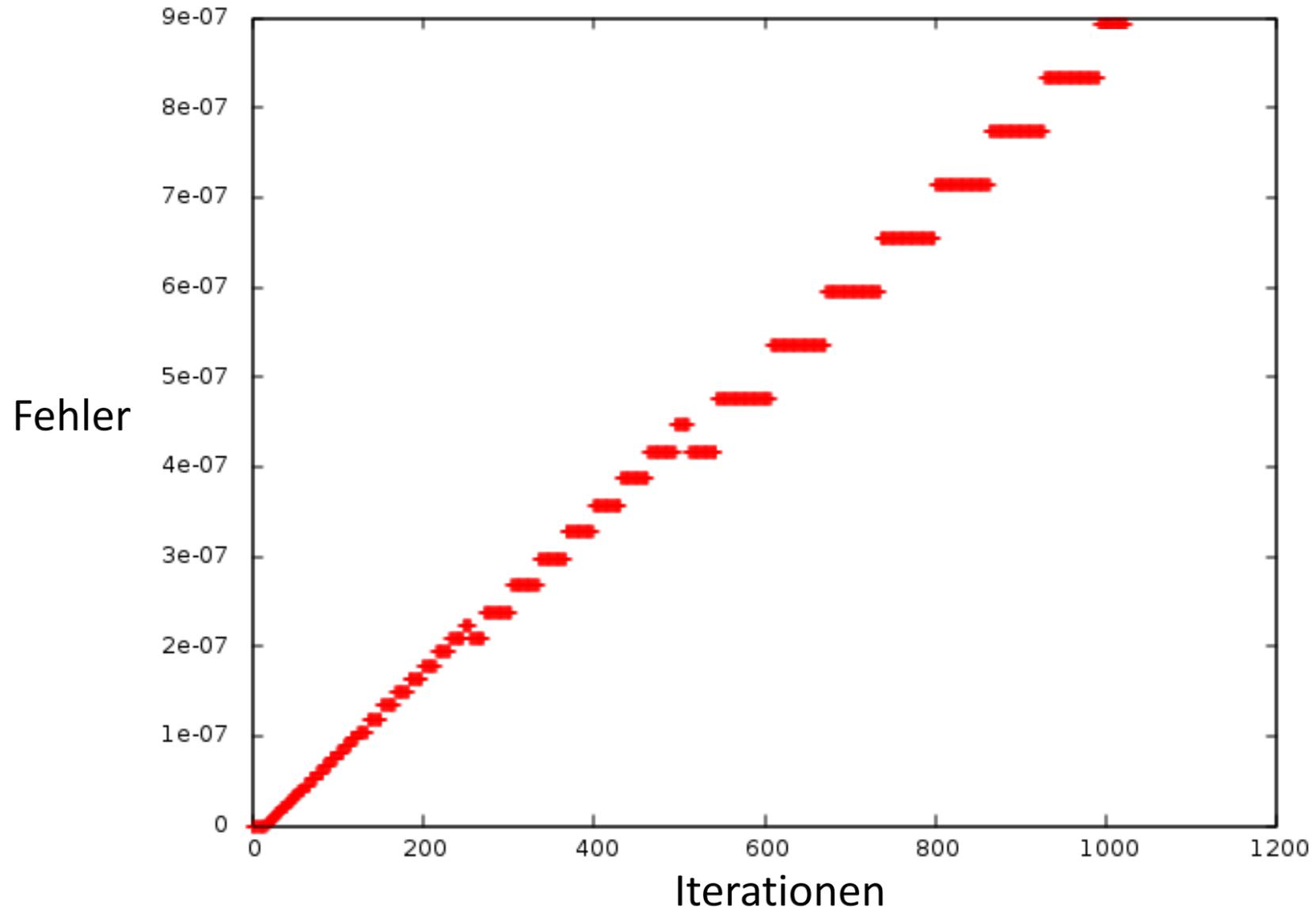
std::cout << v0.back() << std::endl;
std::cout << v1.back() << std::endl;
```

▶ Ausgabe:

1

0.999999

Numerische Ungenauigkeiten

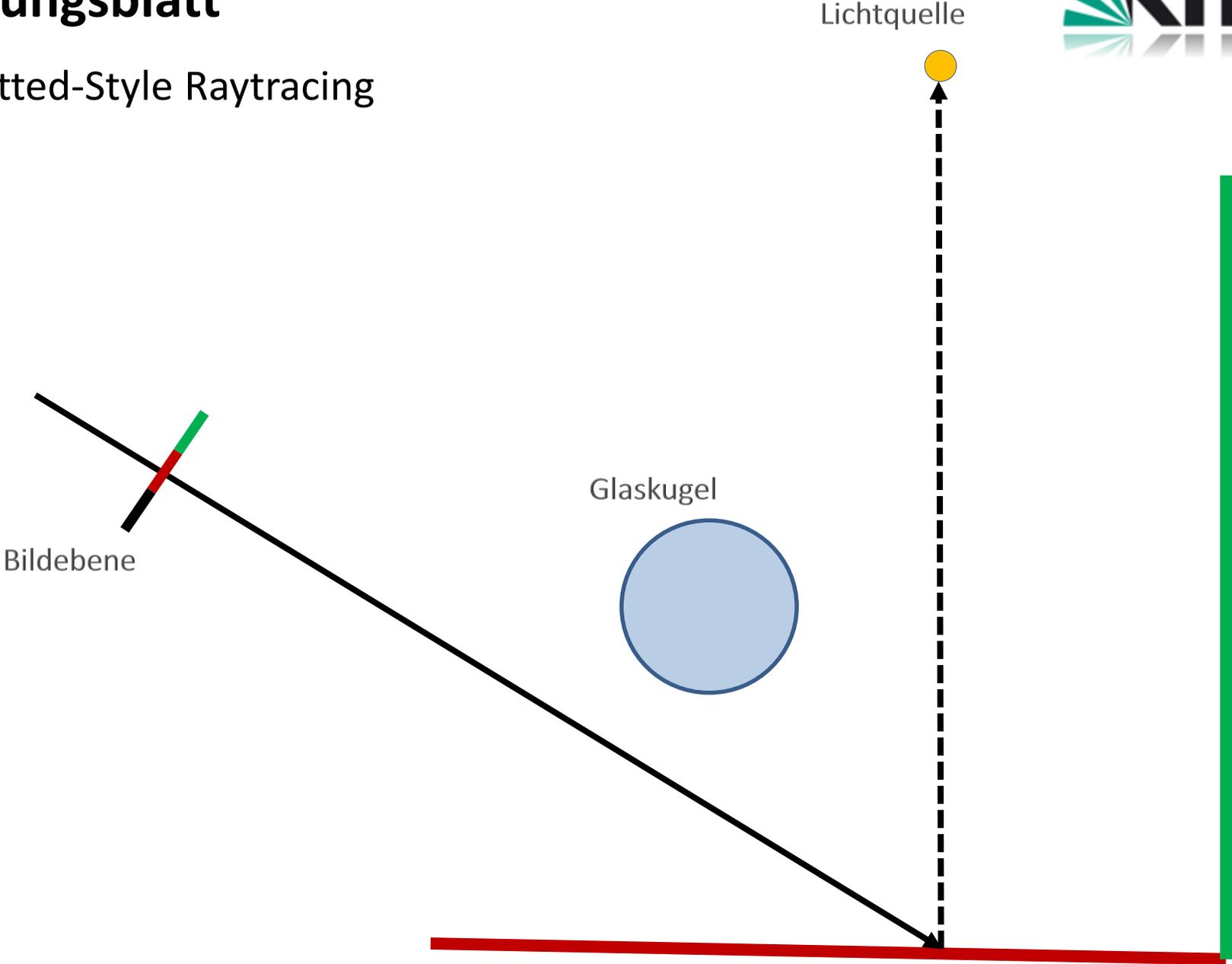


2. Übungsblatt

2. Übungsblatt

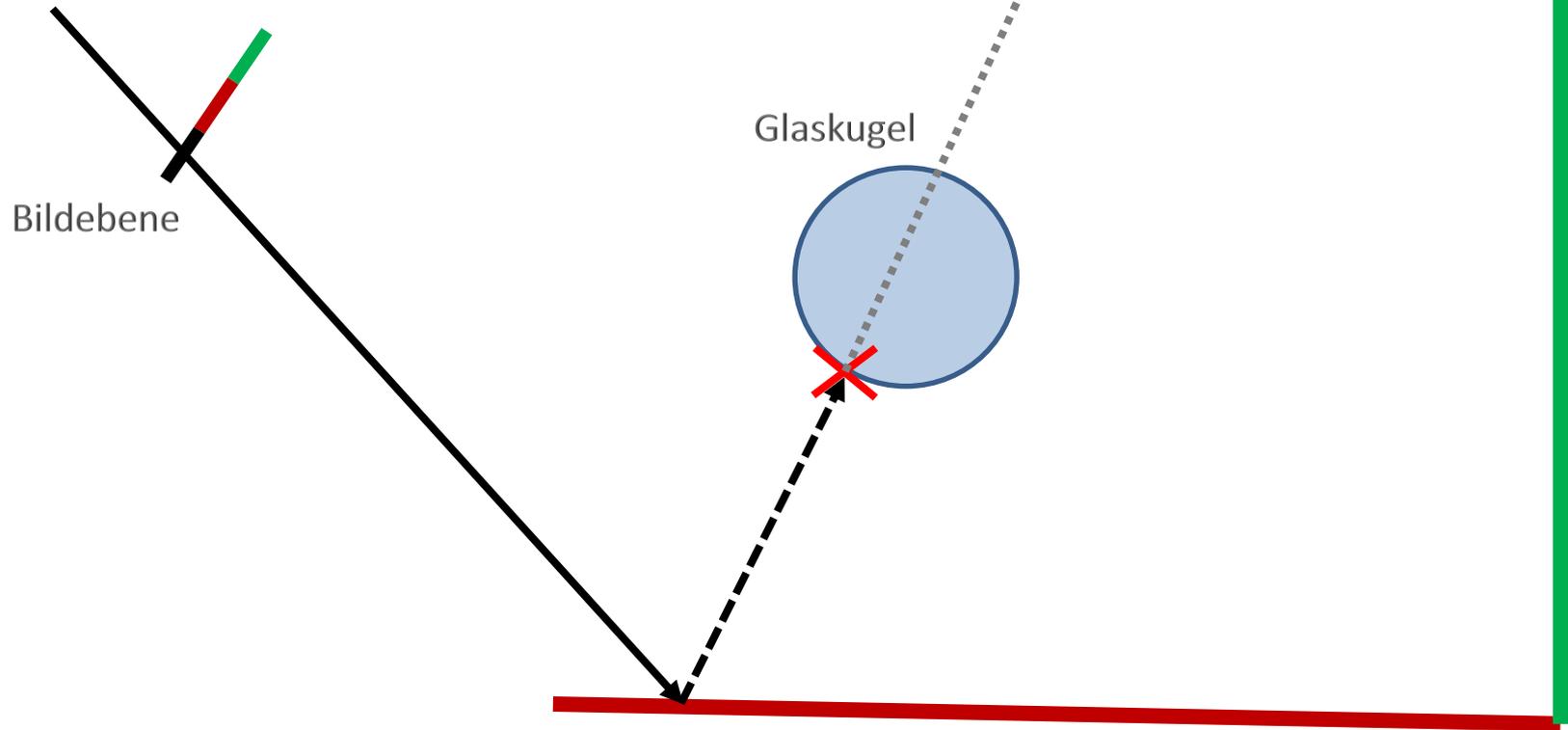


▶ Whitted-Style Raytracing



2. Übungsblatt

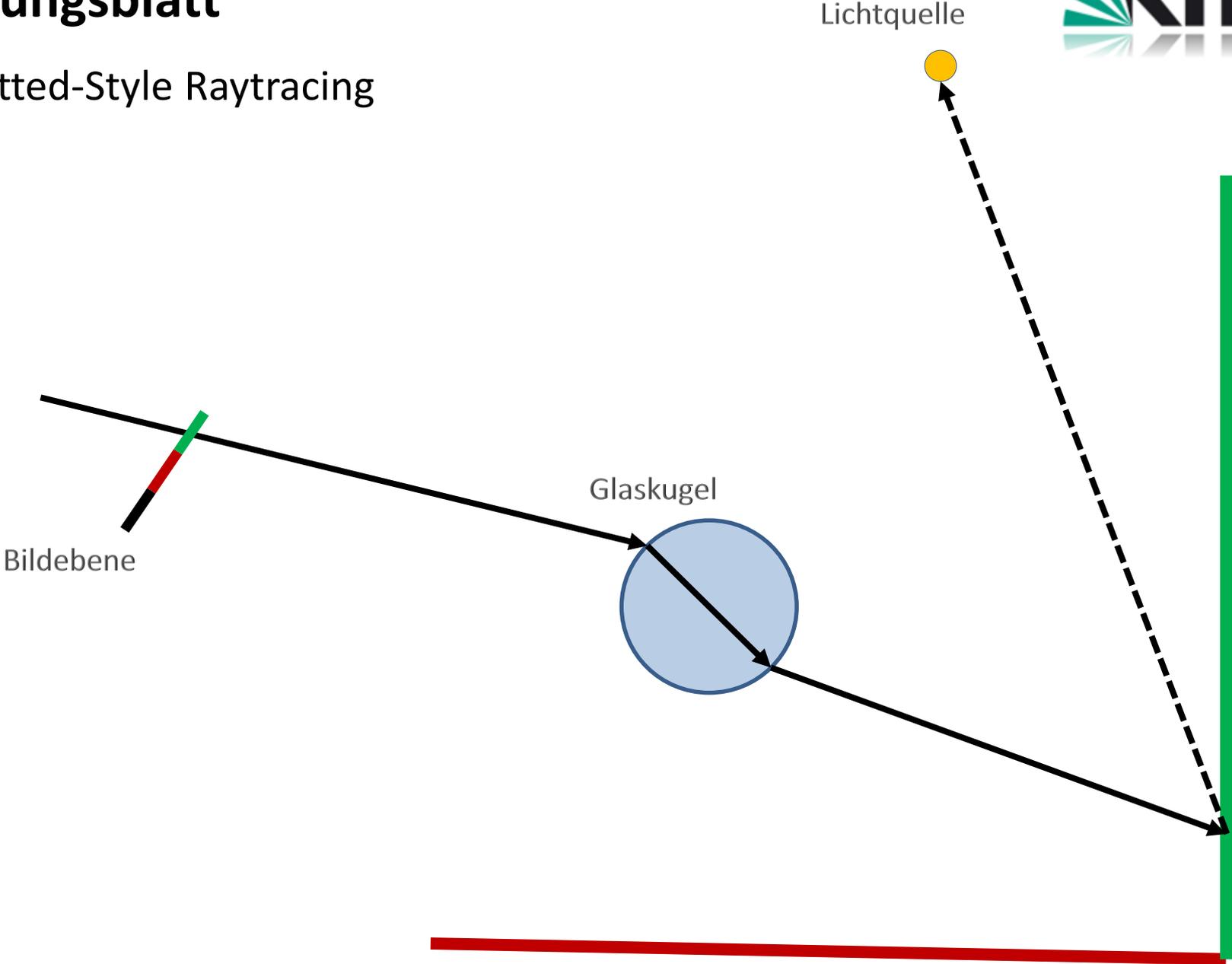
▶ Whitted-Style Raytracing



2. Übungsblatt

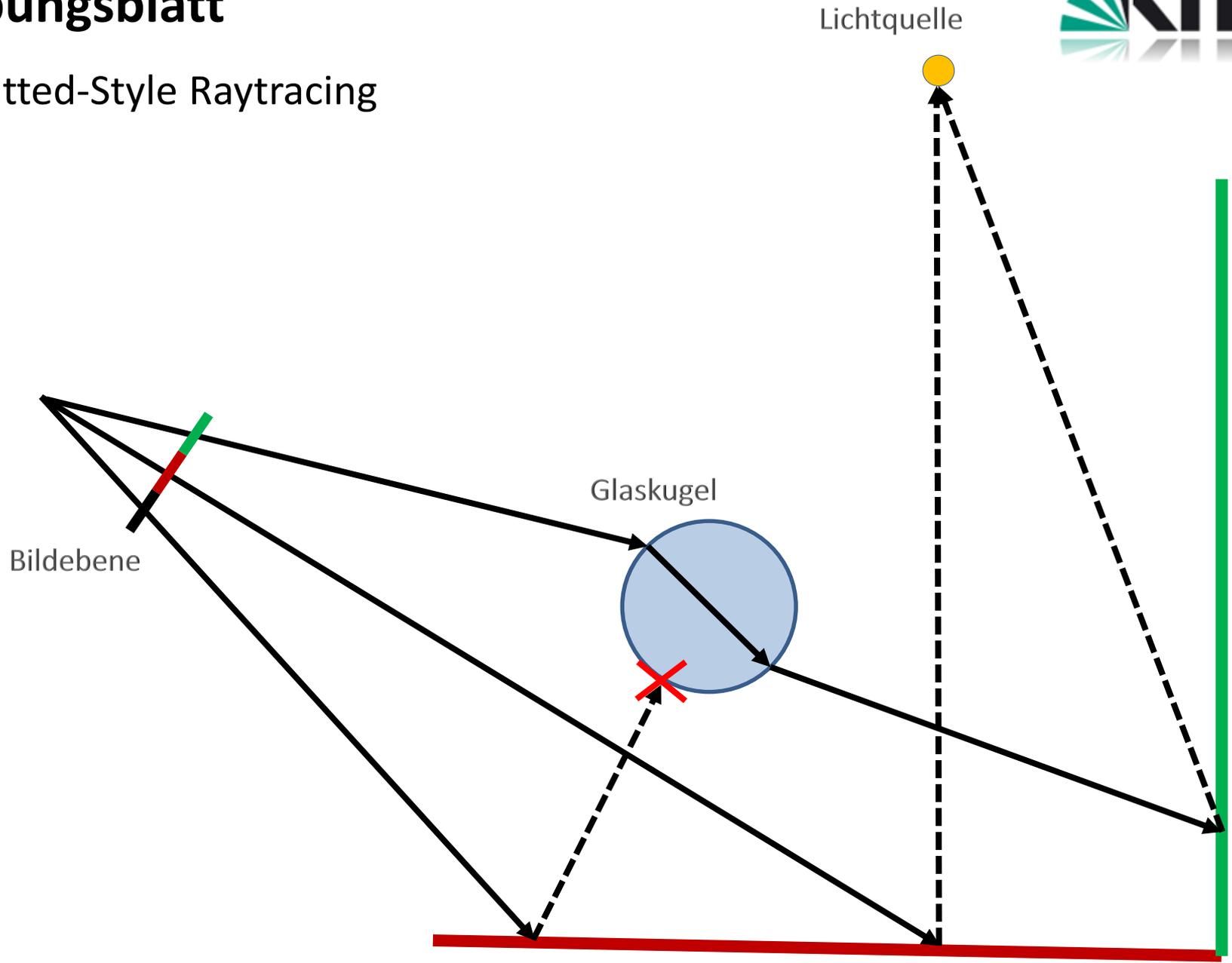


▶ Whitted-Style Raytracing



2. Übungsblatt

▶ Whitted-Style Raytracing



Hinweise zum neuen Übungsblatt

- ▶ Sämtliche Kombinationen testen (Schatten an/aus, Reflexionen an/aus,...)
- ▶ Rekursionen brauchen ein gültiges Ende, sonst entsteht ein Stack Overflow (und das Programm stürzt ab):

