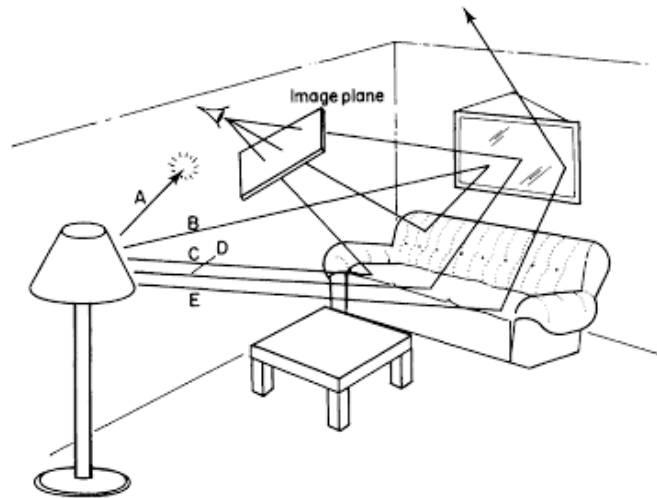


## Projekt 08: Raytracing mit Würfel

Raytracing bezeichnet eine alternative Render-Methode, um virtuelle Szenen auf dem Computer darzustellen. Die Idee basiert auf der Verfolgung von Lichtstrahlen von der Lichtquelle in das Auge des Betrachters. Wenn man jedoch physikalisch korrekt wäre, müsste eine unendliche Anzahl an Strahlen von der Lichtquelle in die Szene verfolgt werden, nur damit wenigstens ein paar davon die Kamera treffen. Der Ansatz von Raytracing ist deshalb, Strahlen von der Kamera aus in die Szene zu senden, Schnittpunkte mit den Objekten zu berechnen und aus diesen Schnittpunkten dann die Farbe zu bilden. Dabei werden die Strahlen von einem "Augpunkt" durch jeden Pixel einer Bildebene in die Szene geschickt. Es ist möglich eine unterschiedliche Menge an Strahlen pro Pixel zu senden. Außerdem kann von jedem Schnittpunkt aus noch ein sogenannter "Schattenfühler" zu jeder Lichtquelle gesendet werden, um zu prüfen, ob die Stelle im Schatten oder Licht liegt. Die normale GPU Pipeline basiert auf dem Verfahren des Rasterizers und kann keine Strahlen in die Szene verfolgen. Deshalb sind Raytracing-Algorithmen größtenteils auf der CPU zu finden und sehr performancelastig. Da die GPU jedoch sehr effizient im Verarbeiten von Vektoren ( Strahlen ) und Matrizen ist, kann mithilfe von OpenCL oder CUDA eine eigene Implementierung erfolgen. Für mehr Infos über Raytracing siehe [2] und [1].



## Ziel

Das Ziel dieses Projekts ist die Implementierung eines Raytracers mit einfachen Objekten in der Szene. Als Objekte sollen Würfel verwendet werden, da diese sich leicht durch mathematisch implizite Funktionen darstellen lassen und deshalb sehr günstig in der Schnittpunktberechnung sind. Da Raytracing auch bedeutet für jeden Schnittpunkt weitere Strahlen auszusenden, können verschiedene Materialien implementiert werden, z.B. Glas, Metall, Plastik oder ähnliches. Die Menge an Effekten kann dabei auf Brechungen, Spiegelungen oder undurchsichtiges Material beschränkt werden. Als Lichtquelle soll eine diffuse Lichtquelle verwendet werden. Diese besteht rein technisch aus einer Ansammlung von punktuellen Lichtquellen und sorgt beim Rendern für weiche Schattenübergänge. Damit die Strahlen jedoch nicht immer weitere Strahlen generieren, ist eine geeignete Endbedingung wichtig, z.B. maximale Baumtiefe von 3 Ebenen. Für OpenGL kann anschließend die entsprechend berechnete Farbe als Pixel gerendert werden. Der Einfachheit halber bitte nur mit Farben arbeiten.

## Geschätzter Aufwand

- CUDA / OpenCL: 80 %
- OpenGL: 20 %

## Themen

- Raytracer in 3D
- Würfel als implizite Objekte
- Eine simple Ebene als Untergrund für alle Würfel
- Eine diffuse Lichtquelle
- Neben undurchsichtigen Objekten, entweder ein Spiegeleffekt oder Brechungseffekt.

## Evaluierung

- Unterschiedliche Menge an Würfeln
- Position der Lichtquelle verändern
- Unterschiedliche Auflösung der Bildebene.

## References

- [1] Jag Mohan Singh and PJ Narayanan. Real-time ray tracing of implicit surfaces on the gpu. *IEEE transactions on visualization and computer graphics*, 16(2):261–272, 2010.
- [2] Martin Zlatuška and Vlastimil Havran. Ray tracing on a gpu with cuda—comparative study of three algorithms. 2010.