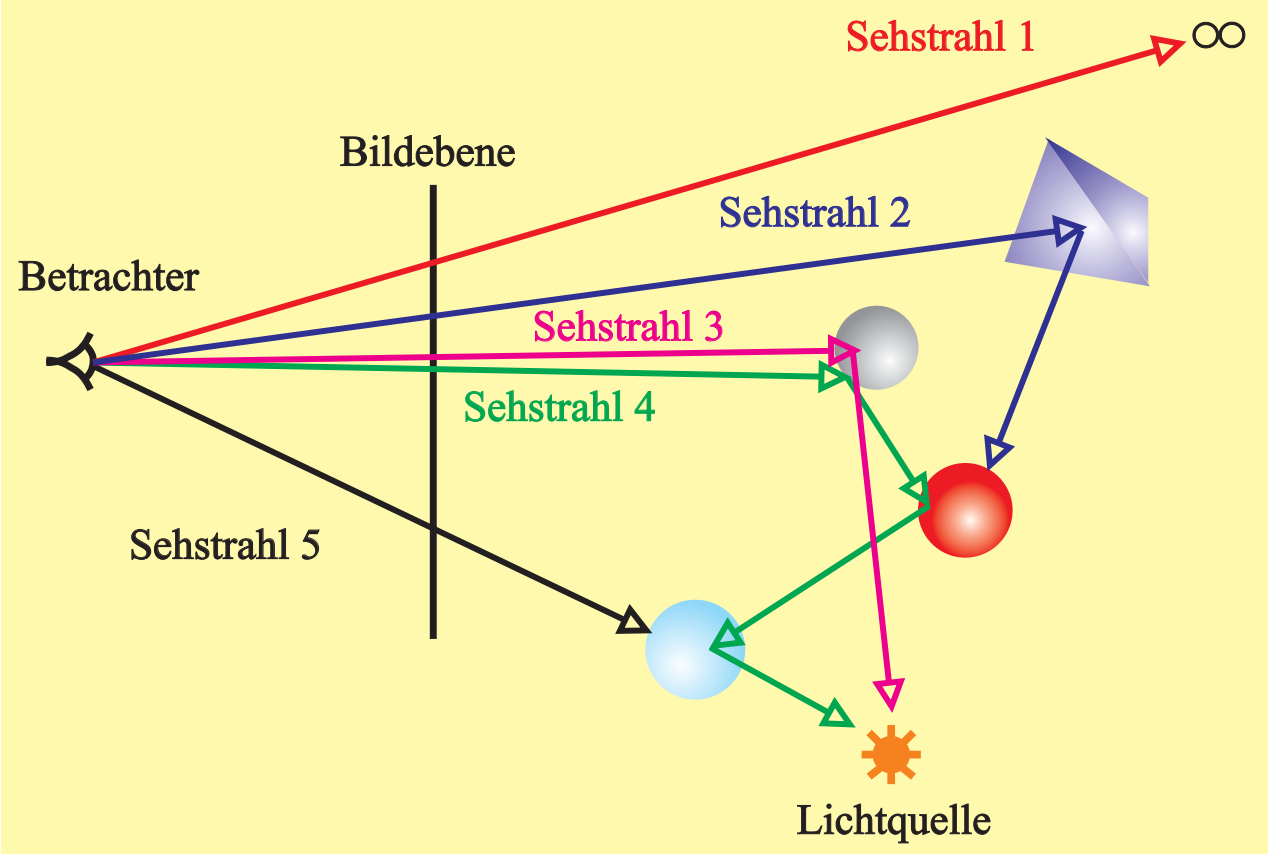
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing

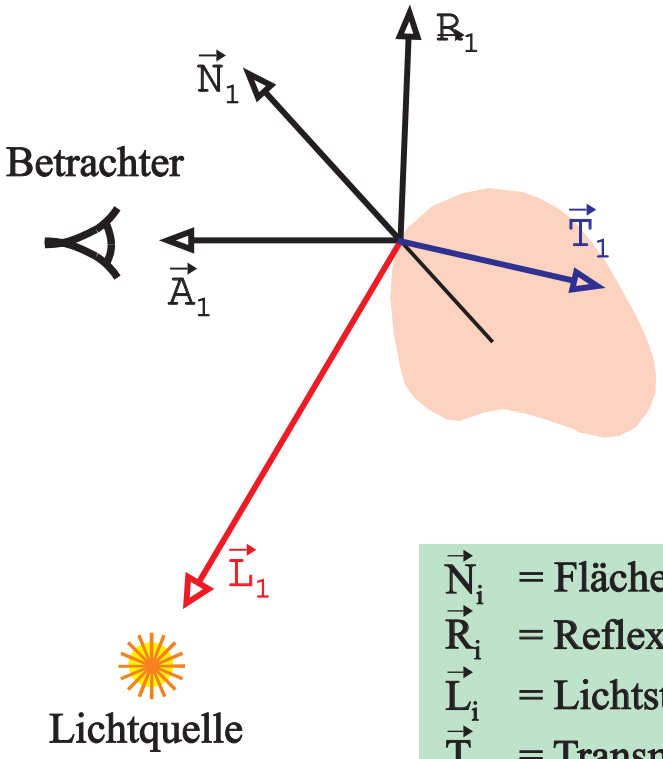
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus, Strahlverläufe (1)

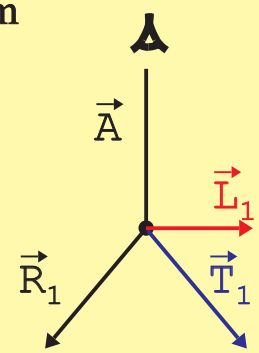


Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus, Strahlverläufe (2)



Strahlbaum



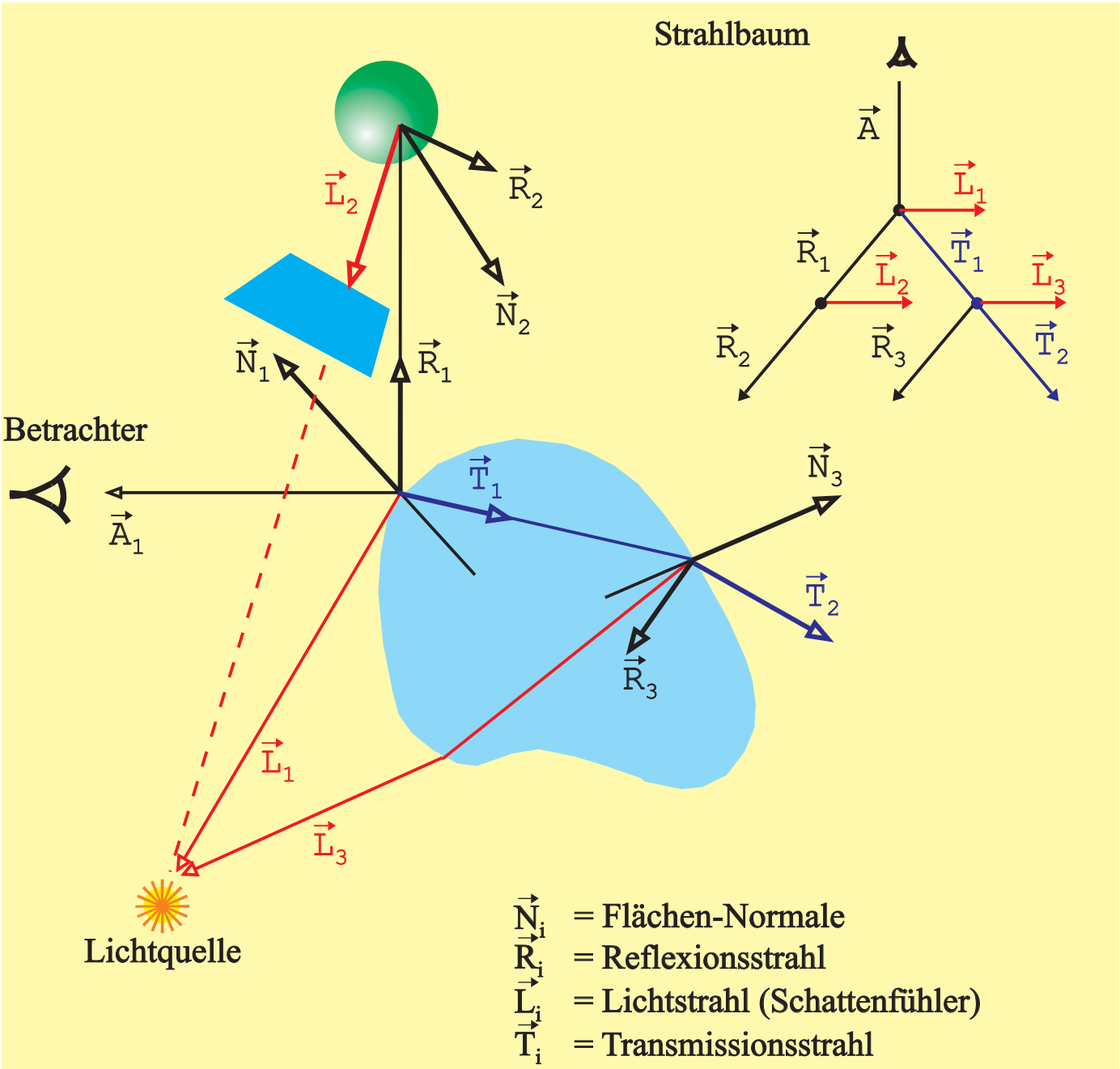
$\vec{N}_i$ = Flächen-Normale
$\vec{R}_i$ = Reflexionsstrahl
$\vec{L}_i$ = Lichtstrahl (Schattenfühler)
$\vec{T}_i$ = Transmissionsstrahl

Prinzip

1. Vom Betrachter aus Sichtstrahl (Primärstrahl) starten, nächsten Schnittpunkt P berechnen.
2. Mit Lichtstrahl (Schattenfühler) zur Lichtquelle bestimmen, ob P im Schatten liegt. Wenn nicht, lokales Beleuchtungsmodell anwenden, um Farbe von P zu bestimmen.
3. Wenn spekulare Reflexion und/oder Transmission vorhanden, von P aus Reflexions- und/oder Transmissionsstrahl (Sekundärstrahlen) starten und jeden Sekundärstrahl wieder als Primärstrahl betrachten (Strahlbaum, Rekursion).
4. Abbruchkriterien: Tiefe des Strahlbaums oder zu geringer Anteil des betrachteten Punktes.

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus, Strahlverläufe (3)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus (Prinzip)

Hauptprogramm und Prozedur Verfolgen

```
PROGRAM Ray_Tracing;
  :
  :
FOR alle Pixel DO
BEGIN
  Sichtstrahl berechnen;
  Baumhöhe:=0;
  Gewicht:=1;

  Verfolgen(Baumhöhe, Gewicht, Sichtstrahl, Farbe);

  Farbe in den Bildspeicher schreiben;
END;
  :
  :
END {Ray_Tracing}.
```

```
PROCEDURE Verfolgen(Baumhöhe, Gewicht, Sichtstrahl, Farbe);

BEGIN
IF (Baumhöhe>=Max_Baumhöhe THEN
  Farbe:=0;
ELSE
  IF Schnitt mit Objekt THEN
    BEGIN
      Berechnung Schnittpunkt und Normale;

      Beleuchtung(Baumhöhe, Gewicht, Sichtstrahl, Farbe);

    END
  ELSE Farbe:=Hintergrundfarbe;
END {Verfolgen};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus (Prinzip)

Prozedur Beleuchtung (1)

```
PROCEDURE Beleuchtung(Baumhöhe, Gewicht, Sichtstrahl, Farbe);

BEGIN

  Farbe:=Anteil ambientes Licht;

  FOR alle Lichtquellen DO
  BEGIN
    Berechnung Schattenstrahl;
    IF Schnittpunkt nicht im Schatten THEN
    BEGIN
      Farbe:=Farbe+Anteil diffuse Reflexion;
      IF Spekularexponent m>0 THEN
      BEGIN
        Berechnung Spekularstrahl;
        Farbe:=Farbe+Anteil spekulare Reflexion;
      END {Spekulare Reflexion};
    END {Diffuse Reflexion};
  END {Lichtquellen};
   :
   :
```

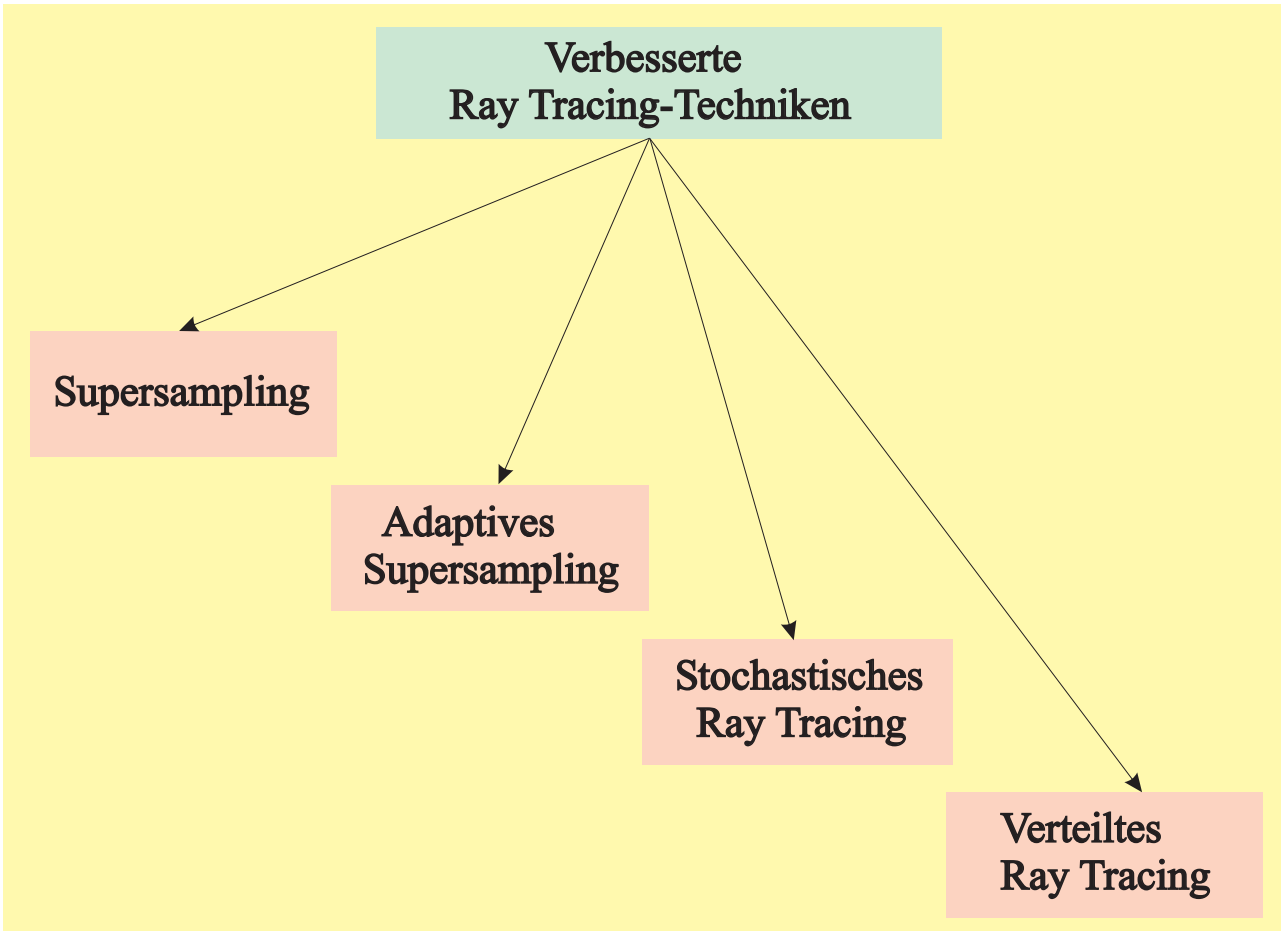
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Algorithmus (Prinzip)

Prozedur Beleuchtung (2)

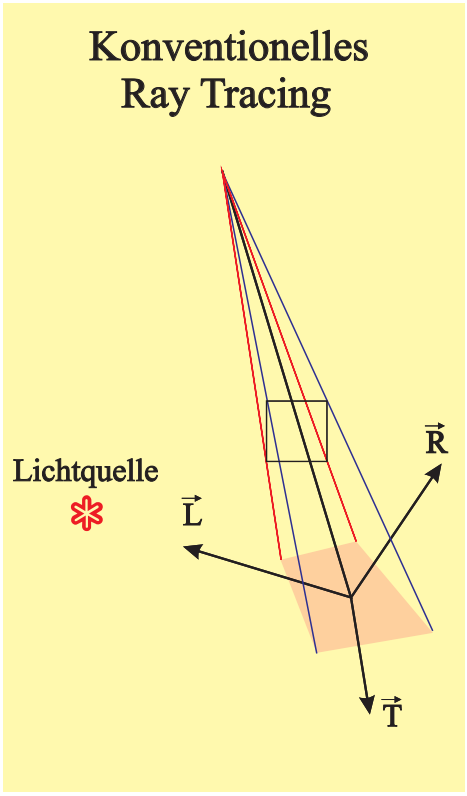
```
  :
  :
IF(k_s*Gewicht>Min_Gewicht) THEN
BEGIN
  Berechnung Spekularstrahl;

  Verfolgen(Baumhöhe+1, k_s*Gewicht,
            Spekularstrahl, Spekularfarbe);

  Farbe:=Farbe+k_s*Spekularfarbe;
END {Globaler Spekularanteil};

IF(k_t*Gewicht>Min_Gewicht) THEN
BEGIN
  Berechnung Transmissionsstrahl;

  Verfolgen(Baumhöhe+1, k_t*Gewicht,
            Transmissionsstrahl, Transmissionsfarbe);

  Farbe:=Farbe+k_t*Transmissionsfarbe;
END {Globaler Transmissionsanteil};

END {Beleuchtung};
```

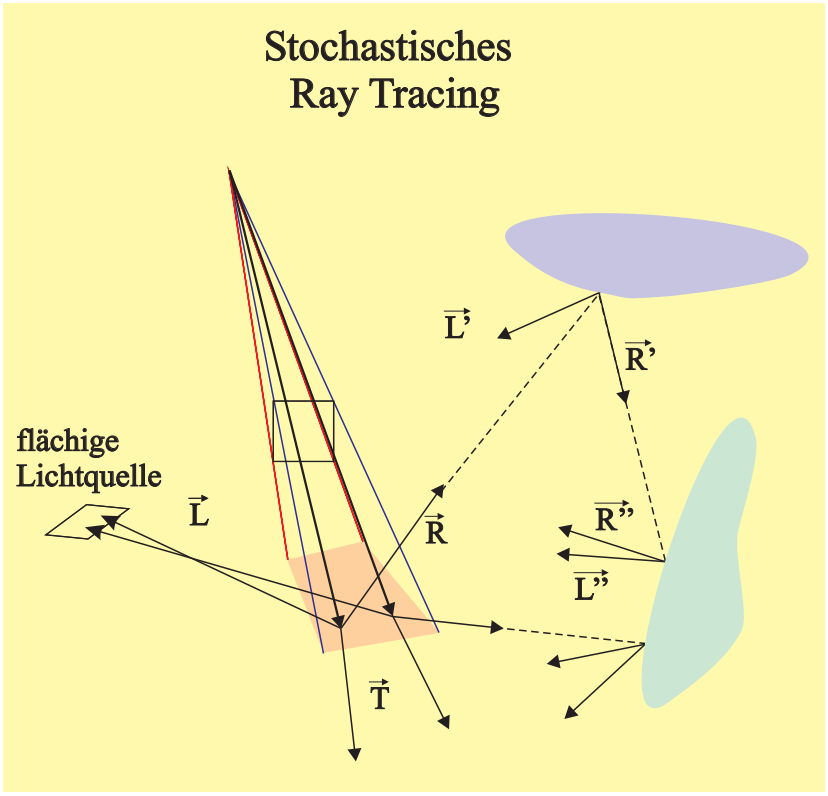
Ray Tracing-Techniken



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Stochastisches Ray Tracing

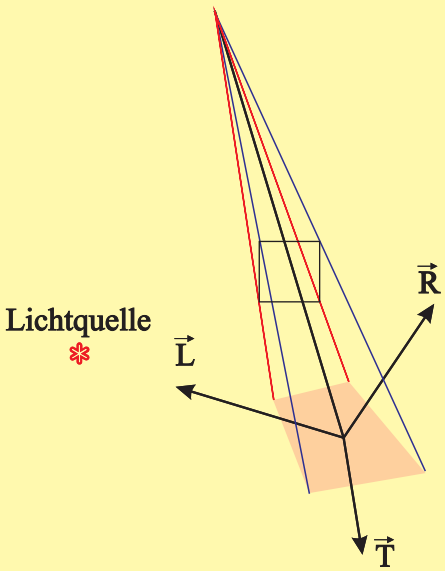




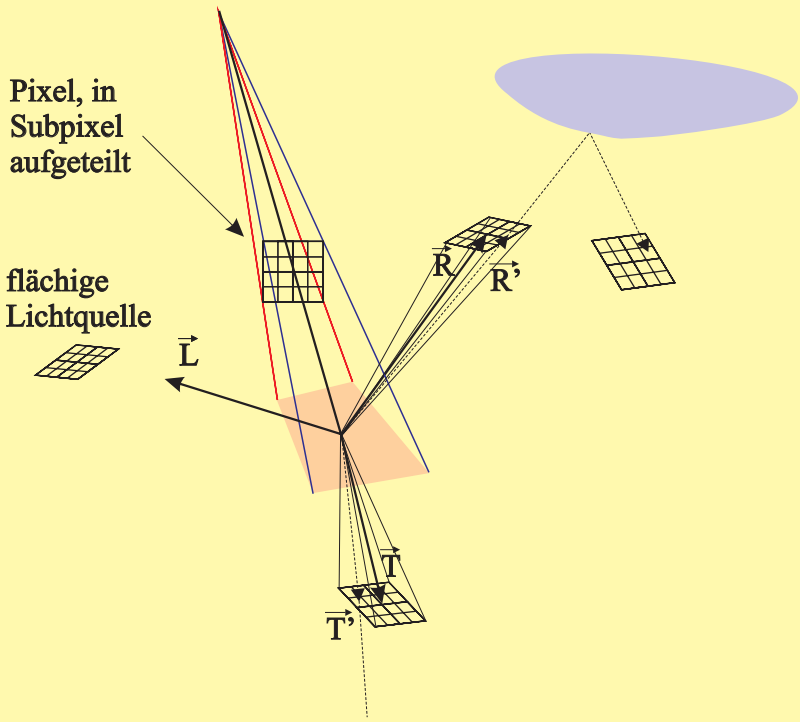
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Verteiltes Ray Tracing (distributed ray tracing)

Konventionelles Ray Tracing



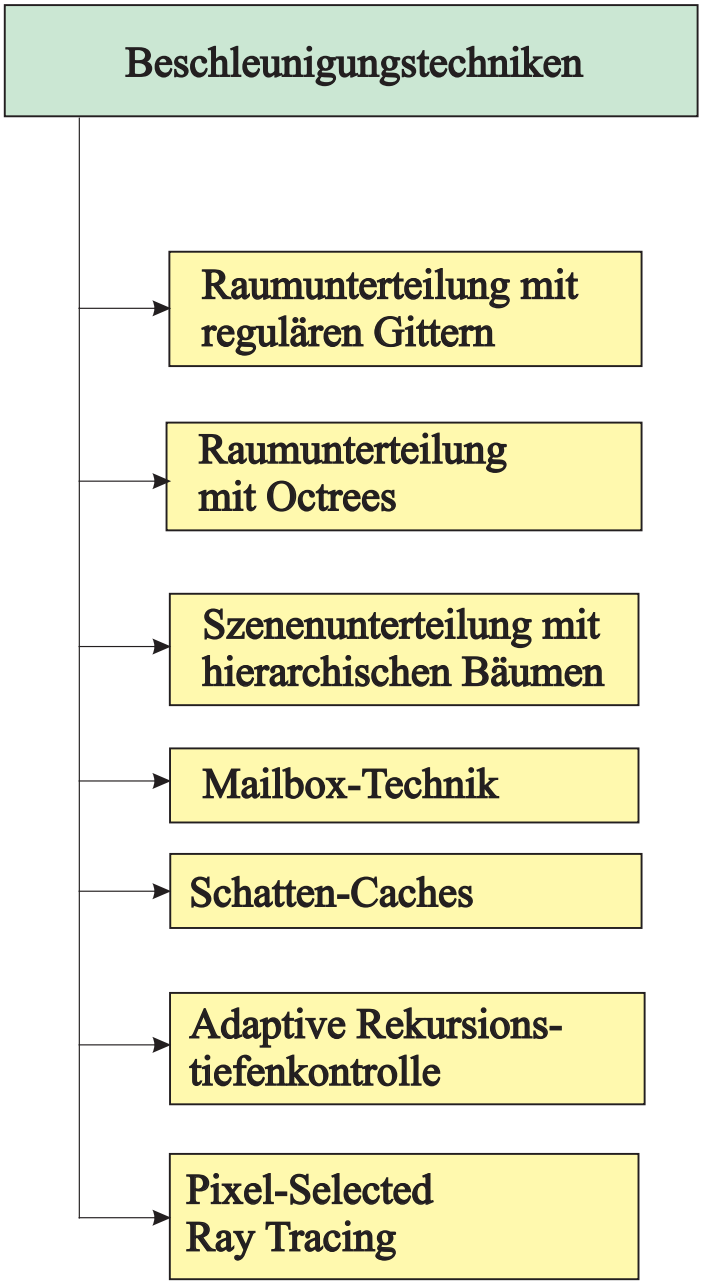
Verteiltes Ray Tracing



Effekte: 1. Weicher Schattenübergang (Penumbra).
2. Simulation von rauhen Oberflächen.
3. Erhöhter Berechnungsaufwand.

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing-Beschleunigungstechniken



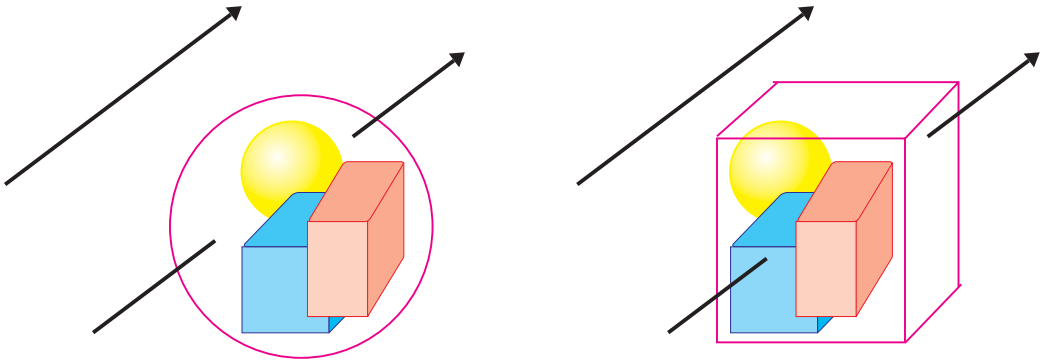
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing

Umgebende Volumen

Ziel: Reduzierung der Schnittpunktberechnungen.

Methode: Einschluß mehrerer Objekte in ein umgebendes Volumen (Bounded Volume), da in einer Szene nur wenige Objekte vom gerade untersuchten Strahl getroffen werden. Üblich sind Kugeln und Quader (Boxen).



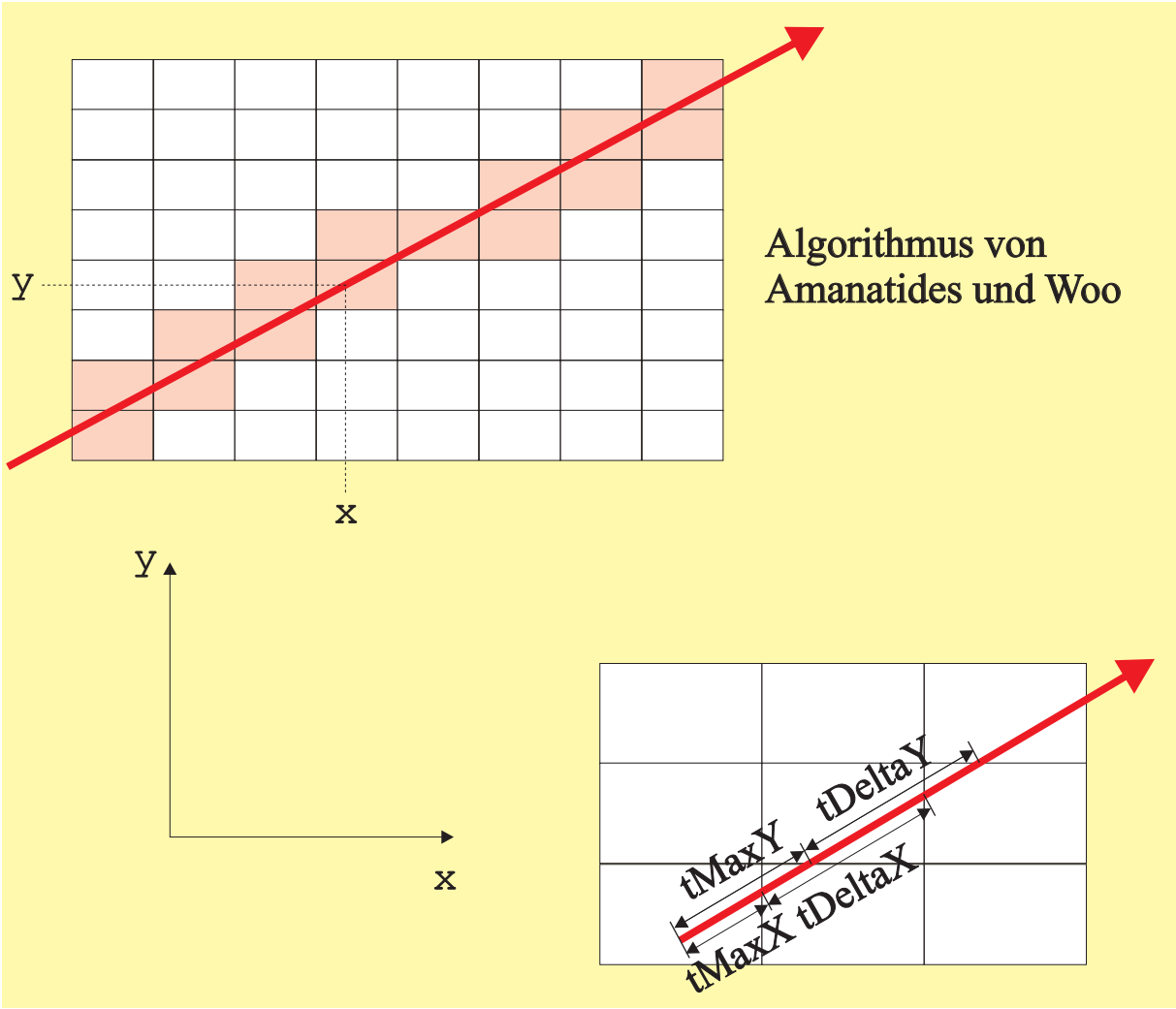
Ergebnis: Wird das umgebende Volumen vom Strahl nicht geschnitten, so gilt das auch für die umschlossenen Objekte.

Hinweis: Für das umgebende Volumen ist nur der Schnittpunkt-Test erforderlich, nicht die genaue Schnittpunktberechnung!

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing

Beschleunigungstechniken
Raumunterteilung mit regulären Gittern



Ray Tracing

Beschleunigungstechniken
Traversierungsalgorithmus für den zweidimensionalen Fall

```
repeat
  if tMaxX<tMaxY then
  begin
    tMaxX:=tMaxX+tDeltaX;
    X:=X+StepX
  end else
  begin
    tMaxY:=tMaxY+tDeltaY;
    Y:=Y+StepY
  end;
  NextPixel(X,Y)
until  (alle Pixel bearbeitet);
```

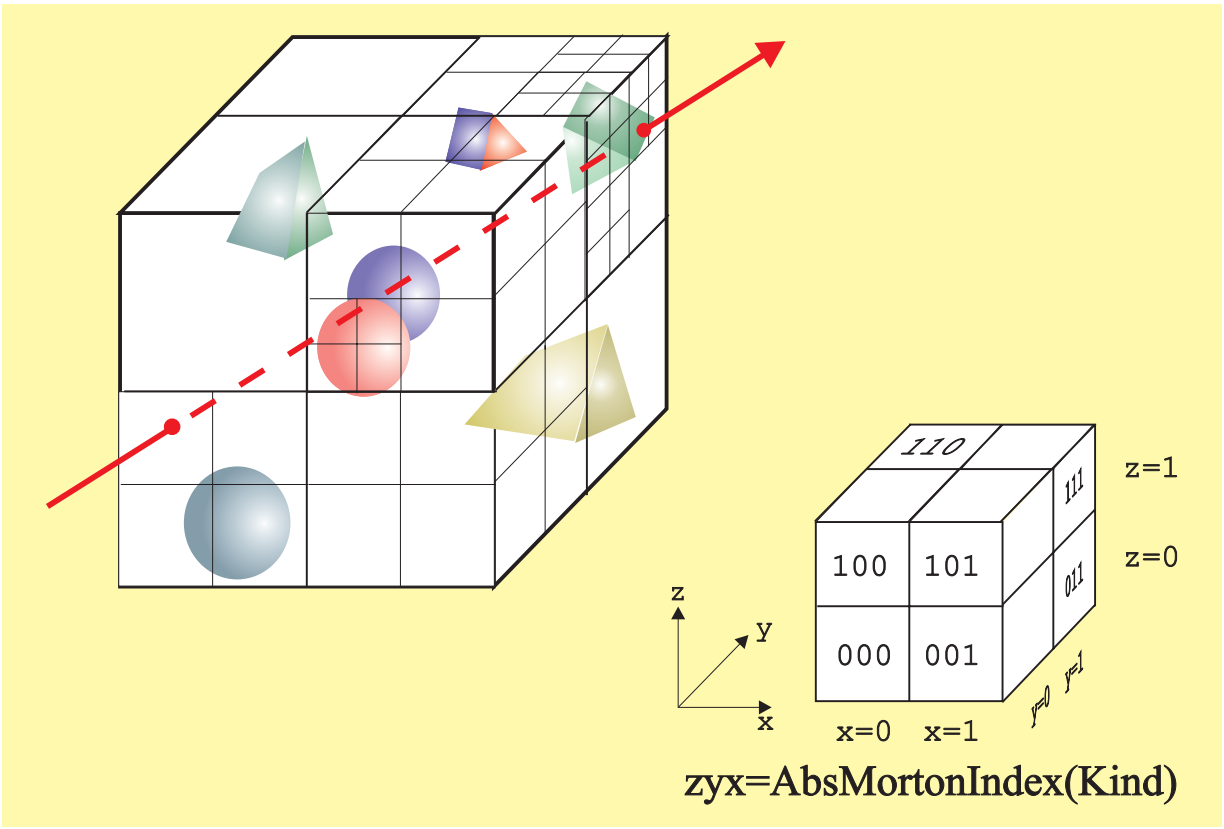
Legende:

X,Y	= Koordinaten des aktuellen Pixels
StepX,StepY	= Schrittweite, +1 oder 1.
tMaxX,tMaxY	= Strahlabstand bis zu nächsten vertikalen bzw. horizontalen Voxelbegrenzung.
tDeltaX,tDeltaY	= Werte die angeben, wieweit man den Strahl von einer horizontalen/vertikalen Voxel-begrenzung zur nächsten verfolgen muß.

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing

Beschleunigungstechniken
Raumunterteilung mit Octrees, SMART-Algorithmus (1)



Ray Tracing

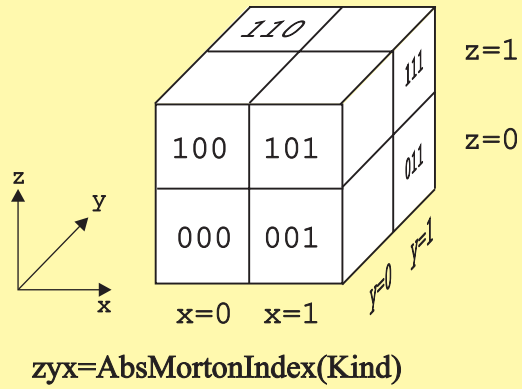
Beschleunigungstechniken
Raumunterteilung mit Octrees, SMART-Algorithmus (2)

```
procedure Traverse(MortonIndex,V_SMART,H_SMART, ...);
begin
  if Knoten(MortonIndex) ist ein Blattvoxel
  then
    Führe Schnittpunkttest des Strahls mit allen Objekten im Blattvoxel
durch
  else
  begin
    // Der aktuelle Knoten ist ein interner Voxelknoten und hat 8 Sub-Knoten
    //(Sub-Voxel)
    Update MortonIndex, V_SMART,  ... für die vertikale Traversierung.
    Traverse(MortonIndex,V_SMART,H_SMART, ...);
    // Elternvoxel  Kindvoxel
    while RelMortonIndex<>7 do
    begin
      Update MortonIndex, H_SMART,  ... für die horizontale Traversierung.
      Traverse(MortonIndex,V_SMART,H_SMART, ...);
       // Kindvoxel  benachbartes Kindvoxel
    end;
  end;
end {Traverse};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

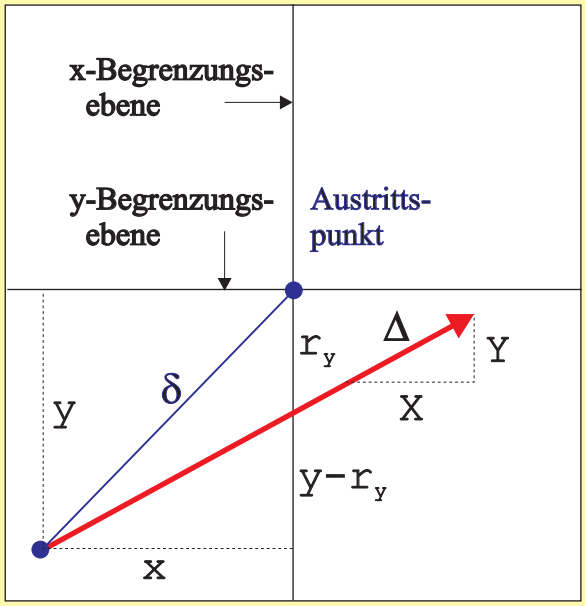
Ray Tracing

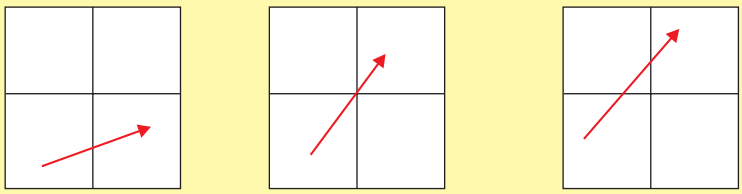
Beschleunigungstechniken
Raumunterteilung mit Octrees, SMART-Algorithmus (3)



zyx=AbsMortonIndex(Kind)

Horizontale Traversierung



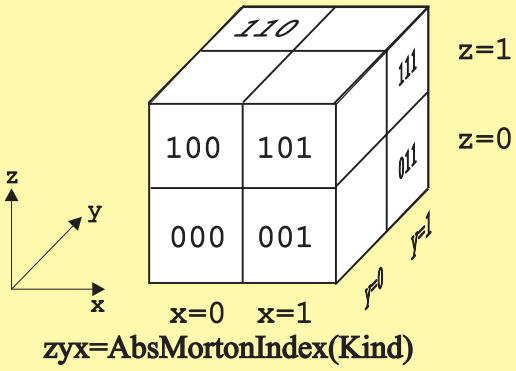


$r_y>0$ $r_y=0$ $r_y<0$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

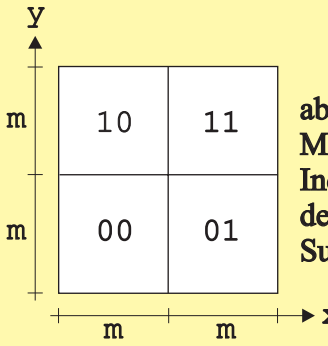
Ray Tracing

Beschleunigungstechniken
Raumunterteilung mit Octrees, SMART-Algorithmus (4)



zyx=AbsMortonIndex(Kind)

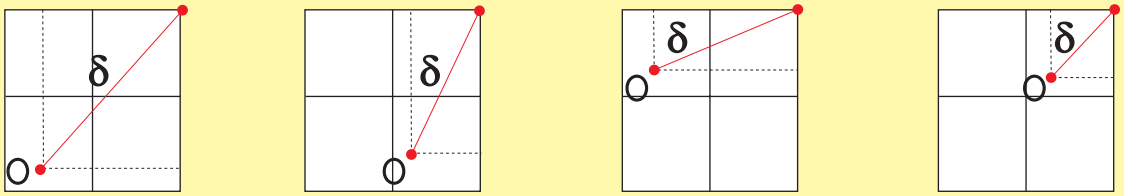
Vertikale Traversierung



absolute Morton-Indizes der Subpixel

$\delta=[x,y],\ x>0,\ y>0$

Austrittspunkt (11)



x>m,y>m
11 XOR 11=00

x<=m,y>m
10 XOR 11=01

x>m,y<=m
11 XOR 11=00

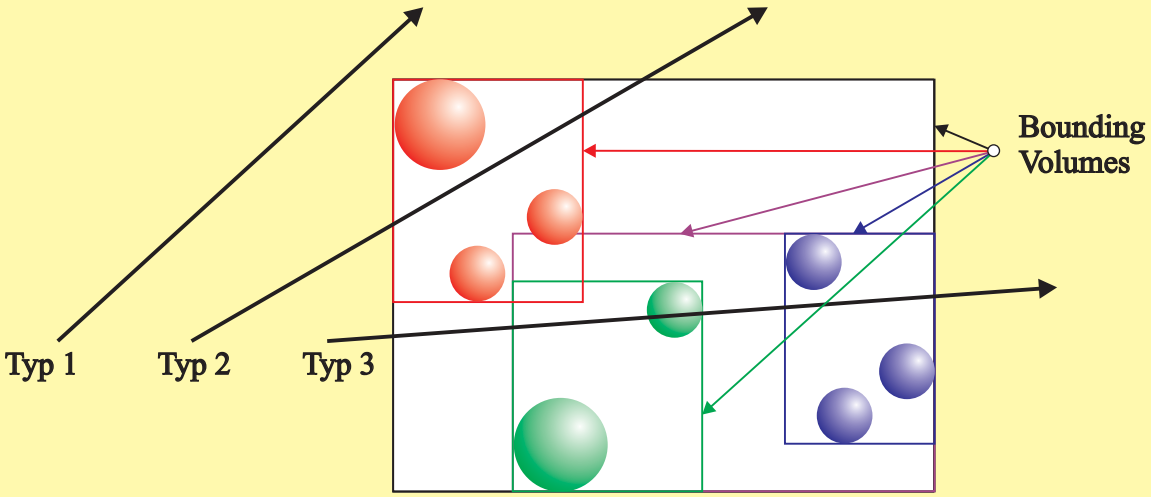
x<=m,y<=m
11 XOR 11=00

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

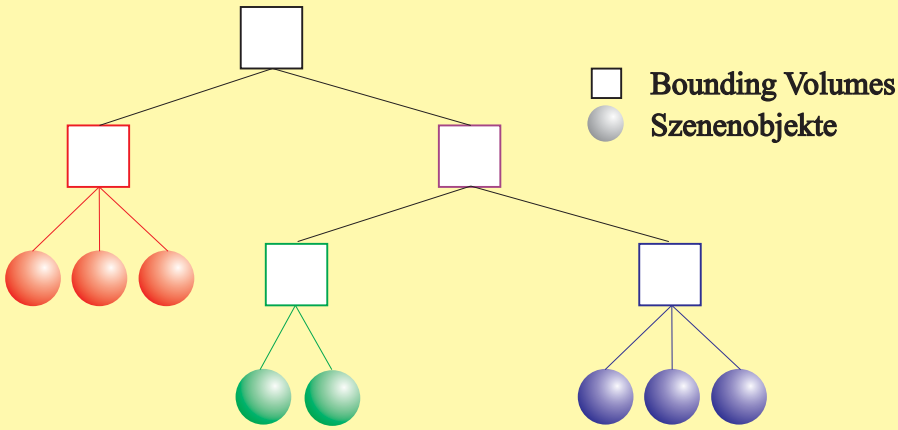
Ray Tracing

Beschleunigungstechniken
Szenenunterteilung mit hierarchischen Bäumen

Mögliche Strahltypen beim Schnitt mit dem Bounding Volume



Bounding Volume Hierarchie



Ray Tracing

Beschleunigungstechniken
Effektiver Traversierungsalgorithmus durch Binärbäume (Prinzip)

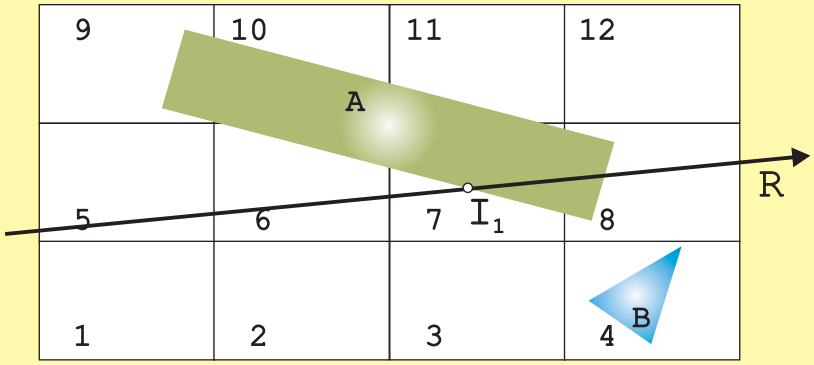
```
procedure Traverse(Strahl, Root);
begin
  Schneide Strahl mit Root und füge Root in Heap ein,
  wenn Schnittpunkt vorhanden;
  mindist:=INFINTY;
  while (Heap nicht leer) do
  begin
    Hole obersten Knoten N vom Heap;
    if (N.dist > mindist) then return;
    if (N ist Blatt des Hierarchiebaums) then
    begin
      Berechne Schnitt von Strahl mit allen Objekten in N;
      mindist:=Strahllänge bis zum nächsten bisher gefundenen
Schnittpunkt;
    end else
    begin
      Für jedes Kind K von N führe aus
      begin
        K.dist:=Strahllänge bis zum Schnittpunkt;
        if (K.dist < mindist) then Füge K in Heap ein
      end;
    end;
  end;
end {Traverse};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

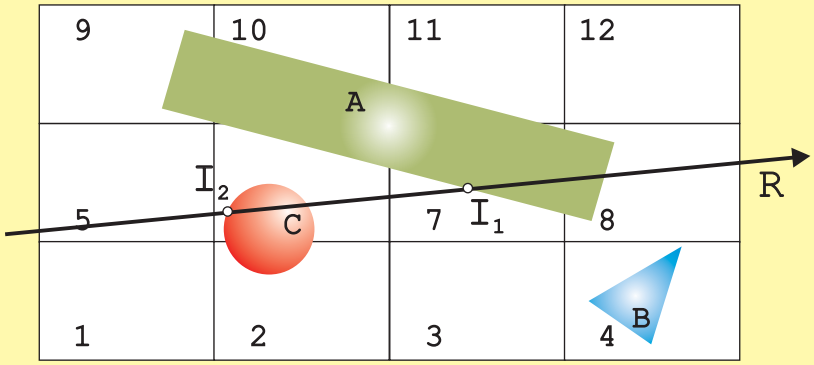
Ray Tracing

Beschleunigungstechniken
Mailbox-Technik

Fall 1



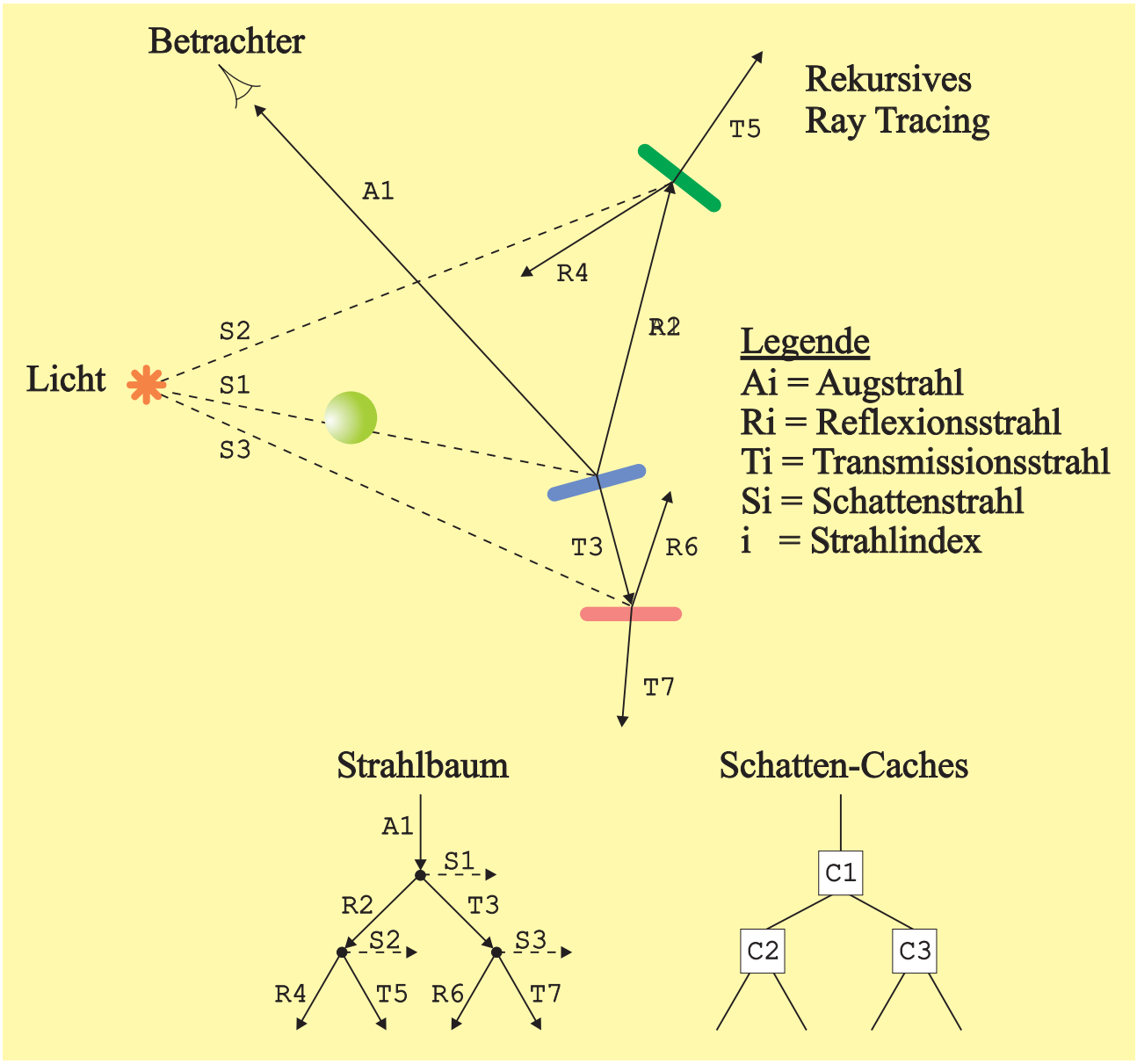
Fall 2



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

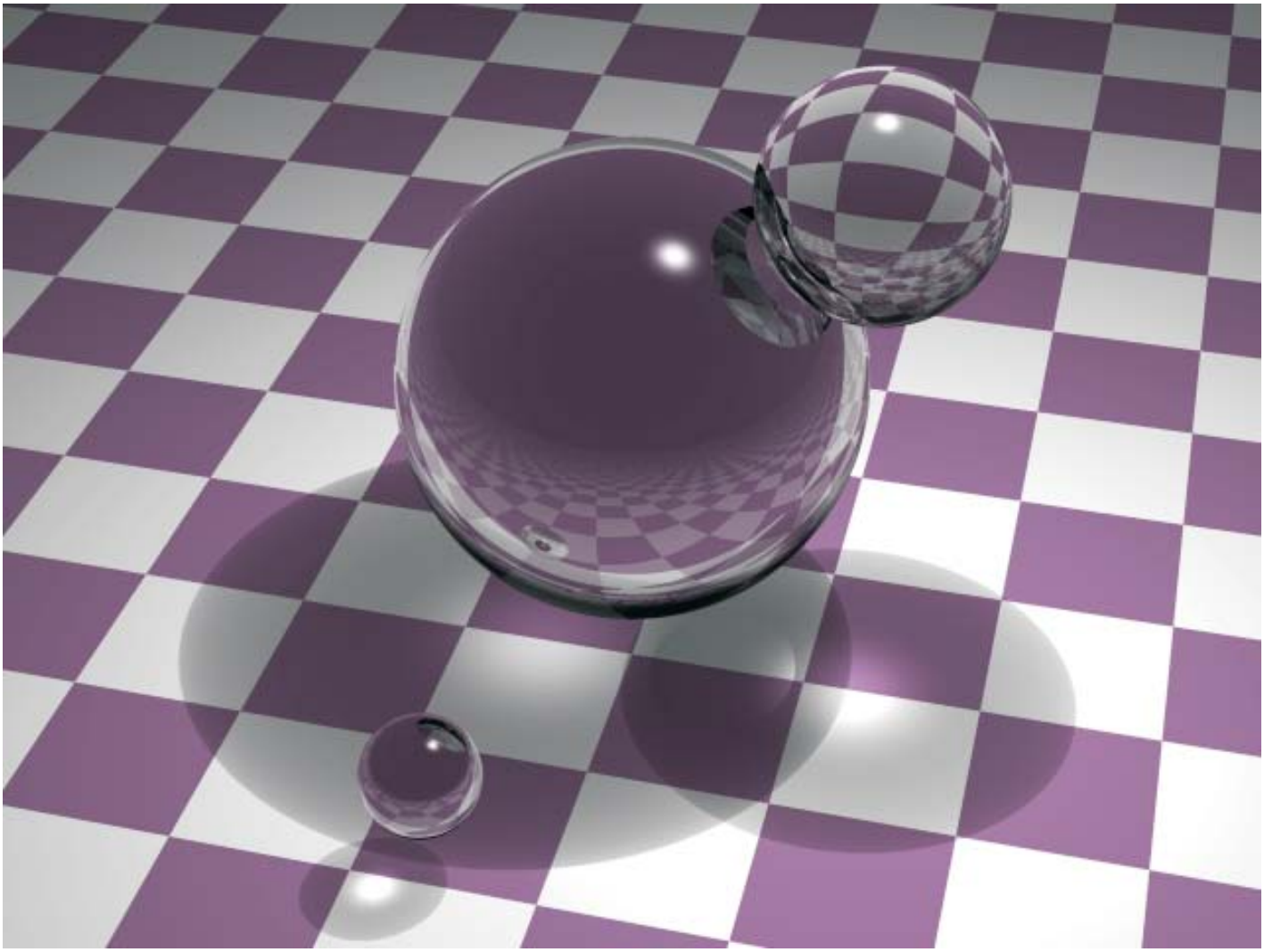
Ray Tracing

Beschleunigungstechniken
Schatten-Caches



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Ray Tracing