

# Interaktive Computergrafik



**Prof. Dr. Frank Steinicke**  
Human-Computer Interaction  
Department of Computer Science  
University of Hamburg



# Interaktive Computergrafik

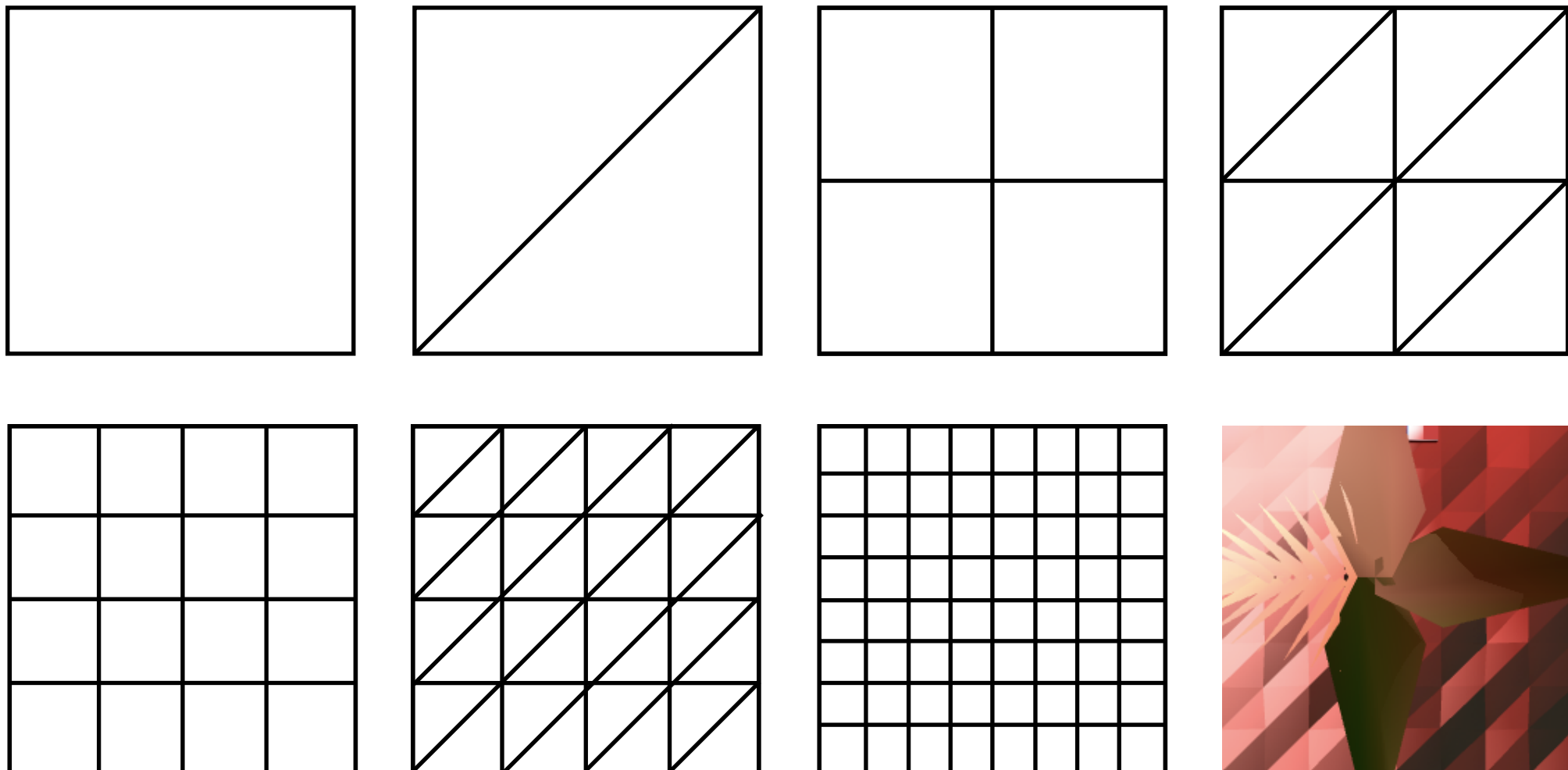
## Übung - Woche 13

# Idee

- (1) Tesselierte Fläche mit  $X$  Dreiecken
- (2) Wellenfunktion anhand  $\sin()$  berechnen
- (3) Normale dynamisch berechnen durch Kreuzprodukt zweier approximierter Punkte

# (1) Tessellation

- rekursive Aufteilung eines Meshes anhand einer mathematischen Funktion



# (1) Tessellation

- In OpenGL: Tessellation Shader
- In WebGL: eigenhändige Unterteilung der Geometrie in Anwendungsprogramm

```
for(let i=-10; i<10; i+=0.5) {  
    for(let j=-10; j<10; j+=0.5) {  
        this.indices.push(...);  
        this.positions.push(...);  
    }  
}
```

# (2) Displacement

## Diamond-square algorithm

- ▶ zur Erzeugung von Höhenfeldern (z.B. Terrain)
- ▶ Tessellation + zufällige Verschiebung neu erzeugter Vertices entlang der y-Achse

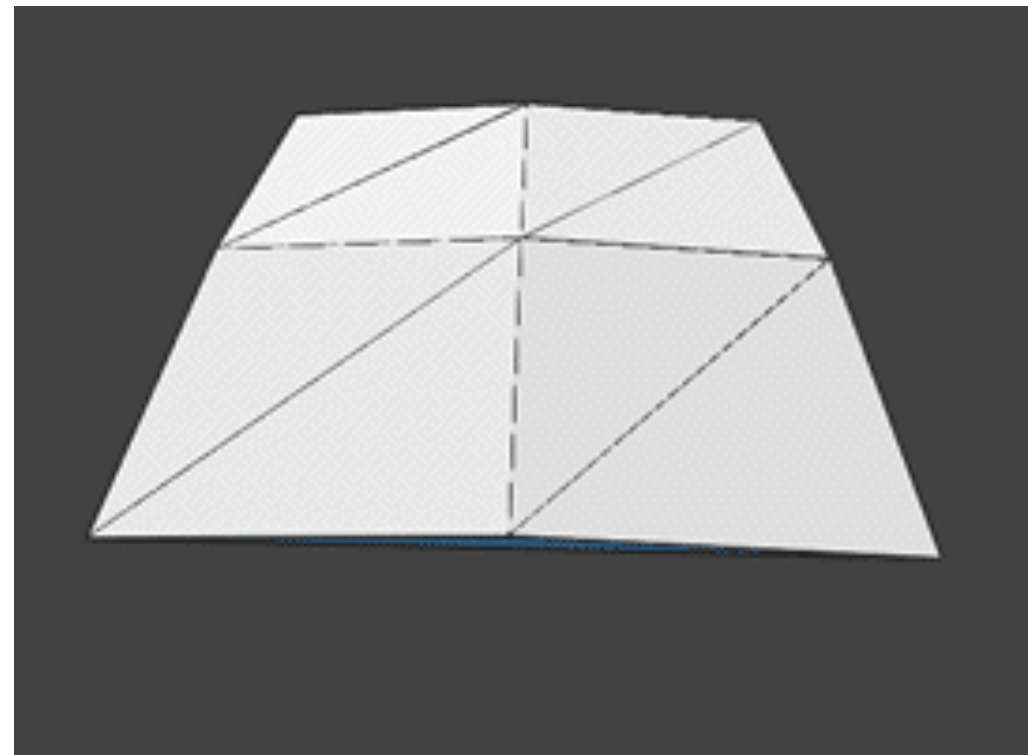
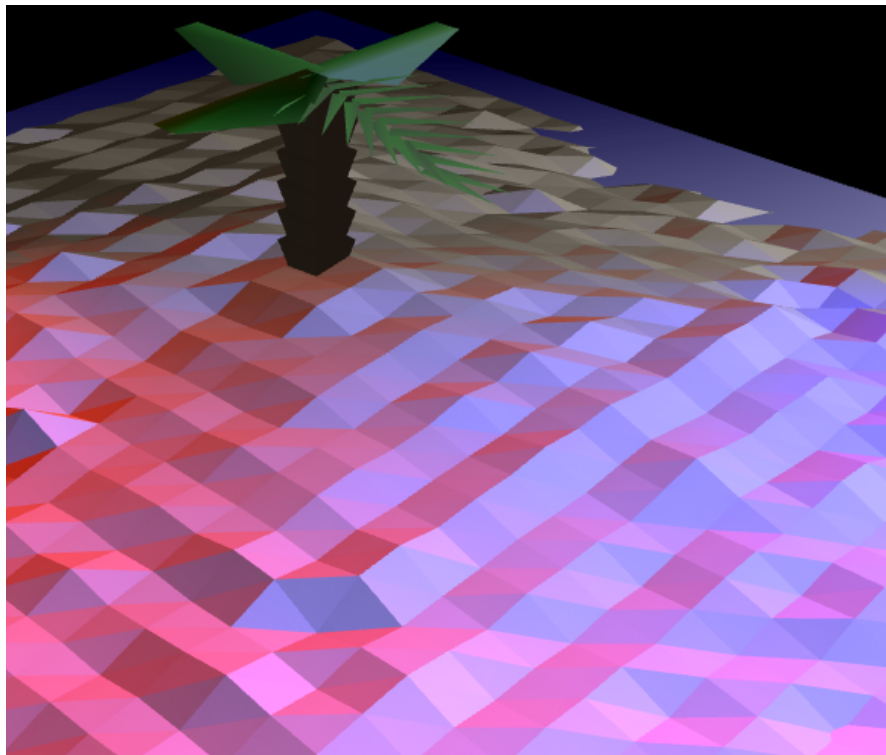
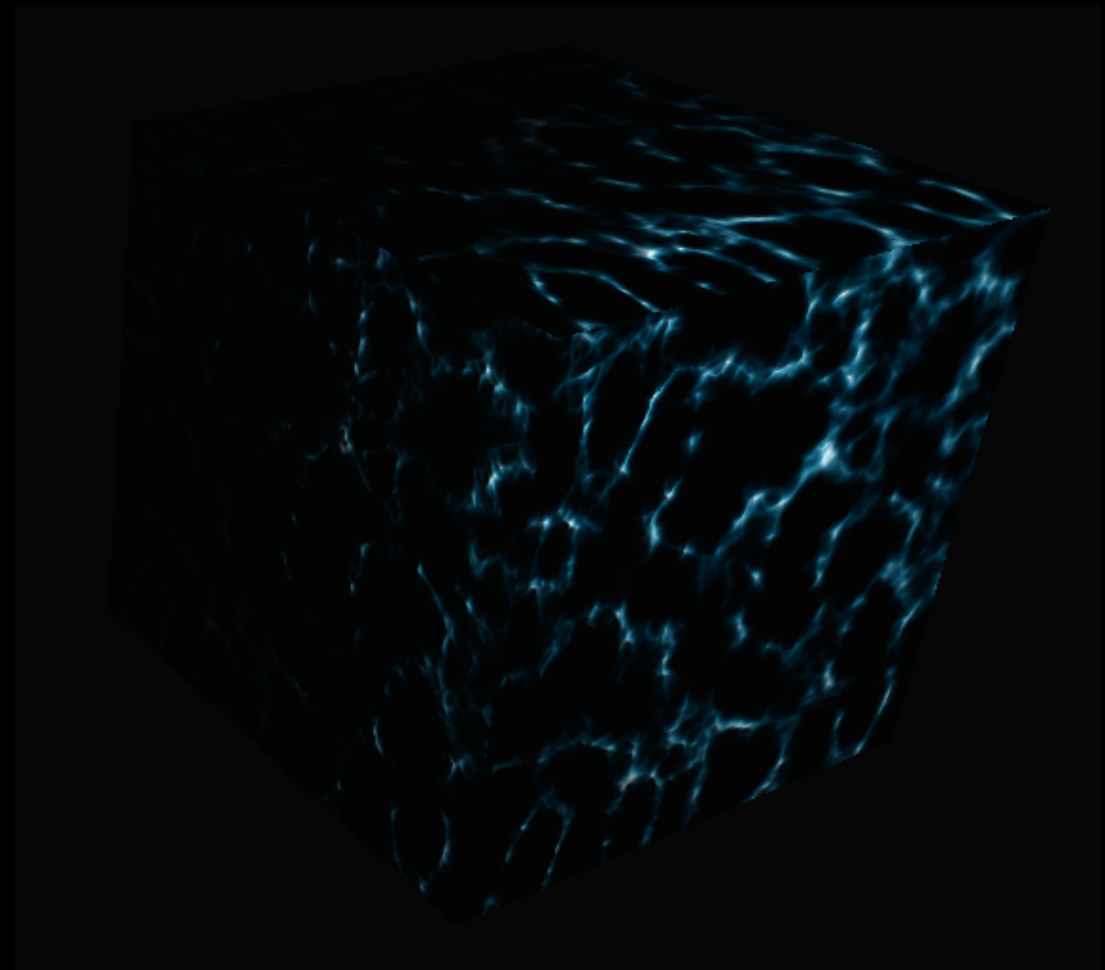
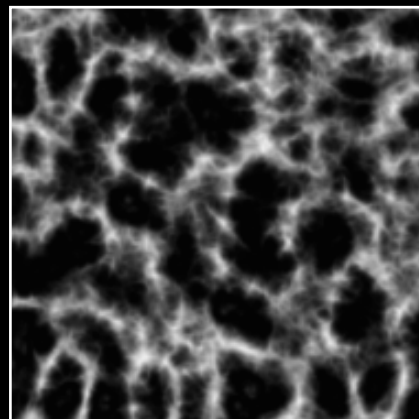
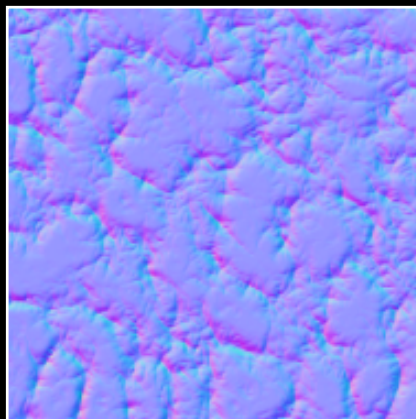
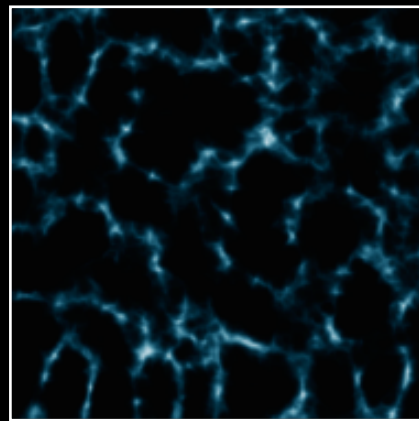
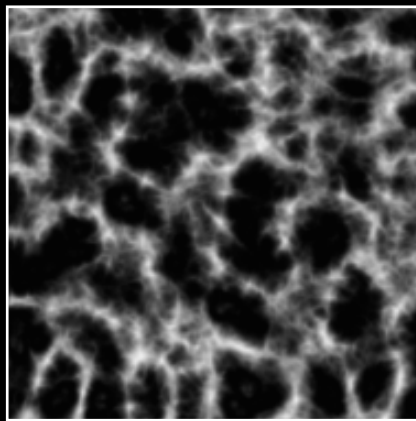


Bild: ICG Gruppe 1.1, WS 17/18

# (2) Displacement Simplex Noise



# (2) Displacement

## Wellenfunktion

- ▶ Verschiebung aller Vertices entlang der y-Achse anhand ihrer xz-Position und einer definierten Wellenfunktion
- ▶ Allgemeine Sinusfunktion (nur für x):

$$f(x) = y = a * \sin(bx + c) + d$$

*a = Streckung/Stauchung in y-Richtung*

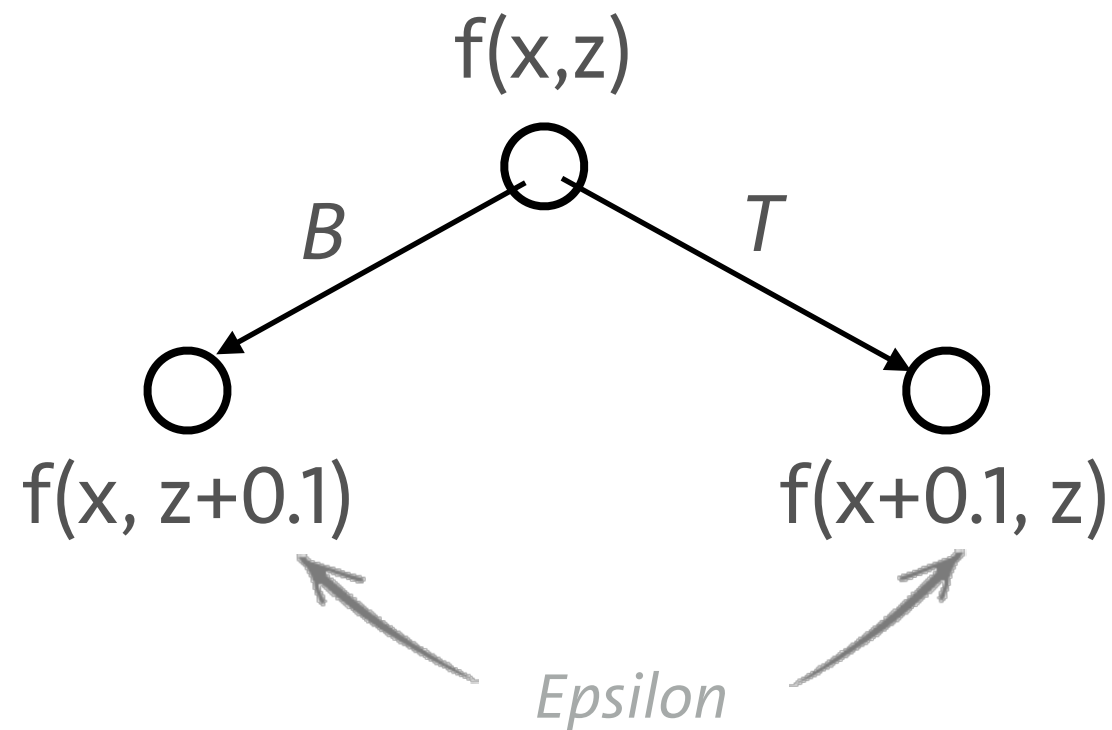
*b = Streckung/Stauchung in x-Richtung*

*c = Verschiebung entlang der x-Achse*

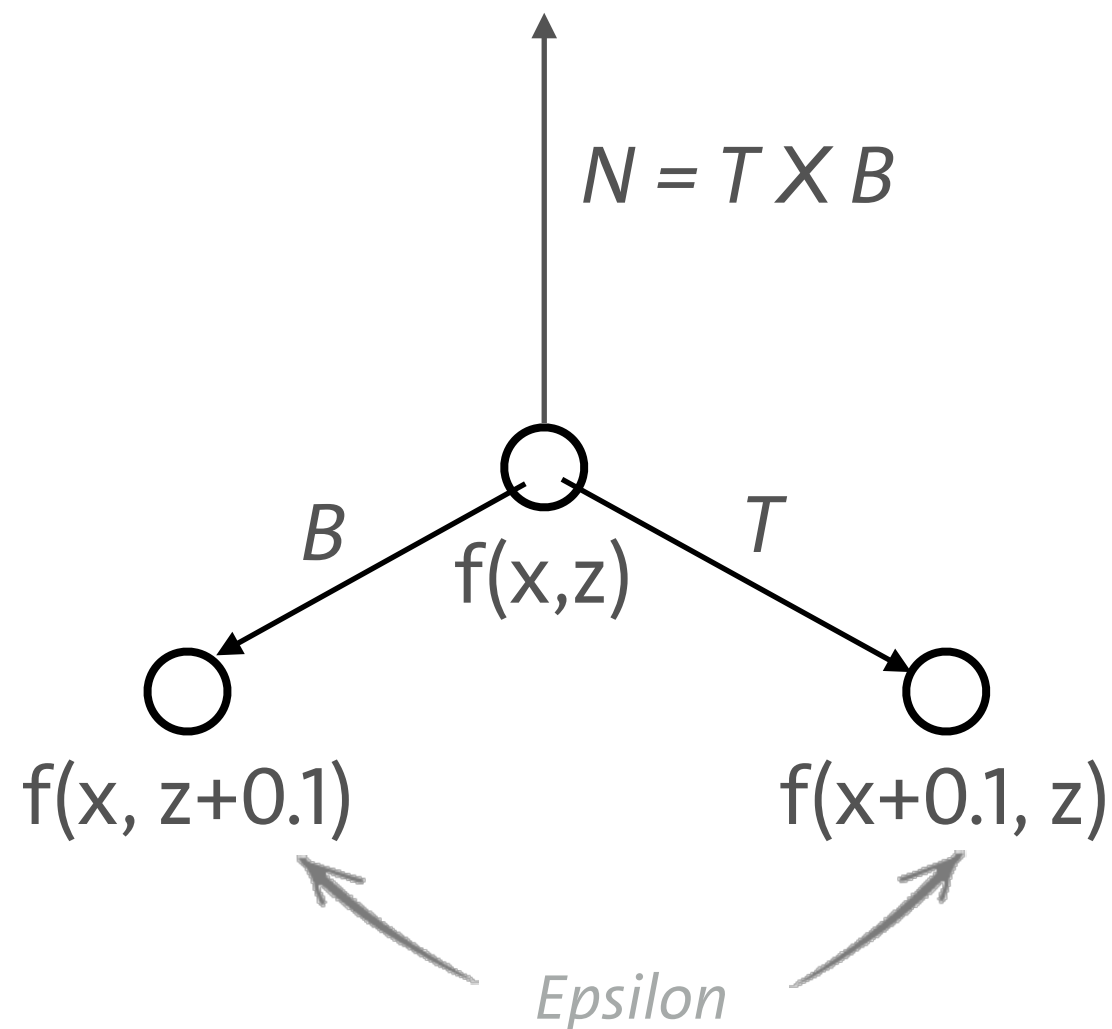
*d = Verschiebung entlang der y-Achse*



# (3) Normalenberechnung



# (3) Normalenberechnung



*Hinweis: Da alte vordefinierten Normalen nicht mehr benötigt werden → entsprechende Variable aus Shader entfernen (sonst Fehlermeldung)*

# Ergebnis

