Moderovacie a renderovacie techniky

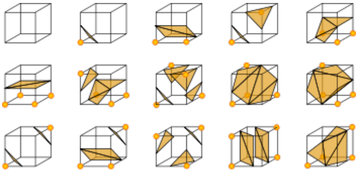
František Dráček
dracek1@uniba.sk

5. októbra 2023

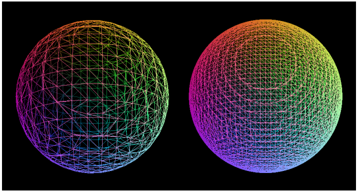
https://github.com/frantisekdracek/Prezentacie/tree/main

Marching cubes

- method for visualizing a conceptual surface called an isosurface
- isosurface is formed from a set of points in 3 space satisfying the equation $v = f(x, y, z)$
- v us called isovalue



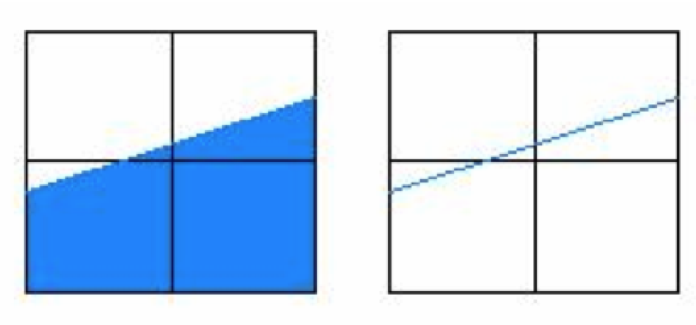
Obr.: Marching Cubes cases



Obr.: Sphere mesh with Marching cubes

Marching squares

- 2D equivalent
- $v = f(x, y)$

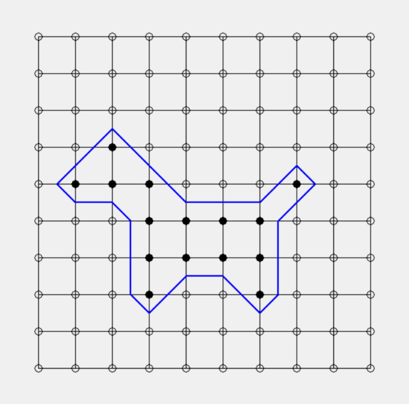


Obr.: Isosurface vs isocurve

Marching cubes

Alogorithm

- create grid with satisfying resolution
- sample function values at edges
- get binary mask-> evaluate whether vertex function value is under or above isovalue

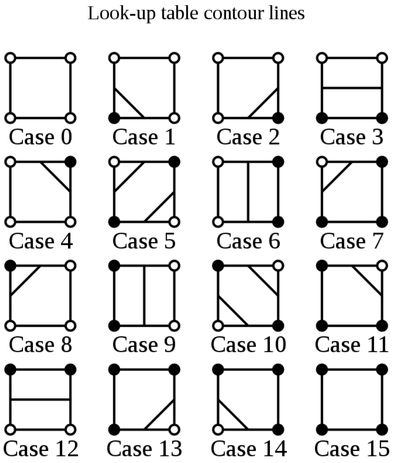


Obr.: Example

Marching cubes

Alogorithm

- evaluate cases and find edge points
- case 5 and case 10 ambiguous

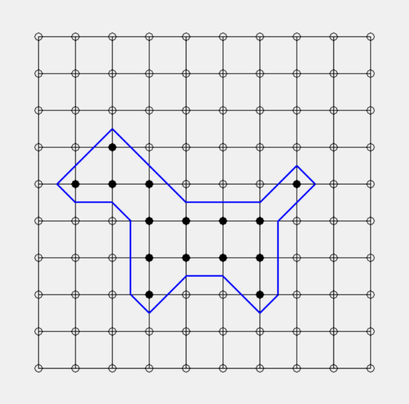


Obr.: Cases

Marching cubes

Alogorithm

- create grid with satisfying resolution
- sample function values at edges
- get binary mask-> evaluate whether vertex function value is under or above isovalue

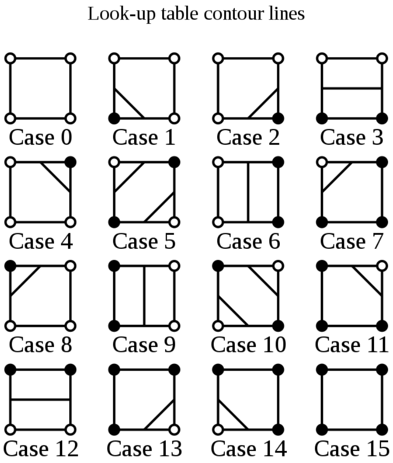


Obr.: Example

Marching cubes

Alogorithm

- draw lines between edge points



Obr.: Cases

Marching cubes

Alogorithm

Linear interpolation:

$$V(t) = (1 - t)V_1 + tV_2, \tag{1}$$

where $t = \frac{v - V_1.f}{V_2.f - V_1.f}$, V_1, V_2 are vertices, v is isovalue and f is function value evaluated at vertex

Marching cubes

Assignment

- implement marching cubes algorithm
- display function $f = (x - x0)^2 + ((y - y0) + \sqrt{|(x - x0)|})^2$

Algorithm

Cases

Square vertices and edges are ordered counterclockwise as 0, 1, 2, 3

```
case_to_edges = {
                #0: [],
                1: [[2, 3]],
                2: [[1, 2]],
                3: [[1, 3]],
                4: [[0, 1]],
                6: [[0, 2]],
                7: [[0, 3]],
                8: [[0, 3]],
                9: [[0, 2]],
                11: [[0, 1]],
                12: [[1, 3]],
                13: [[1, 2]],
                14: [[2, 3]],
                10: [[0, 1], [2, 3]],
                5: [[1, 2], [0, 3]],
                #15: []
        }
```

Thank you!