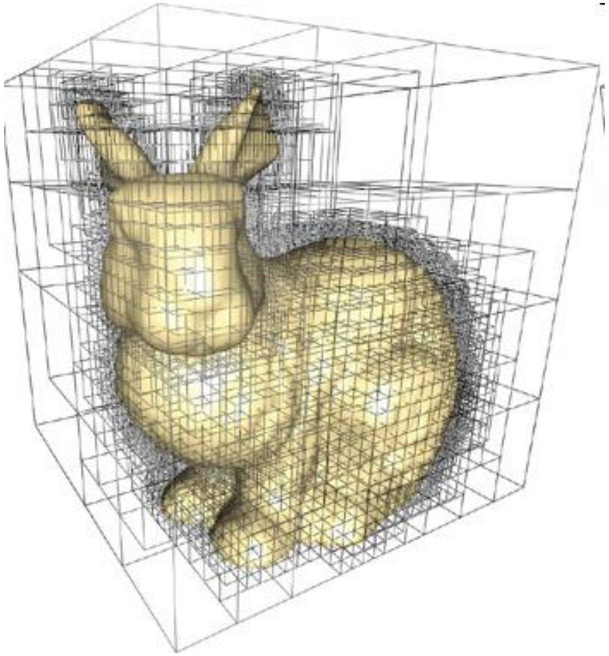
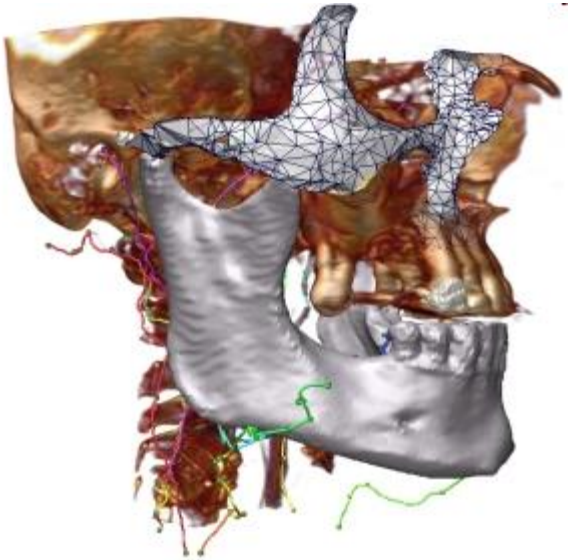
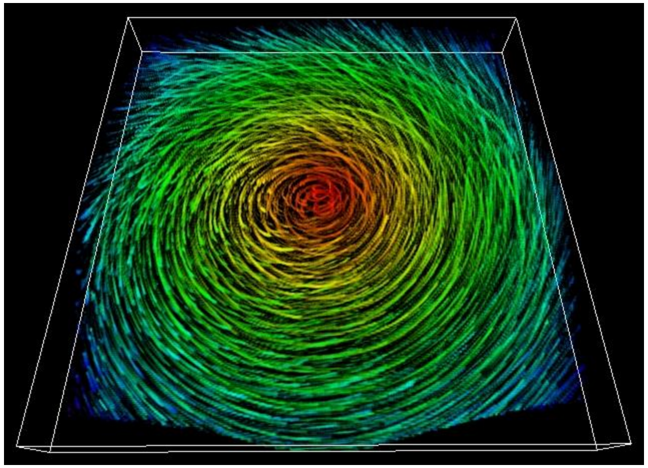
Diskrétne Geometrické Štruktúry

7. Vzdialenostné polia

Martin Samuelčík
samuelcik@sccg.sk, www.sccg.sk/~samuelcik, I4

Reprezentácia objemov

- Najčastejšie pomocou hodnôt v mriežke
  - Mriežka: pravidelná, octree, štvorstenná
  - Hodnoty: binárne, intenzity, vzdialenosti, vektory vzdialeností

Vzdialenostné polia

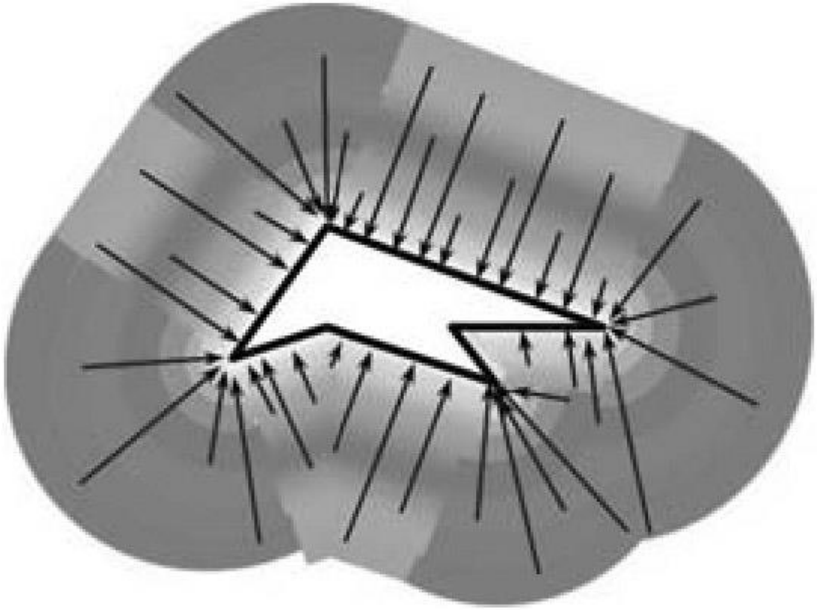
- Funkcia ururčujúca vzdialenosť k danému objektu – d: $R^3 \to R$
- Pre množinu Σ, vzdialenostná funkcia bez znamienka je $\mathrm{dist}_\Sigma(\mathbf{p}) = \inf_{\mathbf{x}\in\Sigma} \|\mathbf{x} - \mathbf{p}\|.$
- Rozšírenie – vzdialenostné vektory
- Vzdialenosť ku objektu S so znamienkom

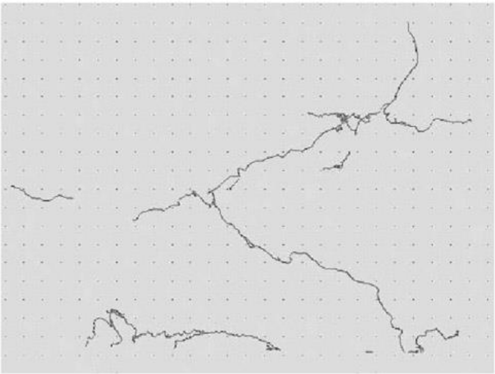
$$\mathrm{d}_S(\mathbf{p}) = \mathrm{sgn}(\mathbf{p}) \inf_{\mathbf{x}\in\partial S} \|\mathbf{x} - \mathbf{p}\|,$$

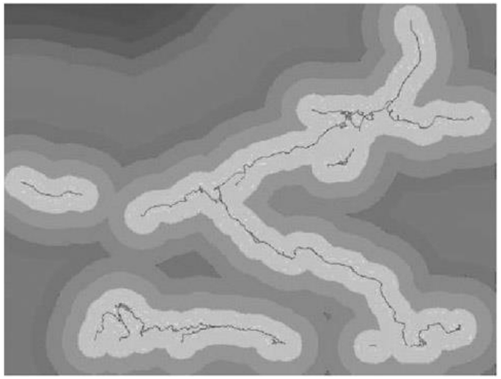
where

$$\mathrm{sgn}(\mathbf{p}) = \begin{cases} -1 & \text{if } \mathbf{p} \in S \\ 1 & \text{otherwise.} \end{cases}$$

Príklady







Vlastnosti

- Izopovrch pre izohodnotu τ: $\{\mathbf{p} | \mathrm{d}(\mathbf{p}) = \tau\}$
- Gradient $\|\nabla \mathrm{d}\| = 1$ pre takmer všetky body, smer gradientu kolmý na izopovrch
- Hesián

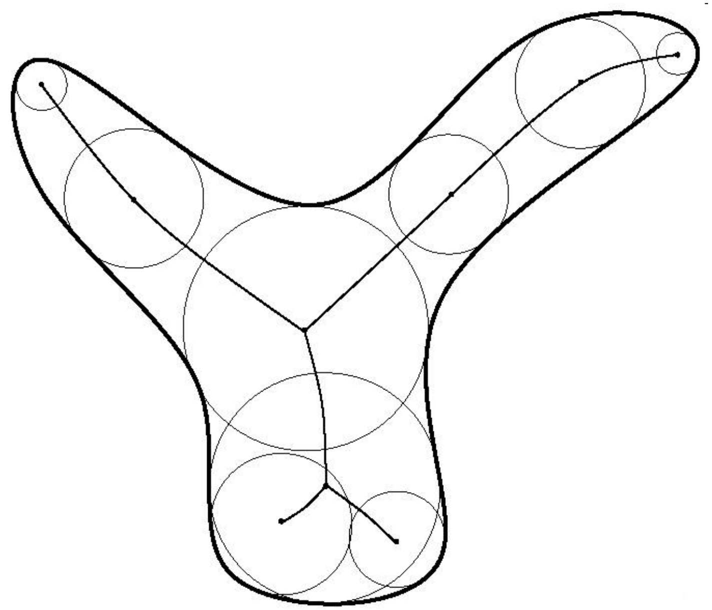
$$H = \begin{pmatrix} \mathrm{d}_{xx} & \mathrm{d}_{xy} & \mathrm{d}_{xz} \\ \mathrm{d}_{yx} & \mathrm{d}_{yy} & \mathrm{d}_{yz} \\ \mathrm{d}_{zx} & \mathrm{d}_{zy} & \mathrm{d}_{zz} \end{pmatrix}$$

- Hlavná krivosť $\kappa_M = \frac{1}{2} \left(\mathrm{d}_{xx} + \mathrm{d}_{yy} + \mathrm{d}_{zz} \right)$
- Gaussova krivosť

$$\kappa_G = \begin{vmatrix} \mathrm{d}_{xx} & \mathrm{d}_{xy} \\ \mathrm{d}_{yx} & \mathrm{d}_{yy} \end{vmatrix} + \begin{vmatrix} \mathrm{d}_{xx} & \mathrm{d}_{xz} \\ \mathrm{d}_{zx} & \mathrm{d}_{zz} \end{vmatrix} + \begin{vmatrix} \mathrm{d}_{yy} & \mathrm{d}_{yz} \\ \mathrm{d}_{zy} & \mathrm{d}_{zz} \end{vmatrix}$$

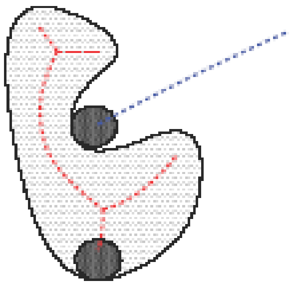
Vlastnosti

- Vzdialenostná funkcia je spojitá
- Problémové body – body rovnako vzdialené od aspoň dvoch bodov na povrchu – cut locus
- Pre C^k povrch, funkcia je C^k v nejakom okolí bodu na povrchu
- Funkcia je diferencovateľná až na body z cut locus



Diskretizácia

- Vzorkovanie vzdialenostnej funkcie
- Rozne formy vzoriek – mriežky, grid
- Je potrebné zachytiť detaily a určiť problémové miesta – cut locus
- Kritérium reprezentovateľnosti – vzdialenosť cut locus od povrchu musí byť väčšia ako rozlíšenie vzorkovania
- Potreba aproximácie gradientu



$$
\begin{aligned}
g^x_{i,j,k} &= d_{i+1,j,k} - d_{i-1,j,k} \\
g^y_{i,j,k} &= d_{i,j+1,k} - d_{i,j-1,k} \\
g^z_{i,j,k} &= d_{i,j,k+1} - d_{i,j,k-1} \\
n_{i,j,k} &= \frac{g_{i,j,k}}{\|g_{i,j,k}\|} .
\end{aligned}
$$

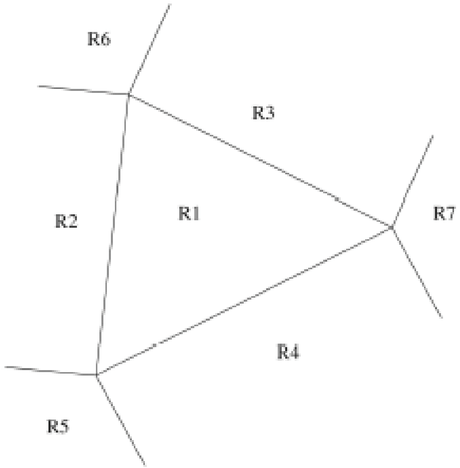
```
struct UniformGridDF
{
        int width, height, depth;
        array3D<float> voxels;
}
```

Výpočet DF (Voxelizácia)

- Distance transform
- Brute-force – pre každý mrežový bod (voxel) sa určí najmenšia vzdialenosť k objektu
- Časovo náročné
- Zlepšenia:
  - Na základe priestorového usporiadania častí objektu sa kontrolujú iba najbližšie objekty
  - Výpočet iba v niektorých častiach a následná propagácia do ďalších častí pomocou transformácií

Voxelizácia z trojuholníkov

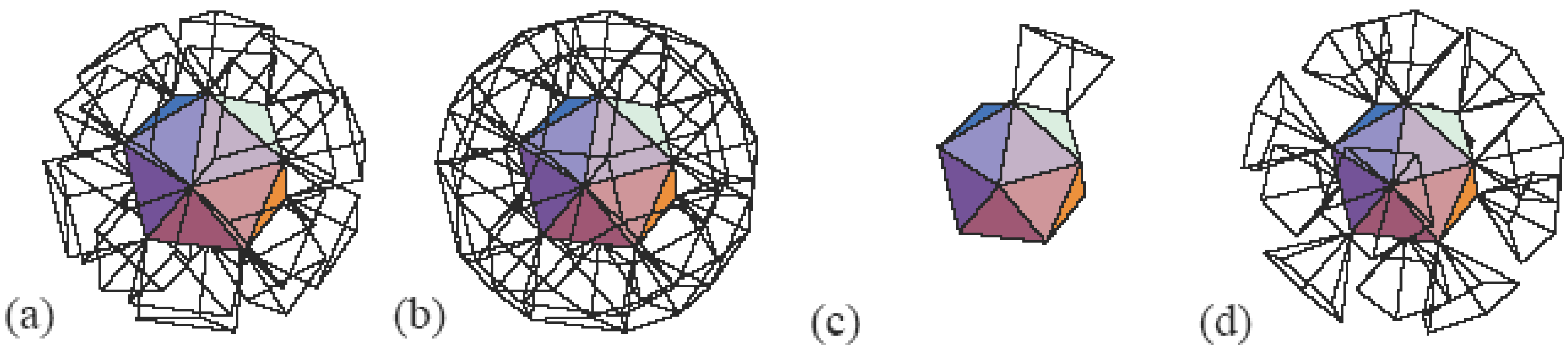
- Trojuholníky tvoria uzatvorené, orientovateľné 2-variety
- Vzdialenosť bodu a trojuholníka – 7 prípadov pri projekcii bodu do roviny trojuholníka
- Urýchlenie prehľadávania – ohraničujúce objemy, octrees



-Trojuholník ABC, bod S, hľadáme vzdialenosť VZD
- bod v rovine trojuholníka X = uA+vB+wC, u+v+w=1
- nech a=(A-C,A-C), b=(A-C,B-C), c = (B-C,B-C), d = (A-C,C-S), e=(B-C,C-S), f=(C-S,C-S)
- nech SX je projekcia S do roviny trojuholníka ABC, SX=sA+tB+(1-s-t)C
- s=(be-cd)/(ac-b^2), t=(bd-ae)/(ac-b^2)
- ak 0≤s≤1, 0≤t≤1, 0≤1-s-t≤1, tak SX je vnútri trojuholníka, hľadaná vzdialenosť je VZD=|S,SX|
- inak ak s<0, treba nájsť v rovine trojuholníka priemet SXS bodu SX na priamku BC
    - SXS=pB+(1-p)C, p=(SX-C,B-C)/(B-C,B-C)
    - ak p<0, tak VZD=|S,C|
    - ak 0≤p≤1 tak VZD=|S, SXS|
    - ak p>1 tak VZD=|S,B|
- podobne pre t<0, (1-s-t)<0

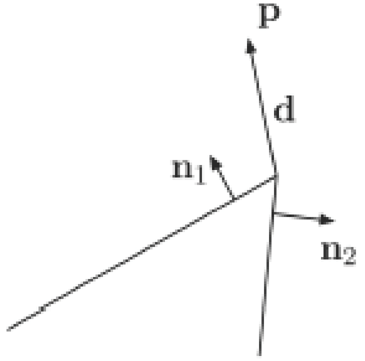
Lokálne metódy

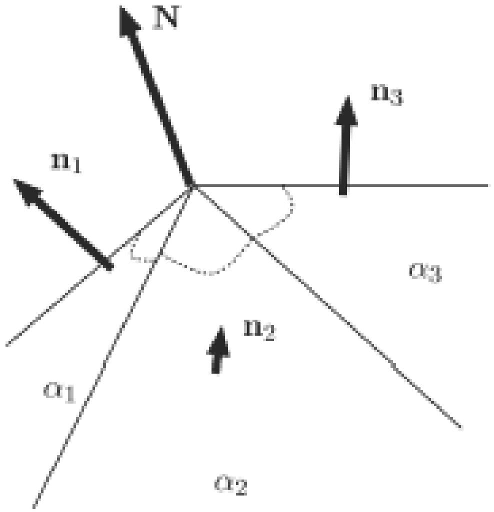
- Pre výpočet vzdialenosti do určitej veľkosti
- Rozšírenie ohraničujúceho objemu trojuholníka o danú hodnotu
- Identifikácia oblastí vzdialených od povrchu o max. danú hodnotu



Výpočet znamienka

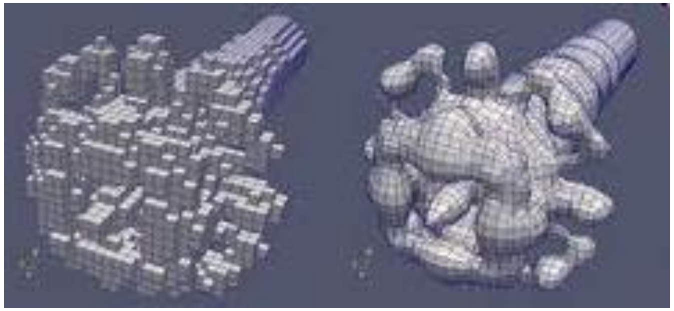
- Pre C^1 povrchy, stačí nájsť skalárny súčin vektora vzdialenosti a normály
- Množina trojuholníkov nie je C^1
- Iný výpočet:
  - Pseudonormály (podľa uhlov)
  - Vysielanie lúča
  - Rezy (konverzia do 2D)





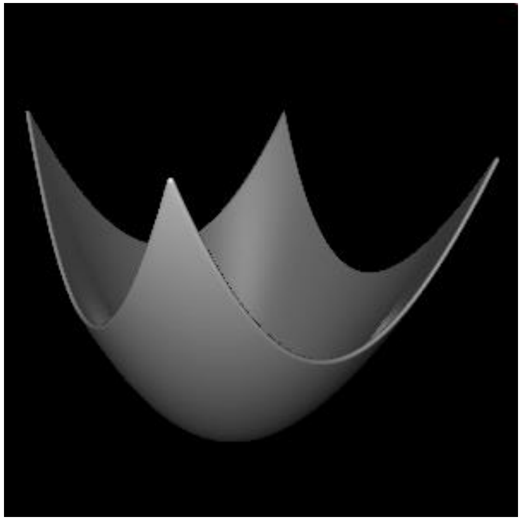
Implicitné povrchy

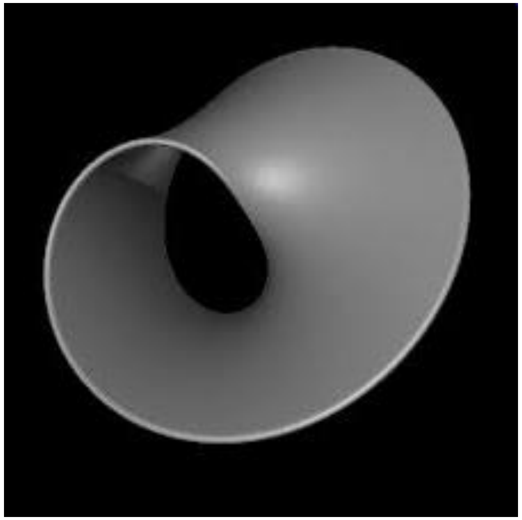
- Izopovrch funkcie $B = \{\mathbf{p} |\ f(\mathbf{p}) = \tau\},$
- Pre niektoré povrchy je f priamo vzdialenostná funkcia (sféra, ...)
- Vzorkovanie funkcie $(f - \tau)/||\nabla f||$
- Presnejšie hľadanie bodu na izopovrchu s najmenšou vzdialenosťou numericky iteratívne v smere gradientu



Parametrické povrchy

- Zmena reprezentácie na implicitnú alebo polyhedrálnu
- Minimalizácia $\mathrm{d}(u,v) = ||\mathbf{S}(u,v) - \mathbf{p}||$
- Numerické riešenia
- Použitie iteratívnych postupov





Distance Transforms

- Výpočet vzdialeností na základe informácií blízko povrchu
- Spôsob prechodu mriežkou:
  - Zametanie – po jednotlivých rezoch a riadkoch
  - Wavefront – od povrchu ku stále väčším vzdialenostiam
- Výpočet pre voxel:
  - Chamfer:
    - Nová vzdialenosť pre voxel je vypočítaná zo vzdialeností okolitých voxelov
  - Vector:
    - Nový vektor vzdialenosti je vypočítaný z vektorov okolitých voxelov
  - Eikonal:
    - Vzdialenosť sa vypĺňa vo voxeloch iteratívnym riešením diferenciálnej rovnice

Inicializácia

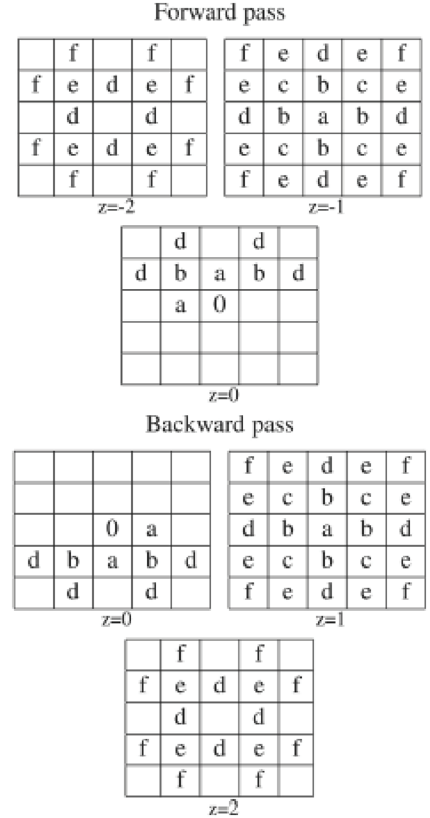
- Na základe výpočtov presnejšími metódami
- Určenie v okolí povrchu $F(\mathbf{p}) = \begin{cases} 0 & \mathbf{p} \text{ is exterior} \\ \infty & \mathbf{p} \text{ is interior,} \end{cases}$
- Ohraničenie povrchu $F(\mathbf{p}) = \begin{cases} \mathrm{d}_S(\mathbf{p}) & \text{in the shell} \\ \infty & \text{elsewhere.} \end{cases}$
- Distance transformation vyplní všetky voxle, kde ešte nemáme určenú vzdialenosť
- Transformácia prechádza celým poľom a vypĺňa ešte nevyplnené údaje
- Niekoľko prechodov, iterácií

Chamfer

- Sweeping:

```
/* Forward Pass */
FOR(z = 0; z < f_z; z++)
  FOR(y = 0; y < f_y; y++)
    FOR(x = 0; x < f_x; x++)
      F[x,y,z] =
 inf_{∀i,j,k∈fp}(F[x+i,y+j,z+k]+m[i,j,k])

/* Backward Pass */
FOR(z = f_z-1; z ≥ 0; z--)
  FOR(y = f_y-1; y ≥ 0; y--)
    FOR(x = f_x-1; x ≥ 0; x--)
      F[x,y,z] =
 inf_{∀i,j,k∈bp}(F[x+i,y+j,z+k]+m[i,j,k])
```



- Wavefront:
  - Prioritná fronta pre voxely s najmenšou vzdialenosťou

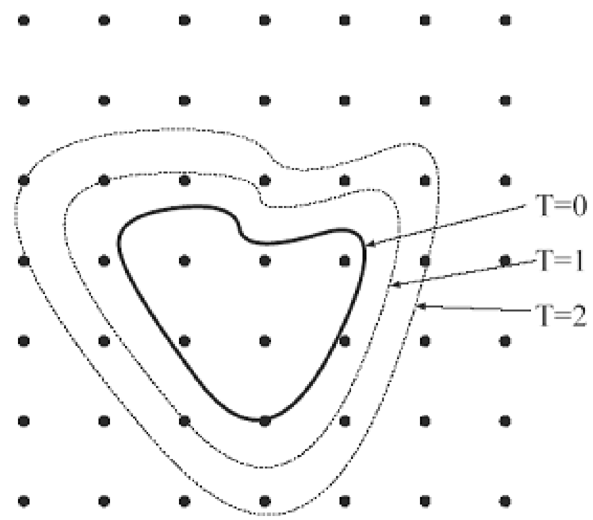
Transform	a	b	c	d	e	f
City Block (Manhattan)	1					
Chessboard	1	1				
Quasi-Euclidean $3 \times 3 \times 3$	1	$\sqrt{2}$				
Complete Euclidean $3 \times 3 \times 3$	1	$\sqrt{2}$	$\sqrt{3}$			
$< a, b, c >_{opt}$ $3 \times 3 \times 3$[102]	0.92644	1.34065	1.65849			
Quasi-Euclidean $5 \times 5 \times 5$	1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{5}$	$\sqrt{6}$	3

Fast Marching Method

- Expandujúci povrch s konštantnou rýchlosťou – nafukovanie balóna
- Čas dorazenia balónika do voxla = vzdialenosť voxla od povrchu
- T – čas príchodu povrchu do x
- F – rýchlosť
  - Pre nás F je konštanta

$$||\nabla T(\mathbf{x})||F(\mathbf{x}) = 1$$

- „frozen" vrchol – už má určenú výslednú vzdialenosť
- „narrow band" – už je určená vzdialenosť, ale ešte nie je výsledná
- H – množina „narrow band" vrcholov, prioritná fronta



Fast Marching Method

```
Initialization()
{
        for each voxel v in I
            {
                Freeze v;
                for each neighbour vn of v
                    {
                        compute distance d at vn;
                        if vn is not in narrow band
                            {
                                tag vn as narrow band;
                                insert (d,vn) in H;
                            }
                        else
                            decrease key of vn in H to d;
                    }
            }
}
```

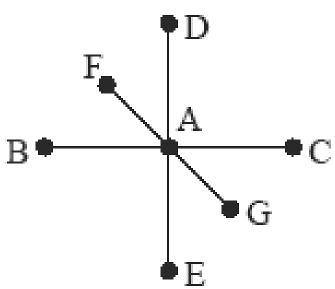
```
Loop()
{
    while H is not empty
        {
            Extract v from top of H;
            Freeze v;
            for each neighbour vn of v
                if vn is not frozen
                    {
                        compute distance d at vn;
                        if vn is not in narrow band
                            {
                                tag vn as narrow band;
                                insert (d,vn) in H;
                            }
                        else
                            decrease key of vn in H to d;
                    }
        }
}
```

$$D_2^{-x}G = \frac{3G[x,y,z] - 4G[x-1,y,z] + G[x-2,y,z]}{2}$$

$$D_2^{+x}G = -\frac{3G[x,y,z] - 4G[x+1,y,z] + G[x+2,y,z]}{2}$$

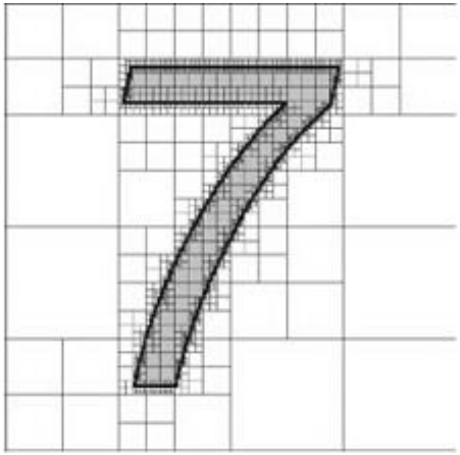
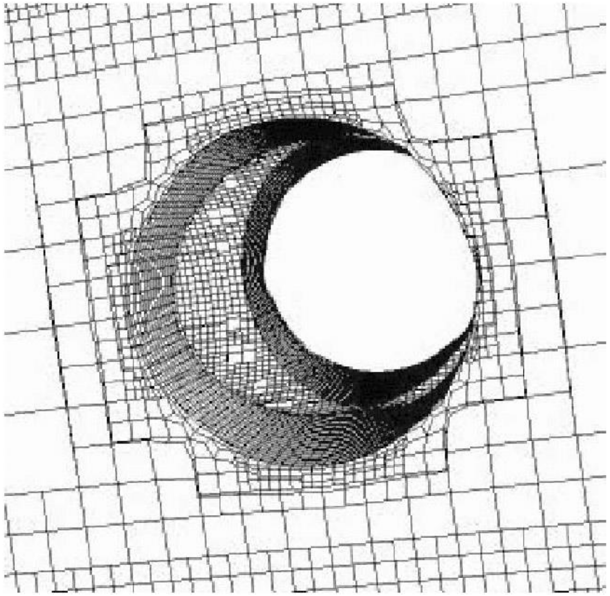
$$1/F^2 = \begin{cases} \max(D_2^{-x}G, -D_2^{+x}G, 0)^2 + \\ \max(D_2^{-y}G, -D_2^{+y}G, 0)^2 + \\ \max(D_2^{-z}G, -D_2^{+z}G, 0)^2 \end{cases}$$

$$||\nabla T||^2 = \begin{cases} \max(V_A - V_B, V_A - V_C, 0)^2 & + \\ \max(V_A - V_D, V_A - V_E, 0)^2 & + \\ \max(V_A - V_F, V_A - V_G, 0)^2 \end{cases}$$



Reprezentácie

- Regulárna mriežka
- Hierarchická mriežka
- Adaptívne pole – octree
- Uchovávanie predchádzajúcej reprezentácie
- Orezávanie hodnôt

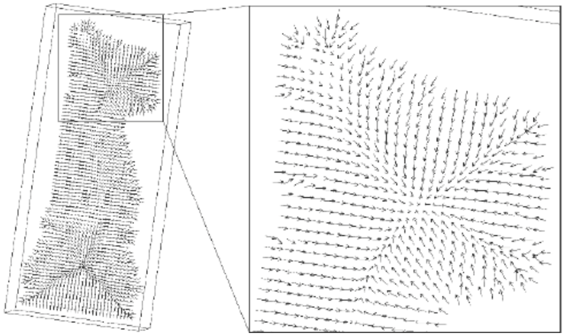




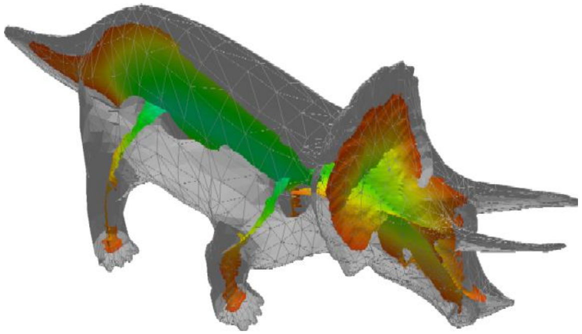
Aplikácie

- Reprezentácia objektov, výhoda oproti binárnym voxelom, napr. approximačné určenie najbližšieho bodu $\mathbf{p}_f = \mathbf{p} - \nabla d_S(\mathbf{p}) d_S(\mathbf{p})$
- Modelovacie schopnosti
- Animácie, morfing
- Image processing
- Fyzikálne simulácie
- Fonty

Kostra, stredná os

- Vytvorenie jednoduchého objektu ktorý aproximuje daný objekt
- Využitie v kinematike, ...
- Analýza DF
  - Hľadanie nespojitostí v derivácii DF
  - Porovnávanie smerových vektorov k najbližším bodom na povrchu
  - Hľadanie najväčších vzdialeností vnútri objektu





Reprezentácia fontov

- http://www.valvesoftware.com/publications/2007/SIGGRAPH2007_AlphaTestedMagnification.pdf
- Namiesto bitmapy sa použije vzdialenostné pole = presnejšia reprezentácia hranice

(a) 64x64 texture, alpha-blended (b) 64x64 texture, alpha tested (c) 64x64 texture using our technique

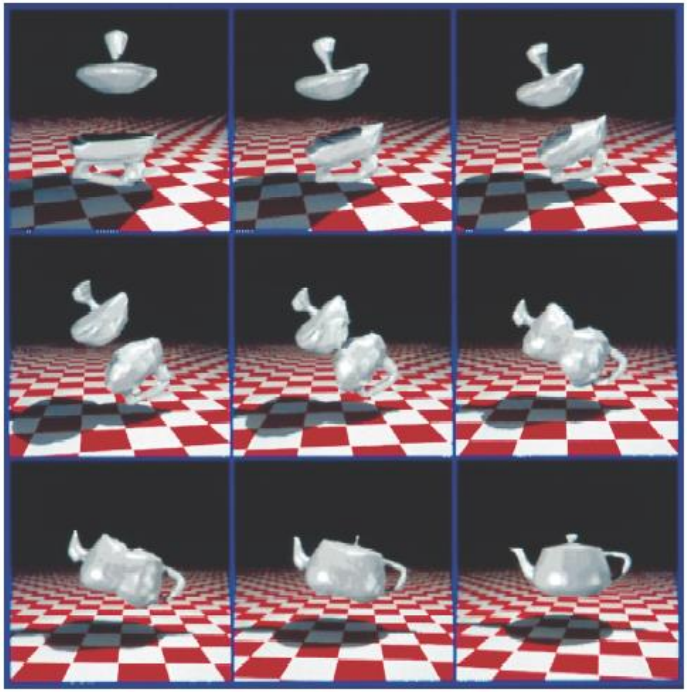
Reprezentácia fontov

(a) High resolution input

(b) 64x64 Distance field

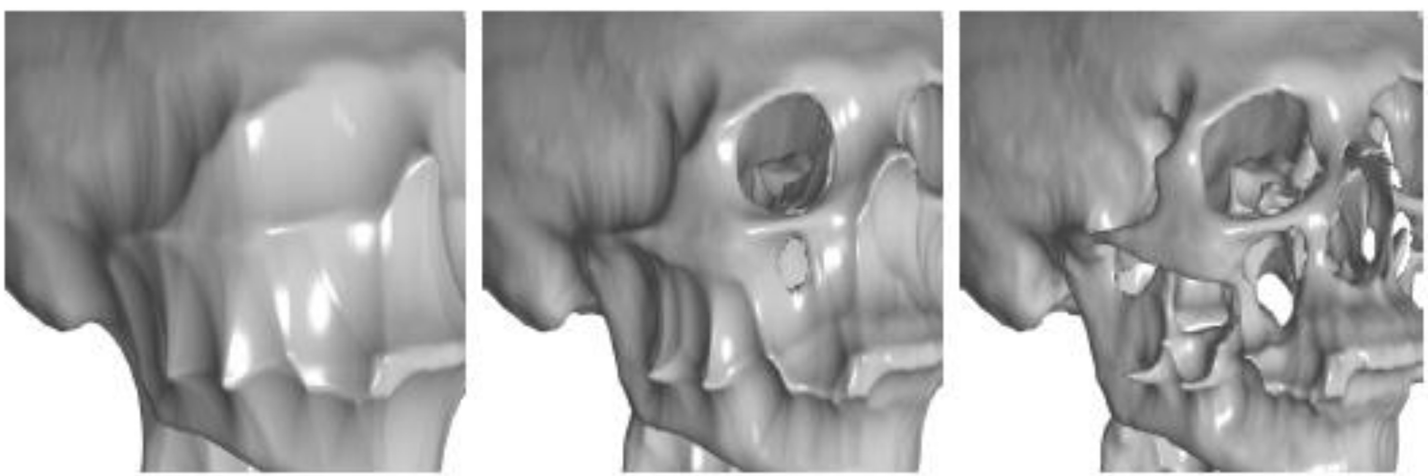
Morfing

- Prechod medzi dvoma objektami v čase
- Nezáleží na rode objektu
- Potrebné zjednotiť aproxoimácie oboch funkcií
- Priloženie objektov



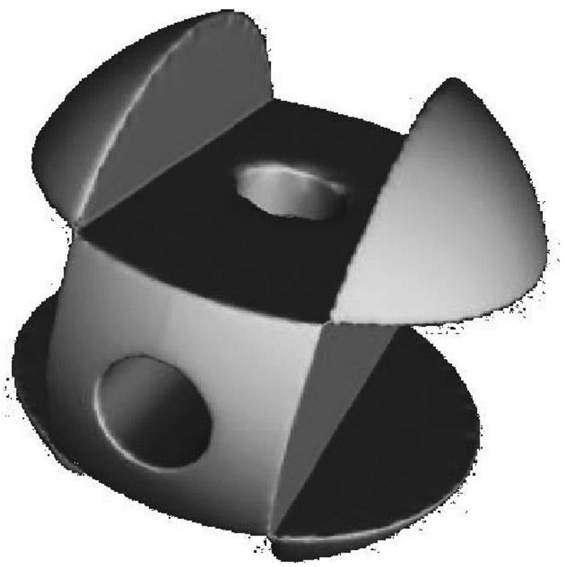
Morfológia

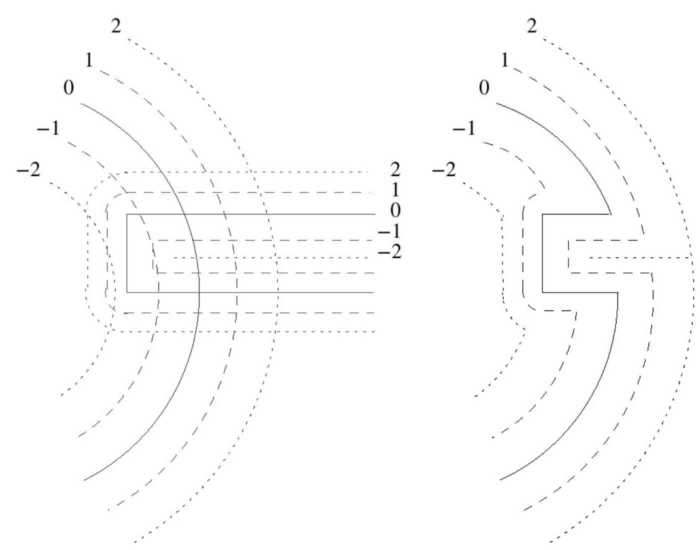
- Operácie pre úpravu diskrétneho signálu
- Erózia $X \ominus B = \{\mathbf{p} | B_{\mathbf{p}} \subset X\}$
- Dilatácia $X \oplus B = \{\mathbf{p} | B_{\mathbf{p}} \cap X \neq \emptyset\}$
- Uzavretie $X \bullet B = (X \oplus B) \ominus B$ $S^{\bullet_r} = \{\mathbf{q} : d'(\mathbf{q}) = -r\}$ $S^{\oplus_r} = \{\mathbf{q} : d(\mathbf{q}) = r\}$
- Otvorenie $X \circ B = (X \ominus B) \oplus B.$



CSG operácie

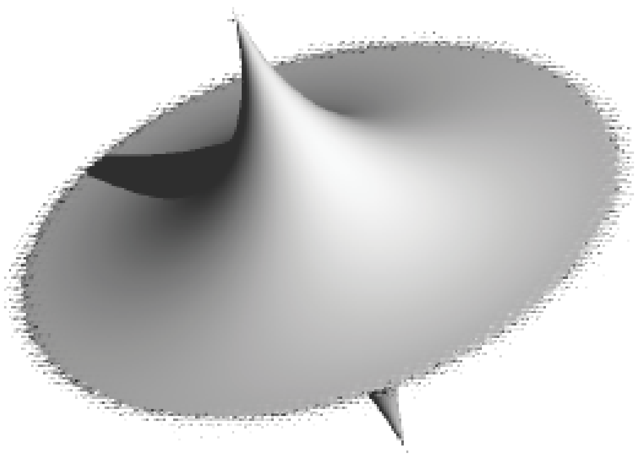
- Jednoduché a rýchle výpočty prienikov, zjednotení a rozdielov, často je daná DF len blízko povrchu
- Zjednotenie - D=min(D1,D2)
- Prienik - D=max(D1,D2)
- Rozdiel - D=max(D1,-D2)
- Iba aproximácia vzdialenosti blízko ostrých rohov

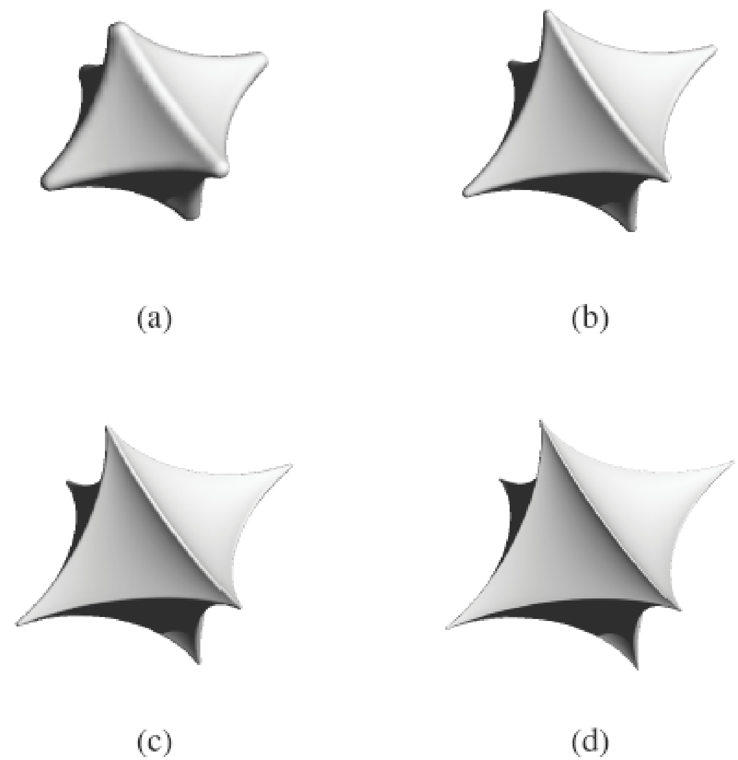




Odstraňovanie artefaktov

- http://www.sccg.sk/~novotny/doc/vg05.pdf
- V miestach s nedostatočným vzorkovaním
- Vyplňovanie vzolov kde nemame dostatočnú informáciu
- Záleží na rozlíšení



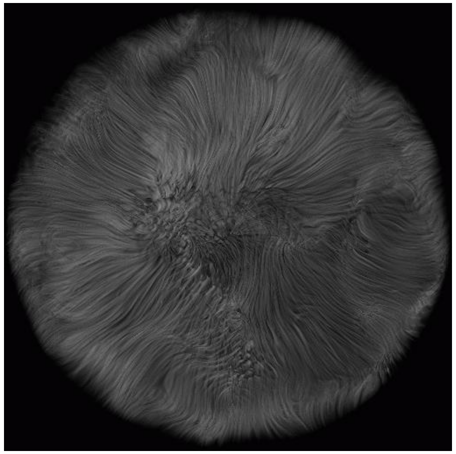


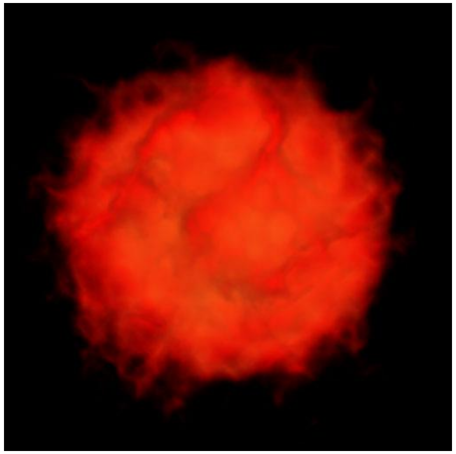
Hypertextúry

- Pridanie detailu nad povrch – srsť, oheň, dym
- Určenie regiónu pre namapovanie textúry

$$D(p) = \begin{cases} 1 & \text{if } \mathrm{d}(p)^2 \le r_i^2 \\ 0 & \text{if } \mathrm{d}(p)^2 \ge r_o^2 \\ \frac{r_o^2 - \mathrm{d}(p)^2}{r_o^2 - r_i^2} & \text{otherwise,} \end{cases}$$

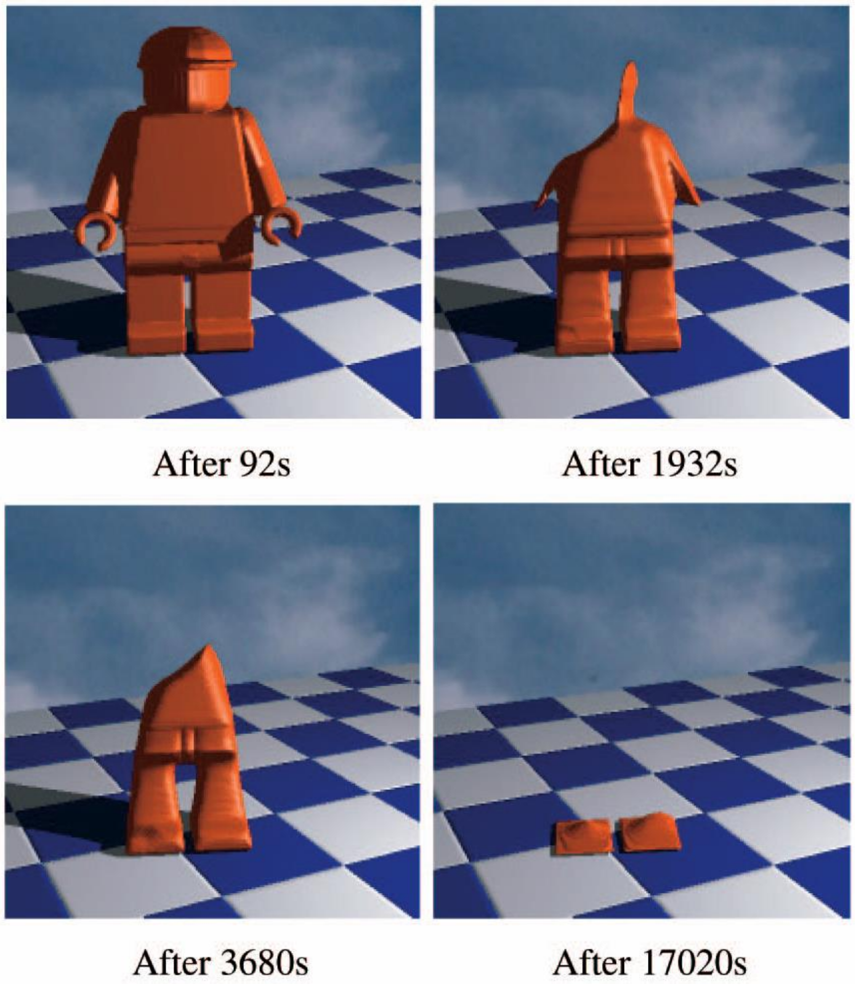
- Pomocou d(p) sa určia ďalšie vlastnosti ako smer, dotyková plocha, tvorba šumu, ...





Spracovanie meshov

- Zjemňovanie povrchov
- Kontrola chyby pri zmene povrchov
- Porovnávanie objektov
- Detekcia kolízií
- Simulácie, animácie

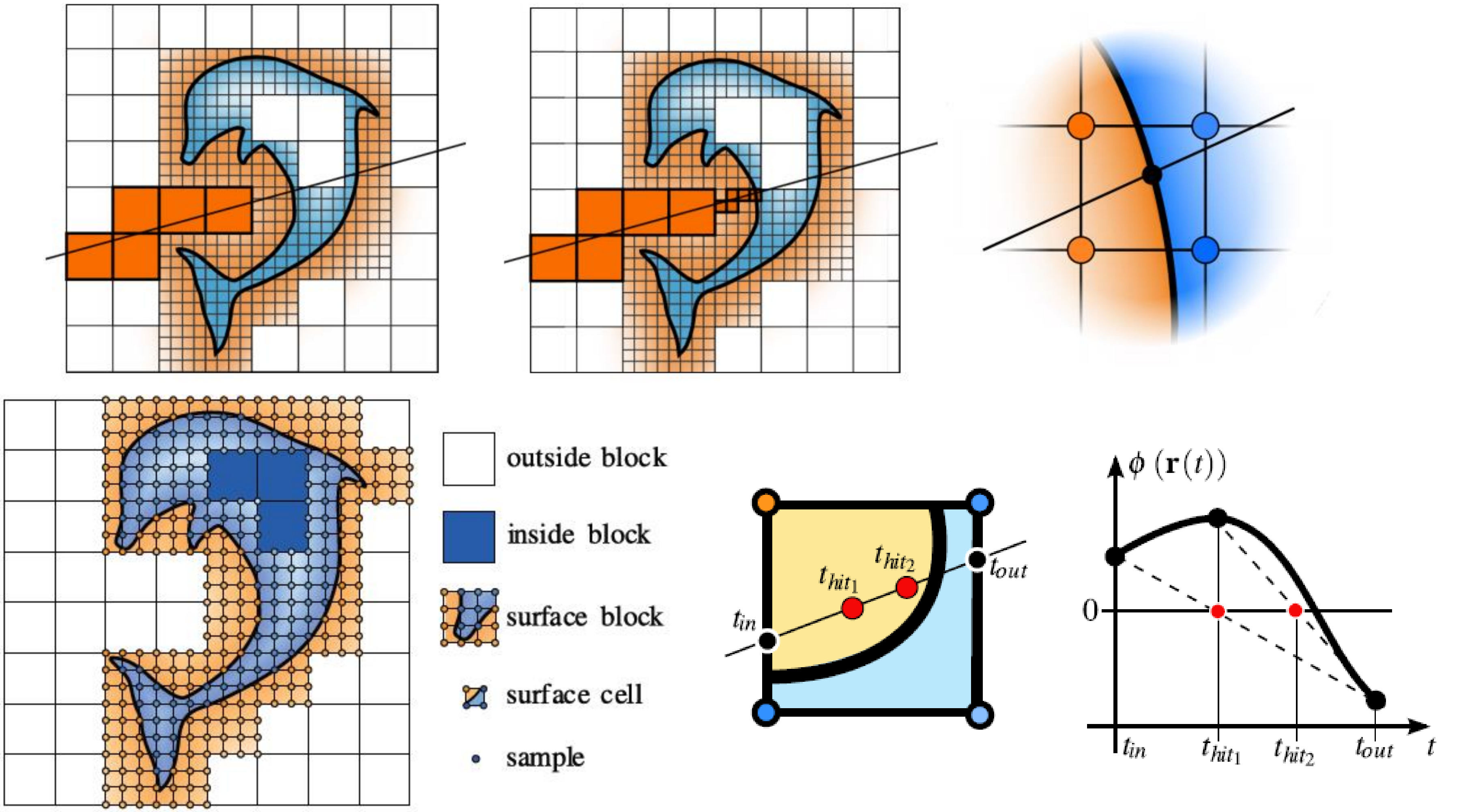


Vizualizácia

- Prevod do iných reprezentácií:
  - Polyhedrálna – marching cubes
  - Mračno bodov – priemety voxlov na povrch
- Vizualizácia 3D objemov
- Raytracing:
  - Určujú regióny kde sa povrch nenachádza
  - Prechádzame priestorom pozdĺž lúča
  - Nájdeme prvý voxel pozdĺž cesty obsahujúci povrch
  - Nájdeme presný prienik pomocou interpolácie vzdialenostnej funkcie

Raytracing

http://dcgi.felk.cvut.cz/_media/en/events/praguecvut-jamriska-ondrej.pdf



Otázky?