

8 Izobličavanje objekata (engl. morphing)

- deformiranje (*warping*)
 - u prostoru slike – 2D
http://alteredqualia.com/xg/examples/synthetic_celebrities_interpolation.html
 - deformiranje projekcije
 - deformiranje objekta – 3D
- područje djelovanja deformacije
 - globalno [/](#)
 - lokalno



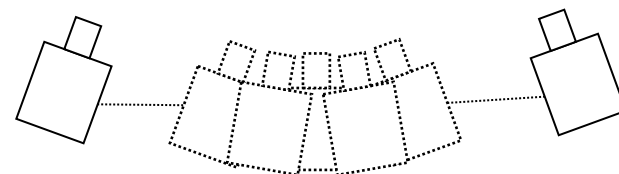
Deformiranje u prostoru slike

- različiti alati za obradu slike (geometrijske transformacije tekstone slike upravljano kontrolnim točkama mreže)
 - <https://testdrive-archive.azurewebsites.net/Graphics/Warp/Default.html>
- deformacije ovisne o sadržaju slike – upravljane deformacije
- <https://www.auduno.com/clmtrackr/examples/facesubstitution.html>
- primjena dubokog učenja u sintezi slike
- http://alteredqualia.com/xg/examples/eyes_gaze.html

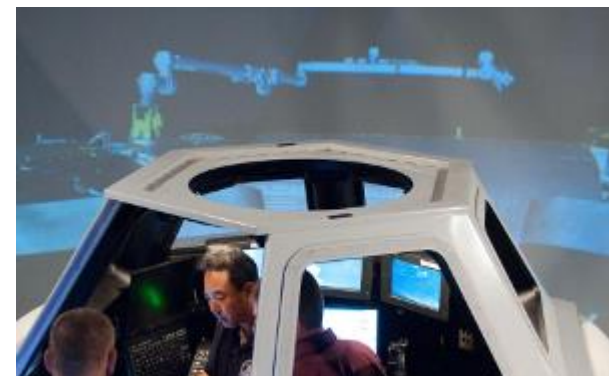


Deformiranje (interpolacija) projekcije

- zadano je nekoliko položaja kamere i pripadne projekcije - interpoliranje među položaja kamere i pripadnih projekcija
- <http://www-2.cs.cmu.edu/~seitz/vmorph/mona.mpg>
- projekcije na fasade
(distorzije ovisno o poziciji projektora)



- prilagođavanje projekcije obliku platna - kalibracija
npr. projekcija na kupolu kod simulatora leta
- prilagođavanje intenziteta projicirane slike
na dijelu preklapanja projekcija kod više projektora

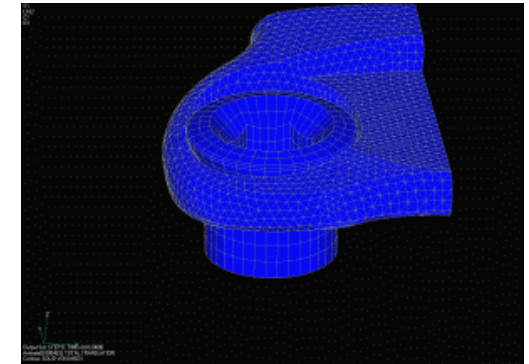
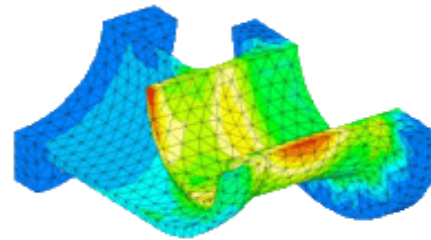


Deformiranje objekta

1. fizikalno temeljene deformacije

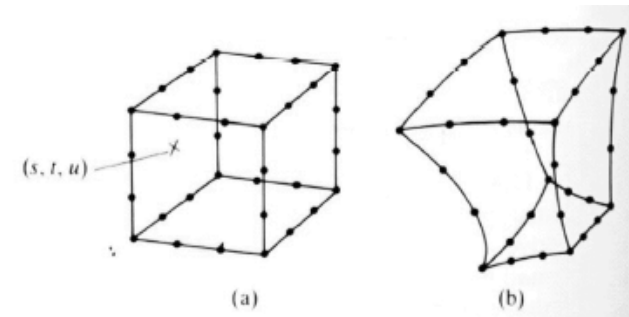
- proračun konačnim elementima (FEM finite element method, FEA, BEA Boundary Element Analysis, Transient Data Animation) <http://vimeo.com/44652965>
(simulacija sudara, dinamika fluida ili polja oko objekta)
- kolizije (sile-deformacije), sustav čestica

<http://fem-mobile.appspot.com/>



2. geometrijski temeljene tehnike

- globalne i lokalne
- nelinearne globalne transformacije
- slobodne deformacije (engl. FFD Free Form Deformations), modeliranje površine (https://jsfiddle.net/human_a/9m8k8v4q/)
npr. dinamički NURBS, točkama kontrolnog poligona oblikujemo površinu i promjene u vremenu



Geometrijski temeljene tehnike

- skaliranje $[x_n \ y_n \ z_n] = [s_x x \ s_y y \ s_z z]$
- nelinearne globalne transformacije

- sužavanje (engl. tapering)

$$[x_n \ y_n \ z_n] = [f(z)x \ f(z)y \ z]$$

- uvijanje (engl. twisting)

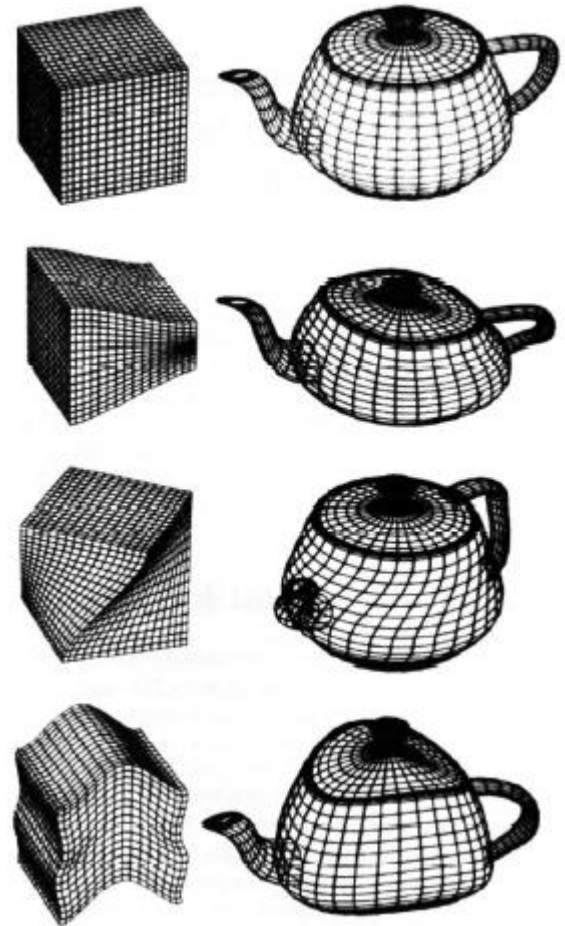
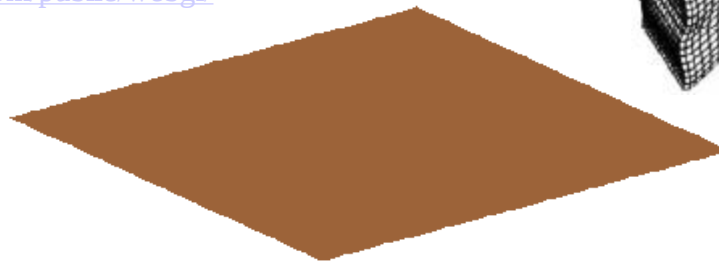
$$[x_n \ y_n \ z_n] = [(x \cos(z \Theta) - y \sin(z \Theta)) \\ (x \sin(z \Theta) + y \cos(z \Theta)) \ z],$$

https://threejs.org/examples/webgl_morphtargets

https://threejs.org/examples/webgl_materials_modified.html

- savijanje (engl. bending)

<http://steveharoz.com/public/webgl/>



8.1 Upravljanje deformiranjem objekta

- upravljanje deformacijom

- upravljanje kosturom (skeletal animation)

- kod upravljanja deformacijom kože koriste se kosti i mišići za upravljanje (*skin – skeleton animation, bone system*)

<http://game.akjava.com/weight/weight.html> <http://game.akjava.com/poseeditor/>

http://lo-th.github.io/olymp/index_diana.html

- upravljanje površinom objekta http://takahirox.github.io/three.js/examples/webgl_morphtargets_human.html

- https://www.blend4web.com/apps/code_snippets/code_snippets.html?scene=morphing

- kontrolne točke

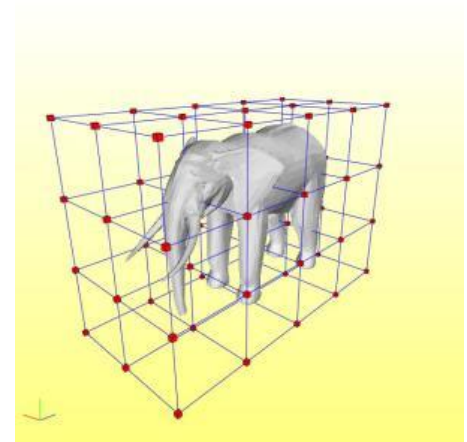
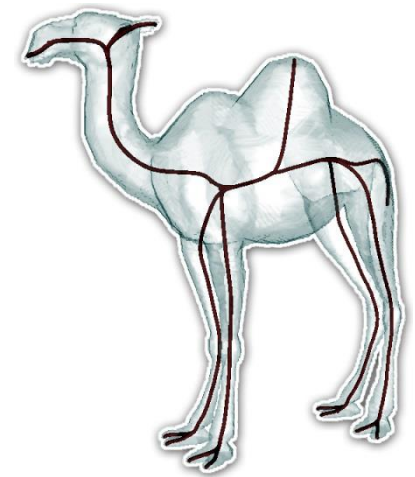
- B-površine su pogodne kada postoji pravilna mreža kontrolnih točaka kojima je površina definirana (gibanje kontrolnih točaka po krivuljama)
 - trokutne mreže ili razasuti podaci nemaju kontrolnu mrežu
 - <https://lucasmajerowicz.github.io/threejs-ffd/app/>

- načini upravljanja deformacijom

- animator određuje promjene - promjene dobivene postupkom modeliranja (extrude → morph) <https://johnrobinson.github.io/sandtoy/>

- fizikalno temeljene promjene/kombinirano http://alteredqualia.com/xg/examples/liquid_face.html

- upravljanje na osnovi snimljenih točaka stvarnog objekta i prenošenje promjena na objekt u računalu koji je isti ili drugačiji (kloniranje izraza lica)



- prednosti i nedostaci B-splajna u animaciji
 - objekt je potrebno razdijeliti u dijelove površine
 - prednost je da je objekt gladak i ima kontrolnu mrežu za upravljanje
 - nedostatak je problem numeričke točnosti na “šavovima” (npr. Toy Story)
- miješanje različitih izraza lica/poza tijela težinskim funkcijama (engl. blend)
 - miješanja neizrazitom logikom (engl. fuzzy logic)

https://threejs.org/examples/webgl_morphtargets_face.html
- razasuti podaci
 - mekani objekti (deformabilni)
 - interpolacija razasutih podataka (engl. scattered data interpolation)



8.2 Dinamika mekanih objekata (Soft Body)

- objekte možemo predstaviti kao pune (volumne) ili samo vanjskom površinom - slojem (obično nije dovoljno)

- <https://lagged.com/play/2566/>
- <https://webgl-examples.appspot.com/square-melon/melon.html>



- simulacija elastičnih (vraćaju se u prvobitni oblik) ili plastičnih deformacija

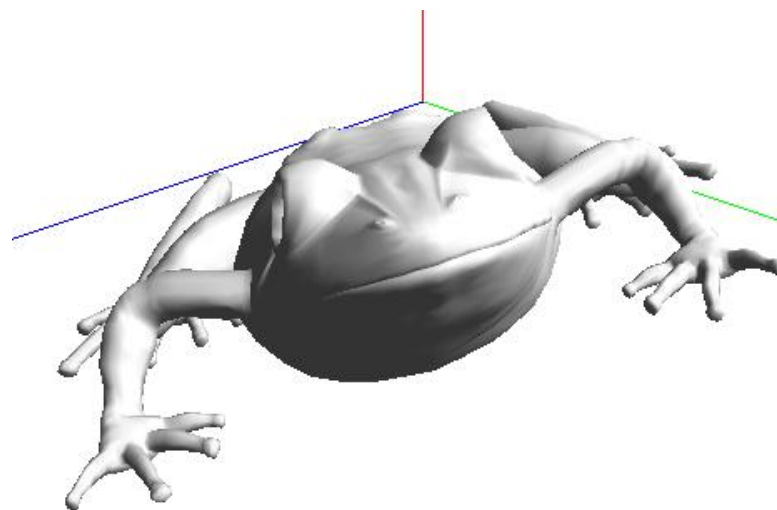
- https://rawcdn.github.com/kripken/ammo.js/99d0ec0b1e26d7ccc13e013caba8e8a5c98d953b/examples/webgl_demo_softbody_volume/index.html
- http://lo-th.github.io/Ammo.lab/#soft_pig

- reprezentacija (aproksimacija) objekta

- konačni elementi, konačni volumeni
<http://egraether.com/demos/soft/index.html>
- samo točke (bez topoloških podataka)

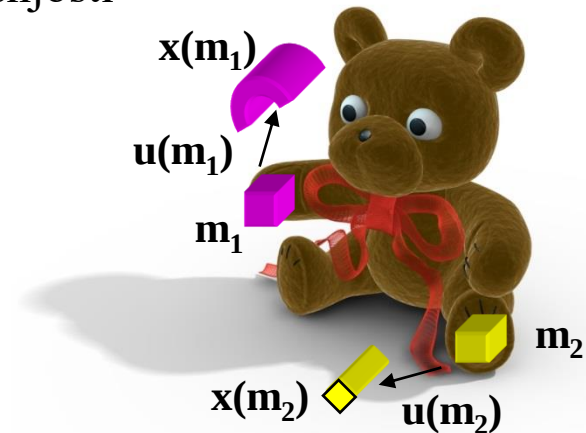
- model sastavljen od masa i opruga

- [Žaba](#)



Reprezentacija objekta

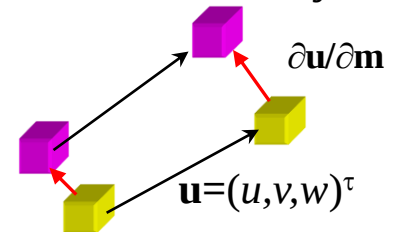
- početni nedeformirani (inicijalni, prvobitni) oblik predstavlja ravnotežno (rest) stanje:
 - skup kontinuirano povezanih gradivnih elemenata $\mathbf{m} \in M$, M je $\mathbf{R}^3$ određenih koordinatama točaka objekta u prvobitnom položaju
 - nove lokacije $\mathbf{x}(\mathbf{m})$ točaka (gradivnih elemenata) nastaju pod djelovanjem sila na objekt – to je vektorsko polje definirano na M
 - vektorsko polje pomaka je $\mathbf{u}(\mathbf{m}) = \mathbf{x}(\mathbf{m}) - \mathbf{m} = (u, v, w)^T$
 - u slučaju diskretizacije M je skup diskretnih koordinata
 - $\mathbf{m}$ su u općem slučaju proizvoljno zakrivljenog oblika i pod utjecajem sila se kontinuirano deformiraju ili se radi aproksimacija konačnim elementima
 - konačni elementi – deformacije se računaju za skup konačnih elementa, a za jedan element se vrijednosti u unutrašnjosti računaju (linearnim) baznim funkcijama
 - elastične sile koje djeluju na objekt ovise o $\mathbf{u}(\mathbf{m})$



Mjera deformacije objekta

- vektori koji predstavljaju pomake gradivnih elementa $\mathbf{u}(\mathbf{m})$ nisu dobra mjera jer komponente mogu postati proizvoljno velike (npr. translateramo objekt a ne deformiramo ga)
- bolja mjera je gradijent vektora deformacija

$$\nabla \mathbf{u} = \begin{bmatrix} u_{,x} & u_{,y} & u_{,z} \\ v_{,x} & v_{,y} & v_{,z} \\ w_{,x} & w_{,y} & w_{,z} \end{bmatrix}$$

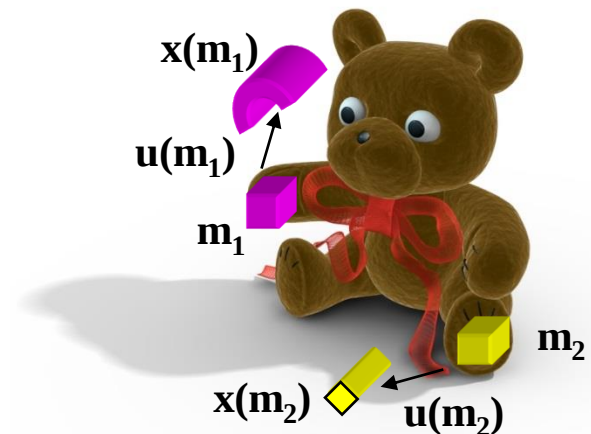


- ova mjera nije invarijantna na rotacije (npr. ako tijelo samo rotiramo $\mathbf{F}=\mathbf{R}$) pa se definira mjera invarijantna na transformacije čvrstog tijela tj.
- Tenzor naprezanja (engl. *Green-Lagrange strain tensor*)

$$\varepsilon_G = \frac{1}{2} \left(\nabla u + [\nabla u]^T + [\nabla u]^T \nabla u \right)$$

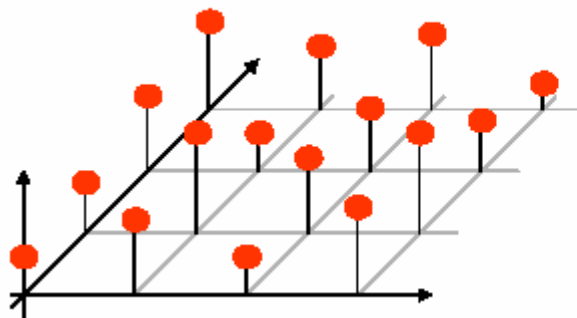
ε_G je nelinearan - kvadratana funkcija od m
Cauchy tenzor naprezanja - aproksimacija

$$\varepsilon_C = \frac{1}{2} \left(\nabla u + [\nabla u]^T \right)$$

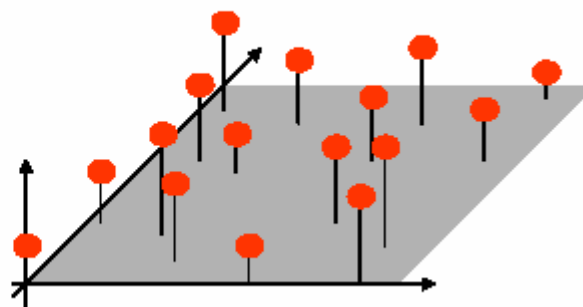


8.3 Interpolacija razasutih podataka (engl. scattered data interpolation)

- želimo mali broj kontrolnih točaka koje definiraju skup razasutih podataka
- kontrolnim točkama želimo upravljati velikim brojem točaka u animaciji (želimo izbjeći topološku strukturu kao kod B-površina)
- dobre rezultate pokazuje korištenje RBF (engl. radial basis functions) u ovoj interpolaciji
- RBF – površina dobivena kombinacijom (zbroy) radijalnih funkcija
- <http://multivis.net/lecture/radial-basis.html>
- https://oosmoxiecode.com/archive/js_webgl/isogrid/
- implicitno definirane površine
- (npr. Metaballs) <http://stemkoski.github.io/Three.js/Metaballs.html> <http://www.ro.me/tech/metaball-playground>



pravilno raspoređeni podaci



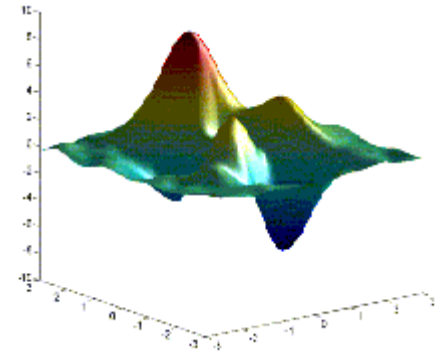
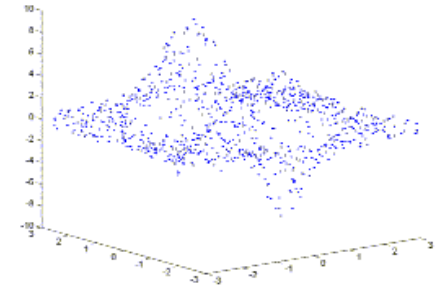
razasuti podaci

Funkcije radijalne baze RBF (engl. radial basis function)

- RBF je funkcija oblika:

$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^N \lambda_i \Phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad \mathbf{x} \in \mathbf{R}^d$$

- N broj točaka koje su zadane
- $s(x)$ nepoznata funkcija koja interpolira N točaka $s(x_i)$
- $p(x)$ polinom niskog stupnja (može biti i nula)
- λ_i težinski koeficijenti koje trebamo odrediti
- x_i središta baznih funkcija
- $\Phi(r)$ bazne funkcije (engl. basis functions) npr:
 - Gaussova funkcija $\Phi(r) = \exp(-c r^2)$, parametar c ,
 - TPS (engl. thin plate spline) $\Phi(r) = r^2 \log(r)$
 - multikvadratna $\Phi(r) = \sqrt{1 + (cr)^2}$, parametar c
 - linearna $\Phi(r) = r$
 - kubna $\Phi(r) = r^3$
 - (zvonolik oblik - radijalni oblik baznih funkcija)
- potrebno je odrediti $p(x)$ i λ_i



Primjer 1D interpolacije RBF funkcijama

- neka je $p(x) = 0$, funkcija $s(x)$ je oblika:

$$s(x) = \sum_{i=1}^N \lambda_i \Phi(|x - x_i|)$$

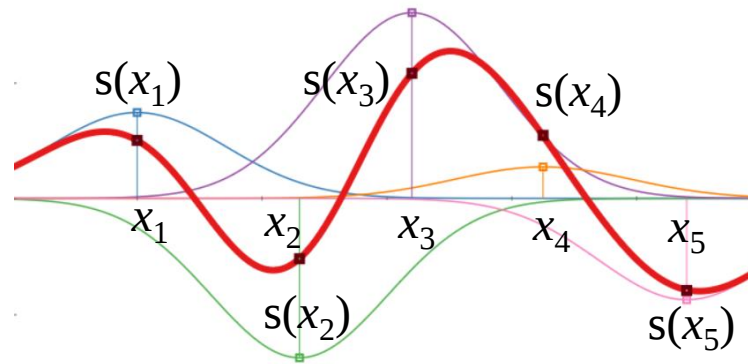
- treba riješiti linearni sustav jednačbi (LU-dekompozicija) i odrediti težine λ_i :

$$\begin{bmatrix} \Phi(0) & \Phi(|x_1 - x_2|) & \dots & \Phi(|x_1 - x_N|) \\ \Phi(|x_2 - x_1|) & \Phi(0) & \dots & \Phi(|x_2 - x_N|) \\ \dots & \dots & \dots & \dots \\ \Phi(|x_N - x_1|) & \Phi(|x_N - x_2|) & \dots & \Phi(0) \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} s(x_1) \\ s(x_2) \\ \dots \\ s(x_N) \end{bmatrix}$$

- sličan postupak je kod neuronskih mreža
 - skup za učenje $\{x_i\}$ $s(x_i)$ određuje funkciju (crvena krivulja)

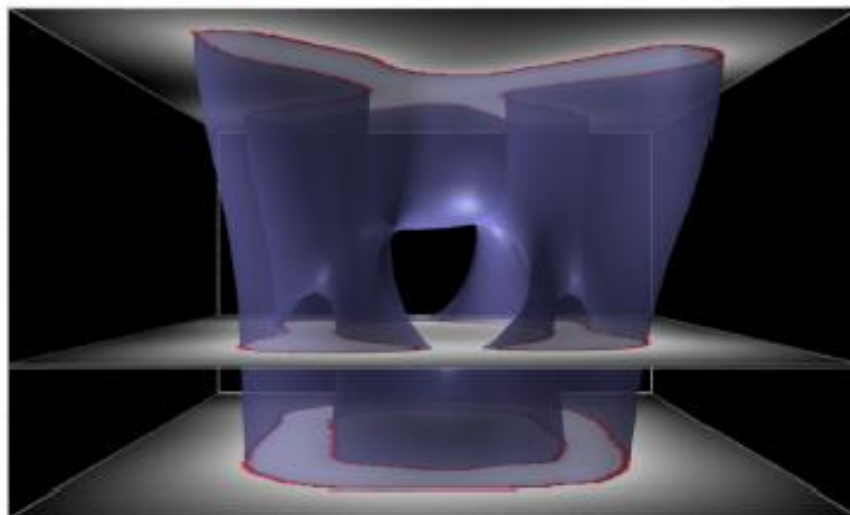
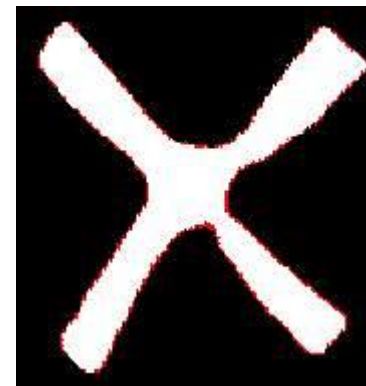
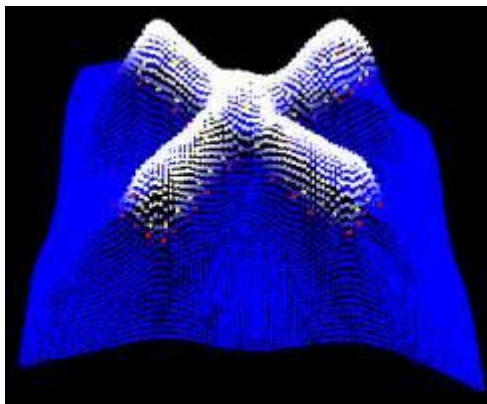
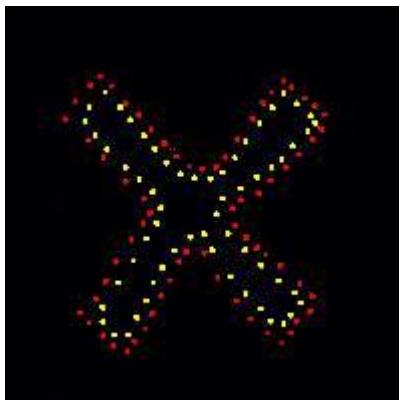
1D <http://multivis.net/lecture/radial-basis.html>

3D <https://www.clicktorelease.com/code/bumpy-metaballs/>



Primjer definiranja objekta u 2D

- žute točke su unutar objekta, a crvene izvan
- http://mathinsight.org/level_sets
- <http://www.zemris.fer.hr/predmeti/rg/diplomski/03Baronica/rezultati.htm>



Primjer 3D (4D) interpolacije RBF funkcijama

- zadana je površina i RBF

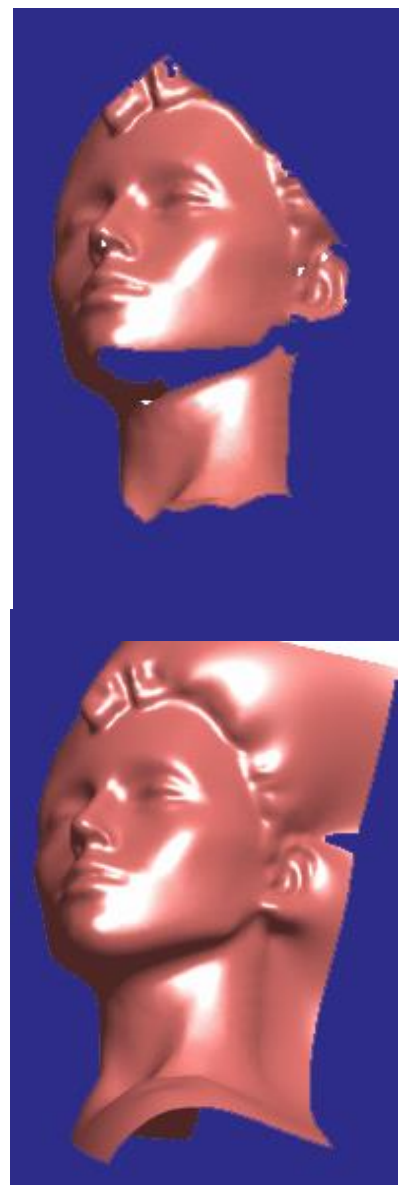
$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{i=1}^N \lambda_i \Phi(|\mathbf{x} - \mathbf{x}_i|), \quad s(\mathbf{x}) = f_i, \quad \Phi(r) = r$$

- sustav je pod-determiniran, pa imamo dodatne uvjete

$$\sum_{i=1}^N \lambda_i = \sum_{i=1}^N \lambda_i x_i = \sum_{i=1}^N \lambda_i y_i = \sum_{i=1}^N \lambda_i z_i = 0$$

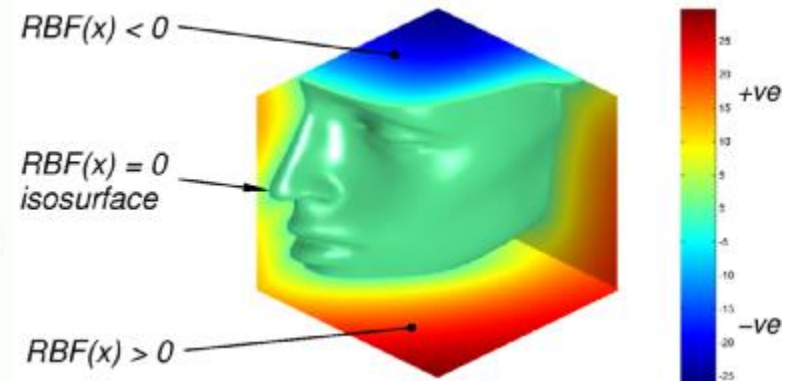
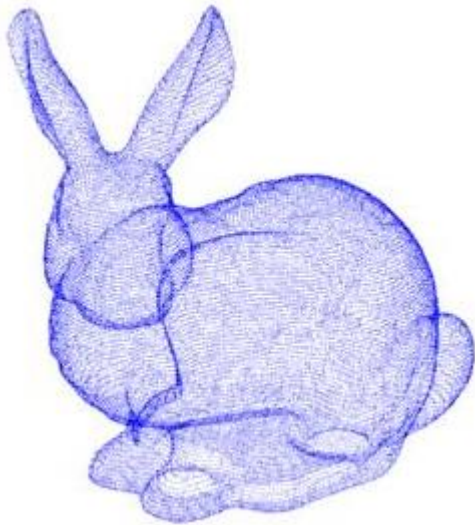
$$\begin{bmatrix} |x_1 - x_1| & \dots & |x_1 - x_N| & x_1 & y_1 & z_1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ |x_N - x_1| & \dots & |x_N - x_N| & x_N & y_N & z_N & 1 \\ x_1 & \dots & x_N & 0 & 0 & 0 & 0 \\ y_1 & \dots & y_N & 0 & 0 & 0 & 0 \\ z_1 & \dots & z_N & 0 & 0 & 0 & 0 \\ 1 & \dots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_N \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ \dots \\ f_N \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- popunjavanje “rupa”



Primjer 3D interpolacije RBF funkcijama

- interpolacija raspršenog skupa točaka (uzorkovanih)
- problem parametrizacije, nije definiran redoslijed (susjedstvo) kao ni u 1D primjeru
- problem jako velikog sustava jednačbi za rješavanje, no matica je slabo popunjena zbog lokalnog djelovanja težinskih funkcija
- rezultat je implicitno definirana RBF površina
- možemo uključiti vrijeme kao još jednu dimenziju

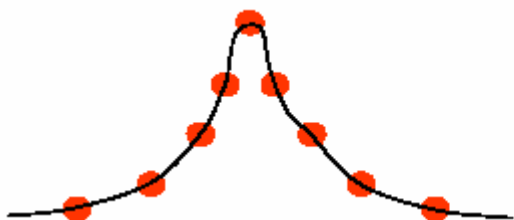


Usmjeravanje čestica:

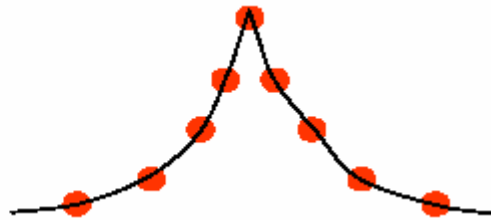
<http://www.iamnop.com/particles/>

Tipični problemi:

- kontradiktoran zahtjev kod interpolacije između:
 - filtriranja šuma
 - dobre lokalne aproksimacije



filtrira šum



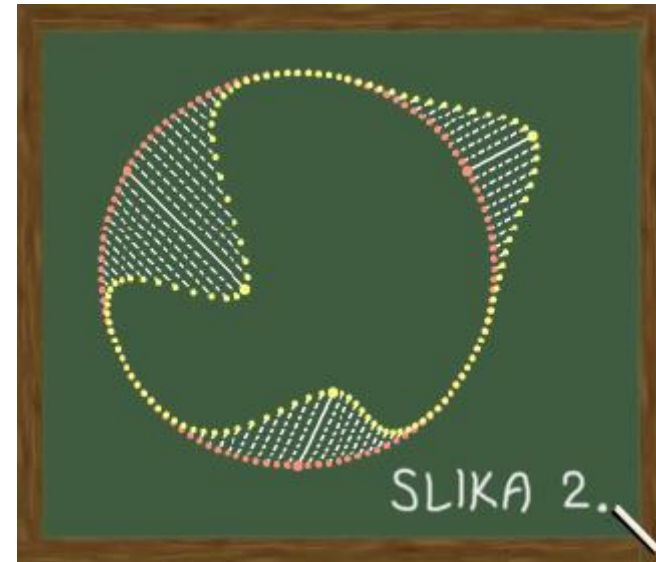
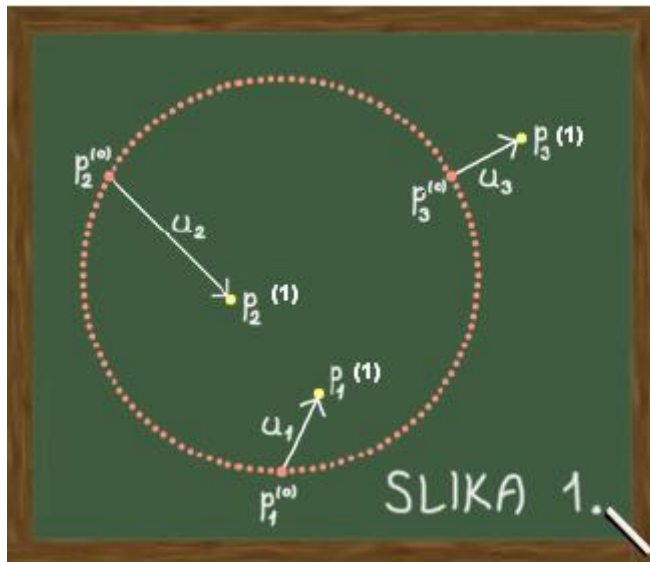
potrebno lokalno
aproksimirati



ne filtrira šum

potrebno
filtrirati šum

- početni skup točaka $P = \{p_i\}$,
- skup za učenje je podskup početnog skupa točaka $P^{(0)} = \{p^{(0)}_i \mid p^{(0)}_i \in P\}$,
- rješavanje sustava RBF i određivanje koeficijenata λ_i , za točke $p^{(0)}_i$
- skup vektora pomaka $U = \{u_i\}$, svakoj točki skupa za učenje pridružen je vektor pomaka
- skup pomaknutih točaka $P^{(1)} = \{p^{(1)}_i\}$,
- pomak ostalih točaka početnog skupa određen je sustavom RBF i izračunatim koeficijenata λ_i
- [Morph](#) [Warp](#)



Lomljenje objekata:

[bunnies](#) [Theend](#)

