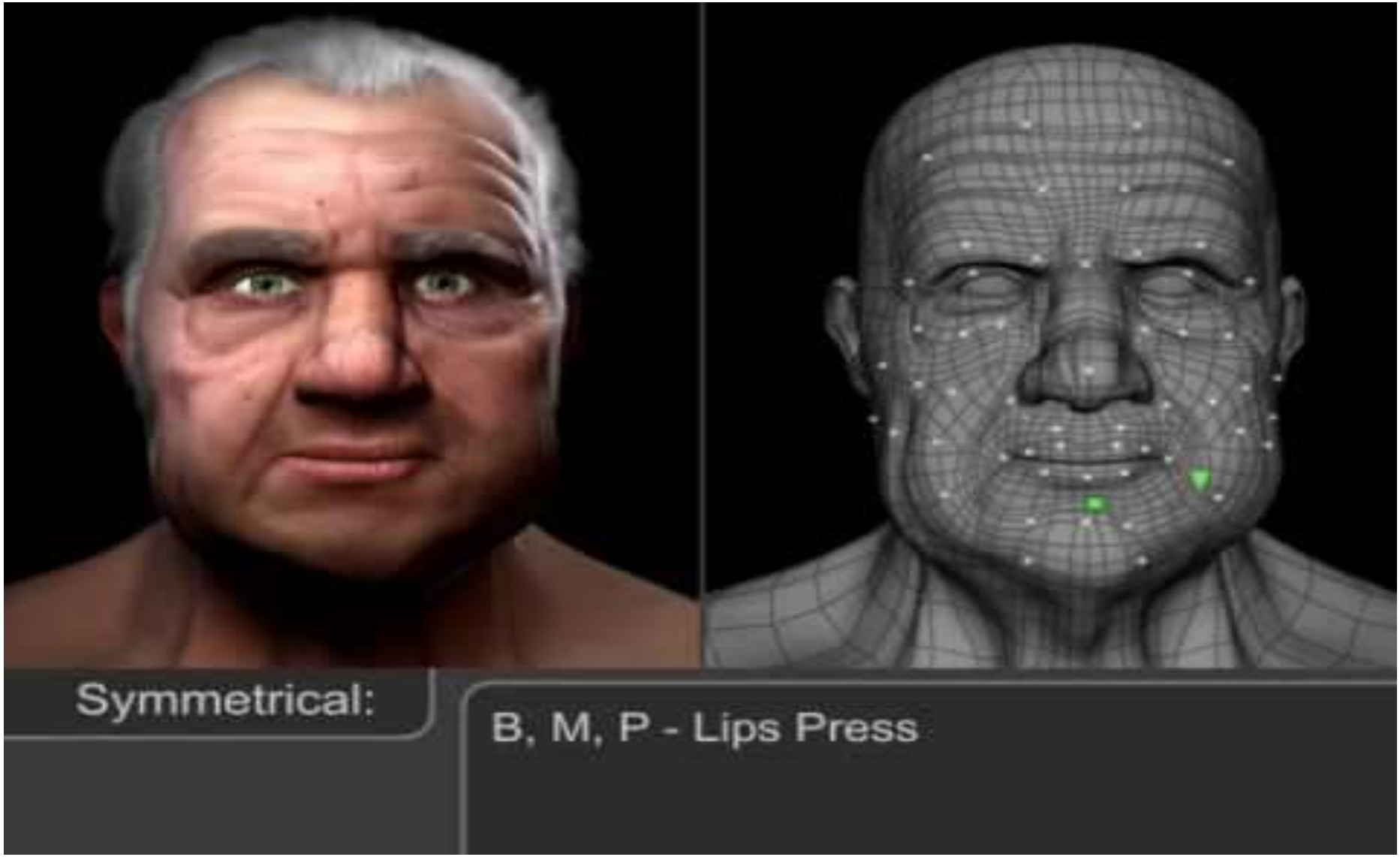
Deformacijos

T120B167 Žaidimų grafinių specialiųjų efektų kūrimas ir programavimas


Rytis Maskeliūnas
Skype: rytmask
Rytis.Maskeliunas@ktu.lt

Paskaitos tema

MULTIMEDIJOS INŽINERIJOS KATEDRA



https://www.youtube.com/watch?v=p06yweBU4fo

Problema

- Nemažai objektų yra nepastovios formos:
  - Želė ☺
  - purvas
  - dujos/skysčiai
  - etc.
- Ką daryti?:
  - Deformuoti geometriškai
  - Keisti objektą fiziškai (plaktuku) ;)

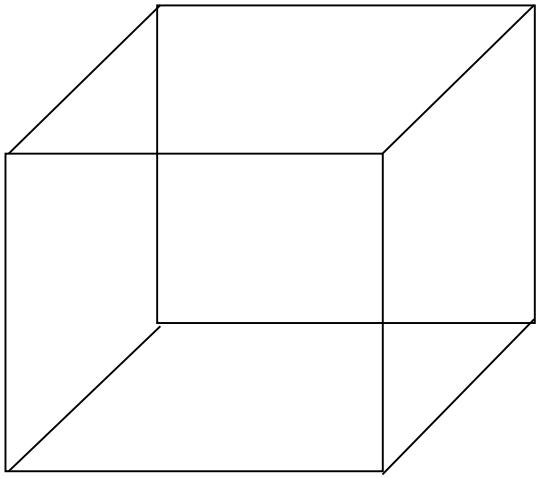
Objektų deformacija

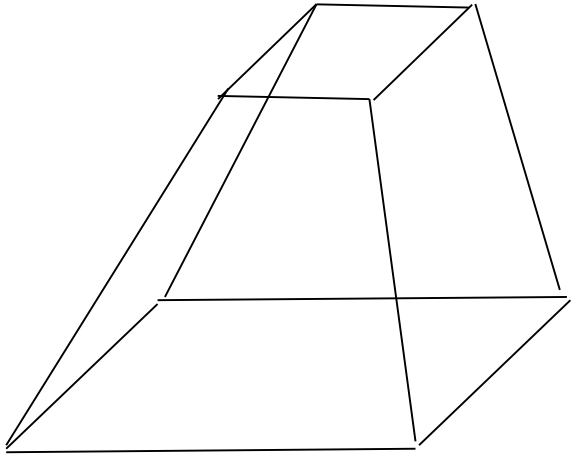
- Ką galime daryti su įprastais geometriniais objektais?
- Gal perdaryti (morph) vieną objektą kitokiu?
  - Paprasta, jei šaltinio objektas ir tas kurį norime gauti turės tokį pat viršūnių skaičių, o taškai bus taip pats sujungti
- Paprastoji formos modifikacija
  - Netolydinė skalė (Non-uniform scale)
  - „Karpymas“ (Shearing)
  - Susijusios erdvės transformacijos

    (General affine transformation)
  - Kraipymai (Warping)

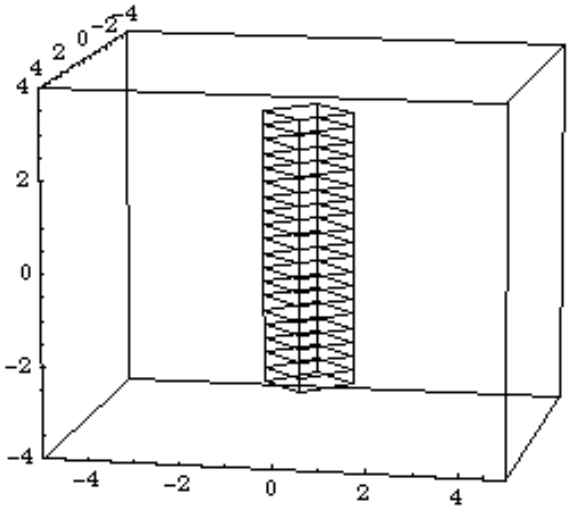
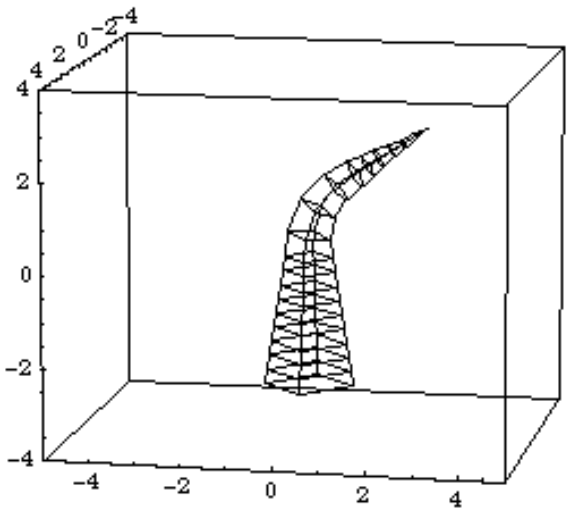
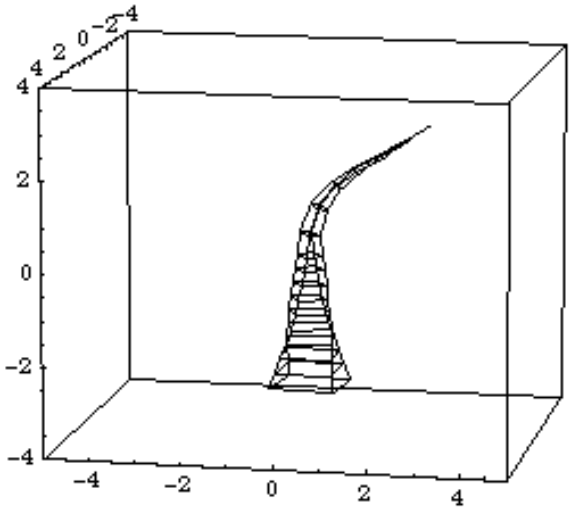
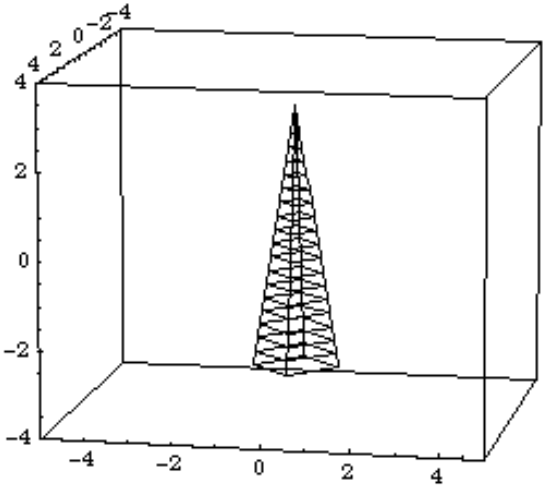
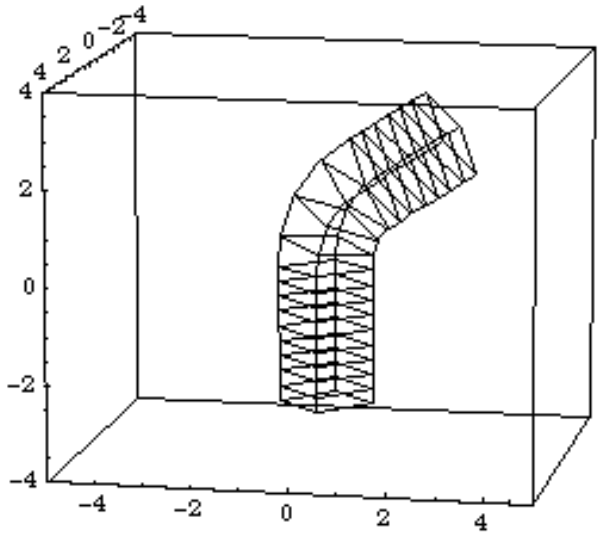
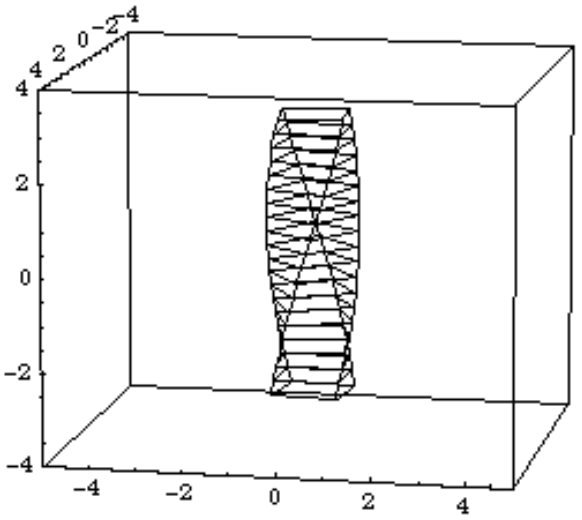
Formos

- Žemėlapiai ir jungimai
  - Nuo viršūnėlės iki viršūnėlės (vertex to vertex)
  - Nuo kraštinės iki kraštinės (Edge to edge)



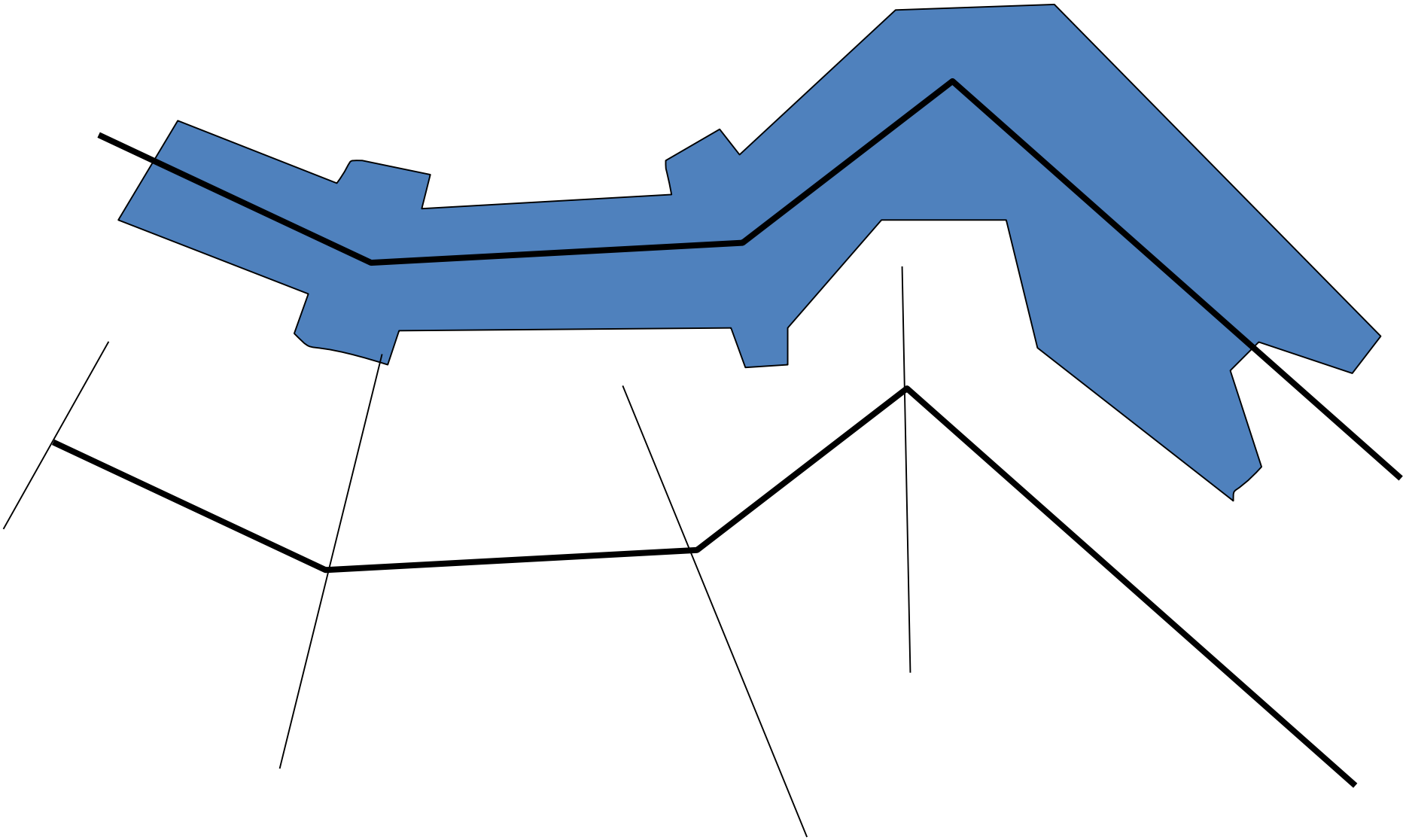


Deformacijos: keletas jų

Skeletinė deformacija

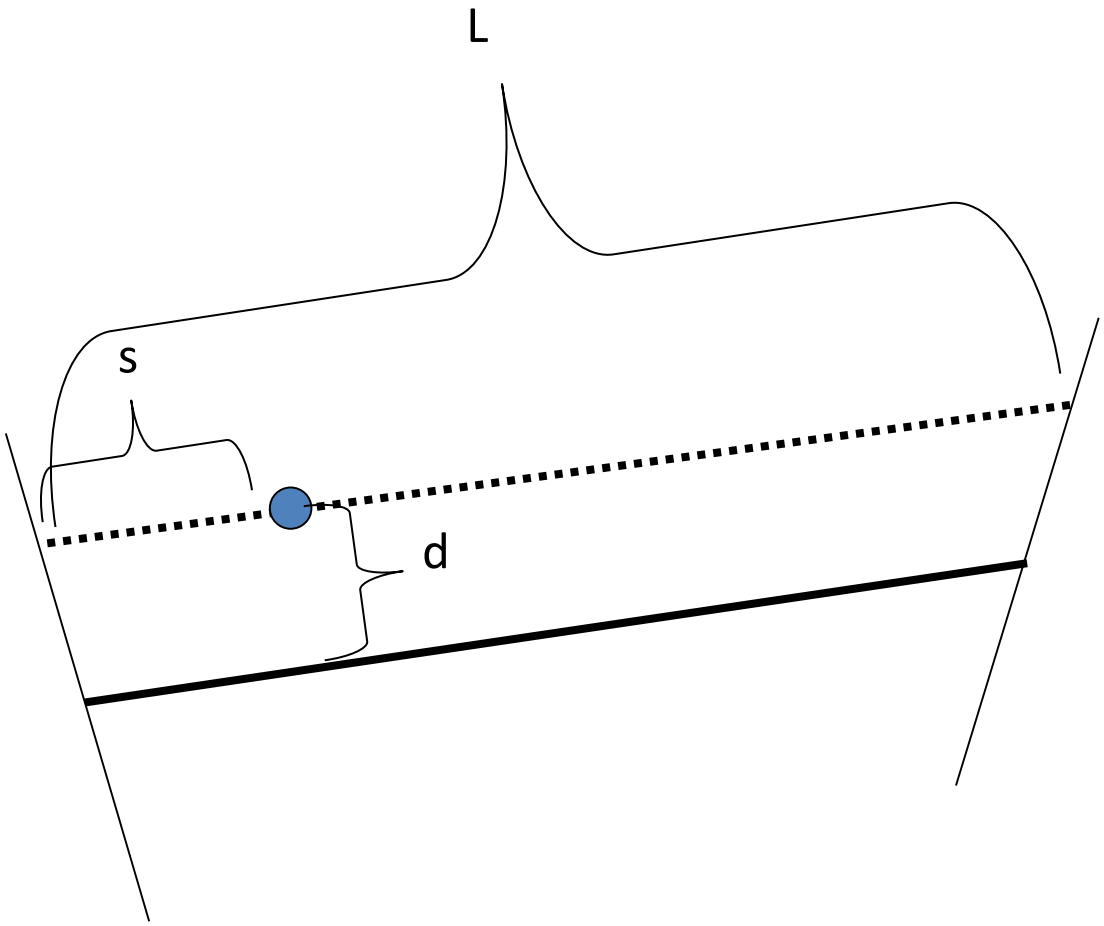
MULTIMEDIJOS INŽINERIJOS KATEDRA



Skeletinė deformacija

https://www.youtube.com/watch?v=lMZb0AatpYE

Skeletinė deformacija



ALGORITMAS:

Paimam objektą
Nusipaišom „nematomas“ linijas
Pririšam viršunėles prie tos linijos
Kraipome liniją kaip mums reikia
Keičiame viršūnėlių poziciją pagal liniją

Geometrinės deformacijos

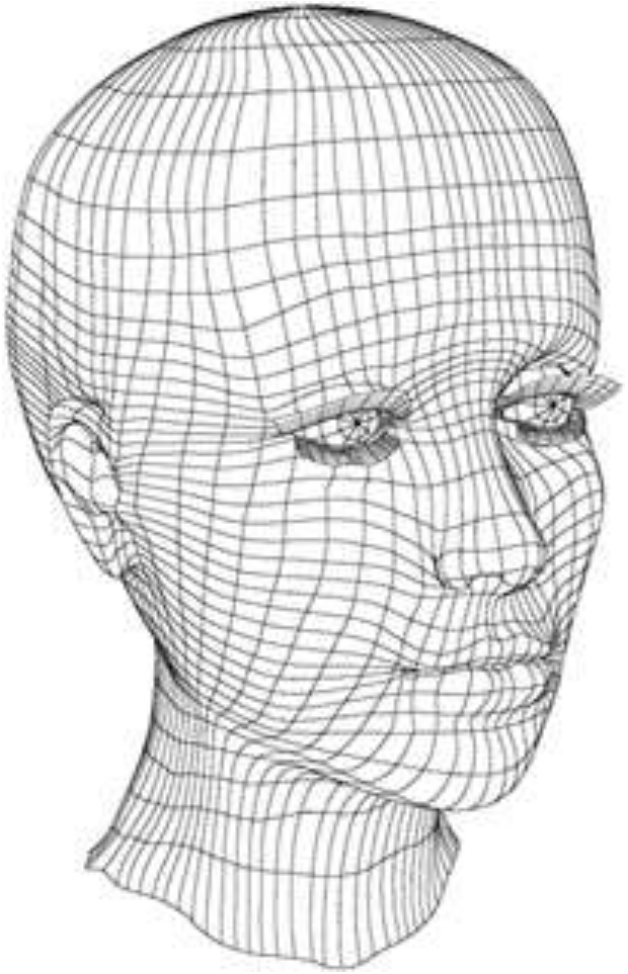
MULTIMEDIJOS INŽINERIJOS KATEDRA

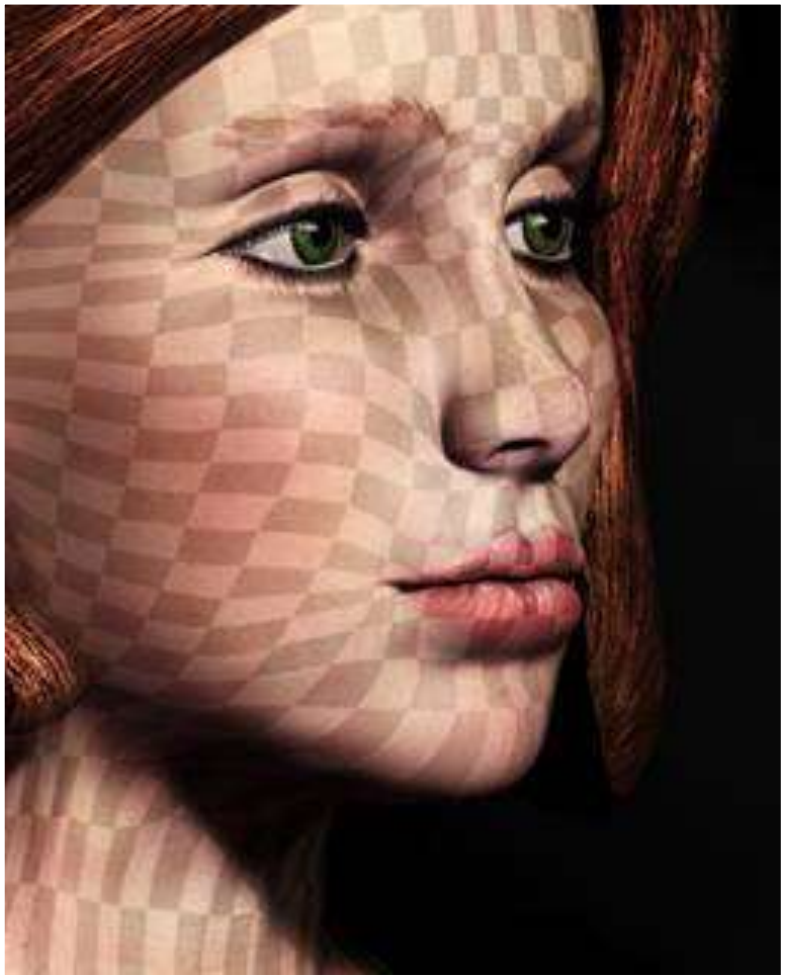
- Galima tiesiogiai deformuoti objekto geometriją
- Du pagrindiniai metodai:
  - kontrolinio taško ar viršūnėlės modifikacija

    (control point / vertex manipulation)
  - erdvės kraipymas

    (space warping)

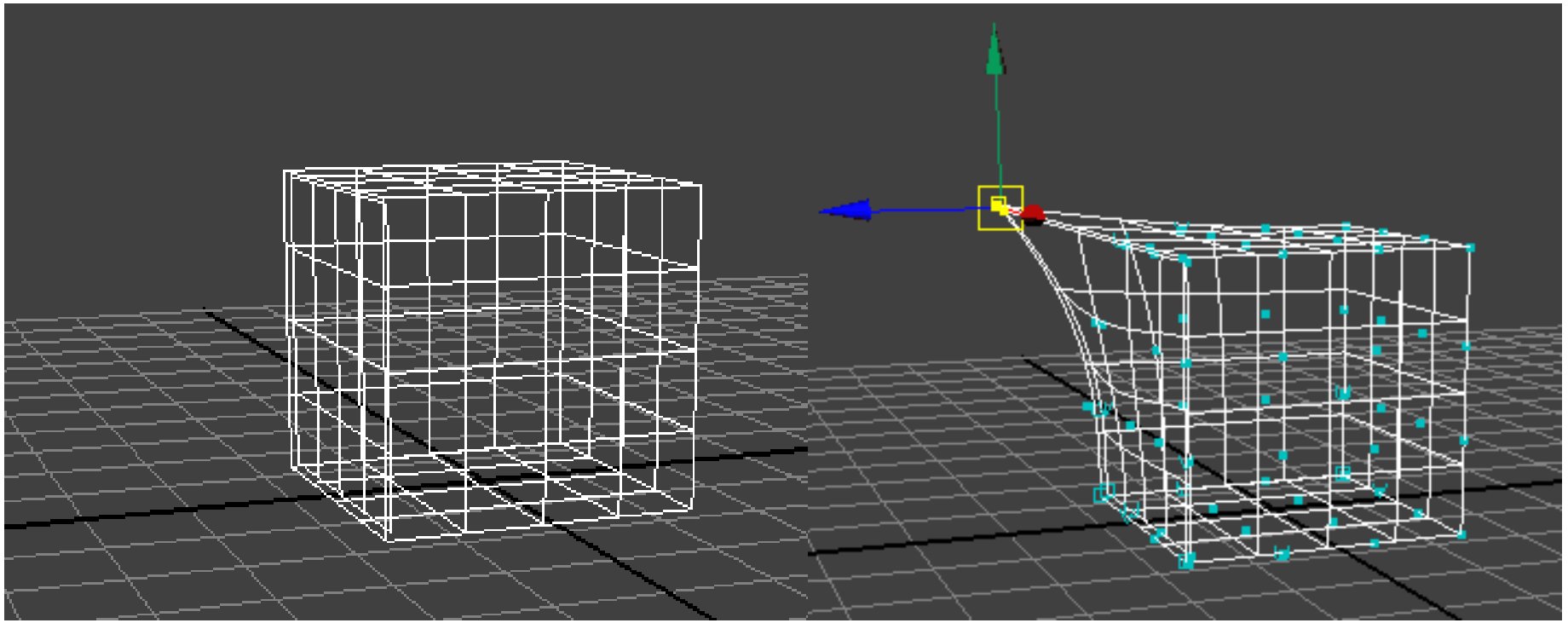
Kontrolinio taško/viršūnėles deformacijos





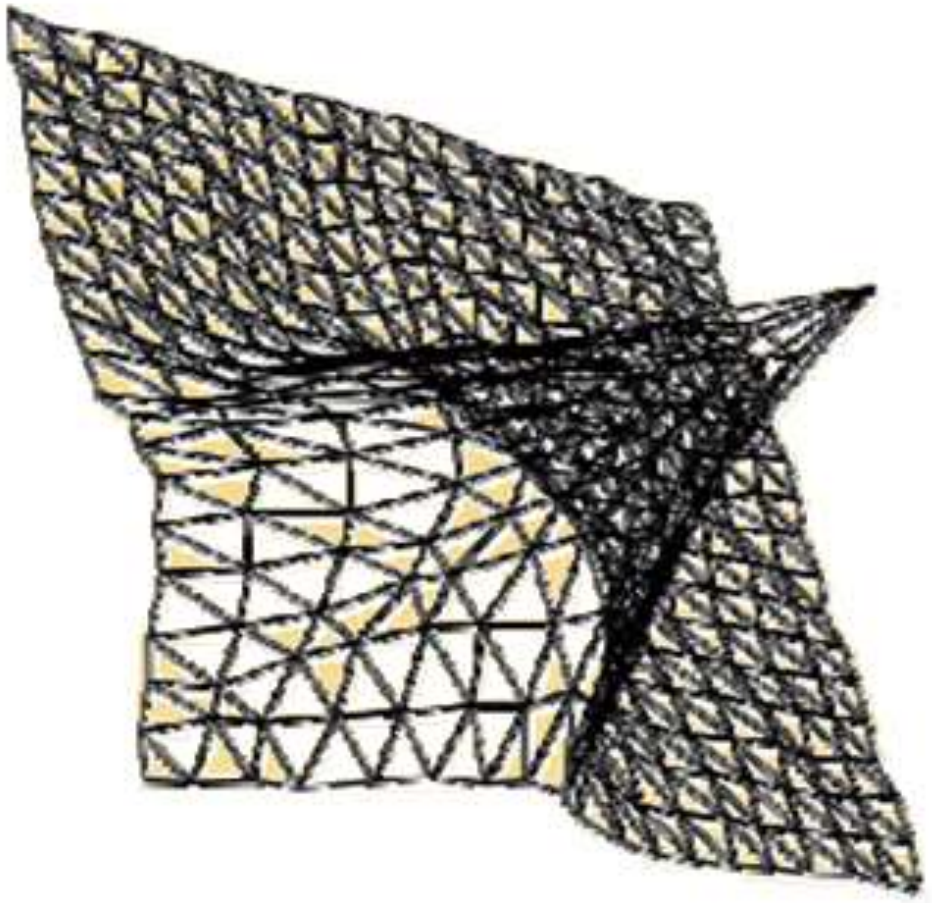
Tiesiogiai redaguojamos viršūnėlės ar paviršiaus taškai

Paviršiaus taškų modifikavimas



Paviršiaus taškų modifikavimas

Tašką *p* ištempiant (pakeičiant) į *W(p)*

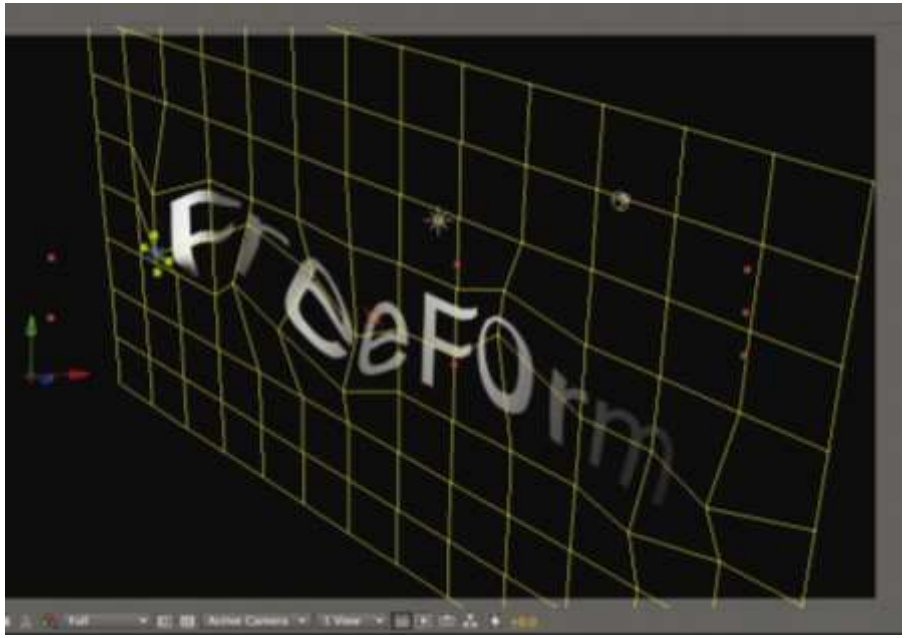


Erdvės kraipymas

Deformuoti objektą deformuojant erdvę kurioje jis yra

Du būdai:

- Netiesinė deformacija (Nonlinear Deformation)
- Laisvos formos deformacija (Free Form Deformation (FFD))

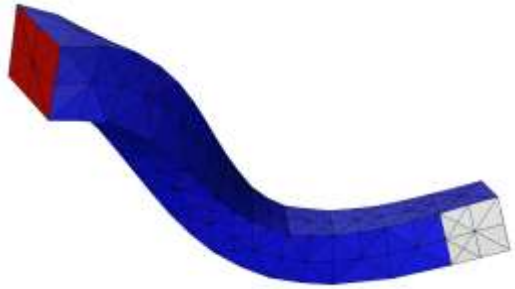


Netiesinė globalinė deformacija

Objektai yra apibrėžti lokalioje objektų erdvėje

Šią erdvę galima deformuoti kombinuojant:

- netolydini masteliavimą (Non-uniform Scaling)
- nusmailinimą (Tapering)
- susukimą (Twisting)
- sulenkimą (Bending)

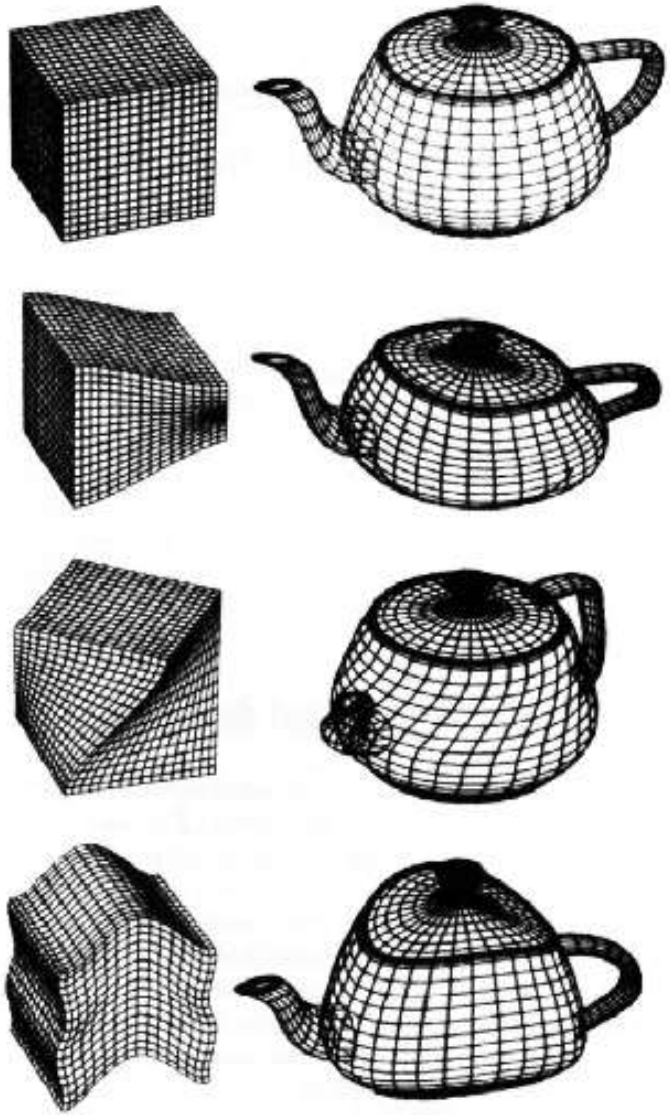


Šiuos apjungus galima gauti sudėtingas figūras ir formas

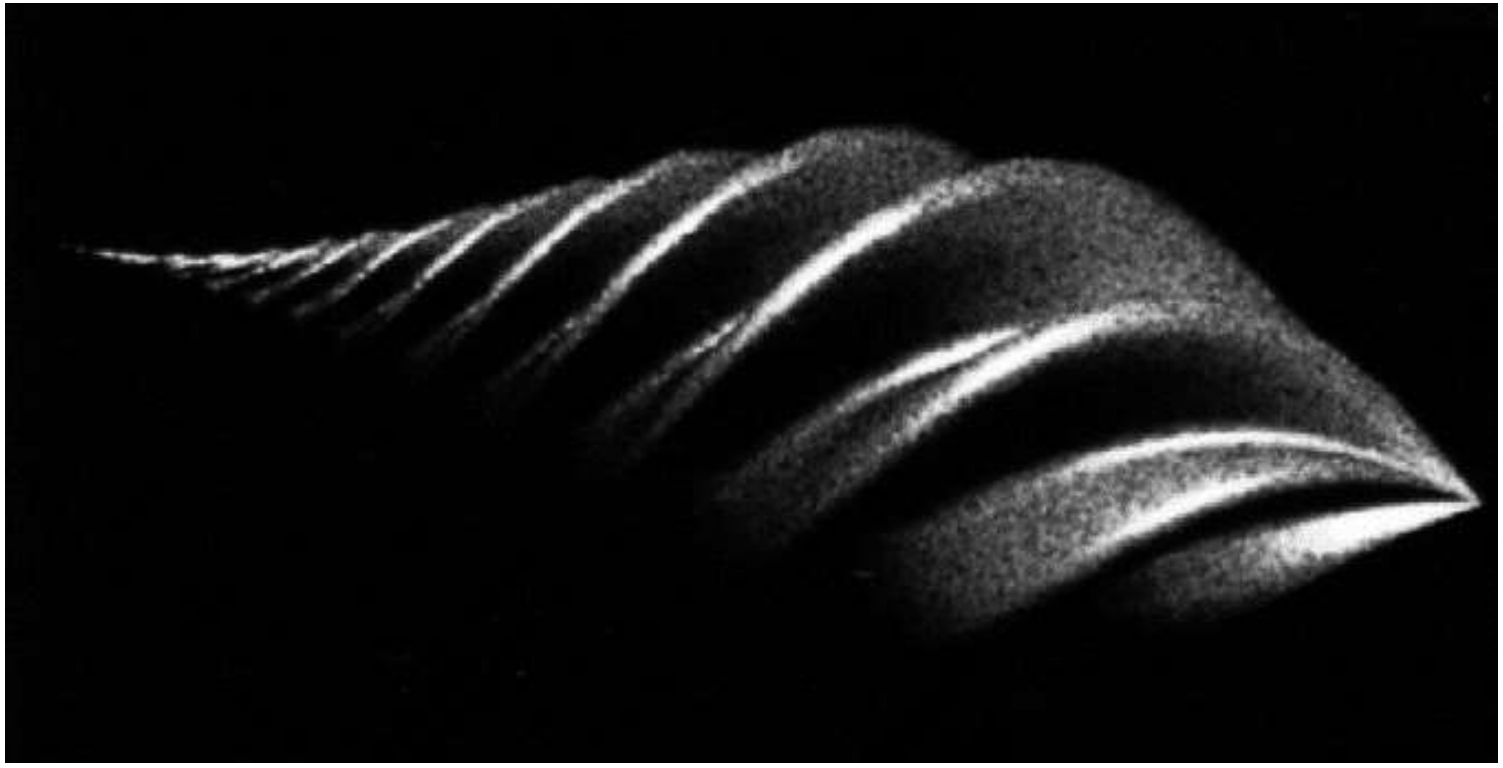
Ir įdomius rezultatus ☺

Netiesinė globalinė deformacija

- original
- tapering
- twisting
- bending



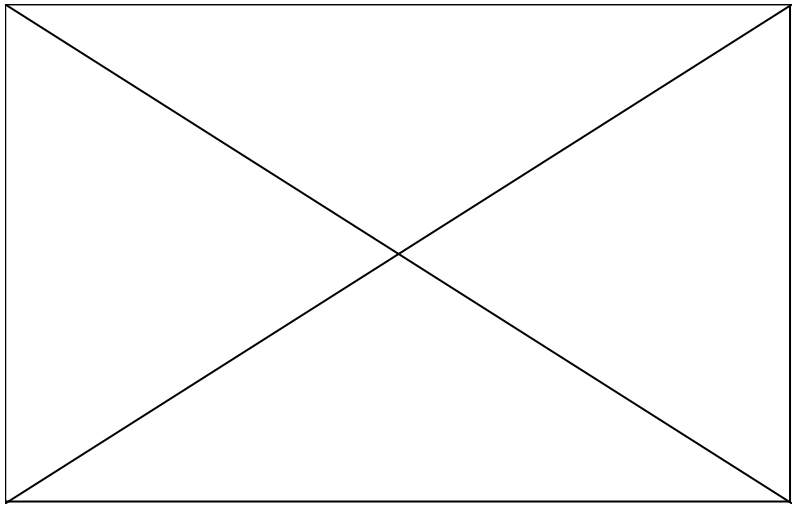
Netiesinė globalinė deformacija



Sunkiau animuoti

Animacija stumdant paviršiaus elementus

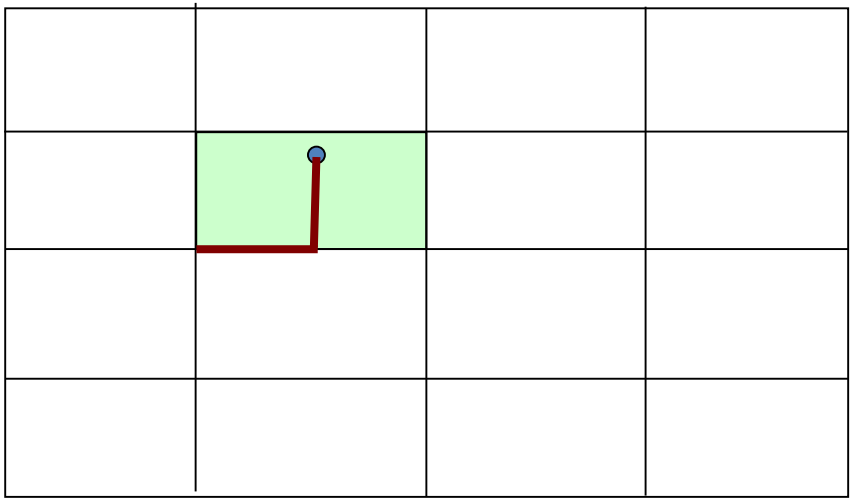
- Sukuriamos per viršūnėles valdomos animacijos kreivės
  - Žinoma pradžia ir galas
- Tiesiog keičiami parametrai laike
  - Keičiamas kampas (Twist angle)
  - Keičiamos konstantos (Scaling constant)
  - Redaguojama verteksų pozicija (Seed vertex position)
- Galioja standartinės 3d taisyklės
  - Kreivių tolygumas, judesio valdymas ir kt.



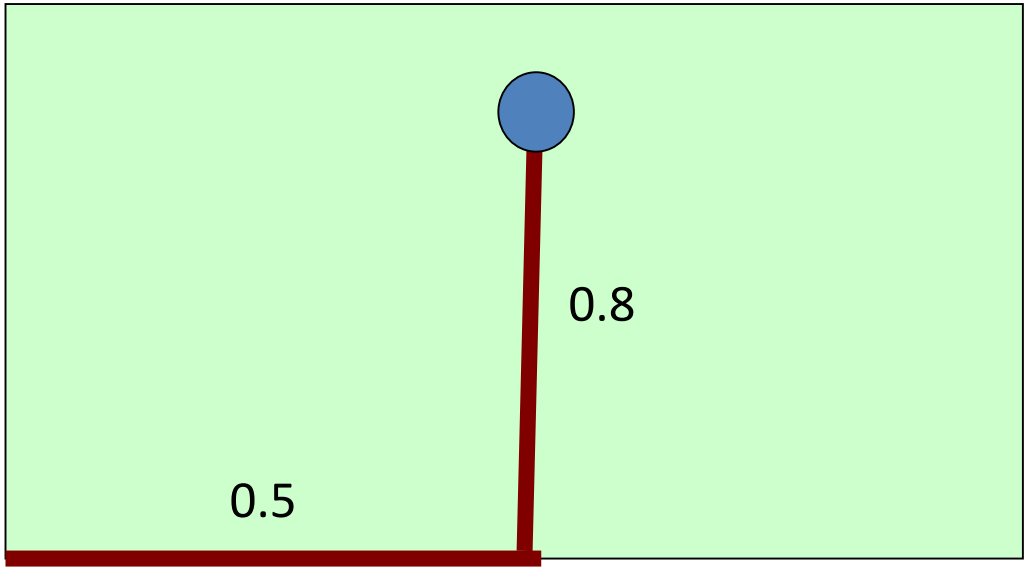
Koordinačių grido deformacija

- Nustatoma lokali koordinačių sistema
- Kažkokiu būdu modifikuojama
- Objektas išlaiko originalias koordinates modifikuotoje erdvėje
- Pvz: 2D grid‘o deformacija
  - Imam kvadratinį 2D gridą ir užnešam ant objekto
  - Pririšam objekto viršūnėles prie grido langelių (sukuriam lokalią koordinačių sistemą)
  - Kraipom gridą (2D viršūnėles)
  - Objekto viršūnėles yra pririšamos prie lokalios 2D grido koordinačių sistemos naudojant bilinear interpoliaciją

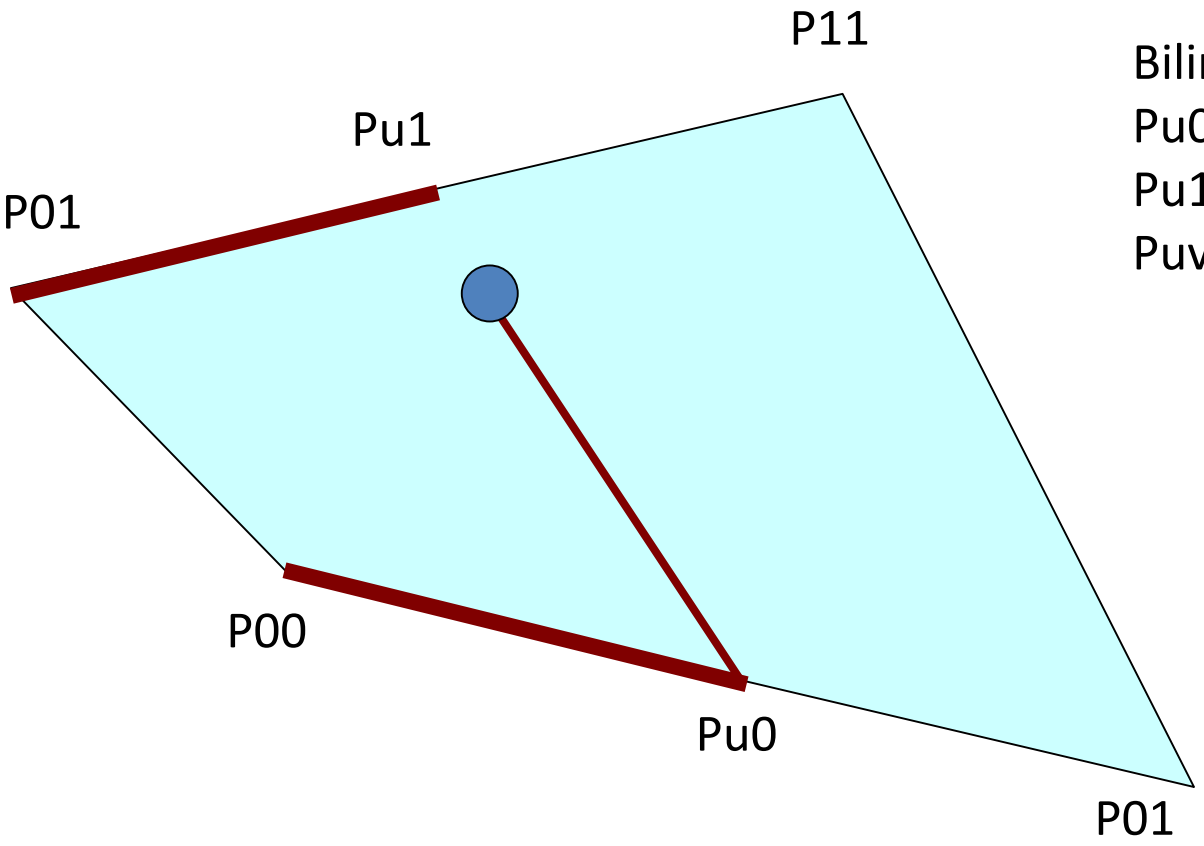
Koordinačių grido deformacija



Kiekvienai viršūnėlei reikia identifikuoti langelį ir lokalias u,v koordinates



Koordinačių grido deformacija



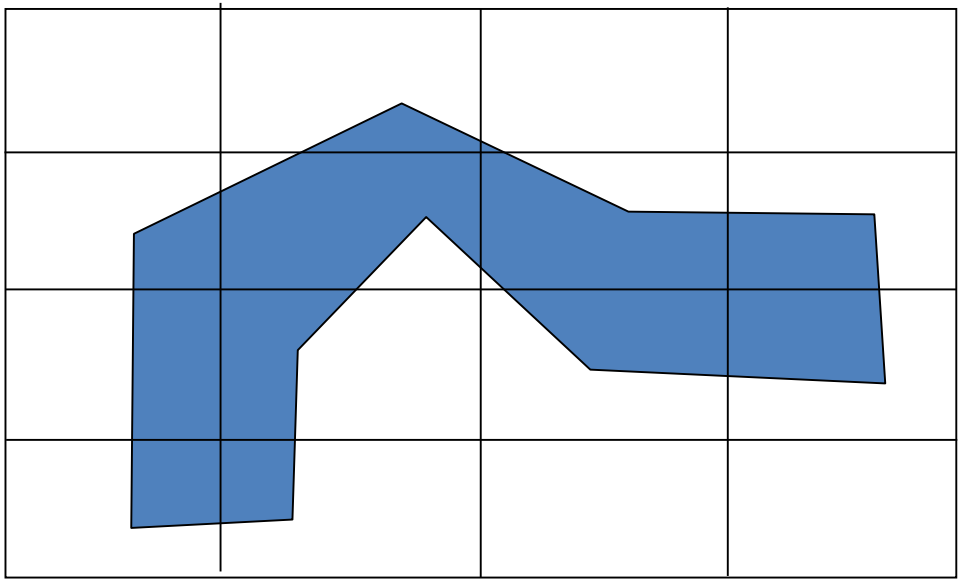
Bilinear interpoliacija

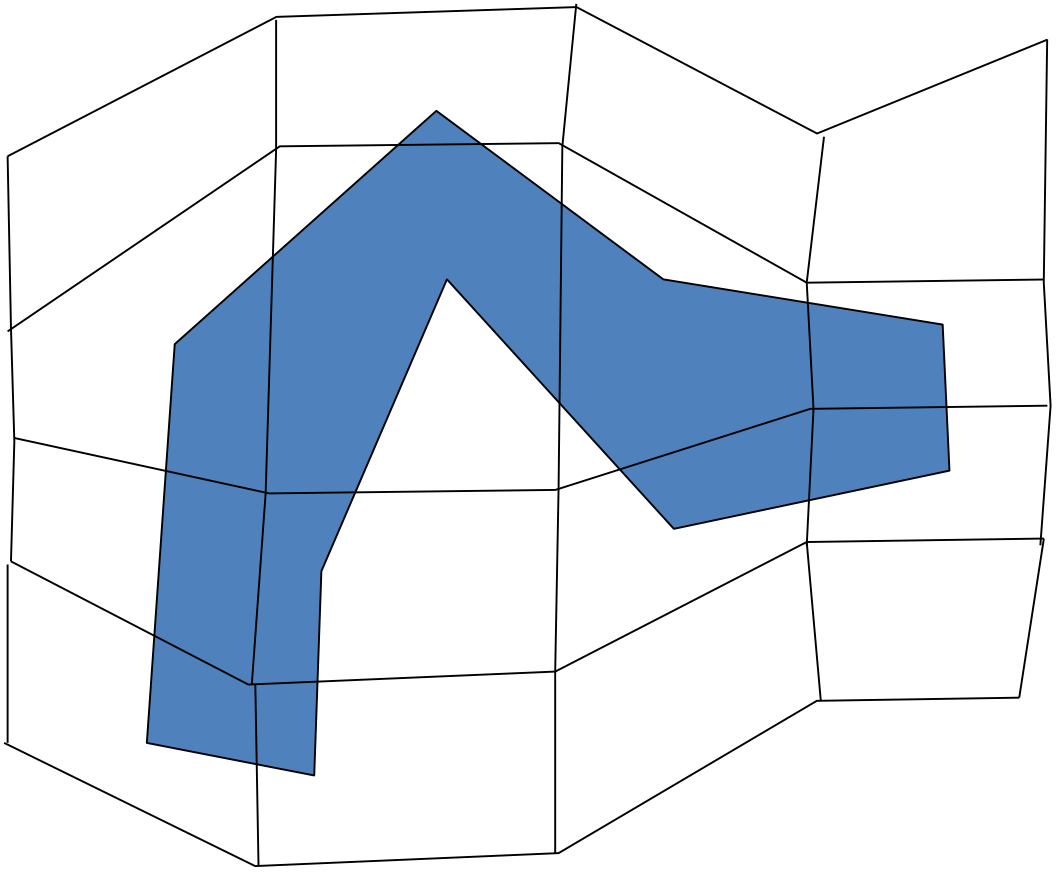
Pu0 = (1-u)*P00 + u*P10

Pu1 = (1-u)*P01 + u*P11

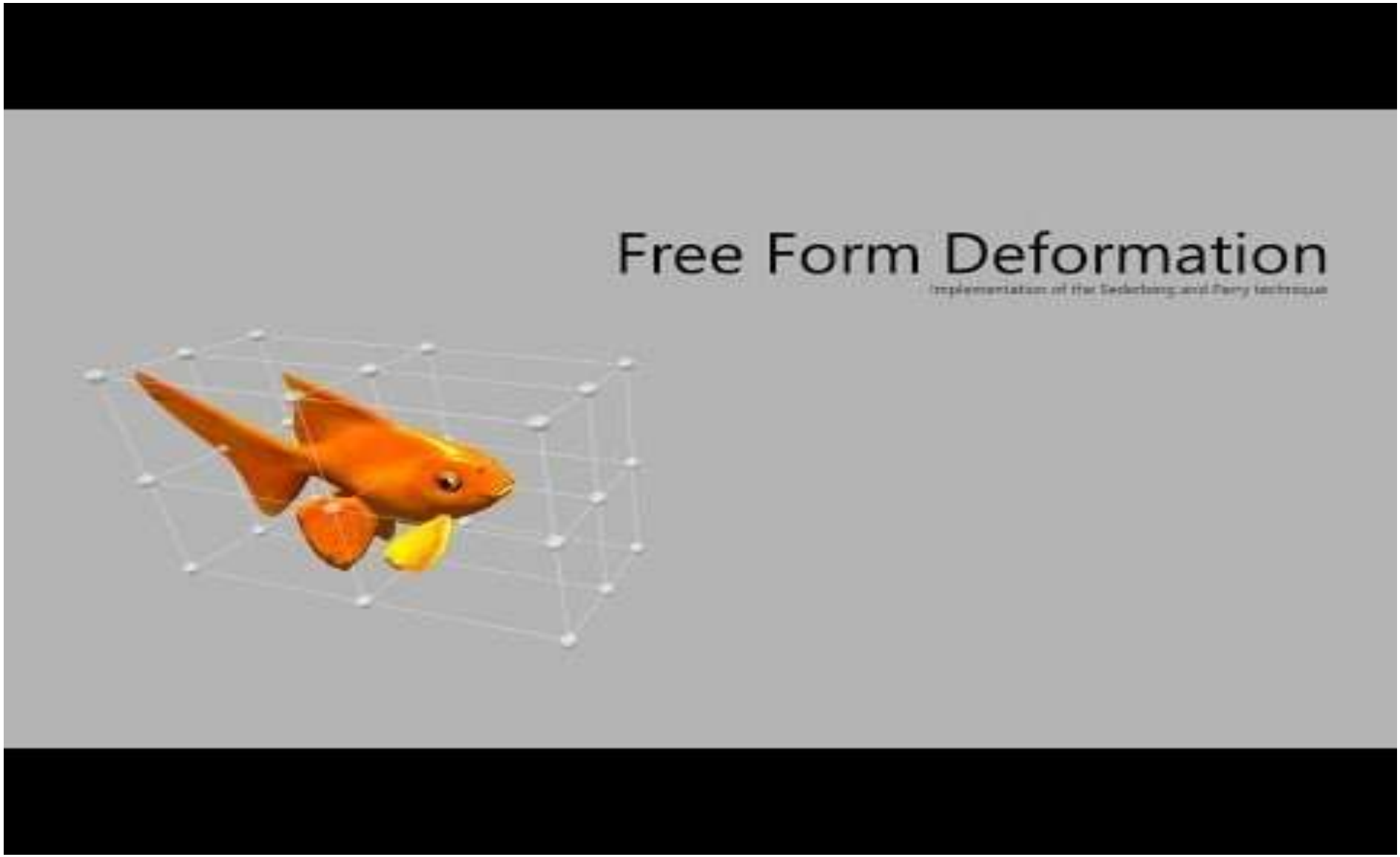
Puv = (1-v)*P0u + v*P1u

Koordinačių grido deformacija





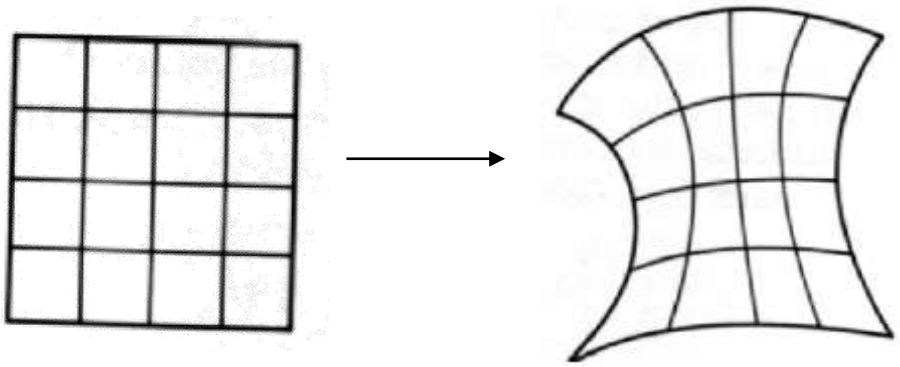
Laisvos formos deformacijos



https://www.youtube.com/watch?v=kkPUU_VXTAc

Laisvos formos deformacija
Free Form Deformation (FFD)

Objektą laisvai galima deformuoti deformuojant jo elementus



Deformacija yra apibrėžta stumdant taškus

Pvz. Guminis objektas

Kas yra laisva forma (free form)?

- **Parametriniai paviršiai yra laisvos formos paviršiai.**
- Ši deformacija yra labai lanksti nes leidžia „laisva valia“ deformuoti modelį.
  - ✓ Galima tampyti bet kokį paviršiaus lopą
  - ✓ Atlikti globalinę ar lokalią deformaciją
  - ✓ Išlaikyti deformacijos tolydumą
  - ✓ Išlaikyti tūrį
  - ✓ Ir kt.

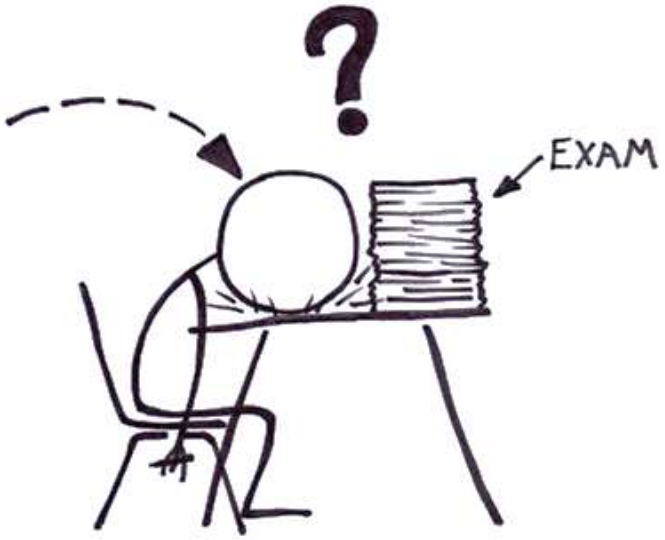
FFD formulės

- **Taškas STU koordinačių sistemoje:**

$$\mathbf{X} = \mathbf{X}_0 + s\mathbf{S} + t\mathbf{T} + u\mathbf{U}.$$

$$s = \frac{\mathbf{T}\times\mathbf{U}\,(\mathbf{X} - \mathbf{X}_0)}{\mathbf{T}\times\mathbf{U}\cdot\mathbf{S}},\quad t = \frac{\mathbf{S}\times\mathbf{U}\cdot(\mathbf{X} - \mathbf{X}_0)}{\mathbf{S}\times\mathbf{U}\cdot\mathbf{T}},\quad u = \frac{\mathbf{S}\times\mathbf{T}\cdot(\mathbf{X} - \mathbf{X}_0)}{\mathbf{S}\times\mathbf{T}\cdot\mathbf{U}}$$

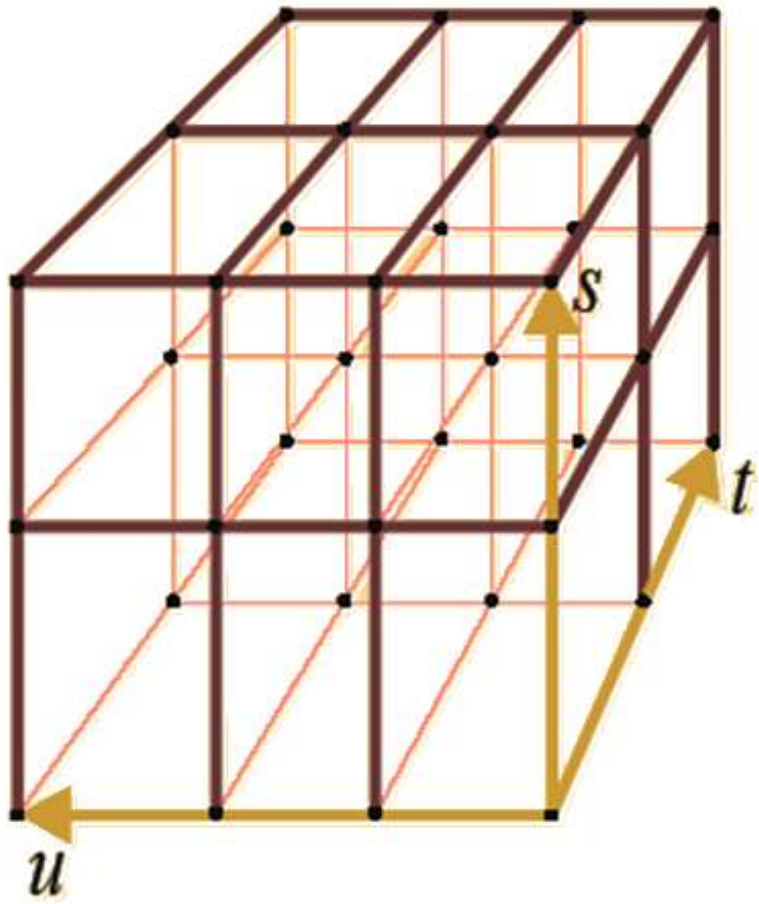
$$0 < s < 1,\ 0 < t < 1 \text{ and } 0 < u < 1.$$

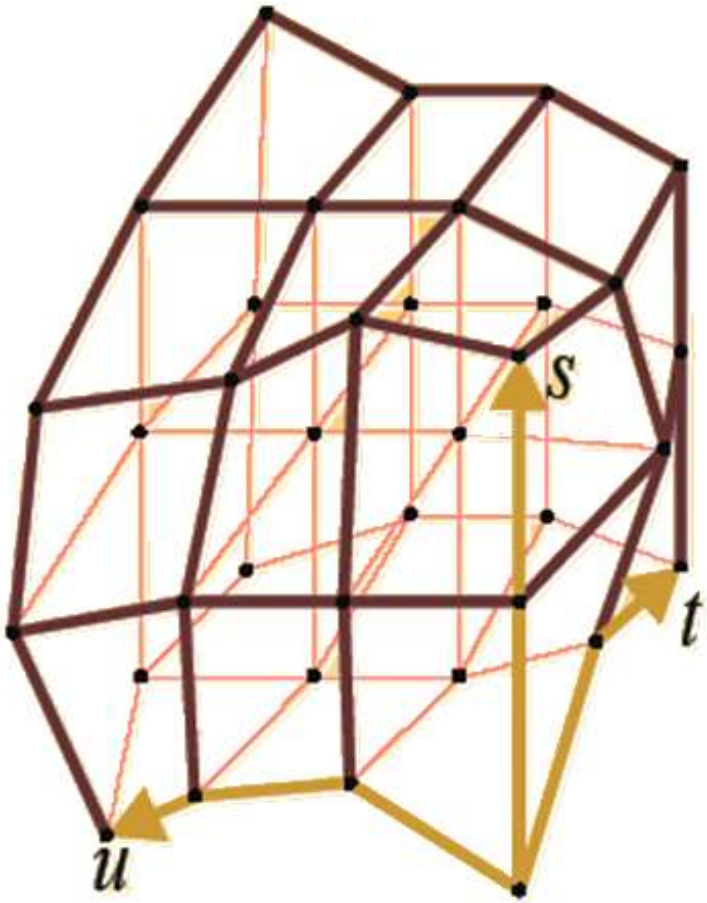


- **Deformuotas taškas:**

$$\mathbf{X}_{ffd} = \sum_{i=0}^{l}\binom{l}{i}(1-s)^{l-i}s^{i}\left[\sum_{j=0}^{m}\binom{m}{j}(1-t)^{m-j}t^{j}\left[\sum_{k=0}^{n}\binom{n}{k}(1-u)^{n-k}u^{k}\mathbf{P}_{i,j,k}\right]\right]$$

Laisvos formos deformacija

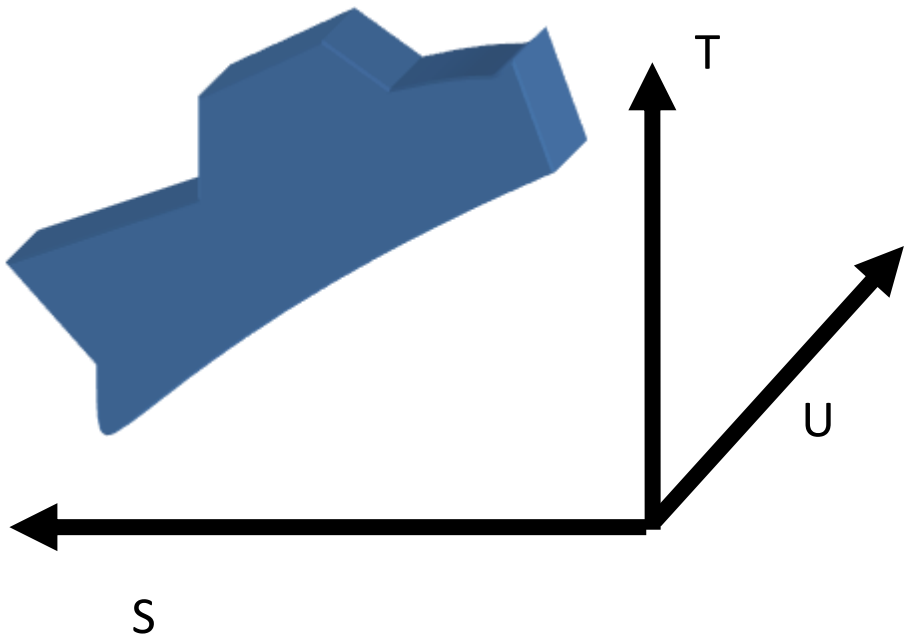




Laisvos formos deformacija

MULTIMEDIJOS INŽINERIJOS KATEDRA

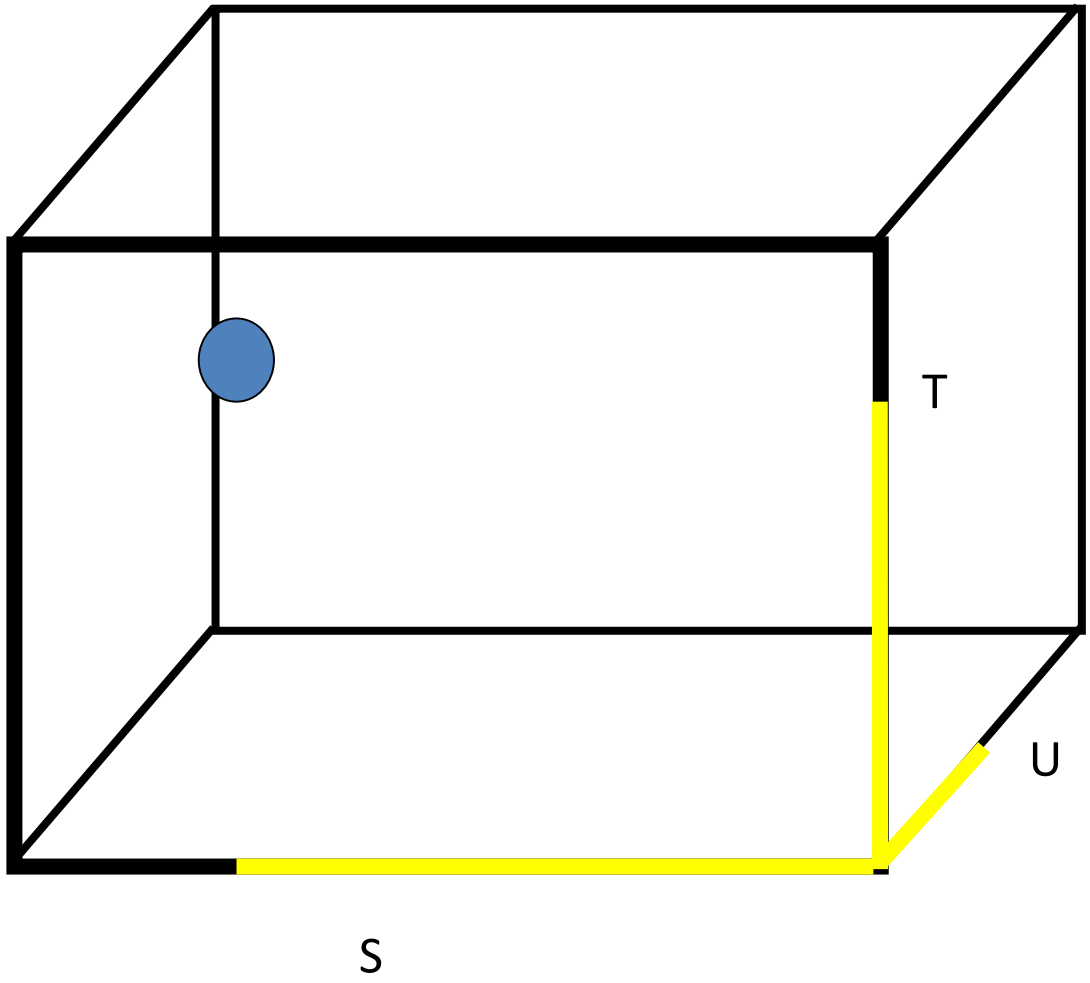
- Apsibrėžiam lokalinę koordinačių sistemą: (S,T,U)



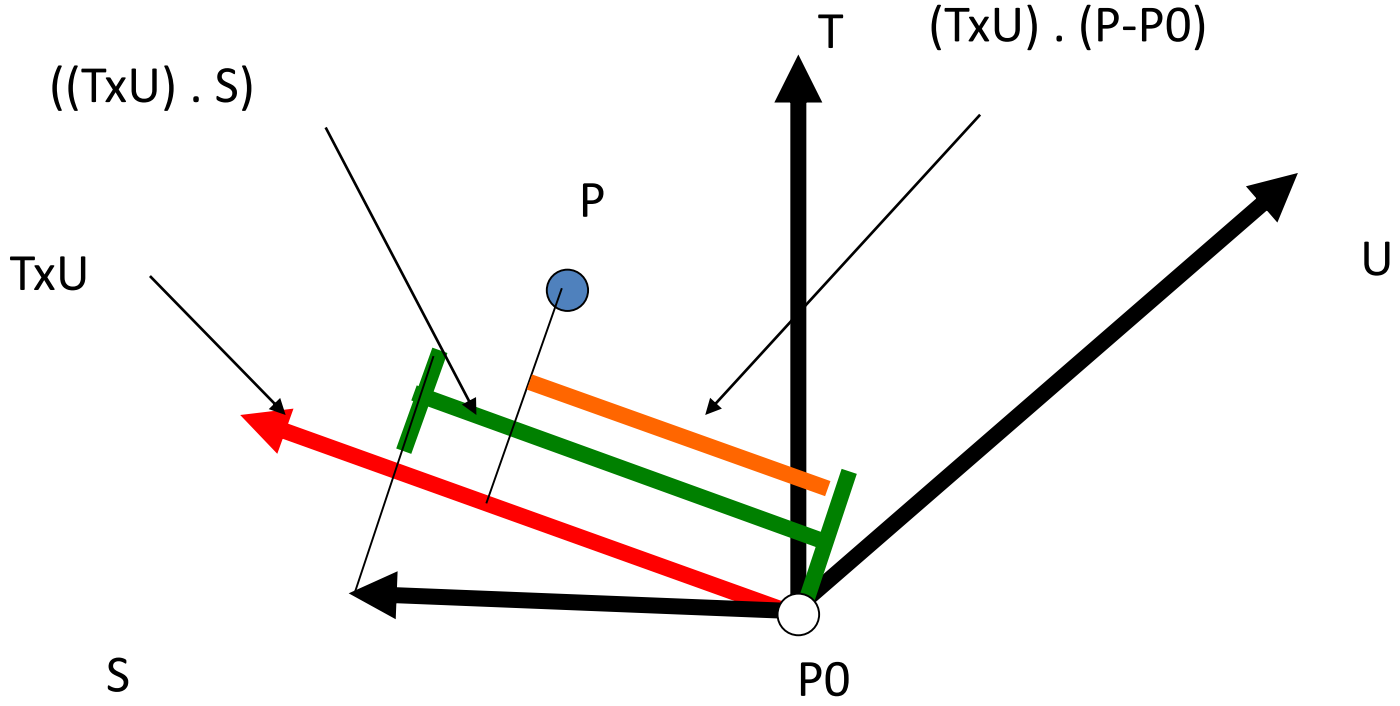
Laisvos formos deformacija

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Užfiksuojam tašką langelyje



Laisvos formos deformacija



Pakreipiam P koordinatę pagal S:

$s = (TxU) . (P-P0) / ((TxU) . S)$

$P = P0 + sS + tT + uU$

Laisvos formos deformacija

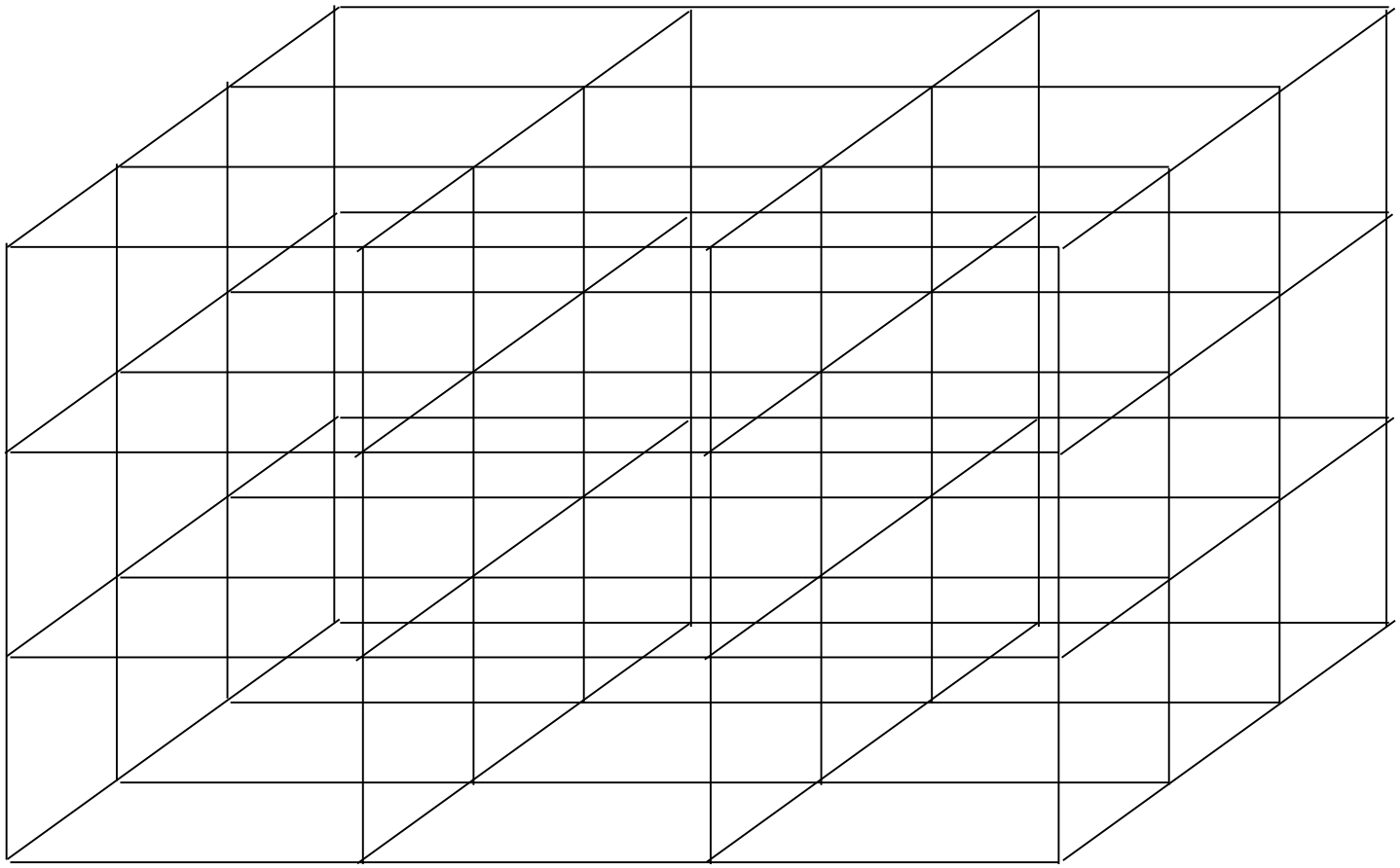
- Pakreipiam P koordinatę pagal S:

$$s = (\vec{T} \times \vec{U})(\vec{P} - \vec{P}_0) / ((\vec{T} \times \vec{U})\vec{S})$$

- Tas pats T,U
- Algoritmas:
  - Panaudojam smulkų gridą
  - Deformuojam to grido taškus
  - Naudojam Bezier interpoliaciją kad gauti naują poziciją
    - Nauji grido taškai laikomi valdymais (control) taškais

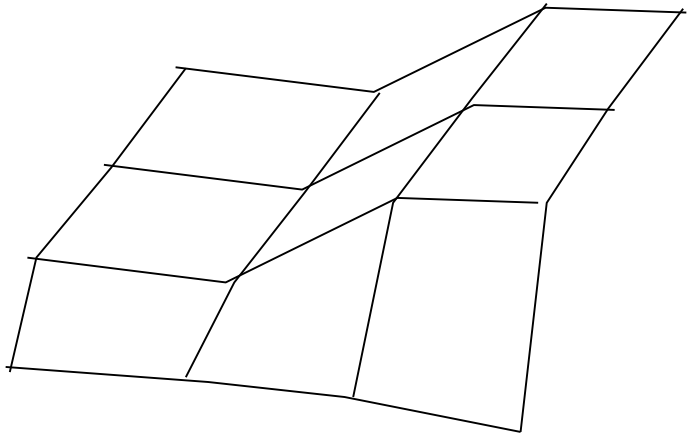
FFD – kontrolinis gridas

- Sukuriam kontrolinį gridą



FFD – taškų perkėlimai

- Judinam kontrolinio grido taškus
- Paprastai naudojama tri-cubic interpoliacija
- Arba Bezier interpoliacija.
- Arba B-spline, Catmull-Romm, tri-linear interpoliacija



FFD algoritmas

Užsikraunam modelį

Padalinam objekto paviršių į gabaliukus

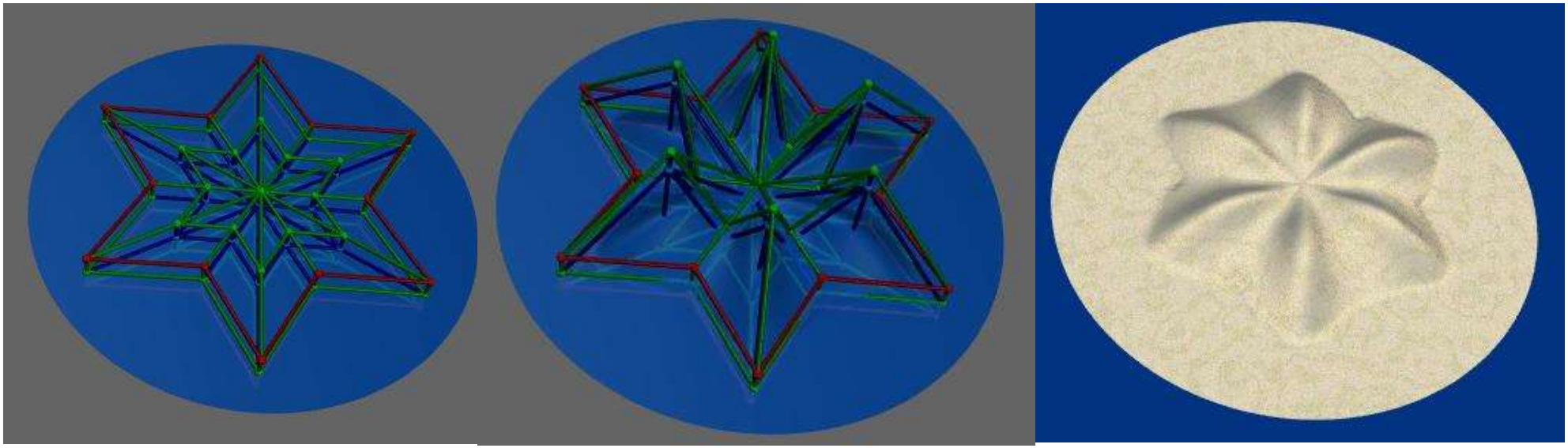
Ciklas 1:
užsikraunam grotelių modelį
uždedam groteles ant reikiamos objekto vietos
nustatom grotelių dimensijas
“užrakinam” groteles

Ciklas 2 :
deformuojam groteles
atnaujinam objekto modelį
end 2

“atrakinam” groteles

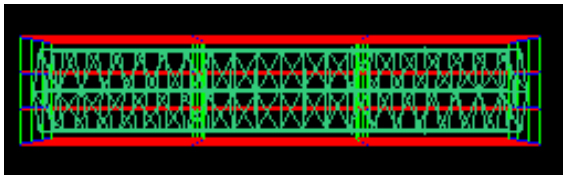
end 1

FFD - pavyzdys



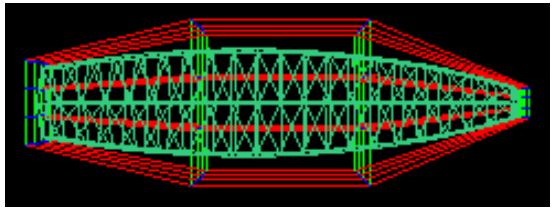
FFD - pavyzdys

1 etapas



Turim cilindrą.
Raudonai parodytas įdėtas FFD blokas ant to cilindro

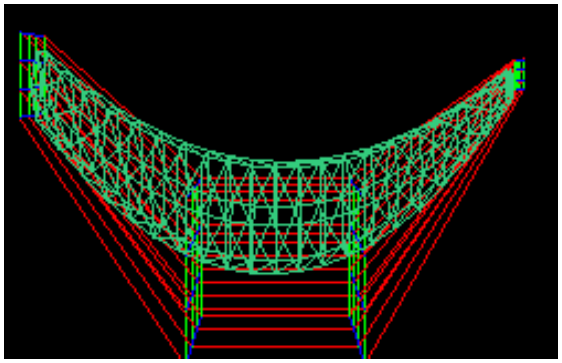
2 etapas



Pastumdom kontrolinius taškus kiekviename gale ir stebim kaip keičiasi vidinis cilindras

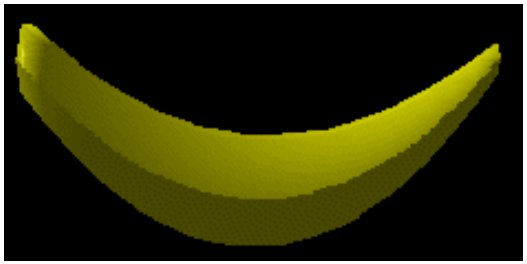
FFD - pavyzdys

3 etapas



pastumiam vidinius kontrolinius taškus žemyn

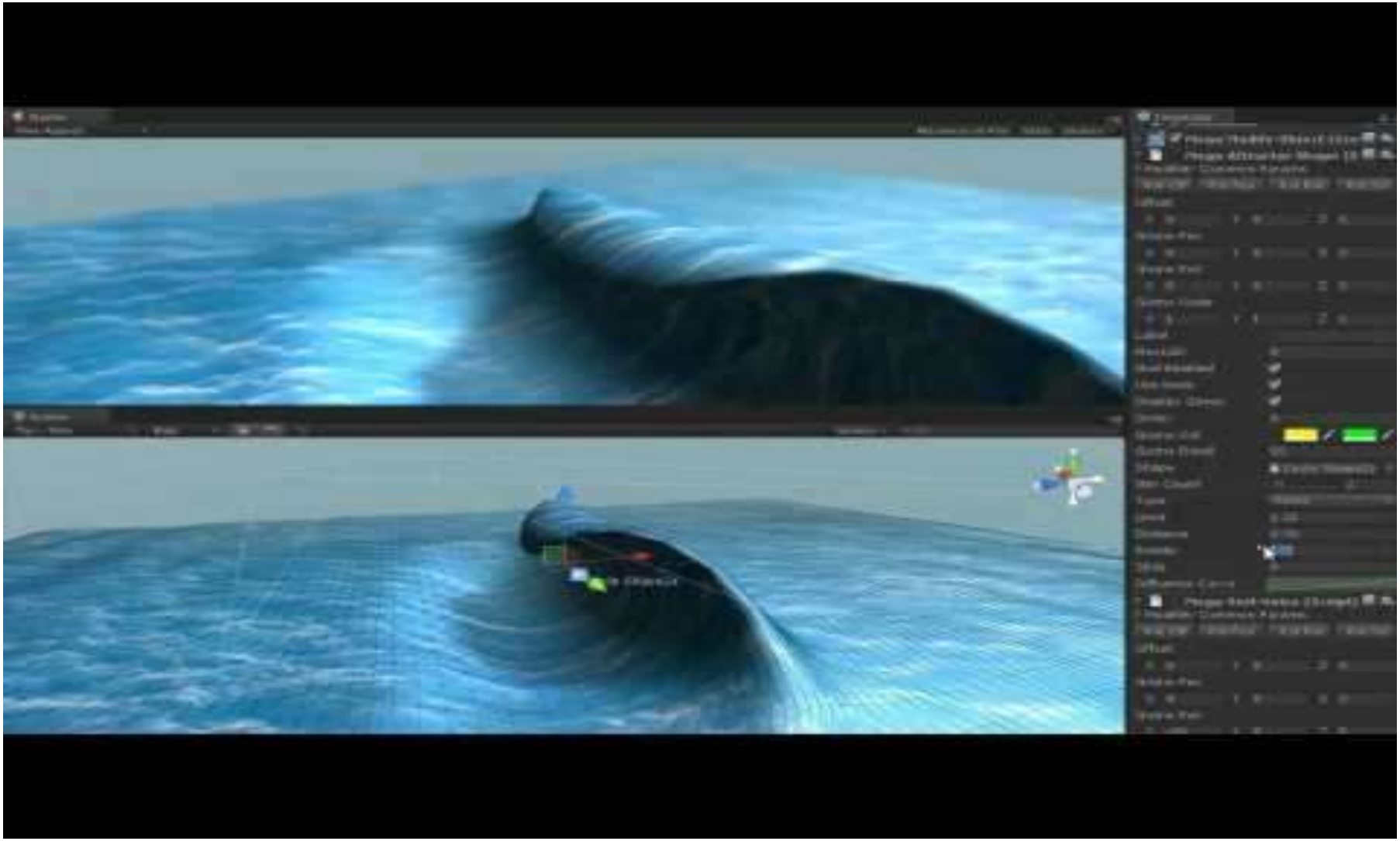
4 etapas



Gauname bananą ;)

Bezier kreivės ir paviršiai

MULTIMEDIJOS INŽINERIJOS KATEDRA

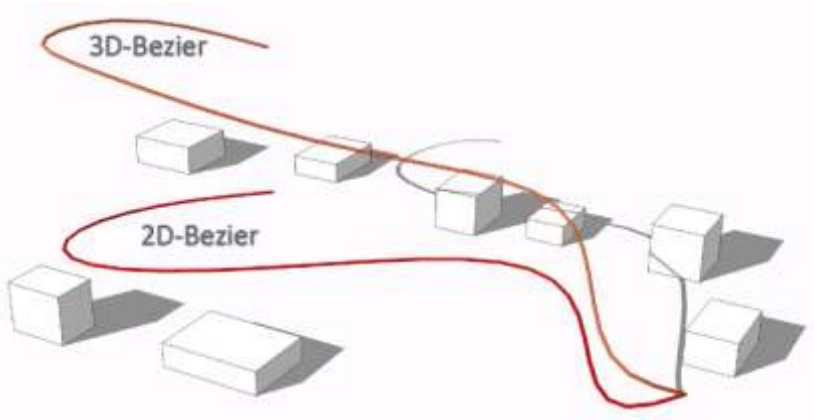


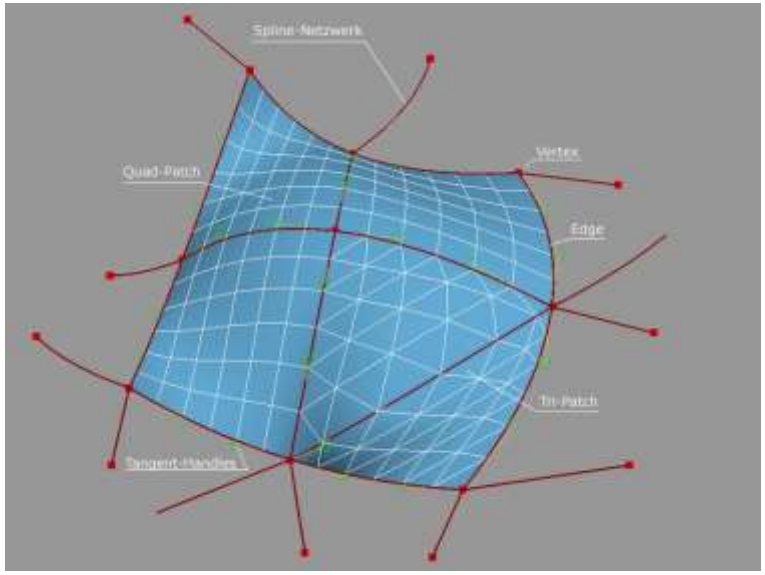
https://www.youtube.com/watch?v=_sFCPfc7mDU

Bezier kreivės ir paviršiai

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Bezier kreivė: 1D
- Bezier plotai: 2D
- Bezier formos (3D)
- Kiekvieną tašką (s,t,u) galima užrašyti valdymo taškais
  - Kubiniu daugianariu per s, t, u (kubinis Bezier)
- Nauja pozicija:
  - Ta pati išraiška, tie patys s,t,u
  - Naujos kontrolinių taškų pozicijos





FFD per Bezier

Grotelėmis apibrėžiamas Bezier tūris

$$\mathbf{Q}(u,v,w) = \sum_{ijk} \mathbf{p}_{ijk} B(u) B(v) B(w)$$

Skaičiuojamos grotelių koordinatės

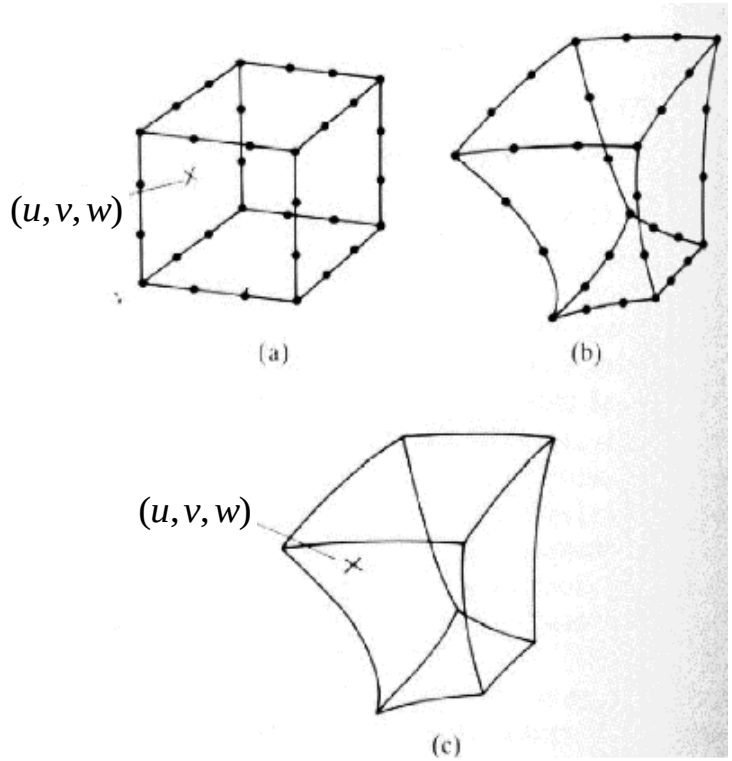
(u,v,w)

Modifikuojami kontroliniai taškai

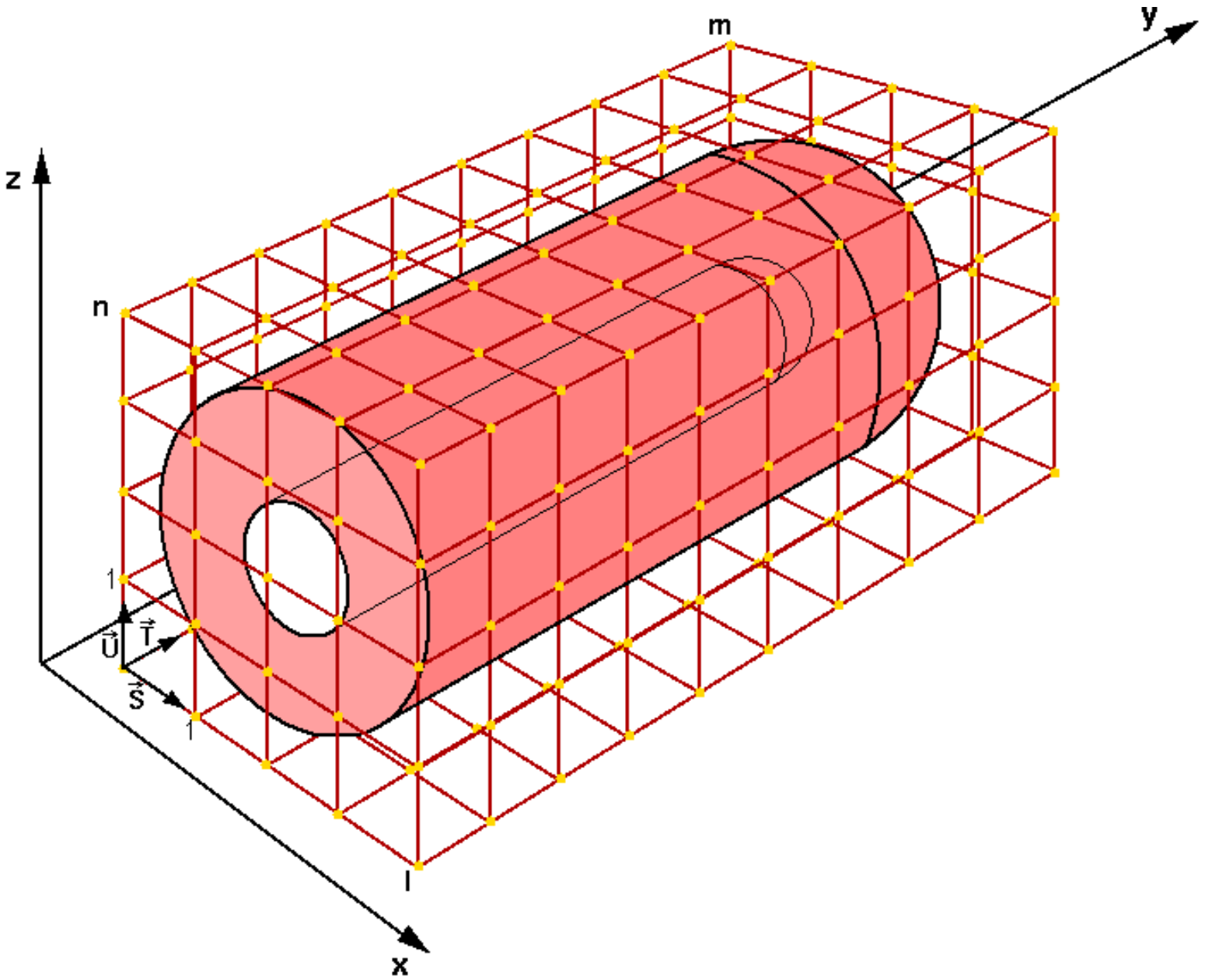
$\mathbf{p}_{ijk}$

Skaičiuojami deformuoti taškai

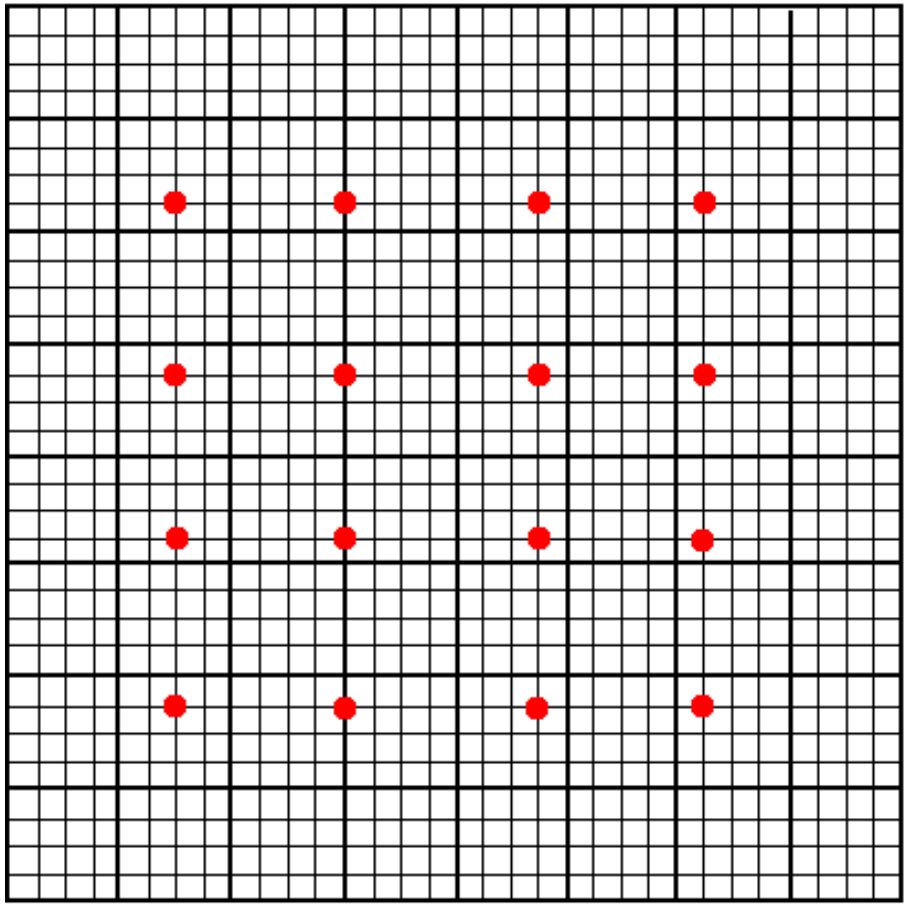
$\mathbf{Q}(u,v,w)$

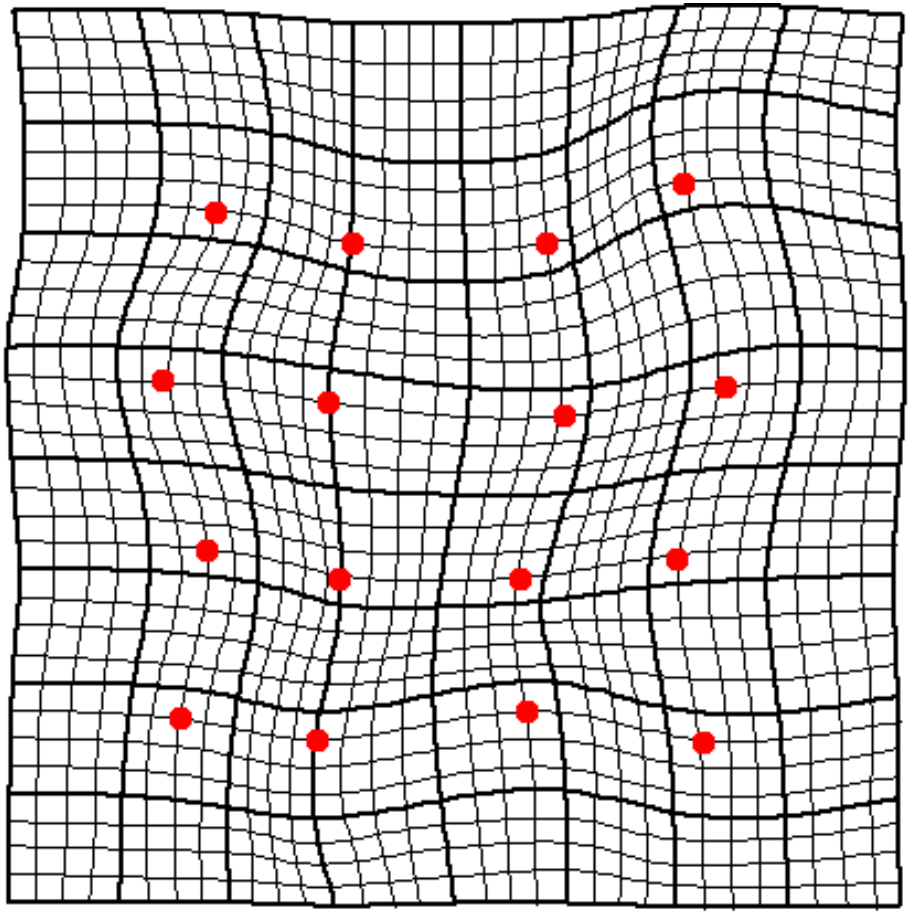


FFD - pavyzdys

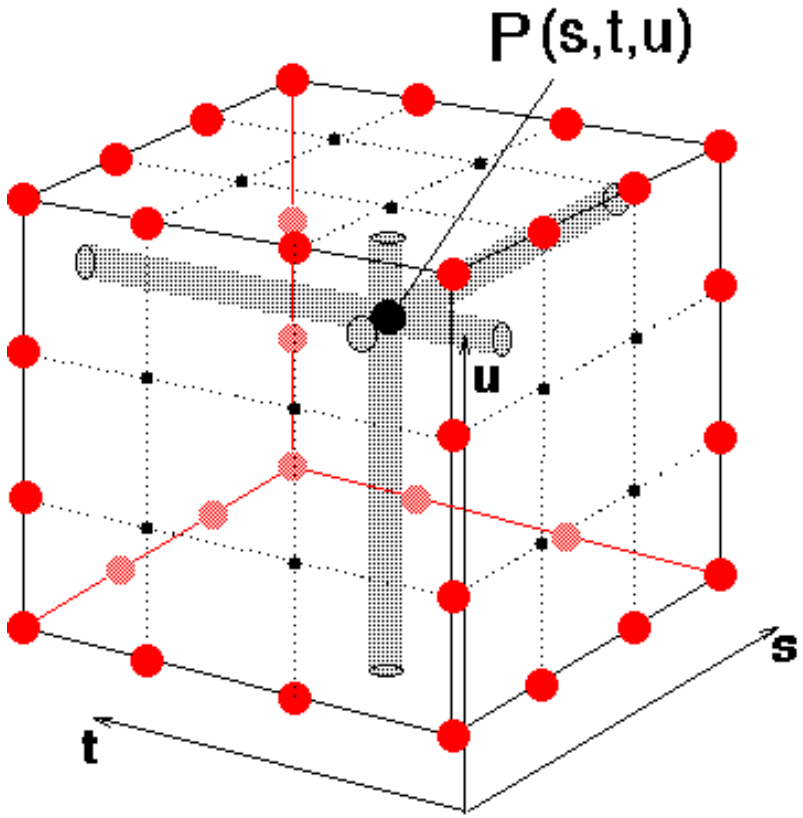


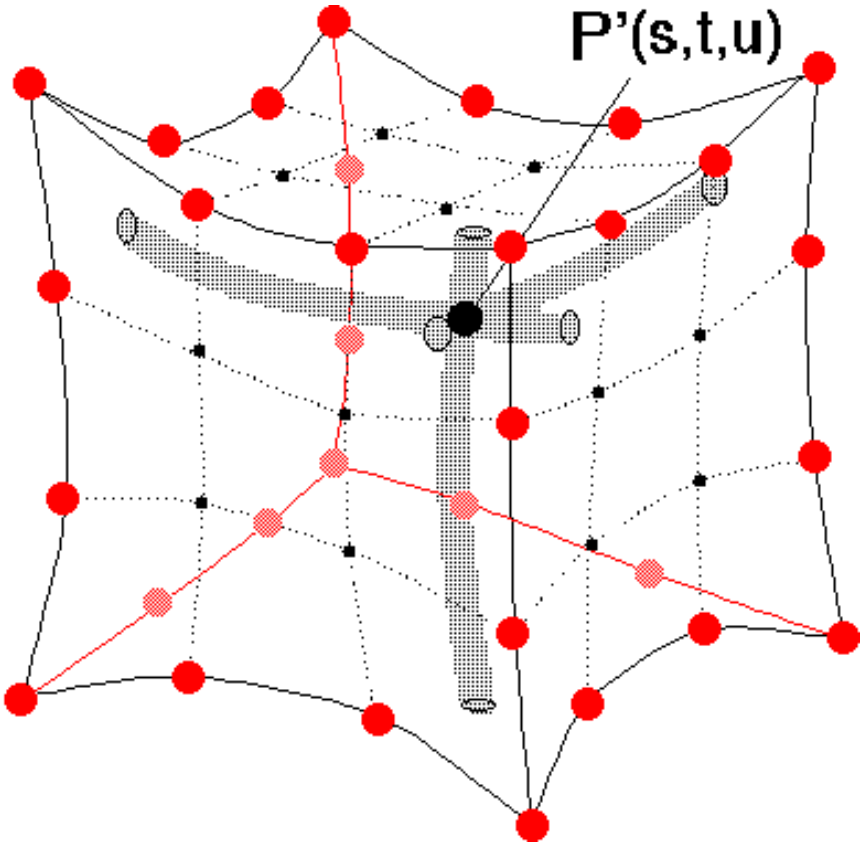
FFD - pavyzdys



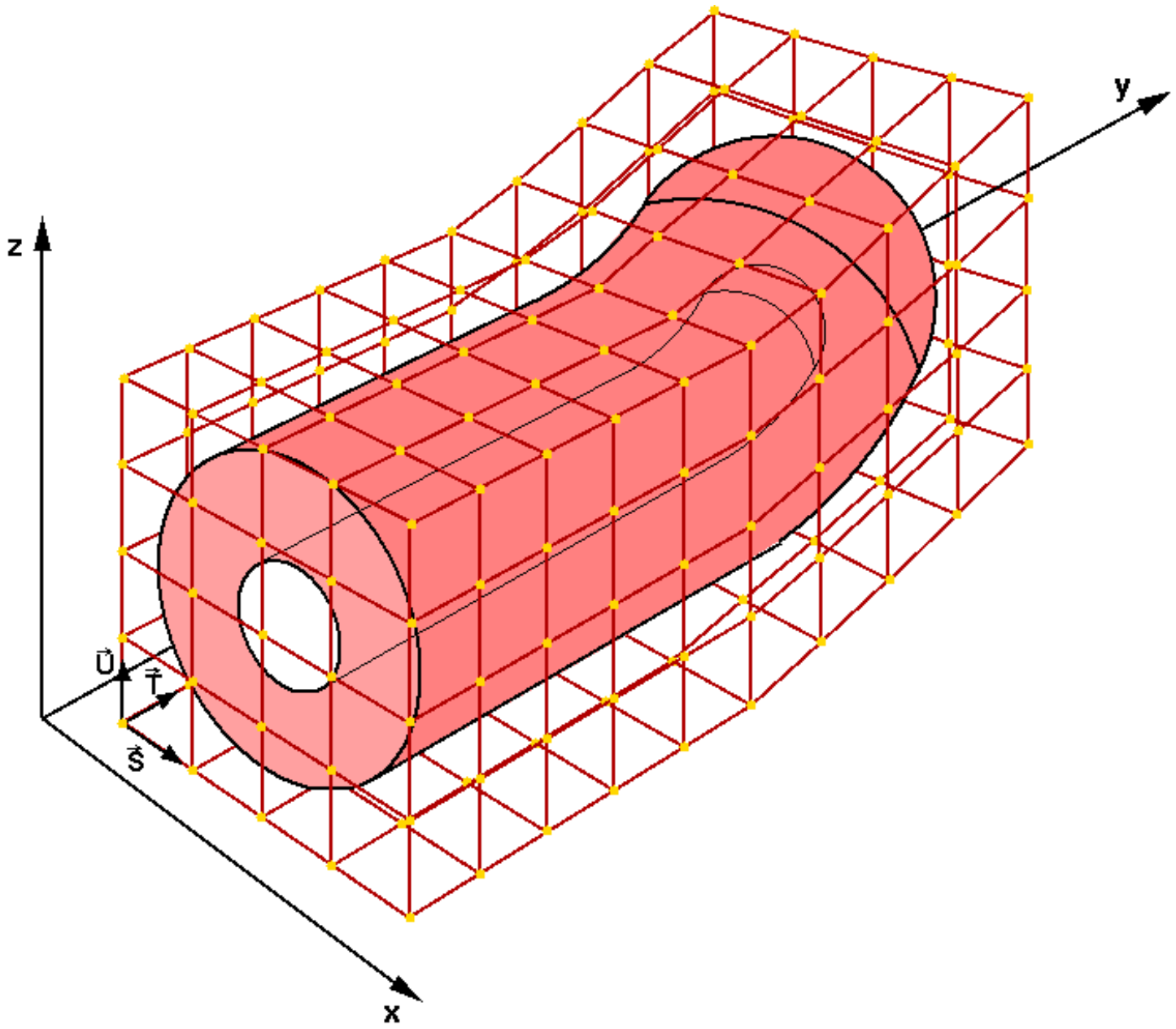


FFD - pavyzdys

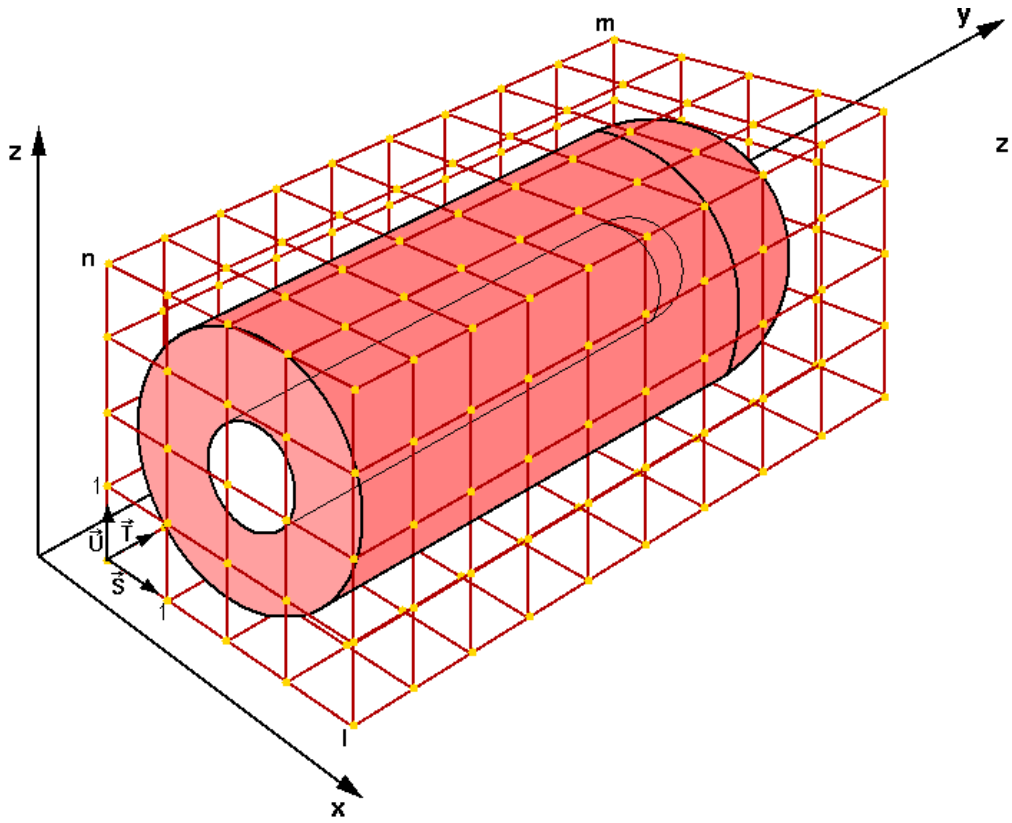


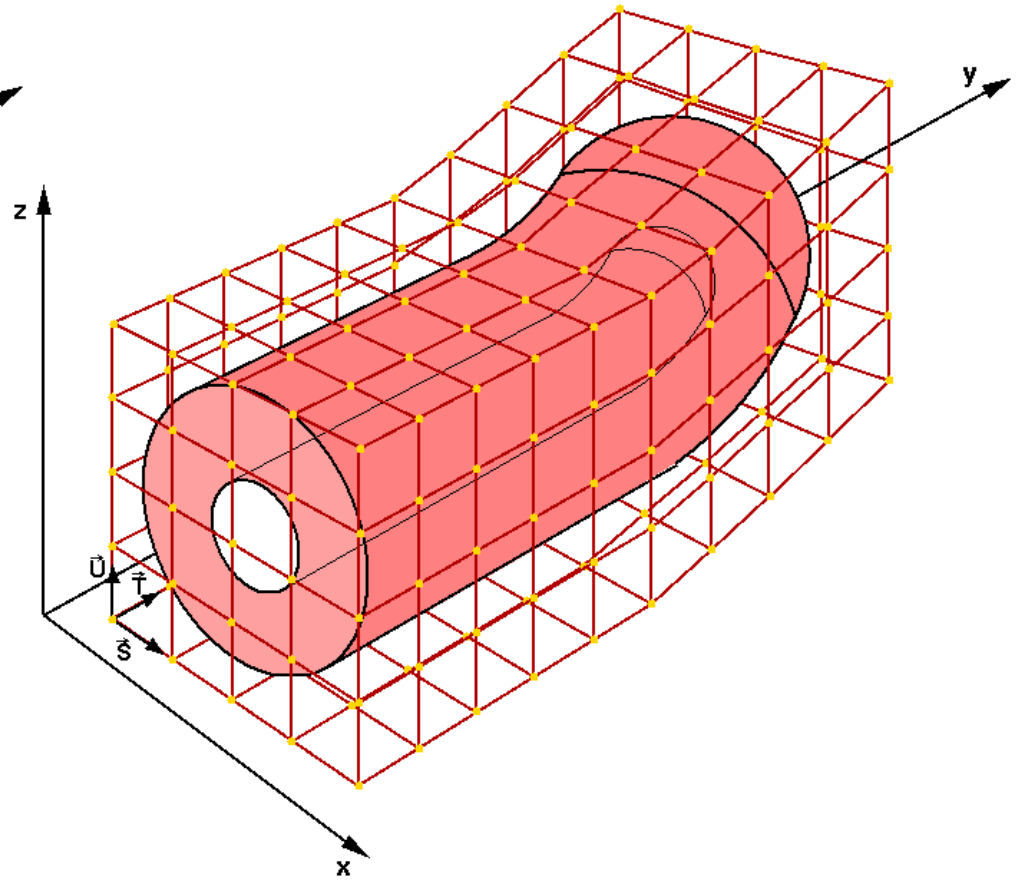


FFD - pavyzdys

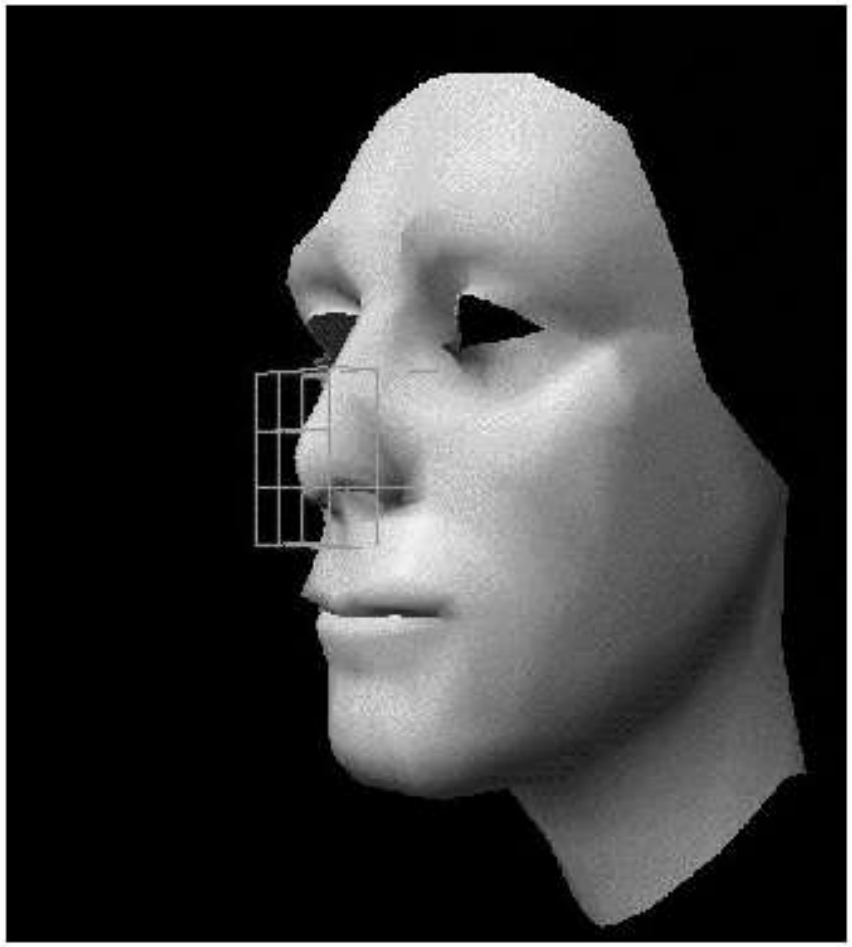


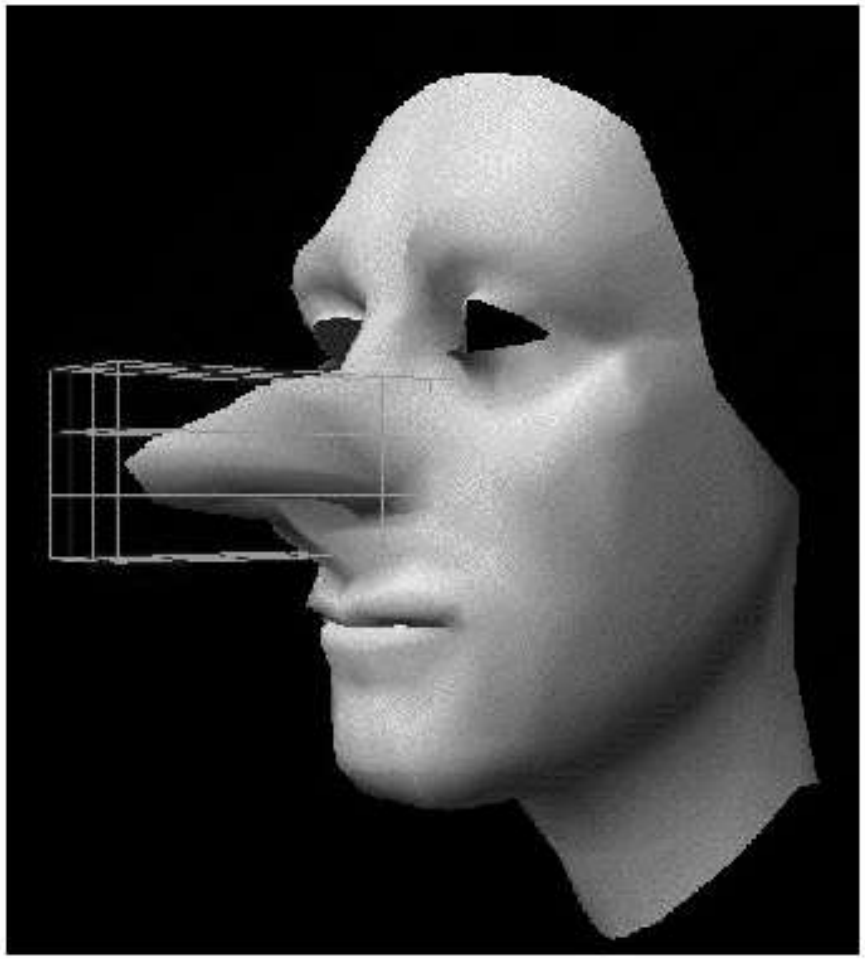
FFD - pavyzdys



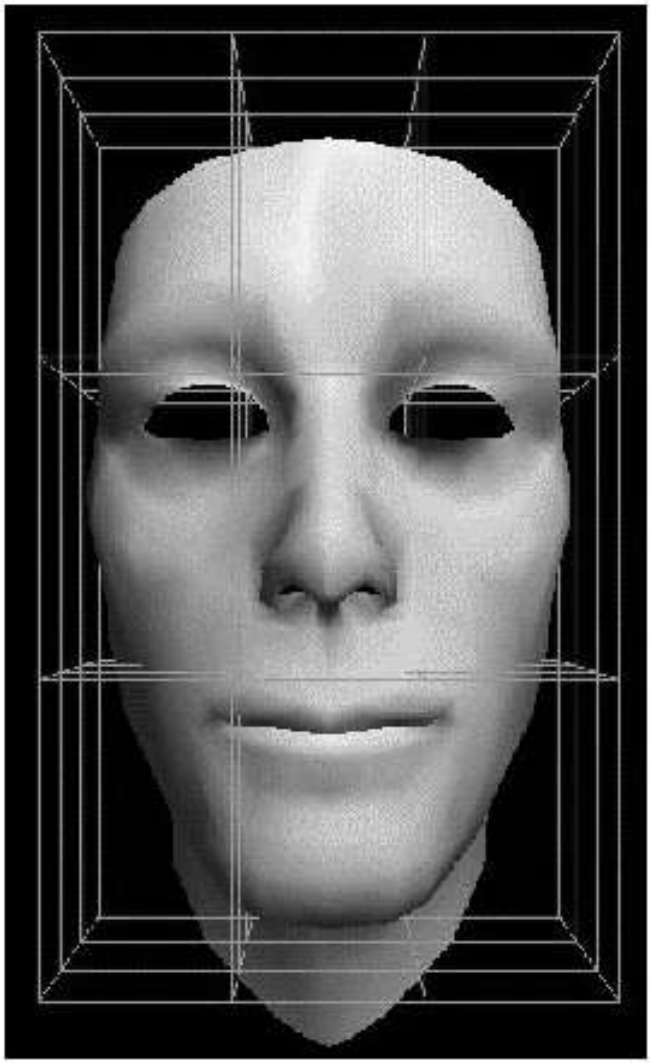


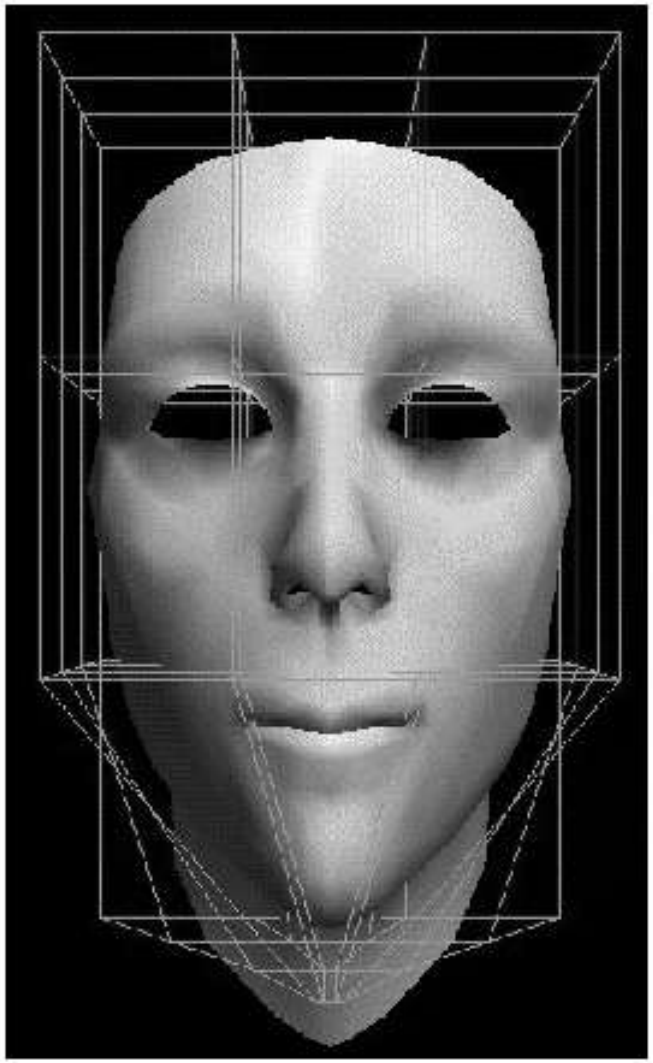
FFD - pavyzdys





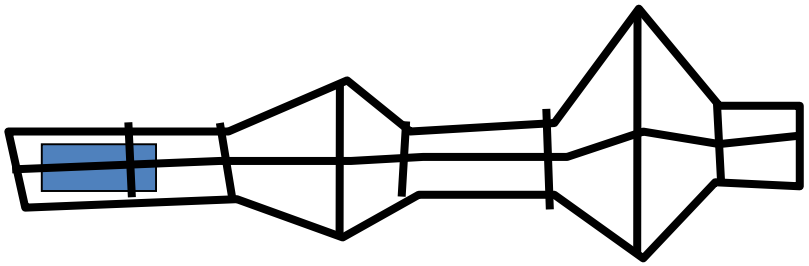
FFD - pavyzdys





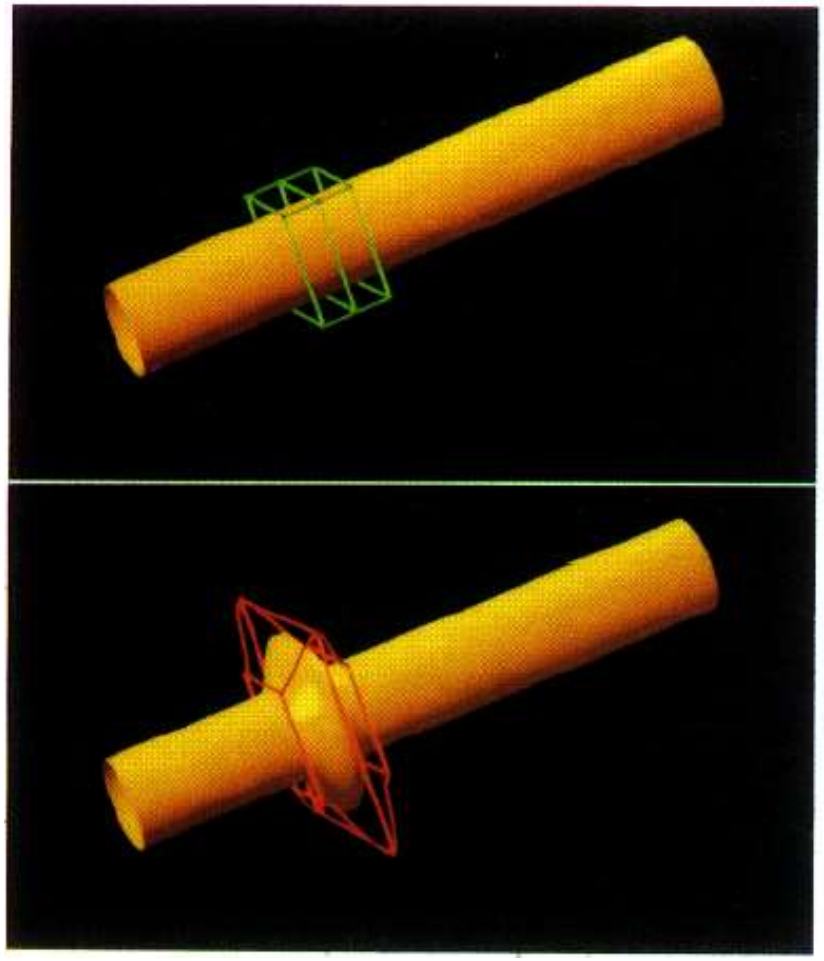
Animacija su FFD

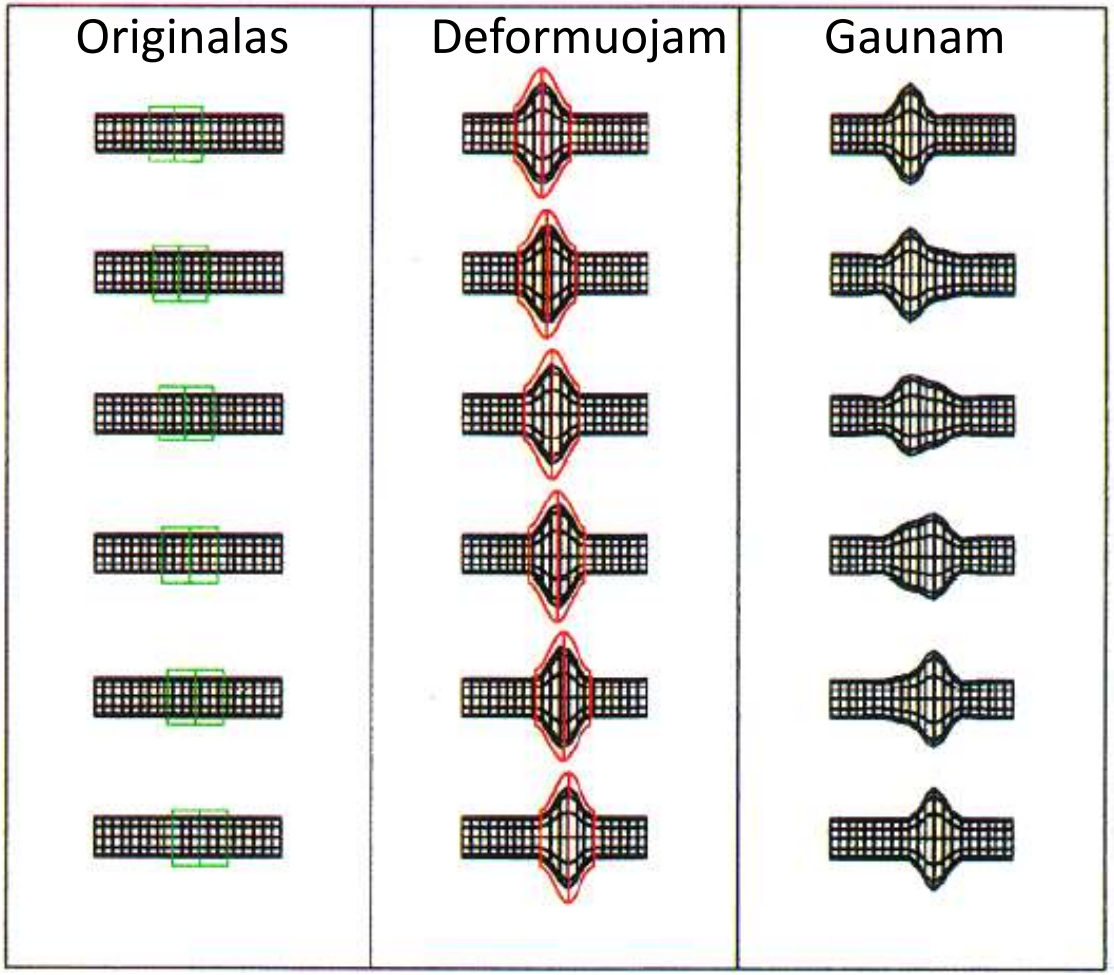
- **Hierarchiniai FFD**
  - Stambių vienetų (Coarse level) FFD modifikuoja viršūnėles ir smulkesnius FFD gridus
- **Objektą galima „pratempti“ per deformuojantį elementą**
  - Arba tampyti patį deformatorių
- **„Gyvai“ modifikuoti FFD kontrolinius taškus**
  - Pavyzdžiui, pagal žaidimo fiziką.



Animacija su FFD

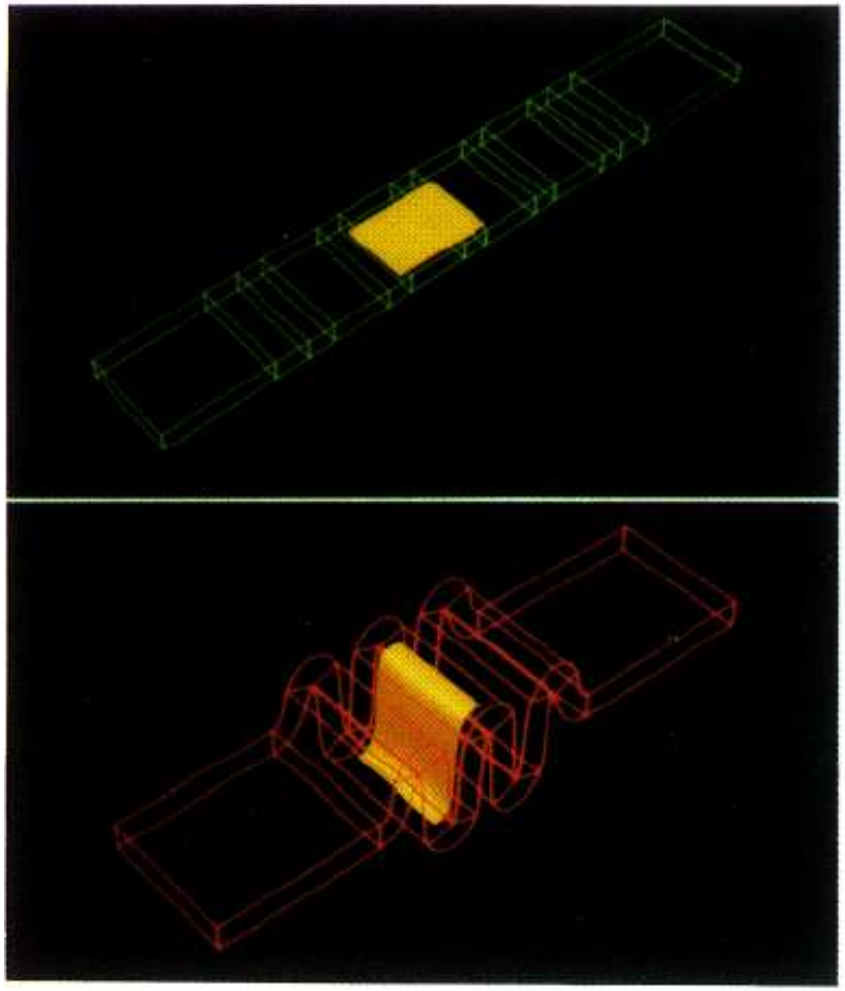
Animaciją sukuria deformacijos procesas

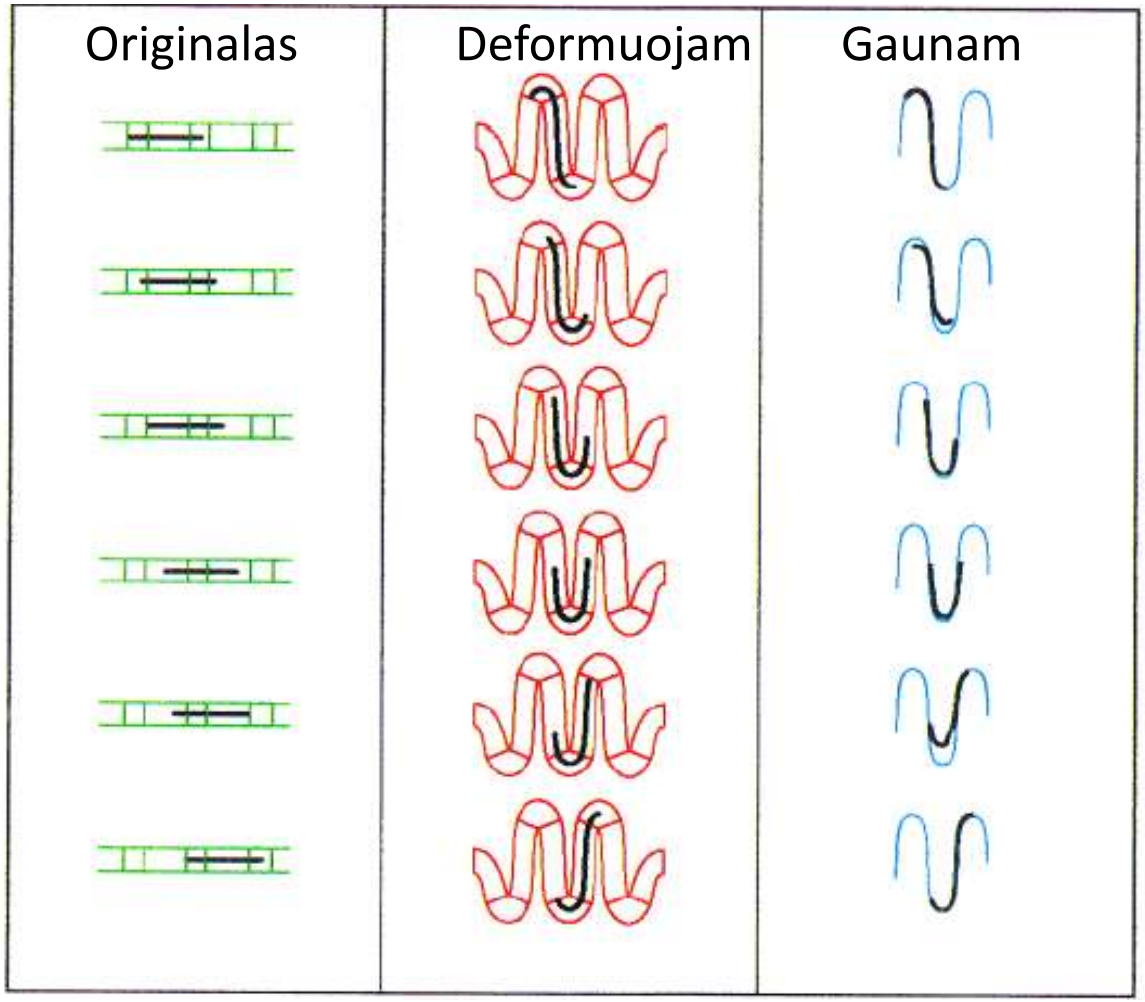




Animacija su FFD

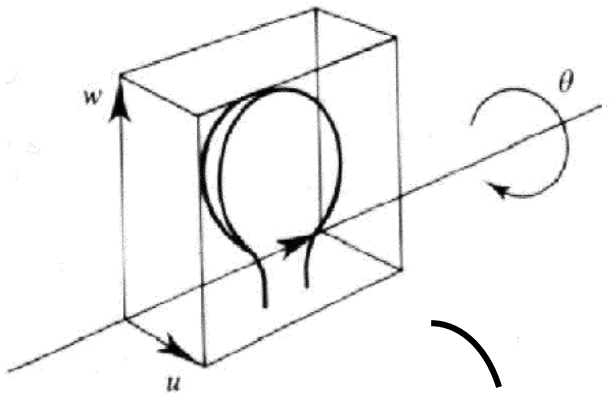
Animaciją sukuria deformacija per groteles



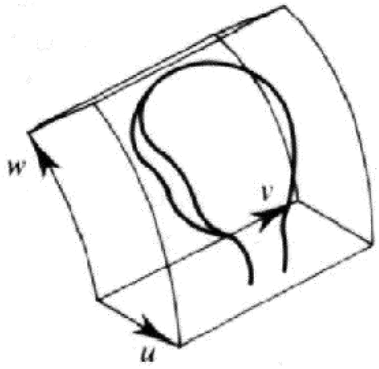


Faktorinės kreivės (Factor Curves)

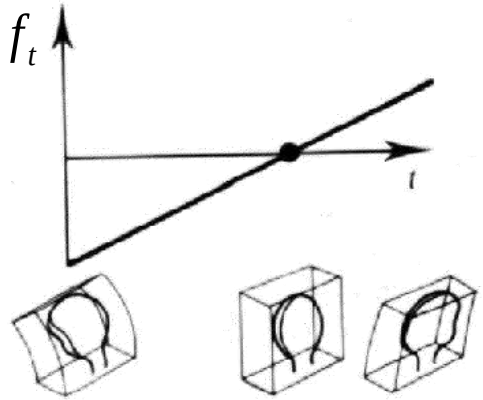
Objekto tranformacija modifikuojama priklausomai nuo to kur ir kada ji panaudojama



$R_w(\theta)$



Space factor curve

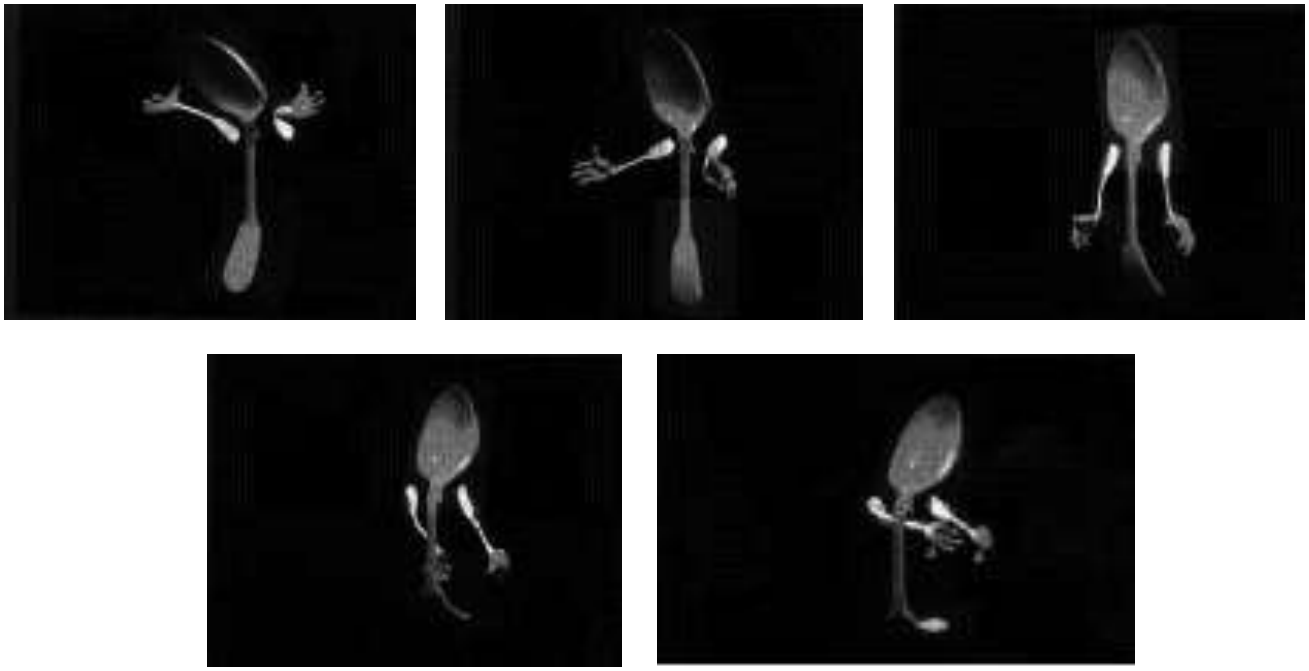


Time factor curve

$$\theta = \theta_0 \; f_w(w) \; f_t(t)$$

Faktorinės kreivės

Dažniausiai naudojama skriptais aprašomoms animacijoms kurti – t.y. labai sudėtingiems judesiams



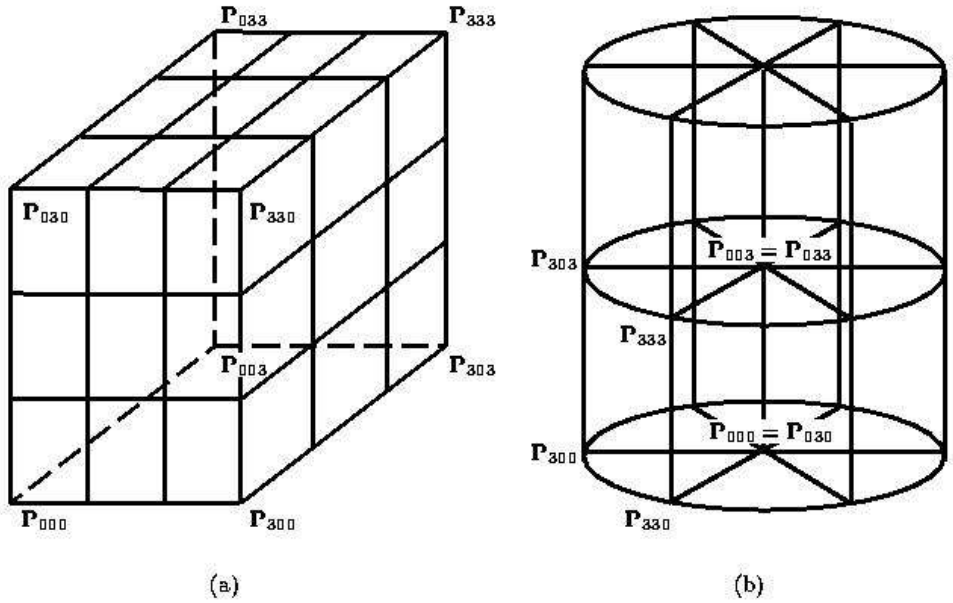
FFD + ir -

- Privalumai:
  - Nepriklausoma paviršiaus geometrija
  - Intuityvu, interaktyvu
  - Efektyvu, lankstu
- Trūkumai:
  - FFD grotelių forma riboja ir neleidžia specifiškai deformuoti (pvz., padaryti griežtos formos iškilumą).

extended FFD

- FFD groteles galima paredaguoti prieš pririšant prie modelio.
  - Perkėlimas
  - Apjungimas
  - Ištrynimas

  Kontrolinių taškų nustatymas



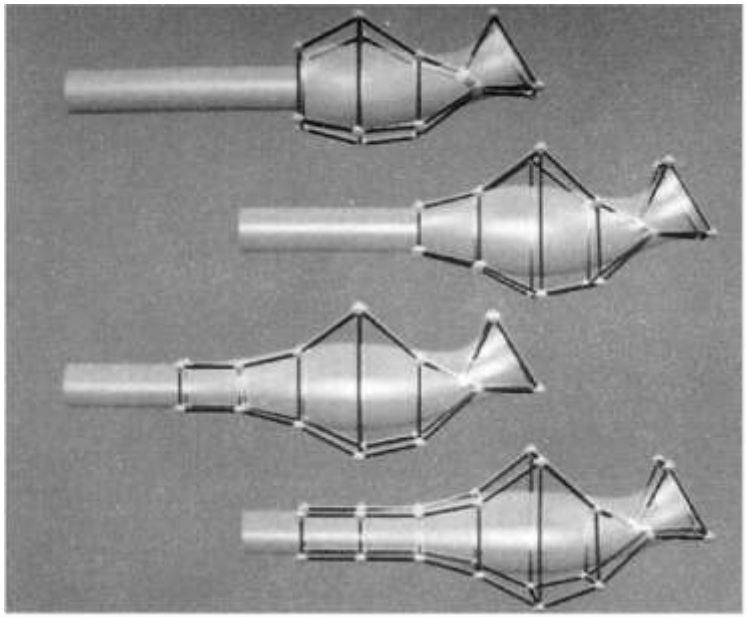
- Panaudoti nestandartines grotelių formas:
  - Prizmė
  - Ketursienis
  - …

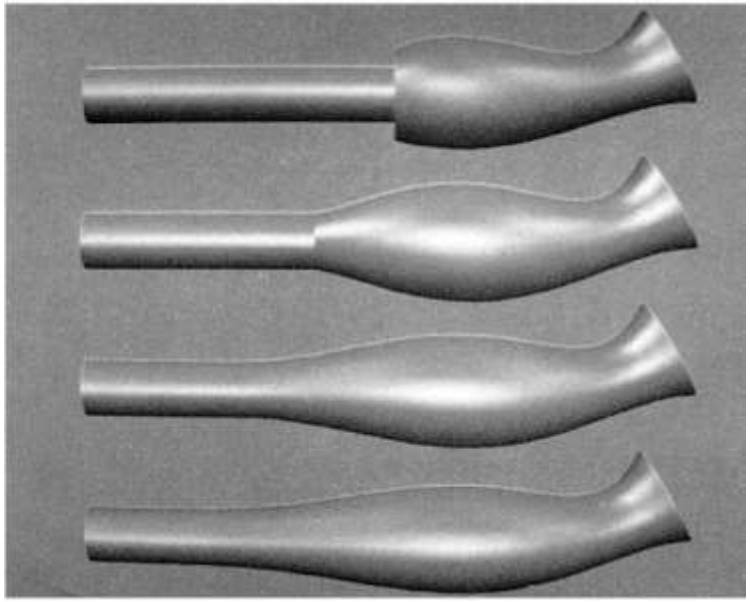
eFFD

- **Galima sukurti groteles sudarytas iš daug grotelių blokų**
- Sukuriamas masyvas iš (3l+1)(3m+1)(3n+1) grotelių blokų.
- Sąveikaujam tik su kampiniais taškais, kiti kontroliniai taškai nustatomi automatiškai.
- Daromas eFFD su kiekvienu bloku, vienas po kito iš eilės.

eFFD

Deformuotų paviršių išlyginimas galimas teisingai nustačius grotelių poziciją ir dimensijas

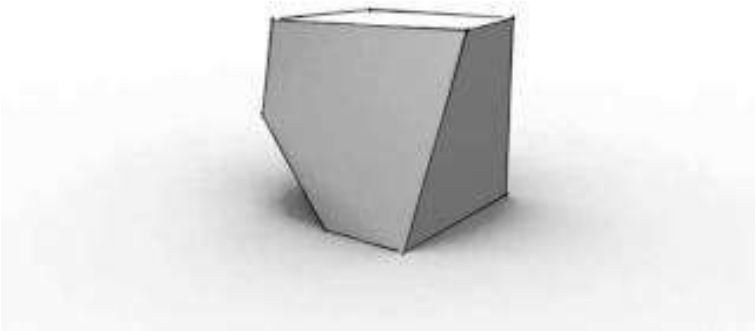




Deformuojam šeideriu

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Kadangi vertex šeideriai gali (ir yra ;) būti panaudoti viršūnėlių apdorojimui ir transformacijai, kone idealu juos naudoti objektų deformacijai.
- Pats procesas yra pakankamai lengvas ir nesudėtingas.
- Pavyzdžiui, turime stačiakampę konstrukciją ir jos dalis norime paploninti ar pastorinti programiškai.
- Šiam tikslui pasidarom vertex šeiderį kuris pastumdys atitinkamas viršūnėles palei jos normalę duotuoju atstumu.

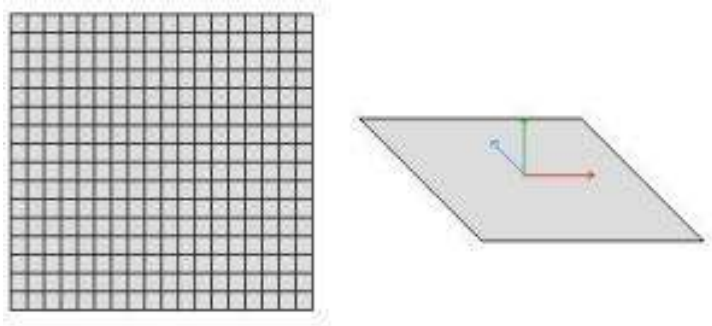


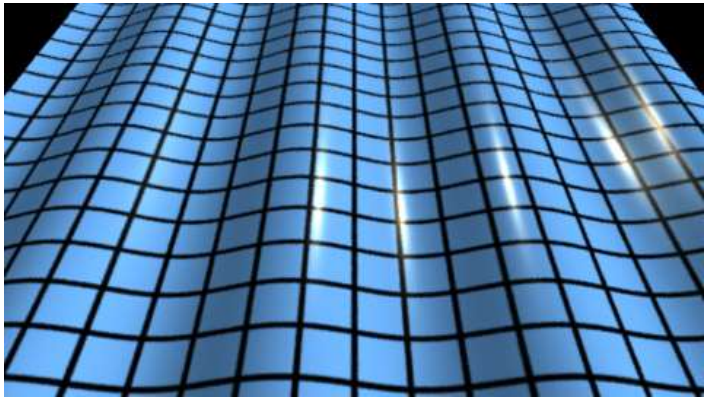
Deformuojam vandenį

MULTIMEDIJOS INŽINERIJOS KATEDRA

Bangos

- Kam animuoti visą vandens objektą, švaistyti skaičiavimų resursus, turėti nerealistinius objektus, jei galima visiškai nesunkiai sumodeliuoti panaudojus vertex šeiderį?
- Imate didelę plokštumą (flat mesh) kuri vaizduos jūsų vandenyną be bangų (arba 3d modelis arba sugeneruojate programiškai).
- Priklausomai kiek norėsite turėti „valdomų“ bangų, atitinkamai parenkate viršūnėlių skaičių.
- Šeideryje judinate tas viršūnėles aukštyn žemyn (y ašis) pagal sinuso ar kosinuso f-ją arba kaip modeliuojate scenarijuje.
- Jei jūsų judesys yra f(y)=sin(y), tai konkrečios viršunėles V_x Y ašis bus apibrėžta taip sin($V_{X.pozicija}$+laiko momentas);





Deformuojam vandenį

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Norite realistiškesnių bangų?
- Padarykite, kad kiekviena banga turėtų judesio skirtumų (random + y)
- Kistų ne tik aukštyn žemyn, bet ir į šonus.
- Maišykite sinuso ir kosinuso funkcijas.
- Kaitaliokite kryptis.
- Naudokite iškilumų žemėlapius, jais kurkite atplaišas ir bangeles.

Darom patys

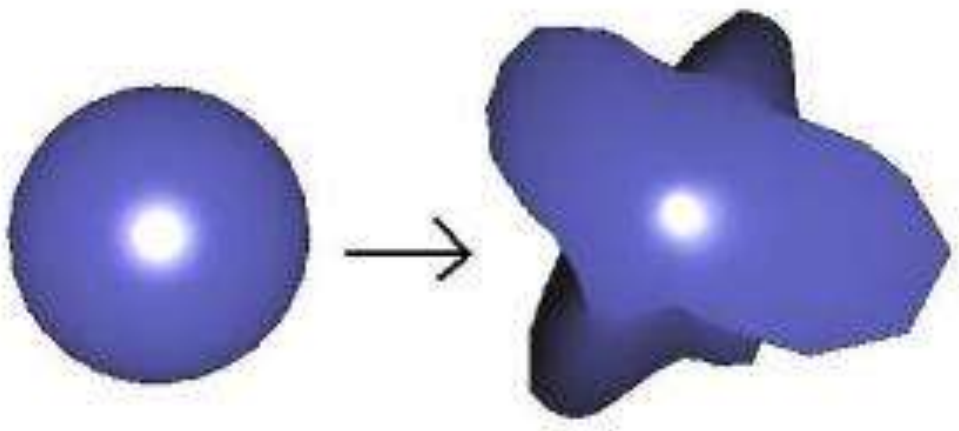
- Įsidedam plane objektą
- Įsidedam vandens šeiderį (kodas moodle)
- Derinam

- Galima kombinuoti su tekstūrų efektais („flowmap“):
- http://algoholic.eu/animated-water-shader/ (kopija moodle)

Deformuojam figūras

Netikrosios sferinės harmonikos

- Šis pavyzdys apjungia anksčiau minėtus taikymus.
- Grafinis objektas (šiuo atveju sfera) yra modifikuojamas panaudojant sinuso/kosinuso funkcijas ir judinant viršūnėles per jų normales taip iškraipant objektą.
- Papildomai įvestas laiko faktorius ir objektas atrodo animuotas.



Deformuojam figūras

- Esminį darbą čia atlieka vertex šeideris (pixel šeideris čia atlieka tik apšvietimą).
- Čia pat galite priderinti ir iškilumų žemėlapius.
- Kadangi objektas kis laike („animuotas“) jums reikės laiko kintamojo, pagal kurio vertę stumdysite viršunėles.
- Paimate prieš tai naudotą šeiderį ir pridedate taimerį.
- Toliau modifikuojate vertex šeiderį ir pridedate viršūnėlių pozicijų stumdymo variacijas.

Darom patys

MULTIMEDIJOS INŽINERIJOS KATEDRA

- Įsidedam objektą
- Jo material'ui pridedam deformacijos šeiderį (yra moodle)
- Derinam

- Reljefo generavimas
- http://catlikecoding.com/unity/tutorials/noise-derivatives/

Meshinator

http://u3d.as/content/mike-mahoney/meshinator-realtime-mesh-deformation/4vC