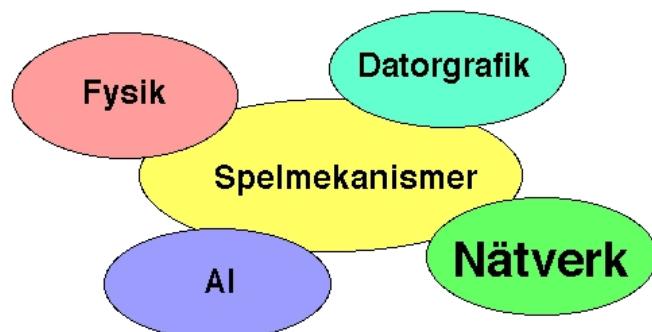


TSBK 03

Teknik för avancerade datorspel

Ingemar Ragnemalm, ISY



Föreläsning 7

Animation

Inverse kinematics

Skinning

Deformerbara objekt



Animation i grundkursen

- Hierarisk modellering (parenting)
- Introduktion till karaktärsanimation
- Partikelsystem (oberoende partiklar)



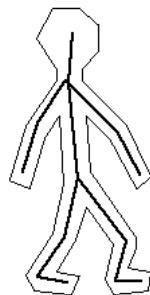
Animationsämnen denna föreläsning

- Inverse kinematics
- Skinning
- Deformerbara objekt



Skin & bones

En 3D-modell av en människa eller liknande djur modelleras med både skinn och ben.



Animationer kan sedan beskrivas enbart genom att röra på benen.



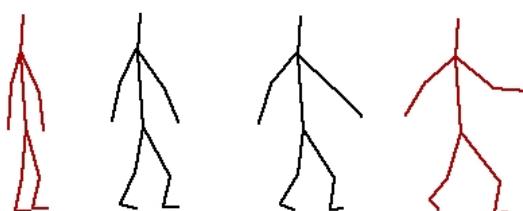
Key-frame animation

Pre-rendered animations

Key-frames are designed at suitable intervals

Frames between keyframes are interpolated (morphed)

Very common method for real-time animation





Key-frame-animation

- Hela kroppen som en animation. Ger problem när man vill kombinera rörelser, t.ex. benrörelse och armrörelse. "Skating".
- Dela upp kroppen i flera delar, olika animationer på delarna.
- Interpolation av animationer. Mjuka övergångar, minska konflikter.



Motion capture

Extremely common in movies!

- Record by natural visuals only
- Tracking markers
- Active sensors on the body

Perfect for pre-generated animations.



Motion capture

By markers or visual: Mainly an image processing problem.

Cheap! (Fairly) easy to set up, no expensive custom hardware.

- Camera calibration
- Proper matching of markers by model.
- Two or more cameras for stereo pair matching.
- Fitting captured data to the skeleton.
- Motion retargeting

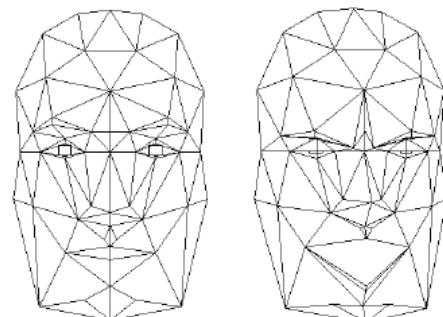


Face animation

Hard problem - we are very sensitive to errors!

Animate by action units
(muscle based) or face
animation parameters
(extreme detail)

FAPs part of the MPEG-4 standard.



The Candide model



Face animation

Parametrized animation (action units, FAPs). Detail level a conflict between animators and communicators

Muscle-based models

Model-based tracking

Lip tracking - detectable boundary

MPEG-4 standard

Retargeting of face animation



Face animation without mocap

Control animation by other means.

- Audio-based animation
- Text-based animation (text-to-speech)
- Manual editing

Audio-based easy to work with for content creators.



Forskning på ISY

Model based coding, ansiktsanimation för kommunikation.

Candide-modellen, berömd tidig låg-poly-modell

**Mocap av ansikten, enbart naturlig scen, realtid
(Jörgen Ahlberg)**

Animation från tal (Goranka Zoric)

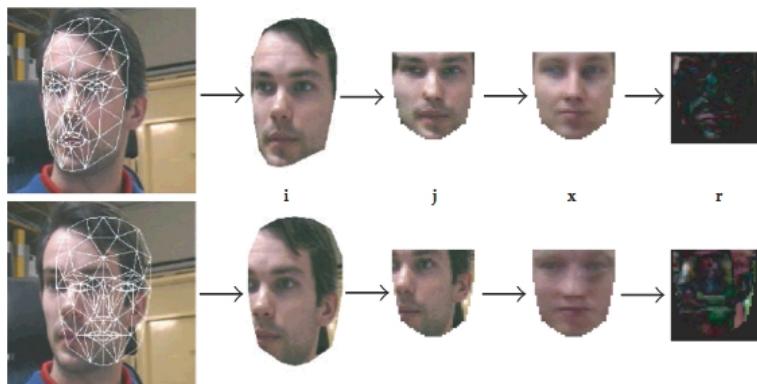


Mocap av ansikten, modellbaserad bildkodning

Modellbaserad, jämför 3D-modell med ansiktstextur med bilden.

**Naturlig scen -
inga markörer**

**Både följer
övergripande
rörelser och
fångar
detaljrörelser**





Egenansikten

SVD-baserad metod, "eigenfaces"

Egenansikten beräknas från databas av ansiktsbilder, bryts ner i mest signifikanta komponenter

Ansikten kan sedan kännas igen med hjälp av databasen, både för identifiering och för att avgöra var i en bild ett ansikte finns.



Kinematik

Rörelse utan krafter.

Forward kinematics: Ange vinklar, position ges av vinklarna. Lätt att implementera, svårt att använda.

Inverse kinematics: Ange position, räkna ut vinklar för att uppnå denna position.

Viktigt i robotik.



Invers-kinematik

- Analytisk beräkning
- Iterativ beräkning med inversa Jacobianen
- CCD (Cyclic coordinate descent)



Invers-kinematik analytiskt

Önskad slutposition anges

Varje led har ett antal frihetsgrader (vinklar)

Överbestämt, olinjärt system

Fungerar för små system, ohanterligt för stora.



Iterativ metod

Beräkna derivata av alla utdata m.a.p. invärden.

Ger en NxM-matris - Jacobianen.

Denna beskriver framåt-kinematiken lokalt.

$$V = J\theta'$$

V derivata av utdata (position), θ' derivata på indata

$$J^{-1}V = \theta'$$

ger oss derivatan på vinklarna.



Iterativ metod

Jacobianen normalt inte symmetrisk! Ej inverterbar!

Pseudoinversen används.

Även en variant baserad på transponat av Jacobianen finns.



Cyclic Coordinate Descent

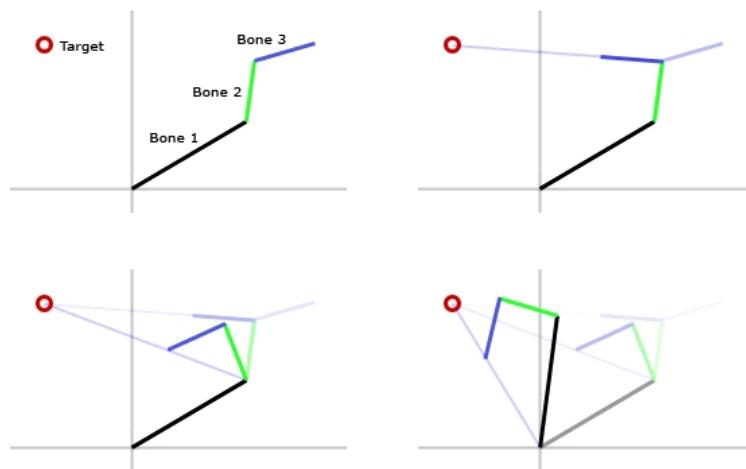
Iterativ metod som riktar ledar explicit. En led i taget vrids åt rätt håll. Arbetar utifrån och in.

Vrid en led i taget så att man kommer så nära målet som möjligt. Starta med den yttersta, jobba mot roten.

Upprepa tills målet nås eller avståndet är litet nog.



CCD, exempel



(<http://www.ryanjuckett.com>)

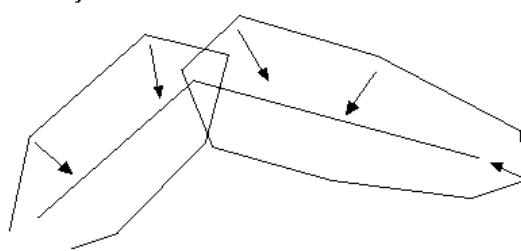
Skinning

Precis som bump mapping: En fråga om
koordinater och transformationer

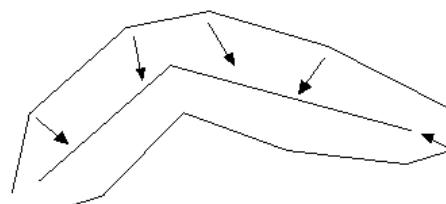
Tre sätt att animera en kropp

- Parenting
- Stitching
- Skinning

Parenting: Separata
stela meshar följer
varje ben

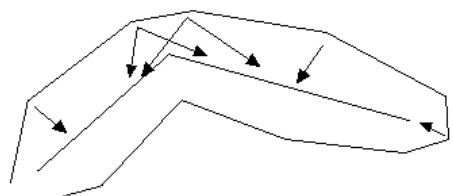


Stitching: Varje
vertex följer *ett* ben



Skinning: En vertex kan
viktas efter flera ben

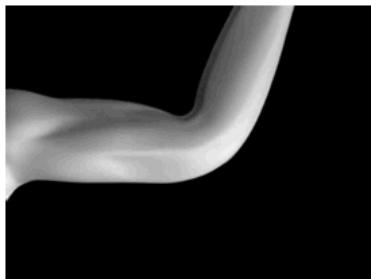
Två närliggande ben,
följ båda med lämplig
viktning



En närliggande, följ
benets transform

Avancerad skinning

Grundläggande algoritmen har vissa svagheter



*<- Collapsing
elbow*
Twist ->



Kan åtgärdas med

- Flera ben i en och samma led
- Shape blending - flera modeller som man morphar mellan
- Volymbevarande tilläggsvillkor

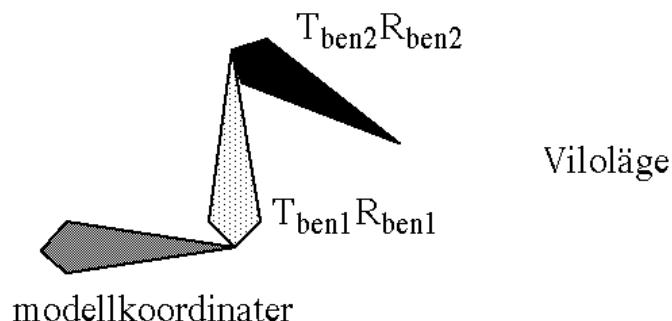
Koordinatsystemen

Mesh given i modellkoordinater

Mesh och skelett givna i viloläge

Hur påförs animation?

Hur beräknar man en vertex modifierade position?



Varje bens position definieras av en transformation, en translation och en rotation:

$$M_{ben2} = T_{ben2}R_{ben2}$$

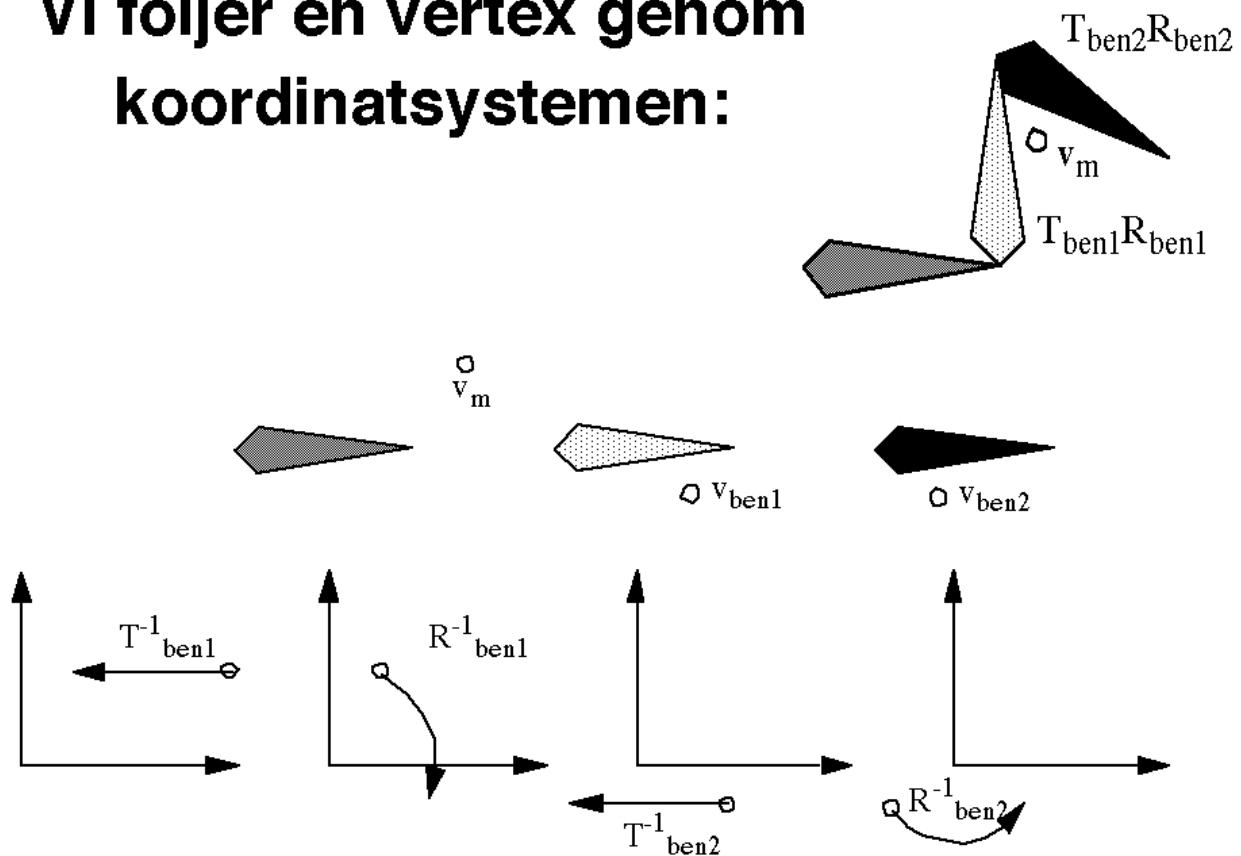
Transformation av vertex från modellkoordinater till benkoordinater:

$$v_{ben2} = M_{ben2}^{-1}M_{ben1}^{-1}v_m$$

Transformation av vertex från benkoordinater till modellkoordinater:

$$v_m = M_{ben1}M_{ben2}v_{ben2}$$

Vi följer en vertex genom koordinatsystemen:



Animation:

Transformera till benkoordinat med viloläget.

Transformera tillbaka till modellkoordinater med modifierade benpositioner

$$\mathbf{v}'_m = M'_{\text{ben}1} M'_{\text{ben}2} M^{-1}_{\text{ben}2} M^{-1}_{\text{ben}1} \mathbf{v}_m$$

Animation oftast rotationer:

Violägestransformation är vilolägets rotation och translation

$$M_{\text{ben}} = T_{\text{vila}} R_{\text{vila}}$$

På denna tillkommer sedan animationens rotation

$$M'_{\text{ben}} = M_{\text{ben}} R_{\text{anim}} = T_{\text{vila}} R_{\text{vila}} R_{\text{anim}}$$

Traversera skelettet för resulterande transformation:

Modellkoordinater till benkoordinater:

$$M_{mb} = \prod M^{-1}_{ben,i}$$

Benkoordinater till modellkoordinater:

$$M_{bm} = \prod M_{ben,i} \cdot R_{anim,i}$$

och hela transformationen kan skrivas

$$v' = M_{bm} M_{mb} \cdot v$$

Viktning av vertexar

Fel i boken! Formel sid 133 är fel.

$$v' = \sum w_i M_i v$$

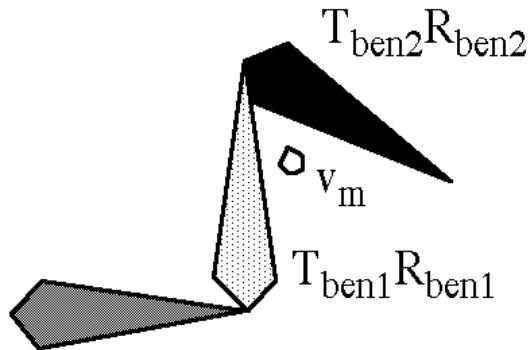
M_i beräknas, enligt ovan, som

$$M_{mbi} = \prod M^{-1}_{ben,j}$$

$$M_{bmi} = \prod M^{-1}_{ben,j} \cdot R_{anim,j}$$

$$M_i = M_{bmi} M_{mbi}$$

Exemplet med två ben ovan. Antag att v_m påverkas av ben1 och ben2.



$$M_1 = M'_{ben1} M^{-1}_{ben1}$$

$$M_2 = M'_{ben1} M'_{ben2} M^{-1}_{ben2} M^{-1}_{ben1}$$

$$v'_m = \sum w_i M_i v = w_1 M_1 v_m + w_2 M_2 v_m$$

Ingemar
Ragnemalm
ingis@isy.liu.se

Beräkna allt i rätt domän

Matriserna kan beräknas *per ben*.

$$\begin{aligned} M_{mb} &= \prod M^{-1}_{ben,i} \\ M_{bm} &= \prod M_{ben,i} R_{anim,i} \\ M_{tot} &= M_{bm} M_{mb} \end{aligned}$$

Transformationen av vertex görs per vertex.

$$v' = M_{tot} \cdot v$$

Ingemar
Ragnemalm
ingis@isy.liu.se

I shader

Per ben-operationer: Görs på CPU

Per vertex-operationer: Görs i vertex shader

Matriserna kan skickas som *uniform*

Vikerna wi är per vertex, skickas som *attribute*

Vad är vinsten?

Mest uppenbar om vertexar är buffrade i VRAM
(display list eller VBO).

Projekt på skinning?

Massor av möjligheter!

Grundläggande skinning som del av ett projekt

Förbättrad skinning, undersök någon metod eller
jämför flera

Auto-generering av skelett från mesh (t.ex. via
sampling av mesh till griddata och
pixelskelettering)

Deformerbara kroppar (intro)

Deformerbara objekt utan ben?

- Tyg
- Skumgummi
- Cellplast
- Gelé
- Ballonger
- Lera
- Vatten

Många olika tekniker

- Förgenererade deformationer
 - Massa-fjäder-system
 - Finita elementmetoden
 - Formmatchning
 - Tryckmodellen
- Implicit modellering (level sets)
 - Punktbaserad animation

Förgenererade deformationer

Vanliga i spel (t.ex Midtown Madness)

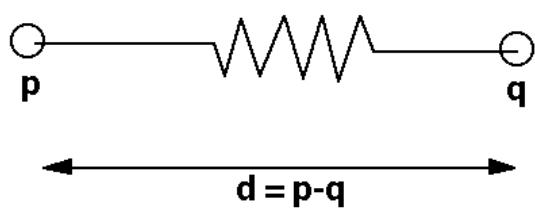
En ren designfråga.

Orealistiskt!

Ingemar
Ragnemalm
ingis@isy.liu.se

Massa-fjäder-system

En uppsättning punktmassor anslutna med fjädrar.



$$|dl| = |p-q|$$

$$\mathbf{f} = -k_s(|dl|-r)$$

$$\mathbf{f} = -k_s(|dl|-r) \cdot \frac{\mathbf{d}}{|dl|}$$

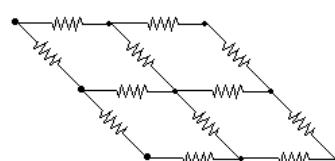
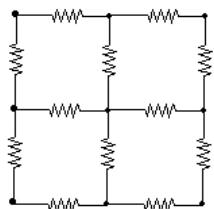
Dämpning kräver även en komponent som är hastighetsberoende.

Ingemar
Ragnemalm
ingis@isy.liu.se

Massa-fjäder-system

Många anslutningar krävs

Detta duger inte:



BÅDA diagonalerna brukar tas med.

Ingemar
Ragnemalm
ingis@isy.liu.se

Massa-fjäder-system

Bra i 2D, för tyg mm

Hyfsat för skumgummi etc, men beräkningstungt

Svårt att klara självkollisioner

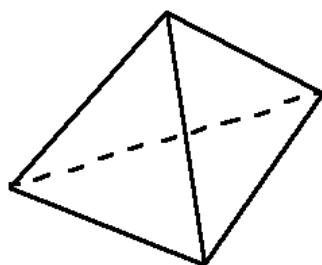
Kan vändas fel utan att det upptäcks

Stabilitetsproblem!

Finita elementmetoden

Bryt ner modellen i tetraedrar

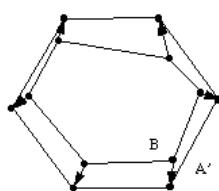
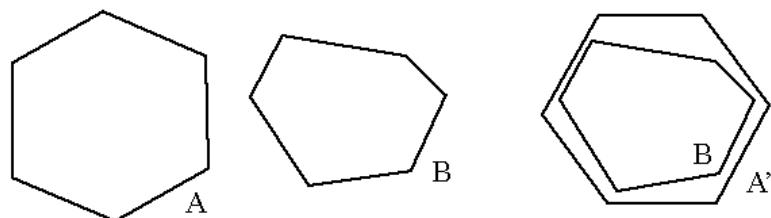
Bra metod, klarar självöverlappningar, inga problem med felaktiga tillstånd.



Formmatchning

Modellen attraheras av ett viloläge

Intressant metod, kan garanteras vara stabil

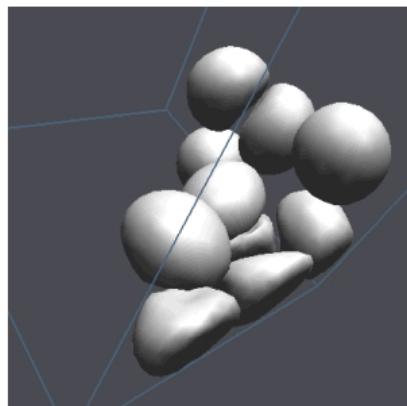


Tryckmodellen

Massa-fjädersystem som 2D-skal plus tryck för innehållet

Mycket bra simulering av

- Vattenballonger
- Celler

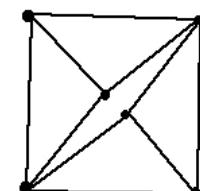
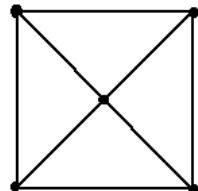
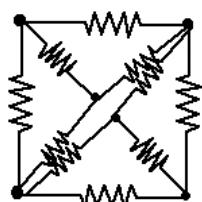
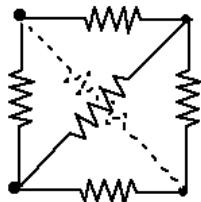


Implicit modellering Punktbaserad animation

Stora partikelsystem kan simulera vatten,
modellera mm mycket realistiskt.

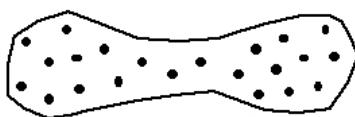
I 2D: iPhone-öl-demot!

Objekt som kan brytas, krossas mm



**Massa-fjädersystem:
Bökigt**

FEM: Ganska lätt



**Punktbaserat: Inbyggt
systemet**

Rendering av punktupppbuggda objekt

Partikelsystem med beroenden

- Rendera som separata billboards

Moln. Enklare vatten. "Luddigt".

- Marching cubes

**Vatten, lera mm. Ger distinkt yta, möjlighet till bra
reflektioner mm.**

Projekt inom deformable bodies

Brett och hett område. Många möjligheter.

Tyngdimering kanske i lättaste laget.

De flesta andra är av intresse.

- **Massa-fjäder:** Lätt att sätta upp, svårt att få det stabilt.
 - FEM: Rättfram och kapabel metod.
 - Formmatching: Stabil och trevlig.
 - Tryckmodellen: Bra för vissa typer av modeller.
- **Punktbaserad:** Beräkningskrävande. GPU-lösning önskvärd.

Inlämningsuppgifter

15. En ytvertex v_m påverkas av två ben med vikterna w_1 och w_2 . Benen kan variera rotationen men inte längden. Beskriv operationerna som behövs för att beräkna dess nya position v_m' . Uppdelning av transformationerna i konstanta och variabla delar skall framgå.

16. Jämför följande metoder för deformable bodies:

- massa-fjäder-modellen
- finita element-metoden
 - formmatching
- punktbaserade modeller

med avseende på

- självöverlapp
- stabilitet
- möjlighet att bryta isär