

Normal Meshes

Antonín Hildebrand [antonin@hildebrand.cz]

Zdroj: Igor Guskov, Kiril Vidimce, Wim Sweldens, Peter Schroder

Obsah přednášky

- **úvod**
- **popis normal mesh reprezentace**
 - **motivace**
 - **normal polylines**
 - **normal meshes**
- **konstrukce algoritmus**
 - **aproximace objektu pomocí normal mesh**
- **diskuse**
- **aplikace**

Bì žné reprezentace

- **závislost na øšeném problému**
- **závislost na implementaci v HW**
- **aproximace povrchu**

- **hranièní reprezentace**
 - - trojúhelníky, quady a jiná primitiva
- **objemové reprezentace**
 - - voxely, ...
- **ostatní reprezentace**
 - - analytický popis
 - - hierarchické pøístupy
 - - hybridní pøístupy

Trojúhelníková reprezentace

- hraničení reprezentace
- jednoduchost
- podpora v HW

- topologie 3D objektu je definována pomocí seznamu trojúhelníků
tj. informacích pozicích vrcholů a jejich propojení
- informace je uložena do seznamů:
 - vertex (x,y,z)
 - face (v1, v2, v3)
 - edge (f1, f2)

Motivace pro novou reprezentaci 1

- parametrizujeme povrch plochy pomocí tří funkcí $x(u,v)$, $y(u,v)$, $z(u,v)$
- p - spojitá plocha
 b - bod na povrchu
 r - tečná rovina v tomto bodě
- Diferenciální geometrie říká:
„ p může být na jistém okolí bodu b popsána jako funkce vzdálenost od tečné roviny r “

plochu tedy popíšeme skalární funkcí $h(u,v)$

Motivace pro novou reprezentaci 2

- geometrická informace v $h(u,v)$ je jednorozměrná
- změnou u,v se pohybujeme po povrchu
funkční hodnota určuje vzdálenost od tečné roviny
- trojúhelníková reprezentace určuje pozice vertexů
jako souřadnice ve 3D prostoru (x,y,z)
=> k popisu jsou potřeba tři veličiny

<0> v trojúhelníkové reprezentaci je do dat
„zamíchána“ i parametrizace
(neboli v x , y a z jsou zakódovány u , v a $h(u,v)$)

Zavedení Normal Mesh reprezentace

- cílem je vytvořit reprezentaci, které bude potřebovat k popisu 1 float na vertex (oproti 3f/v u trojúhelníkové reprezentace)
- tuto reprezentaci autoø nazvali Normal Meshes (dále jen NM)
- co je potřeba využít:
 - rozum .-)
 - ni jakou matematiku a diferenciální geometrii
 - multiresolution reprezentace
 - lokální okolíèka (local frames)

Co je to Normal Mesh ?

- je to hierarchická reprezentace plochy (3D objektu)
- geometrické koeficienty jsou definovány v lokálních okolích a mají pouze skalární složku
- skalární složka má smysl „vzálenost ve smì ru normály“
- geometrický popis objektu je „zhuštì n“ do skalární velièiny -> mohou být použity obecné metody pro zpracování signálu (například pro kompresi)

Normal Polylines 1 - definice

- křivka v rovině : $s(t) = (x(t), y(t))$, kde t je z $[0,1]$
- budeme ji aproximovat lomenou čarou
- značení: $I(p, q)$ - úsečka mezi body p a q
- standardní způsob jak aproximovat křivku je její vzorkování v bodech $s_{j,k}$

$$L_j = \bigcup_{0 \leq k < 2^j} I(s_{j,k}, s_{j,k+1}).$$

- při přechodu od L_j k L_{j+1} se zdvojnásobí počet vzorků
- mezi každé dva body původní lomene čáry vložíme nový vzorek

Normal Polylines 2 - aproximace

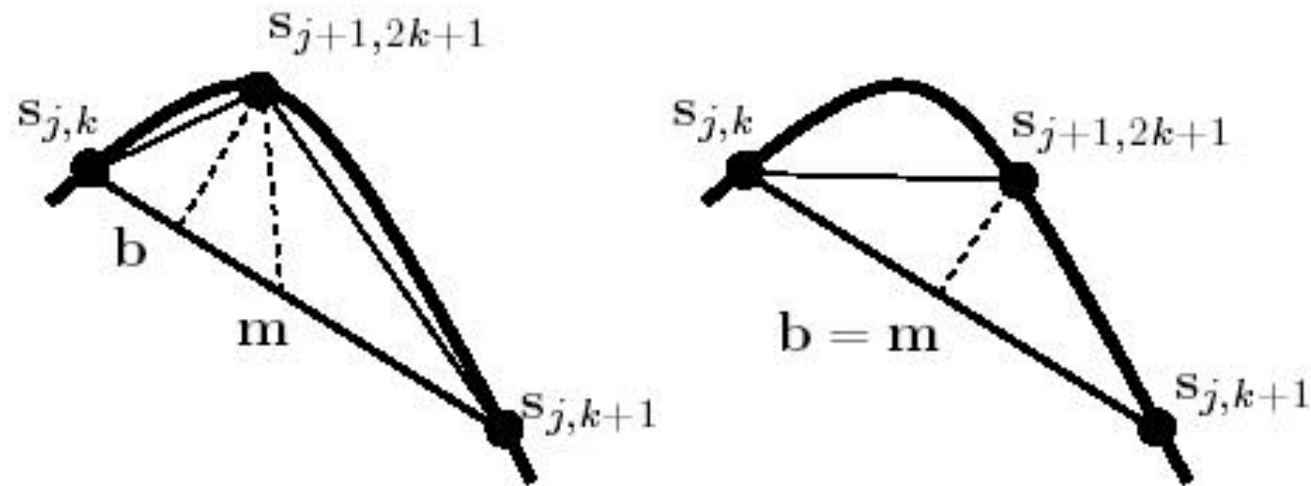


Figure 2: Removing one point $s_{j+1,2k+1}$ in a polyline multiresolution and recording the difference with the midpoint m . On the left a general polyline where the detail has both a normal and a tangential component. On the right a normal polyline where the detail is purely normal.

Normal Polylines 3 – normálnost

- mezi každé dva body původní lomene èáry vložíme nový vzorek
- nový vzorek by normálnì potøeboval na svoje urèení dvě souøadnice
- pokud nový vzorek vztáhneme k nì jakému predikovatelnému bodu m , můžeme urèit:
teèná složka = obsahuje parametrickou informaci
normální složka = obsahuje geometrickou informaci
- cílem je zadefinovat lomenou èáru tak, aby teèná složka byla vždy nulová => normální lomená èára
$\langle o \rangle$ pro midpoint predikci je to splnì no, když každý pøídávaný bod tvoøí se svými sousedy rovnoramenný trojúhleník

Normal Polylines 4 - konstrukce

- $\langle o \rangle$ naprostá vř třina obecných lomených èar není normální
- jak normální lomenou èáru postavit z křivky ?

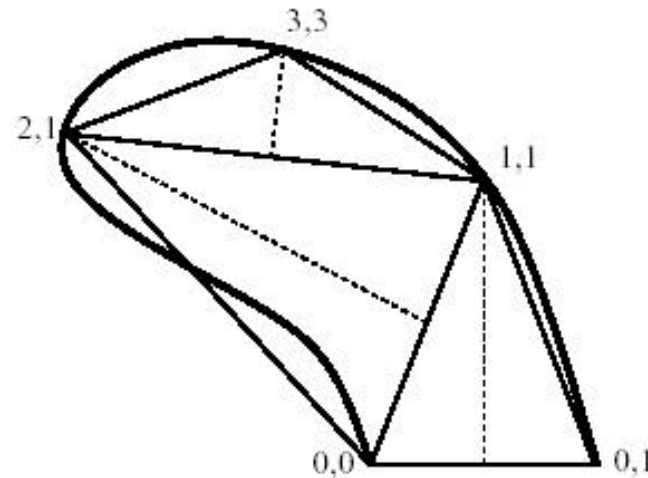


Figure 3: Construction of a normal polyline. We start with the coarsest level and each time check where the normal to the midpoint crosses the curve. For simplicity only the indices of the $s_{j,k}$ points are shown and only certain segments are subdivided. The polyline $(0,0) - (2,1) - (3,3) - (1,1) - (0,1)$ is determined by its endpoints and three scalars, the heights of the Isosceles triangles.

Normal Polylines 5 - shrnutí

- křivka může být aproximována normální lomenou čarou libovolně přesně (rekurzivní dělení)
- na zaznamenání každého bodu lomené čáry je potřeba jedna skalární hodnota (geometrická informace), vztažená k předchozímu hrubšímu stavu
- je možné použít i jiná schémata pro predikci základního bodu
- **<D1>**
Lomená čára je normální, pokud existuje posloupnost odebírání jejich bodů taková, že každý odebraný bod leží v normálním směru od jistého predikovatelného základního bodu. Normální směr a základní bod jsou plně určeny zbývajícími body křivky.

Normal Mesh 1 – zobecnění 2D případu

- přidáme další souřadnici a jsme ve 3D
- značení:
mesh: $M(P,K)$, kde P je seznam N vertexů ve 3D
 K je topologická informace (obecní)
- nechť M_j je hierarchie mesh objektů, tak že $P_j \supset P_{j+1}$
- při odebrání vertexu z $j+1$, musí tento bod ležet v normálním směru od jistého predikovatelného bodu b . Normální směr r a bod b jsou plně určeny stavem j .
- **<D2>**
Mesh je normální, pokud existuje posloupnost odebrání jeho vertexů taková, že každý odebraný vertex leží v normálním směru od jistého predikovatelného základního bodu. Normální směr r a základní bod jsou plně určeny zbývajících vertexy.

Normal Mesh 2 – (naivní) konstrukce

- **<o> obecný mesh není normální => provedeme retriangulaci**
- **konstrukci provedeme podobně jako v případě lomené čáry:**
 - 1. vyjdeme ze základní úrovně (základ mesh 0 - nejhrubší)**
 - 2. v každém kroku určíme základní bod a normální směr**
 - 3. vyhledáme průsečík mezi originálem a normálou**
mohou nastat 3 případy:
 - a) nenalezli jsme žádný průsečík**
 - b) našli jsme jeden průsečík**
 - c) našli jsme více průsečíků**
 - 4. v případě, že máme vybraný průsečík, vytvoříme další úroveň meshu**
- **na další straně rozebereme případy a zvolení vybraného průsečíku podrobněji**

Normal Mesh 3 – případy průsečíků

- a) žádný průsečík
=> normální varianta k danému meshi nejde sestavit
finální budeme řešit zvolením jiného vertexu na povrchu a
zaznamenáním více informací k tomuto vertexu

- b) jeden průsečík
=> ideální situace, vybereme ho jako nový bod

- c) více průsečíků
=> je nutné vybrat nejvhodnější z nich
naivní implementace, která bere první nalezený nebo volí
náhodně jeden z nalezených vede ke špatným výsledkům,
dojde k převrácení některých trojúhelníků a v objektu
vzniknou „díry“
- finální algoritmus musí být robustní v řešení případu a) a c)

Normal Mesh 4 – základ

- pod pojmem základ budu označovat nejhrubší mesh, ze kterého vychází naivní konstrukce
- jednoduchým pozorováním je, že tvar základu ovlivňuje efektivitu metody tedy počet průsečíků nutný na stejnou jemnou aproximaci
- proto bude snahou chytřejšího algoritmu zvolit vhodný základ na kterém normální reprezentaci vybuduje

Normal Mesh 5 – parametrizace 1

- rychlá a robustní hladká parametrizace je důležitý nástroj pro zrobustnění algoritmu
- úloha:
 - R – oblast meshe homomorfního s diskem
 - B – konvexní rovinná oblast
 - hledáme bijektivní zobrazení $u:R \rightarrow B$ s vlastností:
 - 1) na hranici $dR \rightarrow dB$ (po hranicích b_i žíme stejně rychle)
 - 2) minimalizuje funkcionál energie (natahujeme blánu)
- řešení existuje na dalším slajdu !

Normal Mesh 5 – parametrizace 2

- **metoda vychází z práce**

[10] FLOATER, M. S. Parameterization and Smooth Approximation of Surface Triangulations. *Computer Aided Geometric Design 14* (1997), 231–250.

- **u musí vyhovovat:**

$$u(\mathbf{p}_i) = \sum_{k \in \mathcal{V}(i)} \alpha_{ik} u(\mathbf{p}_k), \quad (1)$$

- **dá se řešit pomocí iterative biconjugate method z**

[12] GOLUB, G. H., AND LOAN, C. F. V. *Matrix Computations*, 2nd ed. The John Hopkins University Press, Baltimore, 1983.

- **vlastnosti: konvexita B + alfy jsou vždy pozitivní => no flipping**
- **při dobrém počátečním odhadu metoda konverguje rychle**

Algoritmus - úvod

- **веду robustní algoritmus, pracující v sedmi krocích**
 - 1. vybudování LOD úrovní meshe**
 - 2. postavení základní sítě kóvek**
 - 3. volba globálních vertexů**
 - 4. volba globálních hran**
 - 5. počáteční parametrizace**
 - 6. průsečíkování**
 - 7. narovnání parametrizace**
- **první čtyři kroky jsou přípravou pro robustní hledání průsečíků z naivní konstrukce**
- **body 6 a 7 se provádí cyklicky pro postupné zjemňování**
- **jako ilustrační model byla zvolena molekula viru HIV, jsou na ni vyznačeny jednotlivé fáze**

Algorithmus - molekula

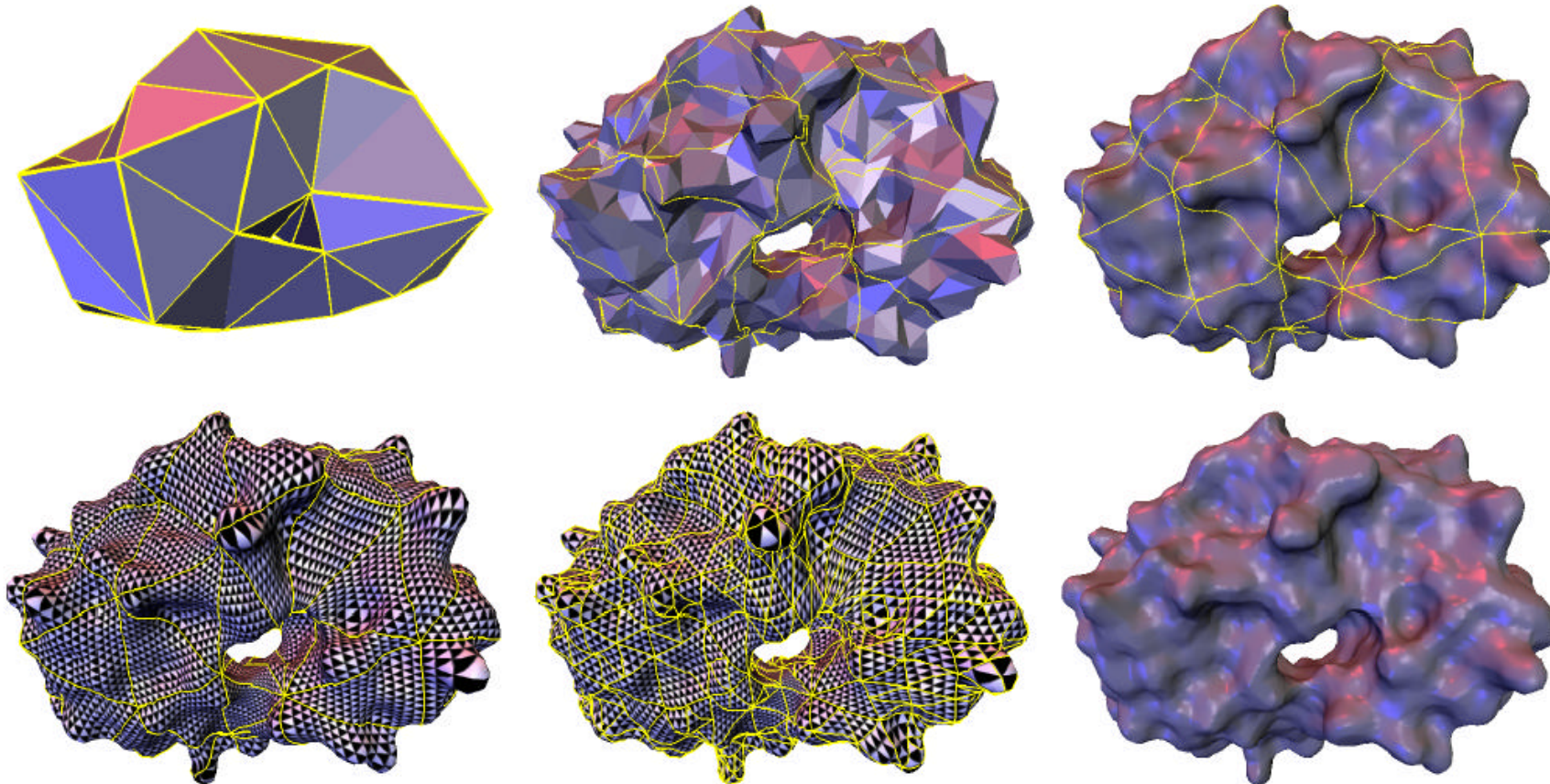
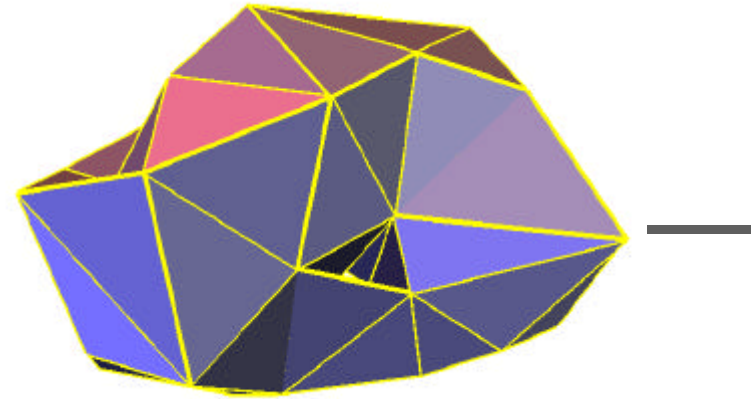


Figure 5: The entire procedure shown for the molecule model. 1. Base domain. 2. Initial set of curves. 3. Global vertex repositioning 4. Initial Parameterization 5. Adjusting parameterization 6. Final normal mesh. (HIV protease surface model courtesy of Arthur Olson, The Scripps Research Institute)

Algoritmus – krok 1



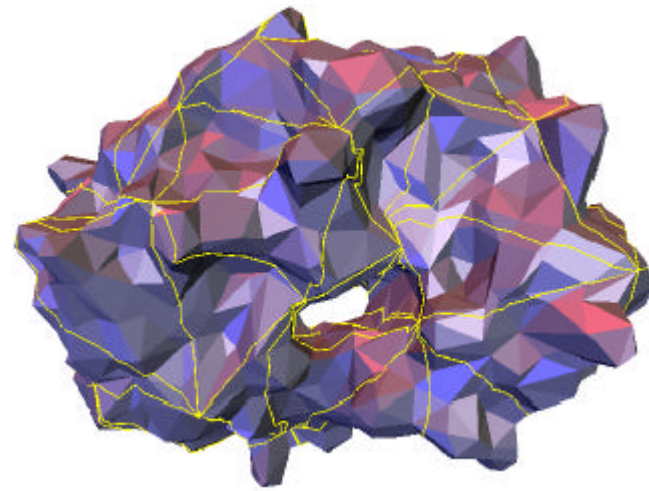
- vybudování LOD úrovní meshe

- **metoda:**

[11] GARLAND, M., AND HECKBERT, P. S. Surface Simplification Using Quadric Error Metrics. In *Proceedings of SIGGRAPH 96*, 209–216, 1996.

- **výstup:**
n zjednodušujících úrovní meshe
nejhrubší úroveň (mesh 0)
použijeme jako odhad základu
- základ bude jen prvním odhadem,
v kroku 3 algoritmu ho postupně zlepšíme

Algoritmus – krok 2



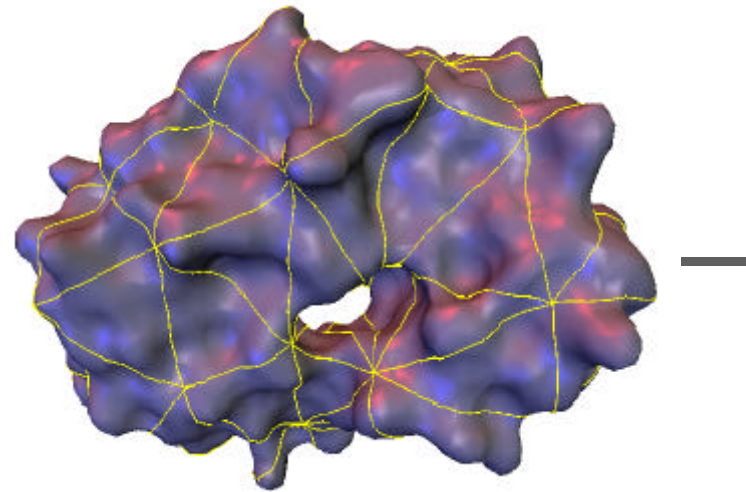
- postavení základní sítě kóvek
- cíl: spojit vertexy základu sítí vzájemně se neprotínajících kóvek a tuto síť propagovat do všech LOD úrovní

- metoda: MAPS parametrizace

[18] LEE, A. W. F., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. MAPS: Multiresolution Adaptive Parameterization of Surfaces. *Proceedings of SIGGRAPH 98* (1998), 95–104.

- MAPS dává bijektivní zobrazení pro 1-prstencové okolí vertexu a retriangulovanou oblasti po jeho odstranění
 - kóvky na každé LOD úrovni jsou definovány jako obrazy hran z mesh 0 v MAPS zobrazení.
- Pozor ne nutně kopírují hrany ve vyšších LODech.

Algoritmus – krok 3a



- volba globálních vertexů
- normal mesh je determinován základem, ze kterého normální reprezentace vychází
- pokusíme se základ z původního odhadu zlepšit
- vybereme vhodnou LOD úroveň (úroveň j) podle které budeme zlepšovat
- pro každý vertex základu zkusíme najít jeho vhodnější jšího náhradníka v j -té úrovni
- náhradníka hledáme v rámci trojúhelníkových plátů sdílejících zkoumaný vertex v j -té úrovni
- tento postup je možné iterovat, ale zkušenosti ukazují, že jeden přechod stačí

Algoritmus – krok 3b

- volba globálních vertexů

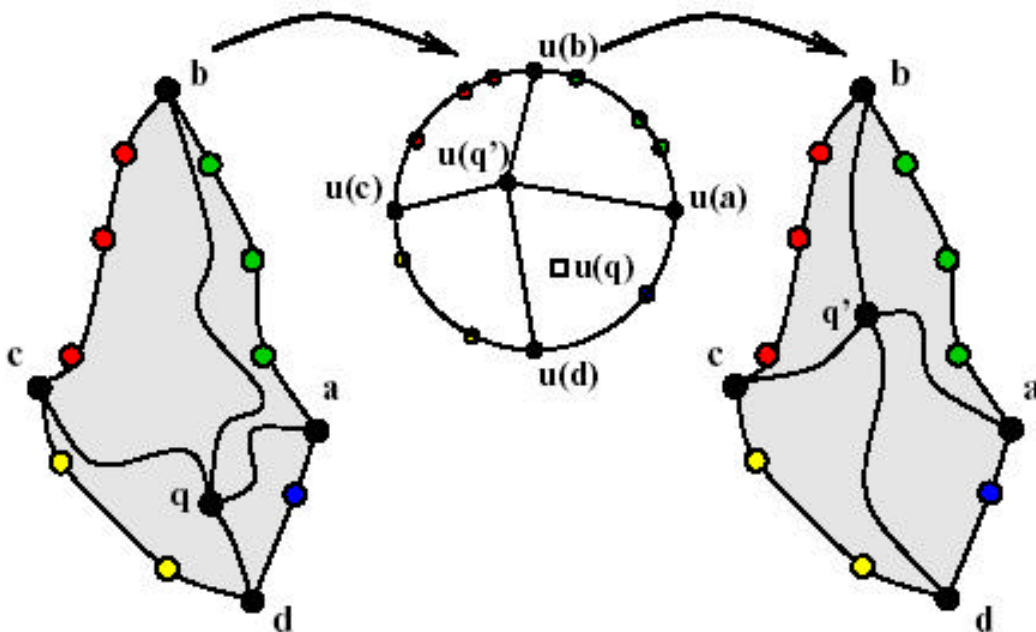
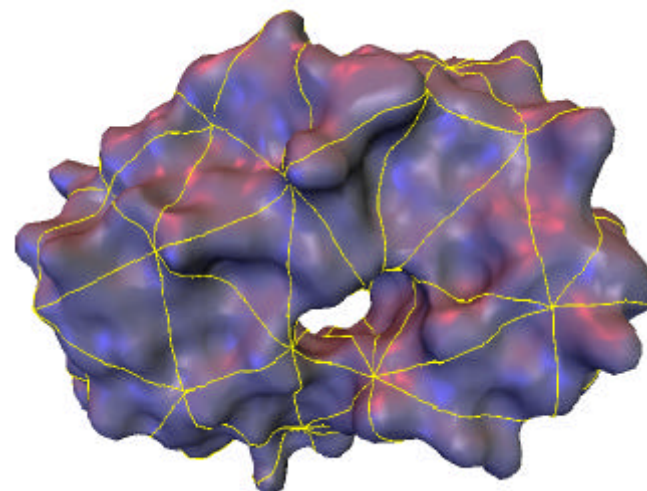
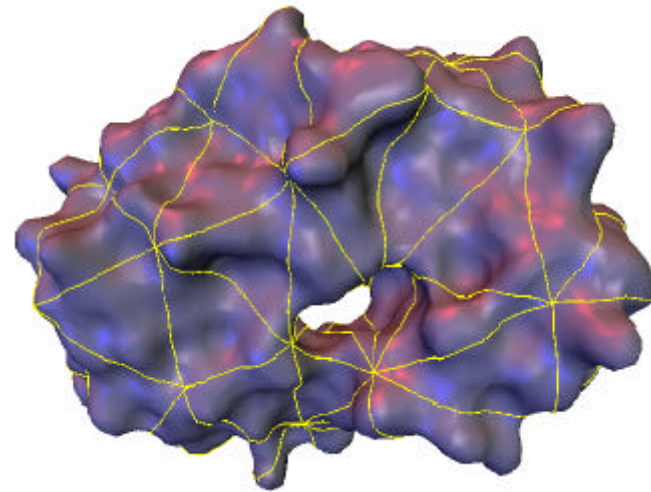


Figure 6: Base domain vertex repositioning. Left: original patches around \mathbf{c}_i , middle: parameter domain, right: repositioned \mathbf{c}_i and new patch boundaries. This is replaced with the vertex whose parameter coordinate are the closest to the center. The inverse mapping (right) is used to find the new position \mathbf{c}_i' and the new curves.

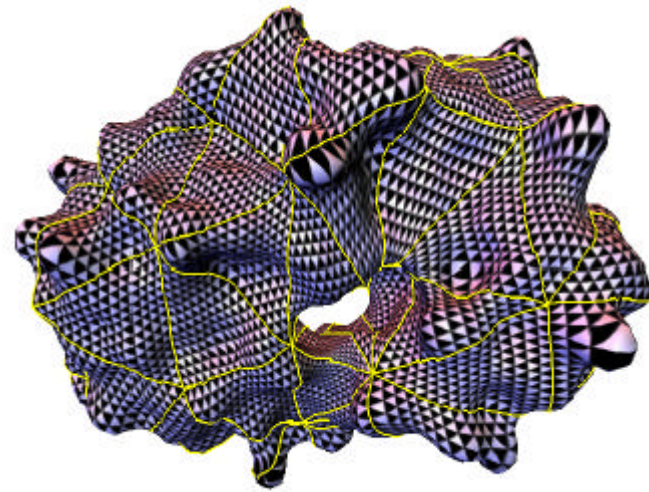
1. zobraz vertexy plátů na kruh
2. zvol vertex nejbliže středu kruhu jako náhradníka
3. náhradní křivky získkej inverzní transformací spojnic náhradníka s původními rohy plátů

Algoritmus – krok 4



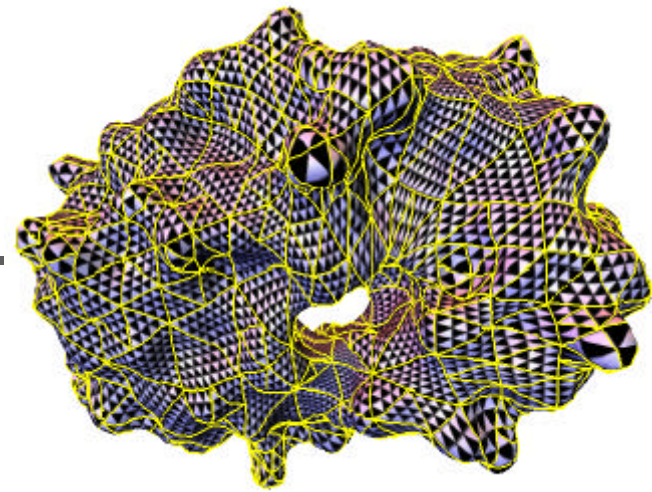
- volba globálních hran
- obraz hran základu na nejjemnější LOD úrovni bude později tvořit hranice plátů normal meshe
- chceme tyto hranice mít co nejhladší pro hladkou parametrizaci
- použijeme podobný postup jako v kroku 3:
- pro každou hranu základu, kterou chceme zlepšit
 1. vezmu pláty, které tuto hranu sdílejí
 2. tyto dva pláty namapuji na kosoúhelník s úhlopříčkou (tak, že zkoumaná hrana je orientována ve směru úhlopříčky)
 3. vzor úhlopříčky je naše hladká hrana

Algoritmus – krok 5



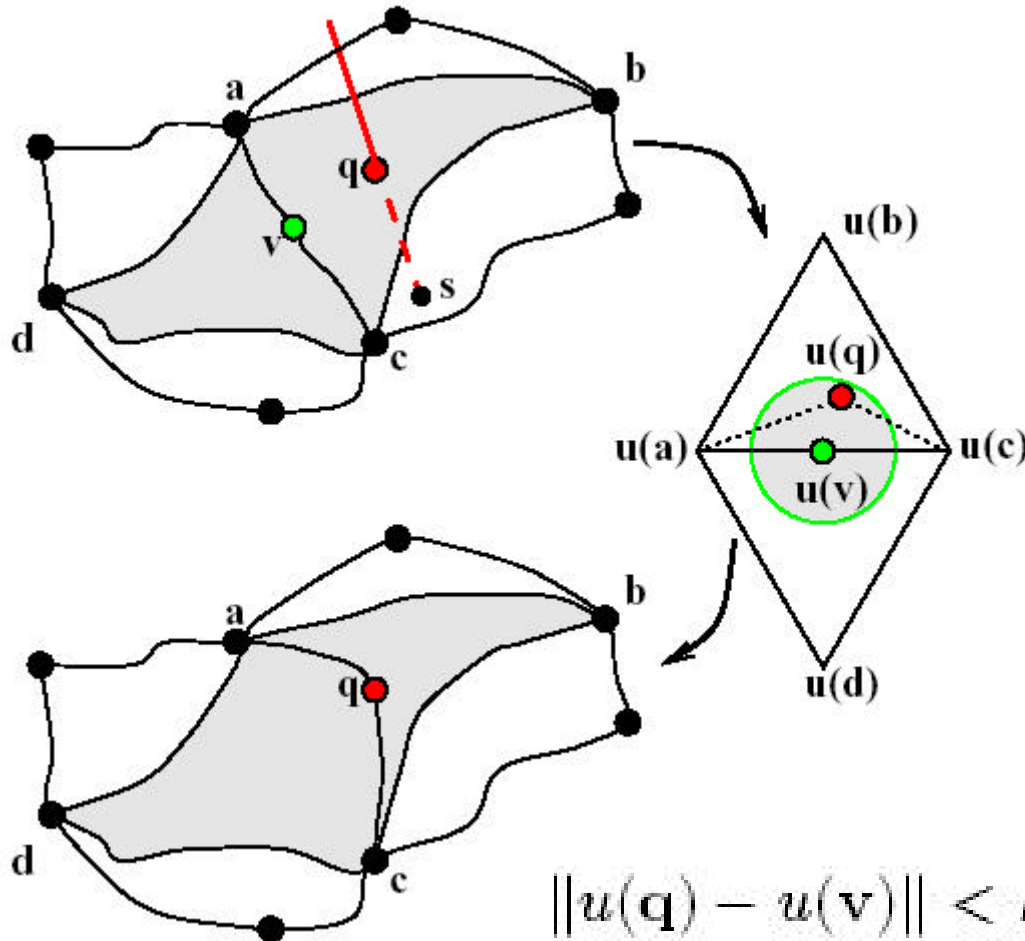
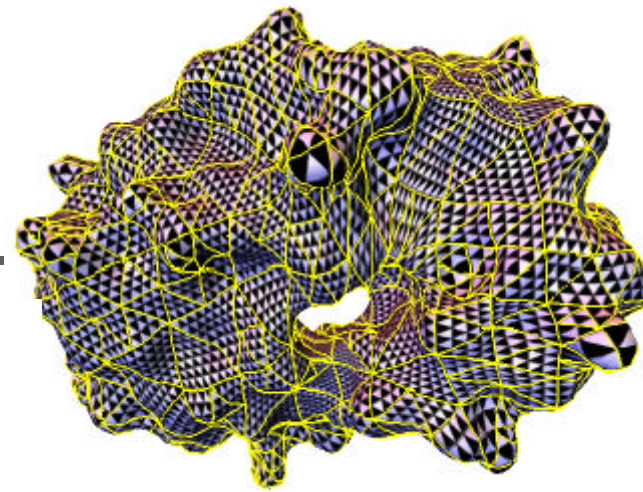
- **poèáteèní parametrizace**
- **máme zvolené globální vertexy a spojeny hladkými globálními hranami**
- **každý vzniklý patch parametrizujeme na trojúhelník
=> dostaneme hladkou parametrizaci celého povrchu**
- **pozn.: pro urychlení konvergence jako poèáteèní odhady parametrizace můžeme použít koeficienty z kroku 4**

Algoritmus – krok 6a



- průsečíkování
- v tomto kroku začínáme stavít reprezentaci ze základní úrovně
- v jednom cyklu, každá hrana j -té úrovně generuje nový vertex pro $(j+1)$ -ní úroveň
- postupujeme stejně jako v naivní konstrukci:
 1. určí základní bod b (jako predikci použij butterfly-subdivision)
 2. aproximuj normálu v b (z j -té úrovně)
 3. najdi průsečík s originálem
- 0 průsečíků => najdi alternativu (obraz středu hrany)
1 průsečík => vezmi ho
více průsečíků => použij parametrizaci pro zvolení toho pravého

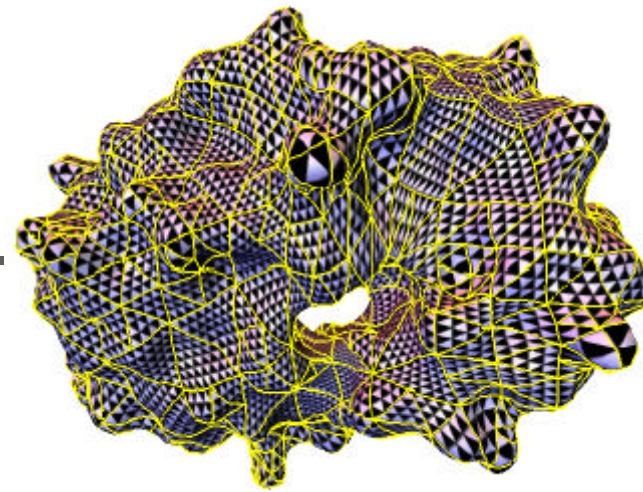
Algoritmus – krok 6b



1. zvol průsečík
nejblíže středu kruhu
v oblasti
parametrizace
2. pokud žádný
nepadne do kruhu
zvol obraz středu
oblasti parametrizace

$$\|u(\mathbf{q}) - u(\mathbf{v})\| < \kappa \|u(\mathbf{a}) - u(\mathbf{v})\|$$

Algoritmus – krok 7a



- narovnění parametrizace
- v každé aplikaci kroku 6 vzniká nový
- nový průsečík se stává v $(j+1)$ -ní úrovni novým vertexem, který byl „vytažen“ ze středu hrany
- cílem je oblast přemapovat tak, aby nový vertex měl parametrizaci $u(v)$ (nyní má $u(q)$)
- použijeme opět MAPS (viz se na slajd výše)
! problém s převrácením trojúhelníku
řešení: 1. bodem q veř lomenou čáru $u(a)$ - $u(q)$ - $u(c)$
2. inverzně ji transformuj na povrch
3. MAPSni $a(q)cb$ a $a(q)cd$ zvlášť

Algoritmus – krok 7b

- každá hrana generuje vertex
- potřebujeme získat síť hladkých křivek spojujících všechny vertexy
- staré křivky zůstanou, přidáme nové inverzním zobrazením ze zjemněného trojúhelníku v parametrickém prostoru

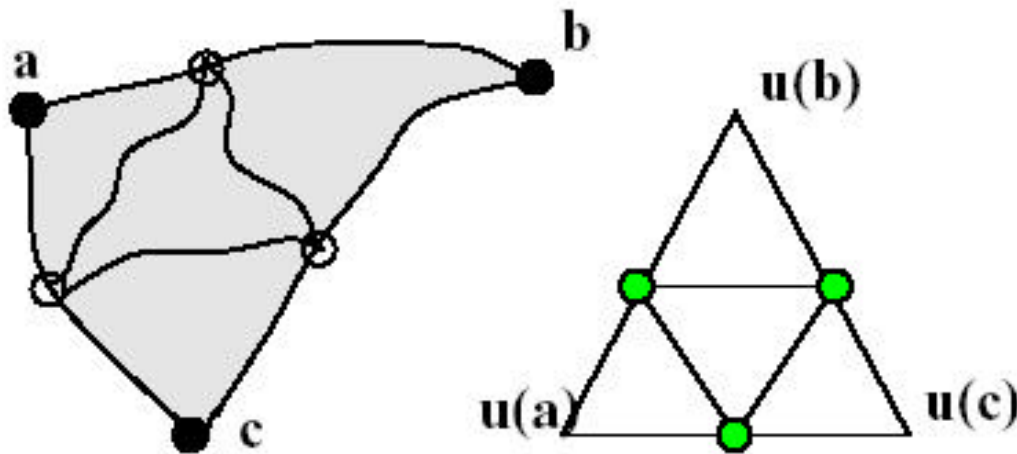
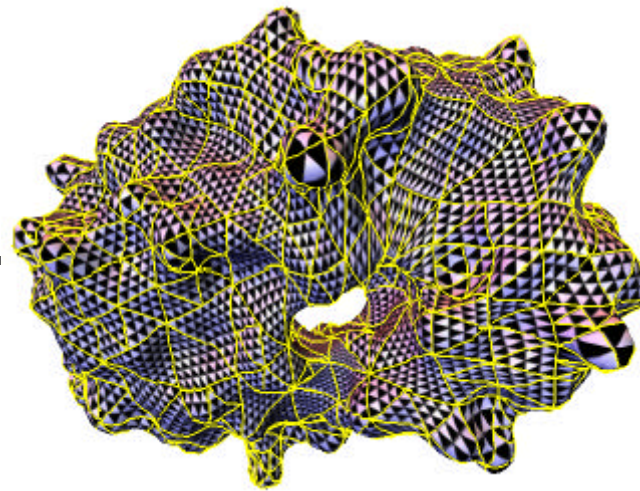


Figure 8: *Face split: Quadrisection in the parameter plane (left) leads to three new curves within the triangular patch (right).*

Algoritmus – diskuse

- průsečíkování je možné provádět adaptivně (dobrý odhad chyby dává například MAPS)
- nastavení parametru κ ovlivňuje počet nenormálních vertexů
- $\kappa = 0 \Rightarrow$ budujeme nenormální mesh založený na globální parametrizaci
- $\kappa = 1 \Rightarrow$ budujeme čistě normální mesh
- autoø experimentálně zjistili, že je dobré na začátku u hrubých LODů dát κ na 0,2 a při postupném zjemňování κ zvyšovat až na 0,6

Algoritmus - výsledek

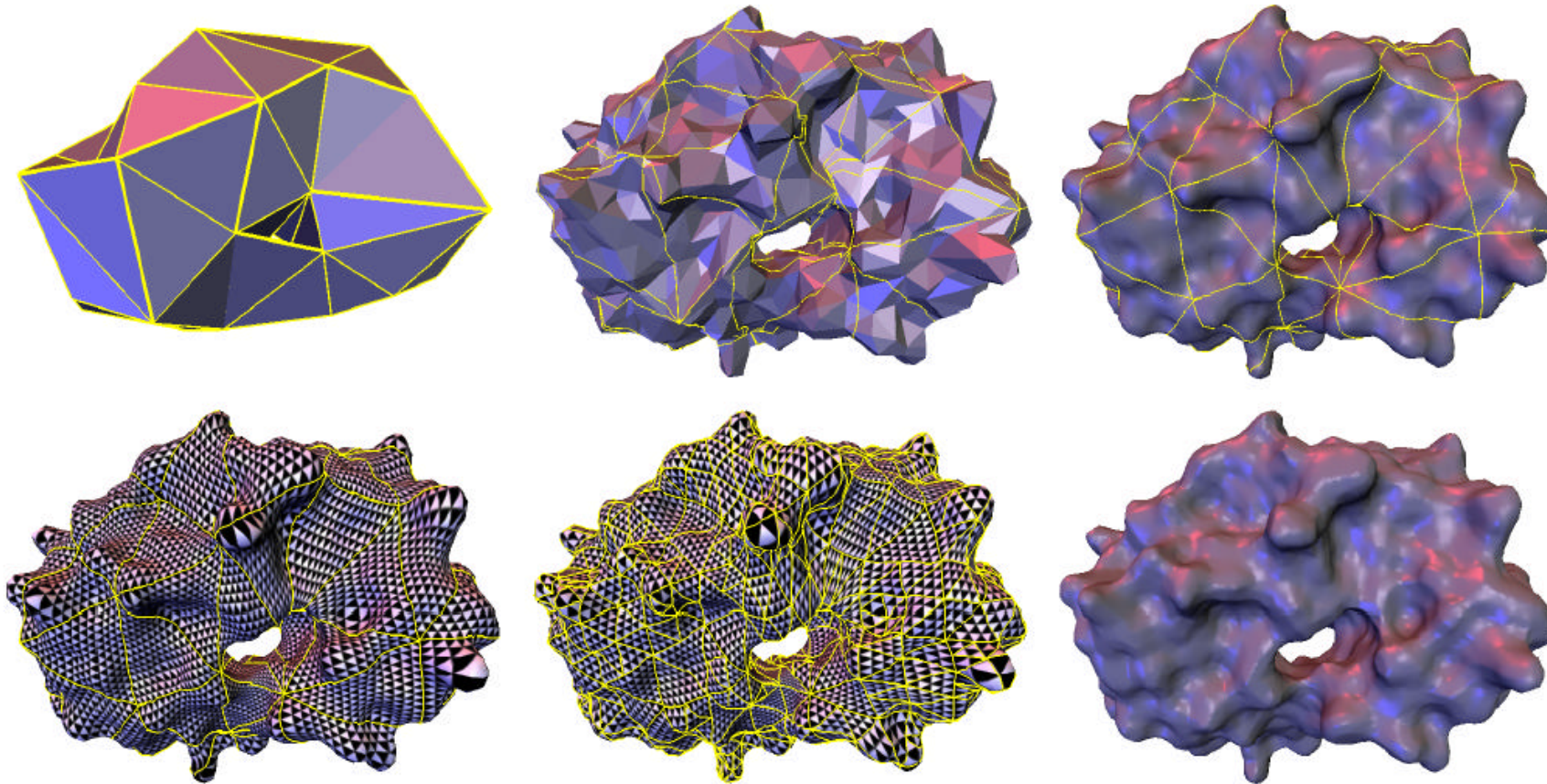


Figure 5: The entire procedure shown for the molecule model. 1. Base domain. 2. Initial set of curves. 3. Global vertex repositioning 4. Initial Parameterization 5. Adjusting parameterization 6. Final normal mesh. (HIV protease surface model courtesy of Arthur Olson, The Scripps Research Institute)

Algoritmus - tabulka

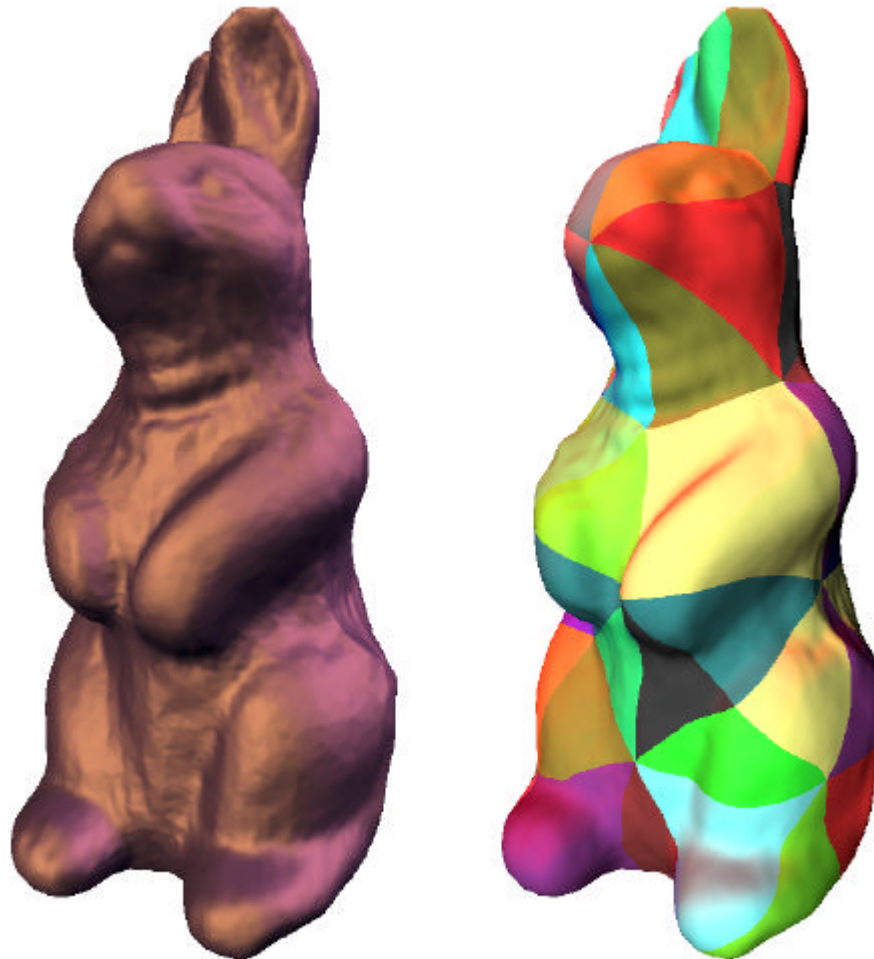
- srovnávací tabulka pro referenční modely

Dataset	Size	Base	Normal mesh size	Not normal (%)	% L^2 error	Time (min)
Feline	49864	156	40346	729 (1.8%)	.015	4
Molecule	10028	37	9521	270 (2.8%)	.075	1.5
Rabbit	16760	33	8235	196 (2.4%)	.037	2
Torus3	5884	98	5294	421 (8.0%)	.03	3
Skull	20002	112	25376	817 (3.2%)	.02	2.5
Horse	48485	234	59319	644 (1.1%)	.004	6.8

Table 1: *Summary of normal meshing results for different models. The normal mesh is computed adaptively and contains roughly the same number of triangles as the original mesh. The relative L^2 errors are computed with the I.E.I.-CNR Metro tool. The times are reported on a 700MHz Pentium III machine.*

Aplikace

- **Kompresa**
- **Filtrování**
- **Texturování**
- **Rendering**



Závěr

- **Normal Mesh je multi-resolution datová struktura**
- **Slouží k efektivní reprezentaci 3D objektů**
- **Tato reprezentace je vhodná pro mnoho aplikací jako komprese, texturování a hw-rendering**

- **A co dál ?**
 - **co tak použít aproximační dělení povrchu ? (místo butterfly)**
 - **co tak vše zobecnit do vyšších prostorů ? (M-dimenzionální manifold v N dimenzionálním prostoru bude potřebovat N-M skalárů na vertex)**
 - **co tak prozkoumat klasické kompresní schémata na normální koeficienty ?**

Reference

GUSKOV, I., VIDIMČE, K., SWELDENS, W., AND SCHRÖDER, P. Normal meshes. *to appear in Proceedings of SIGGRAPH 2000* (2000).

