

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Dynamické změny a editování triangularizovaných geometrických modelů**

Plzeň, 2007

Ondřej Kvasnička

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 21. května 2007

.....  
Ondřej Kvasnička

## **Dynamic modifications and editing of triangulated geometric models**

**Abstract.** The main goal of this diploma thesis is to evaluate suitability of subdivision surfaces for their application in interactive mesh modeling. Great attention is paid to natural ability of subdivision surfaces to represent multiresolution hierarchy of geometric details, and use of this property during mesh manipulation on different levels of details. The modeling tool built as a part of this work offers a basic set of editing operations, including simple copying of geometric details that were previously added to the model and their consecutive pasting to selected parts of the surface.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Reprezentace povrchových modelů</b>	<b>3</b>
2.1	Polygonová síť . . . . .	4
2.2	Parametrické plochy . . . . .	4
2.3	Dělené povrchy . . . . .	5
<b>3</b>	<b>Modelování s dělenými povrchy</b>	<b>7</b>
3.1	Princip dělených povrchů . . . . .	7
3.2	Konkrétní dělicí schémata . . . . .	11
3.3	Catmull-Clarkovy dělené povrchy . . . . .	12
3.4	Limitní poloha bodu a tečná rovina plochy . . . . .	14
3.5	Adaptivní dělení . . . . .	16
3.6	Víceúrovňové modelování . . . . .	17
<b>4</b>	<b>Výběr modelovacích technik</b>	<b>20</b>
4.1	Změna topologie dělených povrchů . . . . .	20
4.2	Změna geometrie dělených povrchů . . . . .	21
4.3	Víceúrovňové modelování . . . . .	22
4.4	„Copy & Paste“ operace . . . . .	22
<b>5</b>	<b>Navržené řešení</b>	<b>24</b>
5.1	Programová implementace . . . . .	24
5.2	Datové struktury . . . . .	24
5.2.1	Požadavky na vlastnosti vstupní řídicí sítě . . . . .	25
5.2.2	Třída <i>QTreeMesh</i> . . . . .	25
5.2.3	Třída <i>OneRing</i> . . . . .	26
5.3	Podpora pro adaptivní dělení . . . . .	27
5.4	Triangulace adaptivně rozdělené sítě . . . . .	28
5.5	Podpora pro víceúrovňové modelování . . . . .	29
5.6	Implementované modelovací operace . . . . .	32
5.6.1	Manipulace s vrcholy děleného povrchu . . . . .	32
5.6.2	Odstranění čtyřúhelníků řídicí sítě . . . . .	33
5.6.3	„Vytažení“ čtyřúhelníku řídicí sítě . . . . .	35
5.6.4	Operace typu „Copy & Paste“ . . . . .	35

<b>6</b>	<b>Výsledky</b>	<b>38</b>
6.1	Přidání detailů do existujícího modelu . . . . .	38
6.2	Modelování na úrovni vstupní řídicí sítě . . . . .	40
6.3	Rychlé přidání detailů pomocí metody „copy & paste“ . . . . .	40
<b>7</b>	<b>Závěr</b>	<b>44</b>
	<b>Literatura</b>	<b>46</b>
<b>A</b>	<b>Definice použitých termínů</b>	<b>48</b>
<b>B</b>	<b>Uživatelský manuál</b>	<b>49</b>
B.1	Orientace ve výhledech tažením kurzoru myši . . . . .	51
B.2	Ovládání klávesnicí a myši . . . . .	52

# Kapitola 1

## Úvod

Jednou ze širokých oblastí nasazení prostorové počítačové grafiky jsou interaktivní modelovací nástroje. Takové programové vybavení je používáno v široké škále odvětví lidské činnosti, zejména pak při výtvarném designu, při návrhu a konstrukci v různých technických oborech a také v různých odvětvích zábavního průmyslu. Z tohoto hlediska můžeme vymezit dvě kategorie. První jsou editory se zaměřením na technické aplikace, kde jsou vyžadovány zejména prostředky pro přesné modelování a nezděná musí modelované objekty splňovat různé požadavky týkající se jejich geometrických vlastností, jakými jsou např. křivost, geometrická spojitost a podobně. Do druhé kategorie patří ty programy, jejichž výstupem jsou modely určené třeba pro zmíněný zábavní průmysl. Zde hledáme spíše takové nástroje, s jejichž pomocí by uživatel co nejjednodušeji a co nejrychleji dokázal svoji myšlenku převést do konečné podoby. K tomu by mu měl pomoci co nejširší výběr nástrojů, které sice nutně nemusí umožňovat geometricky přesné modelování, ale upravovaný model by stále měl mít takové vlastnosti, jako je vizuální hladkost (tam, kde je to žádoucí) a samozřejmě absence nežádoucích artefaktů a děr ve výsledné polygonové síti modelu.

Interaktivní vytváření a upravování geometrických modelů je oblastí počítačové grafiky, ve které stále vzniká mnoho nových metod a přístupů k problému. Většina z nich se omezuje na stále nejrozšířenější 2D vstupní zařízení (obrazovka, myš, tablet) a proto musí mimo jiné nutně řešit otázku, jakým způsobem převádět volbu pozice a směru pohybu kurzoru v 2D prostoru na ekvivalentní polohu a směr v 3D prostoru. Společným cílem při vytváření všech typů interaktivních modelovacích editorů je vytvoření takového nástroje, který umožní uživateli bezprostřední kontrolu nad tvarem upravovaného modelu bez toho, aby musel znát detaily týkající se použitého typu datové reprezentace modelu a implementace použitých modelovacích technik.

Cílem práce je navrhnout a vytvořit modelovací nástroj, který svým charakterem bude patřit do druhé ze dvou výše uvedených kategorií. Vzhledem k obecnému názvu práce bylo nutné konkretizovat zadání a určit typ použité reprezentace modelů. Na doporučení vedoucí práce jsou použity dělené po-

vrchy, později byly jako konkrétní varianta zvoleny Catmull-Clarkovy dělené povrchy. Hlavní důraz je kladen na prozkoumání vlastností této reprezentace a určení vhodnosti jejího použití v interaktivním modelování, která pak bude ověřena v praxi. Součástí řešení je programová knihovna obsahující třídy pro práci s Catmull-Clarkovými dělenými povrchy určenými čtyřúhelníkovou řídicí sítí. Vytvořený interaktivní editor umožňuje provádění základních operací s tímto typem povrchů. Významnou vlastností dělených povrchů je podpora pro víceúrovňové modelování, která rozšiřuje množinu použitelných editačních operací a jejíž implementace je součástí vytvořené programové knihovny.

## Struktura dokumentu

V následující kapitole jsou pro úplnost zmíněny kromě dělených povrchů i další často používané typy povrchových reprezentací modelů. U každého jsou uvedeny nejdůležitější vlastnosti, které ovlivňují vhodnost jeho použití pro potřeby interaktivního modelování. Každý typ reprezentace totiž nastavuje určité mantinely, které vymezují množinu použitelných editačních operací. Třetí kapitola se věnuje obecným vlastnostem dělených povrchů a podrobněji popisuje námi použitou variantu – Catmull-Clarkovy dělené povrchy. Na závěr teoretické části dokumentu jsou s ohledem na použitý typ povrchové reprezentace popsány existující přístupy k modelování. Za touto první, teoretickou částí následuje implementační část, ve které je popsán způsob implementace použitých datových struktur a podporovaných editačních operací. Text uzavírá část diskutující dosažené výsledky. V přílohách je k dispozici seznam definic nejdůležitějších termínů použitých v tomto textu a uživatelský manuál popisující práci s vytvořeným programem.

## Kapitola 2

# Reprezentace povrchových modelů

Abychom mohli efektivně pracovat s povrchovými geometrickými modely, potřebujeme zvolit vhodné datové struktury, kterými budou modely reprezentovány v paměti počítače. Hledáme co nejjednodušší strukturu, pro kterou budeme zároveň schopni navrhnout co nejjednodušší algoritmy, a která nám umožní rychle získat všechny potřebné informace o reprezentovaném modelu. Kromě námi použitých dělených povrchů jsou v této kapitole pro úplnost zmíněny i jiné typy reprezentací povrchových modelů, které by mohly být použity v oblasti interaktivního modelování.

Každý typ povrchové reprezentace geometrických modelů můžeme zařadit do jedné ze dvou hlavních skupin – jsou jimi *explicitní* (také *parametrické*) povrchové reprezentace a *implicitní* povrchové reprezentace [4, 20]. Zmíníme zde i implicitní typy, protože i při implicitním modelování můžeme v určité fázi převést objekt z implicitní reprezentace do explicitní a pracovat s ním dále třeba jako s polygonovou sítí.

Implicitní povrchové reprezentace definují model hranicí mezi „vnitřkem“ a „vnějškem“ modelu. Tato hranice odpovídá povrchu geometrického modelu a je tvořena množinou  $\mathcal{S}$  všech bodů  $x$ , pro které je splněna rovnice v implicitním tvaru  $F(x) = 0$ , tedy  $\mathcal{S} = \{x \in \mathbb{R}^3 \mid F(x) = 0\}$ . Pro body uvnitř modelu platí  $F(x) < 0$ , analogicky pro body vně modelu platí  $F(x) > 0$ . Skutečné objekty vytvářené v implicitní reprezentaci jsou většinou složeny z více jednoduchých objektů, příkladem základního objektu je třeba koule. Tvary těchto stavebních součástí se liší podle konkrétního typu implicitní reprezentace a metody jejich generování.

Typické použití implicitních ploch nalezneme třeba v CSG (*Constructive Solid Geometry*) modelování [13], kde se z jednoduchých objektů typu koule, kvádrů nebo válců vytváří s použitím boolovských operací (průnik, sjednocení, rozdíl) složitější objekty. V tomto případě je jednou z výhod oproti explicitním reprezentacím neexistence samoprůniků po převedení z implicitní reprezentace na explicitní – i když je model v implicitní formě tvořen více než jedním jednoduchým objektem, po jejich zkombinování dostáváme opět jediný objekt s jasně určeným povrchem, který nemá samoprůniky [4]. Po převedení na explicitní



formu (např. triangulací) zůstává tato vlastnost zachována.

Z hlediska použití v interaktivním modelovacím nástroji má implicitní reprezentace také několik nevýhod. Hned při zobrazování implicitně definovaných objektů narazíme na nutnost převodu implicitní reprezentace na explicitní, protože grafická karta pracuje s povrchy explicitně definovanými pomocí množiny trojúhelníků. To může mít velký dopad na plynulost interaktivní práce s modelem – přílišná výpočetní náročnost triangulace implicitně definovaného povrchu by se mohla negativně projevit na délce doby mezi okamžikem provedení nějaké úpravy modelu a vlastním zobrazením změn.

Ovšem vzhledem k tomu, že cílem této práce není vytvořit nástroj, kterým bude možné vytvářet nové modely od základu, ale pouze editor, kterým bude možné provádět úpravy již existujících modelů, přestává být použití implicitní reprezentace objektů rozumnou volbou. Prvním důvodem je malá dostupnost již vytvořených implicitně definovaných modelů, se kterými by se v editoru dalo pracovat, druhým důvodem je fakt, že výše popsané výhody implicitní reprezentace se týkají spíše procesu vytváření nových modelů (viz [14]), ale už méně provádění úprav na existujících modelech.

Ve výsledku nevýhody implicitní datové reprezentace z hlediska použití v interaktivním editoru v našem případě převažují nad výhodami a volba explicitního typu reprezentace, konkrétně dělených povrchů, se z tohoto pohledu jeví jako rozumnější. V následujícím seznamu jsou kromě dělených povrchů uvedeny ještě dvě další typické varianty explicitní reprezentace povrchů – polygonové sítě a parametrické plochy. S oběma z nich sdílí dělené povrchy několik společných vlastností.

## 2.1 Polygonová síť

Klasická polygonová síť je nejuniverzálnějším typem povrchové reprezentace bez ohledu na to, zda mluvíme speciálně o trojúhelníkové síti nebo o síti polygonů s libovolným počtem hran. Dokáže dobře zachytit změny tvaru na jakékoli úrovni detailu díky tomu, že může být libovolně hustá, ale s přibývajícím množstvím detailů v modelu může počet polygonů rapidně růst a s tím roste i datová náročnost. Neposkytuje také bez dalších rozšíření příliš možností pro komplexní manipulace s objektem v situacích, kdy chceme uživateli dát možnost provádět úpravy ve velkém měřítku, alespoň na úrovni ekvivalentní ke změně tvaru parametrických ploch manipulací s jejími řídicími body (viz níže). S výjimkou polygonových sítí tvořených výhradně trojúhelníky může být také problémem to, že jednotlivé polygony obecně nemusí být planární.

## 2.2 Parametrické plochy

Parametrické plochy (konkrétně se velmi často používají Bézierovy bikubické plochy) jsou vhodné pro modelování velkých hladých úseků. Vytváření složitěj-

ších objektů navazováním jednotlivých parametrických ploch se někdy nazývá *plátování*, jednotlivé plochy pak nazýváme *pláty*. Pro vykreslení jednoho plátu nám stačí specifikovat poměrně málo řídicích parametrů, úprava jeho tvaru manipulací s řídicími body je pro uživatele intuitivní a nezanedbatelnou výhodou je i existující hardwarová podpora umožňující implementovat rychlé vykreslování parametrických ploch.

Nejsou ovšem dobře použitelné pro zachycení široké škály detailů v rámci jednoho objektu. Každým řídicím bodem je totiž třeba pohybovat zvlášť a s rostoucím úrovní detailů nutně roste i množství plátů potřebných pro jejich zachycení a s ním samozřejmě počet řídicích bodů. Práce s modelem pak přestává být efektivní. Množství parametrických ploch různé velikosti navíc přináší další problémy v situaci, kdy se snažíme o udržení hladkého navázání na hranicích mezi jednotlivými pláty. U parametrických ploch je navíc nutné, aby síť řídicích bodů tvořila pravidelnou mřížku. To znamená, že pokud chceme přidat nové řídicí body, musíme vždy přidat minimálně jednu celou řádku nebo sloupec řídicích bodů. Výjimku tvoří T-spline povrchy [12], které umožňují přidávat do sítě i jednotlivé body bez nutnosti udržovat pravidelnou mřížkovou strukturu řídicí sítě, ovšem za cenu použití složitějšího matematického aparátu.

V současnosti jsou standardem v oblasti parametrického modelování NURBS plochy (*Non-uniform Rational B-spline Surfaces*) [6, 20]. Jejich nevýhodou je, že s jejich pomocí nelze vymodelovat objekt s libovolnou topologií – samostatná NURBS plocha může mít topologii roviny, válce nebo toroidu (stejně jako jakákoliv jiná parametrická plocha). Všechny ostatní objekty s jinou topologií musí být vytvořeny kombinací více NURBS ploch (podobné výše uvedenému plátování), přičemž složitější objekty jsou často tvořeny ořezanými (*trimmed*) NURBS plochami. S ořezáváním NURBS ploch jsou ale spojeny dvě nevýhody – je výpočetně náročné a náchylné k numerickým chybám [6].

## 2.3 Dělené povrchy

Kombinaci dobrých vlastností parametrických ploch a polygonových sítí nabízí dělené povrchy (*subdivision surfaces*). Pro potřeby modelování představují ideální nástroj, který spojuje hlavní přednost parametrických ploch – konstrukci velkých hladkých ploch složených z jednotlivých plátů a schopnost polygonové sítě zachytit libovolný tvar modelovaného objektu v jakémkoliv detailu. Jeden dělený povrch dokáže zachytit objekt s libovolnou topologií (podobně jako klasická polygonová síť), takže odpadá potřeba skládání složitějších modelů z více částí jako u parametrických ploch. Díky jejich charakteru jsou vhodné pro aplikaci ve víceúrovňovém modelování (*multiresolution modeling*). S dělenými povrchy můžeme nakládat buď jako s polygonovou sítí, nebo jako s parametrickými plochami (díky podobným vlastnostem manipulace s vrcholy děleného povrchu a manipulace s řídicími body parametrické plochy). Tyto dva přístupy přitom můžeme používat současně, vždy tak, jak je to pro nás v dané situaci výhodnější.

Následující kapitola se podrobněji věnuje obecným vlastnostem dělených povrchů, později se zaměřuje na námi použitou variantu (Catmull-Clarkovy dělené povrchy) a vhodnost jejich použití v interaktivním modelování.

# Kapitola 3

## Modelování s dělenými povrchy

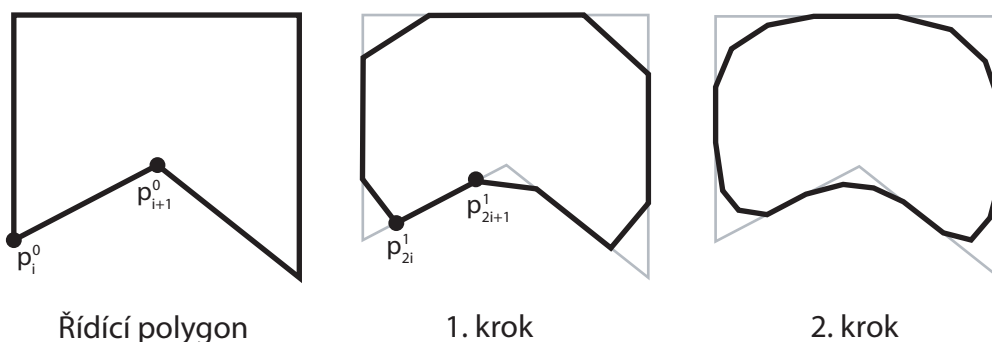
Jak již bylo výše uvedeno, dělené povrchy spojují hlavní přednosti parametrických ploch a polygonových sítí. V následujícím seznamu je podrobněji zmíněno několik nejdůležitějších vlastností obecně platných pro dělené povrchy, které nás budou zajímat v souvislosti s jejich využitím při úpravě a modelování povrchových modelů.

- *Invariance k lineárním transformacím* – tato vlastnost zajišťuje, že transformace, které aplikujeme na síť řídicích bodů ještě před jejím rozdělením, budou mít na výslednou podobu rozdělené sítě stejný vliv, jako kdybychom tytéž transformace aplikovali na samotnou síť později, až po jejím rozdělení.
- *Lokalita změn* – díky lokálnímu charakteru konstrukce dělených povrchů má každý řídicí bod vliv jen na určité malé okolí. To umožňuje provádět uživateli lokální úpravy bez toho, aby se změna projevila na celém modelu. Tato vlastnost je také důležitá, pokud např. chceme použít *level-of-detail* algoritmy nebo dělit plochu adaptivně podle kritérií, která zohledňují např. odchylku v daném místě od limitní plochy.
- *Libovolná topologie vstupní sítě* – ačkoliv konkrétní dělicí schémata mohou mít určitá omezení týkající se vstupní polygonové sítě, obecně lze použít síť libovolné topologie s polygony s libovolným počtem stran.
- Dělené povrchy *mají charakter jak polygonové sítě, tak i parametrických ploch* a díky tomu na ně můžeme aplikovat jak operace používané u polygonových sítí, tak operace používané u parametrických ploch.

### 3.1 Princip dělených povrchů

Postup vytváření dělených povrchů postupným zjemňováním vstupní hrubé sítě lze pro začátek demonstrovat na známém Chaikinově algoritmu pro dělení křivky tvořené přímkovými segmenty. V našem příkladu na obrázku 3.1 máme

na začátku uzavřenou křivku (řídící polygon). V každém kroku jsou v každém segmentu křivky vytvořeny dva nové body, které jsou spojeny, a vznikne tak nová křivka, tvořená dvojnásobným počtem bodů. Tento postup znovu aplikujeme na takto vzniklou křivku, až dosáhneme požadované hladkosti.



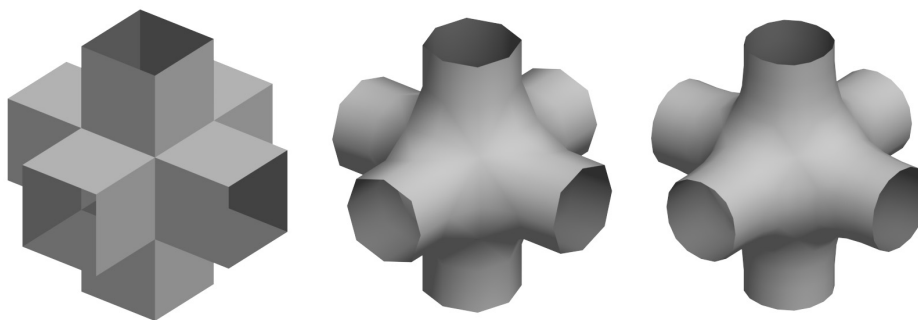
Obrázek 3.1: Chaikinův algoritmus

Dělený povrch popisujeme vstupní sítí řídicích bodů obecně s libovolnou topologií. Základní myšlenkou při její konstrukci je opakované zjemňování hrubé vstupní polygonové sítě (tzv. *řídící sítě*) podle konkrétního *dělicího algoritmu*, přičemž takto dělená síť se blíží ke známé limitní ploše. Postupného zjemňování sítě je docíleno vkládáním nových bodů při každém dělicím kroku a případně přepočítání pozic bodů vytvořených v předchozím kroku. Dělením povrchu vzniká *hierarchie* stále jemnějších ploch, v níž na nejvyšší úrovni leží počáteční řídicí síť a v dalších úrovních se postupně nacházejí povrchy vzniklé opakovaným dělením vstupní řídicí sítě. Obvykle už po několika málo krocích je síť dostatečně detailní pro to, abychom ji mohli např. zobrazovat na obrazovce počítače a přitom získali dostatečně dobrou představu o skutečném tvaru limitní plochy. Příklad děleného povrchu je vidět na obr. 3.2. Je vidět, že už po aplikaci dvou kroků dělicího algoritmu je výsledná síť velice hladká a ostré hrany ze vstupní sítě úplně zmizely.

Konkrétní předpis, podle kterého se provádí dělení plochy v každém kroku, je dán použitým dělicím algoritmem (schématem<sup>1</sup>). Dělicí schémata můžeme obecně rozdělit do dvou skupin podle toho, zda jsou *interpolační* nebo *aproximační*. Interpolační schémata v průběhu dělicího kroku vkládají nové vrcholy na nové pozice a zachovávají pozice vrcholů vytvořených v předchozích krocích. Naproti tomu aproximační schémata kromě vkládání nových bodů mění pozici bodů vytvořených v předchozím kroku. Spline plochy jsou obecně tím typem ploch, které lze popsat aproximačními dělicími schématy.

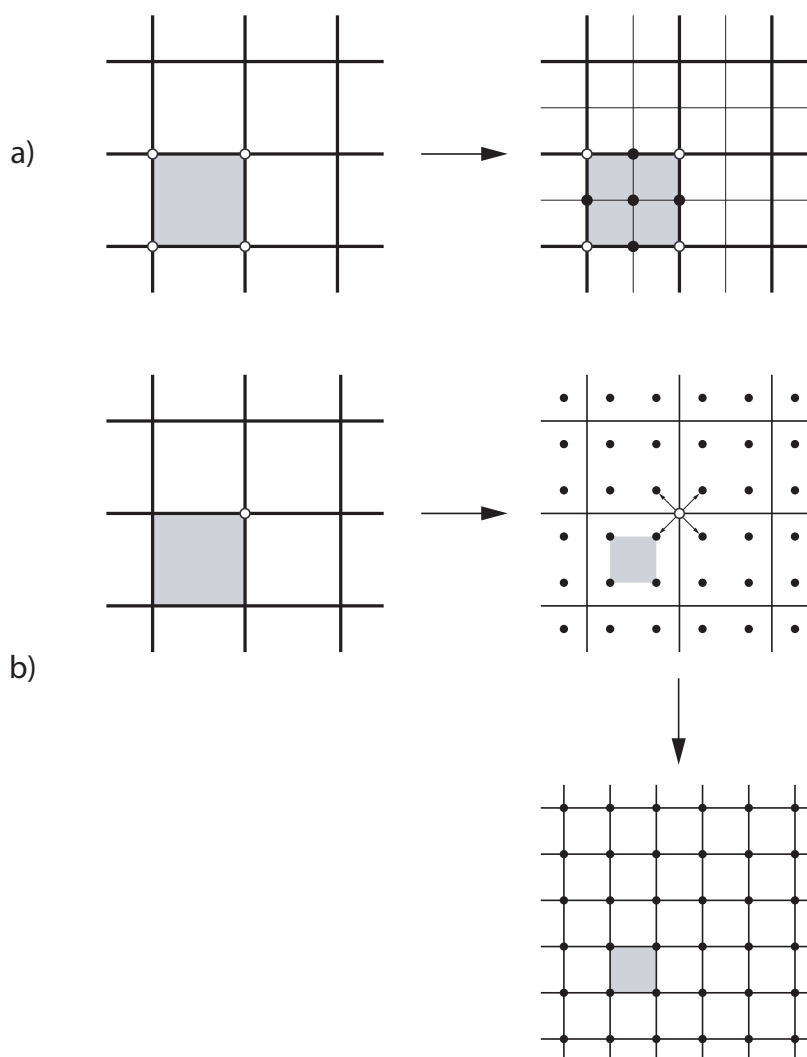
Známa a zavedená dělicí schémata můžeme klasifikovat podle následujících kritérií [18]:

<sup>1</sup>Výrazy *dělicí algoritmus* a *dělicí schéma* mají stejný význam, pokud mluvíme o způsobu, jakým se konstruuje nové body děleného povrchu v závislosti na stávajících okolních bodech.



Obrázek 3.2: Ukázka výstu pů dělicího algoritmu po aplikaci jednoho a dvou dělicích kroků na řídicí síť vlevo.

- *Typ polygonů, ze kterých se skládá generovaná plocha* – ačkoliv již bylo uvedeno, že vstupní (počáteční) řídicí síť může být složena z polygonů s libovolným počtem hran, nejčastěji se používají trojúhelníkové nebo čtyřúhelníkové sítě.
- *Typ dělicího pravidla* – nové vrcholy vznikají buď tak, že jsou vytvořeny v kontextu každého polygonu sítě z předchozího kroku (*face split*) – taková schémata nazýváme *primální*, nebo jsou vytvářeny v kontextu každého vrcholu sítě z předchozího kroku a jeho okolních polygonů (*vertex split*) – tato schémata nazýváme *duální*. Primální schémata se chovají stejně pro trojúhelníkové i čtyřúhelníkové sítě – každý polygon je vždy rozdělen na čtyři nové polygony se stejným počtem hran, nové vrcholy jsou vytvářeny uvnitř polygonu a na jeho hranách. Pozice starých vrcholů jsou v případě aproximačních schémat přepočítávány. Duální schémata vytvářejí pro každý vrchol původní sítě několik nových vrcholů – jeden pro každý polygon, který s tímto vrcholem sousedí. Polygony původní sítě jsou zachovány, nové jsou vytvářeny pro každou hranu a každý vrchol původní sítě. Na obrázku 3.3 je ukázán rozdíl mezi a) primálním a b) duálním schématem pro čtyřúhelníkovou síť. V případě čtyřúhelníkové sítě můžeme použít obě varianty, ale třeba u trojúhelníkové sítě připadá v úvahu jen primální schéma. Duální schéma (pro pravidelnou trojúhelníkovou síť, kde každý vrchol sousedí se šesti trojúhelníky) by v prvním kroku pro každý vrchol řídicí sítě vytvořilo šestiúhelník a pro každou hranu čtyřúhelník.
- *Schéma je buď aproximační, nebo interpolační* – u primálních schémat vzniká pro každý vrchol  $v_i$  polygonové sítě (jak pro vrchol počáteční řídicí sítě, tak i pro každý nově vytvořený vrchol) postupnou aplikací dělicího algoritmu posloupnost vrcholů  $v_i^k = \{v_i^1, v_i^2, \dots, v_i^\infty\}$  konvergující k pozici na limitní ploše. Pokud všechny tyto vrcholy leží na stejné pozici, jde o interpolační schéma, v opačném případě mluvíme o aproximačním



Obrázek 3.3: Dva základní typy dělicího schématu (primální a duální)

schématu. Výhodou interpolačních schémat je jasnější představa o tvaru limitní plochy už po prvním nebo druhém dělicím kroku, protože všechny vytvořené body už leží na limitní ploše. Nevýhodou je, že plochy vytvořené interpolačními dělicími algoritmy jsou ve srovnání s aproximačními méně kvalitní.

- *Limitní plocha je  $C^1$ ,  $C^2$ , ...,  $C^n$  spojitá* – všechna nejčastěji používaná schémata produkují minimálně  $C^1$  spojitě plochy pro případ pravidelné řídicí sítě, některá i  $C^2$  spojitě. V nepravidelných vrcholech tato podmínka splněna nebývá, spojitost v nich je většinou  $C^{n-1}$ , kde  $C^n$  je spojitost pro pravidelnou řídicí síť.

## 3.2 Konkrétní dělicí schémata

V této části popíšeme nejpoužívanější a osvědčené typy dělicích schémat a na základě srovnání jejich vlastností vybereme konkrétní schéma, které použijeme v naší implementaci. Zdrojem informací pro tuto část byl hlavně přehled jednotlivých dělicích schémat v [18].

Podle vlastností uvedených v předchozí části můžeme nejznámější dělicí schémata klasifikovat podle tabulky 3.1. Klasická aproximační schémata typu Loop a Catmull-Clark produkují  $C^2$  spojité povrchy a to se projeví i na kvalitě výstupu – ze srovnání výstupů jednotlivých dělicích algoritmů, kdy jako vstupní síť byla použita krychle, popř. čtyřstěn, vychází tato dvě schémata jako nejlepší z hlediska toho, jak dobře rozdělená síť vypadá a zda si zachovává nějaké tvarové nepravidelnosti nebo ostré hrany vstupní sítě. Interpolační schémata mají naopak tu vlastnost, že čím více konkrétní schéma interpoluje vstupní polygonovou síť, tím méně kvalitní je jeho výstup.

Primální		
	<i>Trojúhelníky</i>	<i>Čtyřúhelníky</i>
<i>Aproximační</i>	Loop ( $C^2$ )	Catmull-Clark ( $C^2$ )
<i>Interpolační</i>	Modified Butterfly ( $C^1$ )	Kobbelt ( $C^1$ )

Duální
Doo-Sabin, Midedge ( $C^1$ )
Biquartic ( $C^2$ )

Tabulka 3.1: Klasifikace nejpoužívanějších dělicích schémat

U aproximačních schémat (zejména u schématu Catmull-Clark aplikovaného na čtyřúhelníkovou síť, kde aproximační plocha je v limitní podobě shodná s kubickou B-spline plochou s řídicí sítí odpovídající vstupní polygonové síti[5]) stojí také za zmínku, že vztah mezi vstupní polygonovou sítí a její aproximací se už po několika dělicích krocích podobá situaci u parametrických ploch v tom smyslu, že vstupní polygonová síť je ekvivalentem řídicí sítě u parametrických ploch a manipulace s body vstupní polygonové sítě tak má na tvar aproximované sítě podobný vliv jako manipulace s řídicími body u parametrické sítě. To může být výhodou pro uživatele, který už někdy s parametrickými plochami pracoval. Interpolační schémata jsou v tomto smyslu pro interaktivní práci méně vhodná, protože síť vytvořená rozdělením vstupní polygonové sítě narozdíl od aproximačních schémat neleží v její konvexní obálce a ztrácí se výše popsaná analogie s parametrickými plochami.

Pokud je vstupní síť tvořena z větší části čtyřúhelníky (v ideálním případě *pouze* čtyřúhelníky), dává podle [18] nejkvalitnější výstup Catmull-Clarkovo



dělicí schéma. Z výše uvedených důvodů je tedy v této práci použita Catmull-Clarkova varianta dělených povrchů. Navíc jsme se rozhodli omezit vstupní množinu dat na polygonové sítě tvořené výhradně čtyřúhelníky – kromě toho, že tím zaručíme lepší kvalitu výstupů, nám to také přinese zjednodušení při návrhu datových struktur a implementaci algoritmů (viz část 5). Toto rozhodnutí samozřejmě s sebou přináší také nevýhody, např. omezení množiny použitelných vstupních dat – složitější modely tvořené výhradně čtyřúhelníky nejsou příliš dostupné.

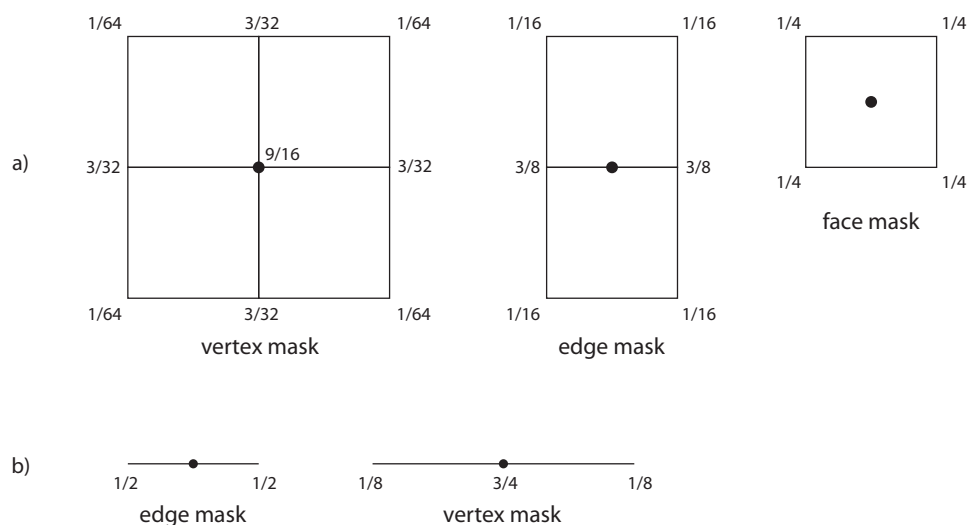
### 3.3 Catmull-Clarkovy dělené povrchy

Jedním z prvních a také jedním z nejčastěji používaných dělicích schémat pro povrchy, založeným na bikubických B-spline plochách, je Catmull-Clarkovo schéma [5]. Původní verze pracuje s pravidelnou čtyřúhelníkovou sítí, takže vrcholy mají standardně počet sousedících čtyřúhelníků  $k = 4$  ( $k$  nazýváme stupeň vrcholu). Plocha vytvořená tímto schématem je všude  $C^2$  spojitá s výjimkou vrcholů, které jsou jiného než standardního stupně, kde je plocha pouze  $C^1$  spojitá. To v našem případě nevádí, protože pro modelování je důležité, abychom mohli vytvářet vizuálně hladké plochy, a v tom nám  $C^1$  spojitost nebrání. Catmull-Clarkovo schéma je primárního typu, takže každý čtyřúhelník je v jednom dělicím kroku rozdělen na čtyři nové.

Na obrázku 3.4 jsou uvedena standardní pravidla pro výpočet nových vrcholů a výpočet nové polohy existujících vrcholů. Tyto *masky* znázorňují, jaký vliv mají na výpočet pozic nových vrcholů a výpočet nových poloh stávajících vrcholů okolní vrcholy. Tmavě je vždy zvýrazněna budoucí poloha nového vrcholu (nebo poloha stávajícího vrcholu při přepočtu jeho polohy), čísla zastupují jednotlivé okolní vrcholy. Čím je číslo u daného vrcholu větší, tím vyšší je jeho vliv na vypočítanou polohu. Horní tři masky (3.4 a) popisují vytváření vnitřních vrcholů – uvnitř čtyřúhelníku (*face mask*), na hraně (*edge mask*) a na pozici vrcholů z předchozího kroku (*vertex mask*). Pro okrajové vrcholy existují dvě speciální pravidla (3.4 b) – pro výpočet nových vrcholů na okrajových hranách (*edge mask*) a pro výpočet nové polohy již existujících okrajových vrcholů (*vertex mask*).

Obrázek 3.5 ilustruje, jakým způsobem se změní topologie polygonové sítě v okolí jednoho bodu po aplikování jednoho (v tomto případě prvního) kroku Catmull-Clarkova dělicího algoritmu. Přerušovanou čarou je nakreslena vstupní řídicí síť, plnou čarou výsledek algoritmu.

Masky uvedené na obrázku 3.4 platí pro čtyřúhelníkovou síť. Pro sítě tvořené i jinými polygony než čtyřúhelníky existují speciální pravidla, díky kterým se polygon s libovolným počtem hran po prvním dělicím kroku rozdělí na čtyřúhelníky. V dalších krocích lze pak už používat standardní pravidla. Rozdělení libovolného polygonu na čtyřúhelníky se docílí podobně jako u vyplněného čtyřúhelníku na obrázku 3.3 a) – nový bod uvnitř polygonu se spojuje s novými body na hranách polygonu tak, že vznikne  $n$  nových čtyřúhelníků, kde

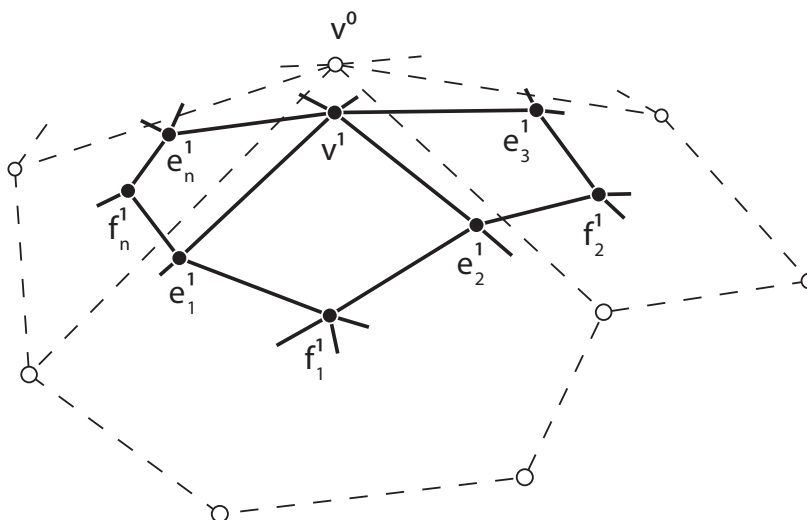


Obrázek 3.4: Základní pravidla (masky) pro vytváření nových vrcholů Catmull-Clarkova dělicího schématu

$n$  je počet hran děleného polygonu, viz také obrázek 3.5. Nový vrchol, který vznikl uvnitř polygonu, má ovšem nestandardní stupeň  $k = n$ , který je roven počtu hran takto rozděleného polygonu. Na obr. 3.6 je uvedena maska pro výpočet nové polohy vrcholu libovolného stupně. Vrcholová maska na obrázku 3.4 a) je pak konkrétní variantou této masky pro  $k = 4$ .

Kromě dosud uvedených výhod, obecně platných pro všechna dělicí schémata, má většina dělicích schémat vlastnosti, které zejména v našem typu aplikace – interaktivním modelovacím nástroji – mohou znamenat omezení v tom smyslu, že uživatel nebude mít možnost vytvořit každý představitelný tvar:

- Ne všechny objekty ve skutečném světě je možné důvěryhodně zachytit jen s pomocí hladkých parametrických ploch (v našem případě pomocí dělených povrchů). Původní dělicí pravidla na toto nepamatují, takže pokud chceme na Catmull-Clarkově ploše vytvořit v určitém místě (nekonečně) ostrou hranu, není to s jejich pomocí možné. K ostrým hranám se lze aspoň vizuálně přiblížit tak, že v požadovaném místě rozdělíme síť do takové míry, že budeme schopni vytvořit hranu, která se bude uživateli na obrazovce jevit jako ostrá. Skutečně ostré hrany lze ale vytvářet až s využitím různých modifikací původních pravidel, které umožňují např. označit libovolnou hranu jako ostrou (tzv. *crease edge*) [6]. Nově vytvářené vrcholy na této hraně pak leží přímo na ní, takže hrana zůstává po celou dobu dělení rovná a ostrá.
- Pokud pracujeme s neuzavřenou plochou, počítají se nové body na jejím okraji podle jiných pravidel než body vnitřní, viz obrázek 3.4. Výsledkem je to, že okrajová křivka limitní plochy je totožná s B-spline



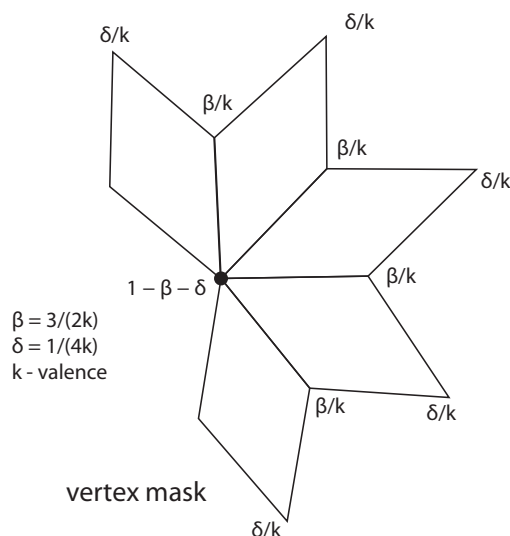
Obrázek 3.5: Změna topologie polygonové sítě v okolí jednoho vrcholu pro aplikování jednoho kroku Catmull-Clarkova algoritmu

křivkou popsanou body ležícími na okraji plochy. V okrajových bodech není otevřená Catmull-Clarkova plocha formálně  $C^1$  spojitá, ale v našem případě to není překážkou, protože plocha je i tak vizuálně dostatečně hladká. Další vlastností okraje otevřené plochy je to, že původně ostré (jak konvexní, tak konkávní) rohy se v průběhu dělení zaoblují (jak již bylo řečeno, tvar okraje odpovídá B-spline křivce). Opět byly navrženy modifikace, které umožňují označit vybrané okrajové vrcholy jako rohové a jejich pozice se pak v průběhu dělení nemění. Takto modifikované masky jsou uvedeny např. v [17].

### 3.4 Limitní poloha bodu a tečná rovina plochy

Dělené povrchy jsou vytvářeny opakovaným dělením počáteční řídicí sítě. Přitom vzniká posloupnost stále hladších ploch konvergující k nějaké limitní ploše. U aproximačních schémat nás narozdíl od interpolačních, kde každý nově vytvořený bod už leží přímo na limitní ploše a jeho poloha je tedy zároveň limitní, bude zajímat, zda jsme schopni nějakým způsobem určit limitní polohu již vytvořených bodů. Díky lokálnímu charakteru dělených povrchů, kdy roli ve výpočtu pozice nového vrcholu hrají jen vrcholy z jeho blízkého okolí, byly vytvořeny vztahy pro určení limitní polohy vrcholu pro všechna běžně používaná aproximační dělicí schémata. Konkrétně pro Catmull-Clarkovo schéma je vztah pro výpočet limitní polohy vrcholu následující [8]:

$$\infty_v = \frac{n^2 v + \sum_{j=1}^n (4e_j + f_j)}{n(n+5)}, \quad (3.1)$$



Obrázek 3.6: Vrcholová maska Catmull-Clarkova schématu pro obecný případ, kdy čtyřúhelníková síť v okolí vrcholu nemusí být pravidelná

přičemž  $v$  je vrchol, pro který počítáme limitní polohu,  $n$  je stupeň (četnost) tohoto vrcholu,  $e_j$  jsou koncové vrcholy hran s počátkem ve vrcholu  $v$  a  $f_j$  jsou vrcholy ležící v protějších (vzhledem k vrcholu  $v$ ) rozích čtyřúhelníků dotýkajících se vrcholu  $v$  (viz obr. 3.5).

Vzhledem k tomu, že na výpočtu limitní polohy bodu se podílejí body  $e_j$  a  $f_j$ , je třeba provést minimálně jeden dělicí krok předtím, než můžeme limitní polohu určit. Pro body vstupní řídicí sítě tedy limitní polohu vypočítat nemůžeme.

V [8] je rovněž uveden vztah pro výpočet tečných vektorů plochy v limitní poloze  ${}^{\infty}v^k$  daného vrcholu  $v^k$ , kde  $k$  je hloubka vrcholu  $v$  v hierarchii dělení. Opět musí platit, že  $k > 0$ , tedy nelze počítat tečné vektory pro vrcholy vstupní řídicí sítě. Označme vektory  $t_0$  a  $t_1$  jako tečné vektory tvořící v daném vrcholu tečnou rovinu. Pak normálový vektor  $\vec{n}$  v daném vrcholu vypočteme podle vztahu

$$\vec{n} = \vec{t}_0 \times \vec{t}_1, \quad (3.2)$$

přičemž vztah pro výpočet tečných vektorů je

$$\vec{t}_i = \sum_{j=0}^{n-1} \left[ A_n \cos\left(\frac{2\pi j}{n}\right) e_{(j+i)\%n}^i + \left( \cos\left(\frac{2\pi j}{n}\right) + \cos\left(\frac{2\pi(j+1)}{n}\right) \right) f_{(j+i)\%n}^i \right], \quad (3.3)$$

kde

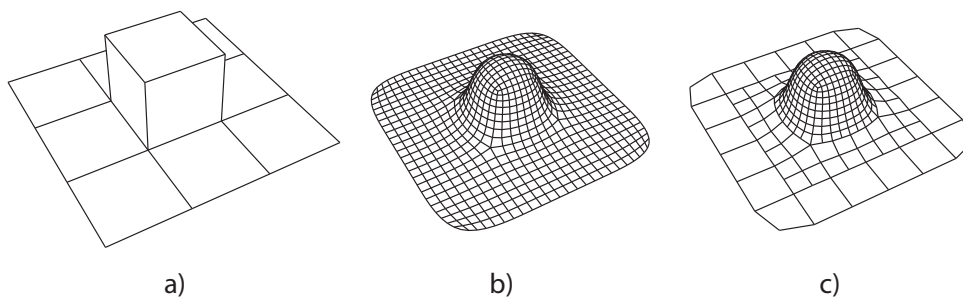
$$A_n = 1 + \cos\left(\frac{2\pi}{n}\right) + \cos\left(\frac{\pi}{n}\right) \sqrt{2 \left(9 + \cos\left(\frac{2\pi}{n}\right)\right)}, \quad (3.4)$$

$n$  je stupeň (četnost) vrcholu  $v$ , v jehož limitní poloze tečné vektory počítáme,  $e_j$  jsou koncové vrcholy hran s počátkem ve vrcholu  $v$  a  $f_j$  jsou vrcholy ležící v protějších (vzhledem k vrcholu  $v$ ) rozích čtyřúhelníků dotýkajících se vrcholu  $v$  (viz obr. 3.5). Takto vypočítaný normálový vektor tvoří v daném bodu společně s tečnými vektory lokální souřadnicový systém, který umožňuje provádět s modelem víceúrovňové editační operace (viz část 3.6).

### 3.5 Adaptivní dělení

Dosud jsme předpokládali, že vstupní polygonová síť byla dělená rovnoměrně, tedy že jsme v každém místě sítě aplikovali stejný počet kroků dělicího algoritmu. Protože ale počet polygonů tvořících dělenou plochu roste s přibývajícím počtem dělicích kroků exponenciálně, dostaneme se díky tomuto naivnímu přístupu po několika krocích do situace, kdy už nám naše výpočetní prostředky nedovolí provést další dělicí krok, ačkoliv bychom chtěli aplikací dalšího kroku například zlepšit vzhled některých jemných detailů.

V průběhu vytváření dělených povrchů tedy počet polygonů roste s každou další aplikovanou iterací dělicího algoritmu exponenciálně. Pokud ale budeme síť dělit jenom tam, kde je plocha křivější, a rovné úseky zůstanou nerozdělené, celkový počet polygonů se sníží a navíc budeme moci plochu v místech, kde je hodně členitá, rozdělit do větší hloubky a lépe zachytit drobné detaily. Na obrázku 3.7 je vidět ukázka rozdílu mezi adaptivním a neadaptivním dělením.



Obrázek 3.7: Rozdíl mezi adaptivním a neadaptivním rozdělením řídicí sítě (a) – neadaptivní varianta (b) je složená z 832 čtyřúhelníků, adaptivní (c) z 388 čtyřúhelníků. V obou případech byla řídicí síť dělena třikrát.

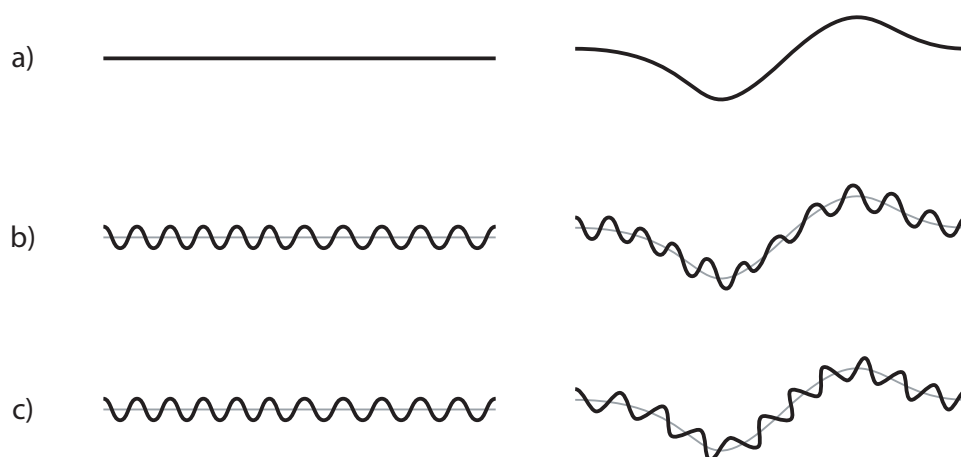
Adaptivní dělení může být obecně aplikováno dvěma způsoby. Jednodušší, ale pro uživatele méně pohodlná, je varianta, kdy si sám označí oblast, kterou chce mít rozdělenou do větší hloubky, a pak provede požadované úpravy. Druhou variantou je automatické adaptivní dělení, které je aplikováno v těch místech sítě, která nesplňují definované kritérium vizuální hladkosti. To znamená, že sledujeme, jak dobře stávající síť aproximuje limitní podobu dělené sítě na základě odchylky od této limitní polohy. V [11] se proces adaptivního dělení řídí podle odchylky každého z vrcholů od jeho limitní polohy. Každý vrchol, jehož odchylka od limitní polohy je vyšší než daná mez, je označen a při dělení polygonů, které obsahují některý z takto označených vrcholů, jsou použita speciální dělicí pravidla. Tento postup se aplikuje při každém dělicím kroku. Jednoduchá metoda pracující s normálovými vektory polygonů je uvedena v [16]. Pro každý polygon se zjišťují velikosti úhlů svíraných mezi jeho normálovým vektorem a normálovými vektory sousedních polygonů. Polygon je rozdělen, pokud je velikost úhlů vyšší než daná toleranční mez, v opačném případě je jeho dělení zastaveno.

Adaptivní dělení lze v interaktivním editoru uplatnit ve dvou situacích – při manipulaci se stávajícím modelem, kdy může dojít k deformaci rovných úseků a pak je vhodné aspoň jedním dělicím krokem zdeformovanou oblast vyhladit, nebo při přidávání nových detailů, kdy je třeba síť rozdělit, aby bylo možné detaily dostatečně věrně zachytit.

### 3.6 Víceúrovňové modelování

Pro víceúrovňové (*multiresolution*) modelovací techniky je charakteristické, že jakákoliv změna objektu na libovolné úrovni detailu nebude mít nežádoucí vliv na detaily v jemnějších úrovních (zkreslení, případně úplné odstranění detailů), které jsou na ní přímo „závislé“. U takto upravovaného modelu musí být známa hierarchie jednotlivých úrovní detailů, od základního nejhrubšího tvaru až po nejjemnější detaily. Konkrétní příklad pro křivku je na obrázku 3.8. Základní přímka (a její zdeformovaný tvar) na obrázku 3.8 a) dostala na obrázcích 3.8 b), c) další úroveň detailu – zvlnění ve směru kolmice k základní přímce. Závislost jemnějších detailů na hrubší úrovni detailů je v tomto případě dána právě orientací vln podle kolmice k základní přímce. Právě části obrázků 3.8 b) a 3.8 c) ilustrují, jakým způsobem se projeví deformace základní přímky (stejná jako na obrázku 3.8 a) na jemnější úrovni detailů. Na obrázku 3.8 b) zůstává po deformaci orientace vln stejná jako u základní přímky. Čím více bychom základní přímku deformovali, tím větší by bylo zkreslení na nižších úrovních detailu. Tato operace tedy nemá vlastnosti použitelné pro víceúrovňové modelování. Na obrázku 3.8 c) je naopak vidět, že po provedení deformace si zvlnění zachovalo původní orientaci vzhledem k základní křivce – v tomto případě jde o víceúrovňové modelování.

V příkladu na obr. 3.8 má křivka jednu přidanou úroveň detailu, která je orientována v závislosti na tvaru základní křivky. Mohli bychom ale chtít

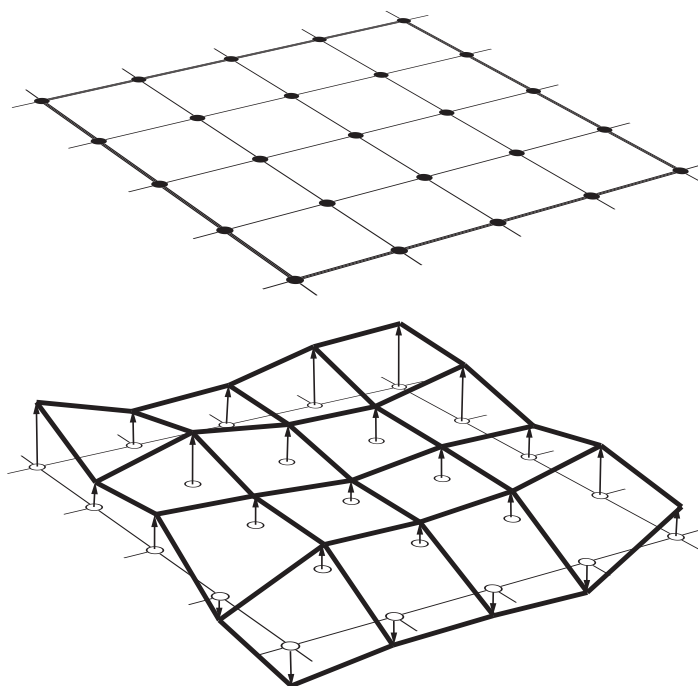


Obrázek 3.8: Změna tvaru křivky při současném zachování jemných detailů

přidat ještě druhou (jemnější) úroveň detailu, která by byla závislá na první úrovni detailu (v případě naší křivky na přidaném zvlnění). Ke zvlnění přidanému v prvním kroku bychom třeba mohli přidat další zvlnění, které by opět bylo orientováno podle kolmice k „nosné“ křivce. Tou by tentokrát už nebyla základní přímka, ale zvlněná křivka přidaná v prvním kroku. Jakákoliv změna tvaru základní přímky by tedy měla vliv nejen na první, ale i na druhou přidanou úroveň detailu, změna tvaru prvního zvlnění by pak měla vliv na orientaci druhého zvlnění. Tímto způsobem bychom mohli přidat libovolné množství stále jemnějších detailů, přičemž každá další úroveň detailů by byla přímo či nepřímo závislá na všech předchozích, včetně základního tvaru, kterým byla v tomto případě přímka. Každá přidaná úroveň detailu typicky vyžaduje zjemnění upravovaného objektu (rozdělení segmentů v případě křivky nebo rozdělení plošek v případě povrchového modelu), aby bylo možné jemnější detaily zachytit.

Důležitý je v případě víceúrovňového modelování způsob, jakým jsou detaily na jednotlivých úrovních reprezentovány. U každého bodu zjemněné křivky nebo plochy potřebujeme vyjádřit jeho posunutí tak, že změna na hrubších úrovních (na kterých je, jak jsme výše uvedli, jemnější úroveň závislá) nezpůsobí ztrátu informace o detailech na jemnějších úrovních. Řešením je vytvoření *lokálního souřadnicového systému* s počátkem v daném bodu, s jehož pomocí posunutí vzhledem k počáteční pozici bodu vyjádříme. V příkladu s křivkou jako lokální souřadnicový systém (s jednou osou) slouží kolmice ke křivce z předchozí úrovně detailu v daném bodu. K uložení posunutí tak stačí pro každý bod znát jednu hodnotu.

U plochy by podobně jako u křivky mohl být lokálním souřadnicovým systémem normálový vektor k ploše v daném vrcholu. Ukázka je vidět na obr. 3.9. Velký čtyřúhelník v horní části byl dvakrát rozdělen, čímž vznikly nové vrcholy. V dolní části obrázku jsou do rozděleného čtyřúhelníku přidány detaily



Obrázek 3.9: Přidání detailů do rozděleného čtyřúhelníku

posunutím vrcholů podél jim odpovídajících normálových vektorů k povrchu (v kladném i záporném směru).

Na základě znalosti tečné roviny (resp. tečných vektorů) v daném vrcholu bychom ovšem mohli vytvořit tříosý lokální souřadnicový systém, který by umožňoval posun do libovolné pozice. Podle vztahů popsanych v části 3.4 pro výpočet tečných vektorů k ploše bychom takový souřadnicový systém pro Catmull-Clarkovy plochy dokázali sestavit s využitím dvou tečných vektorů a normálového vektoru v daném vrcholu sítě.

## Shrnutí

Tato kapitola podrobně popsala teoretický aparát na úrovni, která bude postačovat pro vytvoření konkrétních datových struktur a algoritmů pro práci s Catmull-Clarkovými dělenými povrchy. V následující kapitole bude doplněn krátkým přehledem několika typů modelovacích operací, jejichž konkrétní implementace bude součástí vytvořeného programového vybavení.



# Kapitola 4

## Výběr modelovacích technik

Úkolem této kapitoly je přinést informace o dělených površích z hlediska editování jejich tvaru jak na úrovni geometrie, tak i topologie. Probírá konkrétní typy modelovacích operací, jejichž varianty jsou zahrnuty v navrženém řešení, a krátce se zmiňuje o využití dělených povrchů při rekonstrukci víceúrovňové hierarchie modelů analýzou jejich tvaru. Společně s předchozí kapitolou tvoří teoretický základ pro následující část dokumentu, která se bude věnovat popisu praktické implementace.

### 4.1 Změna topologie dělených povrchů

U dělených povrchů narazíme při pokusu o změnu topologie na omezení, které se například obecné polygonové sítě netýká. V části 3.6 jsme psali, že každá další aproximace<sup>1</sup> vstupní polygonové sítě je závislá na předchozí, protože ve výpočtu poloh nově vytvářených bodů vystupují vrcholy z lokálního okolí patřící do předchozí úrovně dělení. Ve výsledku to znamená, že topologie každé aproximace je závislá na topologii polygonové řídicí sítě (každý čtyřúhelník je při dělení rozdělen na čtyři menší) a tudíž změna topologie na jakékoliv hloubce děleného povrchu se musí promítnout do topologie řídicí polygonové sítě.

Při topologických změnách vstupní řídicí sítě (vytváření „děr“ v síti, odstraňování části sítě, připojování nových částí) lze používat všechny operace, které je možné aplikovat na klasickou polygonovou síť. Změny se dotknou nejen upravované řídicí sítě, ale i všech jejích existujících aproximací, které jsou na ní topologicky závislé. Pokud ale stejným způsobem začneme měnit topologii na jedné z aproximací vytvořených rozdělením vstupní polygonové sítě, je nutné tyto změny topologie provedené na jemnějších úrovních převést na změny topologie řídicí sítě.

Na příkladu vyřezávání děr (*trimming*) do děleného povrchu lze dobře ilustrovat, jakým způsobem se přenáší změny topologie nižších úrovní děleného

---

<sup>1</sup>Zjemněná polygonová síť vzniklá (opakovaným) rozdělením vstupní řídicí sítě

povrchu na počáteční polygonovou řídicí síť. Publikováno bylo několik způsobů vytváření děr v dělených površích [3, 9, 10, 15], ale pro obecnou představu lze všechny popsat tímto zjednodušeným postupem:

1. Uživatel na začátku zakreslí na model tvar díry, kterou chce na vybraném místě vyříznout.
2. Řídicí síť je v okolí křivky upravena (rozdělením a změnou tvarů a velikostí polygonů řídicí sítě) tak, aby bylo možné odstraněním vybraných polygonů aproximovat tvar zakreslené křivky.
3. Odstraněním polygonů „uvnitř“ zakreslené hranice díry vznikne její hrubá aproximace, která je případně upravena přidáním dalších polygonů. V tomto okamžiku je struktura řídicí sítě v lokálním okolí otvoru odlišná od původní podoby a to samé platí i pro všechny aproximace.
4. Struktura řídicí sítě je v okolí vytvořené díry upravena tak, aby lépe zachycovala původní tvar povrchu před vyříznutím díry. Toho se docílí jednak změnou tvaru těch polygonů (trojúhelníků), které mají nevhodné proporce způsobující nerovnosti při dalším dělení [15], a hlavně přidáním posunutí (detailů) na různých úrovních vybraným vrcholům v okolí vyříznuté díry, takže tvar se více přiblíží původní síti před řezáním. Přesnost, s jakou je původní dělený povrch v okolí vytvořené díry aproximován povrchem s dírou, může být ovlivnitelná nastavením hodnoty  $\epsilon$  na libovolně malou velikost, která omezuje maximální odchylku od původního povrchu bez díry [10].

## 4.2 Změna geometrie dělených povrchů

Díky podpoře víceúrovňového modelování mají dělené povrchy výhodu nad klasickými polygonovými modely i pokud jde o operace měnící geometrické vlastnosti modelu. Stejně jako parametrické plochy dovolují provádět úpravy manipulací s řídicími body – u dělených povrchů je takto možné měnit tvar modelu na každé úrovni rozdělené sítě, protože změna polohy libovolného vrcholu rozdělené sítě na libovolné úrovni má vliv na pozici vrcholů v lokálním okolí manipulovaného vrcholu ze všech následujících aproximací.

Nevýhodou při přidávání geometrických detailů na dělené povrchy může být fakt, že i při lokálním rozdělování sítě potřebném k jejich zachycení dochází v daném místě k lokálním změnám tvaru modelu. Čím ostřejší hrany se v daném místě nacházejí, tím větší bude změna tvaru modelu v místě rozdělení. Někdy by uživatel mohl chtít ostré hrany zachovat, což je možné zajistit použitím speciálního dělicího pravidla pro vrcholy na těchto hranách, viz část 3.3 a [6]. Konkrétní ukázky tohoto chování budou uvedeny v realizační části dokumentu.

### 4.3 Víceúrovňové modelování

Dosud jsme se bavili o přidávání geometrických detailů do modelů tvořených polygonovou sítí, kdy vstupní model nemá žádným způsobem explicitně definovanou hierarchii jednotlivých úrovní detailů. Tato hierarchie je vytvářena až při následujících úpravách, takže vstupní model slouží jako řídicí síť vytvářeného děleného povrchu a přidáváním detailů za současného rozdělování vstupní řídicí sítě vznikají nové úrovně detailů a tím i jejich hierarchie.

Někdy bychom mohli chtít aplikovat víceúrovňové transformace na polygonové modely, které již jsou dostatečně detailní, ale nemáme u nich k dispozici hierarchii jednotlivých úrovní detailů. V takovém případě je nutné analyzovat geometrii modelu a postupným rozdělováním modelu na dvě části – hrubší aproximační síť a posunutí jednotlivých vrcholů představující další úroveň v hierarchii detailů – tuto hierarchii vytvořit. V [19] je popsána konkrétní implementace metody tohoto typu společně s odkazy na další literaturu.

Tato práce nemá ambice konkurovat metodám, které informace o víceúrovňové hierarchii získávají zpětně analýzou obecných polygonových modelů za použití pokročilejších matematických prostředků. Implementovaný editor pracuje se vstupním modelem jako se základní řídicí sítí a víceúrovňová hierarchie detailů vzniká až v okamžiku, kdy uživatel nějaké detaily do základní sítě přidá.

### 4.4 „Copy & Paste“ operace

Operace typu „copy & paste“ jsou běžně používané pro práci s různými typy dat, nejčastěji s obrazovými a textovými, a většinou je jejich princip jasně definován. Při práci s povrchovými modely není ale tento typ operace úplně běžný a neexistuje jedna univerzální definice ani metoda, která by se dala použít se všemi používanými typy reprezentace povrchových modelů. U klasických polygonových modelů většinou dochází při kopírování částí modelu a jejich následném vkládání do jiných míst modelu (příp. do zcela jiného modelu) ke změně topologie v daném místě. Při kopírování a následném vkládání kopírovaných částí modelu se můžeme omezit jen na změnu geometrie polygonové sítě a neumožnit změnu její topologie (což by znamenalo, že bychom mohli zkopírované úseky vkládat pouze na místo s topologií shodnou s kopírovaným úsekem a překopírovaly by se pouze souřadnice vrcholů relativně vzhledem k okolí), znatelně tím ale omezíme možnosti, které může u povrchových modelů „copy & paste“ operace nabízet. V části 4.2 jsme psali o topologických změnách dělených povrchů a o tom, že existuje rozdíl mezi prováděním změn v topologii modelu na úrovni řídicí polygonové sítě a na některé z jemnějších úrovní vzniklých jejím rozdělením. To samé platí i pro vkládání zkopírovaných částí do upravovaného modelu, protože jak jsme uvedli výše, operace typu „copy & paste“ obecně mění topologii polygonové sítě modelu.

V [2] je popsána metoda kopírování a vkládání uživatelem vybraných částí polygonového modelu, u kterého není známá hierarchie jednotlivých úrovní detailů, jejíž existence je jinak pro víceúrovňové modelování potřebná. Podobně jako v [19] je součástí této metody oddělení jednotlivých úrovní detailů uživatelem označené části modelu, která má být zkopírována, čímž je vytvořena jejich hierarchie, a při vkládání kopírované části na cílové místo pak uživatel má možnost vybrat si konkrétní úroveň detailů, která bude při vložení použita (společně se všemi jemnějšími úrovněmi na ní závislými).

Použití dělených povrchů společně s jejich podporou pro víceúrovňové modelování přináší při operacích typu „copy & paste“ tu výhodu, že můžeme lehce kopírovat více úrovní detailů najednou, i když se omezíme jen na kopírování geometrické informace. Konkrétní implementace „copy & paste“ metody, která kopíruje pouze geometrickou informaci, je popsána v implementační části věnující se implementovaným modelovacím operacím.

# Kapitola 5

## Navržené řešení

Počínaje touto kapitolou bude popsána konkrétní implementace interaktivního editoru, která vznikla na základě znalostí získaných v předchozích dvou kapitolách. V její první polovině se věnujeme implementovaným datovým strukturám a algoritmům pro práci s Catmull-Clarkovou variantou dělených povrchů (popsanou v kapitole 3), druhá polovina je zaměřena na popis konkrétních implementovaných operací.

### 5.1 Programová implementace

Celá aplikace vytvořená v rámci této práce je naprogramována v jazyku C++ s využitím prostředí Microsoft Visual Studio 2005. Program je rozdělen na dvě hlavní části – *statickou knihovnu*, která obsahuje všechny datové struktury a algoritmy potřebné pro práci s dělenými povrchy, a samotnou spustitelnou aplikaci (*editor*) využívající knihovny MFC (*Microsoft Foundation Class*). Všechny operace s modelem, které je možné v editoru provádět, zpřístupňují třídy statické knihovny. Ta umožňuje práci s Catmull-Clarkovými dělenými povrchy, jejichž řídicí síť je tvořena výhradně čtyřúhelníky a je naprogramována s větším důrazem na nezávislost na použitém překladači a platformě, než je tomu u samotného editoru. Není tedy vázána na použití výhradně s vytvořenou MFC aplikací a operační systém MS Windows. Pro zobrazování modelů v editoru jsou použity funkce grafické knihovny OpenGL.

### 5.2 Datové struktury

V [18] jsou popsány dvě datové struktury vhodné pro reprezentaci speciálně Catmull-Clarkových dělených ploch tvořených výhradně čtyřúhelníky.

První datovou strukturu je vhodné použít v situaci, kdy celý model je dělen konstantně, tedy ve všech místech do stejné hloubky. Pro každý čtyřúhelník pak stačí pro každou úroveň dělení vytvořit dvourozměrné pole o rozměrech  $2^n \times 2^n$ , kde  $n$  je úroveň dělení. Narozdíl od níže popsané datové struktury

není nutné ukládat žádnou informaci o sousednosti čtyřúhelníků vytvořených dělením vstupní polygonové sítě, protože ta je snadno odvoditelná z jejich pozice ve dvourozměrném poli. Tím odpadá určitá režie spojená s ukládáním ukazatelů na datové struktury a hledáním sousedních čtyřúhelníků.

Druhá datová struktura je variantou známého *kvadrantového stromu* a je vhodná pro aplikace, kde používáme adaptivního dělení a očekáváme, že stupeň rozdělení vstupní řídicí sítě se bude v různých částech modelu výrazně lišit. Pokud bychom v takové situaci použili první datovou strukturu, byla by velká část kapacity alokovaných dvourozměrných polí nevyužitá. Pokud bychom pak například v některém místě modelu potřebovali síť rozdělit do větší hloubky, abychom mohli zachytit jemný detail, datové nároky by se rychle staly neúnosnými díky tomu, že model by musel být rozdělen do stejné hloubky na všech místech, a jak už jsme dříve uvedli, počet nových čtyřúhelníků roste v závislosti na stupni dělení exponenciálně.

S ohledem na datové nároky přináší omezení na čtyřúhelníkové sítě další výhodu. Víme, že vstupní řídicí síť je tvořena čtyřúhelníky, které sousedí s nejvýše čtyřmi jinými čtyřúhelníky. Navíc po každém dělicím kroku vzniknou pro každý dělený čtyřúhelník přesně čtyři nové. To přináší určitou úsporu paměti, protože rozměry odpovídajících datových polí jsou konstantní a není nutné ukládat informaci o počtu hran jednotlivých polygonů vstupní řídicí sítě.

### 5.2.1 Požadavky na vlastnosti vstupní řídicí sítě

Pro správnou funkci dělicího algoritmu je třeba splnit dvě základní podmínky týkající se vstupní řídicí sítě. Musí platit, že

- jednu hranu vstupní řídicí sítě mohou sdílet nejvýše dva čtyřúhelníky a
- nesmí nastat situace, kdy dva čtyřúhelníky mají společný vrchol, ale nemají žádnou společnou hranu.

Použité datové struktury jsou navrženy s předpokladem, že tyto podmínky budou splněny pro každou jimi reprezentovanou řídicí síť.

### 5.2.2 Třída `QTreeMesh`

Volba typu datové struktury je v našem případě jasná – v interaktivním editoru lze s jistotou očekávat, že hloubka rozdělení čtyřúhelníkové sítě se bude v různých místech výrazně lišit podle toho, jaké úpravy bude uživatel provádět a jak jemné detaily bude chtít na různých částech editovaného objektu vytvořit. Proto je datová struktura reprezentující upravovanou čtyřúhelníkovou síť založena na struktuře kvadrantového stromu.

Dělený povrch je na nejvyšší úrovni tvořen sítí čtyřúhelníků, z nichž pro každý je vytvořena instance třídy `RootQTree`. Každý ze čtyřúhelníků řídicí sítě má explicitně uvedenou informaci o těch sousedech, se kterými sdílí některou z hran. Kromě toho zná ještě u každého sousedního čtyřúhelníku informaci

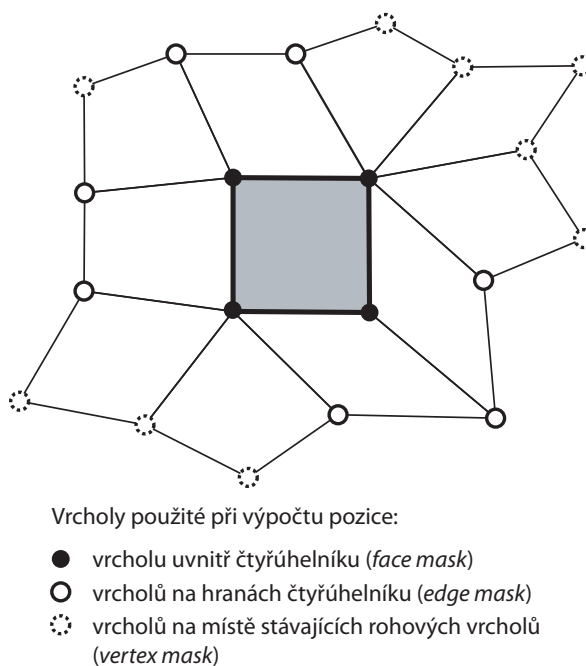
o tom, jakou hranou s ním tento čtyřúhelník sousedí. Tyto údaje stačí k tomu, aby bylo možné získat informaci o lokálním okolí vybraného čtyřúhelníku, kterou budeme potřebovat při dělení sítě. Jelikož každá instance třídy *RootQTree* představuje jeden kvadrantový strom, lze říci, že instance třídy *QTreeMesh* je tvořena „lesem“ kvadrantových stromů (instancí třídy *RootQTree*).

Při prvním rozdělení čtyřúhelníku z řídicí sítě jsou vytvořeny čtyři instance třídy *ChildQTree*. Třídy *ChildQTree* a *RootQTree* mají společného předka – třídu *QTree*. Hlavní rozdíl mezi třídami *ChildQTree* a *RootQTree* je dán tím, že třída *RootQTree* má explicitně uvedeny informace o svém okolí, tedy o sousedních čtyřúhelnících, zatímco u třídy *ChildQTree* je nutné tyto informace vyhledávat procházením hierarchie předků v kvadrantového stromu (pro hledání sousedních čtyřúhelníků v hierarchii kvadrantového stromu je použit algoritmus podobný algoritmu popsánému v [7]). Při dalším rozdělení povrchu už jsou pro každou instanci třídy *ChildQTree* opět vytvořeny čtyři nové instance třídy *ChildQTree*.

Vrcholy děleného povrchu jsou také reprezentovány dvěma třídami se společnou rodičovskou třídou *Vertex* – pro vrcholy vstupní řídicí sítě jsou vytvářeny instance třídy *RootVertex*, pro vrcholy nově vytvářené při dělení jsou použity instance třídy *ChildVertex*. Každý ze čtyřúhelníků (*Root-* i *ChildQTree*) má čtyři odkazy na instance tříd těchto dvou typů.

### 5.2.3 Třída *OneRing*

Masky dělicího schématu se při dělení sítě aplikují na určité lokální okolí daného vrcholu nebo místa, kde má být vytvořen nový vrchol. Catmull-Clarkovo schéma patří do skupiny dělicích schémat, která pro výpočet nových pozic vrcholů používá vrcholy z tzv. *1-okolí* (*1-neighborhood* nebo *one-ring neighborhood*). To znamená, že pro aplikaci masek dělicího schématu potřebujeme znát všechny čtyřúhelníky z okolí děleného čtyřúhelníku, které s ním sdílí aspoň jeden vrchol (viz masky Catmull-Clarkova schématu na obr. 3.4 a 3.6). Jelikož tato informace není ve třídě *QTreeMesh* explicitně uvedena (z toho důvodu, že struktura sítě se může často měnit a také proto, že by dále narostly paměťové nároky, pokud by se u každého čtyřúhelníku udržovala další informace o jeho okolí), je nutné ji v okamžiku potřeby zkonstruovat nalezením prstenců sousedních čtyřúhelníků pro každý z vrcholů děleného čtyřúhelníku. Na obr. 5.1 jsou vidět všechny vrcholy z okolí zvýrazněného čtyřúhelníku, které jsou při jeho dělení použity při aplikaci masek dělicího algoritmu. Vrcholy jsou označeny podle toho, která dělicí maska je využívá při výpočtu pozic nových vrcholů. Třída *OneRing* slouží k získání této informace pro libovolný čtyřúhelník děleného povrchu.



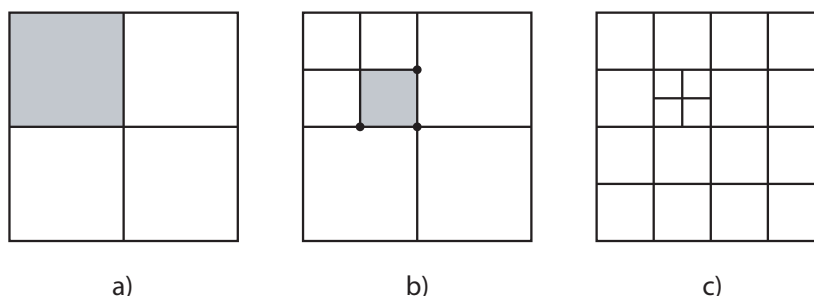
Obrázek 5.1: Část čtyřúhelníkové sítě v okolí prostředního zvýrazněného čtyřúhelníku, která má vliv na polohu vrcholů vytvářených při jeho rozdělení.

### 5.3 Podpora pro adaptivní dělení

Třída *QTreeMesh* umožňuje dělit řídicí síť nerovnoměrně, ovšem ne bez omezení. Důvodem je to, že obecně každé dělicí schéma při výpočtu nového vrcholu nebo při přepočtu polohy vrcholu stávajícího používá vrcholy vypočtené v předchozím dělicím kroku. V našem případě, kdy dělíme síť lokálně (adaptivně), vznikají vrcholy, které mají ve svém okolí nejen vrcholy se stejnou hloubkou dělení, ale také vrcholy z předchozích dělicích kroků. Protože masky dělicího algoritmu (viz masky pro Catmull-Clarkovo schéma na obr. 3.4) pracují s okolními vrcholy z předchozího dělicího kroku, je třeba zajistit, aby tyto vrcholy existovaly. Na obr. 5.2 je vidět, jakým způsobem je toto zajištěno, konkrétně pro Catmull-Clarkovo dělicí schéma. Platí, že před rozdělením jakéhokoliv čtyřúhelníku je třeba zajistit, aby všechny čtyřúhelníky, které s ním sdílejí alespoň jeden vrchol, byly rozděleny do stejné hloubky. Jinak řečeno, hloubky rozdělení jakýchkoliv dvou sousedících čtyřúhelníků ležících v jedné úrovni hierarchie se může lišit nejvýše o jeden stupeň. Na obr. 5.2 a) je vidět počáteční řídicí síť a chceme rozdělit levý horní čtyřúhelník. Všechny ostatní čtyřúhelníky využívající některý z jeho vrcholů jsou ve stejné úrovni hierarchie (v tomto případě v jde o čtyřúhelníky řídicí sítě), takže můžeme čtyřúhelník rozdělit. Výsledkem je síť na obr. 5.2 b). Pokud nyní budeme chtít dále rozdělit nově vytvořený označený čtyřúhelník na obr. 5.2 b), budeme muset nejdříve rozdělit ostatní tři čtyřúhelníky z počáteční řídicí sítě, protože u třech vyznače-



ných vrcholů neexistují všechny okolní vrcholy potřebné pro aplikaci vrcholové masky Catmull-Clarkova algoritmu (viz obr. 3.4 a) – vertex mask). Po rozdělení okolí můžeme označený čtyřúhelník rozdělit (obr. 5.2 c).



Obrázek 5.2: Omezení rozdílu v hloubce dělení v okolí vrcholu

## 5.4 Triangulace adaptivně rozdělené sítě

Zatímco interně je upravovaný model reprezentován sítí čtyřúhelníků, pro potřeby jeho zobrazení na obrazovce počítače je nutné ho převést na síť trojúhelníků, hlavně z důvodu odstranění „trhlin“, které by se v síti objevily na hranách oddělujících čtyřúhelníky rozdělené do různých hloubek.

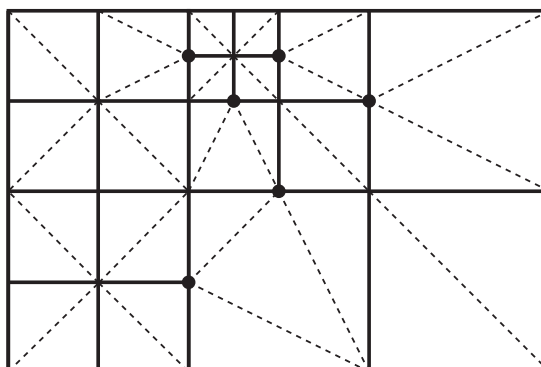
Neadaptivně (konstantně) dělená čtyřúhelníková síť je na trojúhelníkovou síť převeditelná velice snadno. Narozdíl od adaptivně dělené sítě nevznikají tzv. *T-vrcholy*, které mohou způsobit nežádoucí artefakty při zobrazování modelu na obrazovce, takže každý čtyřúhelník se jednoduše převede na dva trojúhelníky.

U nerovnoměrně dělené sítě vzniká několik modelových situací, ve kterých figurují *T-vrcholy* a triangulaci je třeba přizpůsobit tak, aby při vykreslení modelu na obrazovce počítače nevznikaly „trhliny“ (*cracks*) ve hranách s *T-vrcholem*. Ukázkou nerovnoměrně rozdělené sítě s vyznačenými *T-vrcholy* a jednou variantou triangulace sítě viz na obr. 5.3.

Všechny možné varianty, které mohou při triangulaci čtyřúhelníku v souvislosti s *T-vrcholy* nastat, jsou zobrazeny na obr. 5.4 společně s vyznačením odpovídající triangulace. Díky tomu, že hloubka rozdělení sousedních čtyřúhelníků se může lišit maximálně o jednu úroveň (z důvodů uvedených v části 5.3), existuje pouze pět variant podle počtu a pozice *T-vrcholů* v daném čtyřúhelníku.

### Poznámky k implementaci

Trojúhelníková síť pro potřeby zobrazování modelu se vytváří zvlášť pro každou instanci třídy *RootQTree*. V závislosti na maximální zobrazené hloubce

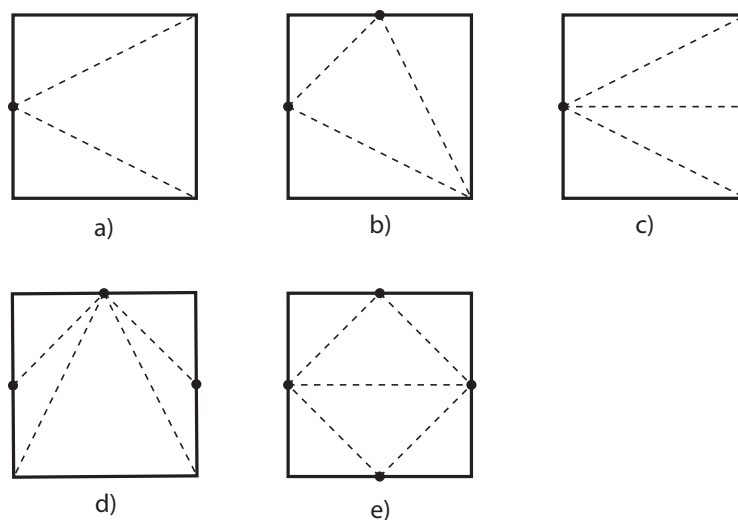


Obrázek 5.3: Ukázka triangulace adaptivně dělené čtyřúhelníkové sítě s vyznačenými T-vrcholy

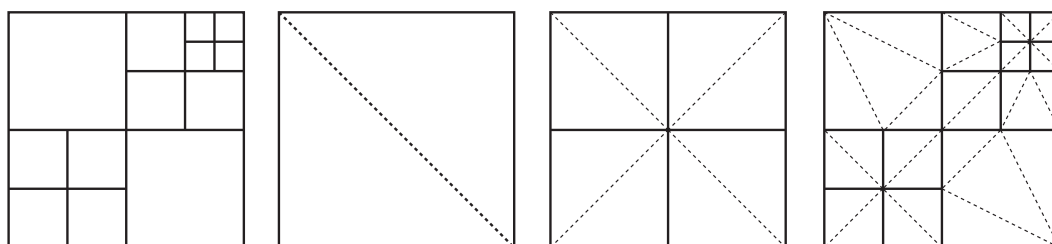
detailů nastavené v editoru (více viz v části 5.6) se vytvoří instance třídy *TriangleMesh*, která obsahuje triangulaci všech uzlů (instancí třídy *QTree*) odpovídajícího kvadrantového stromu, jejichž hloubka nepřesahuje nastavenou maximální hodnotu zobrazené hloubky detailů. Na obr. 5.5 zcela vlevo je vidět jeden nerovnoměrně rozdělený čtyřúhelník ze vstupní řídicí sítě (s maximální hloubkou dělení 3) a jeho tři triangulace, které odpovídají hodnotám maximální zobrazené hloubky detailů 0, 1 a 3. Tato trojúhelníková síť je aktualizována v situaci, kdy se změní struktura hierarchie potomků daného čtyřúhelníku vstupní řídicí sítě (dojde k rozdělení některého ze čtyřúhelníků v jeho hierarchii) nebo když uživatel změní aktuálně zobrazenou hloubku detailů.

## 5.5 Podpora pro víceúrovňové modelování

V části 3.6 je zmíněna nutnost existence nějakého lokálního souřadnicového systému u jednotlivých vrcholů sítě, aby bylo možné s jeho pomocí vyjádřit posunutí daného vrcholu do polohy mimo jeho počáteční pozici a tím zachytit geometrii modelu na různých úrovních detailu. Ze vztahů 3.1 a 3.3 a podle obr. 3.5 je vidět, že vztah pro výpočet limitní polohy vrcholu vyžaduje, aby síť v okolí bodu, pro který limitní polohu počítáme, byla rozdělena rovnoměrně do stejné hloubky. Tento požadavek je totožný s tím, který je popsán v části 5.3, a tudíž pro každý vrchol, u kterého budeme chtít vypočítat limitní polohu a tečné vektory v limitní poloze způsobem popsáným v části 3.4, budeme nejdříve muset tuto podmínku splnit. Jelikož limitní polohu a tečné vektory potřebujeme vypočítat pro každý nově vytvořený vrchol, nelze tento problém řešit stejným způsobem jako v části 5.3. Při každém rozdělení čtyřúhelníku totiž vznikají další nové vrcholy, u kterých je opět nutné vypočítat limitní polohu a tečné vektory, ale k tomu je opět nutné nejdříve rozdělit všechny okolní čtyřúhelníky rovnoměrně do stejné hloubky. Tento postup by se opakoval



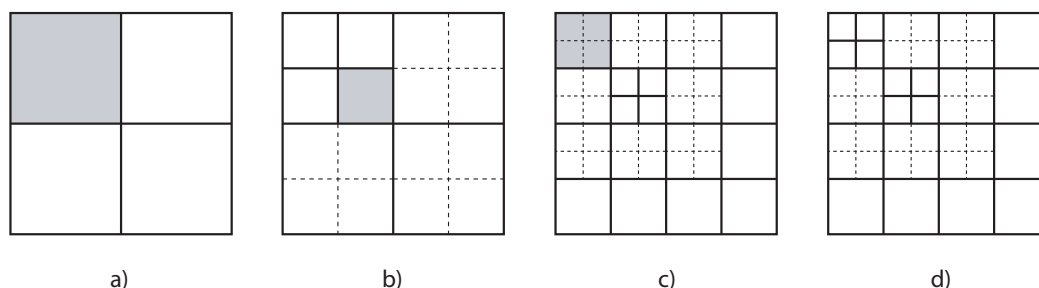
Obrázek 5.4: Pravidla pro triangulaci adaptivně dělené čtyřúhelníkové sítě s T-vrcholy



Obrázek 5.5: Ukázka rozdílných triangulací nerovnoměrně rozděleného čtyřúhelníku vstupní řídicí sítě (zcela vlevo), kdy maximální zobrazená hloubka detailů je postupně nastavena na 0, 1 a 3.

tak dlouho, než by celá síť byla rozdělena rovnoměrně do jedné hloubky, což ovšem není v souladu s naším požadavkem podpory adaptivního dělení.

Proto se pro potřeby výpočtu limitní polohy vrcholu a tečných vektorů rozdělí okolí vrcholu tak, že čtyřúhelníky, které jsou při tomto dělení vytvořeny, jsou označeny jako skryté, stejně tak i vrcholy vytvořené v průběhu dělení čtyřúhelníků. U skrytých vrcholů se pak vypočítá pouze poloha, ale už ne limitní poloha a tečné vektory. Ty se vypočítají až v okamžiku, kdy mají být tyto skryté vrcholy zobrazeny, tj. v situaci, kdy dosud „skryté“ rozdělené čtyřúhelníky mají být rozděleny viditelně, tedy běžným způsobem. Na obr. 5.6 je podobně jako na obr. 5.2 ukázáno, jakým způsobem se bude měnit struktura sítě při postupném dělení vybraných čtyřúhelníků. Skryté rozdělení je znázorněno přerušovanou čarou. Na obr. 5.6 a) je nerozdělená vstupní řídicí síť. Po rozdělení levého horního čtyřúhelníku se zároveň rozdělí i všechny jeho sousední



Obrázek 5.6: Síť se skrytě rozdělenými čtyřúhelníky. Čtyřúhelníky s hranami zakreslenými přerušovanou čarou jsou při zobrazení modelu neviditelné.

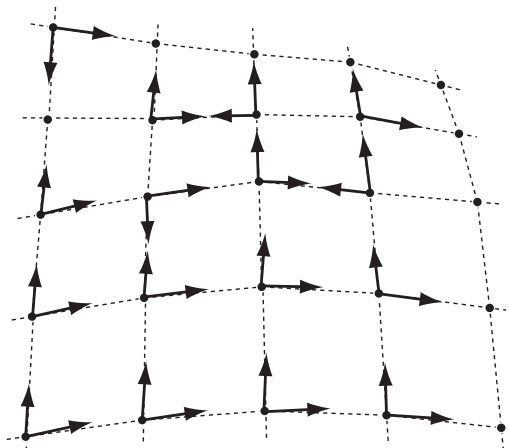
čtyřúhelníky, ale uživatel stále vidí jenom rozdělení levého horního čtyřúhelníku. Na obr. 5.6 c) je vidět situace po rozdělení čtyřúhelníku zvýrazněného na obr. 5.6 b). Skrytě rozdělené čtyřúhelníky řídicí sítě byly nejdříve rozděleny viditelně a poté byl viditelně rozdělen zvýrazněný čtyřúhelník. Nakonec byly skrytě rozděleny všechny čtyřúhelníky sousedící s právě rozděleným čtyřúhelníkem. Na obr. 5.6 d) je pak vidět, že rozdělení čtyřúhelníku zvýrazněného na obr. 5.6 c) už nezpůsobí žádné další dělení okolních čtyřúhelníků, protože všechny už jsou rozděleny do stejné úrovně jako zvýrazněný čtyřúhelník. Dojde jen k zobrazení skrytých čtyřúhelníků a vrcholů a vypočtení limitní polohy a tečných vektorů odkrytých vrcholů.

## Poznámky k implementaci

Lokální souřadnicový systém je odlišný pro vrcholy vstupní řídicí sítě a pro nově vytvářené vrcholy. U vrcholů řídicí sítě je totožný s globálním souřadnicovým systémem, protože limitní polohu vrcholu a jí odpovídající tečné vektory lze podle vztahů uvedených v části 3.4 vypočítat nejdříve pro vrcholy vytvořené prvním rozdělením vstupní řídicí sítě. Pro nově vytvářené vrcholy (instance třídy *ChildVertex*) je tedy lokální souřadnicový systém tvořen normálovým vektorem vypočteným podle vztahu 3.2. Posunutí vrcholu z počáteční pozice v rámci jeho souřadnicového systému je tedy dáno trojicí hodnot v případě vrcholů vstupní řídicí sítě, resp. jednou hodnotou v případě nově vytvářených vrcholů.

Původní myšlenka zahrnout do lokálního souřadnicového systému kromě normálového vektoru i oba tečné vektory (viz závěr části 3.6) se nakonec ukázala být nevhodnou při implementaci „copy & paste“ metody. Hlavním problémem je to, že směry tečných vektorů vypočítaných podle vztahu 3.3 jsou závislé na tom, v jakém pořadí při výpočtu vystupují jednotlivé vrcholy  $e_j$  a  $f_j$ , tedy na pořadí, ve kterém jim byly přiřazeny hodnoty indexu  $j$ . Výsledkem je, že tečné vektory jsou na tečné rovině orientovány souběžně s hranami jednoho z okolních čtyřúhelníků, ale v praxi lze těžko zajistit to, aby u všech vrcholů byly orientovány „shodně“ a při praktickém testování nastávala situ-

ace podobná té na obr. 5.7. Na něm je vidět, že i na pravidelném úseku sítě se může orientace os lokálních souřadnicových systémů jednotlivých vrcholů lišit.



Obrázek 5.7: Rozdílné orientace tečných vektorů u jednotlivých vrcholů na pravidelném úseku sítě

## 5.6 Implementované modelovací operace

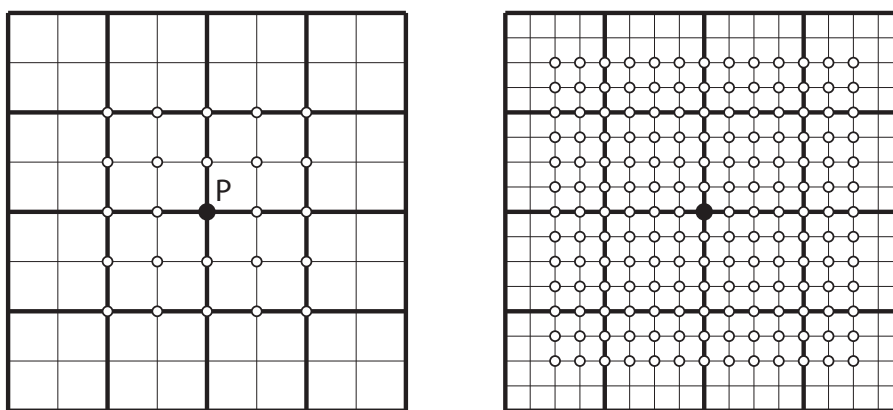
Uživatel má ve vytvořeném editoru k dispozici několik základních operací. Může měnit geometrické vlastnosti modelu manipulací s vrcholy děleného povrchu na libovolné úrovni detailu, přičemž detaily na jemnějších úrovních zůstávají zachovány. Před přidáním detailů může danou oblast povrchu zjemnit rozdělením vybraných čtyřúhelníků. Změnu topologie děleného povrchu může provádět odstraňováním a „vytahováním“ (*extrusion*) vybraných čtyřúhelníků vstupní řídicí sítě. Operace typu „copy & paste“ umožňuje kopírovat detaily mezi čtyřúhelníky ležícími ve stejné úrovni hierarchie. V jednom okamžiku může uživatel provádět změny modelu pouze na jedné úrovni detailů (v editoru je tato hodnota nazvána *active depth*), kterou si mění průběžně podle potřeby.

### 5.6.1 Manipulace s vrcholy děleného povrchu

Manipulace s jednotlivými vrcholy děleného povrchu je nejjednodušší způsob, jakým může v editoru uživatel přidávat do modelu nové detaily a měnit stávající. Je obdoby manipulace s řídicími body u parametrických ploch. Výhodou je to, že u dělených povrchů není omezena pouze na vrcholy vstupní řídicí sítě.

Poloha každého vrcholu děleného povrchu má vliv na omezenou množinu vrcholů ze všech nižších úrovní hierarchie z jeho lokálního okolí, přičemž oblast

vlivu je dána strukturou masek použitého dělicího schématu (viz obr. 3.4 a 3.6). Na obr. 5.8 je pro pravidelnou čtyřúhelníkovou síť ukázáno, jak velkou oblast sítě ovlivňuje poloha jednoho vrcholu. Změna polohy vrcholu P ovlivní polohu vrcholů ze všech nižších úrovní hierarchie dělení v jeho lokálním okolí. Na síti vlevo jsou označeny takto ovlivněné body ležící v hierarchii dělení o jednu úroveň níže, na pravé síti jsou označeny ovlivněné body ležící o dvě úrovně níže. Je vidět, že oblast vlivu je v tomto případě, kdy síť je pravidelná, omezena na velikost  $4 \times 4$  čtyřúhelníky. Obecně platí, že oblast vlivu je omezena hranicí skupiny čtyřúhelníků, která je tvořena čtyřúhelníky dotýkajícími se daného vrcholu a čtyřúhelníky z 1-okolí (viz část 5.2.3 a obr. 5.1) těchto čtyřúhelníků.

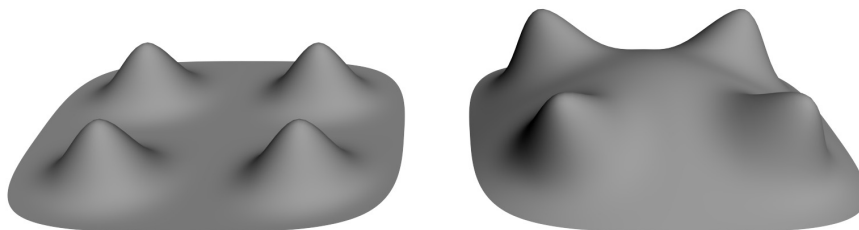


Obrázek 5.8: Znázornění lokálního vlivu vyznačeného vrcholu na polohy vrcholů v nižších úrovních hierarchie konstantně dělené sítě.

Při změně polohy vrcholu zůstávají detaily na jemnějších úrovních v oblasti jeho vlivu zachovány. Povrch na obr. 5.9 vlevo vznikl rozdělením vstupní řídicí sítě dvěma dělicími kroky, přidáním detailů do takto rozdělené sítě změnou polohy vybraných nově vytvořených vrcholů a opětovným rozdělením upravené sítě dalšími třemi kroky, čímž došlo k vyhlazení přidaných detailů. Povrch vpravo vznikl z levého posunutím jednoho z vrcholů vstupní řídicí sítě. Je vidět, že posunutí se projevilo i na jemnějších úrovních obsahujících detaily, jejichž orientace vzhledem k okolí zůstala zachována.

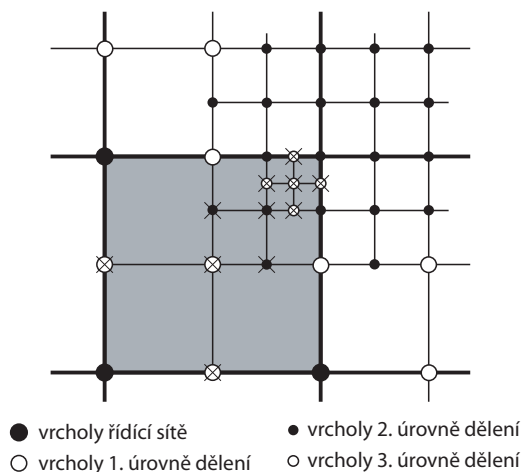
### 5.6.2 Odstranění čtyřúhelníků řídicí sítě

Odstraňováním jednotlivých čtyřúhelníků vstupní řídicí sítě lze v děleném povrchu vytvářet díry. Je přitom možné odstranit i rozdělený čtyřúhelník a to bez ohledu na to, do jaké hloubky jsou rozděleny jeho sousední čtyřúhelníky, protože po odstranění vznikne díra, na jejíž hranici není nutné dodržet podmínku popsanou v části 5.3, která říká, že hloubka rozdělení sousedních čtyřúhelníků se může lišit nejvýše o jeden stupeň.



Obrázek 5.9: Ukázka zachování detailů na jemnější úrovni při změně geometrie na hrubší úrovni.

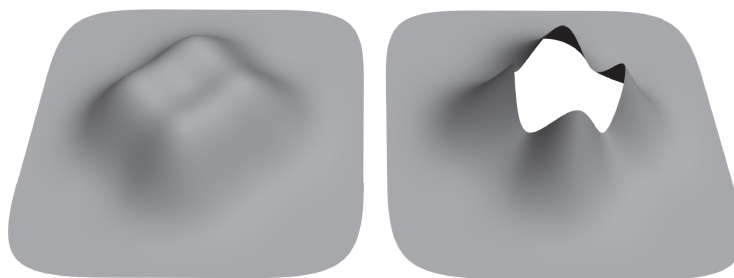
Při odstraňování čtyřúhelníku dojde také k odstranění všech čtyřúhelníků a vrcholů z jeho hierarchie potomků. Zachovány zůstanou pouze vrcholy, které leží na hraně sdílené s některým ze sousedících čtyřúhelníků a které jsou přitom využívány některým ze čtyřúhelníků v hierarchii potomků tohoto sousedícího čtyřúhelníku. Na obr. 5.10 jsou křížkem označeny ty vrcholy, které budou odstraněny společně se zvýrazněným čtyřúhelníkem. Nepřeškrtnuté vrcholy ležící na některé z hran odstraňovaného čtyřúhelníku zůstanou zachovány, protože jsou využívány dalšími čtyřúhelníky z hierarchie některého ze sousedících čtyřúhelníků.



Obrázek 5.10: Vrcholy odstraněné při odstranění zvýrazněného čtyřúhelníku vstupní řídicí sítě jsou označeny křížkem.

Odstraněním čtyřúhelníku se změní topologie vstupní řídicí sítě a proto je nutné aktualizovat pozice těch vrcholů děleného povrchu, na které má tato změna vliv. Pokud byly v okolí odstraňovaného čtyřúhelníku přidány do sítě detaily, bude mít odstranění čtyřúhelníku vliv na jejich tvar, protože při aktualizaci pozic existujících vrcholů budou použity jiná dělicí pravidla (masky)

a vrcholy z odstraněného čtyřúhelníku, které původně také měly vliv na tvar svého okolí, už ve výpočtu aktualizovaných pozic vrcholů vystupovat nebudou. K největší změně tvaru dojde v okolí hran odstraněného čtyřúhelníku (viz ukázkou na obr. 5.11, kdy z původní vstupní řídicí sítě o rozměrech  $3 \times 3$  čtyřúhelníky byl odstraněn prostřední čtyřúhelník).



Obrázek 5.11: Po odstranění čtyřúhelníku z řídicí sítě se změnil tvar okolních detailů.

### 5.6.3 „Vytažení“ čtyřúhelníku řídicí sítě

Při vytažení čtyřúhelníku dochází ke změně topologie vstupní řídicí sítě. Vrcholy vybraného čtyřúhelníku jsou zkopírovány a posunuty o pevně danou vzdálenost zhruba odpovídající délce jedné hrany daného čtyřúhelníku, zároveň jsou vytvořeny čtyři nové čtyřúhelníky, které spojují vytažený čtyřúhelník s řídicí sítí. Na obr. 5.12 jsou srovnány rozdíly ve tvaru děleného povrchu po vytažení vybraného čtyřúhelníku (vpravo) a po posunutí jeho rohových vrcholů (vlevo). Vliv na tvar sítě je u vytažení více lokalizovaný než je tomu u posunutí rohových vrcholů čtyřúhelníku.

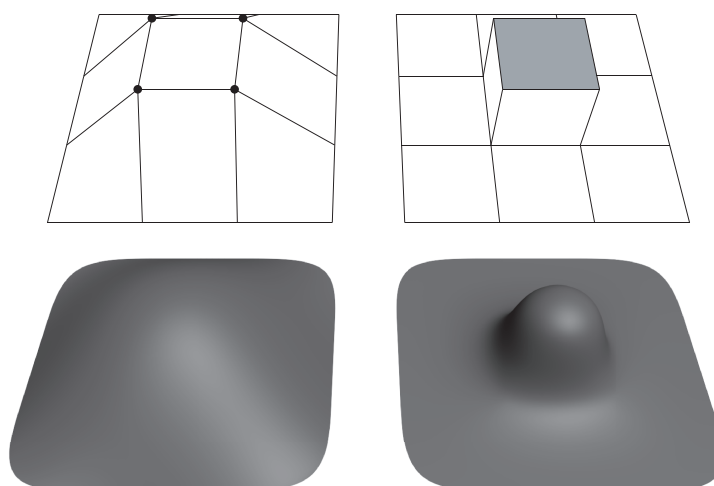
Vytažení nemá takový vliv na tvar okolí, jako je tomu u posunutí rohových vrcholů.

Tato operace je implementována tak, aby bylo možné vytahovat jen ty čtyřúhelníky vstupní řídicí sítě, které ještě nebyly rozděleny. Navíc platí, že ani žádný ze sousedů čtyřúhelníku nesmí být rozdělený. Modelování pomocí vytažení je tedy vhodné použít ještě před přidáváním detailů.

### 5.6.4 Operace typu „Copy & Paste“

V našem případě jde o variantu, kdy je kopírována pouze geometrická informace. Uživatel označí jeden čtyřúhelník z libovolné hloubky rozdělení, který bude sloužit jako zdroj detailů. Kopii detailů z tohoto čtyřúhelníku pak může vložit do jiného čtyřúhelníku, který ovšem musí ležet ve stejné hloubce dělení. Kopírovány jsou hodnoty posunutí podél normálového vektoru jednotlivých vrcholů z kompletní hierarchie potomků zdrojového čtyřúhelníku. Překopírování detailů konkrétně proběhne takto:

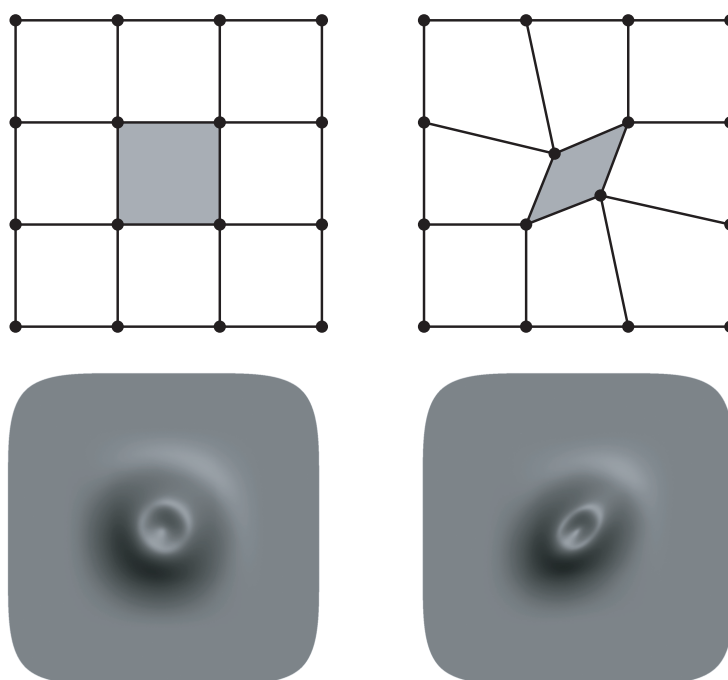




Obrázek 5.12: Rozdíl mezi operací „vytažení“ vybraného čtyřúhelníku vstupní řídicí sítě (vpravo) a změnou polohy jeho vrcholů (vlevo)

1. Cílový čtyřúhelník je nejdříve rozdělen takovým způsobem, aby každý vrchol z hierarchie potomků zdrojového čtyřúhelníku měl svůj protějšek ve vrcholu, který se v hierarchii potomků cílového čtyřúhelníku nachází na stejné pozici. Tím je zajištěno, že cílový čtyřúhelník bude rozdělen minimálně do stejné míry jako zdrojový, aby mohl převzít kopírovanou geometrickou informaci.
2. Hodnoty určující posunutí podél normálového vektoru u zdrojových vrcholů jsou postupně po jednotlivých úrovních překopírovány na odpovídající cílové vrcholy. Pokud již některé cílové vrcholy měly nastaveny nenulové hodnoty posunutí, budou nahrazeny hodnotami ze zdrojových vrcholů. Pokud již byl cílový čtyřúhelník rozdělen do větší hloubky než zdrojový, všechny detaily na těchto jemnějších úrovních (úrovně, které se v hierarchii zdrojového čtyřúhelníku nevyskytují) budou odstraněny, aby zůstal zachován celkový tvar zdrojových detailů.

Nevýhodou tohoto přístupu, kde detaily jsou kopírovány právě mezi dvěma čtyřúhelníky, je deformace kopírovaných detailů v situaci, kdy jsou vloženy na tvarově dostatečně odlišný čtyřúhelník. Na obr. 5.13 je vidět příklad toho, jaký vliv má tvar cílového čtyřúhelníku na deformaci tvaru vložených detailů. V horní řadě jsou vstupní řídicí sítě, v dolní řadě odpovídající dělené povrchy vytvořené několikerým rozdělením sítí z horní řady. Do označeného čtyřúhelníku v levé řídicí síti byly navíc přidány detaily na různých úrovních, které pak byly překopírovány do označeného čtyřúhelníku v pravé řídicí síti. Je vidět, že odlišný tvar cílového čtyřúhelníku ovlivnil tvar vložených detailů tak, že příliš neodpovídá tvaru detailů na zdrojovém čtyřúhelníku. Ke stejné deformaci by ovšem došlo i u levé sítě v případě, že bychom po přidání detailů do čtyř-



Obrázek 5.13: Ukázka deformace kopírovaných detailů v situaci, kdy se tvar zdrojového a cílového čtyřúhelníku výrazně liší

úhelníku dodatečně změnil jeho tvar na tvar podobný čtyřúhelníku z pravé sítě.

Stávající implementace této operace nepodporuje kopírování detailů z více čtyřúhelníků najednou. Pokud bychom chtěli implementovat kopírování detailů z více čtyřúhelníků najednou a přitom používali tříosého souřadnicového systému, narazili bychom na dříve zmiňovaný problém s různou orientací tečných vektorů u jednotlivých kopírovaných vrcholů (viz poznámky k implementaci v části 5.5). Pokud by totiž lokální souřadnicové systémy vrcholů zdrojových čtyřúhelníků měly mezi sebou navzájem jinou orientaci než lokální souřadnicové systémy vrcholů cílových čtyřúhelníků, došlo by při vložení detailů k jejich deformaci.

Nutno dodat, že popsáním způsobem je možné kopírovat pouze ty detaily, které jsou vytvořeny v rámci víceúrovňové hierarchie, tedy ty, které předtím uživatel v editoru do modelu přidal po předchozím aplikování alespoň jednoho kroku dělicího algoritmu. Nelze takto kopírovat části vstupní řídicí sítě, protože pozice a posunutí jejích vrcholů jsou vyjádřeny v rámci globálního souřadnicového systému, který je pro všechny vrcholy stejný a tudíž nerespektuje lokální charakter vstupní řídicí sítě v okolí jednotlivých vrcholů.

# Kapitola 6

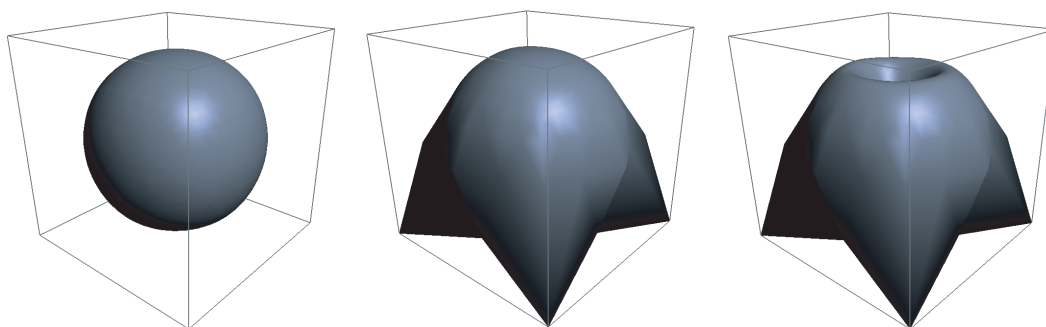
## Výsledky

V této kapitole uzavírající popis navrženého řešení je na několika praktických postupech ukázáno, jak se v práci podařilo využít vlastností dělených povrchů v interaktivním modelování. Všechny soubory, na které je v této části odkazováno, lze nalézt na přiloženém CD. Kromě těchto souborů se na CD také nalézá demonstrační video, kterým zachycuje postup při vytváření jednoduchého modelu. Všechny obrázky v této části pochází z vytvořeného editoru.

### 6.1 Přidání detailů do existujícího modelu

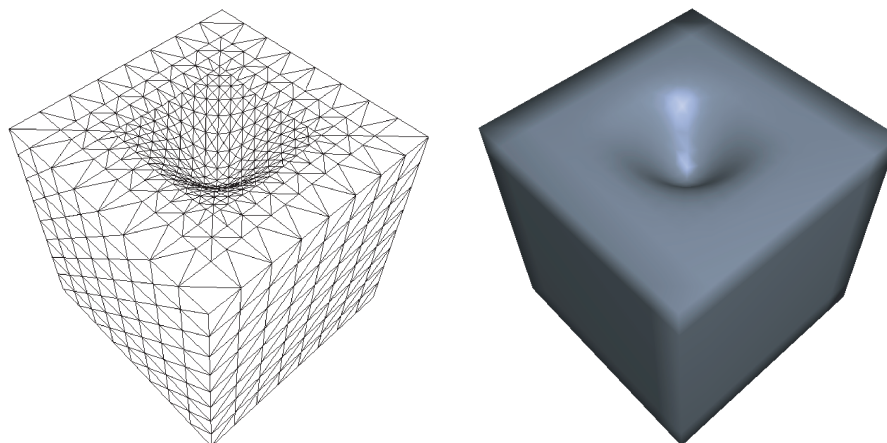
Nejjednodušší operací je přidání detailů do importovaného modelu, který představuje vstupní řídicí síť, na kterou zatím nebyl dělicí algoritmus aplikován. Protože před provedením této operace je nutné model v závislosti na měřítku přidávaných detailů několikrát rozdělit, velmi záleží na tom, jaký počáteční tvar upravovaný model má. Konkrétně hraje roli ostrost jeho hran – čím více ostrých hran model má, tím více se bude jeho tvar měnit během prvního, případně i druhého a dalších dělicích kroků. Extrémním případem je krychle, kde každá její strana je tvořena jedním čtyřúhelníkem. V takovém případě dojde k drastické změně modelu nejen při globálním rozdělení, kdy se tvar rychle přibližuje kouli (viz obr. 6.1 vlevo), ale i při lokálním rozdělení. To je vidět na obr. 6.1 uprostřed. Abychom mohli na horní stranu modelu ve tvaru krychle tvořené šesti čtyřúhelníky přidat detaily, musíme ji nejdříve rozdělit. Je vidět, že i když v tomto případě rozdělíme pouze horní stranu, díky malému celkovému počtu čtyřúhelníků, ze kterých je model tvořen, se zásadně změní tvar celého modelu. Tato deformace původního tvaru modelu může být problémem v situaci, kdy chce uživatel zachovat původní tvar krychle.

Ze zkušenosti získané z výše uvedeného příkladu lze říci, že čím větším počtem čtyřúhelníků je vstupní model tvořen, tím menší vliv bude mít lokální rozdělení vybraných čtyřúhelníků na jeho celkový tvar. Rozdělíme model jen tam, kde chceme přidávat detaily a původní tvar modelu na ostatních místech přitom zůstane nezměněn. Problém z předchozího příkladu lze tedy částečně vyřešit tak, že místo krychle tvořené šesti čtyřúhelníky použijeme model



Obrázek 6.1: Vliv globálního (vlevo) a lokálního (uprostřed) rozdělení vstupní řídicí sítě tvořené malým počtem čtyřúhelníků na změnu jejího původního tvaru

krychle, kde každá její strana je tvořena větším počtem čtyřúhelníků (viz obr. `reflImageResultsBox2`, kde každá strana krychle je tvořena sítí  $8 \times 8$  čtyřúhelníků). Pokud budeme chtít na horní stranu této krychle přidat podobné detaily jako v předchozím případě, opět na daném místě model rozdělíme, tentokrát to už ale bude mít vliv na mnohem menší okolí (viz rozdělenou síť na obr. 6.2 vlevo a její vystínovanou podobu vpravo, detaily byly přidány podobně jako na obr. 6.1 vpravo). Problémům se ovšem opět nevyhneme u hran krychle – při rozdělení některého ze čtyřúhelníků ležících u hrany krychle dojde k nežádoucí lokální změně jejího tvaru.



Obrázek 6.2: Omezení vlivu lokálního rozdělení vstupní řídicí sítě zvýšením počtu jejích čtyřúhelníků

Vhodným námětem pro rozšíření implementace programu by bylo přidání podpory pro tzv. ostré hrany (*crease edges*) zmíněné v části 3.3, které by vyřešilo výše zmíněné problémy. S jejich pomocí by bylo možné už v prvním příkladu s krychlí tvořenou pouze šesti čtyřúhelníky označit všechny její hrany jako ostré a krychle by pak při dělení neměnila svůj tvar.

## 6.2 Modelování na úrovni vstupní řídicí sítě

Pomocí vytahování čtyřúhelníků řídicí sítě lze poměrně snadno k upravovanému modelu přidávat jednoduché části a měnit tak jeho globální tvar. Následující příklad demonstruje, jak lze k modelu vázy přidat úchop, k jehož vytvarování stačí manipulace s vrcholy vstupní řídicí sítě. Tento příklad dobře ukazuje, jak snadno mohou dělené povrchy nahradit klasické parametrické plochy díky podobnému vztahu mezi vrcholy řídicí sítě a tvarem výsledného povrchu.

Na obr. 6.3 vlevo je zdrojový model vázy, ke které byl pomocí operace vytahování čtyřúhelníků řídicí sítě vytvořen úchop (na obrázku uprostřed je vidět v neúplném stavu), a vpravo je vidět hotový a rozdělený model. Změna tvaru modelu v takovémto rozsahu by jen pouhým přidáním detailů k původnímu tvaru vázy bez změny topologie vstupní řídicí sítě nebyla prakticky proveditelná. Zdrojový model je uložen v souboru `vase.qtm`, hotový model je uložen v souboru `vase_handle_final.qtm`.



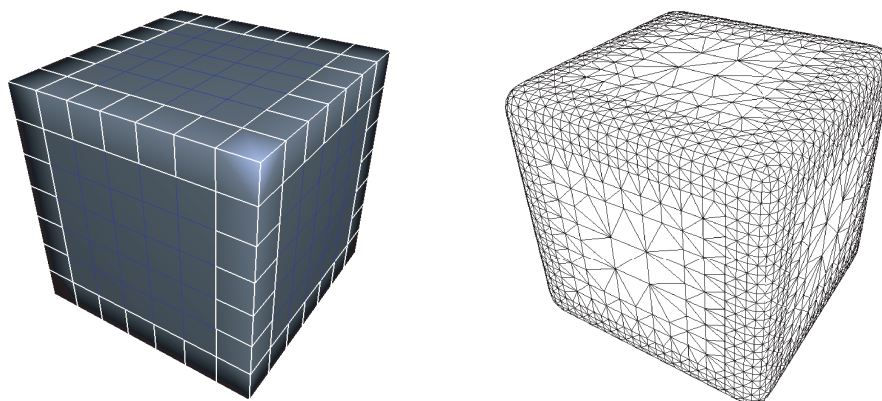
Obrázek 6.3: Přidání úchopu k modelu vázy (změnou topologie vstupní řídicí sítě)

## 6.3 Rychlé přidání detailů pomocí metody „copy & paste“

I přes omezení uvedená v části 5.6.4 je možné použitou implementaci „copy & paste“ metody využít k výraznému urychlení práce v situaci, kdy chceme do modelu přidat na více míst stejné detaily. Následující popis ukazuje, jak jednoduše vytvořit model hrací kostky.

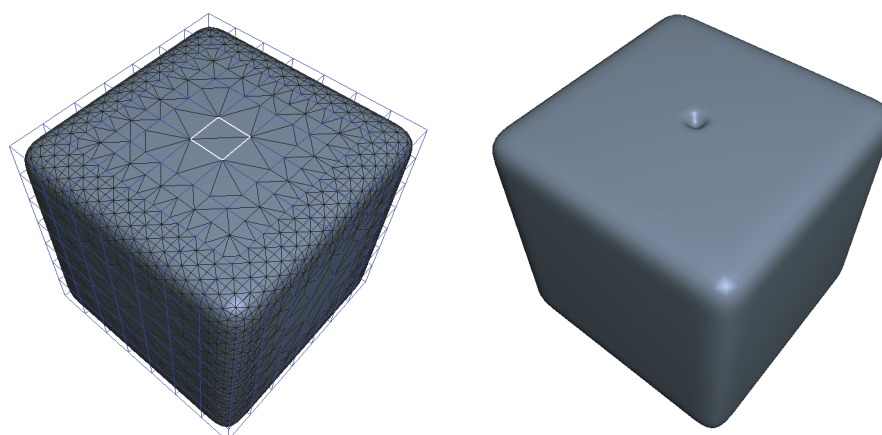
1. Základem bude model krychle se stranami tvořenými sítí  $7 \times 7$  čtyřúhelníků. Je možné ho nahrát ze souboru `dice_1.qtm` nebo vytvořit pomocí panelu s ovládacími prvky v pravé části hlavního okna editoru.
2. Hrany krychle jsou příliš ostré, vyhladit je můžeme rozdělením čtyřúhelníků, které leží v jejich okolí. Nastavte aktivní hloubku na 0 (hodnota

*Active depth* v horní části panelu s ovládacími prvky) a hodnotu *Selection mode* na *Quad* (na stejném místě), vyberte čtyřúhelníky v okolí všech hran krychle (viz obr. 6.4 vlevo) a rozdělte je aspoň dvakrát (stisknutím klávesy **s**). Výsledkem je krychle se zaoblenými hranami (viz obr. 6.4 vpravo). Výsledek tohoto kroku je k dispozici v souboru `dice_2.qtm`.



Obrázek 6.4: První krok postupu pro vytvoření modelu hrací kostky

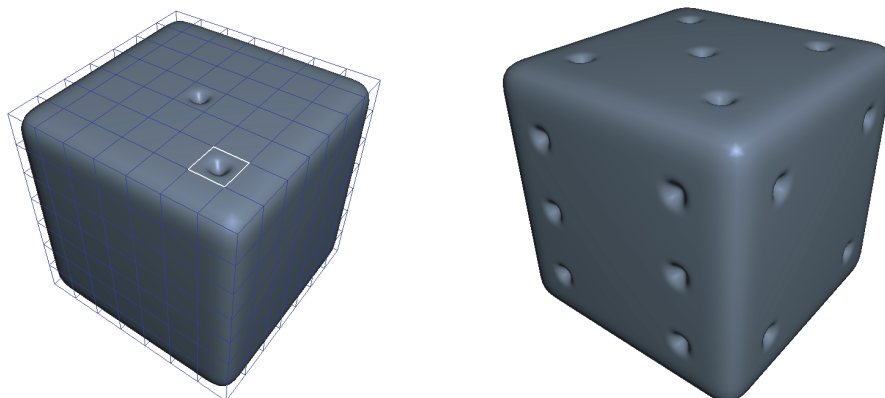
3. Vyberte prostřední čtyřúhelník jedné ze stran krychle (viz obr. 6.5 vlevo) a také ho aspoň dvakrát rozdělte (klávesa **s**). Manipulací s vrcholy tohoto rozděleného čtyřúhelníku (na obrázku to bylo na úrovni 2 a 3) vytvarujte důlek (viz obr. 6.5 vpravo). Výsledek tohoto kroku je k dispozici v souboru `dice_3.qtm`.



Obrázek 6.5: Druhý krok postupu pro vytvoření modelu hrací kostky

4. Změňte aktivní hloubku na 0, označte opět čtyřúhelník, na kterém jste vytvořili důlek a stiskněte klávesu **c**. Tím je vybraný čtyřúhelník označen

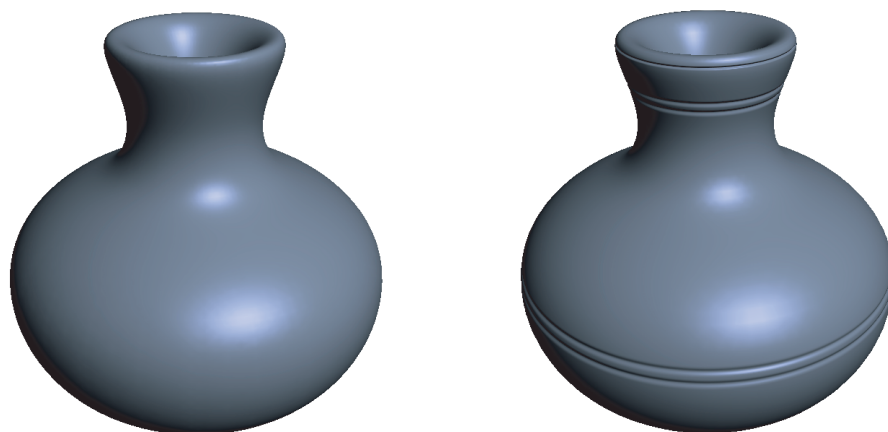
jako zdrojový pro kopírování detailů. Označte jiný čtyřúhelník a stiskněte klávesu `p`. Detaily ze zdrojového čtyřúhelníku se zkopírují na aktuálně vybraný čtyřúhelník (viz obr. 6.6 vlevo). Opakováním rychle vytvoříte kompletní hrací kostku (viz obr. 6.6 vpravo). Konečný výsledek je k dispozici v souboru `dice_4.qtm`.



Obrázek 6.6: Poslední krok postupu pro vytvoření modelu hrací kostky

Podobně lze upravovat jakékoliv modely, ovšem s jejich rostoucí složitostí začíná být práce s nimi za použití změny pozice jednotlivých vrcholů zdlouhavá. Na obr. 6.7 vlevo je vidět původní, několikrát rozdělený model vázy. Na stejném obrázku vpravo je vidět stejný model vázy po přidání detailů pomocí metody „copy & paste“ podobným způsobem jako při vytváření modelu hrací kostky – tedy detail rýhy byl ručně vytvořen pro jeden čtyřúhelník a zbytek vznikl zkopírováním. Na takto velkém modelu už začíná být vkládání detailů zdlouhavé, jelikož vytvoření rýhy po celém obvodu vázy bylo nutné provádět vkládáním kopírovaných detailů po jednotlivých čtyřúhelnících. Model vázy s přidávanými detaily je k dispozici v souboru `vase_details.qtm`, zdrojový model původní vázy v souboru `vase.qtm`.

Na předchozích postupech byly demonstrovány s výjimkou operace mazání čtyřúhelníků vstupní řídicí síť všechny schopnosti vytvořeného editoru. Výběr implementovaných modelovacích operací bohužel není tak široký, aby bylo možné vytvořit výrazně složitější a rozsáhlejší úpravy než ty ve výše uvedených postupech, ale pro základní vyzkoušení modelování s dělenými povrchy je dostačující.



Obrázek 6.7: Přidání detailů na model vázy



# Kapitola 7

## Závěr

V rámci práce se podařilo ukázat, že dělené povrchy jsou zajímavou volbou pro použití v aplikacích interaktivního modelování i bez zapojení pokročilého matematického aparátu nad rámec základních vztahů, který je jinak při aplikaci dělených povrchů v modelování často přítomen. Práce s nimi je po vzoru parametrických ploch intuitivní a navíc nabízejí schopnost reprezentovat víceúrovňovou hierarchii detailů, která je základem pro víceúrovňové modelování.

Implementované operace s dělenými povrchy se omezily jen na základní manipulaci s jednotlivými vrcholy a jednoduché změny topologie vstupní řídicí sítě, ovšem všechny pracují s víceúrovňovou hierarchii detailů, která je vytvářena při přidávání detailů do vstupní řídicí sítě modelu a je v průběhu veškerých úprav modelu udržována. Díky této vlastnosti bylo možné implementovat jednoduchou metodu typu „copy & paste“, která umožňuje kopírování vybraných částí modelu a jejich vkládání na jiná místa. Ačkoliv je její použití principiálně omezené na části sítě se shodnou topologií, ukazuje, jak jednoduše lze využít víceúrovňové hierarchie pro rychlou a účinnou úpravu modelu.

Díky tomu, že jsme se omezili pouze na čtyřúhelníkové řídicí sítě, vyhlí jsme se potenciálním problémům, které jsou v případě Catmull-Clarkových dělených povrchů spojeny se sítěmi tvořenými tvarově nevhodnými polygony, které při dělení mohou způsobovat vytváření artefaktů. Na druhou stranu jsme tím omezili množinu použitelných vstupních dat. V praxi je jen velmi málo polygonových modelů dostupných ve výhradně čtyřúhelníkové variantě, i když mnoho reálných objektů lze s využitím výhradně čtyřúhelníků vymodelovat. Pro vytvoření testovacích dat bychom mohli využít nějaké komerční implementace Catmull-Clarkových dělených povrchů, která umí pracovat s libovolnou vstupní řídicí sítí. Po prvním rozdělení jakékoliv polygonové sítě totiž Catmull-Clarkův algoritmus vyprodukuje síť tvořenou výhradně čtyřúhelníky a tuto síť bychom pak mohli použít jako vstupní řídicí síť v naší aplikaci. Nevýhodou je i v tomto případě to, že hlavně první dělicí krok má největší vliv na změnu tvaru původního modelu.

Omezení na čtyřúhelníky by se také mohlo stát překážkou v okamžiku, kdy bychom chtěli provádět nějaké zásadnější operace se vstupní řídicí sítí, např.

odstraňování vybraných částí „řezáním“ nebo kopírování a vkládání vybraných částí modelu na úrovni vstupní řídicí sítě. V takovém případě bychom sice mohli se vstupní řídicí sítí pracovat jako s klasickou polygonovou sítí, ale při tomto typu operací nevyhnutelně dochází k rozdělování a různým změnám tvaru jednotlivých polygonů a to je bez možnosti použití trojúhelníků obtížné.

## Náměty pro další práci

- Při práci s většími modely se začne projevovat časová náročnost práce se stromovou strukturou, což je ještě umocněno v případě, kdy upravovaný model obsahuje velké množství různě jemných detailů a je proto rozdělen do větších hloubek. K urychlení práce s modelem by výrazně přispělo, kdyby se informace o sousednosti čtyřúhelníků vytvořených dělením vstupní řídicí sítě nezjišťovala procházením stromové struktury, jak je tomu nyní, ale explicitně ukládala do odpovídajících datových struktur a aktualizovala pouze v případě potřeby, i za cenu podstatného zvýšení paměťových nároků, které by to přineslo.
- Přidání podpory pro zmiňované ostré hrany (*crease edges*) by výrazně zvýšilo efektivitu využití vlastností dělených povrchů a poskytlo novou množinu tvarů, které by bylo možné modelovat.
- Odstranění topologického omezení při kopírování vybraných úseků sítě a umožnění kopírování většího úseku sítě najednou než jednoho čtyřúhelníku.
- Současně s přidáváním dostupných operací by bylo nutné rozšiřovat možnosti uživatelského rozhraní editoru (v aktuální podobě velice chybí možnost výběru více vrcholů nebo čtyřúhelníků najednou, např. pomocí obdélníkového výběru), bez kterého jsou pro uživatele všechny implementované operace bezcenné.
- V souvislosti s předchozím bodem by stálo za úvahu použití nějakého volně dostupného „frameworku“ pro vytváření grafických vizualizačních aplikací, čímž by se ušetřil čas strávený při implementaci vlastních prvků grafického rozhraní, které již jsou někde hotové k dispozici.

# Literatura

- [1] BHATTACHARYA, P. *Efficient Neighbor Finding Algorithms in Quadtree and Octree*. M. T. Thesis, Dept. Comp. Science and Eng., India Inst. Technology, Kanpur, 2001.
- [2] BIERMANN, H., MARTIN, I., BERNARDINI, F., ZORIN, D. *Cut-and-Paste Editing on Multiresolution Surfaces*. Proceeding of SIGGRAPH 2002, p. 312–321, 2002.
- [3] BIERMANN, H., MARTIN, I., ZORIN, D., BERNARDINI, F. *Sharp features on multiresolution subdivision surfaces*. Proceeding of Pacific Graphics 2001, 2001.
- [4] BOTSCH, M., PAULY, M., RÖSSL, C., BISCHOFF, S., KOBELT, L. *Geometric Modeling Based on Triangle Meshes*. Eurographics Tutorial Notes, September 2006.
- [5] CATMULL, E., CLARK, J. *Recursively generated B-spline surfaces on arbitrary topological meshes*. Computer-Aided Design 10, p. 350-355, September 1978.
- [6] DEROSE, T., KASS, M., TRUONG, T. *Subdivision Surfaces in Character Animation*. Proceedings of SIGGRAPH 98, p. 85–94, July 1998.
- [7] FRISKEN, S., PERRY, R. *Simple and Efficient Traversal Methods for Quadtrees and Octrees*. Mitsubishi Electric Research Laboratories.
- [8] HALSTEAD, M., KASS, M., DEROSE, T. *Efficient, Fair Interpolation Using Catmull-Clark Surfaces*. Proceedings of SIGGRAPH 93, p. 35–44, August 1993.
- [9] KRISTJANSSON, D., BIERMANN, H., ZORIN, D. *Approximate boolean operations on free-form solids*. Proceedings of SIGGRAPH 2001, August 2001.
- [10] LITKE, N., LEVIN, A., SCHRÖDER, P. *Trimming for subdivision surfaces*. CAGD, 2001.
- [11] MUELLER, H., JAESCHKE, R. *Adaptive subdivision curves and surfaces*. Proceedings of Computer Graphics International '98, p. 48–58, 1998.

- [12] SEDERBERG, T. W., ZHENG, J., BAKENOV, A., NASRI, A. *T-splines and T-NURCCs*. ACM Transactions on Graphics, Vol. 22, Issue 3, p. 477–484, 2003.
- [13] UHLÍŘ, K. *Interaktivní systém pro generování implicitních funkcí a jejich modelování*. Diplomová práce, Katedra informatiky a výpočetní techniky, ZČU, Plzeň, 2001.
- [14] ULLER, M. *Modeling of Free-form Shapes by Sketching*. Master Thesis, Department of Computer Science and Engineering, ČVUT, Praha, 2005.
- [15] WAI KIT ADDY NGAN *Subdivision Models in a Freeform Sketching System*. Bachelor Thesis, Department of Computer Science of Princeton University, May 2001.
- [16] XU, Z., KONDO, K. *Adaptive refinements in subdivision surfaces*. Eurographics '99, Short papers and demos, 239-242, 1999.
- [17] ZORIN, D. *Subdivision on Arbitrary Meshes: Algorithms and Theory*. Lecture Notes, February 2005.
- [18] ZORIN, D., SCHRÖDER, P., DEROSE, T., KOBELT, L., LEVIN, A., SWELDENS, W. *Subdivision for modeling and animation*. ACM SIGGRAPH 2000 Course Notes, 2000.
- [19] ZORIN, D., SCHRÖDER, P., SWELDENS, W. *Interactive multiresolution mesh editing*. Proceedings of SIGGRAPH 97, p. 259–268, 1997.
- [20] ŽÁRA, J., BENEŠ, B., SOCHOR, J., FELKEL, P. *Moderní počítačová grafika*. Brno, Computer Press, 2004. ISBN 80-251-0454-0.

# Příloha A

## Definice použitých termínů

**Aproximace vstupní sítě:** Polygonovou síť vytvořenou aplikováním  $n$  kroků dělicího algoritmu na vstupní polygonovou (řídící) síť nazýváme  $n$ -tou aproximací vstupní polygonové (řídící) sítě.

$C^n$  (**parametrická**) **spojitost:** Říkáme, že křivka  $Q(t)$  je  $C^n$  spojitá, pokud má spojitě derivace podle parametru  $t$  až do  $n$ -tého řádu. Již  $C^1$  spojitost zaručuje *vizuálně* hladké napojení. Analogická definice platí pro parametrické plochy, kde každý bod na ploše je určen hodnotami dvou parametrů, takže podmínka musí být splněna pro oba parametry.

**Dělené povrchy** (subdivision surfaces): Povrchy generované rekurzivním dělením polygonů vstupní řídící polygonové sítě, jehož výsledkem je vyhlazená podoba vstupní sítě (stupeň vyhlazení je přímo závislý na počtu aplikovaných dělicích kroků).

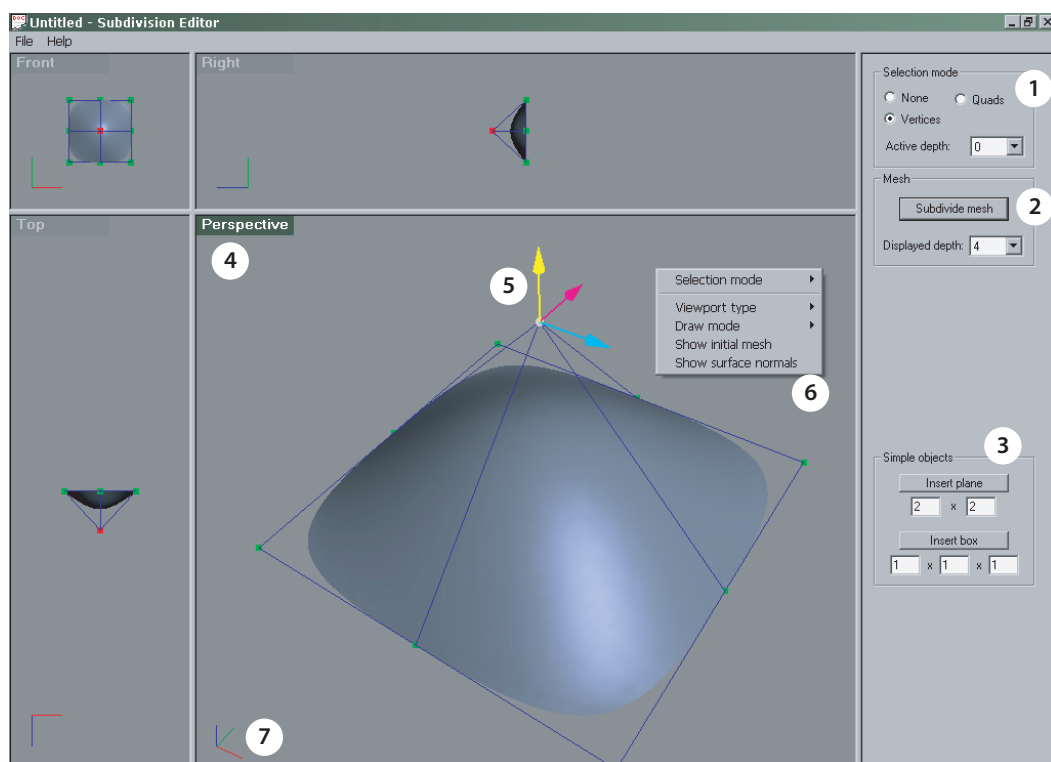
**Dělicí schéma** (dělicí algoritmus): Popisuje způsob, jakým jsou s použitím stávajících vrcholů z lokálního okolí vytvářeny nové vrcholy při dělení polygonové sítě.

**Maska dělicího schématu:** Obrázek znázorňující pro konkrétní dělicí algoritmus (schéma) ty vrcholy z lokálního okolí místa, na kterém vytváříme nový vrchol, které mají vliv na výpočet polohy nového vrcholu. U každého vrcholu masky je uvedena váha vyjadřující jeho vliv na výslednou polohu nového vrcholu.

**Víceúrovňové modelování** (multiresolution modeling): Je použitelné ve spojení s takovou povrchovou reprezentací, která dokáže odděleně zachytit hierarchii více úrovní detailů upravovaného modelu najednou. Při provádění úprav modelu na hrubší úrovni (ve větším měřítku) zachovává detaily na jemnějších úrovních.

# Příloha B

## Uživatelský manuál



Obrázek B.1: Hlavní okno editoru

Na obr. B.1 jsou vyznačeny jednotlivé prvky grafického rozhraní editoru. Okno je rozděleno na dvě hlavní části – vpravo je panel s ovládacími prvky, levá část obsahuje čtyři výhledy, které umožňují prohlížení tvaru modelu z různých směrů.

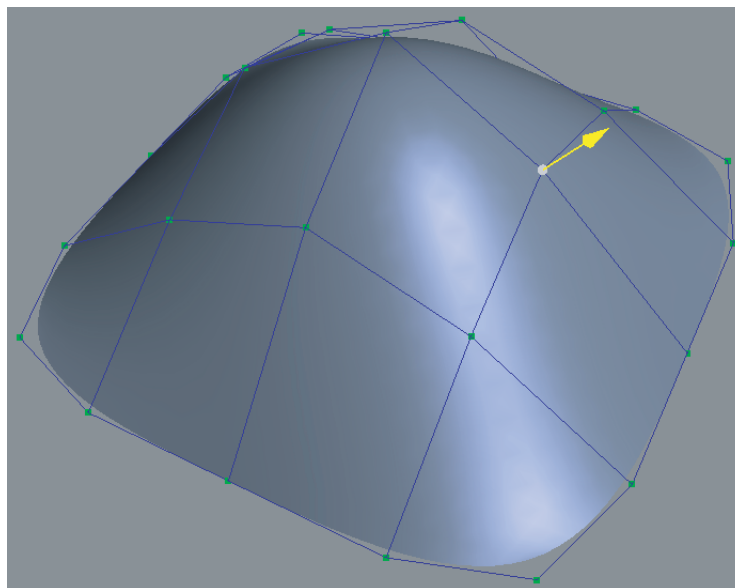
V jednom okamžiku je možné pracovat buď s vrcholy, nebo se čtyřúhelníky upravovaného modelu. Režim výběru se nastavuje v pravém panelu nahoře (číslo 1). Na stejném místě je možné nastavit aktivní hloubku, na které jsou

prováděny úpravy. Hloubka 0 odpovídá vstupní řídicí síti, další hloubky jsou dostupné podle stupně rozdělení modelu. V pravém panelu je dále tlačítka pro globální rozdělení modelu (*Subdivide mesh*) a možnost změny aktuální zobrazené hloubky modelu (*Displayed depth*, číslo 2). Pravý panel ještě nabízí možnost vložení jednoduchých objektů do scény (rámeček *Simple objects*). Jejich vložení dojde k nahrazení stávajícího modelu (číslo 3).

Stisknutí pravého tlačítka myši nad některým z výhledů vyvolá zobrazení kontextové nabídky (číslo 6), ve které jsou k dispozici volby pro změnu režimu výběru (stejně jako volby v pravém panelu), nastavení typu daného výhledu (aktuální typ je zobrazen v levém horním rohu výhledu, číslo 4), nastavení režimu zobrazení v daném výhledu (vystínovaný model, drátový model a kombinace obou předchozích) a povolení zobrazení drátěného modelu vstupní řídicí sítě a normálových vektorů ve vrcholech aktuálně zobrazené hloubky (*Displayed depth* v pravém panelu). V levém dolním rohu každého pohledu jsou zobrazeny směry os odpovídající poloze pomyslné kamery, která scénu v daném pohledu snímá (číslo 7).

Pokud je aktivní režim výběru vrcholů nebo čtyřúhelníků, lze vybírat jednotlivé vrcholy nebo čtyřúhelníky levým tlačítkem myši. Držením klávesy **Shift** je možné vybrat více vrcholů nebo čtyřúhelníků najednou. V režimu výběru vrcholů je možné měnit pozici vybraných vrcholů taháním za barevné šipky (číslo 5). Šipky odpovídají směrovým vektorům lokálního souřadnicového systému vybraných vrcholů. Vrcholy vstupní řídicí sítě je možné posouvat v osách  $x$ ,  $y$  a  $z$  kartézského souřadnicového systému, vrcholy vytvořené při rozdělování vstupní řídicí sítě je možné posouvat podél jim odpovídajících normálových vektorů (viz obr. B.2). Rozdíl mezi změnou polohy vrcholů řídicí sítě a vrcholů vytvořených při dělení je ten, že všechny vrcholy vstupní řídicí sítě jsou posouvány v globálním souřadnicovém systému, zatímco každý vrchol vytvořený při dělení je posouván podle svého vlastního normálového vektoru. Šipky pro posun vybraných vrcholů se zobrazují jen v perspektivním pohledu.

Editor umožňuje ukládat upravené modely (menu *File* → *Save*) ve vlastním formátu (přípona `.qtm`). Použití vlastního formátu je nutné pro uložení informací o rozdělení jednotlivých čtyřúhelníků vstupní řídicí sítě a informací o přidávaných detailech. Kromě toho je možné importovat čtyřúhelníkové modely ve formátu Wavefront Object (přípona `.obj`), které pak slouží jako vstupní řídicí síť modelu, na kterém budou v editoru prováděny úpravy. Jiné než čtyřúhelníkové modely importovat nelze. Upravené modely lze exportovat jako obyčejnou polygonovou síť ve čtyřúhelníkové nebo trojúhelníkové variantě, opět ve Wavefront Object formátu. Pokud je rozdělení modelu nerovnoměrné, může čtyřúhelníková varianta exportované sítě obsahovat trhliny mezi čtyřúhelníky z různých hloubek dělicí hierarchie. V trojúhelníkové variantě trhliny nikdy nejsou.



Obrázek B.2: Nástroj pro změnu polohy vrcholu podél jeho normálového vektoru

## B.1 Orientace ve výhledech tažením kurzoru myši

Stisknuto při tažení	Typ projekce	
	Perspektivní	Pravoúhlá
PTM <sup>1</sup>	natočení modelu	viz Shift + PTM
Ctrl + PTM	přiblížení/oddálení kamery	změna rozměrů projekční roviny
Shift + PTM	horizontální/vertikální posun kamery	

<sup>1</sup>Pravé tlačítko myši.



## B.2 Ovládání klávesnicí a myši

Tyto příkazy jsou použitelné pouze pokud je aktivní některý z výhledu v levé části okna aplikace (jeho název v levém horním rohu je zvýrazněn).

kolečko myši	změna aktivní hloubky
Ctrl + kolečko myši	změna zobrazené hloubky
+	zvýšení aktivní hloubky (pokud není nastavena na maximální)
-	snížení aktivní hloubky (pokud není nastavena na 0)
u	vypnutí režimu výběru vrcholů nebo čtyřúhelníků
i	zobrazení vrcholů z aktivní hloubky a povolení jejich výběru
o	zobrazení sítě čtyřúhelníků z aktivní hloubky a povolení jejich výběru
s	rozdělení vybraných čtyřúhelníků
Shift + s	rozdělení celého modelu do hloubky následující po aktuální maximální hloubce rozdělení
d	odstranění vybraných čtyřúhelníků (pouze pro čtyřúhelníky vstupní řídicí sítě)
e	vytažení vybraného čtyřúhelníku vstupní řídicí sítě (čtyřúhelník ani žádný z jeho sousedů nesmí být rozdělen)
c	kopírování detailů z vybraného (zdrojového) čtyřúhelníku do schránky
p	vložení detailů ze schránky do vybraného čtyřúhelníku (hloubka vybraného čtyřúhelníku musí být shodná s hloubkou zdrojového čtyřúhelníku)