Univerzita Komenského
Fakulta matematiky, fyziky a informatiky

Triangulácie v rovine, teréne a priestore

Dizertačná práca

2010 RNDr. Zsolt Tóth

Univerzita Komenského

Fakulta matematiky, fyziky a informatiky

Katedra algebry, geometrie a didaktiky matematiky

Triangulácie v rovine, teréne a priestore

Dizertačná práca

v odbore doktorandského štúdia:
11-16-9 Geometria a topológia

Školiteľ: Doc. RNDr. Andrej Ferko, PhD.
Doktorand: RNDr. Zsolt Tóth

Bratislava 2010

Univerzita Komenského

Fakulta matematiky, fyziky a informatiky

Katedra algebry, geometrie a didaktiky matematiky

Triangulácie v rovine, teréne a priestore

RNDr. Zsolt Tóth

DIZERTAČNÁ PRÁCA

na získanie vedecko-akademickej hodnosti Philosophiae Doctor
v odbore doktorandského štúdia
11-16-9 Geometria a topológia

Bratislava 2010

Prehlásenie

Čestne vyhlasujem, že dizertačnú prácu som vypracoval samostatne, na základe vlastných vedomostí a použitej literatúry.

....

Poďakovanie

Úprimne ďakujem svojmu školiteľovi doc. RNDr. Andrejovi Ferkovi, PhD. za jeho odborné vedenie, cenné pripomienky a motiváciu pri tvorbe tejto práce.

Vďaka patrí aj mojej rodine a priateľom za ich pochopenie a podporu, ktorú mi poskytli.

Abstrakt

Táto práca sa zaoberá špeciálnou podskupinou optimálnych triangulácií, takzvanými dátovo závislými trianguláciami. Jedná sa o techniku, ktorá vďaka svojim vhodným vlastnostiam má širokú škálu uplatnenia. Hlavným zámerom je skúmanie rekonštrukčných vlastností tejto metódy, a hľadanie aproximácií globálne optimálneho stavu, ktorá je NP-ťažkou úlohou. Práca obsahuje popis existujúcich prístupov a predstavuje rôzne vylepšenia a rozšírenia. Hlavnými prínosmi sú rozšírenie dátovo závislej techniky do vyšších dimenzií a nájdenie paralelného prístupu, ktorá vytvára lokálne optimálne triangulácie pomocou grafického hardvéru. Okrem toho sú uvedené rôzne vylepšenia pre rovinné dátovo závislé triangulácie, kombinácia s výsledkami konvolučných techník v prípade aplikačnej oblasti rekonštrukcie obrazu. Je predstavená aj koncepcia kompresie viacrozmerných dát pomocou dátovo závislej techniky. Výsledky sú analyzované z rôznych aspektov, a na základe vyvodených dôsledkov sú stanovené rôzne výskumné smery pre budúcu prácu.

Abstract

This thesis deals with a special subset of optimal triangulations, called data dependent triangulations. This technique has wide usability because of its properties. The main goal is the investigation of the reconstruction behavior of this method, and the approximation of the globally optimal solution, which is NP-hard to find. The thesis contains the summarization of the existing approaches and introduces several improvements and extensions. The main contributions are the following: extension of the data dependent technique for $n-$*dimensional usage; a parallel approach, which uses the graphics hardware for the locally optimal triangulations generation. Besides of these results there are several improvements presented for the planar data dependent triangulations. For image reconstruction usage is presented an approach which combines the results of the convolution techniques with the results from the data dependent methods. A concept for a multidimensional scattered data compression is also introduced. The results are analyzed from different points of view, and several future work ideas are presented.*

Obsah

Zoznam obrázkov

Zoznam tabuliek

Známe skratky

ABN	–	cenová funkcia (angle beetween normals)
$Card(\boldsymbol{V})$	–	mohutnosť množiny $\boldsymbol{V}$
$\boldsymbol{CH}(\boldsymbol{V})$	–	konvexný obal množiny $\boldsymbol{V}$
CPU	–	centrálna procesorová jednotka
CT	–	počítačová tomografia
$c(\sigma_n)$	–	cenová funkcia n-simplexu
$c(\boldsymbol{T}(\boldsymbol{V}))$	–	cena (váha) triangulácie
DDT	–	dátovo závislá triangulácia
DLP	–	cenová funkcia (deviations from linear polynomials)
DM	–	vzdialenostná mapa
DP	–	cenová funkcia (distances from planes)
DT	–	Delaunayova triangulácia
$ExpROI$	–	rozšírená oblasť vplyvu
FEM	–	metóda konečných prvkov
$GLOP$	–	zovšeobecnený Lawsonov optimalizačný proces
GO	–	genetická optimalizácia
$GPGPU$	–	všeobecné výpočty na grafických procesorových jednotkách
GPU	–	grafická procesorová jednotka
$GSCF$	–	zovšeobecnená Sederbergova cenová funkcia
ID	–	identifikačné číslo hrany
JND	–	cenová funkcia (jump in normal derivatives)
LAT	–	look-ahead triangulácia
LOP	–	Lawsonov optimalizačný proces
LOT	–	lokálne optimálna triangulácia
$MaxGain$	–	maximalizácia prínosu
MWT	–	triangulácia s minimálnou váhou
PLC	–	cenová funkcia (piecewise-linear analogy of curvature)
ROI	–	oblasť vplyvu
SA	–	simulované žíhanie
$SALA$	–	simulované žíhanie s look-ahead rozšírením
SCF	–	Sederbergova cenová funkcia
$SCF\text{-}Apr.$	–	aproximácia Sederbergovej cenovej funkcie
$\boldsymbol{T}(\boldsymbol{V})$	–	triangulácia množiny $\boldsymbol{V}$
$\boldsymbol{T_K}$	–	množina topologických transformácií
$w(\sigma_n)$	–	váha n-simplexu
σ_n	–	n-simplex

Kapitola 1

Úvod

Úlohou výpočtovej geometrie je hľadanie algoritmických riešení pre geometrické problémy a poskytnutie efektívnych nástrojov pri riešení praktických úloh z rôznych vedných odborov. Významnú časť tejto vednej disciplíny tvorí skúmanie triangulácií a ich optimalizácia pre rôzne účely. Optimálne triangulácie majú širokú škálu využiteľnosti v počítačovej grafike ako aj v iných oblastiach vedy. Existujúce poznatky sú veľmi rozsiahle, a preto cieľom tejto práce je zamerať sa na špeciálnu podskupinu optimálnych triangulácií, nazývanú *dátovo závislé triangulácie*. Hlavným účelom ich použitia je rekonštrukcia po častiach lineárnych funkcií definovaných nad diskrétnymi dátami. V praxi to znamená, že táto metóda umožňuje väčšiu kontrolu nad vlastnosťami vytváranej triangulácie daného vstupu.

Cieľom tejto práce je rozširovať poznatky o dátovo závislých trianguláciách a sumarizovať existujúce teoretické vedomosti z hľadiska optimálnych triangulácií. Jedným z hlavných dosiahnutých výsledkov je zovšeobecnenie tejto techniky pre účely rekonštrukcie funkcií nad $n-$dimenzionálnou doménou pomocou simpliciálnej dekompozície domény. Ďalej, významná časť je venovaná paralelizácii rovinného prístupu, a to takým spôsobom, aby sa výpočty triangulácie dali vykonať na grafickej karte. Takisto je poskytnutý podrobný popis možnosti ich využitia pre kompresiu dát, ako aj kombinácie dátovo závislej triangulácie a štandardných rekonštrukčných techník pre účely zväčšenia obrazu.

Aplikačná oblasť dátovo závislých triangulácií je veľmi rozsiahla. Napríklad rekonštrukcia digitálneho obrazu (2D), rekonštrukcia medicínskych dát (3D), rekonštrukcia v čase sa meniacich volumetrických dát (4D), využitie pri predspracovaní dát pre rôzne numerické simulácie, získavanie dodatočnej farebnej informácie pri snímacích čipoch digitálnych kamier atď. Vhodné nastavenie tejto metódy ju umožní prispôsobiť ľubovoľnému rekonštrukčnému problému. Okrem iného môže slúžiť aj na vektorizáciu dát. Raz vytvorená triangulácia poskytuje základnú štrukturalizáciu dát, v ktorej je možné získať neznámu funkčnú hodnotu z interpolácie hodnôt z vrcholov trojuholníka.

Práca je členená do šiestich častí. V druhej kapitole sú zavedené základné pojmy, je definovaná úloha rekonštrukcie, popis dátovo závislých triangulácií, ich vzťah k optimálnym trianguláciám a prehľad optimalizačných techník. Tretia kapitola obsahuje popis vlastných rozšírení, analýza výsledkov je uvedená v štvrtej kapitole. Podnety pre ďalší výskum sú obsiahnuté v piatej kapitole. Šiesta kapitola obsahuje záver.

Kapitola 2

Prehľad problematiky

Cieľom druhej kapitoly je zavedenie základných pojmov a poskytnutie prehľadu problematiky. Zoznámenie čitateľa so základmi optimálnych triangulácií a optimalizačných techník releventné pre uvedenú prácu. Okrem iného obsahuje aj popis topologických transformácií, ktoré budú použité v ďalších kapitolách.

Poznámka. V nasledujúcich úvahách sa bude pracovať nad $n-$rozmerným euklidovským priestorom ozn. $\mathbb{E}^n$, ak to nebude ináč vyznačené.

2.1 Optimálne triangulácie

Zadania úloh súvisiace s trianguláciami môžu byť rôzne. Ako vstup sa najčastejšie zadáva množina vrcholov (2D, terén, 3D), mnohouholník, mnohosten alebo rovinný graf s priamymi hranami[1]. Tieto rozlišnosti zapríčinia odlišné konštrukčné postupy pri hľadaní optimálnej triangulácie vstupu. Mnohokrát dodatočná informácia o rozložení, poznatky o vlastností vstupnej štruktúry uľahčia zostrojenie efektívneho prístupu.

V niektorých prípadoch nie je možné nájsť optimálne riešenie pre daný vstup. Ak to zadanie úlohy dovolí, tak sa pridávajú takzvané *Steinerove vrcholy* k pôvodnému vstupu, a následne prebieha konštrukcia optimálnej siete. Častokrát sa využíva tento prístup pri riešení praktických úloh, a takto sa získava aproximácia optimálneho stavu.

Aby sa zúžil záber tejto práce, zoberie sa nasledujúce obmedzenie: vstupom bude množina vrcholov v rovine, teréne alebo v trojrozmernom priestore a použitie Steinerových vrcholov bude zakázané. Potrebné základné definície pre takto formulovanú úlohu vyzerajú nasledovne.

[1] Množina vrcholov a nepretínajúcich sa hrán v rovine. Užitočné napríklad pri simulácií pomocou *metódou konečných prvkov* (*Finite element method*) - ozn. *FEM*), kde sa zoberie do úvahy aj hranica medzi rôznymi materiálmi.

2.1.1 Základné pojmy

Definícia 2.1.1. Konvexná množina $\boldsymbol{V} \in \mathbb{R}^n$ *je množina takých bodov, pre ktoré platí:*

$$pre\ \forall x, y \in \boldsymbol{V}\ platí\ \lambda x + (1-\lambda)y \in \boldsymbol{V},\ \ kde\ \forall \lambda \in \langle 0, 1 \rangle. \tag{2.1}$$

Konvexným obalom *množiny* $\boldsymbol{V}$*, ozn.* $\boldsymbol{CH}(\boldsymbol{V})$*, sa nazýva najmenšia konvexná množina obsahujúca množinu bodov* $\boldsymbol{V}$*.*

Definícia 2.1.2. *Nech* $a_0, a_1, \ldots, a_n \in \mathbb{E}^n$ *sú affine nezávislé body. Potom konvexný obal ľubovoľnej podmnožiny* $\boldsymbol{A} \subseteq \{a_0, a_1, \ldots, a_n\}$ *sa nazýva* k-simplex, *kde* $k = Card(\boldsymbol{A})$ *(mohutnosť množiny* $\boldsymbol{A}$*). Označuje sa ako* $\sigma_k(\boldsymbol{A})$*. Ľubovoľná podmnožina množiny* $\boldsymbol{A}$ *generuje simplex a nazýva sa stenou simplexu* $\sigma_k(\boldsymbol{A})$*. V prípade, že množina* $\boldsymbol{A}$ *je zrejmá z kontextu, použije sa označenie* σ_k*.*

Definícia 2.1.3. *Množina simplexov v* $\mathbb{E}^n$ *sa nazýva* simpliciálny komplex*, ozn.* $\mathcal{C}$*, ak spĺňa nasledujúce vlastnosti:*

(i) Stena ľubovoľného simplexu z $\mathcal{C}$ *patrí do* $\mathcal{C}$*.*

(ii) Ak $\sigma_k(\boldsymbol{A}), \sigma_l(\boldsymbol{B}) \in \mathcal{C}$*,* $0 \leq k, l \leq n$*, tak platí* $\sigma_k(\boldsymbol{A}) \cap \sigma_l(\boldsymbol{B}) = \sigma_m(\boldsymbol{A} \cap \boldsymbol{B})$*, kde* $k = Card(\boldsymbol{A})$*,* $l = Card(\boldsymbol{B})$*,* $m = Card(\boldsymbol{A} \cap \boldsymbol{B})$*.*

Definícia 2.1.4. *Nech je daná množina* $\boldsymbol{V} \in \mathbb{E}^n$*.* Trianguláciou množiny $\boldsymbol{V}$ *sa nazýva simpliciálny komplex* $\mathcal{C}$*, ktorý spĺňa nasledujúce vlastnosti:*

(i) Každá $0-$*stena (vrchol)* $\mathcal{C}$ *je bod z* $\boldsymbol{V}$*.*

(ii) Priestor pokrytý $\mathcal{C}$ *je* $\boldsymbol{CH}(\boldsymbol{V})$*.*

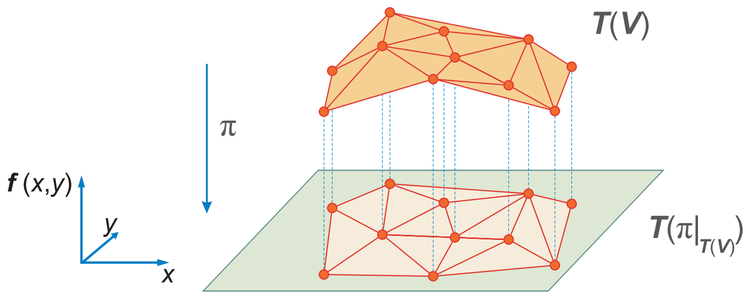
Triangulácia množiny $\boldsymbol{V}$ *bude sa označovať* $\boldsymbol{T}(\boldsymbol{V})$*.*

Definícia 2.1.5. *Nech je daná množina* $\boldsymbol{V} \in \mathbb{E}^{n+1}$ *a* $\pi : \mathbb{E}^{n+1} \to \mathbb{E}^n$*. Ak* $\pi|_{\boldsymbol{V}}$ *je bijekciou, tak dvojica* $(\pi, \boldsymbol{V})$ *sa nazýva* terénom.

Definícia 2.1.6. *Nech je daný terén* $(\pi, \boldsymbol{V})$ *v* $\mathbb{E}^{n+1}$*, a simpliciálny komplex nad množinou* $\boldsymbol{V}$ *označené ako* $\mathcal{C}(\boldsymbol{V})$*. Ak* $\pi|_{\mathcal{C}(\boldsymbol{V})}$ *je trianguláciou množiny* $\pi(\boldsymbol{V})$*, tak* $\mathcal{C}(\boldsymbol{V})$ *sa nazýva* trianguláciou terénu *a označuje sa ako* $\boldsymbol{T}(\boldsymbol{V})$*.*

Poznámka. Z hľadiska topológie triangulácia terénu a triangulácia $\pi|_{\mathcal{C}(\boldsymbol{V})}$ je identická. Triangulácia terénu je vlastne po častiach lineárna plocha nad $\boldsymbol{T}(\pi|_{\mathcal{C}(\boldsymbol{V})})$. Situácia pre $\pi : 3D \to 2D$ je znázornená na Obrázku 2.1. V literatúre tento typ terénu častokrát vystupuje pod menom 2.5D terén. Z matematického hľadiska to nie je korektné, skôr ide o pomenovanie, ktoré intuitívne naznačuje predpis π. Existujú aj oblasti, kde sú definované takéto priestory matematicky korektne, ako napríklad fraktálna geometria, pričom význam je iný.

Kvôli odhadu efektívnosti algoritmov je potrebné si zaviesť nasledujúce pojmy:



Obr. 2.1: Po častiach lineárna plocha $\boldsymbol{T}(\boldsymbol{V})$ nad doménou $\boldsymbol{T}(\pi|_{\boldsymbol{T}(\boldsymbol{V})})$. $\boldsymbol{T}(\boldsymbol{V})$ je trianguláciou terénu, ktorá je určená množinou $\boldsymbol{V}$ a zobrazením π.

Definícia 2.1.7. *Nech $f : \mathbb{N} \to \mathbb{R}$. Symbolom $\mathcal{O}(f(n))$ sa označuje:*

$$\{g : \mathbb{N} \to \mathbb{R} : \exists n_0 \in \mathbb{N} \text{ a } \exists c \in \mathbb{R}^+, \text{že pre } \forall n \geq n_0 : |g(n)| \leq c \cdot f(n)\}. \tag{2.2}$$

Definícia 2.1.8. (Asymptotická horná hranica zložitosti)
Algoritmus má hornú hranicu zložitosti $\mathcal{O}(f(n))$, ak funkcia $g(n)$ počtu jeho operácií v najhoršom možnom prípade je z $\mathcal{O}(f(n))$.

Definícia 2.1.9. *Nech $f : \mathbb{N} \to \mathbb{R}$. Symbolom $\Omega(f(n))$ sa označuje:*

$$\{g : \mathbb{N} \to \mathbb{R} : \exists n_0 \in \mathbb{N} \text{ a } \exists c \in \mathbb{R}^+, \text{že pre } \forall n \geq n_0 : |g(n)| \geq c \cdot f(n)\}. \tag{2.3}$$

Definícia 2.1.10. (Asymptotická dolná hranica zložitosti)
Algoritmus má dolnú hranicu zložitosti $\Omega(f(n))$, ak funkcia $g(n)$ počtu jeho operácií v najlepšom možnom prípade je z $\mathcal{O}(f(n))$.

Špeciálne pre rovinné triangulácie platia nasledujúce vzťahy:

Veta 2.1.1. *Dolná hranica časovej zložitosti ľubovoľnej triangulácie N bodov v rovine je $\Omega(NlogN)$.*

Dôkaz. (náčrt) Vyplýva to z dolnej hranice zložitosti usporiadania N reálnych čísel, pre ktoré platí odhad $\Omega(NlogN)$.

Poznámka. Ak na vstupe je jednoduchý polygón, tak dolná hranica sa zníži na $\Omega(N)$. Viac informácií o triangulácií jednoduchých polygónov sa nachádza v [LZ99].

Tvrdenie 2.1.1. *Nech **V** je množinou vrcholov, **H** je množinou hrán a **T** je množinou trojuholníkov. Nech **B** je množina vrcholov z **V** ležiacich na hranici **CH(V)**. Počet hrán a trojuholníkov ľubovoľnej triangulácie danej množiny v rovine je rovnaký a platí:*

$$\begin{aligned} Card(\boldsymbol{H}) &= 3 \cdot (Card(\boldsymbol{V}) - 1) - Card(\boldsymbol{B}), \\ Card(\boldsymbol{T}) &= 2 \cdot (Card(\boldsymbol{V}) - 1) - Card(\boldsymbol{B}), \end{aligned} \tag{2.4}$$

kde $Card()$ značí mohutnosť množiny.

Dôkaz. Dôkaz tvrdenia vyplýva z Eulerovej formuly a možné ho nájsť v [BY05].

Pre trojrozmerné triangulácie, často nazývané aj *tetrahedralizácie*, platí nasledujúci vzťah:

Tvrdenie 2.1.2. *Nech $\boldsymbol{V}$ je množinou vrcholov v $\mathbb{E}^3$, $\boldsymbol{H}$ je množinou hrán, $\boldsymbol{t}$ je množinou trojuholníkov a $\boldsymbol{T}$ je množinou tetrahedrónov (3-simplexov). Nech $\boldsymbol{B}$ je množina vrcholov z $\boldsymbol{V}$ ležiacich na hranici $\boldsymbol{CH(V)}$. Platí nasledujúce ohraničenie pre ľubovoľnú trianguláciu danej množiny v priestore:*

$$Card(\boldsymbol{V}) - 3 \leq Card(\boldsymbol{T}) \leq \binom{Card(\boldsymbol{V}) - 1}{2} - Card(\boldsymbol{B}) + 2, \tag{2.5}$$

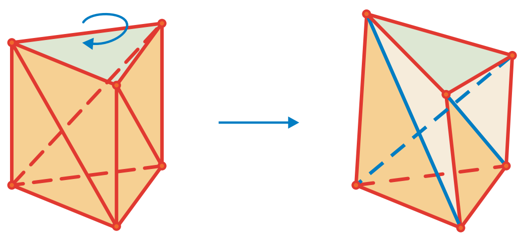
kde $Card()$ značí mohutnosť množiny.

Dôkaz. Dôkaz tvrdenia vyplýva z Eulerovej formuly platiace pre triangulácie v trojrozmernom priestore:

$$Card(\boldsymbol{V}) - Card(\boldsymbol{H}) + Card(\boldsymbol{t}) - Card(\boldsymbol{T}) = 1, \tag{2.6}$$

a možné nájsť v [EPW90].

Na rozdiel od rovinných triangulácií vo vyšších dimenziách nemusí vždy existovať triangulácia daného vstupu. Ako príklad najčastejšie sa uvádza tzv. *Schönhardtov mnohosten* [Sch28], ktorý sa nedá triangulovať (tetrahedralizovať). Tento mnohosten sa konštruuje pomocou trojbokého hranolu, ktorého trojuholníkové podstavy sú rotované takým spôsobom, aby splnili nasledujúci kritérium. Každý vrchol je incidentný (spojený s hranou) so štyrmi ďalšími vrcholmi, ale diagonála k zostávajúcemu vrcholu vždy leží mimo vzniknutého mnohostena. Teda nie je možné spojiť vrcholy, ktoré nie sú incidentné, takým spôsobom, aby takto vzniknutá diagonála prechádzala vnútrom mnohostena. Ilustrácia Schönhardtovho mnohostenu sa nachádza na Obrázku 2.2.



Obr. 2.2: Spôsob vytvárania Schönhardtovho mnohostenu.

Tento mnohosten sa ale dá triangulovať pomocou pridaním jedného Steinerovho vrchola. Vo všeobecnosti platí nasledujúca veta:

Veta 2.1.2. *Ľubovoľný mnohosten je možné triangulovať pomocou $\mathcal{O}(n^2)$ Steinerových vrcholov a $\mathcal{O}(n^2)$ $3-$simpexov (tetrahedronov).*

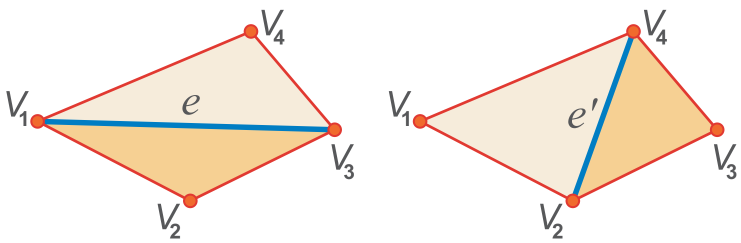
Dôkaz. Dôkaz tvrdenia sa nachádza v [BE95].

Topologické transformácie v trianguláciách

Ďalšie pojmy, ktoré treba zaviesť, sú transformácie triangulácií. Najprv sa zadefinujú transformácie týkajúce sa rovinných triangulácií.

Definícia 2.1.11. *Nech je daná triangulácia množiny $\boldsymbol{V} \in \mathbb{E}^2$, a v nej štvoruholníky vytvorené z dvoch trojuholníkov triangulácie $\boldsymbol{T(V)}$, ktoré obsahujú spoločnú hranu ako svoju diagonálu.* Striktne konvexným štvoruholníkom *v takejto štruktúre sa nazýva konvexný štvoruholník, v ktorom všetky trojice bodov sú nekolineárne.*

Definícia 2.1.12. *Nech platia predpoklady z predchádzajúcej definície. Topologická transformácia, ktorá vymení diagonály v striktne konvexnom štvoruholníku sa bude nazývať* preklápaním *(flip). Situácia je znázornená na Obrázku 2.3.*



Obr. 2.3: Triangulácia striktne konvexného štvoruholníka pred a po aplikácii operácie preklápania, diagonálna e sa vymení za diagonálu e'.

Nech je daná množina všetkých možných triangulácií pre daný vstup ako rovinný graf, kde vrcholmi sú jednotlivé konfigurácie (triangulácie) a hranami sú topologické transformácie. Tento graf sa nazýva grafom všetkých triangulácií nad množinou vrcholov $\boldsymbol{V}$, v skratke *graf triangulácie*. Pre rovinný prípad platí nasledujúce tvrdenie:

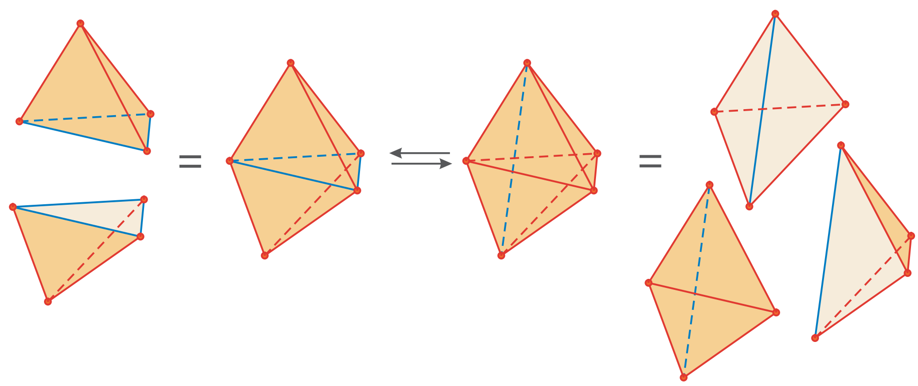
Tvrdenie 2.1.3. *Graf triangulácie pre rovinné prípady s transformáciou preklápania hrán je súvislý.*

Dôkaz. Dôkaz tohto tvrdenia a odhad počtu potrebných preklápaní na transformáciu medzi dvoma trianguláciami sa nachádza v [HOS96].

Poznámka. Napriek svojej jednoduchosti topologická transformácia preklápania hrán je efektívnym a užitočným nástrojom pri hľadaní optimálnych triangulácií. V závislosti od optimalizačného kritéria algoritmy využívajúce túto transformáciu nenájdu vždy optimálne riešenie ale len lokálne optimálny stav. V snahe odstrániť tento hendikep vznikli rôzne rozšírenia operácie preklápania, najznámejším z nich je *vloženie hrany* (*edge insertion*) od Edelsbrunnera a spol. [ETW92]. Základným princípom je vloženie hrany takým spôsobom, aby bol odstránený trojuholník, ktorý má najhoršiu vlastnosť vzhľadom na dané kritérium. Vrcholy hrany sa zvolia tak, aby jedna z nich bola vrcholom trojuholníka, ktorý treba odstrániť. Druhý vrchol sa zvolí tak, aby hrana prechádzala cez daný trojuholník. Následne sa triangulujú oblasti, ktoré vznikli vložením hrany a odstránením hrán, ktoré nová hrana preťala. Algoritmy využívajúce topologickú transformáciu vloženia hrany nájdu globálne optimum vzhľadom na niektoré kritériá, kde prístupy založené na preklápaní hrán nájdu len lokálne optimum (napríklad hľadanie triangulácie, ktorá minimalizuje maximálny uhol [ETW92]). Samozrejme, z hľadiska časovej zložitosti sa jedná o menej efektívny prístup ako v prípade metód založených na preklápaní hrán.

Vo vyšších dimenziách topologické transformácie sú zložitejšie, a existuje viac ich druhov. Je to vďaka rozličným vlastnostiam triangulácií v rovine a vo vyšších dimenziách. Najprv sa uvedie situácia pre 3D, kde sa používajú nasledujúce transformácie:

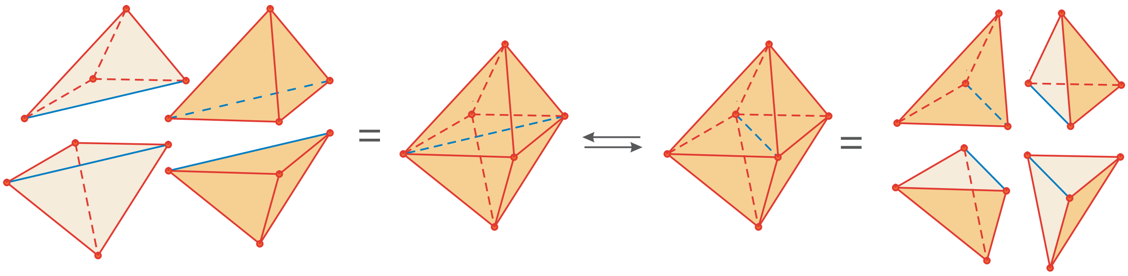
(i) Pre päticu bodov v $\mathbb{R}^3$ formujúce konvexnú bipyramídu - Obrázok 2.4 - spoločná stena (2−stena) môže byť nahradená spoločnou hranou (1−stenou). Táto operácia sa nazýva $2-3$ *preklápanie* ($2-3$ flip). Opačný postup sa nazýva $3-2$ *preklápanie* ($3-2$ flip). Pomocou týchto operácií sa zmení počet tetrahedrónov (3−simplexov), počet hrán a stien triangulácie danej štruktúry. Situáciu znázorňuje Obrázok 2.4.



Obr. 2.4: Ilustrácia topologických transformácií $2-3$ preklápania a $3-2$ preklápania.

(ii) Degenerovaným prípadom je špeciálna konfigurácia šiestich bodov, ktorú tvoria štyri susedné tetrahedrony. Nech štyri zo šiestich bodov sú koplanárne a formujú striktne konvexný štvoruholník, pričom ostatné dva body sú rozdelené rovinou, ktorá je definovaná koplanárnou štvoricou bodov. V takom prípade je možné danú šesticu vrcholov natriangulovať dvomi rôznymi spôsobmi - Obrázok 2.5. Táto transformácia sa nazýva $4-4$ *preklápanie* ($4-4$ flip).

Zovšeobecnenie popísaných preklápacích transformácií vo vyšších dimenziách sa nazýva *bistellárne preklápanie* (*bistellar flip*). Teoretické pozadie týchto transformácií je založené na Radonovej teórii [ES92] z oblasti konvexnej geometrie. Každá transformácia z radu bistellárnych preklápaní môže byť popísaná pomocou projekcie simplexu $\sigma_{n+1} \in \mathbb{E}^{n+1}$ do hyperroviny. Pohľady z antipodálnych bodov projekcie

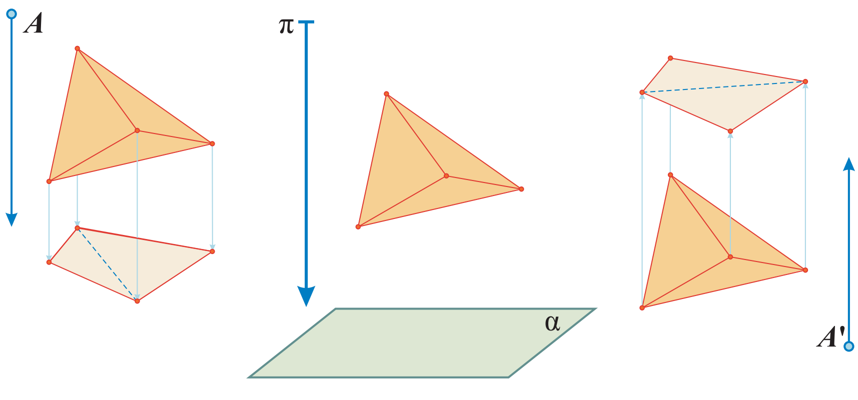


Obr. 2.5: Ilustrácia topologickej transformácie $4-4$ preklápania.

určujú simpliciálne komplexy pred a po bistellárnom preklápaní. Projekcie, ktoré spôsobujú degenerované prípady v projektovanej štruktúre sa dajú popísať ako bistellárne preklápania nižších dimenzií. Na základe tejto teórie sa rozlišujú nasledujúce prípady:

(i) Nech je daný simpliciálny komplex $\mathcal{C}$ v $\mathbb{E}^n$ pozostávajúci z n σ_n simplexov, ktoré majú spoločnú hranu ($1-$stenu). Ak $\mathcal{C}$ má $n+2$ vrcholov ($0-$stien) a je konvexným, tak existujú práve dva spôsoby triangulácie $\mathcal{C}$. Prvá konfigurácia obsahuje spoločnú hranu ($1-$stenu). Druhá konfigurácia nahradí spoločnú hranu s hyperstenou ($(n-1)-$stenou), ktorá je formovaná z vrcholov, ktoré nie sú incidentné s odstránenou hranou.

(ii) Všetky ostatné možnosti sú degenerované prípady a dajú sa popísať pomocou bistellárnych preklápaní nižšej dimenzie.



Obr. 2.6: Projekcia π tetrahedronu do roviny α (v strede) a pohľady z antipodálnych bodov projekcie A, A' (ľavá a pravá časť).

Do skupiny bistellárnych preklápaní patria aj operácie pridávania a odstránenia vrcholov v trianguláciách. Tieto prípady nastanú, ak pri projekcii vrchol padne do vnútra σ_{n-1} simplexu. Princíp bistellárneho preklápania platí aj pre popísané rovinné a 3D topologické transformácie. Napríklad projekcia tetrahedrónu do roviny popisuje topologickú transformáciu v rovine, ilustrácia - Obrázok 2.6.

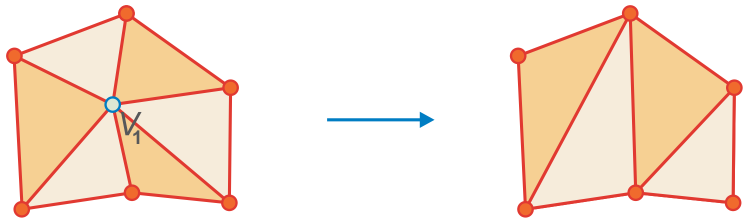
Na rozdiel od rovinných triangulácií, vo vyšších dimenziách nie je dokázané, že graf triangulácií s použitím bistellárnych preklápaní je súvislý. Bolo to dokázané len pre trianguláciu konvexných mnohostenov v 3D, dôkaz sa nachádza v práci [Bes01].

Operácie pre zjednodušovanie triangulácií

Okrem popísaných topologických transformácií sa často stretávať aj operáciami, ktoré majú za úlohu redukovať množstvo vrcholov. Ako motivácia môže slúžiť odstránenie redundancie z dát, kompresia dát, alebo ako predspracovanie modelu pre náročné simulácie.

Definícia 2.1.13. *Nech je daná triangulácia množiny* $\boldsymbol{V} \in \mathbb{E}^2$ *a vrchol* $V_1 \in \boldsymbol{V}$ *ktorý neleží na* $\boldsymbol{CH}(\boldsymbol{V})$. *Topologická operácia, ktorá odstráni z triangulácie vrchol* V_1, *k nemu incidentné hrany a následne trianguluje takto vzniknutý mnohouholník sa nazýva* odstránenie vrchola (vertex removal).

Ak daný vrchol leží na konvexnom obale množiny $\mathbf{V}$, tak po odstránení vrchola sa najprv rekonštruuje konvexný obal a potom sa trianguluje vzniknutý mnohosten. Situáciu vo všeobecnosti znázorňuje Obrázok 2.7.

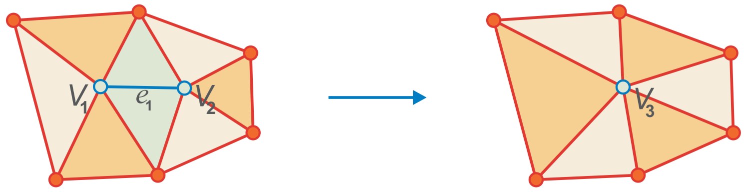


Obr. 2.7: Aplikácia topologickej operácie odstránenie vrchola na vrchol V_1.

Definícia 2.1.14. *Nech je daná triangulácia množiny* $\boldsymbol{V} \in \mathbb{E}^2$ *a hrana* $e_1 \in \boldsymbol{T}(\boldsymbol{V})$, *ktorá nie je incidentná so žiadnym vrcholom ležiace na* $\boldsymbol{CH}(\boldsymbol{V})$. *Topologická operácia, ktorá odstráni z triangulácie hranu* e_1 *takým spôsobom, že zlúči jeho vrcholy do jedného vrchola a odstráni takto vzniknuté degenerované trojuholníky sa nazýva* zlúčenie hrany (edge collapse).

V prípade, že celá hrana leží na $\boldsymbol{CH}(\boldsymbol{V})$, tak sa postupuje ako v prípade uvedenej definície. Zložitejšia je situácia ak len časť hrany leží na konvexnom obale, vtedy v závislosti od pozície zlúčených vrcholov je potrebné rekonštruovať aj konvexný obal. Situáciu vo všeobecnosti znázorňuje Obrázok 2.8.

Poznámka. Existujú aj zložitejšie topologické operácie, ktoré slúžia na zjednodušenie triangulácií, ale pre účely tejto práce uvedené dva prístupy sú postačujúce. Ich rozšírenie do vyšších dimenzií je možné, ale treba si dať pozor aby nevznikli také objekty, ktoré sa potom nedajú triangulovať (viď. Schönhardtov mnohosten, Obrázok 2.2).



Obr. 2.8: Aplikácia topologickej operácie zlúčenie hrany na e_1. Vrchol V_3 vznikol zlúčením vrcholov V_1 a V_2.

Problematika rekonštrukcie a jej vzťah k trianguláciám

Jednou z aplikačných oblastí triangulácií je problematika rekonštrukcie. Jedná sa o veľmi dôležitú úlohu, ktorá sa vyskytuje pri práci s diskrétnymi dátami.

Definícia 2.1.15. *Nech je daná množina* $\boldsymbol{V} = \{V_1, V_2, \cdots, V_n\} \in \mathbb{E}^d$ *a funkcia* $f : \mathbb{E}^d \to \mathbb{E}, d \in \mathbb{N}$. Rekonštrukcia *je spôsob získania funkčnej hodnoty v bode* $V_i \notin \boldsymbol{V}$ *z diskrétnych vzoriek* $f|_{\boldsymbol{V}} = \{f(V_1), f(V_1), \cdots, f(V_n)\}$ *funkcie* f.

Rekonštrukcia teda znamená určenie funkčnej hodnoty v ľubovoľnom bode z dostupných diskrétnych vzoriek. V odbornej terminológii množina $\boldsymbol{V}$ a na nej definované vzorky funkcie f sa nazývajú množinou *rozptýlených dát* (*scattered data*).

Rekonštrukcia rozptýlených dát pomocou triangulácií znamená použitie triangulácie množiny $\boldsymbol{V}$ ako rekonštrukčnú sieť. Nové hodnoty sa získajú ako barycentrické kombinácie dát z vrcholov triangulácie. Z vlastností triangulácií je jasné, že triangulácia ako rozklad oboru hodnôt funkcie f, určuje jednoznačne nové hodnoty. Každý bod z $\boldsymbol{CH(V)}$ leží práve v jednom simplexe σ_n, alebo na hranici viacerých simplexov σ_n, teda funkčná hodnota je určená jednoznačne. Okrem barycentrickej kombinácie dát z vrcholov je možné používať aj iné spôsoby rekonštrukcie, napríklad trojuholníkové splajny atď.

Vzhľadom na to, že rekonštrukcia je často len medzikrokom v postupe rôznych operácií a simulácií, je dôležité, aby bola vykonaná čo najkvalitnejšie. Meranie kvality a určenie adekvátnosti vybranej techniky je závislé od aplikačnej oblasti. Zvyčajne sa však vyžaduje čo najpresnejšia rekonštrukcia významných čŕt, ktoré nesú dôležitú informáciu o správaní funkcie f. Takými oblasťami sú miesta, kde f má v jednom smere väčšiu druhú deriváciu v porovnaní s inými smermi. V terminológii spracovania obrazu to znamená vysokofrekvenčné oblasti, t.j. rekonštrukciu hrán; pri rekonštrukcii 3D medicínskych dát (získané napríklad pomocou počítačovej tomografie) získanie vhodne rekonštruovaných izoplôch atď.

V závislosti od distribúcie množiny $\boldsymbol{V}$ sa dá organizácia dát deliť do skupín: dáta organizované do karteziánskej mriežky (*cartesian grid*), alebo v inom prípade neštruktúrované dáta (*unstructured grid*)[2]. Bez nároku na úplnosť možno uviesť najčastejšie používané rekonštrukčné techniky.

[2]Existuje dôkladnejšie rozdelenie, napríklad v literatúre [vBSF05], ale pre účely tejto práce stačí uvedený spôsob.

Pre dáta organizované do karteziánskej mriežky najčastejším spôsobom rekonštrukcie je použitie takzvaných *konvolučných techník*. Tenzorový súčin 1D rekonštrukčných filtrov alebo iný spôsob výberu jadra však kvalitnú rekonštrukciu nezabezpečuje. Rekonštrukčné chyby nastanú v rekonštrukcii čŕt, ktoré majú inú orientáciu ako súradnicové osi karteziánskej mriežky. Výhodou konvolučných techník je možnosť využitia grafických akcelerátorov [Vio02, Bjo04] a nízka časová náročnosť. Prehľad konvolučných techník sa nachádza v knihách [Gla95, GW06].

Na rekonštrukciu neštruktúrovaných dát sa používajú rôzne numerické metódy. Výsledok týchto prístupov je často vo forme analytických vyjadrení, čo zhoršuje prácu so spojitou reprezentáciou a zvyšuje náročnosť vyčíslenia rekonštruovaných hodnôt. Prehľad numerických rekonštrukčných metód možno nájsť v prácach [Nie93, IA04, Wen05].

Ako alternatíva k uvedeným metódam je geometricky založený prístup, ktorý využíva trianguláciu. Môže slúžiť na rekonštrukciu dát, ktoré sú organizované do karteziánskej mriežky ako aj na rekonštrukciu neštrukturovaných dát. Pomocou topologických zmien je táto technika schopná prispôsobiť trojuholníkovú sieť k črtám reprezentovaných v dátach a takto zvýšiť kvalitu rekonštrukcie. Obrázok 2.9 zobrazuje rekonštrukčný výsledok konvolučnej techniky a geometricky založenej techniky pre aplikačnú oblasť rekonštrukcie obrazu [TVFG07].

Obr. 2.9: 600% magnifikácia pomocou konvolučnej techniky (vľavo) a geometricky založenej techniky (vpravo).

2.1.2 Kritéria optimality

Z doterajšie uvedených tvrdení vyplýva, že pre daný vstup všetky rovinné triangulácie majú rovnaký počet hrán, trojuholníkov. Teda kvalita triangulácie musí byť závislá od tvaru jej trojuholníkov. Najčastejšie sa uvažujú nasledujúce vlastnosti:

— vnútorné uhly,
— výška trojuholníka,
— obsah trojuholníka,
— dĺžka hrán,
— orientácia trojuholníkov (napríklad pre tzv. *fluid flow* simulácie),
— pomer medzi veľkosťou vpísanej a opísanej kružnice pre daný trojuholník.

Na základe týchto faktorov sa hľadá triangulácia s maximálnou, minimálnou hodnotou alebo sumou týchto hodnôt. Zvyčajne sa optimalizuje vzhľadom na 1-2 kritérií naraz, ale napríklad tzv. Delaunayova triangulácia optimalizuje viacero z uvedených kritérií.

Tvar simplexov v trianguláciách hrá dôležitú úlohu aj vo vyšších dimenziách. Veľa simulačných metód vyžaduje vstupnú trianguláciu, spĺňajúcu určité vlastnosti, aby nedošlo k numerickým nepresnostiam. V trojrozmernom priestore sa najčastejšie používajú nasledujúce vlastnosti:

— uhly (priestorové (*solid*), medzi stenami (*dihedral*)),
— pomer medzi najkratšou hranou a veľkosťou opísanej gule pre daný tetrahedrón (3-simplex),
— pomer medzi veľkosťou vpísanej a opísanej gule pre daný tetrahedrón.

Zvyčajne sa hľadá triangulácia s minimálnou alebo maximálnou hodnotou na základe uvedených kritérií. Optimálna triangulácia vzhľadom na dané kritérium sa definuje vo všeobecnosti nasledovne:

Definícia 2.1.16. *Nech $\mathcal{T}$ je množinou všetkých možných triangulácií množiny $\boldsymbol{V}$, a je dané zobrazenie $\Psi : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$. Potom:*

(i) triangulácia $\boldsymbol{T}_1(\boldsymbol{V})$ je kvalitnejšia než triangulácia $\boldsymbol{T}_2(\boldsymbol{V})$ vzhľadom na Ψ, ak $\Psi(\boldsymbol{T}_1(\boldsymbol{V}), \boldsymbol{T}_2(\boldsymbol{V})) < 0$,

(ii) triangulácie sú rovnako kvalitné, ak $\Psi(\boldsymbol{T}_1(\boldsymbol{V}), \boldsymbol{T}_2(\boldsymbol{V})) = 0$,

(iii) triangulácia $\boldsymbol{T}_1(\boldsymbol{V})$ sa nazýva optimálna *alebo* globálne optimálna *vzhľadom na Ψ, ak pre $\forall \boldsymbol{T}_2(\boldsymbol{V}) \in (T)$ platí $\Psi(\boldsymbol{T}_1(\boldsymbol{V}), \boldsymbol{T}_2(\boldsymbol{V})) \leq 0$.*

Z definície vyplýva, že optimálna triangulácia vzhľadom na dané optimalizačné kritérium vždy existuje, ale nemusí byť jedinečná. To jest, môžu existovať rôzne, ale rovnako kvalitné riešenia. Nájdenie optimálnej triangulácie môže vyžadovať vyšetrenie exponenciálneho množstva triangulácií, preto sa často hľadá len jeho aproximácia. Pre tento účel sa zavádza pojem lokálnej optimality, kde sa hľadá splnenie podmienok len vzhľadom na lokálne okolie. Presná definícia lokálnej optimality bude zavedená neskôr. Prehľad optimálnych triangulácií možno nájsť v prehľadových prácach [BE95, Ede00, HD06].

Lokálne optimum mnohokrát dostatočne aproximuje globálne optimálny stav, a pritom jeho konštrukcia je podstatne jednoduchšia, rýchlejšia. V niektorých prípadoch dokonca lokálne optimálny stav indikuje nájdenie globálneho optima.

Získanie väčšej kontroly nad kvalitou triangulácie bolo cieľom rozšírenia optimálnych triangulácií o metódu *dátovo závislých triangulácií* (*data dependent triangulation*, ozn. *DDT*). Táto technika od autorov Dyn a spol. [DLR90] zmenila koncepciu chápania optimalizačných kritérií a umožnila dosiahnuť pozoruhodné výsledky. Nasledujúca časť sa zaoberá so skúmaním vlastností tejto triangulácie.

2.1.3 Dátovo závislé triangulácie

Dátovo závislé triangulácie sú úzko spojené s trianguláciami terénu. Pokiaľ pri rovinných trianguláciách sa berie do úvahy len distribúcia vrcholov, v prípade *DDT* sa využíva aj informácia z priradených funkčných hodnôt. Z matematického hľadiska sa to dá charakterizovať nasledovne. Nech $\boldsymbol{V} = \{x_i, y_i, z_i\}_{i=1}^{m}$ je množina vrcholov, ktorá tvorí terén:

$$f(x_i, y_i) = z_i, \ i = 1, \ldots, m \ , \tag{2.7}$$

to jest, $f(x, y)$ je bijekciou. Úlohou dátovo závislých triangulácií je čo najlepšie aproximovať funkciu $f(x, y)$.

V závislosti od aplikačnej oblasti sa môže jednať o zostrojenie terénnej triangulácie, alebo o konštrukciu rovinnej triangulácie, kde sa využíva dodatočná informácia, zapísaná do funkčných hodnôt. Z hľadiska topológie sú tieto dve triangulácie identické.

Zvyčajne sa považovali dlhé a úzke trojuholníky za nežiaduce elementy v optimálnych trianguláciách. Až dátovo závislé triangulácie zmenili tento pohľad, a poukázali na možnosť využitia týchto prvkov, kde rekonštrukcia funkcie $f(x, y)$ to vyžaduje. Hlavne sa jedná o oblasti, kde druhá derivácia funkcie $f(x, y)$ je v jednom smere výraznejšia v porovnaní s ostatnými smermi. Inými slovami, prudká zmena funkčných hodnôt v jednom smere[3].

Úspech dátovo závislých triangulácií spočíva v priradení tzv. *cenových funkcií* k simplexom v trianguláciách na základe geometrických alebo iných vlastností, teda použitím špeciálnych optimalizačných kritérií. Minimalizáciou sumy všetkých priradených cenových funkcií sa dosiahne triangulácia so žiadanými vlastnosťami. Pri voľbe vhodných cenových funkcií sa to prejavuje výskytom dlhých, úzkych trojuholníkov v oblastiach, kde to vyžaduje aproximácia $f(x, y)$. Pre tento účel existujúce triangulácie nedávali také uspokojivé riešenia, aké poskytuje *DDT*.

V existujúcich dátovo závislých prístupoch sa používa špeciálne ohodnotenie hrán alebo vrcholov. Nižšie uvedené definície sú založené na použití hranovo založených cenových funkciách, vrcholovo založený prístup bude predstavený neskôr.

Definícia 2.1.17. *Váhou* (cenou) *hrany e sa bude označovať číslo $c(e) \in \mathbb{R}$.*

[3]Napríklad takými miestami sú vysokofrekvenčné oblasti (hrany) v prípade digitálnych obrazov.

Definícia 2.1.18. Váha (cena) triangulácie $\boldsymbol{T(V)}$ *je súčtom dĺžok všetkých jej hrán:*

$$c(\boldsymbol{T(V)}) = \sum_{e \in \boldsymbol{T(V)}} \|c(e)\| \ , \tag{2.8}$$

pričom sa používa l_1 norma, teda súčet absolútnych hodnôt.

Poznámka. Existujú prístupy, kde za váhu triangulácie sa berú súčty v iných metrikách [DLR90], najčastejšie sa však ale používa l_1 norma.

Definícia 2.1.19. Triangulácia s minimálnou váhou *(minimum weight triangulation, ozn. MWT) množiny vrcholov* $\boldsymbol{V}$ *je triangulácia, ktorej váha je spomedzi všetkých triangulácií minimálna:*

$$c(MWT(\boldsymbol{V})) = \sum_{e \in MWT(\boldsymbol{V})} \|c(e)\| = \min\{c(\boldsymbol{T(V)})\}, \ \ \forall \boldsymbol{T(V)}. \tag{2.9}$$

Niektoré hrany vždy patria do MWT: najlacnejšia hrana a tie hrany, ktoré nepretína žiadna iná hrana (napr. hrany na konvexnom obale).

Poznámka. Problematika MWT s cenovou funkciou dĺžky hrany patrí medzi známe problémy vo výpočtovej geometrie a má staršie korenie ako dátovo závislé triangulácie. NP-ťažkosť tejto úlohy bola dokázaná len nedávno [Mul06]. Za prekvapujúce možno považovať zistenie, že pridanie Steinerových vrcholov môže priniesť trianguláciu s nižšou váhou [BE95]. Teda pridaním vrchola a k nemu incidentných hrán sa môže znížiť celková dĺžka všetkých hrán! Ilustrácia takejto množiny vrcholov je zobrazená na Obrázku 2.10.

Obr. 2.10: MWT triangulácia pôvodnej množiny na ľavej strane. Na pravej strane, pridaný vrchol V_1, cena MWT tejto množiny je nižšia.

Vzhľadom na to, že MWT predstavuje globálne optimálny stav, je potrebné si zadefinovať aj pojem lokálnej optimality pre túto problematiku.

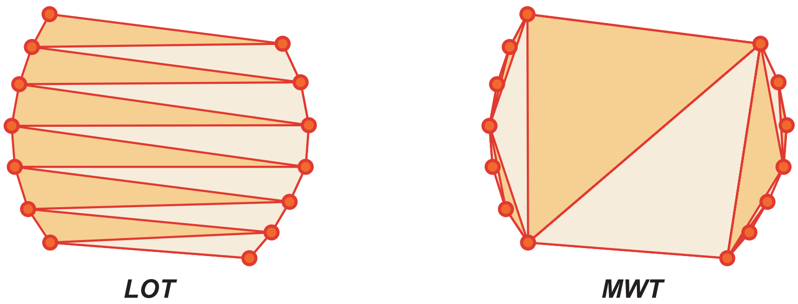
Definícia 2.1.20. *Hrana $e \in \boldsymbol{T(V)}$ sa nazýva* lokálne optimálna, *ak pre štvoruholník vytvorený z trojuholníkov zdieľajúcich hranu e platí jedno z nasledujúcich tvrdení:*

(i) nie je striktne konvexný

(ii) je striktne konvexný a je optimálne triangulovaný (vzhľadom na MWT).

Definícia 2.1.21. Lokálne optimálna triangulácia *ozn. LOT množiny* $\mathbf{V}$ *je trianguláciа, v ktorej každá hrana, ktorá nepatrí do hranice konvexného obalu, je lokálne optimálna.*

Poznámka. Lokálne optimum môže byť značne vzdialené od globálneho optima. Napríklad v prípade problematiky MWT (s cenovou funkciou dĺžka hrany) cena LOT môže byť až $\Omega(n)$ násobok skutočnej ceny MWT, kde n je početnosť množiny vrcholov triangulácie. Ilustrácia takéhoto prípadu je zobrazená na Obrázku 2.11. Ľubovoľná triangulácia dosahuje $\mathcal{O}(n)$ násobok dĺžky MWT. Dôkazy týchto tvrdení sa nachádzajú v [Kir80].



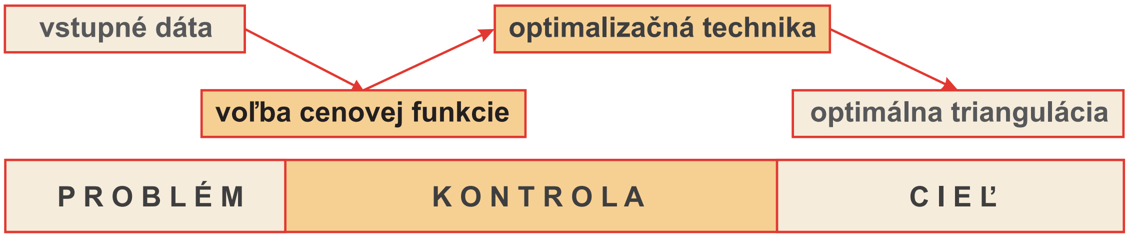
Obr. 2.11: Lokálne optimálna (LOT) a globálne optimálna triangulácia (MWT) množiny.

Definícia 2.1.22. Neutrálna výmena diagonál *v triangulácii je taký proces, pri ktorom sa hrana e vymení za druhú diagonálu v striktne konvexnom štvoruholníku, ktorá e obsahuje ako svoju diagonálu, pričom váha triangulácie sa nemení.*

Definícia 2.1.23. *Nech hrana e je diagonálou striktne konvexného štvoruholníka, vytvoreného s trojuholníkmi obsahujúcimi e ako spoločnú hranu. Druhá diagonála sa bude nazývať* alternatívna diagonála *hrany e.*

Problém nájdenia MWT pre dátovo závislé triangulácie je zovšeobecnením problematiky MWT pre rovinné triangulácie. V rovinnom prípade namiesto cenových funkcií sa minimalizuje dĺžka hrán v danej metrike. Existujú rôzne poznatky o podgrafoch, aproximáciach a heuristikách MWT v rovine [Fer04]. V prípade DDT kvôli špeciálnemu oceneniu jednotlivých hrán (alebo iných komponentov) len časť týchto výsledkov je použiteľná.

Dátovo závislé triangulácie tvoria významnú časť skupiny optimálnych triangulácií. Väčšinu optimálnych triangulácií je možné konštruovať pomocou DDT v závislosti od voľnosti chápania priradenia cien k jednotlivým komponentom triangulácie a od zvolenej optimalizačnej techniky. Ide o veľmi efektívny nástroj pre vytváranie optimálnych triangulácií. Ideálnym prípadom je, ak používateľ má dostatočné informácie o aplikačnej oblasti, kde chce nasadiť DDT. Potom má možnosť zaviesť špeciálne cenové funkcie, a tak získať väčšiu kontrolu nad kvalitou vytváranej siete, ako v prípade iných rovinných triangulácií. Dizajn optimálnej siete pomocou



Obr. 2.12: Priebeh rekonštrukcie pomocou metódy dátovo závislých triangulácií.

DDT naznačuje graf zobrazený na Obrázku 2.12. Nasledujúca časť sa zaoberá prvým prvkom kontroly nad kvalitou, s cenovými funkciami pre hranovo založený prístup.

Cenové funkcie

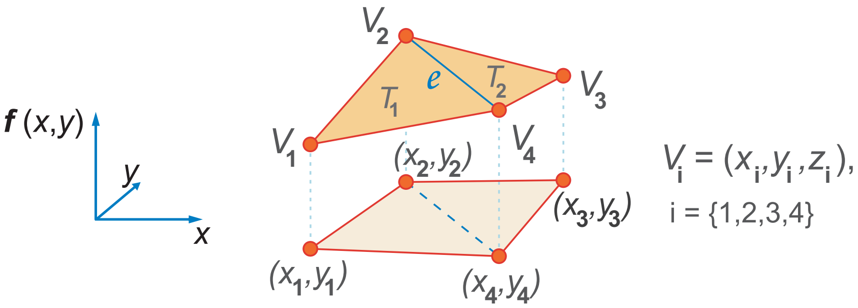
Dôležitosť cenových funkcií nemôže nič lepšie demonštrovať ako ukážka vhodnej a nevhodnej voľby tejto funkcie. Pre účely demonštrácie bola vybraná aplikačná oblasť rekonštrukcie obrazu. Za vrcholy triangulácie sú zvolené umiestnenia pixlov a ich intenzita za funkčné hodnoty. Po výpočte aproximácie MWT s jednotlivými cenovými funkciami výsledné triangulácie boli použité ako rekonštrukčná sieť pri zväčšení obrázkov, výsledok je zobrazený na Obrázku 2.13.

Poznámka. Na určenie funkčnej hodnoty pri farebných obrázkoch sa môže použiť intenzita alebo prístup uvedený v našej predchádzajúcej práci [Tót06], ktorý využíva transformáciu do perceptuálne lineárneho farebného modelu.

Obr. 2.13: Výsledok po použití vhodnej a nevhodnej cenovej funkcie.

Pri definícii cenových funkcií sa najčastejšie používa nasledujúca závislosť: každá hrana DDT triangulácie množiny $\boldsymbol{V}$, ktorá neleží na $\boldsymbol{CH(V)}$, je ohodnotená pomocou informácií získaných práve zo štyroch vrcholov, ktoré sú vrcholmi trojuholníkov obsahujúce danú hranu. Situáciu znázorňuje Obrázok 2.14, kde T_1 a T_2 sú trojuholníky v triangulácii danej množiny, ktoré obsahujú hranu e ako svoju spoločnú hranu. Predpisy rovín obsahujúce T_1 a T_2 sa budú označovať ako $P_1(x, y), P_2(x, y)$ (v krátkosti P_1, P_2):

$$\begin{aligned} P_1(x, y) &= a_1 x + b_1 y + c_1 = z_1, \\ P_2(x, y) &= a_2 x + b_2 y + c_2 = z_2. \end{aligned} \tag{2.10}$$



Obr. 2.14: Ilustrácia pre výpočet cenovej funkcie.

Definície cenových funkcií, ktoré sa nachádzajú v [DLR90, YBS01, BGM04] vyzerajú nasledovne:

ABN (*angle beetween normals*) meria uhol medzi normálami trojuholníkov T_1 a T_2. Nech $\vec{n}_1$ a $\vec{n}_2$ sú označenia týchto normál, pri ich popise je možné vychádzať z gradientov implicitných vyjadrení rovín:

$$\begin{aligned} p_i(x,y,z) &= -a_i x - b_i y - c_i + z_i = 0,\ i = \{1,2\}, \\ \vec{n}_i = \nabla p_i(x,y,z) &= \left(\frac{\partial p_i}{\partial x}, \frac{\partial p_i}{\partial y}, \frac{\partial p_i}{\partial z}\right) = (-a_i, -b_i, 1),\ i = \{1,2\}, \end{aligned} \tag{2.11}$$

potom vyjadrenie uhla medzi normálami môže byť popísané nasledovne:

$$c^{ABN}(e) = cos^{-1}\frac{\vec{n}_1 \cdot \vec{n}_2}{|\vec{n}_1||\vec{n}_2|} = cos^{-1}\frac{a_1 a_2 + b_1 b_2 + 1}{\sqrt{(a_1^2 + b_1^2 + 1)(a_2^2 + b_2^2 + 1)}}\,. \tag{2.12}$$

Snahou c^{ABN} je vytvorenie čo najhladšej po častiach lineárnej plochy.

JND (*jump in normal derivatives*) je ďalšou možnosťou ako optimalizovať hladkosť vytváranej plochy. Cenová funkcia popisuje zmenu v normálach derivácií P_1 a P_2 cez hranu e. Nech $\vec{n} = (n_x, n_y)$ je jednotkový vektor v $\mathbb{E}^2$, ktorý je ortogonálny na smer hrany e, potom platí:

$$\begin{aligned} \frac{\partial P_i}{\partial \vec{n}} = \nabla P_i \cdot \vec{n} &= \left(\frac{\partial P_i}{\partial x}, \frac{\partial P_i}{\partial y}\right) \cdot (n_x, n_y) = (a_i n_x + b_i n_y),\ i = \{1,2\}, \\ c^{JND}(e) &= \left|\frac{\partial P_1}{\partial \vec{n}} - \frac{\partial P_2}{\partial \vec{n}}\right| = |n_x(a_1 - a_2) + n_y(b_1 - b_2)|\,. \end{aligned} \tag{2.13}$$

DLP (*deviations from linear polynomials*) meria chybu medzi lineárnymi polynómami P_1 a P_2 vo vrcholoch V_3 a V_1:

$$\begin{aligned} c^{DLP}(e) &= \|h\|, \\ h &= \begin{bmatrix} |P_1(x_3,y_3) - f(x_3,y_3)| \\ |P_2(x_1,y_1) - f(x_1,y_1)| \end{bmatrix}. \end{aligned} \tag{2.14}$$

DP (*distances from planes*) je veľmi podobná k cenovej funkcii *DLP*, ale tu namiesto vertikálnej vzdialenosti sa používa vzdialenosť medzi vrcholmi V_1 a V_3 od

P_2 a P_1:

$$\begin{gathered} c^{DP}(e) = \|g\| \ , \quad g = \begin{bmatrix} dist(P_1, v_3) \\ dist(P_2, v_1) \end{bmatrix} , \\ dist(P_i, v_j) = \frac{|P_i(x_j, y_j) - f(x_j, y_j)|}{\sqrt{(a_i^2 + b_i^2 + 1)}}, \ i = \{1, 2\}, \ j = \{1, 3\} \ . \end{gathered} \tag{2.15}$$

Poznámka. Uvedené cenové funkcie vznikli pre všeobecné použitie, na rekonštrukciu funkčných hodnôt ľubovoľnej funkcie. Pre niektoré aplikačné oblasti tieto konštrukcie môžu mať nedostatky. Napríklad pri rekonštrukcii obrazu funkčné hodnoty môžu byť zadané ako hodnoty od $0 - 255$ alebo v rozmedzí $0 - 1$, pritom jedná sa o tú istú informáciu. Takéto škálovanie by nemalo mať vplyv na výpočty, ale v prípade cenovej funkcie *DP* sa predsa len výsledné triangulácie líšia. V snahe odstránenia takýchto nedostatkov špeciálne pre problematiku rekonštrukcie obrazu vznikla nasledujúca cenová funkcia.

SCF (*Sederbergs cost function*) je založená na meraní uhla medzi normálami izolínii[4], váhované veľkosťou týchto normál. Nech π je označenie roviny, ktorá je rovnobežná s rovinou xy, a určuje prierez trianguláciou - teda vytvára izolínie. Vzhľadom na to že triangulácia množiny tvorí po častiach lineárnu plochu aj vytvorené izolínie budú po častiach lineárne. Normály týchto izočiar sú vlastne priemetmi normál prislúchajúcich trojuholníkov ($\vec{n}_1$ a $\vec{n}_2$) do roviny π. Situácia je znázornená na Obrázku 2.15. Označenie týchto premietnutých normál bude $\vec{n}_1^\pi$ a $\vec{n}_2^\pi$, ich veľkosť je:

$$\vec{n}_i^\pi = \{a_i, b_i, 0\}, \ i = \{1, 2\}. \tag{2.16}$$

Potom je cenová funkcia definovaná nasledovne:

$$\begin{gathered} c^{SCF}(e) = |\vec{n}_1^\pi||\vec{n}_2^\pi|(1 - cos\alpha) = |\vec{n}_1^\pi||\vec{n}_2^\pi| \left(1 - \frac{\vec{n}_1^\pi \cdot \vec{n}_2^\pi}{|\vec{n}_1^\pi||\vec{n}_2^\pi|}\right) = \\ = |\vec{n}_1^\pi||\vec{n}_2^\pi| - \vec{n}_1^\pi \cdot \vec{n}_2^\pi = \sqrt{(a_1^2 + b_1^2)}\sqrt{(a_2^2 + b_2^2)} - (a_1 \cdot a_2 + b_1 \cdot b_2), \end{gathered} \tag{2.17}$$

kde α značí uhol medzi uhol medzi normálami $\vec{n}_1^\pi$ a $\vec{n}_2^\pi$.

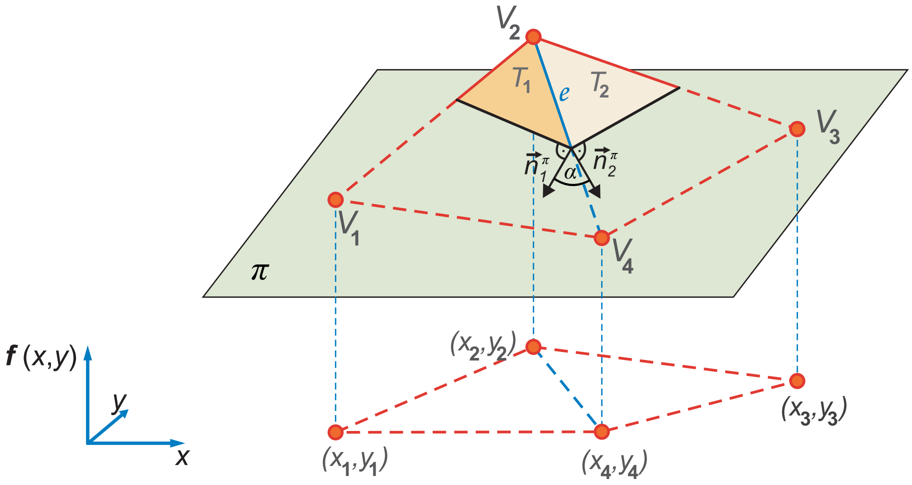
Poznámka. *SCF* má aj ďalšie výhody okrem toho, že je invariantná vzhľadom na škálovanie funkčných hodnôt. Snaží sa zabrániť vzniku dlhých a úzkych trojuholníkov v oblastiach, kde to nie je potrebné, ako napríklad nízkofrekvenčné oblasti v obrazoch.

Na aproximáciou cenovej funkcie *SCF* môže byť použitý nasledujúci predpis (ozn. ***SCF-Apr***) od autorov článku [BGM04]:

$$c^{SCF-Apr} = (|a_1| + |b_1|) \cdot (|a_2| + |b_2|) - (a_1 \cdot a_2 + b_1 \cdot b_2), \tag{2.18}$$

vyčíslenie tejto cenovej funkcie je výpočtovo lacnejšie.

[4] *Izolínia* (*contour line*) je taká čiara, pozdĺž ktorej funkčné hodnoty majú rovnakú hodnotu.



Obr. 2.15: Ilustrácia pre výpočet *SCF*.

Pre účely rekonštrukcie obrazu podľa skúsenosti autorov článku [YBS01] je najúspešnejšia *SCF*. Podľa testov [DLR90] na všeobecné použitie (rekonštrukcia funkčných hodnôt) najlepšie výsledky dávali *ABN* a *JND*. Okrem uvedených cenových funkcií existujú aj ďalšie prístupy, ktoré priraďujú cenu rôznym simplexom (vrcholom, hranám, trojuholníkom) na základe geometrických alebo iných vlastností [DLR90, Bro91, AKTD99].

2.2 Optimalizačné techniky pre dátovo závislé triangulácie

Okrem vhodnej voľby cenovej funkcie kvalitu výslednej triangulácie ovplyní aj výber optimalizačného procesu. Optimalizačné prístupy sa dajú deliť na deterministické a stochastické na základe toho, či využívajú nejaké pravdepodobnostné postupy alebo nie. Prvá časť bude venovaná deterministickým metódam.

2.2.1 Deterministické prístupy

Deterministické metódy značia skupinu algoritmov, kde každý krok je jednoznačne určený a náhodné kroky sú neprípustné. Väčšina existujúcich deterministických dátovo závislých prístupov využíva konštrukciu pomocou iteračných procesov. Pri iteračných metódach dôležitú úlohu hrá voľba úvodnej triangulácie (začiatočný stav). Zvyčajne sa pre tento účel používa *Delaunayova triangulácia* (ozn. *DT*) [Aur91], ktorej vlastnosti budú popísané v nasledujúcej časti spolu s najznámejšou konštrukčnou technikou.

Poznámka. Výhodou deterministických metód je aproximácia *MWT*. Dá sa určiť o koľko je výsledok danej metódy horšia ako kvalita (cena) *MWT*.

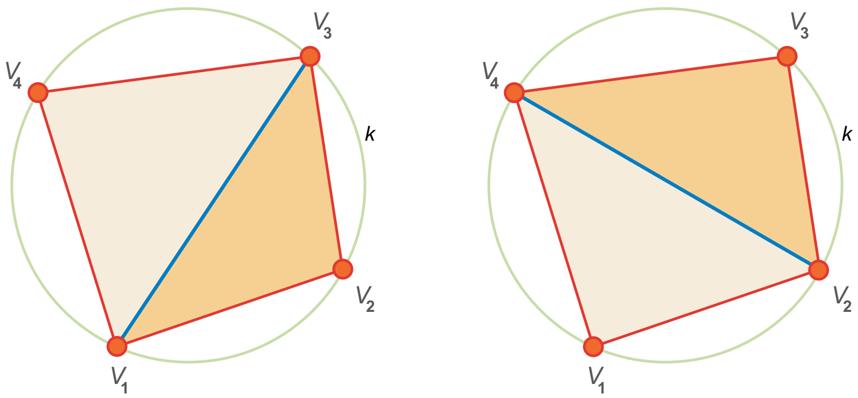
Lawsonova optimalizačná procedúra

Najčastejšie používanou metódou pre vytváranie DDT je *Lawsonova optimalizačná procedúra* (ozn. LOP), ktorá slúži na získanie lokálne optimálnej triangulácie. Jej konštrukcia a výsledná štruktúra je úzko spájaná Delaunayovou trianguláciou, patriace medzi historicky prvé a významné triangulácie aproximujúce MWT. Prvá časť tejto sekcie bude venovaná vlastnostiam a konštrukčným technikám DT, ktoré sa potom využijú aj pri zostrojení LOP.

Delaunayova triangulácia sa často používa v praxi na rôzne účely a to nielen v počítačovej grafike ale aj v iných vedných odboroch. Najčastejšie sa definuje pomocou pojmu *Voronoiov diagram*, ktorý tvorí duálnu štruktúru k DT. Princíp duality a ďalšie vlastnosti Voronoiových diagramov sú podrobne popísané v Aurenhammerovej prehľadovej práci [Aur91].

Definícia 2.2.1. *[Rip90] Nech je daná množina vrcholov* $\boldsymbol{V} = \{V_i \in \mathbb{E}^2; i = 1, \ldots, m\}$*, ktorej prvky sa nazývajú* generátory*.* Voronoiova oblasť *vrcholu* V_i *je množina bodov, ktorá je bližšia (v danej metrike) ku* V_i *ako ku iným vrcholom z* $\boldsymbol{V}$*. Zostrojenie Voronoiovej oblasti pre každý vrchol z množiny* $\boldsymbol{V}$ *rozdelí celý priestor na disjunktné oblasti, ktoré majú spoločné len hranice. Dve Voronoiove oblasti sa nazývajú susedné, ak majú spoločnú hranicu vo výslednej štruktúre. Spojením generátorov susedných Voronoiových oblastí sa získa triangulácia množiny* $\boldsymbol{V}$*, ktorá sa nazýva* Delaunayova triangulácia.

Poznámka. V prípade kocirkularity štyroch (alebo viac) vrcholov z množiny $\boldsymbol{V}$, Delaunayova triangulácia nie je jednoznačne určená, ako to zobrazuje príklad na Obrázku 2.16. V takomto prípade pre danú množinu generátorov (ktoré sú kocirkulárne) sa zvolí ľubovoľná triangulácia. Bolo dokázané že konečným počtom neutrálnych preklápaní sa dá dostať z jednej DT do druhej DT [HOS96].



Obr. 2.16: Nejednoznačnosť DT triangulácie množiny kocirkulárnych generátorov $\{V_1, V_2, V_3, V_4\}$. Obidve triangulácie spĺňajú definíciu DT.

Myšlienka Voronoiových diagramov bola generalizovaná do vyšších dimenzií $\mathbb{E}^n$, $n \in \mathbb{N}$, a teda k nej existujú aj príslušné duálne Delaunayove štruktúry (napríklad v $\mathbb{E}^3$ je to Delaunayova tetrahedralizácia). Dôvodom obľúbenosti Delaunayovej triangulácie v rôznych aplikačných oblastiach sú jej vlastnosti a nízka časová zložitosť konštrukcie. V nasledujúcej časti sú uvedené dôležitejšie vlastnosti rovinných DT, pričom existujú aj ich zovšeobecnenia do vyšších dimenzií [BE95].

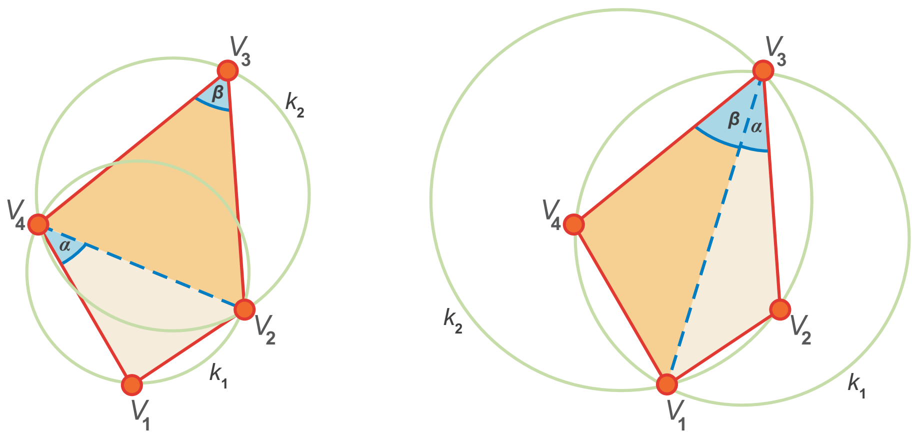
Veta 2.2.1. *(*Kritérium prázdneho kruhu*) Triangulácia* $\boldsymbol{T(V)}$ *množiny bodov* $\boldsymbol{V}$ *v rovine je DT práve vtedy, ak opísaná kružnica ľubovoľného jej trojuholníka neobsahuje vo svojom vnútri bod z* $\boldsymbol{V}$*.*

Dôkaz. Dôkaz tohto tvrdenia sa nachádza v knihe [PS85].

Veta 2.2.2. *(*Maximalizácia minimálneho uhla*) Delaunayova triangulácia maximalizuje minimálny uhol v každom trojuholníku ako aj v celej triangulácii.*

Dôkaz. Dôkaz tohto tvrdenia sa nachádza v knihe [PS85].

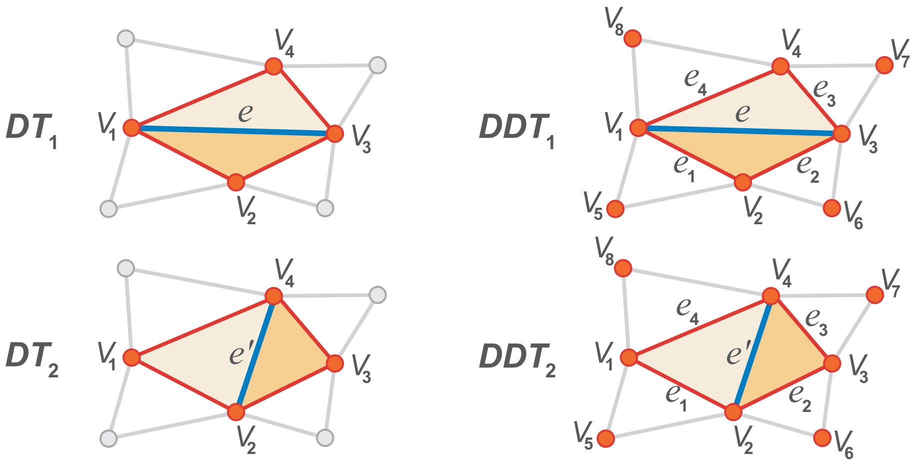
Inými slovami, Veta 2.2.2 znamená, že Delaunayova triangulácia generuje najviac rovnakouhlú (*equiangular*) trianguláciu. Táto vlastnosť je veľmi dôležitá napríklad pre numerickú stabilitu *FEM* výpočtov. Trend preferovania rovnakouhlosti triangulácií je zbadateľný aj v iných aplikačných oblastiach. Až objavenie DDT otváralo diskusiu o využiteľnosti triangulácií s dlhými a úzkymi trojuholníkmi, ktoré vylučujú Vety 2.2.1 a 2.2.2. Kritériá prázdneho kruhu a maximalizácie minimálneho uhla sú ekvivalentné. Na Obrázku 2.17 je uvedený príklad, kde triangulácia vrcholov spĺňa obidve podmienky v prípade vhodnej triangulácie (vľavo), alebo ani jedna z podmienok nie je splnená (vpravo).



Obr. 2.17: Na ľavej strane DT triangulácia, ktorá maximalizuje minimálne uhly α a β, a spĺňa kritérium prázdneho kruhu. Triangulácia zobrazená na pravej strane nespĺňa ani jednu z týchto podmienok.

V prípade DT lokálna optimalita (na základe Vety 2.2.1 a 2.2.2) je iná ako lokálna optimalita v zmysle DDT. Tento rozdiel spočíva v počte potrebných informácií na určenie lokálnej optimality. Nech je daná triangulácia $\boldsymbol{T}(\boldsymbol{V})$ a v nej striktne konvexný štvoruholník $V_1V_2V_3V_4$ s diagonálou e. Situácia je znázornená na Obrázku 2.18.

Výmena tejto diagonály za alternatívnu diagonálu e' v prípade DDT ovplyvní aj cenu okolitých hrán (e_1, e_2, e_3, e_4). To znamená, že optimálna triangulácia $V_1V_2V_3V_4$ zahŕňa v sebe aj vyčíslenie cien hrán, ktoré vytvárajú štvoruholník a vyberie sa tá konfigurácia, ktorej celková cena je menšia. Teda pre určenie lokálnej optimality sa využije informácia z ôsmich vrcholov, v prípade DT je potrebná informácia zo štyroch vrcholov. Najznámejšie spôsoby konštrukcie DT sú nasledujúce.



Obr. 2.18: Závislosť lokálnej optimality hrany od okolia v prípade DT (vľavo) a DDT (vpravo). Označené vrcholy sú použité pri vyhodnotení lokálnej optimality. Cena hrán e_1, e_2, e_3, e_4 závisí od výberu diagonály v štvoruholníku $V_1V_2V_3V_4$.

Asymptoticky najlepšie správanie má Dwyerov algoritmus [Dwy86] s optimálnou časovou zložitosťou $T : \mathcal{O}(n \log n)$. Jedná sa o metódu rozdeľuj a panuj (*divide&conquer*). Jej nevýhodou je numerická nestabilita v niektorých prípadoch a náročnejšia implementácia ako v prípade ostatných metód. Vo všeobecnosti očakávaná časová zložitosť tohto algoritmu je $T : \mathcal{O}(n \log \log n)$.

Ďalším spôsobom konštrukcie DT (ale aj iných LOT) je algoritmus využívajúci postupné pridávanie vrcholov do existujúcej triangulácie. Časová zložitosť tejto techniky nie je optimálna $T : \mathcal{O}(n^2)$, ale pre rovnomerne rozložené body sa dá očakávať lineárne správanie. Kvôli rýchlej lokalizácii pri pridávaní vrcholov sa používajú špeciálne dátové štruktúry, čo zvyšuje pamäťovú náročnosť algoritmu. Vzniklo veľa modifikácií tejto metódy, ich popis a nové vylepšenia z hľadiska časovej a pamäťovej zložitosti je možné nájsť v práci Kolingerovej a Žalika [KZ01].

Posledný z uvedených algoritmov, nazývaný ako *Lawsonova optimalizačná procedúra* [Law77] (známa aj ako lokálna optimalizačná procedúra alebo preklápací algoritmus) patrí medzi obľúbené nástroje výpočtovej geometrie. Táto technika vyžaduje začiatočnú trianguláciu množiny ako vstup (môže byť ľubovoľná), a jej úlohou je optimalizácia siete tak, aby výsledok bol LOT. Jedná sa o iteračnú procedúru, pri ktorej sa testuje lokálna optimálnosť hrán. Iterácia prebieha dovtedy pokiaľ existuje hrana, ktorá nie je lokálne optimálna, konečnosť takéhoto postupu bude dokázaná neskôr. Pseudokód algoritmu je znázornený na Obrázku 2.19. Napriek rozdielnosti medzi DT a DDT, tento algoritmus môže byť použitý pre konštrukciu lokálne optimálnych triangulácií v zmysle DDT. V ďalších úvahách bude použitá Lawsonova optimalizačná procedúra s počiatočnou trianguláciou DT pre účely získania LOT pre DDT, ak to nebude ináč vyznačené.

Poznámka. Nie je zanedbateľná správna voľba začiatočnej triangulácie, závisí od nej kvalita výslednej LOT. Pri nevhodnej počiatočnej triangulácie môže nastať situácia, že výsledná triangulácia síce bude lokálne optimálna, ale ďaleko (počtom preklopení) od globálneho optima. Inými slovami, preklápacie operácie majú lokálny charakter a optimalizačná procedúra sa niekedy "zasekne" v lokálnom optime.

Lawsonova optimalizačná procedúra - LOP
Vstup: počiatočná triangulácia
Výstup: lokálne optimálna triangulácia - LOT

```
{
  while(triangulácia nie je lokálne optimálna){
    if(e nie je lokálne optimálna)
      vymeň e za alternatívnu diagonálu;
  }
}
```

Obr. 2.19: Lawsonova optimalizačná procedúra - pseudokód.

Ako optimalizácia uvedeného postupu sa používa vytvorenie zoznamu hrán, kde sa zaradia hrany, ktorých lokálnu optimalitu je potrebné testovať. V prvom kroku algoritmu sa do zoznamu zaradia všetky hrany, odkiaľ po vykonaní testu lokálnej optimality sú postupne odstránené. V prípade ak hrana nie je lokálne optimálna, tak sa vymení za svoju alternatívnu diagonálu. Vtedy sa do zoznamu zaradia hrany, ktorých lokálna optimalita mohla zmeniť. V každom iteračnom kroku sa testujú hrany z tohto zoznamu, proces skončí, ak zoznam je prázdny.

Poznámka. Ľahké prispôsobenie LOP k danej problematike je dôvodom jej obľúbenosti. Napríklad, ak ohodnotenie hrán vyhovuje kritériu prázdneho kruhu, tak výsledná triangulácia je DT.

Veta 2.2.3. *[Noc02] Počet iterácií v Lawsonovej optimalizačnej procedúre je konečný a maximálny počet preklápaní je $\mathcal{O}(n^2)$.*

Dôkaz. Počet možných triangulácií danej množiny je konečný, a s každou iteráciou sa zmenší váha triangulácie. Teda musí existovať minimum (lokálne alebo globálne) ku ktorému sa uvedeným iteračným spôsobom algoritmus dopracuje. Hrana, ktorá už raz bola odstránená (preklápaná) viac sa už nevyskytne v triangulácii počas priebehu optimalizácie. Z počtu možných hrán medzi bodmi množiny s početnosťou n teda vyplýva ohraničenie $\mathcal{O}(n^2)$.

Výsledkom LOP je aproximácia MWT, nájdenie globálne optimálneho stavu nie je garantované. Vo všeobecnosti problém lokálne optimálnych triangulácií je v tom že sa nedá rozhodnúť či dané riešenie je globálne optimálne[5]. Na riešenie tejto otázky by bolo potrebné preskúmať všetky možné triangulácie. Napríklad MWT je tiež lokálne optimálna triangulácia. Ak by to tak nebolo, tak vymenením diagonál vo vnútri striktne konvexných štvoruholníkov by sa dalo dosiahnuť trianguláciu s menšou váhou než MWT.

Veta 2.2.4. *[Noc02] Pre ľubovoľnú množinu vrcholov* $\boldsymbol{V}$ *platí, že hrana, ktorá sa nachádza v každej* LOT *množiny* $\boldsymbol{V}$*, patrí aj do* MWT*. Naopak ak platí, že hrana sa nenachádza v žiadnej* LOT *množiny* $\boldsymbol{V}$*, tak ani do* MWT *nemôže patriť.*

Dôkaz. Táto veta platí aj pre DDT, tvrdenie priamo vyplýva z lokálnej optimality MWT.

Dôsledkom uvedenej vety je možnosť rozdelenia hrán do troch disjunktných množín: hrany vyskytujúce sa v každej LOT; hrany, ktoré nie sú v žiadnej LOT; hrany, ktoré sa vyskytujú len v niektorých LOT. Takto je možné získať doteraz najväčší známy podgraf MWT nazývaný ako *LMT-skeleton* [CKS96].

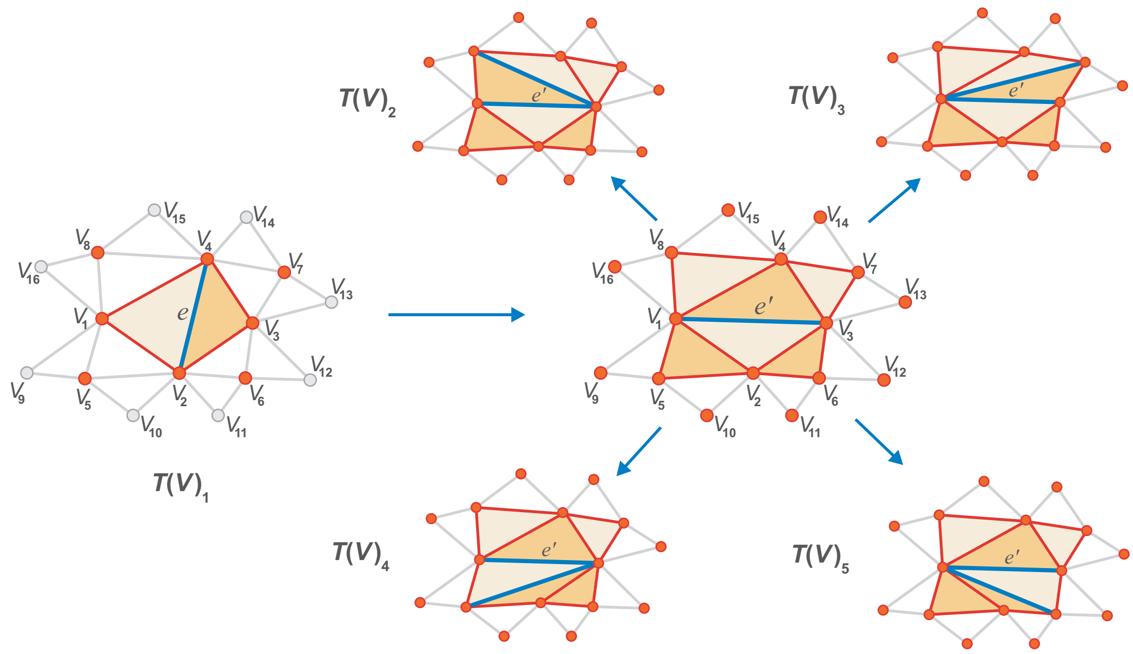
V prípade rovinných triangulácií cieľom je získanie súvislého podgrafu MWT, ktorý obsahuje v sebe len také netriangulované oblasti, ktoré sa dajú reprezentovať ako jednoduchý polygón. Jednoduché polygóny v rovine sa dajú optimálne triangulovať v čase $T : \mathcal{O}(n^3)$ Gilbertovým algoritmom, jeho popis sa nachádza v [PS85]. Vo všeobecnosti LMT-skeleton negarantuje tieto predpoklady. Ďalším obmedzením je, že pre DDT prístupy nie je možné zovšeobecniť Gilbertov algoritmus.

Look-ahead rozšírenie

Rozšírenie Lawsonovej optimalizačnej procedúry pod názvom *look-ahead* (ozn. LAT) poskytli autori Yu a spol. v práci [YBS01]. Ich motiváciou bolo zlepšenie rekonštrukcie izočiar pri rekonštrukcii obrazov.

Hlavnou ideou úpravy pôvodného algoritmu je v prehľadávaní hlbšieho okolia testovanej hrany pri minimalizácii ceny triangulácie. Modifikácia sa vzťahuje na konfigurácie keď testovaná hrana, ozn. e, je diagonálou striktne konvexného štvoruholníka (ozn. $V_1V_2V_3V_4$), ale kvôli hodnôt cenových funkcií nie je možné ju vymeniť za alternatívnu diagonálu e'. LOP by v takej situácii pokračoval ďalej v testovaní ďalších hrán. Rozšírenie LAT spočíva v podmienečnom preklápaní e na hranu e', s tým, že zároveň jedna z hrán $V_1V_2V_3V_4$ sa tiež preklopí a cena triangulácie tak celkovo poklesne. Teda okrem preklápania e na e' sa postupne testujú jednotlivé hrany z $V_1V_2V_3V_4$ pokiaľ sa nenájde riešenie s nižšou cenou ako pôvodná konfigurácia s hranou e. V prípade že ani jedna zo štyroch možností nie je vyhovujúca, sa použije triangulácia s hranou e. Situácia je znázornená na Obrázku 2.20.

[5]Delaunayova triangulácia je špecifická v tomto smere, tu nájdenie lokálne optimálneho stavu garantuje aj globálnu optimalitu.



Obr. 2.20: Hľadanie optimálnej triangulácie okolia hrany e pomocou LAT.

Na vyčíslenie lokálnej optimality pri LOP prístupe bola použitá cena z piatich hrán (dáta z ôsmich vrcholov). Rozšírenie LAT je v tomto smere náročnejšie a potrebuje vyčísliť cenu trinástich hrán (dáta maximálne až zo šestnástich vrcholov). Pseudokód algoritmu je uvedený na Obrázku 2.21.

Poznámka. Vyčíslenie optimality jednotlivých prípadov pri skúšobnom preklápaní hrán z $V_1V_2V_3V_4$ je výpočtovo drahá operácia. Preto pri prvom nájdenom riešení, ktoré dáva nižšiu cenu, algoritmus pokračuje ďalej v nasledujúcom kroku. Dalo by sa hľadať a vybrať aj najefektívnejšie riešenie, alebo rovno hladať globálne optimálnu trianguláciu pre mnohouholník vytvorený z ôsmich vrcholov. Pravdepodobne by takto získaná váha triangulácie bola nižšia ako v prípade uvedeného prístupu.

Počet iterácií v LAT je konečný. Vyplýva to z faktu, že počet triangulácií danej množiny je konečný, a pri každej iterácii sa zmenší celková váha. Ďalej platí, že cena triangulácie konverguje rýchlejšie (alebo aspoň rovnako rýchlo) ako v prípade LOP. Teda v priemernom prípade dá sa očakávať lepšie aproximačné správanie. Nájdenie globálneho optima však ani v tomto prípade nie je garantované. Vo všeobecnosti platí nasledujúci vzťah medzi cenami triangulácií množiny $\boldsymbol{V}$:

$$c(MWT(\boldsymbol{V})) \leq c(LAT(\boldsymbol{V})) \leq c(LOP(\boldsymbol{V}))\,. \tag{2.19}$$

V testoch [YBS01] modifikácia LAT vytvárala v každej iterácii lacnejšiu trianguláciu než LOP. Autori poznamenali, že niekedy už po prvej iterácii bol výsledok dostatočne kvalitný. Ako aplikačná oblasť bola zvolená rekonštrukcia obrazu.

Poznámka. Uvedený look-ahead prístup sa môže označiť ako look-ahead prvého stupňa. Generalizácia tejto myšlienky, ktorá optimalizuje väčšie okolie danej hrany,

Look-ahead triangulácia - LAT
Vstup: počiatočná triangulácia
Výstup: look-ahead triangulácia

```
{
  while(zmena váhy triangulácie ≠ 0){
    if(e nie je lokálne optimálna)
      vymeň e za alternatívu diagonálu;
    else if(e leží v striktne konvexnom štvoruholníku ){    //ozn. V1V2V3V4
      Σ_original = suma cenových funkcií 13 hrán;           //pre hranu e a jej okolie
      vymeň e za alternatívnu diagonálu;
      for(všetky hrany f ∈ V1V2V3V4){
        if(f leží v striktne konvexnom štvoruholníku){
          f vymeň za alternatívnu diagonálu;
          Σ_changed = suma cenových funkcií 13 hrán v zmenenej triangulácii;
          if(Σ_changed < Σ_orginal)
            goto riadok 2;
          else
            vráť naspäť hranu f;
        }
      }
      vráť naspäť hranu e;    //nenašlo sa lepšie riešenie
  }
}
```

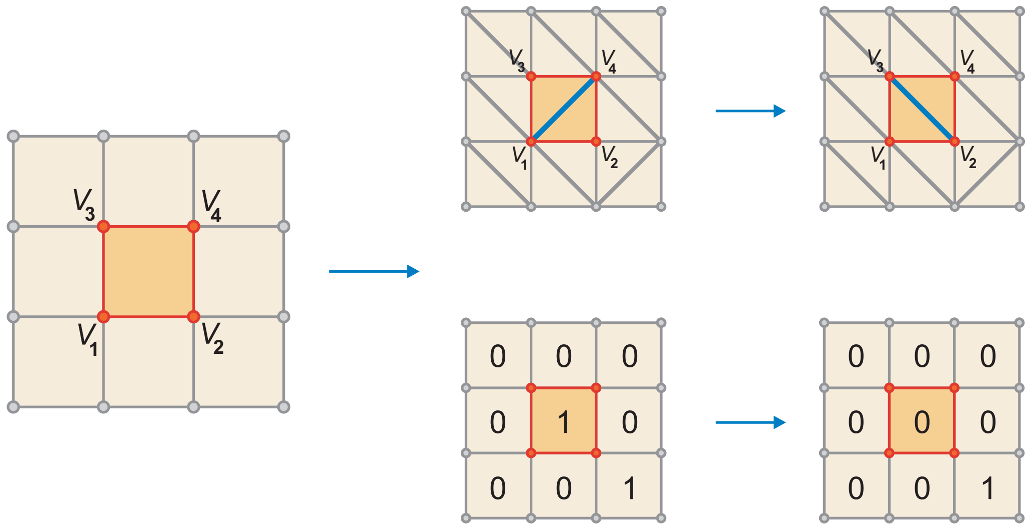
Obr. 2.21: Look-ahead triangulácia - pseudokód.

môže priniesť kvalitnejšiu rekonštrukciu a rýchlejšiu konvergenciu. Cenou za to je väčšia výpočtová zložitosť pri hľadaní optimálnej konfigurácie okolia v jednotlivých iteráciách. Dá sa očakávať rýchlejšiu konvergenciu ceny triangulácie, čo sa týka počtu iteračných krokov. Na druhej strane je tiež pravdepodobné, že look-ahead nižšieho rádu s väčším počtom iteračných krokov môže produkovať z hľadiska kvality porovnateľné výsledky.

Poznámka. Okrem uvedeného look-ahead prístupu existujú aj iné formy využitia tejto myšlienky. Kolingerová v [Kol04] predstavila aproximáciou rovinnej *MWT*, ktorá kombinuje myšlienku greedy prístupu s look-ahead vyhľadávaním pridaných hrán.

DDT na úrovni pixlov

Ďalšia metóda patriaca do skupiny deterministických prístupov pochádza od autorov článku [SW04]. Prínosom ich metódy je skôr v praktickom vnímaní *DDT* a nejedná sa o snahu dosiahnuť lepšiu aproximáciu *MWT*. Autori nazvali túto metódu *DDT na úrovni pixlov* (*pixel level DDT*) vzhľadom na voľbu aplikačnej oblasti, ktorou je rekonštrukcia obrazu. Pri konštrukcii triangulácie sa využíva dodatočná informácia, že dáta sú organizované do karteziánskej mriežky. Z toho vyplýva aj limitácia vzhľadom na distribúciu vstupnej množiny, pre ľubovoľnú vstupnú množinu nie je možné uplatniť tento prístup.



Obr. 2.22: Základná štruktúra *DDT* na úrovni pixlov (vľavo), triangulácia jednotlivých štvorcov (stred), a zohľadnenie orientácie okolitých diagonál (vpravo).

Algoritmus *DDT* na úrovni pixlov pracuje nasledujúcim spôsobom. Na začiatku rozdelí obraz do karteziánskej mriežky, kde každý štvorec je vytvorený zo štyroch susedných pixlov. Za pozíciu pixla sa berie jeho umiestenie a intenzita je braná ako funkčná hodnota. Časť takejto štruktúry je zobrazená v ľavej časti Obrázku 2.22. Následne triangulácia vytvorených štvorcov (napr. $V_1V_2V_3V_4$) prebieha na základe jednoduchého testu. Vloží sa diagonála, ktorej diferencia funkčných hodnôt koncových bodov je menšia, teda:

$$\min(|f(V_1) - f(V_3)|, |f(V_2) - f(V_4)|) \ . \tag{2.20}$$

Pri výbere vhodnej diagonály sa využije informácia len zo štyroch vrcholov. Adekvátna rekonštrukcia hrán v obraze vyžaduje viac informácií, a preto takto vložené diagonály nevystihujú vždy dostatočne smerovanie hrán. Ako snaha odstrániť tento nedostatok vzniklo rozšírenie metódy, kde digitálny obraz sa vníma ako Markovove náhodné pole [PL89]. V zmysle tejto teórie sa predpokladá takzvaná lokálna stacionarita, teda že intenzita pixlov je závislá od svojho okolia (susedných pixlov) a nezávislá od ostatnej časti. V tomto prípade to znamená že pri výbere diagonály sa dá zohľadniť výber diagonál z okolitých ôsmich susedných štvorcov (v zmysle osemsusednosti). Nech na základe orientácie vybranej diagonály sa priradia k štvorcom hodnoty {0,1}, konkrétny príklad je zobrazený na Obrázku 2.22 v strede dole. Ak v ôsmich okolitých štvorcoch je orientácia hrán signifikantná v jednom smere, tak výber na základe cenovej funkcie sa nahradí orientáciou signifikantnej väčšiny, vpravo dole v Obrázku 2.22. Takto sa využije informácia zo šestnástich vrcholov pri výbere vhodnej diagonály. Z praktického hľadiska to znamená zvýšenie kvality rekonštrukcie hranových oblastí. Pseudokód takto modifikovaného algoritmu je znázornený na Obrázku 2.23.

DDT na úrovni pixlov
Vstup: digitálny obraz
Výstup: *DDT* triangulácia na úrovni pixlov

```
{
  vytvor karteziánsku mriežku z digitálneho obrazu;
  for(pre všetky štvoruholníky)
    trianguluj štvoruholník optimálne vložením vhodnej diagonály;
  for(pre všetky štvoruholníky)
    if(orientácia diagonál v okolitých štvoruholníkoch je signifikantné &&
    orientácia diagonály skúmaného štvoruholníka je opačná){
      vymeň diagonálu za alternatívnu hranu;
    }
}
```

Obr. 2.23: *DDT* triangulácia na úrovni pixlov - pseudokód.

Poznámka. Zo spôsobu konštrukcie algoritmu vyplýva, že *DDT* na úrovni pixlov je praktickým inžinierskym riešením, ktoré využíva filozofiu *DDT* s jednoduchou cenovou funkciou. Jeho výhodou je nízka časová náročnosť výpočtov, ktoré sú na úrovni náročnosti výpočtov bilineárnej interpolácie. To umožňuje použitie tejto techniky v reálnom čase. Vizuálna kvalita výsledkov je nižšia ako v prípade doteraz uvedených *DDT* metód, ale pritom je ešte stále nad úrovňou výsledkov dosiahnutých pomocou bilineárnej interpolácie. Využitie uvedeného prístupu na zlepšenie výstupov z digitálnych kamier bolo uvedené v [SW03].

2.2.2 Stochastické prístupy

Okrem deterministických metód existujú aj prístupy, ktoré sa snažia pri hľadaní *MWT* využiť náhodné kroky. Preto sa táto skupina označuje ako *stochastické prístupy*. Motiváciou je snaha dosiahnuť takú trianguláciu, ktorá sa dostane bližšie ku globálnemu optimu ako to umožnia existujúce deterministické prístupy. Riadenie náhodných krokov prebieha pomocou nastavenia parametrov. Úspešnosť a rýchlosť nájdenia optimálneho stavu je silne závislá na vhodnom nastavení týchto hodnôt.

Simulované žíhanie

Najznámejším stochastickým prístupom pre oblasť *DDT* je *simulované žíhanie* (*simulated annealing*, ozn. *SA*) [Sch93]. Jedná sa o iteratívnu techniku, ktorá je založená na poznatkoch o priebehu fyzikálnych procesov v prírode. Hľadanie triangulácie

MWT je analógiou hľadania globálne minimálneho energetického stavu kryštálov tekutín, pomocou postupného chladenia. Simulované žíhanie ako optimalizačná metóda má široké uplatnenie v rôznych vedných odboroch, kde sa vyskytuje problematika optimalizácie, ako napríklad fyzika, matematika, informatika alebo ekonómia. Vhodným nastavením konverguje rýchlejšie ako metódy, ktoré vždy berú do úvahy len najlepšie lokálne riešenia.

Základným princípom SA je snaha vyhýbať sa lokálne optimálnym riešeniam, ktoré môžu vzbudzovať dojem globálnej optimality. Obeťami takýchto pomýlení sa často stávajú výsledky, ktoré poskytujú deterministické metódy. Na rozdiel od týchto prístupov okrem zlepšujúcich krokov sa akceptujú aj zhoršujúce kroky v jednotlivých iteráciách. *Krokom* sa nazýva výmena diagonál v striktne konvexnom štvoruholníku. Pri každom iteračnom kroku sa nastaví hranica možných zlepšujúcich krokov. Zhoršujúce kroky sa akceptujú náhodne a rastúcim počtom iteračných krokov sa zmenšuje pravdepodobnosť ich vykonania (toto sa nazýva "*chladiaci proces*"). Pseudokód algoritmu je znázornený na Obrázku 2.24.

Simulované žíhanie - SA
Vstup: počiatočná triangulácia
Výstup: *SA* triangulácia

```
{
  for(k=1; k ≤ ntemp; k++){
    t_A = r^k t_0;
    for(l=1; l ≤ nlimit; l++)
      while(počet dobrých krokov ≤ glimit){
        vyber náhodne hranu e z triangulácie;
          if(existuje alternujúca diagonála ku e){
            if(e nie je optimálne triangulovaný){
              vymeň diagonálu;
            }
            else{
              náhodný výber čísla φ, 0 ≤ φ ≤ 1;
              if(φ ≤ e^(-c(e)/t_A))
                vymeň diagonálu;
            }
          }
      }
  }
}
```

Obr. 2.24: Simulované žíhanie - pseudokód.

Vysvetlenie parametrov v uvedenom pseudokóde:

— *ntemp* - počet iterácií (v koľkých krokoch je teplota znížená), štandardné nastavenie tejto hodnoty je okolo 20.

— *nlimit*, *glimit* - počet vykonaných krokov v každej iterácii, počet vykonaných správnych krokov v každej iterácii. Zvyčajne sa volia tieto čísla ako 5− alebo 10− násobok počtu hrán v triangulácii.

— t_0, t_A - štartovacia teplota (zvyčajne 0.1), aktuálna teplota.

— $r-$rýchlosť "*chladenia*", jeho hodnota je zvyčajne 0.95 alebo 0.9.

— pod označením termínom "*dobrý krok*" sa rozumie taká výmena diagonál, ktorá zredukuje celkovú váhu triangulácie.

Simulované žíhanie je veľmi citlivé na nastavenie parametrov. Ak sa spraví príliš veľa správnych krokov na začiatku, tak s veľkou pravdepodobnosťou sa nájde len lokálne optimum. Zvolením štartovacej teploty sa dá nastaviť rýchlosť konvergencie. Pri zvolení rýchlej konvergencie sa preskúša menej možností, čím sa zvýši pravdepodobnosť, že metóda nájde len lokálne optimum. Na druhej strane pomalá konvergencia znamená veľkú časovú náročnosť výpočtov. Vzhľadom na špeciálne rozloženie bodov je rozumné experimentovať s nastaveniami parametrov *SA*, lebo uvedené hodnoty boli navrhnuté pre všeobecný prípad.

Poznámka. Zovšeobecnené simulované žíhanie sa nazýva aj ako Metropolisov algoritmus, základy v tomto smere boli publikované v práci [MRR*53]. Ako autori metódy simulovaného žíhania sa najčastejšie uvádzajú Černý [Č85] a Kirkpatrick a spol. [KGV83].

Genetická optimalizácia

Ďalší stochastický prístup ponúka možnosť ako skombinovať rôzne optimalizačné metódy a kritériá. Jedná sa o kombináciu deterministickej a stochastickej metódy, ktorá je založená na napodobňovaní evolúcie, a preto sa nazýva *genetická optimalizácia*, ozn. *GO*. Je to iteračná technika, kde v jednotlivých iteračných krokoch sa vytvára množina triangulácií, takzvaná "*populácia*". Pre účely dosiahnutia čo najlepšej aproximácie globálneho optima sa používajú podobné princípy ako v prípade evolúcie, kríženie a mutácia. Priebeh optimalizácie vyzerá nasledovne.

V prvom kroku sa vygeneruje množina triangulácií pre daný vstup, napríklad pomocou doteraz popísaných algoritmov. V terminológii evolučnej teórie sa táto množina nazýva "*úvodnou populáciou*". Ku každému prvku vygenerovanej množiny sa priradí hodnota podľa toho, ako dobre aproximuje cieľový stav, *MWT*. Táto hodnota vyjadruje pravdepodobnostnú hodnotu schopnosti "*reprodukcie*" v zmysle evolúcie. V každom iteračnom kroku sa vytvára nová populácia. Pričom počet členov v každej generácii počas celého procesu je konštantný. Prvky sú vybrané z prechádzajúcej generácie podľa ich pravdepodobnostných hodnôt. To spôsobí, že lepšie triangulácie sa tam vyskytujú viackrát a nevhodné sa eliminujú.

"*Prekríženie*" (*crossover*) je jeden z procesov, ktoré vplývajú na vývoj populácie. Jedná sa o binárnu operáciu, ktorá skombinuje dva prvky populácie (triangulácie), pričom prvky sa vyberajú náhodne. Inými slovami ide o výmenu genetickej informácie. Jej priebeh vyzerá nasledovne: obidve triangulácie sa rozdelia priamkou a potom sa skombinujú časti, ktoré zostali na rôznych stranách po rozdelení. Nech sú dané triangulácie A a B, a ich rozdelenia náhodnou priamkou na A_1, A_2, B_1, B_2. Výsledné triangulácie budú vytvorené z jednotenia $A_1 \cup B_2$ a $A_2 \cup B_1$. Z hľadiska výpočtovej náročnosti je táto operácia najviac náročná - $T : \mathcal{O}(\log n)$. Ďalšia operácia ktorá slúži na modifikáciu populácie je unárna operácia "*mutácie*". V celej populácii pre každú hranu s malou pravdepodobnosťou (zvyčajne $p < 0.01$) sa spraví preklápanie hrany. Táto operácia má zložitosť $T : \mathcal{O}(n)$. Týmto sa dosiahne mierna mutácia (modifikácia) jednotlivých prvkov populácie.

Ani princíp genetickej optimalizácie nezaručí nájdenie globálneho optima v prípade problematiky *MWT*. Zvolenie takej úvodnej populácie triangulácií, ktorá veľmi dobre aproximuje daný problém, skrýva v sebe nebezpečie, že sa nájde len lokálne optimum. Ďalším pozorovaním je, že rôznorodosť tejto populácie zrýchli konvergenciu *GO*. Presný popis tejto metódy je možné nájsť v [Kol99, KF01], pseudokód algoritmu je znázornený na Obrázku 2.25.

Genetická optimalizácia - GO
Vstup: množina vrcholov
Výstup: optimálna triangulácia

{
  vyrátaj úvodnú populáciu P_0;
  inicializuj čas $T = 1$;
  while(nie je splnená ukončovacia podmienka**)**{
    $T = T + 1$;
    aplikuj operácie mutácie a prekríženia na celú populáciu;
    vyber novú generáciu P_T z P_{T-1};
  }
  vráť najlepšie riešenie problému;
}

Obr. 2.25: Genetická optimalizácia - pseudokód.

Ukončovacou podmienkou optimalizačného procesu môže byť určitý počet generácií alebo malá odlišnosť v ohodnotení generácií. Za výsledok optimalizácie sa vyberie triangulácia s najvýhodnejšími vlastnosťami z poslednej populácie (alebo spomedzi všetkých populácií, v závislosti od spôsobu implementovania). Genetická optimalizácia je asymptoticky lacnou trianguláciou - $T : \mathcal{O}(n \log n)$, ak si vytvorenie počiatočnej populácie nevyžiada vyššiu časovú náročnosť ako $T : \mathcal{O}(n \log n)$. Napriek tomu aj pri voľbe lacných triangulácií (napríklad *DDT* na úrovni pixlov) zostáva úloha vyriešiť náročnú operáciu prekríženia triangulácií v množstve generácií. Autori odporúčajú 100 generácií po 50 členov populácie. To znamená preskúmanie až 5000 triangulácií, čo je výpočtovo veľmi náročné, a preto praktické využitie tejto metódy je nevýhodné. A to aj napriek tomu, že voľba jednotlivých operácií môže byť aj iná, dávajúc veľkú pestrosť GO.

2.2.3 Alternatívne prístupy

Okrem uvedených deterministických a stochastických prístupov existujú aj ďalšie techniky ktoré sú relevantné pre túto prácu.

Ko-triangulácia

Metóda založená na iteratívnom zjemnení siete pre veľké objemy dát bola predstavená v článku [WWT*98] pod názvom *ko-triangulácia*. Motiváciou je aproximácia rôznych rozptýlených dát nad tou istou doménou, ako napríklad rôzne klimatické, geologické či medicínske dáta.

Základnou myšlienkou je použitie aproximácie, ktorá je vytvorená pomocou po častiach lineárnych funkcií vo vyšších dimenziách. Pre N dátových množín, ktoré reprezentujú N rôznych $D-$rozmerných nezávislých funkcií, ko-triangulácia zostrojí $D-$rozmerný po častiach lineárny objekt v $D+N$ rozmernom priestore. N ortogonálnych projekcií tohto objektu dáva dobrú aproximáciu pôvodných N dátových množín. Takto získaný $D-$rozmerný lineárny objekt poskytuje aproximáciu pre N rôznych funkcií definovaných na tom istom obore.

V prípade aproximácie obrazu umiestnenie pixlov reprezentuje spoločnú doménu (rovinné dáta) na ktorom sú zadefinované tri nezávislé funkcie (farebné informácie RGB). V 5D priestore sa zostrojí 2D po častiach lineárna funkcia tak, že ortogonálne projekcie do tých súradnicových osí, ktoré nesú farebnú informáciu, sú trianguláciami.

V praxi vytvorenie ko-triangulácie vyzerá nasledovne. Zoberie sa prvá množina rozptýlených dát a vytvorí sa jej aproximácia pomocou triangulácie (pričom sa využíva Delaunayova triangulácia). Potom sa postupne zoberú ďalšie množiny rozptýlených dát a ku vrcholom existujúcej triangulácie sa pridávajú ďalšie vrcholy tak, aby dostatočne aproximovali danú množinu (miera aproximácie sa dopredu určí). Každý vrchol obsahuje N rôznych funkčných hodnôt, v prípade obrázkov farebné informácie (hodnoty RGB). Presný spôsob zostrojenia ko-triangulácie sa nachádza v [WWT*98].

Poznámka. Použitie tejto metódy na rekonštrukciu funkčných hodnôt nie je dobrou voľbou. Nahradenie Delaunayovej triangulácie dátovo závislými metódami by bolo zaujímavejšie pre tento problém. Pre účely aproximácie obrazu ďalšie vylepšenia by mohlo priniesť voľba iného farebného priestoru ako RGB, napríklad použitie perceptuálne lineárneho modelu $L^*a^*b^*$.

Vrcholovo založené ohodnotenie

Pri popise cenových funkcií bola naznačená existencia aj iných prístupov ako priradenie cien k hranám. Metóda založená na ohodnotení vrcholov bola predstavená v článku [Bro91]. Za cenu triangulácie sa v takomto prípade považuje suma cenových funkcií všetkých vrcholov $V_i \in \boldsymbol{V}$:

$$c(\boldsymbol{T(V)}) = \sum_{V_i \in \boldsymbol{T(V)}} \|c(V_i)\|, \tag{2.21}$$

kde $c(V_i)$ značí cenovú funkciu vrchola V_i. Pri konštrukcii vrcholovo založených cenových funcií sa dajú využiť poznatky z hranovo založených prístupov. Za cenu vrchola je možné zvoliť sumu cien incidentných hrán, alebo sumu štvorcov cien týchto hrán.

Okrem adaptácií už existujúcich cenových funkcií sa dajú vytvoriť aj nové cenové funkcie. Autor vyššie spomenutého článku zadefinoval za cenu vrchola *po častiach lineárnu analógiu krivosti*, ozn. *PLC* (*piecewise-linear analogy of curvature*). Nech je daný vrchol $V_j \in \boldsymbol{V}$, a k nemu incidenté hrany e_1 až e_m zoradené okolo vrchola V_j. Normály trojuholníkov (označené $\vec{n}_i$) obsahujúce V_j môžu byť vyjadrené pomocou smerových vektorov hrán e_i, označené ako $\vec{l}_i$, $i = 1, \cdots, m$, nasledujúcim spôsobom:

$$\vec{n}_i = \vec{l}_i \times \vec{l}_{(i+1) \bmod m} \ , \ i = 1, \cdots, m \, . \tag{2.22}$$

Po častiach lineárna verzia normálového vektora rekonštruovanej plochy vo vrchole V_j sa bude označovať $\vec{n}$. Dá sa ho vyčísliť sčítaním normál trojuholníkov, ktoré sú incidentné s V_j:

$$\vec{n} = \sum_{i=1}^{m} \frac{\vec{n}_i}{|\vec{n}_i|}. \tag{2.23}$$

Cenová funkcia pre vrchol V_j v zmysle *PLC* je definovaný nasledovne:

$$c(V_j) = \sum_{i=1}^{m} \left(cos^{-1} \left(\frac{\vec{n} \cdot \vec{n}_i}{|\vec{n}||\vec{n}_i|} \right) \right)^2 . \tag{2.24}$$

Uvedená definícia platí pre nedegerovaný prípad, keď vrchol V_j neleží na konvexnom obale množiny $\boldsymbol{V}$ (v priemete do roviny). Modifikácia, ktorá pokrýva aj tieto možnosti je triviálna a možné ju nájsť v [Bro91].

Celkovo je možné konštatovať, že konštrukcia cenových funkcií nie je taká intuitívna ako u hranovo založených *DDT*. Na druhej strane, jedná sa o všeobecnejší prístup, adaptácia existujúcich hranovo založených metód je bezproblémová.

Princíp získania *LOT* triangulácie pre vrcholovo založené *DDT* je analógiou Lawsonovej optimalizačnej techniky, ktorá bola popísaná v odseku 2.2.1. Aj v tomto prípade sa iteratívne testuje lokálna optimalita hrán a vykoná sa preklápanie, ak sa takto zníži celková cena triangulácie.

Z výsledkov testov na syntetických dátach uvedených v spomenutom článku vyplýva, že kvalita výslednej triangulácie je mierne lepšia, respektíve porovnateľná s výsledkami z hranovo založených metód. Nie sú ale uvedené porovnanie časov potrebné pre výpočty LOT. Dá sa ale predpokladať, že vrcholovo založený prístup vyžaduje výrazne náročnejšie výpočty kvôli vyčísleniu cenových funkcií.

Poznámka. Delaunayovu trianguláciu je možné vytvoriť pomocou vrcholovo založeného prístupu, ak za cenu vrchola je braný súčet Sobolevovej semi-normy trojuholníkov, ktoré sú incidentné s daným vrcholom. Sobolevova semi-norma trojuholníka v DDT je plocha trojuholníka v xy priemete, krát veľkosť gradientu interpolačnej plochy trojuholníka v DDT interpolácii [Rip90, Bro91].

Využitie *DDT* pri aproximácii rozptýlených dát

Kombinácia simulovaného žíhania a dátovo závislej techniky pre účely aproximácie rozptýlených dát bola predstavená v práci [KH01]. Optimálna sieť je vytvorená nasledujúcim spôsobom. Najprv sa určí počet vrcholov, ktorými bude reprezentovaná pôvodná množina dát. Výber týchto vrcholov je riadený metódou simulovaného žíhania (využíva sa iným spôsobom, ako to bolo uvedené v časti 2.2.2), je to vďaka snahe vyhnúť sa lokálne optimálnym riešeniam. Po dosiahnutí dostatočnej kvality výberu, vzhľadom na definovanú chybovú normu, metóda postupuje do druhej fázy, kde sa určí optimálna triangulácia pre vybraté vrcholy. Tu sa používa dátovo závislá technika. Autori poznamenali, že je možné súčasne optimalizovať výber vrcholov z množiny dát a hľadať optimálnu trianguláciu. V tomto prípade je výpočtová náročnosť metódy vyššia. Pre demonštráciu zvolili kompresiu obrazu, výsledky sú zobrazené na Obrázku 2.26.

Obr. 2.26: Aproximácia obrázku pomocou 800 vrcholov (ľavá strana) a pomocou 3200 vrcholov (pravá strana). Pôvodný obrázok mal rozlíšenie 256 × 256 pixlov. Zdroj obrázku [Kre].

Podobný prístup publikovali autori článku [BHJ99], kde skombinovali simulované žíhanie so zjemňovaním siete (*mesh refinement*). Výber vrcholov je riadený pomocou metódy postupného zjemňovania začiatočnej triangulácie (triangulácia konvexného obalu), a z toho vyplýva, že je aj viac limitovaný ako v predchádzajúcom prístupe. Optimálna triangulácia daných vrcholov je vyrátaná pomocou simulovaného žíhania.

Poznámka. Prehľad problematiky možno uzavrieť pozorovaním, že v danej oblasti prebieha intenzívny výskum a že všeobecne používané metódy sa zatiaľ len hľadajú.

Kapitola 3

Vlastné rozšírenia

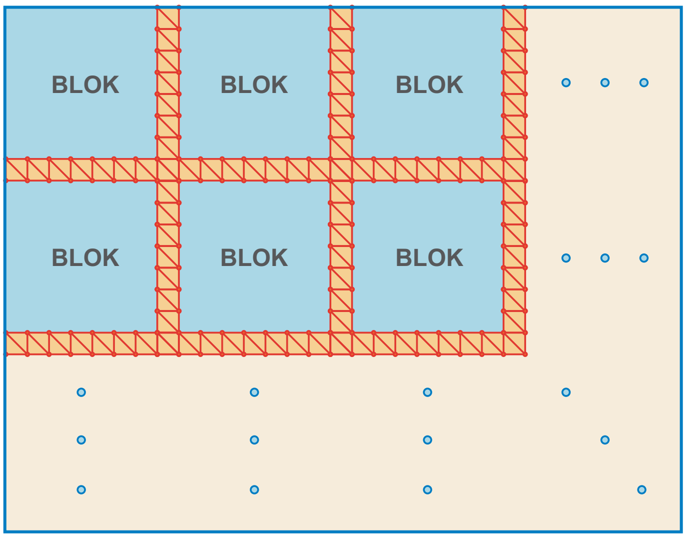
Časť popísaných algoritmov a prístupov z nasledujúcej kapitoly bola publikovaná v prácach [Tót04, Tót06, TVFG07, CTS*].

3.1 Rozdelenie na bloky

Metódy uvedené v prehľadovej časti sú kompromisom medzi rýchlosťou nájdenia riešenia a kvalitou výsledku. Napríklad simulované žíhanie pomocou vhodne nastavených parametrov je schopné nájsť riešenie blízke ku globálnemu optimu, na druhej strane DDT na úrovni pixlov je veľmi efektívna z hľadiska výpočtovej náročnosti. V snahe kombinovať výhodné vlastnosti existujúcich prístupov vznikol nasledujúci návrh, ktorý je možné aplikovať na dáta organizované do karteziánskej mriežky.

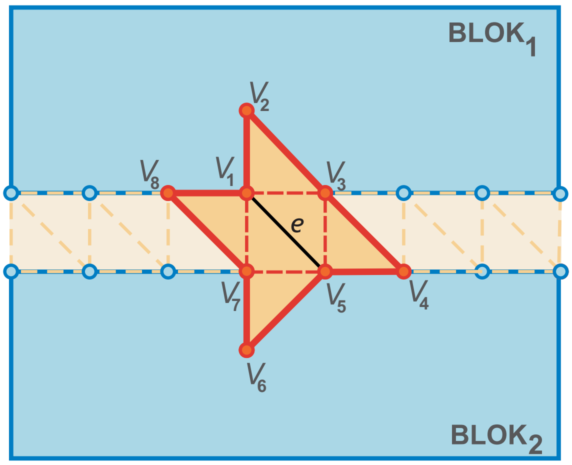
Základnou myšlienkou a cieľom je vytvorenie takej štrukturalizácie dát, ktorá umožní reálne použitie dátovo závislej techniky, a pritom nedegraduje rekonštrukčné vlastnosti, ako napríklad DDT na úrovni pixlov. Štrukturalizácia dát je založená na skúsenostiach o správaní DDT metód pri rekonštrukcii obrazu. Pri tejto aplikačnej oblasti euklidovská dĺžka hrán v 2D priemete výsledných triangulácií je málo rozptýlená. Teda tieto triangulácie majú skôr lokálny charakter, extrémne dlhé hrany sa v nich vyskytujú len výnimočne. Na tomto základe je postavená myšlienka rozdeliť obraz na bloky - obdĺžniky, pričom každý vrchol musí ležať práve v jednom bloku. Vo vnútri jednotlivých blokov sa vyrieši aproximácia MWT už so známymi metódami, alebo sa vygenerujú všetky možnosti a vyberie sa z nich optimálne riešenie. V nasledujúcej fáze je potrebné prepojiť susedné bloky. Ako počiatočná štruktúra triangulácie medzi blokmi bude braná regulárna triangulácia, situácia je znázornená na Obrázku 3.1.

Hlavným problémom je vyriešenie prepojenia jednotlivých blokov takým spôsobom, aby to vzbudzovalo vizuálne príjemný dojem, a pritom by nešlo o výpočtovo náročný proces. Na tento účel môže byť použitý podobný princíp preklápania hrán ako v prípade metódy look-ahead. Nech *zoznam aktívnych hrán* označuje množinu hrán, ktoré nepatria do žiadneho bloku a v počiatočnej (regulárnej) triangulácii majú inú smerovú orientáciu ako súradnicové osi. Ide o diagonály štvorcov, ktoré



Obr. 3.1: Štruktúra segmentovania na bloky, regulárna počiatočná triangulácia medzi jednotlivými blokmi.

sa využívajú na prepájanie blokov. Na Obrázku 3.2 je zvýraznená hrana e, ktorá je členom zoznamu aktívnych hrán. K danej hrane sa zoberie okolie takej veľkosti ako v prípade look-ahead prístupu. Táto oblasť je vyznačená s mnohouholníkom vytvoreným z ôsmich vrcholov $V_1V_2V_3V_4V_5V_6V_7V_8$, ďalej označené len ako $\square V_{1-8}$. Na Obrázku 3.2 je to zvýraznený hrubou čiarou. Je potrebné si uvedomiť, že vrcholy V_2 a V_6 závisia od výsledkov triangulácií blokov $Blok_1$ a $Blok_2$. Vyrátanie optimálnej triangulácie mnohouholníka $\square V_{1-8}$ zabezpečí vizuálne akceptovateľné prepojenie susedných blokov, pričom sa nemenia ostatné hrany zo zoznamu aktívnych hrán. Uvedený postup je potrebné aplikovať na všetky hrany zo zoznamu aktívnych hrán, čím sa získa dátovo závislé prepojenie všetkých blokov. Pseudokód algoritmu je znázornený na Obrázku 3.3.



Obr. 3.2: Hrana e zo zoznamu aktívnych hrán a jej okolie, mnohouholník $\square V_{1-8}$, ktorej optimálna triangulácia zabezpečí prepojenie blokov $Blok_1$ a $Blok_2$.

Dátovo závislá triangulácia pomocou rozdelenia na bloky
Vstup: digitálny obraz
Výstup: triangulácia

```
{
   rozdeľ obraz na bloky;
   trianguluj jednotlivé bloky;
   vytvor regulárnu trianguláciu medzi blokmi;
   vytvor zoznam aktívnych hrán;
   for(pre hrany zo zoznamu aktívnych hrán)
      vyrátaj optimálnu trianguláciu pre okolie hrany; // okolie ako v prípad look-ahead prvého stupňa
}
```

Obr. 3.3: Dátovo závislá triangulácia pomocou rozdelenia na bloky - pseudokód.

Veľkosťou a spôsobom triangulácie jednotlivých blokov je možné riadiť výpočtovú náročnosť. Príliš malé bloky indikujú horšiu kvalitu, nie je možné zostrojiť takú trianguláciu, ktorá by dostatočne vystihla vysokofrekvenčné oblasti. Veľké bloky zase môžu spomaliť výpočet, v závislosti od spôsobu zvolenej metódy triangulácie.

Ďalšou významnou výhodou uvedenej metódy je, že jednotlivé bloky sa dajú optimalizovať paralelne. Súčasný vývoj hardveru ako aj nové štandardy (napríklad *OpenCL* [Khr] - spojenie výpočtovej sily grafických kariet a procesora) potvrdzujú dôležitosť takéhoto riešenia. Prepojenie blokov nemôže byť vykonané paralelne, ale výpočtová náročnosť tohto kroku je nižšia ako samotná triangulácia blokov. Pri súčasnom trende vývoja hardveru sa dá očakávať, že v budúcnosti sa bude dať riešiť rekonštrukcia obrazu pomocou uvedenej metódy v reálnom čase, čo by znamenalo možnosť uplatnenia v širokej škále aplikácií.

Spôsob triangulácie jednotlivých blokov je možné zvoliť v závislosti od potreby danej aplikačnej oblasti. V prípade rekonštrukcie obrazu pomocou hranových detektorov sa dá určiť významnosť blokov. Homogénne oblasti môžu byť triangulované pomocou *DDT* na úrovni pixlov a bloky, kde hranové detektory naznačujú vysokofrekvenčné oblasti sofistikovanejším spôsobom. Rozličný spôsob triangulácie blokov môže byť výhodou aj v prípade použitia triangulácií pri tvorbe textúry. Prioritu môže určiť vzdialenosť plochy od kamery, na ktorý je daný triangulačný blok aplikovaný ako textúra. Bloky nachádzajúce sa bližšie ku kamere by mali byť kvalitnejšie triangulované ako tie vzdialenejšie.

Uvedená metóda je aplikovateľná len na dáta organizované do karteziánskej mriežky, ako napríklad digitálny obraz alebo digitálne výškové mapy (*digital elevation map*). V prípade inej organizácie dát by bolo potrebné nájsť efektívne riešenie ako deliť dáta do blokov a ako tie bloky prepojiť. Vyriešenie týchto problémov nemusí byť priamočiare a výpočtová náročnosť môže prevýšiť výhody získané rozdelením na bloky.

Poznámka. Výsledná triangulácia nie je lokálne optimálna kvôli spôsobu prepojenia blokov. Treba si poznamenať, že splnenie tejto podmienky by v krajnom prípade mohlo znamenať zmenu celej triangulácie danej množiny. To znamená, že nie je možné nájsť taký spôsob prepojenia blokov, ktorý by zaručil lokálnu optimalitu a nízku výpočtovú náročnosť. V prípade rekonštrukcie obrazu uvedený spôsob prepojenia sa javí ako vhodný kompromis medzi rýchlosťou a kvalitou riešenia.

3.2 Simulované žíhanie s využitím look-ahead stratégie

Ako už bolo uvedené v prehľadovej časti simulované žíhanie v prípade dátovo závislých triangulácií slúži na získanie aproximácie *MWT*. Pri vhodnom nastavení parametrov s veľkou pravdepodobnosťou dosiahne globálne optimálny stav, pričom je to za cenu vyššej výpočtovej náročnosti. Zefektívnenie *SA* je cieľom nižšie uvedeného rozšírenia. V pôvodnom prístupe [Sch93] je myšlienka simulovaného žíhania spájaná s operáciou preklápania hrán. Dá sa to vnímať ako aplikáciu stochastického optimalizačného prístupu na *LOP*. Chápanie vzťahov v takomto kontexte je kľúčom k rozšíreniu, ktoré spája myšlienku simulovaného žíhania s look-ahead prístupom (ďalej to bude označené ako *SALA*) namiesto Lawsonovej metódy. Znamená to nasledujúcu zmenu.

V prípade klasického *SA* prístupu, ak bola testovaná hrana e lokálne optimálna, tak s určitou pravdepodobnosťou sa táto lokálna optimalita porušila výmenou diagonál. Ak sa tento zhoršujúci krok nevykonal, algoritmus pokračoval ďalej v spracovaní ďalších hrán. Aplikácia look-ahead prístupu bude znamenať ďalšie spracovanie hrany e, ak nenastane zhoršujúci krok vďaka simulovanému žíhaniu. V takomto prípade sa bude hľadať optimálne riešenie pomocou look-ahead prístupu pre hranu e a jej okolie. Takto sa dá očakávať zvýšenú rýchlosť konvergencie a lepšiu aproximáciu *MWT*. Pseudokód algoritmu *SALA* je zobrazený na Obrázku 3.4, pričom význam parametrov *ntemp*, *nlimit*, *glimit*, t_0, t_A, r je totožný s významom parametrov *SA*.

Nastavenie parametrov pre *SALA* môže byť iné ako v prípade *SA*, lebo má iné správanie (rýchlosť konvergencie a aproximačné vlastnosti). Ďalším novým parametrom môže byť aj stupeň look-ahead prístupu (dalo by sa uvažovať o zvýšení tohto stupňa a analyzovať jeho vplyv na správanie algoritmu). Úlohou zostáva nájdenie vhodných parametrov pomocou vykonania rozsiahlych experimentov pre rôzne aplikačné oblasti a distribúcie vrcholov. Toto je ale nad rámec a možnosti tejto práci, v prípade rekonštrukcie obrazu stačí použitie look-ahead optimalizácie hrany prvého stupňa. Uvedenú modifikáciu je možné uplatniť aj na iné účely ako je rekonštrukcia obrazu, je to všeobecnejšia metóda ako prístup rozdelenie na bloky.

Simulované žíhanie s využitím look-ahead - *SALA*
Vstup: počiatočná triangulácia
Výstup: *SALA* triangulácia

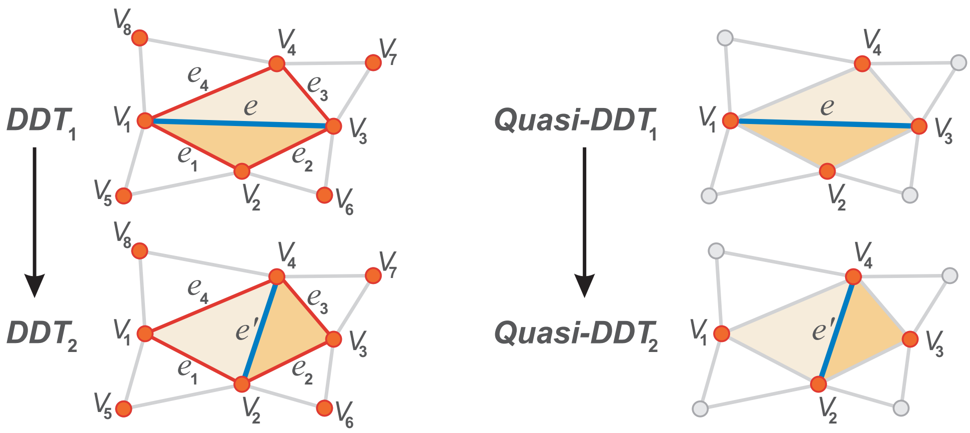
```
{
 for(k=1; k ≤ ntemp; k++){
  t_A = r^k t_0;
  for(l=1; l ≤ nlimit; l++)
   while(počet dobrých krokov ≤ glimit){
    vyber náhodne hranu e z triangulácie;
    if(existuje alternatívna diagonála ku e){
     if(striktne konvexný štvoruholník, ktorý obsahuje e nie je optimálne triangulovaný){
      vymeň diagonálu;
     }
     else{
      náhodný výber čísla φ, 0 ≤ φ ≤ 1;
      if(φ ≤ e^(−c(e)/t_A))
       vymeň diagonálu;
      else
       look-ahead optimalizácia e;
     }
    }
   }
 }
}
```

Obr. 3.4: Simulované žíhanie s využitím look-ahead prístupu - pseudokód.

3.3 Quasi - *DDT*

Zdrojom ďalšej metódy sú empirické poznatky získané z aplikačnej oblasti rekonštrukcie obrazu. Najznámejší prístup - Lawsonov optimalizačný proces - pri rekonštrukcii obrazu často vytvára v oblasti hrán tzv. “*dvojité hrany*”, medzi ktorými vygeneruje vrstvu trojuholníkov. Tieto trojuholníky slúžia na rekonštrukciu hladkých zmien cez hrany. Z teoretického hľadiska je to správny výsledok, metóda sa snaží vytvárať adekvátny farebný prechod cez hrany, ale práve tieto oblasti obsahujú mnoho artefaktov. Výsledok takejto rekonštrukcie je zobrazený na Obrázku 3.6 v hornom riadku.

Pri experimentovaní s rôznymi zmenami a nastaveniami sa podarilo nájsť prístup, ktorý zmierňuje množstvo vyššie spomenutých artefaktov. Zmena oproti pôvodnému prístupu spočíva v chápaní lokálnej optimality hrán. Napríklad pri cenovej funkcie *SCF* lokálna optimalita hrany sa štandardne posudzuje na základe ceny piatich hrán, nech testovaná hrana je označená e na Obrázku 3.5 ľavá časť. V takomto prípade je potrebné vyčísliť cenové funkcie $c(e), c(e_1), c(e_2), c(e_3), c(e_4)$ pred a po preklápaní hrany e. V novom prístupe lokálna optimalita je vyčíslená len na základe cenovej funkcie testovanej hrany e, t.j. $c(e)$ na Obrázku 3.5 pravá časť.



Obr. 3.5: Ilustrácia k lokálnej optimalite hrany e v prípade *DDT* - ľavá časť, a v prípade *Quasi-DDT* - pravá časť.

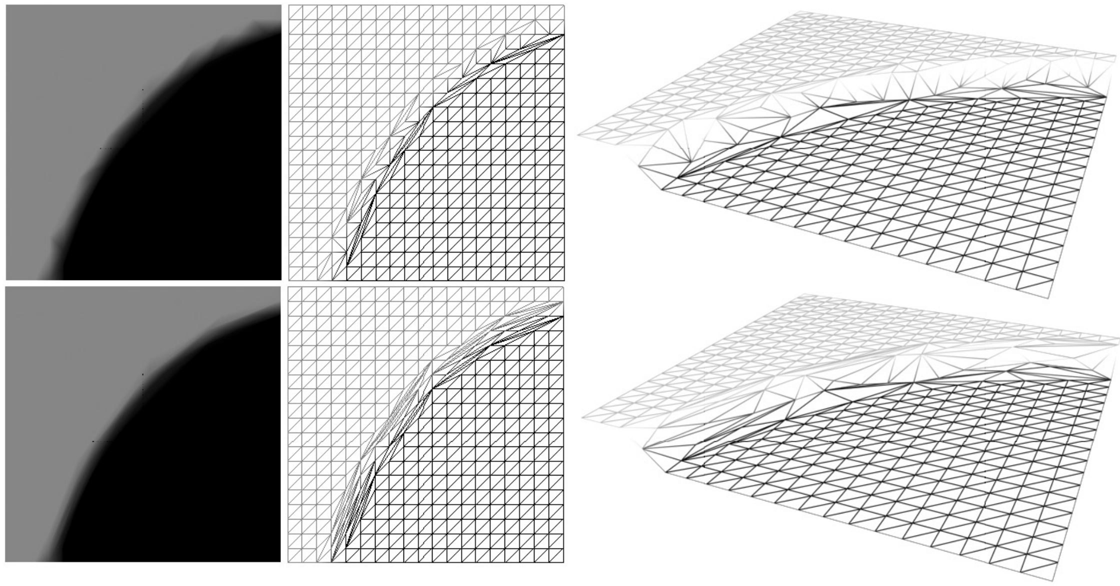
Takéto preklápanie hrany skrýva v sebe nebezpečenstvo, že celková váha triangulácie môže vzrásť. Zmeny v $c(e_1), c(e_2), c(e_3), c(e_4)$ nie sú brané do úvahy, a celkový prínos získaný preklápaním hrany e môže byť tým pádom negovaný. To znamená, že v tomto prípade nie je možné hovoriť o aproximácii *MWT*. Uvedenú zmenenú lokálnu optimalitu hrán je možné skombinovať s existujúcimi optimalizačnými technikami ako napríklad *LOP*, *LAT*, *SA*. Výsledné triangulácie nemusia byť lokálne optimálne. Tieto prístupy sa budú označovať predponou *Quasi-*, napríklad *Quasi-LOP* pre *LOP*, atď.

Napriek uvedeným negatívnym vlastnostiam táto metóda má praktický význam pre oblasť rekonštrukciu obrazov. V oblastiach, kde sa vyskytujú v obrazoch hrany, v niektorých prípadoch vytvárajú priaznivejšie výsledky ako klasické *DDT* metódy. Ako príklad je uvedené porovnanie medzi *LOP* a *Quasi-LOP* na Obrázku 3.6. V prípade *Quasi-LOP* (dolný riadok Obrázku 3.6) je vytvorený len jeden pás trojuholníkov a takto sa zlepší kvalita rekonštrukcie. Aj výpočtová náročnosť je nižšia ako v prípade *DDT* metód, vyčíslenie lokálnej optimality hrany je menej náročné.

Z praktického hľadiska ide o významnú skupinu metód, ktorá poskytuje alternatívu ku popísaným triangulačným prístupom. Pri rekonštrukcii vytvára artefakty iného charakteru ako klasické *DDT* metódy. V situáciách, kde uvedené metódy vytvárajú nepriaznivé výsledky, môžu priniesť kvalitnejšiu rekonštrukciu.

3.4 Rozšírenie *DDT* do vyšších dimenzií

V porovnaní s rovinným prípadom optimálne triangulácie vo vyšších dimenziách sú menej preskúmané. Je to aj vďaka rozličným (zložitejším) vlastnostiam a menšiemu



Obr. 3.6: Rekonštrukcia hrany pomocou *LOP* (horný riadok), *Quasi-LOP* (dolný riadok) metódy pri použití cenovej funkcie *SCF*.

počtu aplikačných oblastí. Tejto tematike sú venované časti prehľadových prác [BE95, Ede00].

V prehľadovej časti bol popísaný vzťah triangulácií *DT* a *LOP* pre rovinný prípad. V trojrozmernom priestore ale priamočiare zovšeobecnie Lawsonovho optimalizačného procesu vo vytváraní *DT* zlyháva. Algoritmus využívajúci postupné pridávanie vrcholov pre generovanie *DT* v trojrozmernom priestore bol predstavený v článku od Joe [Joe91]. Jedná sa o špeciálnu modifikáciu, rozšírenie prístupu, ktorá existuje pre rovinný prípad a využíva tiež inkrementálne pridávanie vrcholov. Napriek tomu ani táto metóda nevyhovuje pre účely vytvárania dátovo závislých triangulácií vo vyšších dimenziách.

Vo všeobecnosti Delaunayova triangulácia vo vyšších dimenziách nemá také priaznivé vlastnosti ako v rovinnom prípade, kde sa výsledok tejto triangulácie využíva na rôzne simulácie pomocou metódy konečných prvkov. Vo vyšších dimenziách sa tieto triangulácie často využívajú len ako úvodné triangulácie, ktoré sú ďalej optimalizované, aby sa z nich odstránili simplexy nevhodného tvaru. Odstránenie týchto elementov je dôležité kvôli rôznym numerickým aspektom. Vytváranie vhodných triangulácií v 3D pre metódu konečných prvkov bolo popísané v článkoch [MJ02, MTG04].

Ďalšie využitie triangulácií vo vyšších dimenziách môže byť snaha aproximovať pôvodnú dátovú množinu s menším počtom dát. Triangulácie zachovávajúce významné črty vo volumetrických dátach boli predstavené v prácach Marchesin a spol. [MDM04] a Roxborough a Nielson [RN00]. Autori zvolili prístup využívajúce zjemňovanie siete kombinované s metódou rozdelenia najdlhšej hrany (*longest edge split method*). Dôsledkom čoho tvar výsledných tetrahedronov je ale limitovaný kongruentnými kópiami tetrahedronov zo začiatočnej štruktúry. Cieľom dátovo závislých triangulácií nie je aproximovať dané dáta, ale korektne ich rekonštruovať. Kvôli tomu poznatky z uvedených zdrojov sa nedajú využiť v prípade konštrukcie $n-$dimenzionálnej DDT.

Dátovo závislá technika pre trojrozmerný priestor bola predstavená v práci Lee [Lee00]. Ako optimalizačná technika sa zvolila simulované žíhanie, a bola použitá topologická transformácia $3-2$ preklápanie. Za cenu triangulácie je braný súčet cenových funkcií, pričom cenové funkcie sú priradené k stenám tetrahedronov. Počet stien v trojrozmernej triangulácii je závislý od spôsobu triangulácie, čo v uvedenom prístupe nie je zahrnutý. Publikované výsledky takéhoto zovšeobecnenia DDT do trojrozmerného priestoru nie sú úplne korektné.

Rozšírenie dátovo závislých triangulácií do vyšších dimenzií uvedené v tejto časti je výsledkom spolupráce s TU Wien a bol publikovaný v [TVFG07].

3.4.1 Problém rozšírenia a jeho riešenie

Existujúce dátovo závislé prístupy z 2D sú založené na konštantnom počte simplexov v trianguláciách. Tento počet je nezávislý od spôsobu triangulácie vrcholov, vzťahy medzi počtami simplexov (vrcholy, hrany, trojuholníky) sú uvedené vo vzťahu (2.4). Vo vyšších dimenziách takéto tvrdenie neplatí, počet $k-$simplexov ($k = 1, \ldots, n$) sa mení v závislosti od voľby triangulácie. Nie je možné priradiť cenovú funkciu k jednotlivým $k-$simplexom, tak aby sa dalo korektne minimalizovať súčet týchto cenových funkcií. Je potrebné nájsť také riešenie, ktoré je nezávislé od topologických transformácií, pri ktorých sa zmenia počty $k-$simplexov.

Vrcholovo založené DDT vyhovujú stanoveným cieľom, lebo počet vrcholov ($0-$simplexov) nezávisí od spôsobu triangulácie. Návrh cenových funkcií v 2D pre ohodnotenie vrcholov nie je ľahký a intuitívny. Vo vyšších dimenziách táto úloha sa stáva ešte komplikovanejšou a zatiaľ neexistuje jej riešenie. Spôsob optimalizácie triangulácie v tomto prípade je tiež otvorenou otázkou, preto je potrebné hľadať iné možnosti.

Z vlastností triangulácií vyplýva, že $\boldsymbol{CH(V)}$ sa nezmení topologickými transformáciami ani vo vyšších dimenziách. Teda objem $\boldsymbol{CH(V)}$ zostáva nezmenený, čo vedie k myšlienke váhovať objem $n-$simplexov. Na základe ich vlastností, ako vystihujú významné črty reprezentované v dátach, by mala byť priradená váhová funkcia. Nižšia váha pritom bude znamenať lepšie správanie. To znamená, že úloha $n-$dimenzionálnej DDT sa dá napísať ako minimalizáciu váhovaného objemu triangulácie:

$$c(\boldsymbol{T(V)}_{optimal}) = \min_{\boldsymbol{T(V)}\in\Omega} \left(\sum_{\sigma_n \in \boldsymbol{T(V)}} (V(\sigma_n) \cdot w(\sigma_n)) \right), \tag{3.1}$$

kde σ_n je $n-$simplex, $V(\sigma_n)$ je objem $n-$simplexu a k nemu priradená váha je označená ako $w(\sigma_n)$.

Uvedená formulácia je ekvivalentná problému MWT pre rovinný prípad. Minimalizácia predstaveným spôsobom je nezávislá od spôsobu triangulácie množiny $\boldsymbol{V}$. Napriek tomu uvedená myšlienka je použiteľná len v prípade, ak sa podarí nájsť vhodné váhové funkcie. Ich hľadaniu je venovaná nasledujúca časť práce.

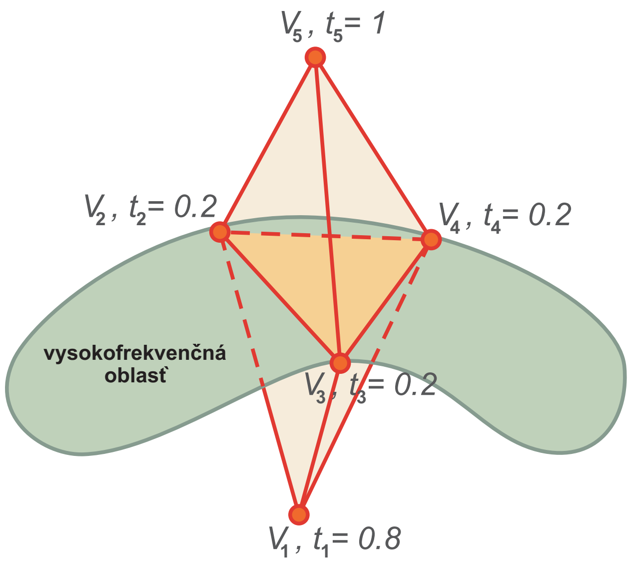
3.4.2 Váhové funkcie

Určenie vhodných váhových funkcií je závislé od voľby aplikačnej oblasti. Vo všeobecnosti ale sa dá preferovať riešenie, ktoré rekonštruuje vysokofrekvenčné oblasti, dbá na kvalitnú rekonštrukciu častí medzi rôznymi oblasťami v dátach.

Jeden z takýchto prístupov je založený na variancii dát. Snaha zachovať oblasti s nízkou varianciou znamená korektnú rekonštrukciu vysokofrekvenčných oblastí. Pre $n-$rozmerné triangulácie to znamená generovanie hyperstien ($(n-1)-$simplexov) s nízkou varianciou. Teda váhová funkcia je založená na variancii funkčných hodnôt z vrcholov $n-$simplexu:

$$w(\sigma_n) = Variance(t_{n_1}, t_{n_2}, \ldots, t_{n_{n+1}})\,, \tag{3.2}$$

kde $w(\sigma_n)$ je váhová funkcia pre $n-$simplex σ_n a t_{n_i} $(i = 1, \ldots, n+1)$ sú funkčné hodnoty priradené k vrcholovom $n-$simplexu. Situácia pre trojrozmerný prípad je ilustrovaná na Obrázku 3.7. Pre vrcholy V_1, V_2, V_3, V_4, V_5 existujú dva spôsoby triangulácie (možné konfigurácie sú zobrazené na Obrázku 2.4). Voľba načrtnutá na Obrázku 3.7 produkuje nižší váhovaný objem ako druhá možná konfigurácia, a lepšie vystihuje črty reprezentované v dátach.



Obr. 3.7: Príklad korektnej tetrahedralizácie vzhľadom na varianciu stien, hodnoty $t_i, i = 1, \ldots, 5$ označujú funkčné hodnoty v daných vrcholoch.

Ďalšou možnosťou je použitie existujúcich cenových funkcií z rovinných prístupov. Cenové funkcie popísané v odseku 2.1.3 sú založené na vlastnosti, že každá hrana (ktorá neleží na ***CH(V)***) je obsiahnutá práve v dvoch trojuholníkoch. Vo vyšších dimenziách platí podobné tvrdenie. Každá hyperstena ($(n-1)-$ stena), neležiaca na ***CH(V)*** je obsiahnutá práve v dvoch $n-$simplexoch, a teda existuje štruktúra $n+2$ vrcholov (vytvorená z dvoch $n-$simplexov), ktorá je analogická k rovinnému prípadu. V trojrozmernom prípade to znamená, že pre každú stenu existujú dva tetrahedrony, a z nich vytvorená štruktúra je bipyramída. Pre tieto hyperstena môžu byť existujúce rovinné cenové funkcie jednoducho zovšeobecnené do vyšších dimenzií. Cieľom je ale určenie váhovej funkcie pre $n-$simplex, pričom k dispozícii sú len informácie o hyperstenách. Vo všeobecnosti platí, že každý $n-$simplex obsahuje práve $n+1$ hyperstien ($(n-1)-$simplexov). Ak za váhovú funkciu je považovaný priemer zovšeobecnených cenových funkcií určených pre $n+1$ hyperstien, tak výsledkom je váhová funkcia s hľadanou vlastnosťou. Ako konkrétny príklad sa uvedie zovšeobecnenie cenovej funkcie *SCF*, ktorá je popísaná pre rovinný prípad v odseku 2.1.3, bude to označené ako *GSCF*. Pre $n-$dimenzionálny prípad *GSCF* meria uhol medzi gradientmi nadrovín, ktoré obsahujú $n-$simplexy zdieľajúce spoločnú $(n-1)-$stenu v $(n+1)-$dimenzionálnom priestore (obor hodnôt a funkčná hodnota) násobené veľkosťami gradientov:

$$c^{GSCF}(\sigma_{n-1}) = \|\nabla P_1\| \cdot \|\nabla P_2\| \cdot (1-\cos(\alpha)) = \|\nabla P_1\| \cdot \|\nabla P_2\| - \nabla P_1 \cdot \nabla P_2\,, \quad (3.3)$$

kde $\nabla P_1, \nabla P_2$ sú gradienty nadrovín v $(n+1)$ rozmernom priestore obsahujúce dané $n-$simplexy, $\|\nabla P_1\|$ a $\|\nabla P_2\|$ sú veľkosti gradientov, a α je uhol medzi nimi. Rovnice hyperrovín $P_1(x)$ a $P_2(x)$ sú:

$$P_i(x) = a_{i,1}x_1 + a_{i,2}x_2 + \ldots + a_{i,n}x_n + a_{i,n+1}\,, \quad i = \{1,2\}\,. \quad (3.4)$$

Gradienty hyperrovín sú dané nasledujúcim vzťahom:

$$\nabla P_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,n})\,, \quad i = \{1,2\}\,, \quad (3.5)$$

a veľkosti gradientov sú:

$$\|\nabla P_i\| = \sqrt{a_{i,1}^2 + a_{i,2}^2 + \ldots + a_{i,n}^2}\,, \quad i = \{1,2\}\,. \quad (3.6)$$

Pre daný $n-$simplex váhová funkcia, ktorá používa zovšeobecnené cenové funkcie, vyzerá nasledovne:

$$w(\sigma_n) = \frac{\sum_{\sigma_{n-1}\in\sigma_n} c^{GSCF}(\sigma_{n-1})}{n+1}\,. \quad (3.7)$$

Aj ostatné uvedené cenové funkcie môžu byť zovšeobecnené podobným spôsobom ako je tu uvedené v prípade *SCF*. Geometrické vlastnosti by sa dali tiež využiť pri tvorbe váhových funkcií. Napríklad tvar hyperstien môže byť použitý ako faktor pri priemerovaní vo vzťahu 3.7.

3.4.3 Optimalizačný proces

Po definovaní matematického modelu *DDT* a váhových funkcií pre viacrozmerné prípady zostáva určenie a popis optimalizačného procesu, ktorý bude vytvárať trianguľácie s požadovanou vlastnosťou. Voľba padla na zovšeobecnenie *LOP* do vyšších dimenzií, táto metóda bude označená ako *GLOP* (*Generalized Lawson's optimization procedure*), teda zovšeobecnený Lawsonov optimalizačný proces.

V rovinnom prípade *LOP* používa topologickú transformáciu preklápania hrán, vo vyšších dimenziách namiesto toho bude použitá operácia bistellárneho preklápania (popísané v odseku 2.1.1). Vzhľadom na to, že vo vyšších dimenziách nie je dokázaná súvislosť grafu triangulácií s použitím bistellárnych preklápaní, je potrebné zavedenie pojmu *množina topologických transformácií* ozn. $\mathbf{T}_K$. Táto množina bude obsahovať topologické transformácie, vzhľadom na ktoré bude vykonaná optimalizácia. Znamená to všeobecnejší popis. $\mathbf{T}_K$ môže byť zvolená ľubovoľne[1], konvergencia algoritmu bude zaručená, nezávislá od voľby tejto množiny. V každom kroku sa zníži váhovaný objem, pričom existuje triangulácia s minimálnym váhovaným objemom. Uprednostňuje sa voľba takej množiny $\mathbf{T}_K$, pre ktorú graf triangulácie je súvislý a jednotlivé topologické transformácie majú za úlohu odstrániť špecifický $k-$simplex, kde $k = 1, \ldots, n-1$, a zmenia topológiu $n-$simplexov obsahujúcich odstránený $k-$simplex. Voľba bistellárnych preklápaní sa zdá byť vhodnou, aj napriek nedokázanej súvislosti grafu triangulácie. Vzhľadom na to, že pojem lokálnej optimality bol definovaný pre hrany v rovine, je potrebné predefinovať daný pojem:

Definícia 3.4.1. *Nech je daná triangulácia* $\boldsymbol{T(V)}$ *pre* $n-$*dimenzionálne rozptýlené dáta,* $k-$*simplex z* $\boldsymbol{T(V)}$, $k = 1, \ldots, n-1$, *sa nazýva* lokálne optimálna *vzhľadom na danú* $\mathbf{T}_K$ *a cenovú funkciu, ak platí jedno z nasledujúcich tvrdení:*

(i) daný $k-$*simplex nemôže byť odstránený z triangulácie pomocou topologickej transformácie z množiny* $\mathbf{T}_K$.

(ii) existuje topologická transformácia z množiny $\mathbf{T}_K$*, ktorá odstráni daný* $k-$*simplex (daný* $k-$*simplex je* preklápateľný*), ale cena triangulácie sa nezníži po aplikovaný tejto transformácie.*

$n-$*dimenzionálna triangulácia sa nazýva lokálne optimálna ozn.* LOT*, ak aplikácia topologickej transformácie z* $\mathbf{T}_K$ *nezníži jej váhovaný objem.*

Základný princíp *GLOP* prístupu je identický so spôsobom fungovania techniky *LOP*. Takisto vyžaduje úvodnú trianguláciu, za ktorú je možné zvoliť *DT* kvôli vhodným vlastnostiam. Cez iteračné kroky sa odstraňujú $k-$simplexy z triangulácie ($k = 1, \ldots, n-1$), ktoré nie sú lokálne optimálne, pokiaľ sa nedosiahne

[1] $\mathbf{T}_K$ môže obsahovať také topologické transformácie, ktoré zmenia trianguláciu množiny $\boldsymbol{V}$ na inú trianguláciu $\boldsymbol{V}$. Počet vrcholov sa nemôže zmeniť, pridanie alebo odstránenie vrcholov je zakázané.

lokálne optimálna triangulácia. Podobne ako pri rovinnom prípade je možné urýchliť algoritmus pomocou zoznamu $k-$simplexov, ktorých lokálna optimalita sa mohla zmeniť v priebehu optimalizácie. Takto nie je potrebné vyšetrovať lokálnu optimalitu všetkých $k-$simplexov v každom iteračnom kroku. Pseudokód algoritmu je znázornený na Obrázku 3.8.

Zovšeobecnený Lawsonov optimalizačný proces - GLOP
Vstup: rozptýlené dáta
Výstup: *LOT* tetrahedralizácia

```
{
 vytvor počiatočnú tetrahedralizáciu T(V);
 Cost = c(T(V));
 oldCost = Cost + 1;
 List_active = {∀σ_k ∈ T(V), k = 1,...,n − 1};
 List_candidate = ∅;
 while(Cost < oldCost){
   for(pre všetky členy List_active){
     if(ak nie je lokálne optimálna){
       použi transformáciu z T_K;
       odstráň z List_active;
       for(všetky σ_k, ktorých lokálna optimalita mohla zmeniť){
         if(σ_k nepatrí do List_active a List_candidate)
           pridaj σ_k do List_candidate;
       }
     }
     else
       odstráň σ_k z List_active;
   }
   oldCost = Cost;
   Cost = c(T(V));
   List_active = List_candidate;
   List_candidate = ∅;
 }
}
```

$List_{active}$ - zoznam $k-$simplexov, ktoré sú testované v danom iteračnom kroku
$List_{candidate}$ - zoznam $k-$simplexov, ktorých lokálna optimalita sa mohla zmeniť

Obr. 3.8: Zovšeobecnený Lawsonov optimalizačný proces - pseudokód.

Popísaný algoritmus konverguje k lokálnemu optimu, lebo v každom iteračnom kroku sa zníži váhovaný objem a počet možných triangulácií je konečný. Iná voľba $\mathbf{T}_K$ môže urýchliť konvergenciu algoritmu a zvýšiť mieru aproximácie *MWT*. Pre trojrozmerný prípad vhodným rozšírením bistellárnych preklápaní môže byť špeciálna transformácia odstránenie hrany (nazývané ako *edge-removal*). Slúži na odstránenie hrany z tetrahedralizácie, nezávisle od počtu tetrahedronov, ktoré ju obsahujú (na rozdiel od $3-2$ preklápania). Môže sa skladať zo série bistellárnych preklápaní. Popis tejto topologickej transformácie, ako aj návod efektívnej optimalizácie, je uvedený v publikácii [She02].

Namiesto *GLOP* je možné použiť aj iné optimalizačné techniky pre *DDT*, ako napríklad simulované žíhanie. V prípade genetickej optimalizácie treba predefinovať operáciu "*prekríženia*". Dalo by sa uvažovať aj o zovšeobecnení look-ahead techniky, táto úloha ale je komplikovanejšou ako v prípade prístupov simulovaného žíhania alebo genetickej optimalizácie.

3.5 Kompresia viacrozmerných dát pomocou *DDT*

Zvládnutie enormného nárastu objemu dát je bežným problémom v rôznych aplikačných oblastiach v súčasnosti [Pol00]. Častokrát tempo vývoja hardveru zaostáva za týmto nárastom a metódy, ktoré slúžia na prácu s dátami, algoritmicky už dosiahli svoju dolnú hranicu. Tento problém týka sa aj výpočtovej geometrie a aplikácie spájané s optimálnymi trianguláciami nie sú výnimkou. Ako príklad je možné uviesť tetrahedralizácie. Pri práci s tetrahedrálnymi sieťami je potrebné pracovať s topológiu triangulácie, čo vyžaduje veľa pamäte a diskového priestoru v prípade skladovania výsledkov. Ďalšou výzvou sú vizualizácia siete alebo rôzne simulácie na základe topológie tetrahedralizácie, napríklad *FEM*. Tieto ťažkosti slúžia ako motivácia k ďalšej úlohe, ktorou je potreba reprezentácie triangulácií s menším počtom trojuholníkov (simplexov) a pritom zachovávanie významných čŕt v dátach. Z hľadiska triangulácií sa jedná o zjednodušenie triangulačnej siete, a z pohľadu dát je to možné vnímať ako stratovú kompresiu. Nasledujúci návrh rieši tento problém pomocou použitia dátovo závislého prístupu.

Riešenie tejto úlohy je úzko spojené s výsledkami z predošlej časti, kde bola predstavená viacrozmerná dátovo závislá triangulácia, príslušná cenová funkcia a optimalizačná technika. Takýto prístup zabezpečuje zostrojenie triangulácie, zachovávajúcej významné príznaky. V kombinácii s technikou zjednodušenia sietí (*mesh decimation*) je možné dosiahnuť stanovený cieľ. Pomocou postupného odstránenia jednotlivých vrcholov sa dosiahne triangulácia s menším počtom simplexov.

Z hľadiska úspešnosti kľúčovou úlohou celého prístupu je ohodnotenie vrcholov podľa ich dôležitosti. Pre tento účel je možné zovšeobecniť spôsob vrcholovo založeného ohodnotenia pre *DDT* (predstavené v časti 2.2.3) do vyšších dimenzií. Pre $n-$rozmernú trianguláciu to znamená prevedenie váhovaných objemov (informácia, ktorá je k dispozícii) do vrcholov. Cena vrchola je váhovaným priemerom ohodnotenia $n-$simplexov, v ktorých je daný vrchol obsiahnutý. Určenie adekvátneho pomeru váhovania, ako aj možnosť použitia geometrických vlastností, môžu byť predmetom ďalšieho výskumu.

Po odstránení vrcholov z triangulácie treba takto vzniknuté mnohosteny znova triangulovať. V rovinnom prípade táto úloha má vždy riešenie, vo vyšších dimenziách to ale neplatí (napríklad Schönhardtov mnohosten pre trojrozmerný priestor). Znamená to, že pred odstránením vrchola je nutné otestovať či vznikne triangulovateľný mnohosten. Okrem možnosti hľadania triangulácie vzniknutého mnohostena

Návrh kompresie viacrozmerných dát pomocou DDT
Vstup: rozptýlené dáta
Výstup: zjednodušená tetrahedrálna sieť, kompresia

```
{
  vyrátaj lokálne optimálnu DDT trianguláciu;
  ohodnoť vrcholy tetrahedralizácie;
  while(nie je splnená ukončovacia podmienka){
    zober vrchol V s najmenšou priradenou hodnotou ;
    if(ak odstránením V vznikne mnohosten, ktorý sa dá tetrahedralizovať){
      trianguluj okolie odstráneného vrchola:
      dosiahni lokálne optimálny stav;   //opcionálne
      aktualizuj ohodnotenie vrcholov;
    }
    else
      zober nasledujúci prvok s najmenšou priradenou hodnotou;
  }
}
```

Obr. 3.9: Stratová kompresia dát, zachovávajúca významné príznaky - náčrt algoritmu.

je tu možnosť použitia topologickej operácie zlúčenie hrany. Odstránenie daného vrchola je možné vnímať ako zlúčenie $1-$simplexu (hrany) do jedného z konečných vrcholov hrán, s ktorými je odstránený vrchol incidentný. Táto operácia sa nazýva aj ako *half edge collapse*. Viac informácií o efektívnom použití zlúčenia hrany na zjednodušenie sietí sa nachádza v práci [KE00]. V prípade, že daný vrchol sa nedá odstrániť, tak sa vyberie ďalší vrchol zo zoznamu.

Vzniknutá triangulácia po odstránení vrcholov nemusí byť lokálne optimálna. Vo všeobecnosti obnova lokálnej optimality v zmysle definície 3.4.1 je výpočtovo drahá operácia, v tomto prípade sa dajú zohľadniť nasledujúce okolnosti:

(i) Pôvodná triangulácia je lokálne optimálna. Po odstránení vrchola stačí vyšetriť lokálnu optimalitu takto vzniknutých nových simplexov a ich susedných simplexov, pričom toto závisí od voľby cenovej funkcie. Každou novou topologickou operáciou (bistelárnym preklápaním), ktorá sa aplikuje na trianguláciu, sa ale rozširuje množina simplexov, ktorých lokálna optimalita sa mohla zmeniť. Inými slovami, môže to spustiť reťazovú reakciu, výsledkom čoho je dátovo závislá optimalizácia veľkej časti triangulácie, čo je časovo náročné. Spôsob voľby odstráneného vrchola ale zaručí, že daný vrchol bol obsiahnutý v $n-$simplexoch s nízkymi váhovanými objemami, t.j. leží v nízkofrekvenčnej oblasti. Znamená to, že triangulácia, ktorá vznikne zlúčením hrany, bude s veľkou pravdepodobnosťou blízko k lokálne optimálnemu riešeniu, teda netreba aplikovať veľa zmien.

(ii) Druhá možnosť takisto zohľadňuje výber vrcholov, ktoré sa majú odstrániť. Ako to už bolo vyššie spomenuté, tieto vrcholy ležia v nízkofrekvenčných oblastiach. Optimalizácia triangulácie, ktorá vznikne po odstránení vrcholov, nie je nutne potrebná v tejto fáze. Stačí vyčísliť informáciu, ktorá je potrebná na ohodnotenie vrcholov, ktorých cena sa mohla zmeniť. Obnova lokálnej optimality sa môže vykonať po odstránení určitého počtu vrcholov.

Počet vrcholov alebo obmedzenie, týkajúce sa ceny odstránených vrcholov, môže byť vhodnou ukončovacou podmienkou pre algoritmus. Ďalšou možnosťou je meranie kvality, napríklad pomocou rôznych metrík, ktoré budú predstavené neskôr v kapitole 4.

Oproti metódam, ktoré používajú postupné zjemňovanie sietí (napríklad prístupy [MDM04, RN00]) má uvedený návrh určité nevýhody. Jeho časová náročnosť je vyššia, lebo sa snaží dosiahnuť výsledok tak, že zostrojí optimálne riešenie pre daný vstup a postupne odstraňuje vrcholy. Je to výpočtovo oveľa náročnejšie ako práca s jednoduchými trianguláciami v prípade metód, ktoré sú založené na postupnom zjemňovaní. Na druhej strane výhodou je spôsob kompresie (odstránenie vrcholov) a následne hľadanie optimálnej triangulácie. Uvedený dátovo závislý návrh poskytuje sofistikovaný spôsob riešenia týchto problémov.

3.6 Paralelný výpočet *DDT* pomocou grafického hardvéru

Ďalšou oblasťou bolo skúmanie možnosti paralelného výpočtu dátovo závislých triangulácií. Motiváciou je smerovanie vývoja hardveru. V posledných rokoch najväčší výpočtový výkon poskytujú grafické karty, ozn. *GPU* (*graphics processing unit*), a nie procesorové jednotky, ozn. *CPU*(*central processing unit*). Architektúra *GPU* je založená na veľkom počte paralelných procesorových jednotiek (rádovo desiatky - stovky, a ich počet neustále rastie), ktoré majú odlišnú stavbu ako *CPU*. Jedná sa o vysoko špecializované procesorové jednotky, ktoré podporujú špeciálne inštrukčné sady a sú optimalizované na efektívne zobrazovanie grafickej informácie. Evolúcia *GPU*[2] časom priniesla príležitosť využitia ich výkonu aj na iné účely než pôvodne boli určené, táto výskumná oblasť sa označuje *GPGPU* (*General-Purpose computation on Graphics Processing Units*) [LHK*04]. Jedná sa o pomerne novú oblasť, ktorá ale rastie pomerne dynamicky.

Úlohou v tomto prípade je hľadanie paralelného dátovo závislého prístupu, ktorú je možné implementovať pomocou *GPGPU* prístupu. Využitie sily *GPU* vyžaduje určité obmedzenia v implementácii, hlavne sa to týka práce s pamäťou.

Poznámka. Pre korektnosť je potrebné uviesť, že táto časť práce sa zrodila pred 3 rokmi (2007), kedy situácia okolo architektúry *GPU* bola odlišná ako v súčas-

[2]Príchod programovateľných jednotiek, tzv. *shader units*, bol tým významným krokom.

nosti. Vtedy programovateľné časti grafických kariet tvorili *vertex-shader* a *pixel-shader*, z hľadiska *GPGPU* výpočtov to znamená značné obmedzenia. Odvtedy evolúcia architektúry grafických kariet a vznik špeciálnych ovládačov ako napríklad *CUDA* [NVi] od NVidie značne uľahčili tvorbu *GPGPU* prístupov. Celkové smerovanie vývoja v tejto oblasti naznačuje vznik štandardu *OpenCL* [Khr], ktorý zjednocuje spôsob práce s *GPU* a *CPU* a je zameraný na efektívne vykonanie paralelných výpočtov.

Podľa dostupných poznatkov sa paralelizáciou dátovo závislých triangulácií nezaoberala žiadna výskumná skupina doteraz (či už na *CPU* alebo *GPU*). Z oblasti optimálnych triangulácií sú známe prístupy, ktoré generujú Voronoiove diagramy (duálna štruktúra k *DT*) v diskrétnom priestore [HCK*99, FG06, RT06] alebo Delaunayovu trianguláciu [RTCS08] (z Voronoiovho diagramu) na *GPU*. Tieto metódy sú založené na proximite vo Voronovoivých diagramoch a pracujú v diskrétnom priestore, preto nie sú použiteľné pre tvorbu *DDT*.

Snahou nasledujúcej časti je zostrojenie paralelnej verzie Lawsonovho optimalizačného procesu, ktorý bol predstavený v odseku 2.2.1.

3.6.1 Návrh paralelného prístupu

Paralelný výpočet optimálnej triangulácie znamená súčasné vykonávanie viacerých optimalizačných krokov. V prípade *LOP* sa jedná o súčasné preklápanie hrán, v závislosti od ich lokálnej optimality. Táto operácia bez obmedzení ale môže narušiť topológiu, a vytvárať neplatnú trianguláciu. Preto pre korektný popis navrhovaného algoritmu je potrebné zaviesť ďalšie pojmy.

Definícia 3.6.1. *Nech je daná triangulácia $\boldsymbol{T}(\boldsymbol{V})$ množiny $\boldsymbol{V}$. Pod pojmom* susedia hrany $e \in \boldsymbol{T}(\boldsymbol{V})$ *sa rozumie množina takých hrán z $\boldsymbol{T}(\boldsymbol{V})$, ktoré vytvárajú s hranou e trojuholník patriace do $\boldsymbol{T}(\boldsymbol{V})$.*

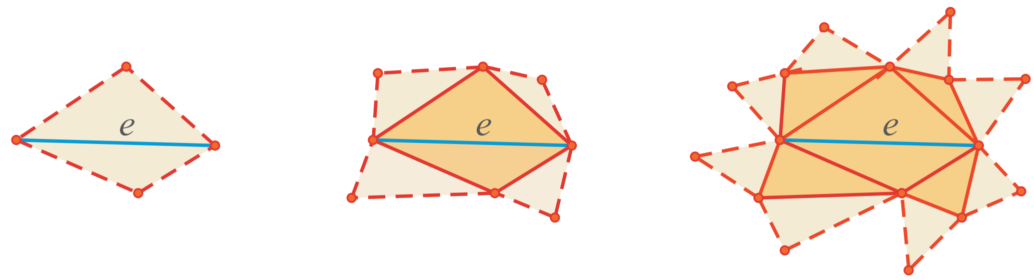
Definícia 3.6.2. Okolie hrany e stupňa-1 *označuje množinu susedov hrany e (ozn.* 1-okolie*). Na základe tohto označenia sa zadefinuje* okolie n-tého stupňa hrany e *pre $n \in \mathbb{N}$ a $n > 1$, ako množina pozostávajúca z:*

(i) množiny hrán z $(n-1)$-okolia hrany e,

(ii) množiny susedných hrán k hranám z $(n-1)$-okolia hrany e.

Pre danú hranu $e \in \boldsymbol{T}(\boldsymbol{V})$ okolia rôzneho stupňa sú znázornené na Obrázku 3.10.

Lokálna optimalita hrany e v dátovo závislých trianguláciách je ovplyvnená výberom cenovej funkcie. V prípade *SCF* jej lokálna optimalita je závislá od cenových funkcií z piatich hrán (hrana e a jej susedia). Znamená to, že preklápanie nejakej hrany z 2-okolia hrany e môže zmeniť jej lokálnu optimalitu. Vo všeobecnosti táto oblasť bude pomenovaná ako *oblasť vplyvu* ozn. *ROI* (*region of influence*)



Obr. 3.10: Ilustrácia 1−okolia (na ľavo), 2−okolia (v strede) a 3−okolia (na pravo) hrany e.

hrany e, a teda bude značiť množinu takých hrán, ktorých zmena (preklápanie) môže spôsobiť zmenu lokálnej optimality e.

V prípade paralelizácie *LOP* je potrebné si nájsť množinu takých hrán, ktoré je možné preklápať súčasne. *GPU* je vytvorený z veľkého počtu paralelných procesorových jednotiek, preto je dôležité hľadať také riešenie, ktoré vie plne využiť túto architektúru, aj s danými obmedzeniami. Modifikovaná Lawsonová metóda bude členená do troch častí:

(i) *Vytváranie kandidátov* - zistenie, ktoré hrany nie sú lokálne optimálne,

(ii) *Akceptovanie, zamietnutie kandidátov* - výber súčasne preklápateľných (akceptovaných) hrán,

(iii) *Paralelné preklápanie hrán* - samotné paralelné preklápanie akceptovaných hrán.

Tieto tri časti tvoria jednu iteráciu v ponímaní klasického *LOP* prístupu. Popisu jednotlivých častí budú venované nasledujúce odseky.

Vytváranie kandidátov

Tento krok zahŕňa vyčíslenie lokálnej optimality všetkých hrán (či ležia v striktne konvexných štvoruholníkoch a výpočet cenových funkcií). Na základe týchto informácií je možné rozdeliť hrany do dvoch skupín:

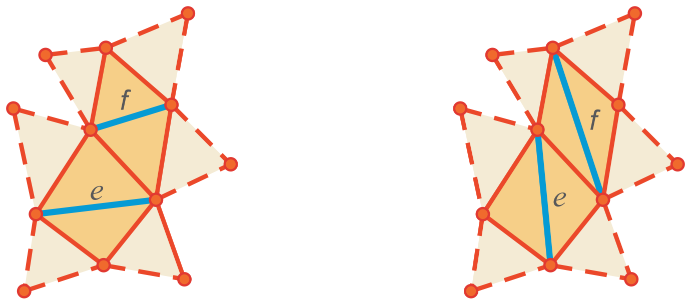
(i) Množina hrán, ktoré nie sú lokálne optimálne, t.j. ich preklápaním je možné získať trianguláciu s nižšou váhou. Označenie týchto hrán bude *kandidáti.*

(ii) Množina lokálne optimálnych hrán. Tieto hrany v danej iterácii už nebudú ďalej spracované.

Úlohou prvej časti je vytvorenie množiny kandidátov. Výberu súčasne preklápateľných hrán z tejto množiny je venovaný ďalší odsek textu. Všetky výpočty, ktoré boli vyššie uvedené, sa dajú paralelne vykonať bez obmedzení. Výpočtovo najdrahšou operáciou je určenie lokálnej optimality na základe cenových funkcií. Je potrebné totiž vyčísliť, aký prínos by malo preklápanie danej hrany na cenu triangulácie. V prípade použitia *SCF* to znamená prácu s 2−okolím testovanej hrany.

Akceptovanie, zamietnutie kandidátov

Lokálna optimalita danej hrany je závislá od preklápaní hrán v jej *ROI*. Pri sekvenčnom spracovaní to nespôsobuje problémy, ale v prípade paralelného spracovania môže nastať nasledujúca konfigurácia. Nech je daná triangulácia a v nej dve hrany e, f, pričom nech f leží v *ROI* hrany e (z čoho vyplýva, že aj e leží v *ROI* hrany f). Súčasné spracovanie hrán spôsobuje, že preklápanie jednej hrany môže zmeniť lokálnu optimalitu tej druhej, pričom tá druhá hrana je preklápaná na základe tých nezmenených údajov. Situácia je znázornená na Obrázku 3.11. Riešením je výber takých hrán, ktorých *ROI* sa nepretínajú. Tým je zabezpečené, že sa neporuší štruktúra triangulácie a konvergencia k lokálnemu optimu.



Obr. 3.11: Hrany e a f pred (vľavo) a po (vpravo) paralelnom preklápaní hrán. V prípade hrany e, hrana f by nemala byť preklápaná, platí to aj opačne.

Na základe vyššie uvedeného spôsobu je možné hrany rozdeliť do dvoch disjunktných množín:

(i) *zamietnuté hrany* - hrany, ktoré nebudú preklápané,

(ii) *akceptované hrany* - množina hrán, ktoré budú preklápané paralelne.

Pre rozdelenie hrán do vyššie uvedených množín je potrebné zaviesť pomocné premenné, tzv. *identifikačné čísla* hrán (ozn. *ID*). Tieto identifikátory sú priradené ku každej hrane na začiatku optimalizačného procesu tak, aby každá hrana mala jedinečné *ID*. Pri preklápaní danej hrany *ID* zdedí alternatívna diagonála (vzniknutá hrana).

Rozdelenie kandidátskych hrán, teda ich zamietnutie alebo akceptovanie prebieha nasledovne. Nech hrana e je z množiny kandidátov, pre jej klasifikáciu je potrebné spracovať všetky hrany z *ROI* e. Ak existuje aspoň jedna akceptovaná hrana v tejto oblasti, tak e musí byť zamietnutá. Ináč sa porovnávajú hodnoty *ID* všetkých kandidátskych hrán z *ROI* hrany e. V prípade, že e má najmenšiu *ID*, tak sa akceptuje (pridá sa do množiny akceptovaných hrán) a odstráni sa z množiny kandidátov. Zostala ešte konfigurácia, kde e nemá najmenšiu hodnotu *ID* a neexistuje akceptovaná hrana v jej *ROI*. Vtedy testovaná hrana zostane v množine kandidátov a bude akceptovaná alebo zamietnutá v nasledujúcich iteráciách.

Akceptovanie a zamietnutie sa vykoná iteratívne a skončí vtedy, keď množina kandidátov je prázdna. Konvergencia takéhoto prístupu vyplýva z toho, že v každej iterácii sa odstráni minimálne jedna hrana z množiny kandidátskych hrán. Vždy existuje hrana s minimálnou hodnotou *ID*. Pri reálnych dátach sa dá predpokladať oveľa rýchlejšia konvergencia.

Poznámka. Uvedený postup je navrhnutý tak, aby sa výpočty dali zrealizovať na *GPU*. Treba však poznamenať, že by sa dal zostrojiť aj sofistikovanejší a efektívnejší prístup. Uvedená úloha, výber akceptovaných hrán, sa dá vnímať totiž aj ako farbenie grafu, ktorá sa skúma a rieši v teórii grafov. Vzhľadom na obmedzenia týkajúce sa architektúry *GPU* a veľký počet procesorových jednotiek je však možné považovať navrhnutý prístup za adekvátny.

Paralelné preklápanie hrán

V poslednom kroku prebieha súčasné preklápanie hrán z množiny akceptovaných hrán. Vďaka predspracovaniu v predošlom kroku je zabezpečené, že nenastanú konflikty a vytvorená triangulácia bude korektná. Vzhľadom na preklápacie operácie je potrebné aktualizovať dátové štruktúry v okolí zmenených hrán, veľkosť týchto okolí závisí od veľkosti *ROI*.

Súhrn

Zopakovanie uvedených krokov zaručuje vytvorenie lokálne optimálnej triangulácie. Konvergencia k lokálne optimálnemu stavu je dokázateľná podobne ako v prípade *LOP*, cena triangulácie klesá v každom kroku a minimum musí existovať. Pseudokód algoritmu je znázornený na Obrázku 3.12.

Oproti pôvodnému *LOP* je vnútorná iterácia navyše - výber súčasne preklápateľných hrán v časti akceptácia a zamietnutie kandidátov. To je cena za paralelný prístup. Dá sa predpokladať, že pri malom počte paralelných výpočtových jednotiek (napríklad súčasné *CPU* disponujú s 2–4 jadrami) to nie je efektívne. Pri *GPU* sa ale dá spoľahnúť na vyšší počet výpočtových jednotiek. Úlohou testov bude overiť, či si väčší počet procesorov vie poradiť so stratou, ktorá je spôsobená paralelizáciou prístupu.

3.6.2 Náčrt implementácie na *GPU*

V predošlej časti bol predložený návrh paralelnej optimalizácie *DDT* prístupu. Voľba jednotlivých krokov algoritmu úzko súvisí s architektúrou *GPU*, preto je dôležité uviesť náčrt implementácie. Nejedná sa popritom o podrobný návod, ale skôr poslúži na to, aby čitateľ ľahšie pochopil dôvody dizajnu metódy. Samotná implementácia na *GPU* a návrh dátových štruktúr je prácou Dr. Červeňanského, hlavného autora článku [CTS*], kde tieto výsledky boli publikované.

Paralelný Lawsonov optimalizačný proces
Vstup: iniciálna triangulácia *T(V)*
Výstup: *LOT* triangulácia

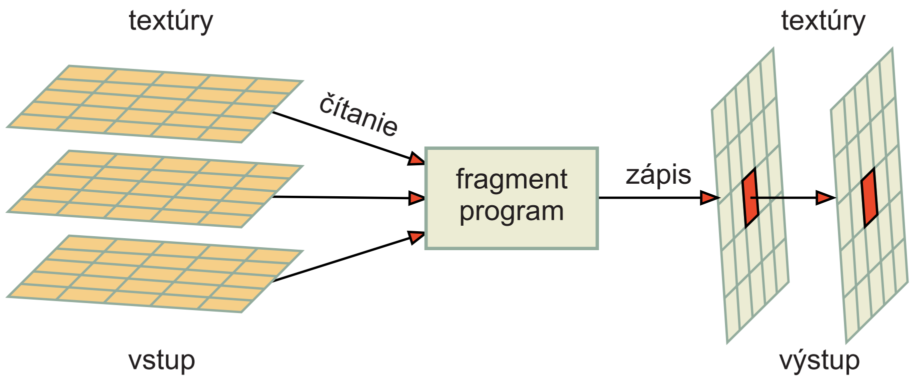
```
{
  while(T(V) nie je lokálne optimálna){
    EdgeList_candidate = ∅;
    EdgeList_accepted = ∅;
// – Krok 1 –
    for(pre ∀ hrany e ∈ T(V))
      if(e nie je lokálne optimálna)
        pridaj e do EdgeList_candidate;
// – Krok 2 –
    while(∃e ∈ EdgeList_candidate){
      if(∃ hrana f ∈ EdgeList_accepted také, že f ∈ ROI hrany e)
        odstráň e z EdgeList_candidate;    // zamietni hranu e
      else{
        if(ID hrany e < min(ID všetkých kandidátskych hrán z ROI hrany e)){
          pridaj e do EdgeList_accepted;    // akceptuj hranu e
          odstráň e z EdgeList_candidate;
        }
        else{
        }    // e zostane kandidátom
      }
    }
// – Krok 3 –
    for(∀e ∈ T(V)){
      if(e ∈ EdgeList_accepted){
        preklop hranu e;
        aktualizuj dátovú štruktúru hrany e;
      }
      else
        if(∃ akceptovaná hrana v 1-okolí hrany e) // sused hrany e
          aktualizuj dátovú štruktúru hrany e;
    }
}
```

$EdgeList_{candidate}$ - zoznam kandidátskych hrán
$EdgeList_{accepted}$ - zoznam akceptovaných hrán

Obr. 3.12: Pseudokód paralelnej verzie Lawsonovej optimalizácie, ktorá je implementovateľná na *GPU*.

V čase, keď vznikla implementácia, väčšina *GPGPU* výpočtov sa vykonávala v takzvaných *fragment programoch* (známy aj ako *fragment shader*). Je to programovateľná časť *GPU*, ktorá je štandardne podporovaná výrobcami grafických kariet. Použitie fragment programu vyžaduje zvolenie takých dátových štruktúr, ktoré sú s ním kompatibilné. Pôvodne slúži na manipuláciu s textúrami, a preto hlavným vstupom do programu sú textúry. Každá textúra sa skladá z jednotiek (*texelov*), kde sú uložené štyri hodnoty, v normálnom prípade farebná informácia (R,G,B kanály) a priehľadnosť. Fragment program má odlišný prístup k textúram, ako je to v prípade *CPU* prístupu do operačnej pamäte. Čítanie je umožnené z viacerých textúr a z ľubovoľného texelu, ale je rozdiel v zápise. Tam je potrebné dopredu určiť jednu pozíciu, kam sa zapíše výsledok fragment programu, pričom zápis môže byť do viacerých textúr, ale len na danú pozíciu - situácia je znázornená na Obrázku 3.13.

Je to signifikantné obmedzenie, ktoré má veľký vplyv na implementáciu a dizajn algoritmu.



Obr. 3.13: Práca fragment programu so vstupnými a výstupnými textúrami.

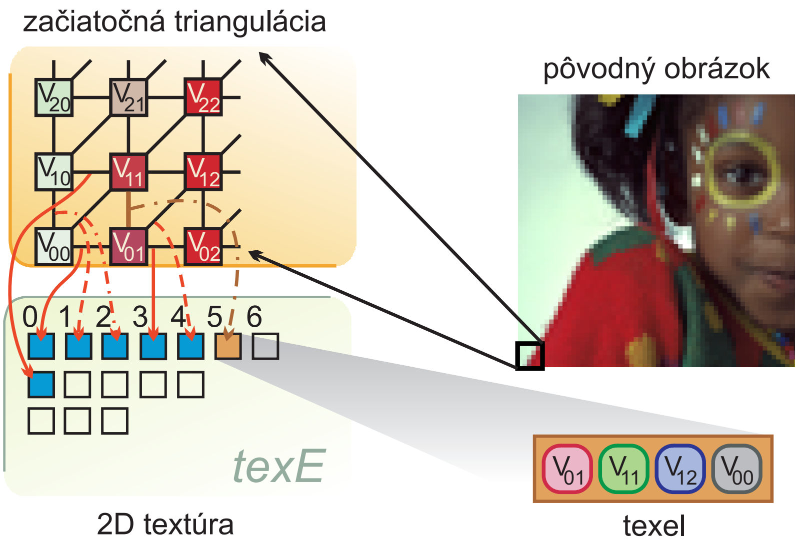
Návrh dátovej štruktúry

Reprezentácia triangulácii pomocou textúr vyžaduje špeciálnu dátovú štruktúru. Voľba vrchola za základnú jednotku a udržiavanie zoznamu incidentných hrán sa zdá byť prirodzenou voľbou. Počet incidentných hrán (teda stupeň vrchola) je ale premenlivý, v závislosti od vykonaných preklápaní. Práca s dynamickými štruktúrami pri *GPGPU* výpočtoch je náročná a preto tento prístup nie je vhodný. Namiesto toho sa zvolí za základnú jednotku dátovej štruktúry hrana triangulácie. Počet hrán je totiž v prípade rovinných triangulácií nemenný.

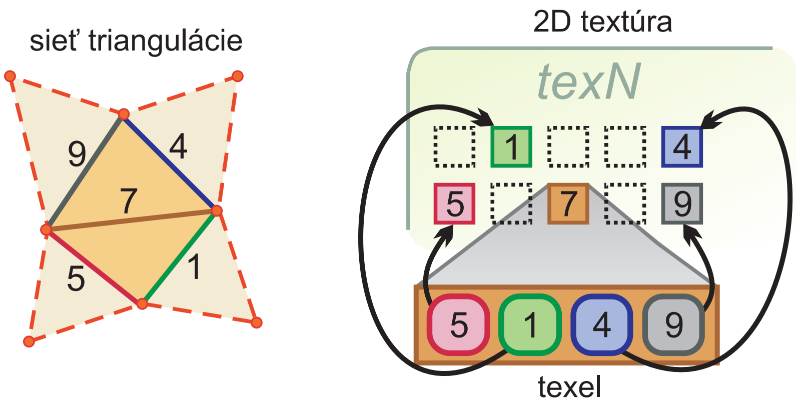
Každý texel bude obsahovať informáciu o vrcholoch, ktoré vytvárajú danú hranu (ozn. ako *hlavné vrcholy*). Na uchovanie topológie triangulácie je potrebné uložiť aj informáciu a ďalších dvoch vrcholoch, ktoré spolu s hlavnými vrcholmi určia trojuholníky, v ktorých daná hrana leží. Tieto vrcholy sa budú nazývať *susedné vrcholy* danej hrany. Teda texel bude obsahovať štyri vrcholové identifikátory, ktoré budú slúžiť na určenie pozícii a farebných údajov jednotlivých vrcholov. Zoznam hrán bude uložený do textúry, ktorá je označená $texE$, celková štruktúra je ilustrovaná na Obrázku 3.14.

Pri vyčíslení lokálnej optimality hrany je potrebné získať aj cenu okolitých hrán (z 1−okolia). Preto je potrebné ukladať do textúry aj ďalšie informácie, kde budú uložené *ID* štyroch susedných hrán. Označenie tejto textúry bude $texN$ a jej ilustrácia sa nachádza na Obrázku 3.15. Na začiatku optimalizačného procesu je táto informácia vygenerovaná na základe počiatočnej triangulácie. Vzhľadom na distribúciu vrcholov pospájaním susedných vrcholov sa dá táto triangulácia vygenerovať veľmi rýchlo. Znázornenie takejto triangulácie je zobrazené na Obrázku 3.14 vľavo hore.

Okrem uvedených textúr $texE$, $texN$ je nutné mať aj ďalšiu textúru, kde sa ukladá informácia o kandidátskych hranách, o akceptácii a zamietnutí hrán. Navrhnutý algoritmus pracuje s touto textúrou tak, že súčasne potrebuje z nej čítať a do nej zapisovať. Toto v prípade *GPU* nie je umožnené, preto je potrebné mať dve identické



Obr. 3.14: Začiatočná triangulácia (vlavo hore). Vytváranie textúry *texE* zo začiatočnej triangulácie (vlavo), a jeden texel z tejto štruktúry (vpravo dole). $V_{i,j}$ označujú vrcholy triangulácie.



Obr. 3.15: Informácia o susedoch hrán, uložená v textúre *texN*. Čísla značia *ID* jednotlivých hrán.

textúry, ozn. $texC_1$ a $texC_2$. Z jednej sa len číta a do druhej sa len zapisuje, pričom na konci iterácii sa informácie v nich synchronizujú. Texely týchto textúr obsahujú *ID* hrany, na ktoré sa vzťahuje informácia a údaje o tom, či je daná hrana kandidátom, akceptovaná alebo zamietnutá.

Náčrt implementácie jednotlivých krokov vyzerá nasledovne.

Vytváranie kandidátov

Prvým krokom je zistenie lokálnej optimality jednotlivých hrán. Znamená to vyčíslenie cenových funkcií a zistenie, či je daná hrana preklápateľná (či leží v striktne konvexnom štvoruholníku). Pre tieto účely je potrebný prístup k textúram *texE* a *texN*, výsledok, či je daná hrana kandidátom, sa zapíše do $texC_1$. V prípade, ak počet kandidátov je nula, tak algoritmus dokonvergoval do lokálne optimálneho stavu riešenia.

Akceptovanie, zamietnutie kandidátov

V druhom kroku sa akceptujú alebo zamietajú kandidáti. Tento iteračný proces striedavo číta a zapisuje z textúr $texC_1$ a $texC_2$, pričom prebieha ich synchronizácia na konci iteračných krokov. Zavedie sa pojem *nespracovaná hrana*, značí to hranu, ktorá ešte nebola označená ako akceptovaná alebo zamietnutá. V jednotlivých iteračných krokoch sa získavajú údaje o nespracovaných hranách, či sú kandidátmi a ich *ID* hodnota. Pre testovanú hranu sú potrebné tieto údaje pre hrany z jej *ROI*. Na základe vyššie opísaného algoritmu sa postupne znižuje počet nespracovaných hrán. Konečný výsledok, informácia o paralelne preklápateľných hranách, sa nachádza v jednej z textúr $texC_1$ a $texC_2$, do ktorej sa v poslednom iteračnom kroku zapisovalo.

Paralelné preklápanie hrán

Preklápanie hrany v uvedenej dátovej štruktúre znamená výmenu hlavných a susedných vrcholov v *texE*. Ďalej je potrebné zmeniť informácie o susedných hranách v *texN*. Situácia je podobná ako v predchádzajúcom kroku, je potrebné naraz čítať a zapisovať do tých istých textúr, dá sa to riešiť kópiami textúr. Okrem toho vstupom je aj informácia o akceptovaných hranách, teda jedna z textúr $texC_1$ a $texC_2$.

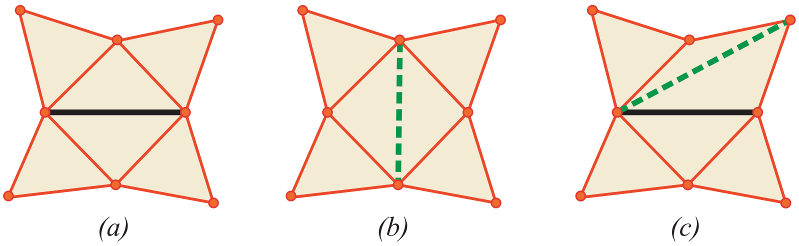
Ako už bolo spomenuté, hlavným obmedzením pri fragment programoch je zapisovanie na pevne vopred stanovené miesto. Pri preklápacích operáciách môžu nastať nasledujúce konfigurácie:

(i) Nech hrana e je zamietnutá, teda sa nepreklápa. Na výstup sa zapíšu pôvodné hodnoty, táto konfigurácia je zobrazená v časti (a) Obrázku 3.16.

(ii) Druhým prípadom je, že hrana e sa preklápa, to znamená, že výmenu hlavných a susedných vrcholov je potrebné zaznamenať do *texE*. Takisto treba zapísať korektné údaje o susedných hranách do *texN*. Konfigurácia je zobrazená na Obrázku 3.16 (b).

(iii) Posledná možnosť zahŕňa konfiguráciu, kde hrana e sa nepreklápa, ale niektorá z jej susedných hrán áno, Obrázok 3.16 (c). Je potrebné prehľadávať susedné hrany e. V prípade, že niektorá z týchto hrán je akceptovaná, tak je potrebné zmeniť informáciu o susedných vrcholoch hrany e v *texE* a zapísať informáciu o nových susedoch hrany e do *texN*.

Poznámka. Popísaná implementácia vyžaduje grafickú kartu s podporou Shader Model verzie 3.0 alebo vyššie. Táto technológia je bežne dostupná vo väčšine starších grafických kariet a je štandardom v prípade nového hardvéru.



Obr. 3.16: Konfigurácie pri paralelnom preklápaní, preklápanie hrany je označené prerušovanou čiarou. (a) - zamietnutá hrana, žiadne preklápanie; (b) - akceptovaná hrana (preklápanie); (c) - preklápanie susednej hrany.

3.6.3 Vylepšenie základného princípu

Výsledky z popísaného paralelného prístupu vykazujú určité nedostatky oproti pôvodnému LOP. Cena dosiahnutej lokálne optimálnej triangulácie je zvyčajne vyššia ako v prípade výsledkov sekvenčnej techniky, teda horšie aproximuje MWT. Ďalej, v prípade rekonštrukcie obrazu produkuje pravidelne sa objavujúce artefakty vo vysokofrekvenčných oblastiach. Takéto rekonštrukčné chyby sú zobrazené na Obrázku 3.17 *(c)*.

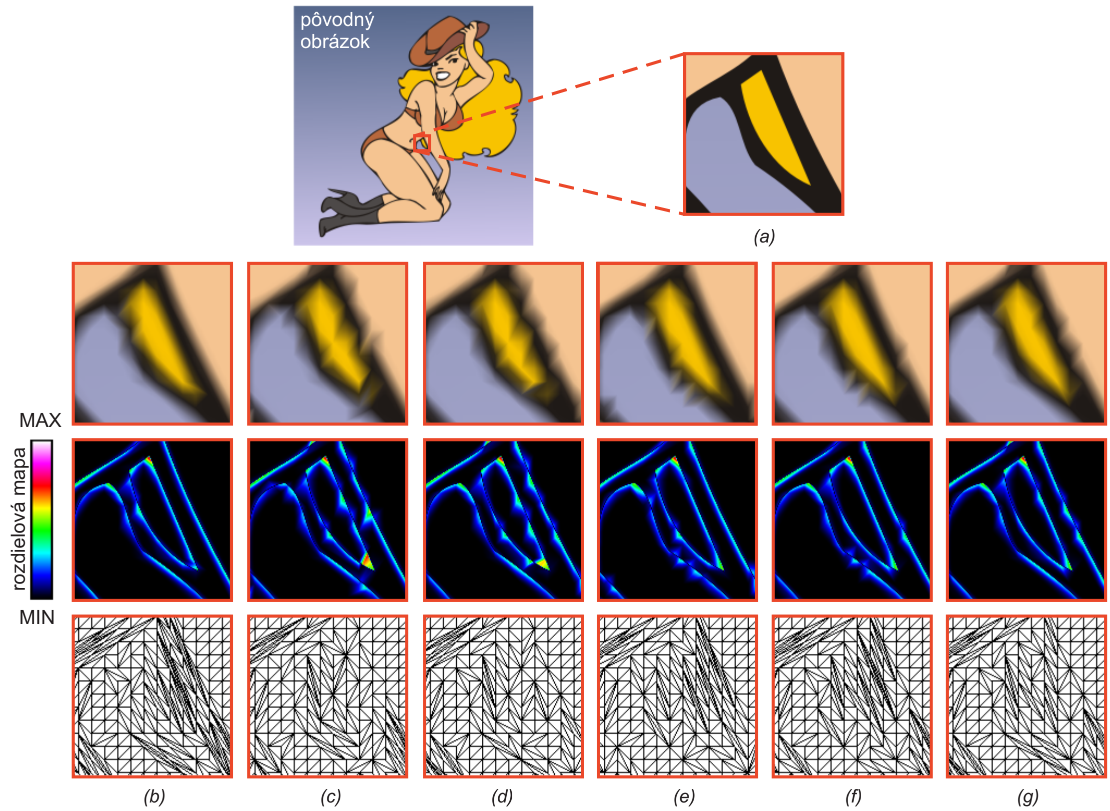
Cieľom nižšie uvedených modifikácií je snaha dosiahnuť lepšiu aproximáciu MWT a eliminovať množstvo rekonštrukčných chýb v oblastiach hrán. Výsledky jednotlivých zlepšení sú znázornené na Obrázku 3.17, kde v strednom riadku sú vyrátané rozdielové obrazy v perceptuálne lineárnom farebnom priestore.

Rozšírenie oblasti *ROI*

V prípade rekonštrukcie obrazu vizuálne artefakty sú spôsobené nesprávne orientovanými trojuholníkmi vo vysokofrekvenčných oblastiach. Tieto trojuholníky sa vyskytujú v pravidelných intervaloch, vzdialené od seba zhruba vo veľkosti *ROI*. Súčasné spracovanie blízko seba ležiacich hrán spôsobuje, že hrany ktoré ležia medzi nimi, sa dostanú do lokálne optimálneho stavu. Tieto hrany spôsobujú vo výslednej triangulácie vznik artefaktov. V prípade sekvenčného spracovania takéto zaseknutie sa hrán nenastane.

V snahe zabrániť vzniku takýchto konfigurácií sa zväčší *ROI* hrán vo fáze akceptovania a zamietnutia, riadky 9–20 pseudokódu na Obrázku 3.12. V prípade použitia cenovej funkcie *SCF* to znamená zväčšenie 2$-$okolia na 3$-$okolie. Táto zmena čiastočne potlačí množstvo vzniknutých artefaktov, a aj cena triangulácie je nižšia ako v prípade pôvodného paralelného prístupu. Označenie tejto modifikácie bude *Exp-ROI*, výsledok tejto metódy je zobrazený na Obrázku 3.17 (e).

Vďaka použitiu väčšieho *ROI* okolia rastie výpočtová náročnosť metódy. Ďalšie pozorovania viedli k zisteniu, že hrany, ktoré spôsobili problémy v pôvodnej verzii algoritmu, boli preklápané hlavne v prvých dvoch iteráciách. Preto je postačujúce použiť rozšírené *ROI* len v prvých iteráciách a potom môže byť použitý základný



Obr. 3.17: Rasterizovaný vektorový obrázok (pôvodný obrázok), a jej 1200% magnifikácia pomocou rôznych techník: (b) *CPU* založený *DDT*; (c) základná *GPU* verzia; (d) *GPU MaxGain*; (e) *GPU ExpROI*; (f) *GPU ExpROI3*; (g) *GPU ExpROI MaxGain*. Rasterizácia vektorového obrázku v zväčšenom rozlíšení (a). Stredný riadok obsahuje rozdielové obrazy, príslušné triangulácie sú zobrazené v dolnom riadku.

prístup. Označenie tejto modifikácie bude *ExpROIN*, kde N značí počet iterácií v ktorej sa používa rozšírený *ROI*. Výsledok z hľadiska vizuálnej kvality je podobný ako výsledok z *ExpROI*, ukážka sa nachádza na Obrázku 3.17 (f).

Maximalizácia prínosu (ozn. *GPU MaxGain*)

Zlepšenie miery aproximácie *MWT* je cieľom ďalšej modifikácie, jej základnou ideou je zmena spôsobu výberu akceptovaných hrán. V pôvodnej verzii pri tomto výbere sa využíva hodnota *ID* kandidátskych hrán. V modifikácii sa namiesto toho zoberie tá kandidátska hrana, ktorej preklápanie má väčší prínos pre cenu triangulácie. Táto informácia je vyrátaná už vo fáze *vytvárania kandidátov*, a v tomto prípade sa to znova využije. V prípade, ak existuje viac hrán s rovnakým prínosom, tak sa postupuje ďalej na základe hodnoty ich *ID*.

Takto modifikovaný algoritmus konverguje rýchlejšie. Môže sa nastať situácia, že vďaka rýchlejšej konvergencii sa prístup zasekne v horšom lokálnom minime ako v prípade techník, kde cena konverguje pomalšie ku globálnemu optimu. Tento jav bol pozorovaný napríklad pri minimalizačných úlohách v kombinácií s metódou simulovaného žíhania [Sch93]. Tento prípad hrozil, ak parametre boli nevhodne nastavené, a konvergencia bola príliš rýchla. Podobná modifikácia ako maximalizácia prínosu bola spomenutá v knihe [HD06], kde bola uplatnená táto myšlienka pre neparalelnú verziu Lawsonovho prístupu.

V ďalších častiach textu táto modifikácia bude označená ako *MaxGain*. Na Obrázku 3.17(d) je zobrazený výsledok vytvorený s uvedenou modifikáciou. Zvyčajne dosahuje nižšiu cenu triangulácie ako dáva pôvodná paralelná technika.

Poslednou modifikáciou je kombinácia uvedených zmien. V súlade s očakávaniami táto kombinácia produkuje trianguláciu s najnižšou váhou a najlepšie potláča vizuálne chyby, jej výsledok je zobrazený na Obrázku 3.17(g).

3.7 Kombinácia *DDT* a konvolučných techník pre účely rekonštrukciu obrazu

V predchádzajúcich úvahách sa najčastejšie spomína aplikačná oblasť rekonštrukcia obrazu. Použitie dátovo závislej techniky pre tento účel je zvyčajne spojené s potrebou zväčšenia obrazu, vzhľadom na kvalitnú rekonštrukciu vysokofrekvenčných oblastí. Táto metóda ako väčšina techník je závislá od kvality vstupu. V reálnom svete digitálne obrazy sú častokrát zašumené alebo ich kvalita je poznačená stratovou kompresiou. Výsledkom toho je neadekvátna kvalita rekonštrukcie v nízkofrekvenčných oblastiach alebo na miestach, kde nie sú jasne orientované hrany v obraze. Inými slovami šum je zosilnený. Pri extrémnom zväčšení na týchto miestach je viditeľná trojuholníková štruktúra, čo môže spôsobiť vizuálne rušivo, príklad takejto rekonštrukcie je zobrazený na Obrázku 3.18. Výsledky konvolučných techník sú viac priaznivé v týchto oblastiach, majú schopnosť potlačiť šum. Oba prístupy majú svoje výhody aj nevýhody. Hlavnou myšlienkou a úlohou je nájdenie takej rekonštrukcie, ktorá sa vo vysokofrekvenčných oblastiach správa ako *DDT* a v nízkofrekvenčných ako konvolučná technika.

Jedna z možností je skombinovanie výsledkov rôznych rekonštrukčných techník pomocou stmeľovacej funkcie. Vyžaduje to vykonanie rôznych operácií, celkový proces je možné deliť na tri fázy:

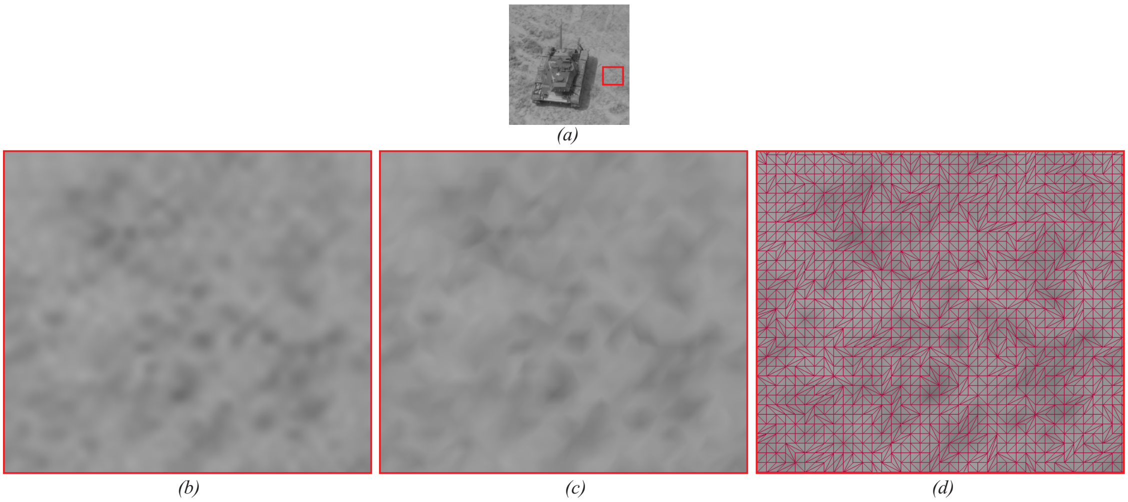
(i) *Predspracovanie* - odhad hranových oblastí po zväčšení obrazu,

(ii) *Rekonštrukcia* - rekonštrukcia obrazu pomocou *DDT* a konvolučnej techniky,

(iii) *Kombinácia výsledkov* - spojenie výsledkov pomocou stmeľovacej funkcie.

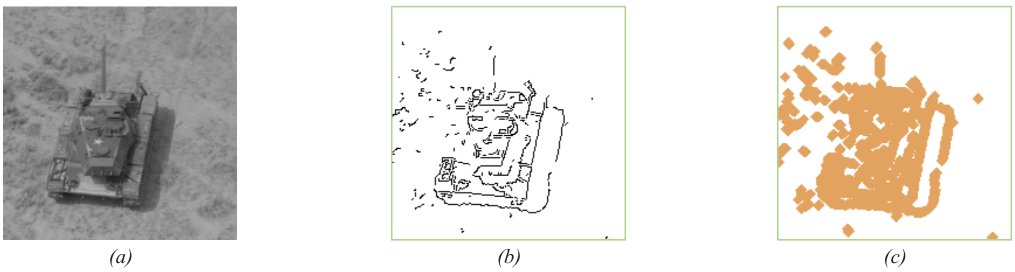
3.7.1 Predspracovanie

Prvou úlohou je identifikovanie vysokofrekvenčných oblastí v obraze. Pre tento účel slúžia hranové detektory, existuje ich veľké množstvo a rôzne druhy. Výsle-



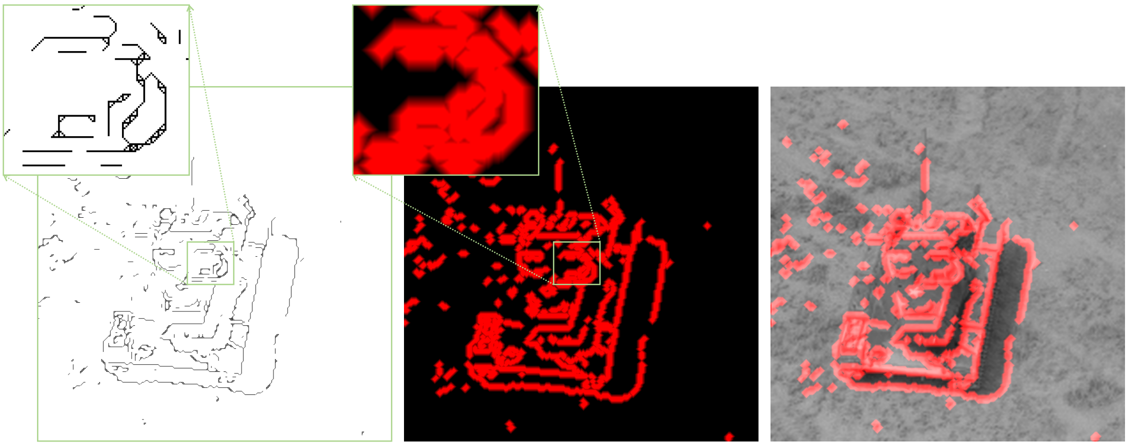
Obr. 3.18: Rekonštrukcia nízkofrekvenčných oblastí pomocou Lanczosovho filtra (b), a dátovo závislej metódy *LOP* (c) pri 1600% zväčšení (spolu s výslednou trianguláciou (d)). Pôvodný obrázok (a).

dok bude použitý pre odhad hranových oblastí vo zväčšenom obrázku. Voľba preto padla na *Cannyho hranový detektor* [GW06], kvôli jeho vhodným vlastnostiam. Tento hranový detektor využíva ako predspracovanie konvolúciu obrazu s *Gaussianom* [GW06], vďaka čomu potláča falošné detekcie spôsobené zašumením. Taktiež spĺňa kritérium jednej odozvy, t.j. hrany sú reprezentované 1 pixel hrubou čiarou. V porovnaní s bežne používanými hranovými detektormi je viac odolný voči nastaveniu parametrov, dá sa ľahšie nastaviť taký parameter, aby kvalita detekcie bola adekvátna pre rôzne typy vstupov a nevyžadovala zásah používateľa. Výpočtová náročnosť je vyššia ako v prípade jednoduchých detektorov, akými sú napríklad *Sobelov* alebo *Robertsov operátor* [GW06], ale vzhľadom na výpočtový čas dátovo závislej triangulácie je toto navýšenie zanedbateľné. Výsledok z Cannyho detektoru je zobrazený na Obrázku 3.19 (b), bude sa to označovať ako Tex_{EDR} (*EDR* - *edge detector result*), pôvodný obrázok je zobrazený v časti (a).



Obr. 3.19: Výsledok hranovej detekcie pomocou Cannyho detektora (b) z obrazu (a). V časti (c) sú vyznačené oblasti, ktoré treba triangulovať.

Kritérium jednej odozvy uľahčuje vytvorenie odhadu hranových oblastí. Na základe faktoru zväčšenia (vstupný údaj) sa zväčší výsledok z hranového detektora. Táto operácia prebieha tak, akoby Tex_{EDR} obsahovala vektorové údaje. Pixle reprezentujúce hranové oblasti sú brané ako vrcholy. Najprv sa vyrátajú nové pozície vrcholov vzhľadom na dané zväčšenie, a spoja sa vrcholy, ktoré boli susedné v Tex_{EDR} (susednosť na základe 8−susednosti jednotlivých pixlov v Tex_{EDR}). Vrcholy, ktoré nemajú žiadneho suseda, sa vynechajú a odstránia, aby v týchto oblastiach bola použitá konvolučná technika. Výsledok takto získaného odhadu hrán je zobrazený na Obrázku 3.20 (a), bude sa to označovať ako Tex_{EE} (*EE - estimated edges*).



Obr. 3.20: Odhad hrán pri 400% magnifikácii (a), na základe toho sa vytvorí vzdialenostná mapa (b). Vizualizované hranové oblasti vo výslednom obrázku (c), kde sú kombinované výsledky z *DDT* a z konvolučnej metódy.

Na kombináciu výsledkov nestačí odhad hrán, ale je potrebné aj určenie hranových oblastí, respektíve segmentácia obrazu na základe od vzdialenosti od odhadnutých hrán. Nazýva sa to vzdialenostná mapa (*distance map*) v počítačovej grafike. Na vytvorenie vzdialenostnej mapy môže byť použitý klasický floodfill algoritmus, využívajúci 4−susednosť. Je potrebné dodať, že takto získaný výsledok je len aproximáciou korektného výsledku, existujú aj sofistikovanejšie algoritmy ako *Danielssonov algoritmus* [Dan80], ktoré vrátia vzdialenosť na základe euklidovskej metriky. Po porovnaní výsledkov z týchto dvoch prístupov (respektíve skúšanie floodfill metódy s 8−susednosťou) sa rozhodlo pre využitie floodfill algoritmu so 4−susednosťou. Vplyv na výsledok celého prístupu nie je znateľný a merateľný, a preto použitie Danielssonovho prístupu s vyššou výpočtovou náročnosťou nie je potrebné. Výsledok je zobrazený na Obrázku 3.20 (b), vzdialenostná mapa sa bude označovať ako Tex_{DM} (*DM - distance map*). Intenzita červenej farby znamená vzdialenosť od odhadnutých hrán, menšia intenzita znamená väčšiu vzdialenosť. Nie je potrebné vybudovanie kompletnej vzdialenostnej mapy, stačí len tá časť, kde sa budú zmiešavať výsledky rekonštrukcií *DDT* a konvolučnej techniky.

3.7.2 Rekonštrukcia

Nasledujúcim krokom je rekonštrukcia obrazu. Výber konvolučnej metódy môže byť závislý od kvality vstupu, pre veľmi zašumené obrázky je vhodný napríklad Gaussian ako konvolučná maska, v inom prípade môže byť zvolený napríklad Lanczosov filter. Výpočet dátovo závislej techniky nie je potrebné aplikovať na celý obraz, stačí len optimalizácia hranových oblastí, ako je to zobrazené na Obrázku 3.19(c). Na základe Tex_{DM} sa dajú tieto oblasti ľahko získať, veľkosť pritom závisí od voľby stmeľovacej funkcie a bude analyzovaná v časti *Kombinácia výsledkov*. Výhodou, okrem toho, že sa takto zmenší počet vrcholov, je aj možná paralelizácia. S veľkou pravdepodobnosťou takto vzniknú od seba oddelené oblasti, ktoré sa dajú nezávisle triangulovať.

3.7.3 Kombinácia výsledkov

Záverečnou fázou je kombinácia výsledkov pomocou stmeľovacej funkcie. Nech zväčšené obrázky majú rozmer x_{max} a y_{max}, a hodnota $f(x, y)$ značí vzdialenosť od hrán na základe Tex_{DM}, $x = 1, \ldots, x_{max}$ a $y = 1, \ldots, y_{max}$. Pomocou *ratio* je značený pomer, o koľko bol daný obraz zväčšený, napríklad v prípade 400% magnifikácie jej hodnota je 4. Predpis stmeľovacej funkcie, ktorá lineárne kombinuje výsledky rekonštrukčných techník, vyzerá nasledovne:

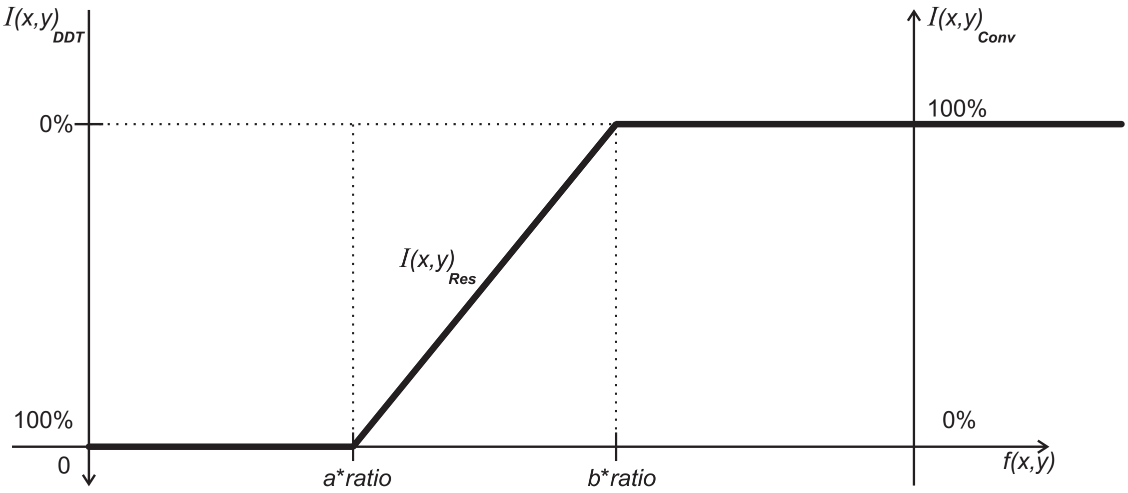
$$I(x,y)_{Res} = \begin{cases} I(x,y)_{DDT}\,, \ ak \ \ f(x,y) < a * ratio \\ (1-t) * I(x,y)_{DDT} + t * I(x,y)_{Conv}, \\ \quad ak \ \ a * ratio \leq f(x,y) \leq b * ratio \\ I(x,y)_{Conv}\,, \ ak \ \ f(x,y) > b * ratio, \end{cases} \tag{3.8}$$

$$t = \frac{f(x,y) - a * ratio}{(b-a) * ratio}, \ \ a, b \in \mathbb{R}, \ \ a, b > 0, \ \ b > a, \tag{3.9}$$

kde vo vzťahu (3.8) $I(x, y)_{Res}$, $I(x, y)_{DDT}$ a $I(x, y)_{Conv}$ značia farebné informácie v jednotlivých obrazových bodoch ($Res-$výsledný obraz, $DDT-$výsledok dátovo závislej metódy, $Conv-$výsledok konvolučnej metódy), (x, y) je umiestenie pixlov. Parametre a, b pomocou informácií z Tex_{DM} určia oblasti, kde sa kombinujú výsledky rekonštrukčných techník. Uvedená stmeľovacia funkcia je po častiach lineárna, jej priebeh je znázornený na Obrázku 3.21 s hrubou čiarou.

Empiricky zistená hodnota parametrov a, b je $a = 1$ a $b = 3$. Zatiaľ nebol vykonaný detailný test vplyvu týchto parametrov na kvalitu výsledného obrazu. Uvedená stmeľovacia funkcia bola navrhnutá pre účely zväčšenia obrazu o 200% a viac. Namiesto po častiach lineárnej funkcie je možné zvoliť použitie ľubovoľnej funkcie.

Poznámka. Z popísaných prístupov neboli publikované metódy uvedené v častiach 3.3 (*Quasi-DDT*), 3.5 (kompresia viacrozmerných dát) a 3.7 (kombinácia *DDT* a konvolučných techník). Techniky rozdelenie na bloky (časť 3.1) a *SALA* (časť 3.2)



Obr. 3.21: Po častiach lineárna stmeľovacia funkcia, vyjadrená vzťahom (3.8).

boli publikované v [Tót06]. Rozšírenie DDT do vyšších dimenzií (časť 3.4) bolo publikované v [TVFG07]. Paralelný výpočet DDT pomocou grafického hardvéru bola akceptovaná na publikáciu [CTS*].

Kapitola 4

Analýza výsledkov

Nasledujúca časť práce je venovaná analýze výsledkov. Za aplikačnú oblasť bola zvolená rekonštrukcia obrazu, pričom sa používali štandardné testovacie obrázky z oblasti spracovania obrazu. Z týchto obrázkov boli vytvorené rôzne testovacie množiny. V prípade verifikácie dátovo závislého prístupu v 3D boli použité syntetické dáta. Implementácia bola realizovaná v prostredí Borland C++ Builder 5.0 a Microsoft Visual Studio .NET 2003 pre platformu Microsoft Windows. Použili sa pritom rôzne softvérové knižnice ako *OpenCV* 1.0 (*Open Source Computer Vision*) [Int], *CGAL* 3.01 (*Computational Geometry Algorithms Library*) [CGA] a *VTK* 5.0 (*Visualization ToolKit*) [Kit]. Výsledky pre konvolučné techniky boli vytvorené pomocou softvérového balíka *ImageMagick* [Ima]. Testy boli vykonané na nasledujúcej počítačovej zostave: Intel Core 2 6400 (2.13GHz) CPU, 2GB RAM, NVidia GeForce 7900 GTX, operačný systém Microsoft Windows XP SP2.

Poznámka. Pre lepšie porovnanie jednotlivých výsledkov je odporúčané pozrieť si aj digitálnu verziu práce, ktorá sa nachádza na domovskej stránke dizertačného projektu, odkaz je uvedený v Dodatku A.

4.1 Charakteristika porovnávacích kritérií

Objektívne hodnotenie kvality rekonštrukcie je náročnou úlohou, ktorá je závislá od aplikačnej oblasti. Preto je dôležité určiť metódy, ktoré budú slúžiť na vyhodnotenie výsledkov.

4.1.1 Váha triangulácie

Z matematického hľadiska dôležitú informáciu dáva dosiahnutá váha výslednej triangulácie. Základná hypotéza je nasledujúca: za ideálnu rekonštrukciu je považovaná triangulácia s minimálnou váhou. Toto kritérium je vhodné na porovnanie medzi triangulačnými prístupmi. Problémom sú však porovnávania s inými prístupmi, ako napríklad s konvolučnými technikami. Ďalej, na základe tejto informácie nie je možné určiť adekvátnosť voľby cenovej funkcie. Na druhej strane tento postup je nezávislý od aplikačnej oblasti.

4.1.2 Perceptuálne metriky

Vo väčšine prípadov pre jednotlivé aplikačné oblasti existujú metódy, ktoré majú za úlohu porovnať výsledky rekonštrukcií. V prípade rekonštrukcie obrazu takým meradlom sú takzvané *perceptuálne metriky*, popísané v [Cec02, WB02, WACBS04, Win05]. Zvyčajne sa používajú na vyhodnotenie kvality kompresie alebo iných operácií, kde je potrebné zmerať zmenu kvality obrazu. Hlavným nedostatkom použitia perceptuálnych metrík je nutnosť zhody veľkostí porovnaných obrázkov. V tomto prípade to znamená, že obrázok najprv musí byť prevzorkovaný na nižšie rozlíšenie (zmenšenie), potom zväčšený naspäť na veľkosť pôvodného obrázku pomocou testovaných rekonštrukčných techník. Následne rekonštruované obrazy sa porovnávajú s pôvodným obrázkom. Z uvedených informácií vyplýva, že výsledná hodnota môže byť ovplyvnená aj spôsobom zmenšenia. V snahe minimalizovať tento vplyv, jednotlivé obrázky boli zmenšené nasledujúcim spôsobom. Používala sa bilineárna interpolácia, obrázok bol iteratívne zmenšený vždy na polovičnú veľkosť, podobným spôsobom ako je vytvorená textúra pre techniku mipmapping. Z toho vyplýva, že v nasledujúcich testoch sa vždy bude jednať o zväčšenie 2^n násobok veľkosti zmenšeného obrazu, $n \in \mathbb{N}$, teda 200%, 400%, 800%, 1600%, atď. Matematická definícia použitých perceptuálnych metrík vyzerá nasledovne:

Korelácia(*Correlation - C*):

$$C = 1 - \frac{\sum_{i,j} |x_{ij} - y_{ij}|}{N \cdot I_{max}} \quad , \tag{4.1}$$

kde x_{ij} a y_{ij} sú farebné informácie pixlov z pôvodného a z rekonštruovaného obrazu, $i = 1, \ldots, x_{max}$ a $j = 1, \ldots, y_{max}$. N značí počet pixlov v obrázku, teda $N = x_{max} \times y_{max}$; I_{max} je hodnota maximálnej možnej intenzity, napríklad 255 ak hodnoty sú brané z intervalu $< 0, 255 >$.
Krížová korelácia (*Cross Correlation - CC*):

$$CC = \left| \frac{\sum_{i,j} x_{ij} y_{ij} - N\bar{x}\bar{y}}{\sqrt{(\sum_{i,j} x_{ij}^2 - N\bar{x}^2)(\sum_{i,j} y_{ij}^2 - N\bar{y}^2)}} \right| \quad , \tag{4.2}$$

kde $\bar{x}$ a $\bar{y}$ sú stredné hodnoty:

$$\bar{x} = \frac{1}{N} \sum_{i,j} x_{ij} \ , \quad \bar{y} = \frac{1}{N} \sum_{i,j} y_{ij} \quad . \tag{4.3}$$

Stredná kvadratická odchýlka (*Mean Square Error - MSE*):

$$MSE = \frac{1}{N} \sum_{i,j} (y_{ij} - x_{ij})^2 \quad . \tag{4.4}$$

Odstup signál-šum (*Signal-to-Noise Ratio - SNR*):

$$SNR(dB) = 10 \cdot log \frac{\sum_{i,j} x_{ij}^2}{N \cdot MSE} \ . \tag{4.5}$$

Vrcholový odstup signál-šum (*Peak Signal-to-Noise Ratio - PSNR*):

$$PSNR(dB) = 10 \cdot log \frac{I_{max}^2}{MSE} \ . \tag{4.6}$$

Univerzálny koeficient kvality obrazu (*Universal Image Quality Index - UIQI*):

$$UIQI = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)(\bar{x}^2 + \bar{y}^2)} \ , \tag{4.7}$$

kde:

$$\begin{gathered} \sigma_x^2 = \frac{1}{N-1}\sum_{i,j}(x_{ij} - \bar{x})^2, \ \ \sigma_y^2 = \frac{1}{N-1}\sum_{i,j}(y_{ij} - \bar{y})^2, \\ \sigma_{xj} = \frac{1}{N-1}\sum_{i,j}(x_{ij} - \bar{x})(y_{ij} - \bar{y}) \ . \end{gathered} \tag{4.8}$$

Štrukturálna podoba (*Structural Similarity - SSIM*):

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{(\bar{x}^2 + \bar{y}^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \ , \tag{4.9}$$

kde C_1 a C_2 sú konštanty, ich hodnota je v tomto prípade $C_1 = 6.5025$, $C_2 = 58.5225$ (pričom intenzita jednotlivých pixlov je z $< 0, 255 >$). Detailný popis významu týchto konštánt sa nachádza v článku [WACBS04].

Vyššia hodnota uvedených perceptuálnych metrík znamená vyššiu kvalitu rekonštrukcie, jedinou výnimkou je metrika *MSE*, kde nižšia hodnota znamená lepší výsledok. Členenie uvedených metrík môže byť nasledujúce: štatisticky založené metódy (*C*, *CC*, *MSE*, *SNR*, *PSNR*), metódy založené na poznatkoch o ľudskom vnímacom systéme (*UIQI*, *SSIM*).

4.1.3 Rozdielové obrazy

Ďalšia metóda, použiteľná špeciálne pre oblasť rekonštrukcie obrazu, je vyhotovenie rozdielového obrazu, to jest odčítavanie výsledku jednotlivých techník od pôvodného obrázku. Aj v tomto prípade je nutnosť zhody veľkosti porovnaných obrázkov. Takýto rozdielový obraz je zobrazený na Obrázku 4.1, označený ako *rozdielový obraz (RGB)*. Odčítavanie hodnôt bolo v tomto prípade uskutočnené vo farebnom priestore *RGB*. Za korektnejšie možné považovať výsledky zmerané v perceptuálne lineárnom farebnom priestore, v takomto prípade rozdiel medzi dvoma farbami viac zodpovedá skutočnostiam ľudského vnímacieho systému. Pre tento účel bola použitá aproximácia vzdialenosti vo farebnom priestore $L^*u^*v^*$, ktorá vyzerá nasledovne:

$$\begin{gathered}\triangle(L*u*v*)_{apr} = \sqrt{\left(2+\frac{\bar{r}}{256}\right)\triangle R^2 + 4\triangle G^2 + \left(2+\frac{255-\bar{r}}{256}\triangle B^2\right)},\\ \bar{r} = \frac{R_1+R_2}{2},\\ \triangle R = R_1 - R_2, \triangle G = G_1 - G_2, \triangle B = B_1 - B_2,\end{gathered} \tag{4.10}$$

kde $R_i, G_i, B_i, i = \{1,2\}$, značia jednotlivé farebné informácie v RGB farebnom priestore. $\triangle(L*u*v*)_{apr}$ je aproximáciou vzdialenosti medzi dvoma farbami v $L*u*v*$ farebnom priestore. Pre lepšiu orientáciu vo výsledkoch tento rozdiel bol namapovaný na farebnú škálu. Takto získaná vizualizácia rozdielového obrazu je zobrazená na Obrázku 4.1, označené ako *rozdielový obraz (L*u*v*)*. Rozdielové obrazy dávajú dôležitú informáciu o rozložení rekonštrukčných chýb. Pomocou perceptuálnych techník je možné vyčísliť množstvo rekonštrukčných chýb, ale o ich distribúcií nie je žiadna informácia. Tieto dve hodnotiace kritéria sa v tomto zmysle dopĺňajú.

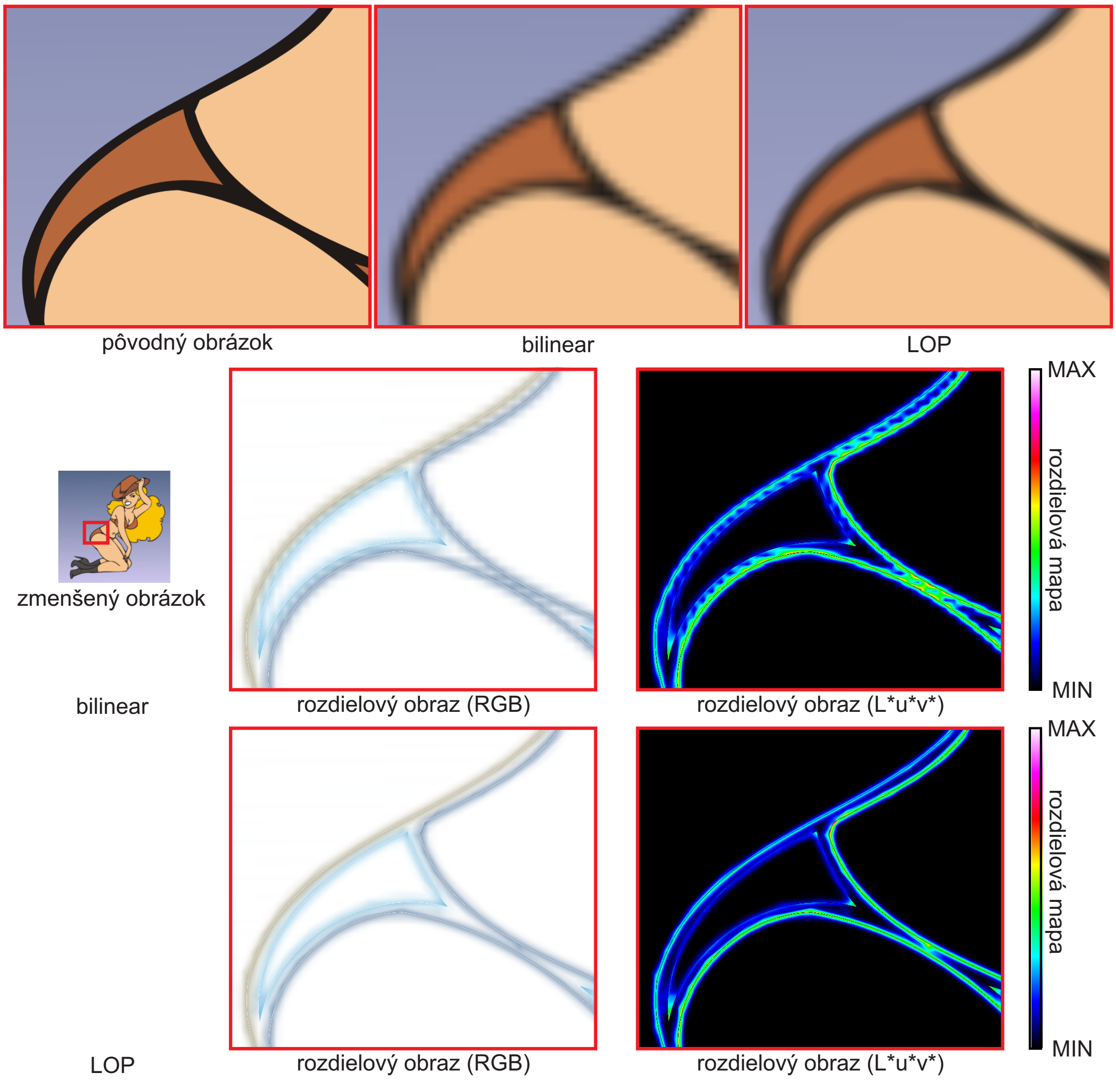
4.2 Analýza vplyvu cenových funkcií

Medzi viacero faktorov, ktoré ovplyvňujú kvalitu rekonštrukcie v prípade dátovo závislých triangulácií patrí aj voľba cenovej funkcie. Tento výber je závislý od aplikačnej oblasti a nie je možné určiť všeobecne ideálnu voľbu. Nasledujúci test bol zameraný na rekonštrukciu obrazu, pričom sa používali cenové funkcie predstavené v časti 2.1.3. Za optimalizačnú techniku bola zvolená *LOP*. Merania boli vykonané na 20-ich obrazoch, pričom sa išlo o magnifikáciu jednotlivých obrázkov o 400%, respektíve 800%. Táto dátová množina bude označená ako *Dataset 1*. Veľkosti zmenšených obrázkov sa pohybovali od 64×60 do 180×144 pixlov. Výsledky boli porovnané pomocou perceptuálnych techník, priemerné hodnoty jednotlivých metrík sú zobrazené v Tabuľke 4.1.

Meranie kvality pomocou preceptuálnych metrík - *Dataset 1*							
	C **(%)**	***CC*** **(%)**	***MSE***	***SNR*** **(dB)**	***PSNR*** **(dB)**	***UIQI***	***SSIM***
ABN	96,198	94,131	280,582	18,618	24,526	0,35687	0,63553
DLP	96,176	94,111	282,684	18,582	24,491	0,35505	0,63423
DP	96,168	94,113	283,047	18,579	24,487	0,35479	0,63414
JND	96,131	94,140	283,637	18,566	24,475	0,35416	0,63383
SCF	96,148	94,129	282,573	18,583	24,492	0,35523	0,63430

Tabuľka 4.1: Priemerná kvalita výsledkov z *Dataset 1* na základe perceptuálnych metrík pre rôzne cenové funkcie.

Podľa dostupnej literatúry jediné podobné porovnanie pre oblasť rekonštrukcie obrazu bolo prezentované v práci [YBS01]. V tomto prípade sa jednalo len

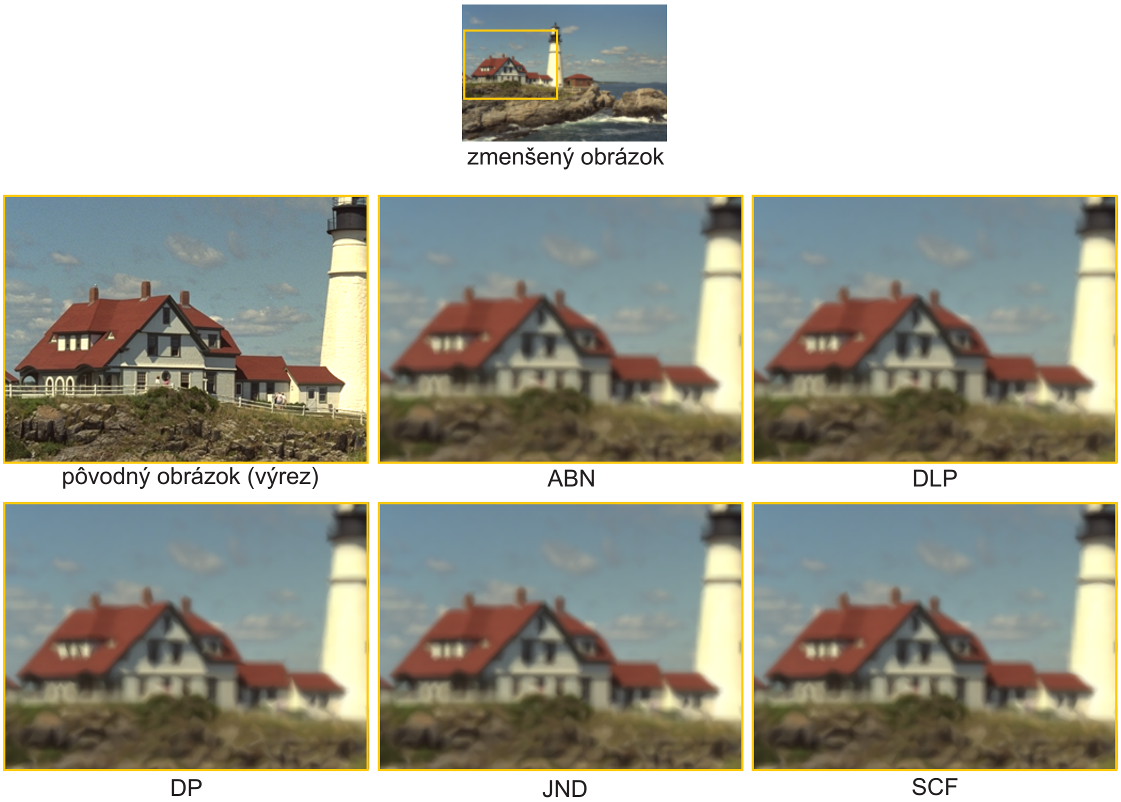


Obr. 4.1: Rozdielové obrazy získané v rôznych farebných priestoroch. Na ukážku boli použité rekonštrukčné výsledky, kde sa zväčšili obrázky o 1600% pomocou bilineárnej interpolácie a pomocou techniky *LOP*.

o porovnanie na základe subjektívnych pozorovaní autorov, a neboli uskutočnené merania ohľadom kvality výsledkov. Podľa autorov spomenutého článku najlepšie správanie ukázala cenová funkcia *SCF*. Na základe výsledkov, ktoré sú prezentované v Tabuľke 4.1, nie je možné potvrdiť toto tvrdenie. Rozdiely medzi jednotlivými cenovými funkciami sú veľmi malé, a nie je možné to považovať za štatisticky signifikantný rozdiel.

Ďalšie merania boli vykonané za účelom zistiť koreláciu medzi výsledkami jednotlivých cenových funkcií. Bola zistená signifikantná podobnosť, vzhľadom na to že výsledky sa väčšinou pohybovali medzi hodnotami 99,7% - 99,9%. Toto bolo potvrdené aj pomocou rozdielových obrazov medzi rekonštrukčnými výsledkami.

Vizuálna kvalita výsledkov jednotlivých cenových funkcií je veľmi podobná aj na základe subjektívneho vnímania. Najlepší vizuálny dojem vzbudili cenové funkcie



Obr. 4.2: Porovnanie výsledkov dosiahnutých pomocou rôznych cenových funkcií pri 400% magnifikácii.

ABN a *SCF*. Ukážka výsledkov z *Dataset 1* je zobrazená na Obrázku 4.2, kde nie je možné si všimnúť veľké rozdiely medzi jednotlivými metódami. Podľa skúseností je možné konštatovať, že ak vo výslednom obraze sa prejavujú artefakty, je vhodné skúsiť inú cenovú funkciu. Rozdiely medzi jednotlivými cenovými funkciami spočívajú v distribúcií artefaktov, ale nie v ich množstve.

Pomocou dostupných prostriedkov nebolo možné zostaviť poradie podľa kvality výstupu, v prípade rekonštrukcie obrazu testované cenové funkcie dávajú porovnateľné výsledky. V nasledujúcich meraniach bola použitá cenová funkcia *SCF*. Tento výber možno považovať za korektný, nakoľko sa nepodarilo ukázať jednoznačnú prevahu ani jednej cenovej funkcie.

4.3 Porovnanie *DDT* a konvolučných techník

Okrem cenových funkcií dôležitý vplyv má na výsledok dátovo závislých triangulácií aj použitá optimalizačná technika. Na testovanie bolo vybratých sedem dátovo závislých techník, z toho tri existujúce (*LOP*, *LAT*, *SA*) a štyri nové prístupy (*Quasi-LOP*, *GLOP*, *Quasi-SA*, *SALA*). Tieto metódy boli porovnané s nasledujúcimi konvolučnými metódami: box filter (interpolácia pomocou najbližšieho suseda), bilineárna interpolácia, b-spline filter, Mitchellov filter, Lanczosov filter. Na vytvorenie výsledkov konvolučných techník sa používal softvérový balík *ImageMagick* [Ima]. Výsledky boli zmerané na 20-ich obrázkoch z *Dataset 1* a porovnané pomocou perceptuálnych metrík. Dosiahnuté priemerné hodnoty sú zobrazené v Tabuľke 4.2.

Meranie kvality pomocou preceptuálnych metrík - *Dataset 1*							
	C (%)	*CC* (%)	*MSE*	*SNR* (dB)	*PSNR* (dB)	*UIQI*	*SSIM*
box filter	95,988	93,443	308,978	18,115	24,024	0,24529	0,60582
bilineárna interpolácia	95,957	93,798	296,307	18,351	24,259	0,33831	0,62593
b-spline filter	95,730	92,983	332,833	17,809	23,718	0,30255	0,60401
Mitchell filter	96,028	93,951	288,989	18,472	24,380	0,34724	0,63011
Lanczos filter	96,275	94,583	259,666	18,981	24,890	0,38404	0,64910
LOP	96,148	94,129	282,573	18,583	24,492	0,35523	0,63430
LAT	96,180	94,173	281,457	18,603	24,511	0,35614	0,63493
Quasi-LOP	96,044	93,915	290,782	18,451	24,360	0,34203	0,62929
GLOP	96,059	93,873	293,439	18,408	24,317	0,33834	0,62734
SA	96,030	93,986	287,568	18,488	24,397	0,33929	0,62814
Quasi-SA	95,964	93,795	295,575	18,363	24,272	0,33042	0,62356
SALA	96,104	94,027	284,715	18,538	24,447	0,34456	0,63068

Tabuľka 4.2: Priemerná kvalita výsledkov z *Dataset 1* na základe perceptuálnych metrík pre rôzne rekonštrukčné metódy.

Na základe týchto výsledkov najlepšie dopadol Lanczosov filter, ktorého v kvalite nasledujú triangulačné prístupy *LAT* a *LOP*. Všetky perceptuálne metriky určili toto poradie na prvých troch priečkach[1]. Najhoršie výsledky dávali box filter (interpolácia pomocou najbližšieho suseda) a b-spline filter, čo sa aj dalo očakávať. Spomedzi stochastických prístupov najlepšie výsledky dosahovala metóda *SALA*.

Množstvo rekonštrukčných chýb je len jeden z aspektov vnímania kvality rekonštrukcie. Vzhľadom na ľudský vnímací systém, veľmi dôležitú úlohu zohráva aj samotná distribúcia rekonštrukčných artefaktov. Ako už bolo spomenuté na začiatku práce, konvolučné techniky v tomto smere zaostávajú za dátovo závislými metódami. Distribúcia artefaktov v prípade *DDT* je menej systematická a z tohto dôvodu tieto chyby pôsobia menej rušivo. Konkrétne rekonštrukčné výsledky z *Dataset 1* sú zobrazené na Obrázkoch 4.3, 4.4. V prípade výsledkov z konvolučných techník sú rozpoznateľné osovo orientované, pravidelne sa opakujúce chyby vo vysokofrekvenčných oblastiach, až na výnimku b-spline filtru, ktorý spôsobil rozmazanie celého obrazu, vrátane hranových oblastí. Toto je možné ukázať aj pomocou rozdielových obrazov, na to aby to ale bolo dobre viditeľné je potrebné väčšia miera magnifikácie, ako napríklad 1600% zväčšenie zobrazené na Obrázku 4.1. Z hľadiska subjektívneho vnímania najlepšie výsledky dávajú *LOP*, *LAT* a *SALA*; z konvolučných techník Lanczosov a Mitchellov filter.

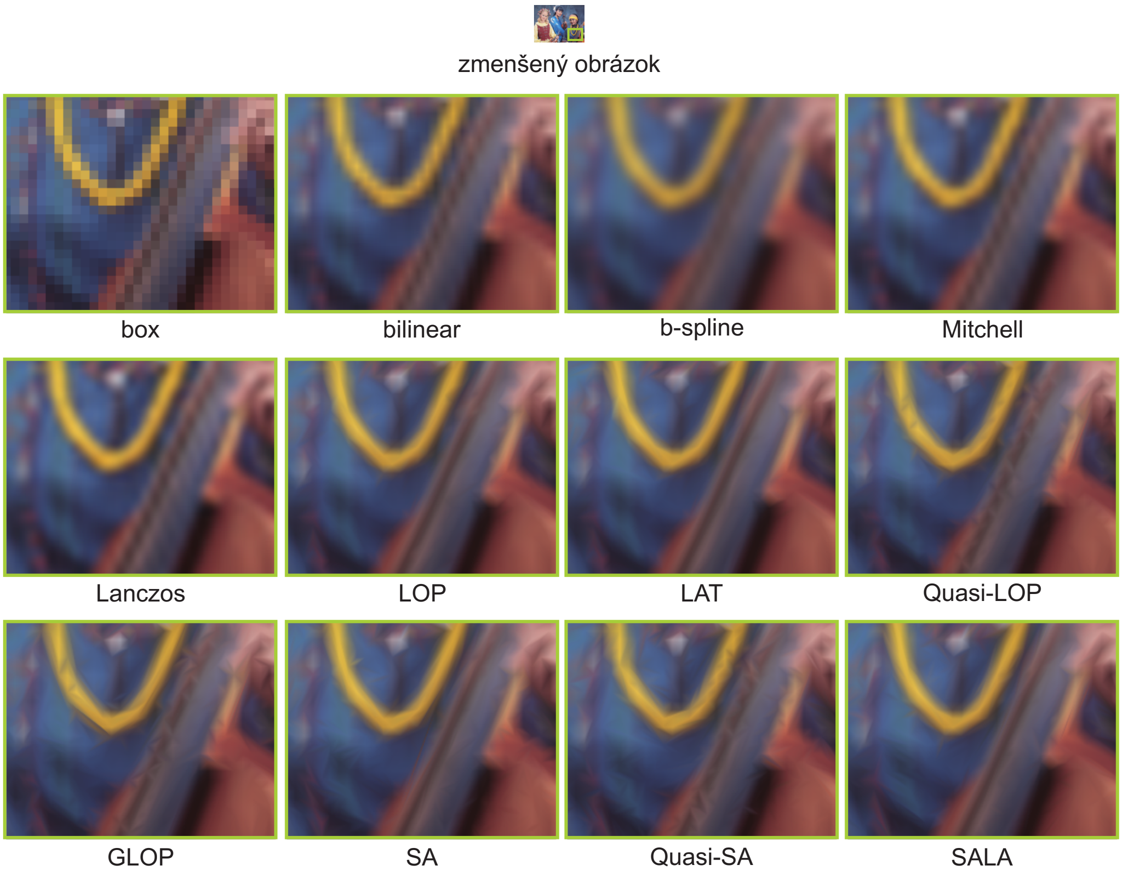
Ďalším spôsobom hodnotenia kvality je cena dosiahnutej triangulácie. V tomto prípade sa jedná len o porovnanie medzi dátovo závislými metódami. Z týchto úvah je vynechaná metóda *GLOP*. Dôvodom je, že cena triangulácie je rátaná iným spôsobom ako v prípade ostatných techník, kde sa používala cenová funkcia *SCF*. Tieto

[1] V prípade metriky *MSE* nižšia hodnota znamená lepšiu rekonštrukciu.

Obr. 4.3: Rekonštrukcia pri 400% zväčšení pomocou rôznych metód.

výsledky sú zobrazené v Tabuľke 4.3. *Začiatočná cena* značí váhu úvodnej triangulácie, *konečná cena* váhu výslednej triangulácie. Miera zlepšenia ceny triangulácie v percentách je označená ako *zlepšenie*. Najnižšiu cenu dosiahla metóda *LAT*, ktorú nasledovali *LOP* a *SALA*. Zaostávanie stochasticky založených metód je prekvapujúce, podrobnejšia analýza bude poskytnutá v ďalších častiach práce.

Výsledky rekonštrukcie závisia aj od typu testovaných dát (v prípade rekonštrukcie obrazu: miera zašumenia, miera štruktúrovanosti obsahu obrazu, atď.). Dátovo závislým metódam nevyhovujú zašumené obrazy, v takýchto prípadoch je pravdepodobnosť vzniku rekonštrukčných artefaktov oveľa vyššia (lebo zvýrazňujú aj šum). Veľmi dobré výsledky sú dosiahnuté pre obsahovo štruktúrované obrazy, kde sa vyskytuje množstvo hrán. Predspracovanie na odstránenie šumu môže byť dôležitou úlohou pri presadení *DDT* založených techník v komerčnej sfére.



Obr. 4.4: Rekonštrukcia pri 800% zväčšení pomocou rôznych metód.

4.4 Analýza nových algoritmov

Nasledujúca časť obsahuje analýzu výsledkov implementovaných nových algoritmov.

4.4.1 Porovnanie *Quasi-LOP* a *GLOP* s metódou *LOP*

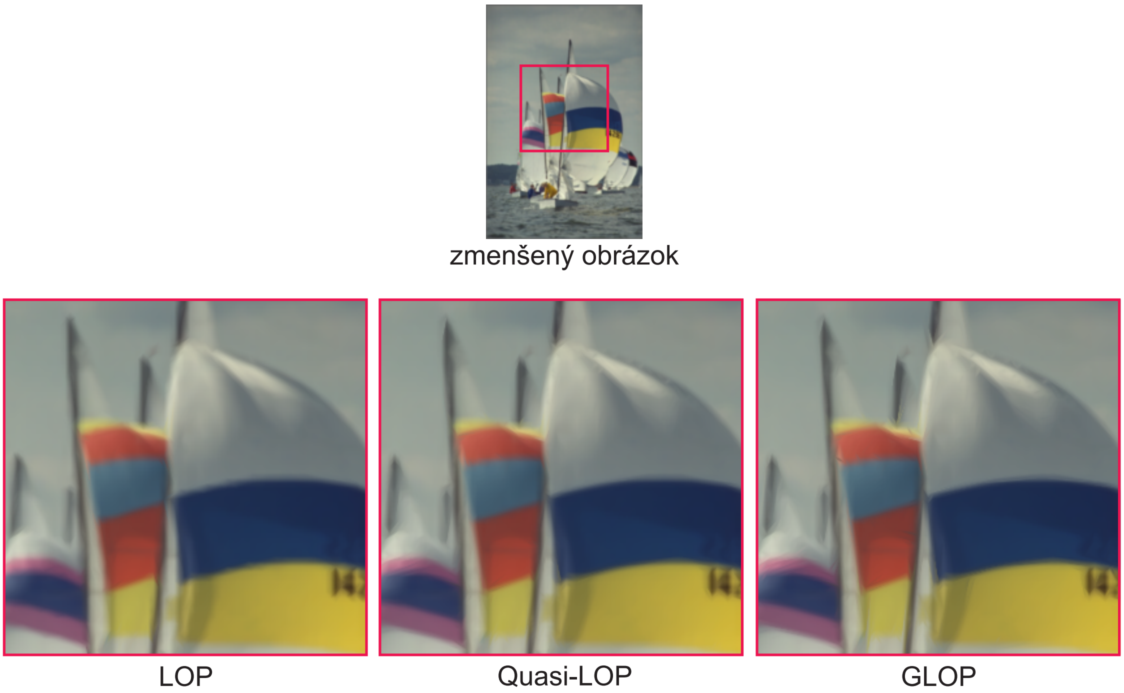
Metódy *Quasi-LOP* a *GLOP* sú založené na princípe fungovania Lawsonovho optimalizačného procesu, a preto tieto techniky budú posudzované spoločne s výsledkami *LOP*. V prípade *LOP* a *Quasi-LOP* bola použitá cenová funkcia *SCF*, *GLOP* bola vytvorená pomocou váhovania obsahu trojuholníkov na základe variancie funkčných

Meranie kvality - *Dataset 1*			
	začiatočná cena	**konečná cena**	**zlepšenie (%)**
LOP	513,117	221,559	231,594
LAT	513,117	203,316	252,374
Quasi-LOP	513,117	379,953	135,047
SA	513,117	327,789	156,539
Quasi-SA	513,117	522,159	98,268
SALA	513,117	273,754	187,437

Tabuľka 4.3: Priemerné ceny triangulácií pre obrázky z *Dataset 1*.

hodnôt. Na testovanie bola použitá testovacia množina *Dataset 1*.

Z hľadiska perceptuálnych metrík najlepšie výsledky dávala *LOP*, ktorú nasledovali *Quasi-LOP* a *GLOP*, výsledky sú zobrazené v Tabuľke 4.2. Rozdiely medzi *Quasi-LOP* a *GLOP* neboli markantné. V prípade *Quasi-LOP* horšie výsledky sa dajú vysvetliť vďaka rozličnému chápaniu lokálnej optimality jednotlivých hrán. Kvalitu *GLOP* metódy ovplyvňovala voľba váhovej funkcie. Nahradenie váhovej funkcie metódou, ktorá využíva zovšeobecnené cenové funkcie (vzťah 3.7), by mohol priniesť lepšie výsledky.



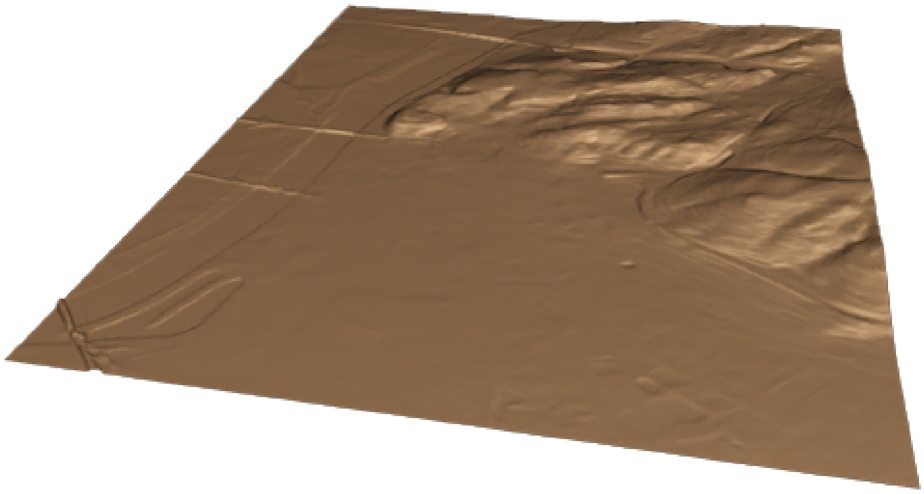
Obr. 4.5: Rekonštrukčné výsledky techník *LOP*, *Quasi-LOP* a *GLOP* pri 400% zväčšení.

Porovnanie vizuálnej kvality výsledkov *LOP*, *Quasi-LOP* a *GLOP* podľa subjektívneho vnímania je ťažkou úlohou. Ukážky sú zobrazené na Obrázkoch 4.5, 4.6. Vo všeobecnosti je možné tvrdiť, že poradie je rovnaké ako v prípade perceptuálnych metrík, teda najlepšiu kvalitu dosahuje *LOP*. Rozdiely sú ale závislé od typu obrazu, a v niektorých prípadoch *Quasi-LOP* a *GLOP* dosahujú vizuálne prijateľnejšiu kvalitu. Predovšetkým sa jedná o hranové oblasti, kde hrana je jednoznačne určená a nevyskytujú sa jemné detaily. V týchto prípadoch *Quasi-LOP* a *GLOP* niekedy dávajú lepšie výsledky ako *LOP*, ktorá v týchto oblastiach občas produkuje artefakty v snahe lepšie modelovať farebný prechod cez danú hranu. Na druhej strane *LOP* jednoznačne lepšie rekonštruuje jemné detaily, alebo združenie hranových oblastí, ktoré ležia blízko k sebe. Ďalej, vo výsledkoch metódy *GLOP* sa často objaví nasledujúci vizuálne zavádzajúci jav. Vďaka zvolenej váhovej funkcii v obraze sa zvýrazňujú izočiary, a vytvárajú sa izolované jednofarebné oblasti. Vizuálne to vytvára podobný dojem, ako keď napríklad 24 bitový farebný obraz je zobrazený na monitore, ktorý podporuje len 16 bitovú farebnú hĺbku.

Obr. 4.6: Rekonštrukčné výsledky techník *LOP*, *Quasi-LOP* a *GLOP* pri 800% zväčšení.

Na základe dosiahnutej ceny triangulácie je možné porovnať len metódy *LOP* a *Quasi-LOP*. V prípade *GLOP* to vylučuje rozlišný spôsob ohodnotenia triangulácie. Výsledné ceny sú uvedené v Tabuľke 4.3. Z pozorovaní možno konštatovať, že najväčší pokles ceny nastane v prvých iteráciach, a to pri všetkých metódach. *LOP* dosahuje nižšiu priemernú cenu triangulácie, čo sa aj dalo očakávať. V prípade *Quasi-LOP* konvergencia prístupu nie je zaručená, a algoritmus končí v tej iterácii, keď celková cena triangulácie ďalej už neklesne. Možno by bolo vhodné vyskúšať aj inú ukončovaciu podmienku, ako napríklad fixne stanovený počet iterácií. Z hľadiska spôsobu konštrukcie *Quasi-LOP* sa dá očakávať, že dosiahnutá cena bude vždy horšia ako v prípade *LOP*. Neznamená to ale, že vďaka svojej charakteristike sa nemôže *Quasi-LOP* vyhýbať lokálne optimálnym riešeniam, ktoré môžu mať vyššiu cenu.

Z aspektu optimálnych triangulácií možno jednoznačne preferovať metódy *LOP* a *GLOP*, a to vďaka zaručenej konvergencii k lokálne optimálnemu stavu. To ale neznamená, že pre praktické účely nie je vhodná metóda *Quasi-LOP*. Okrem rekonštrukcie obrazu môže byť použitá aj v iných oblastiach. Napríklad na Obrázku 4.7 je zobrazená časť výškovej mapy mesta Bratislavy, rekonštruovaná pomocou *Quasi-LOP* metódy s cenovou funkciou *SCF*. Reálne použitie dátovo závislých prístupov vyžaduje aj akceptovateľnú výpočtovú zložitosť, ktorá je silnou stránkou predstavených nových techník *Quasi-LOP* a *GLOP*. V našej implementácii dosiahli výrazne rýchlejšie časy ako metóda *LOP* (*Quasi-LOP* zhruba 10×, *GLOP* približne 5×). Rozdiely v rýchlostiach výpočtov sú vďaka inému chápaniu lokálnej optimality a rozdielnemu spôsobu váhovania triangulácie, výsledné priemerné časy sú zobrazené v Tabuľke 4.4.



Obr. 4.7: *Quasi-LOP* triangulácia výškovej mapy Bratislavy. Dáta poskytol Magistrát mesta Bratislavy.

Meranie rýchlosti výpočtov - *Dataset 1*			
	LOP	*Quasi-LOP*	**GLOP**
optimalizácia (s)	5,567	0,536	1,149

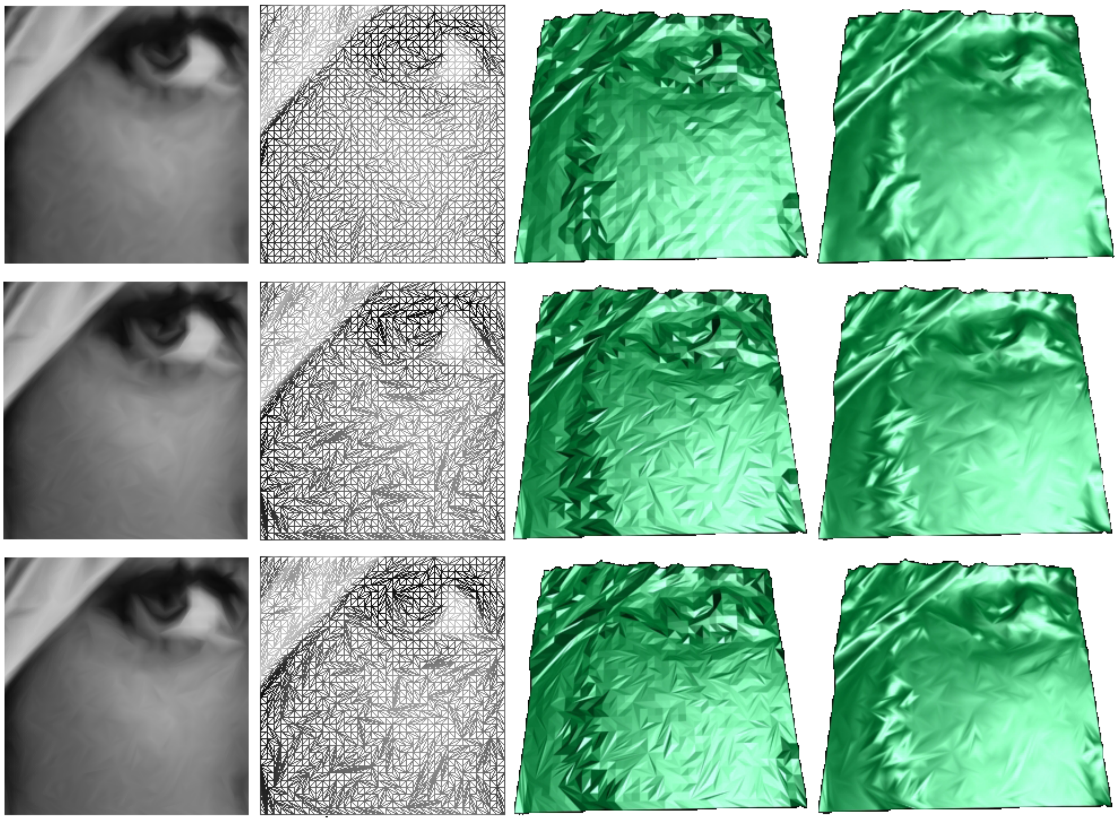
Tabuľka 4.4: Priemerné hodnoty rekonštrukčných časov pre *Dataset 1*.

4.4.2 Porovnanie *Quasi-SA* a *SALA* s metódou *SA*

Nové techniky *Quasi-SA* a *SALA* vychádzajú zo simulovaného žíhania, a preto podobne ako v prechádzajúcom prípade budú tieto metódy posudzované spoločne. Za testovaciu množinu bola zvolená *Dataset 1*, pričom sa vo všetkých prístupoch bola použitá cenová funkcia *SCF*. Spomínané stochasticky založené techniky sú citlivé na nastavenie parametrov, používali sa štandardné nastavenia, popísané v časti 2.2.2.

Výsledok analýzy pomocou perceptuálnych techník je zobrazený v Tabuľke 4.2. Najlepšie hodnoty dosiahla metóda *SALA*, ktorú nasledovala *SA* a nakoniec prístup *Quasi-SA*. Zaostávanie *Quasi-SA* je výraznejšie ako rozdiely medzi *SALA* a *SA*.

Z hľadiska vizuálnej kvality správanie stochastických prístupov sa dá charakterizovať nasledovne pre aplikačnú oblasť rekonštrukcie obrazu. V porovnaní s deterministickými prístupmi vytvárajú častejšie artefakty, príčinou toho je odlišný priebeh optimalizácie. Zhoršujúce kroky vytvoria v štruktúre triangulácie také nezvratné zmeny, ktoré sú z hľadiska vizuálnej kvality neakceptovateľné. V oblastiach, kde sa nevyskytujú hrany (napríklad jemné farebné prechody), by optimálnym riešením

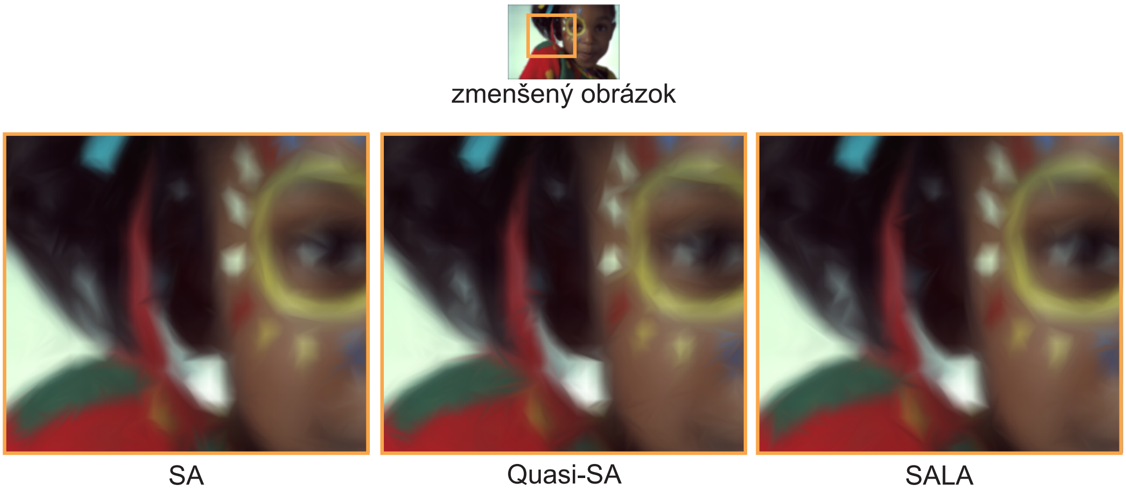


Obr. 4.8: 600% zväčšenie pomocou rôznych *DDT* techník, *LOP* - prvý riadok, *SA* - druhý riadok, *SALA* - tretí riadok. V prvom stĺpci sú zobrazené rekonštrukčné výsledky, v druhom príslušná trojuholníková sieť, tretí a štvrtý stĺpec obsahujú vizualizované výškové mapy.

(z hľadiska vizuálnej kvality) bola triangulácia blízka ku *DT*. Tu stochastické algoritmy vytvoria dlhé a úzke trojuholníky, ktoré sa postupom optimalizácie napravia nesprávne, ako je to zobrazené na Obrázku 4.8 (kvôli porovnaniu je uvedený aj rekonštrukčný výsledok *LOP*).

Poradie metód podľa subjektívneho vnímania vizuálnej kvality je rovnaké ako poradie určené pomocou perceptuálnych metrík. Najlepšie výsledky dosahuje *SALA*, ktorej kvalita sa približuje ku kvalite výsledkov z deterministických metód. Ukážky výsledkov zo stochastických prístupov sú zobrazené na Obrázkoch 4.9, 4.10. *SALA* zvyčajne vytvára menšie množstvo artefaktov v oblastiach hrán ako *SA* alebo *Quasi-SA*.

Ďalším faktorom vnímania kvality je cena dosiahnutej triangulácie. Význam stochasticky založených metód, využívajúce myšlienku simulovaného žíhania, je z hľadiska aproximácie *MWT* veľký. Vo všeobecnosti tieto techniky dosahujú nižšiu cenu ako deterministické metódy. Dosiahnuté ceny sú zobrazené v Tabuľke 4.3, a sú vyššie ako v prípade deterministických metód. Dôvodom toho môže byť zlé nastavenie parametrov, alebo špeciálna charakteristika rekonštruovaných dát (príbuznosť



Obr. 4.9: Rekonštrukčné výsledky pomocou použitia stochastických prístupov, pri 800% zväčšení.

s Markovovými náhodnými poľami). Priemerná rýchlosť výpočtu jednotlivých prístupov je uvedená v Tabuľke 4.5. Najrýchlejšia metóda je *Quasi-SA*, ktorú nasleduje *SA*, najpomalšia technika medzi stochastickými prístupmi je *SALA*.

Meranie rýchlosti výpočtov - *Dataset 1*			
	SA	***Quasi-SA***	**SALA**
optimalizácia (s)	24,384	10,772	40,361

Tabuľka 4.5: Priemerné hodnoty rekonštrukčných časov pre *Dataset 1*.

V prípade stochastických metód okrem dosiahnutej ceny je zaujímavý aj samotný priebeh optimalizácie. Priemerný vývoj ceny je zobrazený na Obrázku 4.11, pre jednotlivé obrázky z *Dataset 1* priebeh je zobrazený na Obrázku 4.12. Z týchto grafov je viditeľné, že vo všeobecnosti cena triangulácie v prvých iteráciach v prípade *SALA* výrazne poklesne, v prípade *Quasi-SA* výrazne rastie, pre *SA* je tento pokles mierny. V ďalších iteráciach cena klesá zhruba rovnakou rýchlosťou. V prípade *Quasi-SA* sa častokrát stane, že dosiahnutá cena je vyššia ako cena začiatočnej triangulácie. Z toho vyplýva, že pre *Quasi-SA* štandardné nastavenia nie sú vhodné, respektíve *Quasi-* metódy nie sú vhodné na aproximáciu *MWT*. V prípade *SALA* rýchly pokles ceny v prvých iteráciach naznačuje, že experimentovanie s nastaveniami parametrov by mohlo priniesť nižšie cenu výslednej triangulácie. Príliš rýchla konvergencia zabráni metódam simulovaného žíhania dosiahnuť globálne optimum. Vykonanie testov takéhoto charakteru je nad rámec rozsahu tejto práci.

So štandardnými nastaveniami parametrov, testované stochastické procesy nepriniesli adekvátne výsledky. Je to vďaka špeciálnemu rozloženiemu dát, a príbuznosti chápania digitálnych obrazov a Markovových náhodných polí. *SALA* produkovala najnižšiu celkovú váhu triangulácie spomedzi stochastických procesov. Ukázalo sa aj horšie správanie *Quasi-SA* ako *SA*, čo sa dalo predpokladať. Nová heuristika

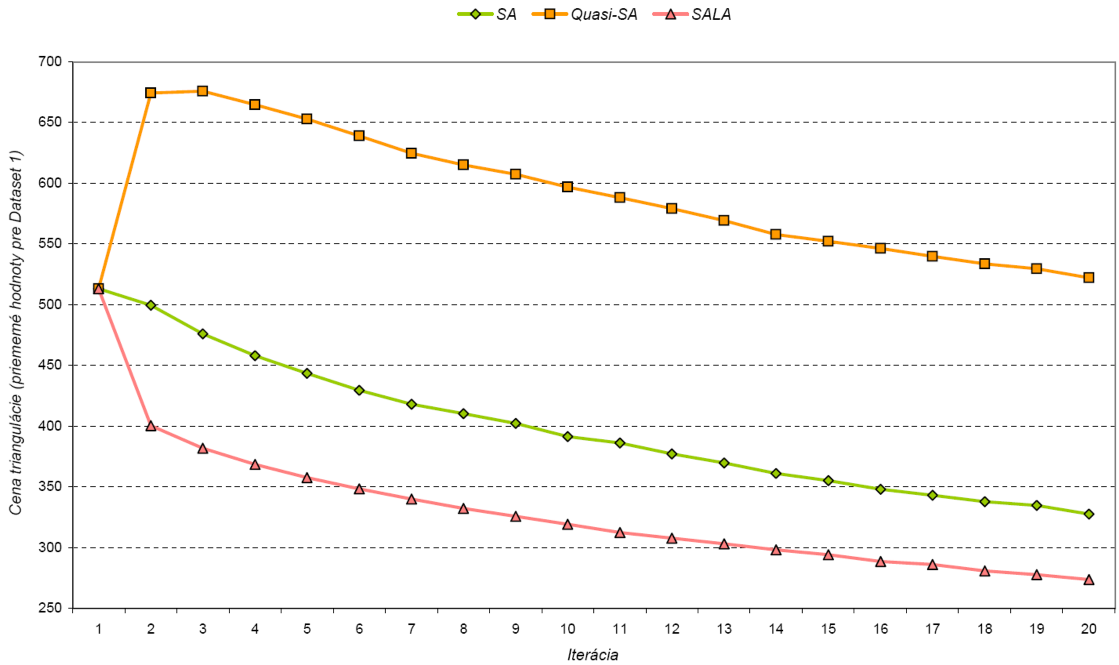
Obr. 4.10: Rekonštrukčné výsledky pomocou použitia stochastických prístupov, pri 800% zväčšení.

SALA môže priniesť kvalitné výsledky pre iné typy rekonštrukčných úloh ako je rekonštrukcia obrazu.

4.5 Aplikácia *DDT* na syntetické 3D dáta

Medzi vlastnými rozšíreniami bola predstavená metóda *GLOP*, testovaná v predchádzajúcich častiach pre aplikačnú oblasť rekonštrukcie obrazu. Primárnou motiváciou pri vzniku tejto techniky bolo rozšírenie dátovo závislých triangulácií do vyšších dimenzií. Pre verifikáciu bola implementovaná *GLOP* aj pre trojrozmerný prípad, podobne ako v prípade rovinnej verzie bolo použité váhovanie na základe variancie. Ponúknuté výsledky sú rekonštrukciou syntetických 3D dát pomocou trilineárnej interpolácie a dátovo závislou technikou *GLOP*. Testovacie volumetrické dáta mali rozlíšenie 32×32×32. Pri vizualizácií výsledkov a vytváraní konvolučných výsledkov bola použitá knižnica *VTK* 5.0 [Kit]. Ukážky sú zobrazené na Obrázkoch 4.13, 4.14.

V prípade dátovo závislej techniky rekonštrukčné chyby nie sú tak relevantné ako vo výsledkoch trilineárnej interpolácie. Konvolučné techniky podobne ako v rovinnom prípade vytvárajú artefakty vo vysokofrekvenčných oblastiach, v tomto prípade na plochách, ktoré nie sú osovo orientované. Aplikácia zovšeobecnených cenových funkcií namiesto váhovania na základe variancie by mohla priniesť ďalšie zvýšenie kvality rekonštrukcie. Ako aplikačná oblasť pre 3D *DDT* môže slúžiť napríklad rekonštrukcia medicínskych dát z *CT*. Subjektívne vnímanie kvality by sa dalo nahradiť rozšírením perceptuálnych metrík do 3D.



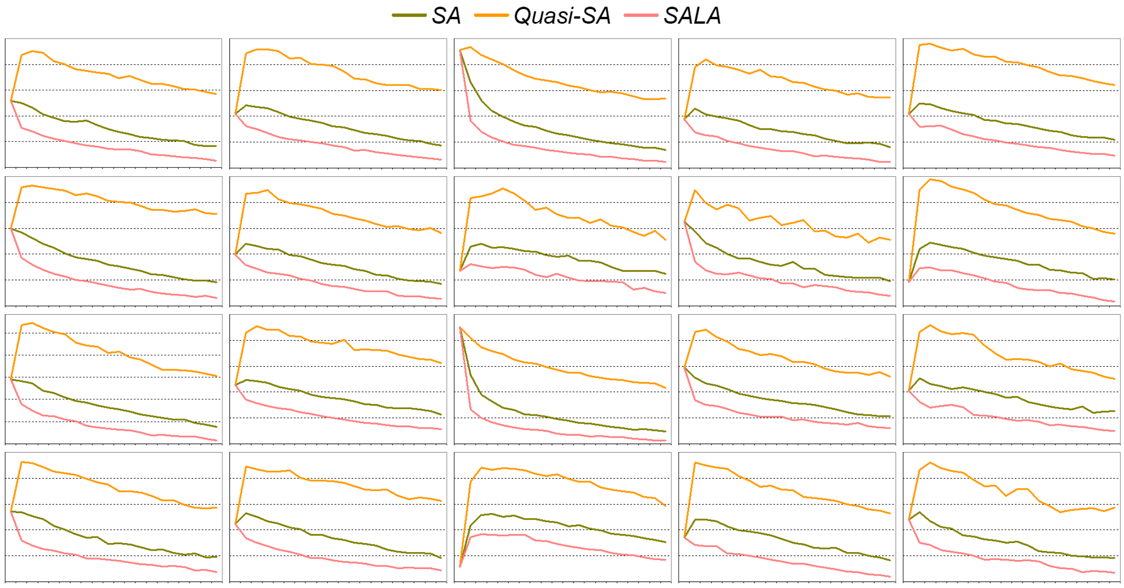
Obr. 4.11: Priemerný vývoj ceny pre *SA*, *Quasi-SA* a *SALA* na základe výsledkov rekonštrukcie *Dataset 1*.

4.6 Analýza paralelnej verzie *DDT*

V sekcii 3.6 bola predstavená paralelná verzia dátovo závislého prístupu, analýzou tejto metódy sa zaoberá nasledujúca časť. Na testovanie boli vytvorené dve sady obrázkov, ktoré budú označené *Dataset 2* a *Dataset 3*. *Dataset 2* obsahuje 12 obrázkov, skoro každý z nich je používaný ako štandardný testovací obázok v oblasti spracovania obrazu. *Dataset 3* pozostáva z 8 obrázkov, ktoré boli vytvorené vo vektorovom editore, a následne rasterizované. Jednotlivé obrázky boli najprv zmenšené a následne zväčšené naspäť na pôvodnú veľkosť, podobne ako v prípade *Dataset 1*. Pri zväčšení sa používali nasledujúce metódy: *LOP*, paralelné dátovo závislé techniky, konvolučné techniky (bilineárna interpolácia, b-spline a Lanczosov filter) vytvorené pomocou *ImageMagick* [Ima]. Miera zväčšenia bolo 400% a 800%, pričom rozlíšenie jednotlivých (zmenšených) obrázkov sa pohybovalo od 64×64 do 250×243 pixlov.

Implementácia paralelnej verzie bola vykonaná Dr. Červeňanským pomocou jazyka *OpengGL*, v prípade fragment programu boli použité dva rôzne jazyky *OpenGL Shading Language* (*glsl*) [Ros05] a *C for graphics* (*Cg*) [FK03]. Výsledky boli zmerané na nasledujúcej počítačovej zostave: Intel Pentium 4 3.0GHz CPU, 1GB RAM s nasledujúcimi grafickými kartami NVidia GeForce 8800 GTS (veľkosť memórie 640MB), NVidia GeForce 9600GT (veľkosť memórie 512MB), ATI Radeon HD 4770 (veľkosť memórie 512MB).

Pri analýze vizuálnej kvality boli použité perceptuálne metriky, podobne ako v predchádzajúcich prípadoch. Výsledky týchto meraní pre *Dataset 2* a *Dataset 3*



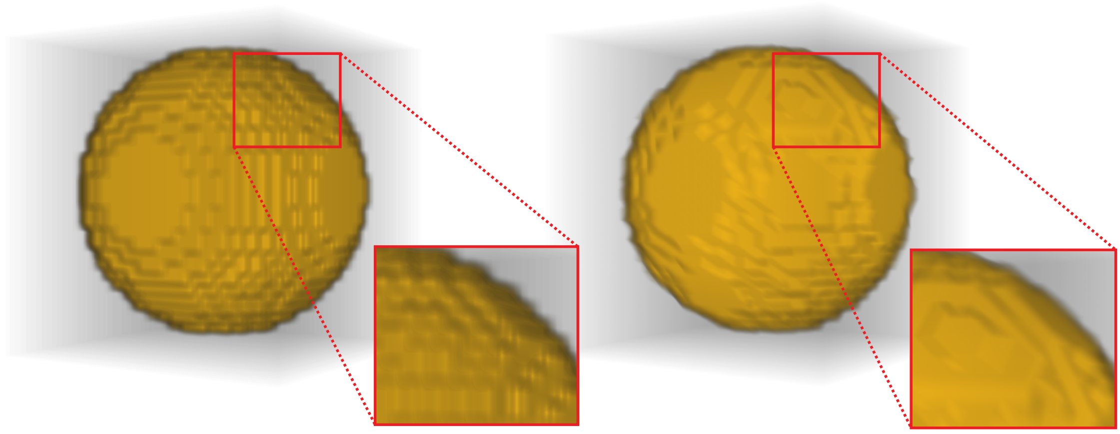
Obr. 4.12: Vývoj ceny pre rekonštrukciu jednotlivých obrázkov z *Dataset 1* pomocou metód *SA*, *Quasi-SA* a *SALA* (*Y*-os predstavuje cenu, *X*-os iterácie).

sú uvedené v Tabuľkách 4.6, 4.7. V prípade *Dataset 2* najlepšie výsledky dosiahol Lanczosov filter a *LOP*, na treťom mieste sa striedali metódy *GPU ExpROI*, *GPU ExpROI2 MaxGain* a *GPU ExpROI3 MaxGain*. Jednotlivé modifikácie základného *GPU* prístupu v každom prípade dosahovali lepšie výsledky ako pôvodná technika, označená v tabuľke ako *GPU pôvodná*. Zložitejšie je to s výsledkami z *Dataset 3*. Najlepšie výsledky dosahovali *LOP* a Lanczosov filter v závislosti od danej perceptuálnej metriky. Na druhej priečke sa nachádza *LOP* a *GPU ExpROI2 MaxGain*, na treťom mieste sa striedali metódy *GPU ExpROI2 MaxGain*, *GPU ExpROI3 MaxGain*. Aj v tomto prípade, až na niektoré výnimky, jednotlivé modifikácie dosiahli lepšie výsledky ako základný *GPU* prístup.

Meranie kvality pomocou perceptuálnych metrík - *Dataset 2*							
	C **(%)**	***CC*** **(%)**	***MSE***	***SNR*** **(dB)**	***PSNR*** **(dB)**	***UIQI***	***SSIM***
bilineárna interpolácia	96,0142	94,7434	294,428	17,7444	24,1102	0,36111	0,63596
b-spline filter	95,3072	93,3080	377,732	16,6075	22,9733	0,29662	0,59566
Lanczos filter	96,3722	95,5652	249,824	18,5715	24,9374	0,41005	0,66093
CPU LOP	96,1783	95,2463	267,577	18,2452	24,6110	0,38757	0,65186
***GPU* pôvodná**	96,1418	95,1506	272,882	18,1580	24,5238	0,38303	0,64873
GPU ExpROI	96,1573	95,1779	270,689	18,1989	24,5647	0,38485	0,65006
GPU MaxGain	96,1485	95,1618	272,204	18,1713	24,5372	0,38335	0,64913
ExpROI MaxGain	96,1587	95,1725	270,879	18,1914	24,5572	0,38472	0,64999
ExpROI2 MaxGain	96,1600	95,1693	270,729	18,1952	24,5611	0,38478	0,65007
ExpROI3 MaxGain	96,1590	95,1728	270,801	18,1927	24,5586	0,38478	0,65003

Tabuľka 4.6: Priemerná kvalita výsledkov z *Dataset 2* na základe perceptuálnych metrík pre rôzne rekonštrukčné techniky.

Konkrétne výsledky sú zobrazené na Obrázkoch 4.15 a 4.16. U konvolučných tech-



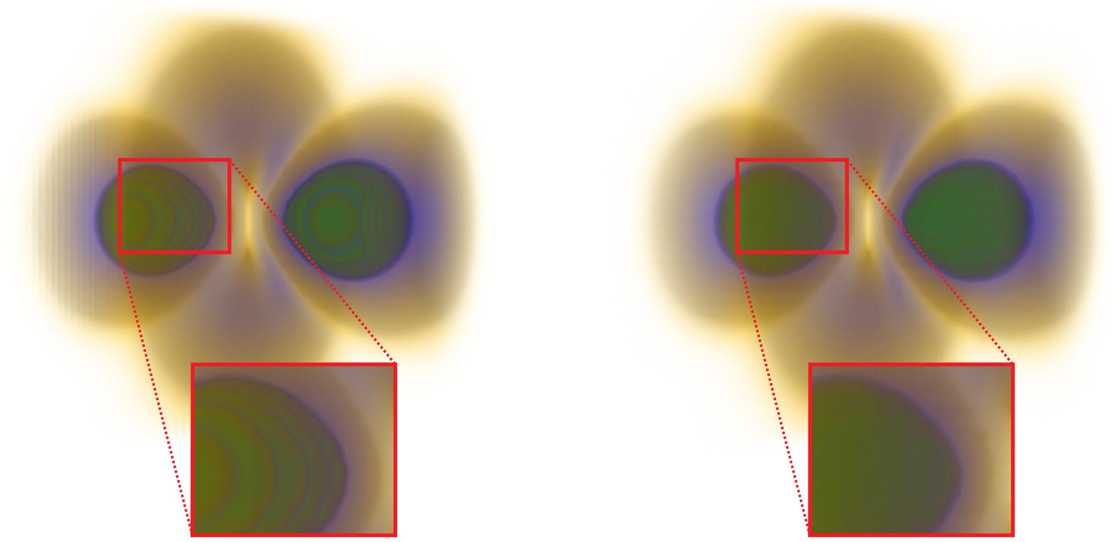
Obr. 4.13: Rekonštrukčné výsledky pre syntetické volumetrické dáta. Trilineárna interpolácia (vľavo) a 3D *GLOP* (vpravo).

Meranie kvality pomocou perceptuálnych metrík - *Dataset 3*							
	C (%)	*CC* (%)	*MSE*	*SNR* (dB)	*PSNR* (dB)	*UIQI*	*SSIM*
bilineárna interpolácia	97,8322	97,1942	343,465	21,2149	23,6096	0,69532	0,90194
b-spline filter	97,1695	95,8883	500,506	19,7293	22,1240	0,62283	0,87987
Lanczos filter	98,1455	97,9042	253,483	22,5276	24,9224	0,57387	0,90993
CPU LOP	98,2470	97,7710	268,149	22,3261	24,7208	0,78067	0,92038
***GPU* pôvodná**	98,1522	97,6275	286,916	22,0282	24,4229	0,77578	0,91565
GPU ExpROI	98,1715	97,6450	283,838	22,0616	24,4563	0,77722	0,91658
GPU MaxGain	98,1570	97,6162	287,492	22,0196	24,4143	0,77605	0,91593
ExpROI MaxGain	98,1982	97,6887	278,899	22,1660	24,5607	0,77827	0,91796
ExpROI2 MaxGain	98,2057	97,6992	277,238	22,1865	24,5812	0,77855	0,91831
ExpROI3 MaxGain	98,2007	97,6920	278,375	22,1702	24,5650	0,77837	0,91806

Tabuľka 4.7: Priemerná kvalita výsledkov z *Dataset 3* na základe perceptuálnych metrík pre rôzne rekonštrukčné techniky.

ník sú pozorovateľné pravidelne sa opakujúce rekonštrukčné chyby vo vysokofrekvenčných oblastiach. Zo subjektívnych pozorovaní a z výsledkov uvedených v tabuľkách sa dá vyvodiť nasledujúci záver. Najlepšie výsledky medzi paralelnými prístupmi dosiahli modifikácie *GPU ExpROI2/3*, ktoré nasledujú *GPU MaxGain* a *GPU ExpROI MaxGain*.

Ako už bolo spomenuté, cena výslednej triangulácie je dôležitým ukazovateľom kvality, vyplýva z nej miera aproximácie *MWT*. V Tabuľkách 4.8, 4.9 sú uvedené údaje, ktoré popisujú tieto výsledky. *Začiatočná cena* značí cenu začiatočnej triangulácie, v tomto prípade sa jedná o Delaunayovu trianguláciu, ktorá sa dá veľmi ľahko generovať vzhľadom na distribúciu vrcholov. *Konečná cena* značí cenu triangulácie po dosiahnutí (lokálne) optimálneho stavu, pre ľahšiu orientáciu vo výsledkoch je uvedená aj miera takto získaného zlepšenia, označené ako *zlepšenie*. V tabuľkách sa nachádza aj počet preklápacích operácií, ktoré boli vykonané jednotlivými prístupmi, označené ako *počet preklápaní*. Najnižšiu cenu v oboch prípadoch dosiahla metóda *CPU LOP*, ktorú nasleduje skupina prístupov so zhruba rovnakou cenou -



Obr. 4.14: Rekonštrukčné výsledky pre syntetické volumetrické dáta. Trilineárna interpolácia (vľavo) a 3D *GLOP* (vpravo).

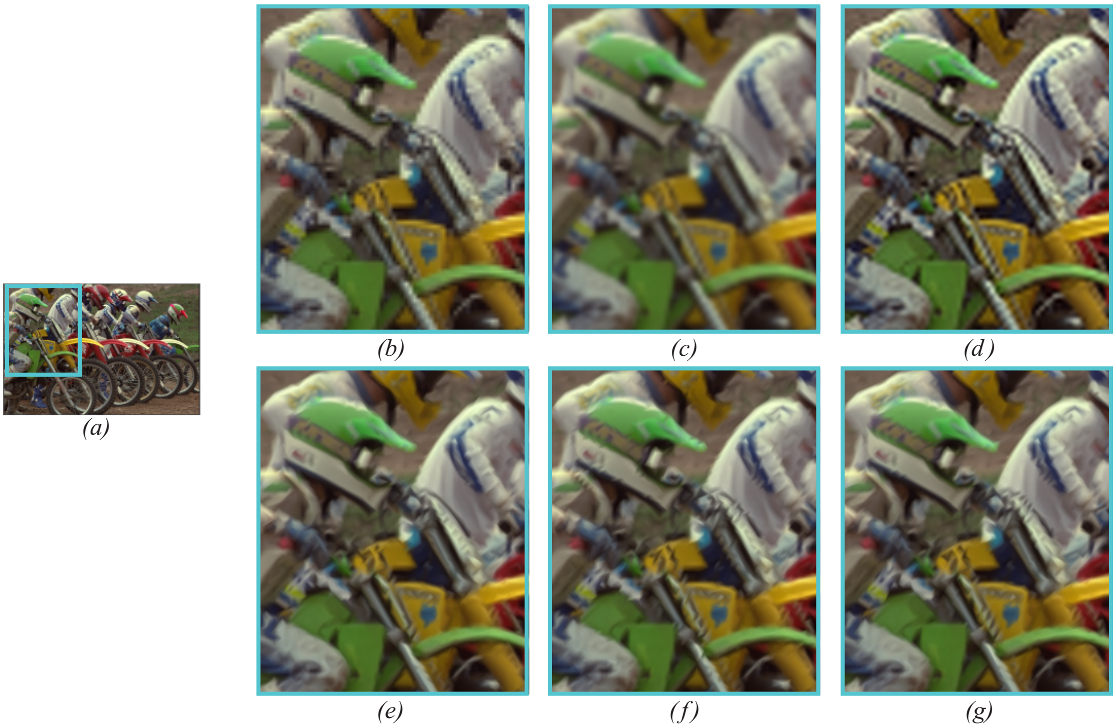
GPU ExpROI MaxGain, *GPU ExpROI2 MaxGain*, *GPU ExpROI3 MaxGain*, *GPU ExpROI4 MaxGain*.

Meranie kvality - *Dataset 2*				
	začiatočná cena	**konečná cena**	**zlepšenie (%)**	**počet preklápaní**
CPU LOP	1549,901	691,641	224,090	17727
***GPU* pôvodná**	1549,901	825,867	187,670	16440
GPU ExpROI	1549,901	795,837	194,751	17089
GPU MaxGain	1549,901	777,825	199,261	14808
GPU ExpROI MaxGain	1549,901	753,367	205,730	15066
GPU ExpROI2 MaxGain	1549,901	754,047	205,544	15109
GPU ExpROI3 MaxGain	1549,901	753,288	205,752	15105
GPU ExpROI4 MaxGain	1549,901	753,445	205,709	15092

Tabuľka 4.8: Priemerné ceny triangulácií pre obrázky z *Dataset 2*.

Hlavnou motiváciou pre paralelizáciu bolo využitie výpočtovej sily *GPU*. Porovnanie výpočtových časov jednotlivých prístupov bude delené do troch časti: *inicializácia*, *optimalizácia*, *kompletizácia*. Inicializácia zahŕňa načítanie obrazu a vytvorenie príslušných dátových štruktúr. V prípade *GPU* je potrebné okrem toho nahrať vytvorené textúry do pamäti grafickej karty. V optimalizačnej fáze prebieha vytváranie lokálne optimálnej triangulácie. V poslednej fáze je triangulácia uložená na disk, konvertovaná do vlastného formátu, v prípade paralelného prístupu je okrem toho potrebné stiahnuť textúry z pamäte *GPU*. Výsledné časy sú zobrazené v Tabuľkách 4.10, 4.11. *Celkovo* značí sumu týchto hodnôt, všetky časy sú v sekundách, a značia priemerné hodnoty pre *Dataset 2* a *Dataset 3*.

Z uvedených výsledkov vyplýva, že paralelný prístup je rýchlejší až o 6–10 krát ako pôvodná sekvenčná technika *CPU LOP* v prípade *Dataset 2*. Pri rekonštrukcii obrázkov z *Dataset 3* (vytvorené vektorovým editorom) je tento rozdiel ešte väčší

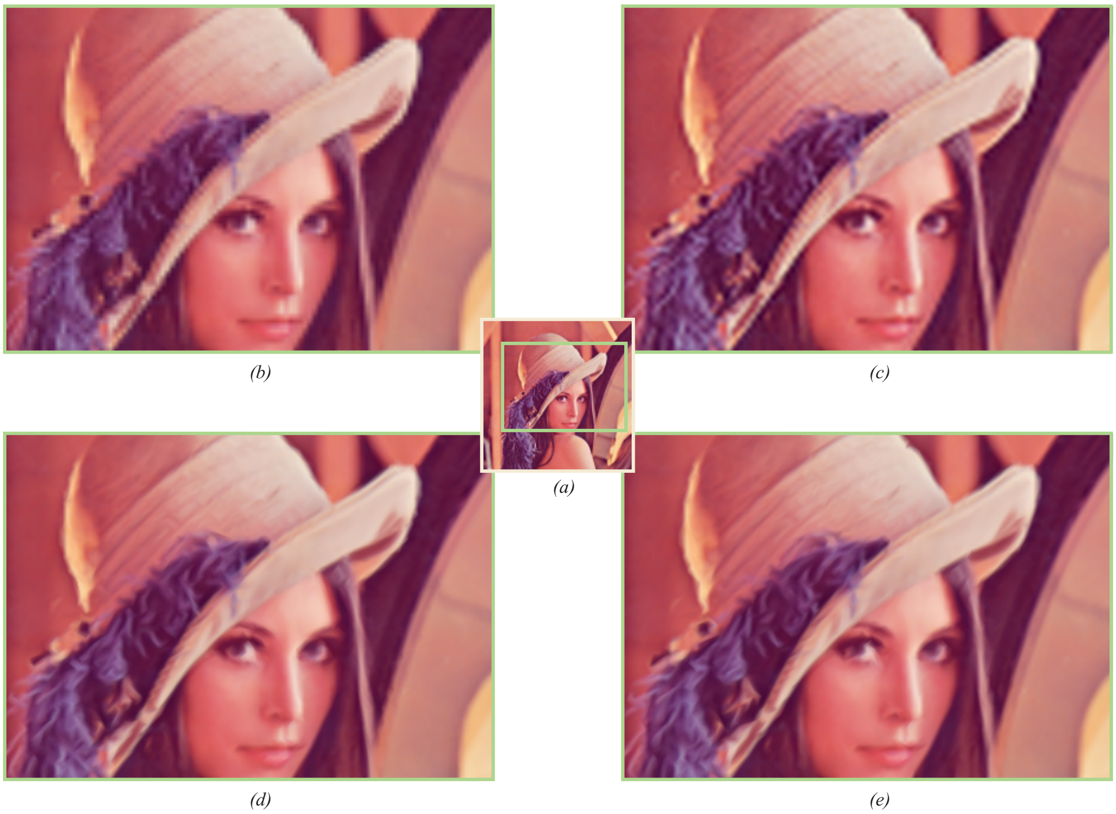


Obr. 4.15: Rekonštrukčné výsledky pri 400% zväčšení (a) zmenšený obrázok, (b) bilineárna interpolácia, (c) b-spline filter, (d) Lanczosov filter , (e) *CPU LOP*, (f) *GPU* pôvodná technika, (g) *GPU ExpROI MaxGain* modifikácia.

(25–30 násobok). Podrobnejšou analýzou sa zistilo, že tento markantný rozdiel bol spôsobený vďaka vysokému počtu iteračných krokov v prípade *CPU LOP*. V posledných iteráciach sa preklápal len malý počet hrán, ktoré potom následne umožnili preklápanie ďalších hrán. Vzhľadom na to, že *Dataset 2* obsahuje dáta z reálneho sveta, je v nich obsiahnutý šum, respektíve štrukturalizácia obrázkov je zložitejšia, a preto sa tento jav neobjavil. Paralelný prístup si vďaka súčasnému preklápaniu hrán lepšie poradil s touto situáciou. Za korektné možno považovať faktor urýchlenia 6 až 10 násobok oproti pôvodnému *LOP* prístupu, vzhľadom na to, že *Dataset 2* je bližšie k reálnym dátam.

Meranie kvality - *Dataset 3*				
	začiatočná cena	**konečná cena**	**zlepšenie (%)**	**počet preklápaní**
CPU LOP	1549,762	341,809	453,400	8626
***GPU* pôvodná**	1549,762	432,785	358,090	7263
GPU ExpROI	1549,762	436,413	355,113	7423
GPU MaxGain	1549,762	420,602	368,463	6696
GPU ExpROI MaxGain	1549,762	365,502	424,009	7054
GPU ExpROI2 MaxGain	1549,762	360,335	430,089	7161
GPU ExpROI3 MaxGain	1549,762	363,982	425,779	7118
GPU ExpROI4 MaxGain	1549,762	363,736	426,068	7106

Tabuľka 4.9: Priemerné ceny triangulácií pre obrázky z *Dataset 3*.



Obr. 4.16: Rekonštrukčné výsledky pri 400% zväčšení (a) zmenšený obrázok, (b) bilineárna interpolácia, (c) Lanczosov filter, (d) *CPU LOP*, (e) *GPU ExpROI2 MaxGain* modifikácia.

Neočakávaným výsledkom je nižší výpočtový čas prístupu *GPU ExpROI MaxGain* v porovnaní s modifikáciami *GPU ExpROI2/3/4 MaxGain*. Očakávalo sa, že prístupy, kde sa používa rozšírená oblasť *ROI* len v prvých pár iteráciách budú vykonané rýchlejšie ako metóda, kde sa rozšírená oblasť používa počas celého priebehu optimalizácie - *GPU ExpROI MaxGain*. Dá sa to vysvetliť tým, že zmena fragment programu je časovo náročná operácia na grafických kartách, a práve po prvých pár iteráciach sa dochádza k takejto zmene v prípade metód *GPU ExpROI2/3/4 MaxGain*.

Na získanie vyššie uvedených výsledkov bola použitá implementácia pomocou jazyku *Cg*, z čoho vyplýva obmedzenie použitia grafických kariet len s *GPU* od výrobcu NVidia. Kvôli tomu vznikla ďalšia implementácia v jazyku *glsl*, kde už neplatí obmedzenie ohľadom výrobcu hardvéru. Na testovanie boli použité všetky obrázky z *Dataset 2* a *Dataset 3*, priemerné hodnoty výpočtových časov sú uvedené v Tabuľke 4.12. Tieto údaje značia celkový čas, v predošlých tabuľkách sú tieto hodnoty označené ako *celkovo*.

Z výsledkov vyplýva, že nie je veľký rozdiel medzi rôznymi implementáciami, pre NVidia GeForce 8800 *GPU* sú tieto odchýlky minimálne. Novšia grafická karta od NVidia, GeForce 9600 *GPU*, produkovala približne o 35% nižšie výpočtové časy.

Meranie rýchlosti výpočtov - *Dataset 2*				
čas (s)	**inicializácia**	**optimalizácia**	**kompletizácia**	**celkovo**
CPU LOP	0,2126	6,8346	0,2216	7,2689
***GPU* pôvodná**	0,4076	0,2526	0,0897	0,7498
GPU ExpROI	0,5183	0,6199	0,0936	1,2318
GPU MaxGain	0,4531	0,2775	0,0872	0,8178
GPU ExpROI MaxGain	0,6028	0,4702	0,0859	1,1589
GPU ExpROI2 MaxGain	0,6017	0,5183	0,0910	1,2109
GPU ExpROI3 MaxGain	0,5990	0,5274	0,0898	1,2163
GPU ExpROI4 MaxGain	0,6013	0,5340	0,0859	1,2212

Tabuľka 4.10: Priemerné hodnoty rekonštrukčných časov pre *Dataset 2*.

Meranie rýchlosti výpočtov - *Dataset 3*				
čas (s)	**inicializácia**	**optimalizácia**	**kompletizácia**	**celkovo**
CPU LOP	0,3159	32,1041	0,3313	32,7513
***GPU* pôvodná**	0,4142	0,3047	0,1457	0,8645
GPU ExpROI	0,5207	0,5262	0,1483	1,1952
GPU MaxGain	0,4348	0,3253	0,1510	0,9112
GPU ExpROI MaxGain	0,6095	0,5362	0,1460	1,2917
GPU ExpROI2 MaxGain	0,6045	0,5702	0,1487	1,3233
GPU ExpROI3 MaxGain	0,6040	0,5758	0,1430	1,3228
GPU ExpROI4 MaxGain	0,6095	0,5702	0,1430	1,3227

Tabuľka 4.11: Priemerné hodnoty rekonštrukčných časov pre *Dataset 3*.

Tretia grafikcá karta, na ktorej sa testovali výsledky je od firmy ATI model Radeon HD 4770. V tomto prípade niektoré výsledky sú lepšie a niektoré výrazne horšie ako v predchádzajúcich prípadoch. Je to spôsobené architektúrou grafickej karty, vďaka čomu niektoré vnútorné operácie majú odlišnú efektívnosť. Ceny výsledných triangulácií sa líšia len nepatrne, a tieto malé zmeny sú spôsobené pravdepodobne vďaka rozlišnej vnútornej presnosti jednotlivých operácií.

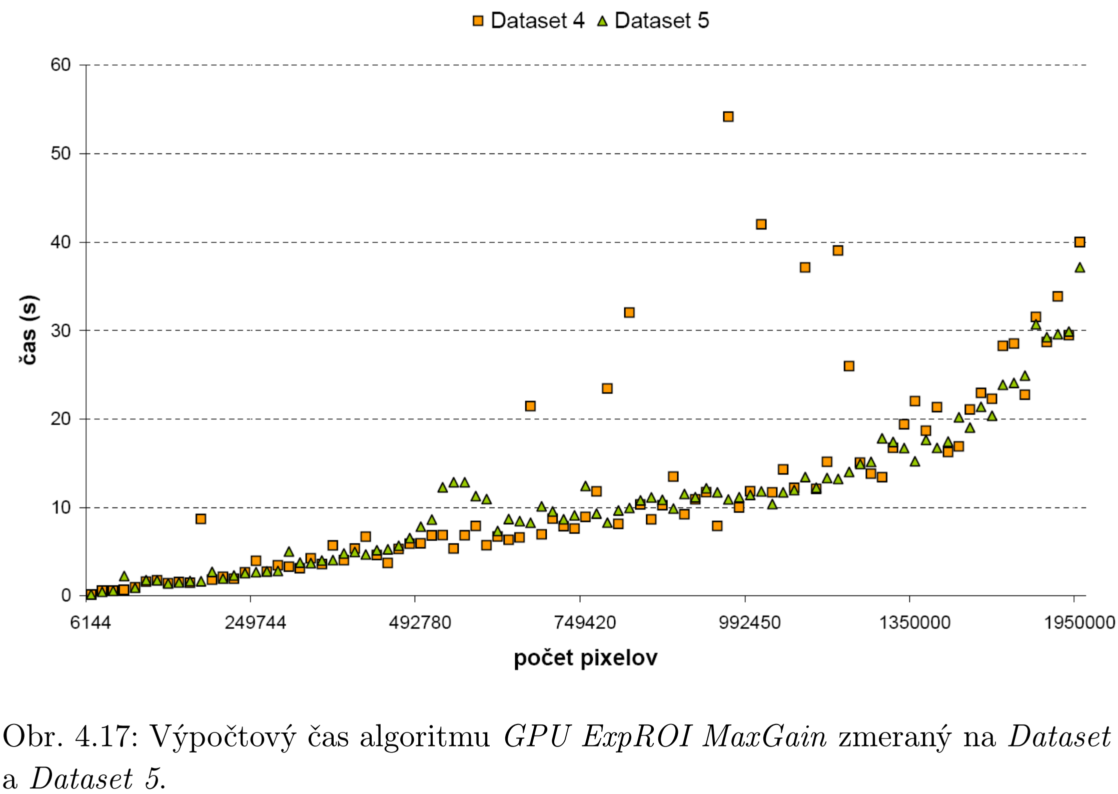
Analýza časovej zložitosti predstavených paralelných metód je závislá od rôznych faktorov, ako napríklad voľba cenovej funkcie alebo distribúcia dát. Tieto parametre závisia od aplikačnej oblasti, a takáto analýza je nad rámec cieľov a možností tejto práce. Analýza časovej zložitosti pre podobný prístup ako akceptovanie a zamietnutie kandidátskych hrán bola predstavená v článku [JPMp92], a môže slúžiť ako

priemerná hodnota	**NVIDIA GeForce 8800GTS**	**NVIDIA GeForce 9600 GT**	**ATI Radeon HD 4770**
***GPU* pôvodná**	0,873	0,574	0,608
GPU ExpROI	1,225	0,885	1,354
GPU MaxGain	0,866	0,551	0,644
GPU ExpROI MaxGain	1,222	0,820	3,962
GPU ExpROI2 MaxGain	1,169	0,741	4,027
GPU ExpROI3 MaxGain	1,184	0,753	4,020
GPU ExpROI4 MaxGain	1,209	0,762	4,015

Tabuľka 4.12: Porovnanie výpočtového času paralelného prístupu na rôznych *GPU*, implementácia pomocou jazyku *glsl*.

štartovací bod pri komplexnej analýze časovej zložitosti.

Na získanie odhadu časovej zložitosti je možné zvoliť aj empirický prístup. Pre tento účel bola vytvorená testovacia sada pozostávajúca z 91 obrázkov, označená ako *Dataset 4*, pričom jednotlivé obrázky mali rôzne rozmery. Na Obrázku 4.17 sú zobrazené výsledné rekonštrukčné časy pre metódu *GPU ExpROI MaxGain*, v závislosti od veľkosti jednotlivých obrázkov. Pre obrázky malej a strednej veľkosti sa dá očakávať lineárne správanie, v prípade obrázkov väčšieho rozmeru očakávaná časová zložitosť nemá lineárnu charakteristiku. Ďalej, dajú sa pozorovať ojedinelé prípady, kde výpočtové časy značne prevýšia hodnoty z ich okolia. Toto bolo spôsobené pravdepodobne nevhodnou distribúciou kandidátskych hrán z hľadiska paralelného spracovania. V porovnaní s početnosťou testovacej množiny množstvo prípadov, kde sa nastala problematická distribúcia, nie je štatisticky signifikantné. Pri overení uvedenej hypotézy sa postupovalo nasledovne. Z *Dataset 4* sa vybral jeden obrázok, ktorý bol následne prevzorkovaný na veľkosti jednotlivých obrázkov z *Dataset 4*. Takto získaná množina obrázkov sa bude označovať ako *Dataset 5*. Rekonštrukčné časy pre túto množinu sú uvedené na Obrázku 4.17. Dosiahnuté časové výsledky sú podobné ako v prípade *Dataset 4* s tým rozdielom, že sa nevyskytujú také prípady, kde by výsledné časy výrazne prevýšili okolité hodnoty.



Obr. 4.17: Výpočtový čas algoritmu *GPU ExpROI MaxGain* zmeraný na *Dataset 4* a *Dataset 5*.

4.7 Zhrnutie

Z hľadiska aplikačnej oblasti rekonštrukcie obrazu je možné konštatovať, že výsledné DDT triangulácie majú lokálny charakter. Teda výpočtovo lacné aproximácie MWT dávajú vizuálne príjemné výsledky. Tieto pozorovania platia aj pre rekonštrukčné úlohy, kde dáta majú charakter Markovových náhodných polí. Z hľadiska rekonštrukcie obrazu využívanie DDT prístupov na magnifikačné účely prináša výhody. V tomto prípade možné odporúčať použitie deterministicky založených metód. Stochastické metódy sú vhodné pre rekonštrukčné úlohy, kde miera aproximácie MWT je dôležitá. Ďalej, je možné konštatovať, že paralelné dátovo závislé prístupy vzhľadom na súčasný vývoj hardvéru sú užitočným nástrojom pri tvorbe lokálne optimálnych triangulácií.

Pri popise porovnávacích kritérií bola vyslovená hypotéza, že nižšia cena triangulácie znamená lepšiu rekonštrukciu. Keď sa porovnávajú výsledky, z Tabuliek 4.2 a 4.3 vidieť, že poradie podľa kvality triangulačných prístupov je rovnaké. Vyplýva z toho potvrdenie tejto hypotézy, a to za predpokladu, že perceptuálne metriky sú objektívnym meradlom kvality.

Kapitola 5

Podnety pre ďalší výskum

Výskumná oblasť dátovo závislých triangulácií je málo preskúmaná. Ponúknuté rozšírenia majú za úlohu ju posunúť bližšie ku praktickému využitiu. Okrem iného bolo predstavené $n-$dimenzionálne rozšírenie tejto techniky, čo otvára priestor pre využitie v mnohých aplikačných oblastiach. Do nasledujúceho zoznamu sú zaradené myšlienky, ktoré môžu byť predmetom ďalšieho výskumu:

— Z teoretického hľadiska zaujímavou výzvou môže byť konštrukcia DT vo vyšších dimenziách pomocou zovšeobecneného Lawsonovho procesu. Úlohou je nájdenie takej množiny topologických transformácií $\boldsymbol{T}_K$ a váhovej funkcie, ktorá by produkovala trianguláciu s požadovanými vlastnosťami.

— Rekonštrukcia v čase sa meniacich volumetrických dát a flow-visualization sú aplikačné oblasti, kde sa dá preveriť správanie DDT v 4D. Motiváciou môže byť, že rekonštrukčné chyby metód zo spracovania obrazu v prípade v čase sa meniacich volumetrických dát sa prejavujú viac ako pri rekonštrukcii statických snímok.

— V časti 3.5 bol predstavený návrh kompresie viacrozmerných dát pomocou DDT. Implementácia a porovnanie výsledkov s inými metódami by mohlo byť ďalším smerovaním.

— V uvedených úvahách umiestnenie vrcholov sa považovalo za fixné. Zmenou pozície vrcholov sa otvára celý rad možností. Napríklad, dalo by sa to využiť na ostrenie hrán v prípade rekonštrukcie obrazu. Znamenalo by to nájdenie obrysov v obraze (ich vektorizáciu), vyšetrovanie tvarov trojuholníkov v obrysových oblastiach a modifikáciu pozícií ich vrcholov. V tomto prípade by bolo potrebné predspracovanie, aké bolo uvedené pri kombinácii konvolučných techník s výsledkami dátovo závislých prístupov.

— Rekonštrukcia videosekvencie je ďalšou oblasťou, kde dátovo závislé techniky môžu nájsť uplatnenie. Motiváciou môže byť napríklad nízke rozlíšenie vyžadujúce zväčšenie. Zaujímavé by bolo skúmanie možnosti využitia triangulácie z predchádzajúcej snímky, ako úvodnú trianguláciu. Takýto prístup by mohol výrazným spôsobom urýchliť rekonštrukciu sekvencií, kde susedné snímky sú veľmi podobné, respektíve pohyb objektov v scéne je pomalý. V takomto prípade by stačilo aplikovať DT ako úvodnú trianguláciu len v kľúčových snímkach (key frames), a v ostatných by sa používal vyššie spomínaný princíp - DDT. Spájanie tejto myšlienky s paralelným spracovaním by mohlo priniesť nízke rekonštrukčné časy. Ako alternatíva k uvedenému prístupu, môže byť rekonštrukcia videosekvencie ako 3D dáta, teda organizácia jednotlivých snímok tak, aby spolu tvorili volumetrické dáta. V tomto prípade ale výpočtový čas by bol oveľa vyšší.

— Podrobná analýza $GLOP$ v 3D je ďalším kandidátom pre budúci výskum. V časti 4.5 bola predstavená len verifikácia tohto prístupu, na základe subjektívneho vnímania. V prípade rekonštrukcie dát z CT by mohli byť použité podobné nástroje a analýzy ako v prípade rekonštrukcie obrazu. Vyžadovalo by to rozšírenie perceptuálnych metrík a rozdielových obrazov do 3D, čo z teoretického hľadiska neznamená problém.

— V časti 3.7 bol predstavený princíp kombinácie výsledkov dátovo závislých metód a konvolučných prístupov. Spôsob kombinácie je riadený pomocou parametrov, ktorých nastavenie nebolo podrobne testované. Takáto analýza by mohla byť prínosom, nielen teoretickým, ale aj praktickým.

— Podrobné skúmanie dátovo závislých prístupov z hľadiska aproximácie MWT. V prípade rekonštrukcie obrazu bolo zistené, že stochastické prístupy dosahovali vyššiu cenu ako deterministické metódy. To otvára diskusiu ohľadom nastavenia parametrov, ktorými sú riadené tieto procesy. Analýza týchto nastavení pre rôzne distribúcie vrcholov by mohla byť náplňou ďalšieho výskumu.

— Prínos dátovo závislých triangulácií môže byť významný aj pre vedeckú oblasť spracovania obrazu, napríklad ich využitie v kombinácií s waveletovými prístupmi, kde sa využívajú triangulácie [Lee02, Tri03].

— Predstavený paralelný dátovo závislý prístup z časti 3.6 bol implementovaný pomocou fragment programu, z čoho vyplĺňajú určité obmedzenia. Vzhľadom na vývoj architektúry grafického hardvéru, množstvo týchto bariér už je odstránených. Efektívnejšie využitie hardvéru umožňujú napríklad *CUDA* [NVi] alebo *OpenCL* [Khr]. Pomocou týchto štandardov by bolo možné navrhnúť efektívnejšie paralelné riešenia. Tento výskumný smer je veľmi perspektívny a dôležitý vzhľadom na predpokladané smerovanie vývoja hardvéru.

Kapitola 6

Záver

Predložená dizertačná práca je venovaná skúmaniu špeciálnej podskupiny optimálnych triangulácií, dátovo závislým trianguláciam. Cieľom bolo sumarizovať poznatky z tejto oblasti, navrhovať nové optimalizačné prístupy a zovšeobecniť myšlienku dátovo závislých triangulácií do vyšších dimenzií.

Prehľad problematiky vrátane popisu základných pojmov, uvedenie dátovo závislých triangulácií a existujúcich optimalizačných techník bol predstavený v druhej kapitole. Snahou bolo priblížiť dátovo závislé triangulácie z hľadiska aproximácie globálne optimálneho stavu *MWT*. Prínos práce sa nachádza v nasledujúcej, tretej kapitole. Z tematického a publikačného hľadiska dosiahnuté výsledky sa dajú deliť do troch skupín.

Prvú skupinu tvoria rozšírenia dátovo závislej techniky o nové metódy pre rovinný prípad. Sem patrí koncepcia rozdelenie na bloky (časť 3.1) pre dáta organizované do karteziánskej mriežky, alebo stochastická optimalizačná technika založená na kombinácii simulovaného žíhania a look-ahead prístupu (časť 3.2). Ďalej, bola predstavená nová skupina triangulačných metód, pomenované ako *Quasi-DDT* (časť 3.3), ktorá slúži skôr k praktickému použitiu a nie ako aproximácia *MWT*. Časť týchto metód bola publikovaná v práci [Tót06].

Druhú skupinu tvoria prístupy zaoberajúce sa viacrozmernými priestormi. Jedným z hlavných prínosov práce je rozšírenie dátovo závislých triangulácií do vyšších dimenzií cez minimalizáciu váhovaného objemu, prezentované v časti 3.4. Pre tento prístup bola predstavená aj váhová funkcia v odseku 3.4.2. Ako príklad optimalizačného prístupu bolo uvedené zovšeobecnenie Lawsonovho optimalizačného procesu v odseku 3.4.3. Podobne by bolo možné zovšeobecniť aj iné optimalizačné metódy. Princíp využitia týchto myšlienok pre účely kompresie viacrozmerných dát bol popísaný v časti 3.5. Výsledky, okrem koncepcie dátovo závislej kompresie, boli publikované v [TVFG07].

Poslednú skupinu tvoria pragmatické riešenia, ktoré majú za účel posúvať dátovo závislé techniky bližšie k reálnemu použitiu, pre aplikačnú oblasť rekonštrukcie obrazu. Do tejto skupiny patrí ďalší významný prínos práce, paralelný výpočet dátovo závislých triangulácií pomocou grafického hardvéru, popísané v časti 3.6. Táto metóda je všeobecná a možné ju použiť pri vytváraní ľubovoľnej lokálne optimálnej triangulácie. Pre aplikačnú oblasť rekonštrukcie obrazu bola predstavená metóda, ktorá kombinuje rekonštrukčné výsledky konvolučných techník a dátovo závislých metód, nachádza sa to v časti 3.7. Publikácia paralelného prístupu je akceptovaná do časopisu *Computers & Graphics* [CTS*].

Detailná analýza implementovaných metód z hľadiska rôznych aspektov je uvedená v štvrtej kapitole. V prípade rovinných metód za aplikačnú oblasť bola zvolená rekonštrukcia obrazu, a boli použité štandardné testovacie obrázky zo spracovania obrazu. Podnety pre ďalší výskum boli predstavené v piatej kapitole, nachádzajú sa tu myšlienky, ktoré podľa subjektívneho odhadu by sa dali dotiahnuť do publikačnej podoby. Uvedená výskumná činnosť bola podporovaná z viacerých grantov, ktoré sú uvedené v Dodatku A.

Okrem uvedených rozšírení a nových prístupov bolo otázkou aj reálna použiteľnosť dátovo závislých metód v praxi. Testy v tomto smere boli vykonané pre oblasť rekonštrukcie obrazu, konkrétne pre operáciu zväčšenia obrazu. Z hľadiska množstva rekonštrukčných chýb dátovo závislé prístupy dávajú porovnateľné výsledky ako najlepšie konvolučné techniky. Ďalším aspektom vnímania kvality je distribúcia týchto chýb, kde dátovo závislé prístupy dávajú vizuálne prijateľnejšie výsledky. Vplyv distribúcie artefaktov na kvalitu je ale otázkou subjektívneho vnímania, ktorú je ťažké hodnotiť matematickými meradlami. Práve preto, ako objektívny záver je možné konštatovať, že dátovo závislé techniky sú vhodnou alternatívou ku konvolučným metódam. Napriek tomu ich rozšírenosť v oblasti počítačovej grafiky je malá, vzhľadom na výsledky ktoré dosahujú. Cieľom tejto práce bolo aj poukázanie na tento fakt, a šíriť túto myšlienku vo vedeckej komunite.

Na začiatku 4. kapitoly v odseku 4.1.1 bola stanovená hypotéza, že pomocou ideálnej voľby cenovej funkcie a nájdením triangulácie s minimálnou váhou je kvalita výsledku rekonštrukcie maximalizovaná. Preverenie tejto hypotézy pre oblasť rekonštrukcie obrazu bola prezentovaná v kapitole štyri, v analýze výsledkov. Tu sa preukázalo, že poklesom ceny triangulácie rastie kvalita rekonštrukčných výsledkov, na meranie kvality sa pritom používali perceptuálne metriky. Je to pozoruhodný výsledok, vzhľadom na to, že ideálnu cenovú funkciu pre túto oblasť nie je možné zostrojiť. Ďalším obmedzením je dokázaná NP-ťažkosť úlohy hľadania *MWT* triangulácie. To môže slúžiť ako ďalšia motivácia pri hľadaní aproximácií či návrhu stochastických procesov.

Literatúra

[AKTD99] ALBOUL L., KLOOSTREMAN G., TRAAS C., DAMME R. V.: *Best data-dependent triangulations.* Tech. Rep. TR-1487-99, 1999.

[Aur91] AURENHAMMER F.: Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys 23*, 3 (1991), 345–405.

[BE95] BERN M., EPPSTEIN D.: Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, second ed., no. 4. World Scientific, 1995, pp. 47–123.

[Bes01] BESPAMYATNIKH S. N.: Enumerating triangulations of convex polytopes. In *Proc. Conf. Discrete Models: Combinatorics, Computation, & Geometry (DM-CCG 2001)* (2001), Cori R., Mazoyer J., Morvan M.„ Mosseri R., (Eds.), no. AA in Discrete Mathematics & Theoretical Computer Science Proceedings, pp. 111–122.

[BGM04] BATTIATO L., GALLO G., MESSINA G.: SVG rendering of real images using data dependent triangulation. In *In Proceedings of Spring Conference on Computer Graphics 2004* (2004), pp. 191–198.

[BHJ99] BARNES J. C., HAMANN B., JOY K. I.: *An edge-preserving, data-dependent triangulation scheme for hierarchical rendering.* Scientific Visualization - Methods and Applications. Springer-Verlag, 1999, pp. 1–10.

[Bjo04] BJORKE K.: High-quality filtering. *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics* (2004), 391–415.

[Bro91] BROWN J.: Vertex based data dependent triangulations. *Computer Aided Geometric Design 8*, 3 (1991), 239–251.

[BY05] BOISSONNAT J.-D., YVINEC M.: *Algorithmic Geometry.* Cambridge University Press, ISBN 0-521-56529-4, 2005.

[Cec02] CECH P.: Perceptual metrics and image compression. In *Proceedings of Spring Conference on Computer Graphics* (2002), pp. 39–40.

[CGA] CGAL: *Developers manual*, CGAL 3.1 ed. http://www.cgal.org. Webpage accessed at November 2004.

[CKS96] CHENG S.-W., KATOH N., SUGAI M.: A study of the LMT-skeleton. In *ISAAC '96: Proceedings of the 7th International Symposium on Algorithms and Computation* (1996), Springer-Verlag, pp. 256–265.

[CTS*] CERVEŇANSKÝ M., TÓTH Z., STARINSKÝ J., FERKO A., ŠRÁMEK M.: Parallel GPU-based data-dependent triangulations. *Computers&Graphics*. accepted at 9th January 2010.

[Dan80] DANIELSSON P.-E.: Euclidean distance mapping. *Computer Graphics and Image Processing 14* (1980), 227–248.

[DLR90] DYN N., LEVIN D., RIPPA S.: Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10 (1990), 137–154.

[Dwy86] DWYER R.: A simple divide-and-conquer algorithm for computing Delaunay triangulations in $\mathcal{O}(nloglogn)$ expected time. In *SCG '86: Proceedings of the second annual symposium on Computational geometry* (New York, NY, USA, 1986), ACM, pp. 276–284.

[Ede00] EDELSBRUNNER H.: Triangulations and meshes in computational geometry. In *Acta Numerica* (2000), pp. 133–213.

[EPW90] EDELSBRUNNER H., PREPARATA F. P., WEST D. B.: Tetrahedrizing point sets in three dimensions. *J. Symb. Comput. 10*, 3-4 (1990), 335–347.

[ES92] EDELSBRUNNER H., SHAH N. R.: Incremental topological flipping works for regular triangulations. In *SCG '92: Proceedings of the eighth annual symposium on Computational geometry* (1992), ACM Press, pp. 43–52.

[ETW92] EDELSBRUNNER H., TAN T., WAUPOTITSCH R.: A polynomial time algorithm for the minmax angle triangulation. *SIAM J. Sci. Stat. Comp. 13* (1992), 994–1008.

[Fer04] FERKO A.: Solving the minimum weight triangulation problem. Habilitation Lecture Notes, Comenius University. Bratislava, 2004.

[FG06] FISHER I., GOTSMAN C.: Fast approximation of high order Voronoi diagrams and distance transforms on the GPU. *Journal of Graphics Tools 11* (2006), 39–60.

[FK03] FERNANDO R., KILGARD M. J.: *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. Addison-Wesley Professional, 2003.

[Gla95] GLASSNER A.: *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995.

[GW06] GONZALEZ R. C., WOODS R. E.: *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[HCK*99] HOFF K. E., CULVER T., KEYSER J., LIN M., MANOCHA D.: Fast computation of generalized Voronoi diagrams using graphics hardware. *Proceedings of ACM SIGGRAPH* (1999), 277–286.

[HD06] HJELLE O., DAHLEN M.: *Triangulations and Applications, Series: Mathematics and Visualization*. Springer, 2006.

[HOS96] HANKE S., OTTMANN T., SCHUIERER S.: The edge-flipping distance of triangulations. *Journal of Universal Computer Science 2*, 8 (1996), 570–579.

[IA04] ISKE A., ARNOLD V. I.: *Multiresolution Methods in Scattered Data Modelling*. Springer Verlag, 2004.

[Ima] IMAGEMAGICK:. http://www.imagemagick.org. Webpage accessed at September 2009.

[Int] INTEL COORPORATION: *OpenCV: Open Computer Vision*. http://www.intel.com/technology/computing/opencv/overview.htm. Webpage accessed at February 2006.

[Joe91] JOE B.: Construction of three-dimensional Delaunay triangulations using local transformations. *Comput. Aided Geom. Des. 8*, 2 (1991), 123–142.

[JPMp92] JONES M. T., PLASSMANN P. E., MCS-P P.: A parallel graph coloring heuristic. *SIAM J. Sci. Comput 14* (1992), 654–669.

[KE00] KRAUS M., ERTL T.: Simplification of nonconvex tetrahedral meshes. In *Hierarchical and Geometrical Methods in Scientific Visualization* (2000), Springer-Verlag, pp. 185–196.

[KF01] KOLINGEROVÁ I., FERKO A.: Multicriteria-optimized triangulations. *The Visual Computer 17*, 6 (2001), 380–395.

[KGV83] KIRKPATRICK S., GELATT C. D., VECCHI M. P.: Optimization by simulated annealing. *Science, Number 4598, 13 May 1983 220, 4598* (1983), 671–680.

[KH01] KREYLOS O., HAMANN B.: On simulated annealing and the construction of linear spline approximations for scattered data. *IEEE Transactions on Visualization and Computer Graphics 7*, 1 (2001), 17–31.

[Khr] KHRONOS GROUP: *The OpenCL specification.* http://www.khronos.org/opencl/. Webpage accessed at October 2009.

[Kir80] KIRKPATRICK D. G.: A note on Delaunay and optimal triangulations. *Inform. Process. Lett.*, 10 (1980), 127–128.

[Kit] KITWARE INC.: *Documentation*, VTK 5.0 ed. http://public.kitware.com/VTK/. Webpage accessed at March 2006.

[Kol99] KOLINGEROVÁ I.: Genetic approach to data dependent triangulations. In *Proceedings of Spring Conference on Computer Graphics* (1999), pp. 229–238.

[Kol04] KOLINGEROVÁ I.: Lookahead search to the minimum weight triangulation. Manuscript, 2004.

[Kre] KREYLOS O.: Approximation examples - Lena. http://graphics.cs.ucdavis.edu/~okreylos/ResDev/SplineApproximation/Examples/index.html. Webpage accessed at December 2009.

[KZ01] KOLINGEROVÁ I., ZALIK B.: Improvements to randomized incremental Delaunay insertion. *Computers & Graphics*, 26 (2001), 477–490.

[Law77] LAWSON C. L.: Software for c^1 surface interpolation. In *Mathematical Software III, J. Rice ed.* (1977), Academic Press, pp. 161–194.

[Lee00] LEE K.: Three-dimensional medical image modeling of scattered data based on data-dependent criteria. In *Proc. SPIE Vol. 4117, Vision Geometry IX* (2000), pp. 91–99.

[Lee02] LEE S. M.: *Wavelet-Based Multiresolution Surface Approximation from Height Fields.* Dissertation thesis, Virginia Polytechnic Institute and State University, 2002.

[LHK*04] LUEBKE D., HARRIS M., KRÜGER J., PURCELL T., GOVINDARAJU N., BUCK I., WOOLLEY C., LEFOHN A.: GPGPU: general purpose computation on graphics hardware. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes* (2004), ACM, p. 33.

[LZ99] LAMOT M., ZALIK B.: An overview of triangulation algorithms for simple polygons. In *IV '99: Proceedings of the 1999 International Conference on Information Visualisation* (Washington, DC, USA, 1999), IEEE Computer Society, pp. 153–158.

[MDM04] MARCHESIN S., DISCHLER J. M., MONGENET C.: 3D ROAM for scalable volume visualization. In *2004 IEEE Symposium on Volume Visualization and Graphics (VV'04)* (2004), IEEE Computer Society, pp. 79–86.

[MJ02] MAHMOOD R., JIMACK P. K.: Locally optimal unstructured finite element meshes in three dimensions. In *ICECT'03: Proceedings of the third international conference on Engineering computational technology* (Edinburgh, UK, UK, 2002), Civil-Comp press, pp. 31–32.

[MRR*53] METROPOLIS N., ROSENBLUTH A., ROSENBLUTH M., TELLER A., TELLER E.: Equations of state calculations by fast computing machines. *Journal of Chemical Physics 21*, 6 (1953), 1087–1092.

[MTG04] MÜLLER M., TESCHNER M., GROSS M.: Physically-based simulation of objects represented by surface meshes. In *Proceedings of Computer Graphics International CGI'04* (2004), pp. 26–33.

[Mul06] MULZER W.: Minimum weight triangulation is NP-hard. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry* (New York, NY, USA, 2006), ACM Press, pp. 1–10.

[Nie93] NIELSON G. M.: Scattered data modeling. *IEEE Comput. Graph. Appl. 13*, 1 (1993), 60–70.

[Noc02] NOCIAR M.: *Triangulácie v rovine a teréne.* Master's thesis, Comenius University, Bratislava, 2002.

[NVi] NVIDIA: *CUDA: Compute Unified Device Architecture Programming Guide, Version 2.0.* http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf. Webpage accessed at July 2008.

[PL89] POTOCKÝ R., LAMOŠ F.: *Pravdepodobnosť a matematická štatistika.* Alfa, Bratislava, Slovensko, 1989.

[Pol00] POLEC J.: *Vybrané metódy kompresie dát.* Univerzita Komenského, 2000.

[PS85] PREPARATA F. P., SHAMOS M. L.: *Computational geometry: an introduction.* Springer-Verlag, 1985.

[Rip90] RIPPA S.: Minimal roughness property of the Delaunay triangulation. *Comput. Aided Geom. Des. 7*, 6 (1990), 489–497.

[RN00] ROXBOROUGH T., NIELSON G. M.: Tetrahedron based, least squares, progressive volume models with application to freehand ultrasound data. In *VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)* (2000), IEEE Computer Society.

[Ros05] ROST R. J.: *OpenGL(R) Shading Language (2nd Edition).* Addison-Wesley Professional, 2005.

[RT06] RONG G., TAN T.-S.: Jump flooding in GPU with applications to Voronoi diagram and distance transform. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2006), ACM Press, pp. 109–116.

[RTCS08] RONG G., TAN T.-S., CAO T.-T., STEPHANUS: Computing two-dimensional Delaunay triangulation using graphics hardware. In *SI3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games* (2008), ACM Press, pp. 89–97.

[Sch28] SCHÖNHARDT E.: Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Math. Annalen*, 98 (1928), 309–312.

[Sch93] SCHUMAKER L. L.: Computing optimal triangulations using simulated annealing. In *Selected papers of the international symposium on Free-form curves and free-form surfaces* (1993), Elsevier Science Publishers B. V., pp. 329–345.

[She02] SHEWCHUK J. R.: Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. Unpublished manuscript, 2002.

[SW03] SU D., WILLIS P.: Demosaicing of colour images using pixel level data-dependent triangulation. In *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003* (2003), IEEE Computer Society, p. 16.

[SW04] SU D., WILLIS P.: Image interpolation by pixel-level data-dependent triangulation. *Comput. Graph. Forum 23*, 2 (2004), 189–202.

[Tri03] TRISIRIPISAL P.: *Image approximation using triangulation.* Master of science thesis, Virginia Polytechnic Institute and State University, 2003.

[Tót04] TÓTH Z.: *Rekonštrukcia obrazu pomocou triangulácie.* Master's thesis, Comenius University, Bratislava, 2004.

[Tót06] TÓTH Z.: Towards an optimal texture reconstruction. In *CESCG 2000-2005 Best Papers Selection* (2006), Wimmer M., Ferko A., Szirmay-Kalos L.„ Hauser H., (Eds.), pp. 197–212.

[TVFG07] TÓTH Z., VIOLA I., FERKO A., GRÖLLER M. E.: N-dimensional data-dependent reconstruction using topological changes. In *Topology-based Methods in Visualization* (2007), Hauser H., Hagen H.„ Theisel H., (Eds.), Springer, pp. 183–198.

[vBSF05] ŽÁRA J., BENEŠ B., SOCHOR J., FELKEL P.: *Moderní počítačová grafika 2. vydání.* Praha: Computer Press, 2005.

[Č85] ČERNY V.: A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications 45*, 1 (1985), 41–52.

[Vio02] VIOLA I.: *Applications of Hardware-Accelerated Filtering in Computer Graphics.* M.sc. thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Apr. 2002.

[WACBS04] WANG Z., A. C. BOVIK H. R. S., SIMONICELLI E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612.

[WB02] WANG A., BOVIK A.: A universal image quality index. *IEEE Signal Processing Letters 9*, 3 (2002), 81–84.

[Wen05] WENDLAND H.: *Scattered data approximation.* Cambridge University Press, 2005.

[Win05] WINKLER S.: *Digital Video Quality: Vision Models and Metrics.* Wiley, 2005.

[WWT*98] WEIMER H., WARREN J., TROUTNER J., WIGGINS W., SHROUT J.: Efficient co-triangulation of large data sets. In *IEEE Visualization 98* (1998), pp. 119–126.

[YBS01] Yu X., Bryan B. S., Sederberg T. W.: Image reconstruction using data-dependent triangulation. *Computer Graphics and Applications 21*, 3 (2001), 62–68.

Dodatok A

Vedecké aktivity

Publikácie

Červeňanský M., Tóth Z., Starinský J., Ferko A., Šrámek M.: Parallel GPU-based data-dependent triangulations. *Computers & Graphics.* Prijaté v januári 2010.

Tóth Z., Viola I., Ferko A., Gröller M. E.: N-dimensional data dependent reconstruction using topological changes. In Topology-based Methods in Visualization (2007), Hauser H., Hagen H., Theisel H., (Eds.), *Springer*, ISBN 978-3-540-70822-3, pp. 183–198.

Tóth Z.: Towards an optimal texture reconstruction. In CESCG 2000-2005 Best Papers Selection (2006), Wimmer M., Ferko A., Szirmay-Kalos L., Hauser H., (Eds.), *Österreichische Computer Gesellschaft Wien*, ISBN 3-85403-204-8, pp. 197–212.

Tóth Z.: Illustration of data dependent triangulation reconstruction technique, *Animations of Computer Graphics exhibition in Proceedings of Spring Conference on Computer Graphics* (2006), Častá-Papiernička, ISSN 1335-5694, pp. 104.

Tóth Z. *Image reconstruction using triangulations. Študentská vedecká konferencia FMFI UK 2004* (Bratislava, SR - Brno, CZ), 2004.

Kvalifikačné práce

Tóth Z.: Triangulácie v rovine, teréne a priestore. *Práca k dizertačnej skúške*, Univerzita Komenského, Bratislava SR, 2006.

Tóth Z.: Dátovo závislé triangulácie. *Rigorózna práca*, Univerzita Komenského, Bratislava SR, 2005.

Tóth Z.: Rekonštrukcia obrazu pomocou triangulácie. *Diplomová práca*, Univerzita Komenského, Bratislava SR, 2004.

Ocenenia

Štipendium ESF: Skvalitnenie vedeckej práce doktorandov a postdokov na FMFI UK, 2006

Cena Rektora Univerzity Komenského za najlepšiu diplomovú prácu v odbore matematika

2. miesto ŠVOČ 2004 Česko-Slovenské záverečné kolo - v sekcii Teoretická Informatika

Laureat Študentskej vedeckej konferencie FMFI UK (za prácu Image Reconstruction Using Triangulations) v školskom roku 2003/2004

2nd Best Paper Award (za článok Towards an Optimal Texture Reconstruction, CESCG 2004)

Tosiyasu Lawrence KUNII Award 2004

Zahraničné pobyty

December 2004 - Február 2005, September 2005 Austria - Vienna, Institute of Computer Graphics and Algorithms, Technische Universitat Wien.

Spoluorganizátor podujatí

ASO Workshop on Natural Phenomena Visualization using Unstructured Grid 2005, aso.sccg.sk

TopoInVis Workshop 2005, http://www.vrvis.at/topo-in-vis/2005/

Grantová činnosť

APVV - Tools for processing and visualization of tomographic and confocal data - APVV-20-056105, (2007)

Grant Univerzity Komenského - Dátovo závislá kompresia tetrahedrálnych sietí - UK/392/2007, (2007)

Grant Univerzity Komenského - Rekonštrukcia viacrozmerných dát pomocou triangulácie - UK/362/2006, (2006)

VEGA - Výpočtová geometria pre real-time rendering - VEGA grant no. 1/3083/06, (2006-2007)

ASO sustainable cooperation grant - Natural Phenomena Visualization using Unstructured Grid - (2005), http://aso.sccg.sk

APVT - Virtuálna Bratislava - APVT-20-025502, (2002 - 2004)

Domovská stránka dizertačného projektu

http://dmpc.dbp.fmph.uniba.sk/∼zsolt/res.html