

1 Obsah

1 OBSAH	1
2 ÚVOD	2
3 MODEL GTOPO30	2
3.1 CO JE TO MODEL GTOPO30	2
3.2 CHARAKTERISTIKA DAT GTOPO30.....	3
3.3 POPIS BALÍKU DAT	6
3.4 PŘESNOST MODELU GTOPO30.....	7
3.5 CO JSME POUŽILI.....	7
4 PROSTŘEDÍ MVE	7
4.1 ZÁKLADNÍ CHARAKTERISTIKY	7
4.2 POUŽITÉ MVE	8
4.3 MODULY MVE	8
5 IMPORT GTOPO30 DAT	9
5.1 VSTUPNÍ PARAMETRY	9
5.2 ČTENÍ Z-OVÝCH SOUŘADNIC	11
5.3 URČENÍ NORMÁLY V BODĚ	11
5.4 INDEXOVÁNÍ TROJÚHELNÍKŮ	12
5.5 VÝPOČET NORMÁLY TROJÚHELNÍKU	13
6 IMPLEMENTACE MODULU GLOBELOADER	14
6.1 POUŽITÉ DATOVÉ TYPY	14
6.2 STRUKTURA MODULU	18
7 POUŽITÍ MODULU GLOBE LOADER	20
7.1 MODUL V MVE EDITORU	20
7.2 SPOJENÍ MODULU GLOBELOADER A DECIMATION	21
8 VÝSLEDKY	22
8.1 GLOBE LOADER S MODULEM RENDERER	22
8.2 GLOBE LOADER S MODULEM DECIMACE.....	26
9 PODĚKOVÁNÍ	31
10 ZÁVĚR	31
11 LITERATURA	33
PŘÍLOHA A	34

2 Úvod

Už velice dlouhou dobu provází člověka touha poznávat nepoznané a to již poznané nějakým způsobem zaznamenat, aby se k tomu mohl kdykoliv vrátit. Obě tyto touhy dohromady stáli u zrodu první mapy, kterou kdy člověk udělal. Už od nepaměti se mapy vyvíjí, zpřesňují, vkládá se do nich více a více informací, ale přesto se mapy dlouhou dobu zobrazovali jen ve dvou rozměrech. Jako by člověk zapomněl, že existuje ještě minimálně rozměr třetí. Prvním takový průlomem do třetího rozměru v kartografii se stal globus. Globus má však tu nevýhodu, že není vidět věrný reliéf krajiny. Je samozřejmě možné reliéf s určitou přesností vymodelovat, ale je to příliš pracné a příliš málo detailní. Tento problém však odstranil příchod počítačů. Vzhledem k velkému nasazení milionů pracovníků z oboru výpočetní technika je dnes možné s velkou přesností popsat a vymodelovat v počítači napodobeninu Země i s jejím reliéfem. Díky spolupráci výpočetních a vesmírných středisek se dnes také změnil způsob získávání údajů o naší planetě. Důkazem je tomu soubor informací použitý pro tuto bakalářskou práci, který vznikl na několika pracovištích po celém světě.

3 Model GTOPO30

3.1 Co je to model GTOPO30

GTOPO30 je digitální výškový model Země. Tento model je výsledkem práce U.S. Geological Survey's EROS Data Center v Sioux Falls v severní Dakotě v USA. Model byl získán výpočtem průměrné nadmořské výšky v pravidelném úhlovém krokování po 30 sekundách, což je přibližně 1 km. Jeden bod reprezentuje tedy průměrnou nadmořskou výšku jednoho kilometru čtverečního. GTOPO30 je poměrně starý model, který byl poprvé veřejně publikován v roce 1993. Od té doby byly některé lokality znovu nasnímány, upraveny a zasazeny zpět do modelu GTOPO30. Tato inovovaná verze je nyní poskytována pro potřeby široké veřejnosti na internetových stránkách NASA institutu v USA a mohla posloužit jako vstupní data pro tuto bakalářskou práci.

3.2 Charakteristika dat GTOPO30

GTOPO30 je model, který pokrývá celý zemský povrch od 180° západní do 180° východní délky a od 90° jižní do 90° severní šířky. Krokování úhlu zemské elevace je 30 sekund v obou směrech, což nám dává model o celkových rozměrech 43,200 sloupců a 21,600 řádků. Každý bod reprezentuje určitou nadmořskou výšku, a to od -407 do 8,752 metrů. Body moří a oceánů mají symbolickou hodnotu -9999 metrů, což slouží k jednoduché eliminaci povrchu pevnin od vodní plochy. Veškerá pevnina n hranici s vodní plochou má nadmořskou výšku aspoň 1 m, což v praxi znamená, že i v případě změny hodnoty vodní hladiny na 0 m, rozeznáme obrys pevniny. V důsledku krokování modelu po 30 sekundách (přibližně 1 km v obou směrech) se podle očekávání nezobrazí ostrůvky o rozloze menší než je jeden výškový bod.

Elektronická podoba GTOPO30, s kterou pracujeme je rozdělena na 33 částí. Povrch od 90° severní do 60° jižní šířky má ekvidistantní dělení po 40° ve směru zemské délky a po 50° zemské šířky (počítáno od 180° z.d. a 90° s.š.). Druhá část od 60° do 90° jižní šířky je dělena po 60° ve směru délky a není už dále dělena ve směru šířky. V balíku dat poskytované na internetu je ještě zvlášť část pokrývající celou oblast Antarktidy. Jednotlivé části jsou nazvány podle příslušné pozice levého horního rohu oblasti (např. w180n90 je první oblast s levým horním rohem ležícím na 180° z.d. a 90° s.š.).

Rozlišení jednotlivých oblastí je jednoduše odvoditelné a je to 4,800 sloupců a 6,000 řádků pro oblasti nad 60° j.š. a 7,200 sloupců a 3,600 řádek pro oblasti ležící pod touto hranicí. Sousedící části se nepřekrývají, takže jsou v konečném důsledku velice snadno napojitelné.

Na internetových stránkách tohoto projektu naleznete pro každou oblast archiv složený z 8 souborů. Všechny nesou příslušná jména oblasti (viz. výše) a liší se od sebe pouze koncovkou.

DEM	soubor výškových dat
HDR	hlavičkový soubor pro DEM
DMW	soubor světu
STX	soubor statistik
PRJ	soubor informací o projekci
GIF	vystínovaný obrázek oblasti
SRC	soubor zdrojů informací
SCH	hlavičkový soubor pro SRC

Tab. 1 Soubory v balíku dat

Soubor DEM

V souboru s příponou DEM je uložen výškový model oblasti, a to tak, že jeden výškový bod je reprezentován hodnotou ve formátu znaménkového integeru. Dlužno podotknout, že data jsou uložena ve smyslu MSB-LSB, tedy jinak než jsme na to u počítačů PC zvyklí. Data jsou uložena v souboru po řádcích tzn. celý první řádek, následovaný druhým řádkem...

Soubor HDR

Hlavičkový soubor pro DEM je v ASCII textové podobě a obsahuje informace o oblasti. Setkáme se zde s následujícími symboly

BYTEORDER	pořadí bytů u jednoho výškového bodu
LAYOUT	organizace složek v souboru (pro DEM není důležité)
NROWS	počet řádků
NCOLS	počet sloupců
NBANDS	počet složek v souboru
NBITS	počet bitů na bod
BANDROWBYTES	počet bytů v řádce a složce
TOTALROWBYTES	celkový počet bytů v řádce
BANDGAPBYTE	počet bytů mezi jednotlivými složkami
NODATA	hodnota určena pro maskovací účely
ULXMAP	zemská délka středu levého horního rohu
ULYMAP	zemská šířka středu levého horního rohu
XDIM	x-ový rozměr jednoho bodu (úhel)
YDIM	y-ový rozměr jednoho bodu

Tab. 2 Popis souboru .HDR

Soubor DMW

Soubor DMW je ASCII textový soubor a obsahuje informace o souřadnicích oblasti. Je zde jen pro potřebu určitých geografických systémů a je velice podobný (po informativní stránce) souboru HDR.

Soubor STX

V tomto souboru jsou jen informativní statistická data. Čísla udávají postupně počet složek, minimální výšku, maximální výšku, průměrnou hodnotu a odchylku bodů. Je opět v ASCII text podobě.

Soubor PRJ

Informace o projekci naleznete v tomto souboru. Je v ASCII textové podobě a ukážeme si jej na příkladu.

Projection	GEOGRAPHIC
Datum	WGS84
Zunits	METERS
Units	DD
Spheroid	WGS84
Xshift	0.0000000000
Yshift	0.0000000000
Parameters	

Tab. 3 Popis souboru .PRJ

Soubor GIF

Stínovaný obrázek je v balíku kvůli orientaci. Byl získán z původního projektu s úhlem pootočení 240 sekund a jeho rozlišení je 600 x 750 bodů pro oblasti nad 60° j.š. a 900 x 450 bodů pro oblasti pod touto hranicí.

Soubor SRC

Je to binární soubor, který má strukturu podobnou jako DEM, ale jeden bod zde má velikost 1 byte (DEM má 2 byty). Hodnota bodu znamená, kterého zdroje bylo použito pro DEM GTOPO30.

0	Ocean
1	Digital Terrain Elevation Data
2	Digital Chart of the World
3	USGS 1-degree DEM's
4	Army Map Service 1:1,000,000-scale maps
5	International Map of the World 1:1,000,000-scale maps
6	Peru 1:1,000,000-scale map
7	New Zealand DEM
8	Antarctic Digital Database

Tab. 4 Popis zdrojů jednotlivých map

Soubor SCH

Hlavičkový soubor je ASCII textový soubor. Je velice podobný hlavičkovému souboru pro DEM soubor, a proto jej nebudeme dále popisovat.

3.3 Popis balíku dat

Veškerá data jsou rozdělena podle výše uvedeného schématu na 33 částí, z nichž každá obsahuje 8 souborů. Tyto soubory jsou uloženy v archivech jak na přiloženém CD tak na internetových stránkách NASA institutu. Archivy jsou dvojnásobně komprimované, a to nejprve TAR komprimací a následně GZIP komprimací.

Velikosti rozbalených DEM souborů, tedy souborů, jenž my budeme využívat jsou následující:

Oblast	Velikost souboru
nad 60° j.š.	57600000
pod 60° j.š.	51840000

Tab. 5 Velikosti souborů jednotlivých částí mapy

3.4 Přesnost modelu GTOPO30

Jako každý model i model GTOPO30 je zatížen určitou chybou. Největší chyby jsou způsobeny nejednotností zdrojů poskytujících potřebné údaje. Uvedeme zde proto orientační tabulku chyb jednotlivých zdrojů.

DTED	30
DCW	160
USGS DEM	30
AMS maps	250
IMW maps	50
Peru map	500
N.Z. DEM	15
ADD	různé

Tab. 6 Přesnosti mapy podle jednotlivých zdrojů

3.5 Co jsme použili

Pro naše potřeby bohatě posloužili jen soubory DEM. S těmi se dále pracuje a cílem celé práce je zachovat jejich tvar i přesto, že nám zdánlivě nelogické rozdělení mapy působilo nemalé potíže. Data je tedy možné získat z různých zdrojů a program by s nimi měl bez problémů pracovat.

4 Prostředí MVE

4.1 Základní charakteristiky

MVE (Modular Visualization Environment) je zjednodušeně vizualizační systém dovolující uživateli řešit daný problém pomocí skládání a propojování základních primitiv (modulů). Pro daný problém může existovat více řešení a uživatel má tak možnost porovnat výsledky a průběh několika z nich a na základě toho si vybrat nejúčinnější metodu pro jeho řešení. Modularita navíc přináší pro mírně pokročilé uživatele

jednoduchou cestu, jak do systému přidat svůj vlastní modul a tím dovoluje rozšiřování řady základních primitiv. Další výhodou MVE je to, že úlohy mohou být v prostředí spuštěny paralelně v jednotlivých vláknech.

4.2 Použité MVE

Pro účely této bakalářské práce bylo použito nekomerčního MVE prostředí pro systém Win98 / NT. Toto MVE vzniklo na Západočeské Univerzitě a skládá se z MVE Editoru a RunTime. Moduly jsou zde realizovány pomocí dll knihoven, které musí dodržovat konvenci danou tvůrci MVE.

4.3 Moduly MVE

Jak jsem již uvedl, moduly jsou do MVE editoru importovány pomocí dll knihoven. Dll knihovna musí obsahovat minimálně následující funkce

```
Get_Modules()  
Free_DLL_Descr()  
  
ModuleName_SETUP_FUNC()  
ModuleName_MAIN_MODULE_FUNC()  
ModuleName_FREE_SETUP_DATA()  
ModuleName_FREE_DATA()  
ModuleName_FREE_STATE()  
ModuleName_HELP_FUNC()
```

Jedna dll knihovna může obsahovat i více modulů a každý z nich musí mít aspoň těchto pět základních funkcí.

```
ModuleName_SETUP_FUNC()  
ModuleName_MAIN_MODULE_FUNC()  
ModuleName_FREE_SETUP_DATA()  
ModuleName_FREE_DATA()  
ModuleName_FREE_STATE()  
ModuleName_HELP_FUNC()
```

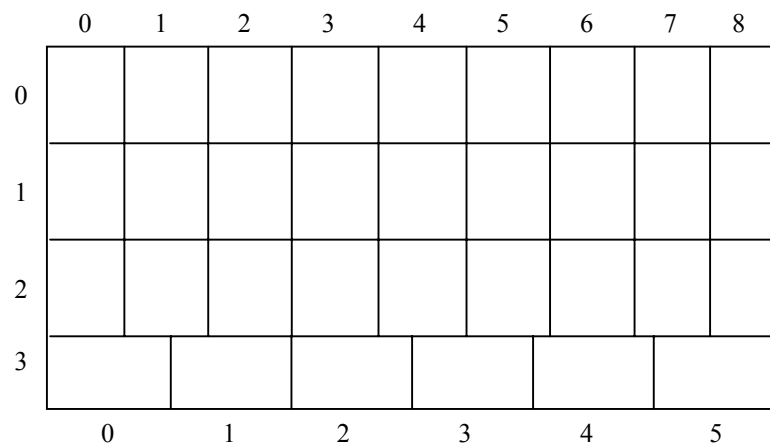
Příslušné moduly se popisují ve funkci Get_Modules(), pro kterou jsou k dispozici makra. V této funkci se také definují vlastnosti modulu (vstupy / výstupy, formát vstupů / výstupů).

5 Import GTOPO30 dat

5.1 Vstupní parametry

Vzhledem k podobě vstupních dat (rozdělení modelu GTOPO30) je nutné si zjistit informace o tom, z kterých souborů bude čteno, na jaké pozici a kam do výsledné sítě bodů se bude zapisovat.

Jednotlivé části modelu (dále jen pláty) jsou oindexovány podle následujícího schématu:



Obr. 1 Rozložení souborů na celkové mapě

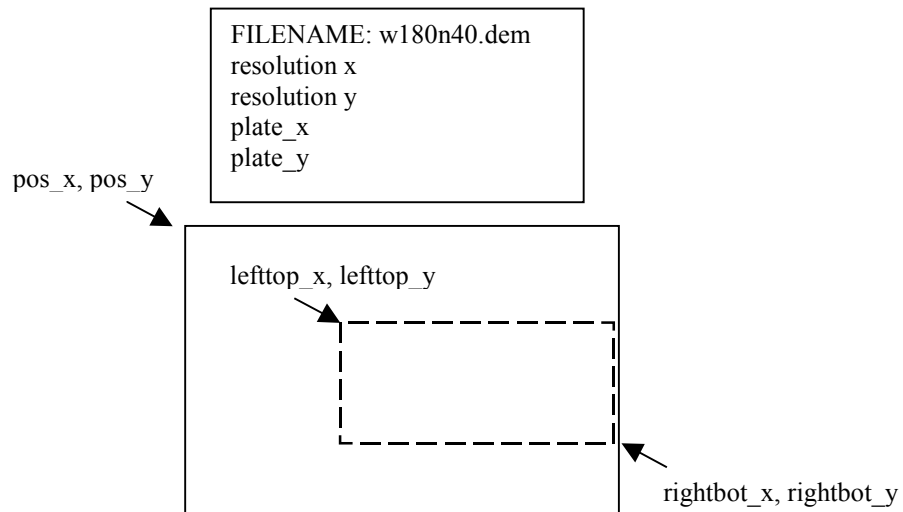
Takto pomocí souřadnic bodu můžeme jednoduše zjistit odpovídající indexy a z těchto indexů pak dále:

- jméno plátu
- souřadnice plátu (levý horní roh) v celém výseku
- relativní souřadnice (souřadnice vzhledem k levému hornímu rohu)

K určení všech informací nám tedy stačí souřadnice dvou bodů, které určují obdélník, ty si převedeme na indexy a postupným výpočtem po řádcích získáme všechny výše uvedené informace (jméno, souřadnice plátu) a ještě relativní souřadnice levého horního a pravého

dolního bodu výseku na mapě. Nakonec ještě potřebujeme rozlišení celého výseku tzn. kolik bodů leží mezi souřadnicemi vrcholů obdélníka.

Jednoduchým přepočtem indexů získáme z těchto dat informace o tom, v jakém souboru číst, kde číst a kam ve výstupní síti zapisovat.



Obr. 2 Reprezentace jednoho plátu a potřebné údaje

resolution_x, reolution_y	rozlišení celého výseku
plate_x, plate_y	rozlišení celého plátu
lefttop_x, lefttop_y	relativní souřadnice levého horního rohu
rightbot_x, rightbot_y	relativní souřadnice pravého dolního rohu
pos_x, pos_y	pozice plátu v celém výseku
FILENAME	jméno souboru

Tab. 7 Vysvětlivky k Obr. 2

5.2 Čtení z-ových souřadnic

Čteme po řádcích a na místě

$$\text{POINT}[x, y] = \text{FILE}[(\text{lefttop_y} + y) * \text{res_x} + x]$$

... pro $x = 0..(\text{rightbot_x} - \text{lefttop_x} - 1)$
 ...pro $y = 0..(\text{rightbot_y} - \text{lefttop_y} - 1)$

Zapisujeme po řádcích

$$\text{WRITE}[(\text{pos_y} + \text{lefttop_y} + y) * \text{res_x} + \text{pos_x} + \text{left_x} + x] = \text{POINT}[x, y]$$

... pro $x = 0..(\text{rightbot_x} - \text{lefttop_x} - 1)$
 ... pro $y = 0..(\text{rightbot_y} - \text{leftbot_y} - 1)$

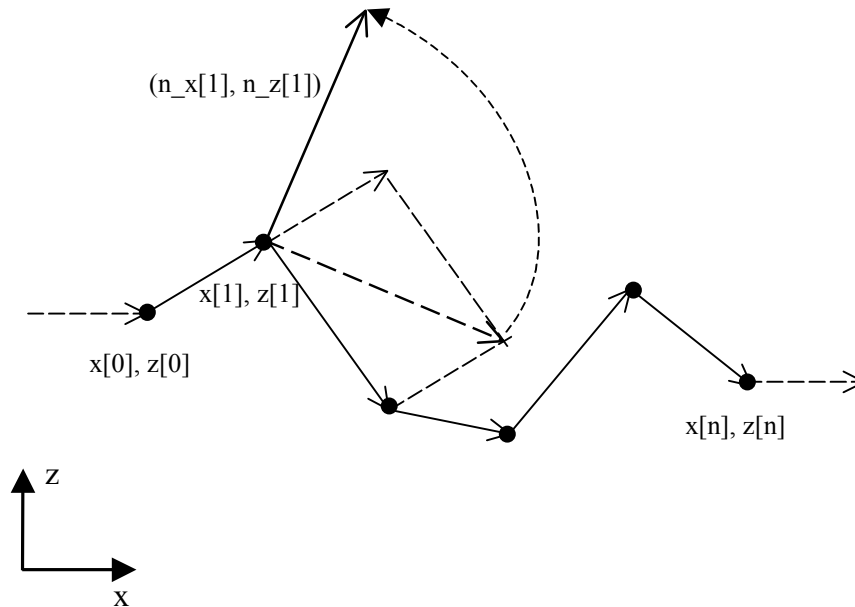
Takto získáme síť bodů, která je připravena pro výpočet normál bodů a indexů trojúhelníků.

5.3 Určení normály v bodě

Normálu v bodě lze získat různými způsoby. Řešení výpočtu normály je rozděleno na tři fáze:

- výpočet složky xz všech normál (problém přenášíme z prostoru do roviny)
- výpočet složky yz všech normál
- součet jednotlivých složek

Na následujícím obrázku si ukážeme jak celý proces probíhá.

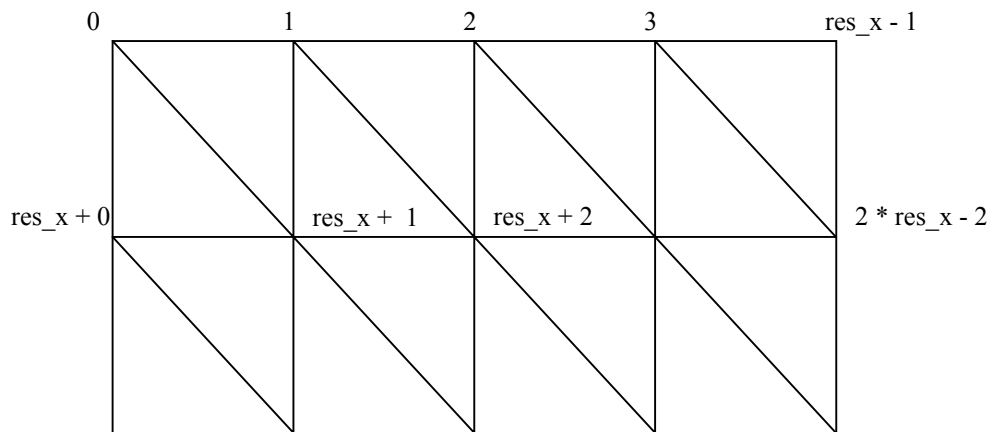


Obr. 3 Výpočet xz složky normály jednoho bodu

V bodech na začátku a na konci se počítá s vektorem rovnoběžným s osou x . Stejně tak jako se vypočetly normály v rovině xz , vypočtou se i normály v rovině yz . Jednoduchým vektorovým součtem těchto dvou složek pak získáme kýžené normály b bodech.

5.4 Indexování trojúhelníků

Vyplnění pole indexů vrcholů všech trojúhelníků je další krok v konstrukci trojúhelníkové sítě. Zvolil jsem klasické uspořádání trojúhelníků podle následujícího obrázku.



Obr. 4 Indexování trojúhelníků v síti

Takže spojujeme body o indexech

$$\text{apex}[0][2 * y * (\text{res_x} - 1) + 2 * x] = y * \text{res_x} + x$$

$$\text{apex}[1][2 * y * (\text{res_x} - 1) + 2 * x] = y * \text{res_x} + x + 1$$

$$\text{apex}[2][2 * y * (\text{res_x} - 1) + 2 * x] = y * \text{res_x} + x + \text{res_x}$$

$$\text{pro } x = 0 .. \text{res_x} - 2$$

$$\text{pro } y = 0 .. \text{res_y} - 2$$

a druhý trojúhelník

$$\text{apex}[0][2 * y * (\text{res_x} - 1) + 2 * x + 1] = y * \text{res_x} + x + \text{res_x}$$

$$\text{apex}[1][2 * y * (\text{res_x} - 1) + 2 * x + 1] = y * \text{res_x} + x + 1$$

$$\text{apex}[2][2 * y * (\text{res_x} - 1) + 2 * x + 1] = y * \text{res_x} + x + \text{res_x} + 1$$

$$\text{pro } x = 0 .. \text{res_x} - 2$$

$$\text{pro } y = 0 .. \text{res_y} - 2$$

5.5 Výpočet normály trojúhelníku

Jednoduchým vektorovým součinem vektorů získaných ze stran trojúhelníků získáme normálu trojúhelníku.

6 Implementace modulu GlobeLoader

6.1 Použité datové typy

Jak již bylo zmíněno, dll knihovna musí obsahovat jisté prerekvizity. Tomu jsou také uzpůsobeny datové typy pro komunikaci mezi MVE a modulem. Zde je jejich popis:

Struktura popisující všechny moduly v DLL knihovně

```
typedef struct {
    int IO_Type;
    char *Descr;
} IO_CONN;
```

IO_Type

číslo datového typu pro daný vstup/výstup (Undefined 0, Points 1, Triangles 2, Tetrahedras 3, Image 4, Volume 5, Freq 6, COSFreq 7)

*Descr

stručný popis (nejlépe tak 1 řádka - v editoru použito jako ToolTip)

```
typedef struct {
    int IO_Num;
    IO_CONN *IO_Types;
} IO_DESCR;
```

*IO_Types

pole (ukazatel na první prvek pole) struktur typu IO_CONN

IO_Num

počet struktur v poli IO_Types

```
typedef struct {
    int Module_Type;
    char *Module_Name;
    char *Module_Descr;
    IO_DESCR *Inputs;
    IO_DESCR *Outputs;
} MODULE_DESCR;
```

Module_Type

číslo typu modulu (Loader 1, ComputingModule 2, Saver 3, Renderer 10)

*Module_Name

jméno modulu (používá se jako předpona ve jménech funkcí modulu)

*Module_Descr

stručný popis modulu (nejlépe tak 1 řádka - použito jako ToolTip)

*Inputs

ukazatel na strukturu IO_DESCR s informacemi o vstupech modulu

*Outputs

ukazatel na strukturu IO_DESCR s informacemi o výstupech modulu

```
typedef struct {
    int Num_Descr;
    MODULE_DESCR *Descriptors;
} DLL_DESCR;
```

Num_Descr

počet struktur v poli Descriptors

*Descriptors

pole struktur typu MODULE_DESCR (popisy modulů).

Datové struktury pro zapouzdření dat

```
typedef struct {
    int Data_Type;
    void *Data;
    unsigned long Data_Length;
    void *Header;
    unsigned long Header_Length;
    int Data_State;
} DATA_DESCR;
```

Data_Type

typ obsažených dat (na jaký typ dat ukazuje pointer Data)(Undefined 0, Points 1, Triangles 2, Tetrahedras 3, Image 4, Volume 5, Freq 6, COSFreq 7)

*Data

ukazatel na samotná data

Data_Length

objem dat - pouze informační (nemá vliv na běh)

*Header

ukazatel na "hlavičku" dat (info může být obsažena ve struktuře Data)

Header_Length

délka hlavičky - pouze informační (nemá vliv na běh)

Data_State

stav dat (před spuštěním modulu je stav všech vstupních dat nastaven na 1, pokud modul po ukončení zpracování již vstupní data nepotřebuje musí nastavit tuto hodnotu na 0) (0 - Data již nejsou potřeba (je možno je uvolnit), 1 - Data jsou stále potřeba (nelze uvolnit), 2 – Možný update dat v rendereru (pouze pro renderer, jinak nepoužívat), 3 – Data jsou právě aktualizována (pouze pro renderer, jinak nepoužívat)

```
typedef struct    {
    int Num_Descr;
    DATA_DESCR *Descriptors;
} DATA_DESCRIPTOR;
```

Num_Descr

počet struktur v poli Descriptors

*Descriptors

pole struktur DATA_DESCR reprezentující všechna vstupní/výstupní data modulu

```
CUSTOM_MODULE_DATA
typedef struct {
    void *Data;
    unsigned long Data_Length;
} CUSTOM_MODULE_DATA;
```

*Data

ukazatel na data s uživatelským nastavením modulu. Musí se jednat o konzistentní blok dat (tj. struktura na kterou ukazuje Data nesmí obsahovat již žádné další ukazatele, pokud obsahuje řetězce tak statické)

Data_Length

přesná délka bloku dat s nastavením na který ukazuje Data

```

typedef struct {
    T_Float      xmin, xmax, ymin, ymax, zmin, zmax;
    T_Float      Pos_x, Pos_y, Pos_z;
    T_Float      Rot_x, Rot_y, Rot_z;
    T_Float      Dis_x, Dis_y, Dis_z;

    // static part
    T_Index      NV_M;
    T_Index      NT_M;
    T_Index      NV;
    T_Index      NT;
    T_Status     Orientation;

    // dynamic part
    P_Coord      P_VCoord[MAX_DIM];
    P_Coord      P_VNorm[3];
    P_Vertex_Status P_VStatus;

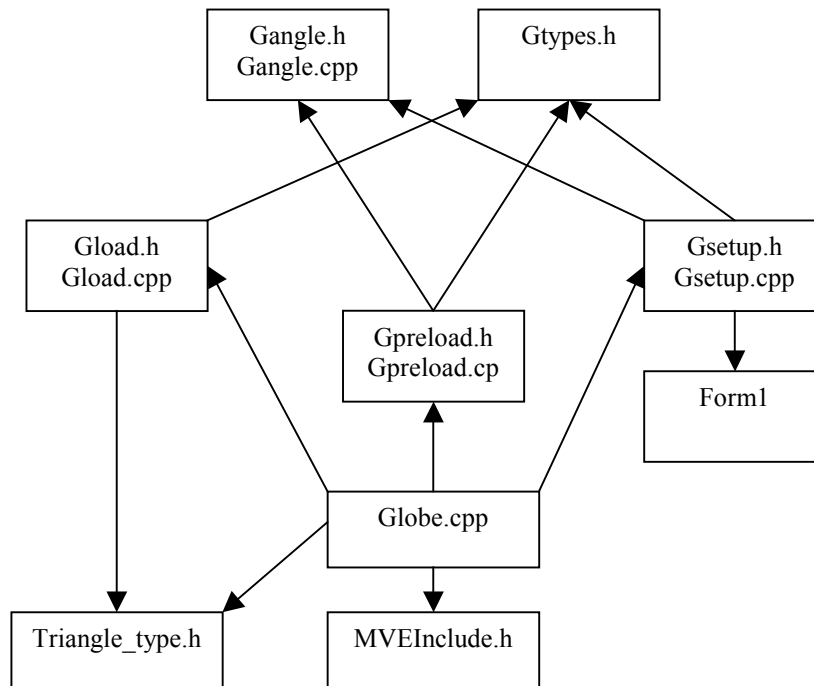
    P_Index      P_TV[3];
    P_Index      P_TT[3];
    P_Coord      P_TNorm[3];
    P_Triangle_Status P_TStatus;
    T_Byte      z_valid;
} T_Triangle_Mesh, *P_Triangle_Mesh;

```

Tato struktura je velice přehledná , a proto není třeba příliš mnoho psát. `NV_M` je počet vertexů , `NT_M` je počet trojúhelníků , `P_VCoord[MAX_DIM]` jsou pole souřadnic bodů , `P_VNorm[3]` jsou tři pole souřadnic normál v bodech , `P_TV[3]` jsou tři pole indexů trojúhelníků , `P_TNorm[3]` jsou pole souřadnic normál trojúhelníků.

6.2 Struktura modulu

Abych se vyhnul v následujícím odstavci konfliktu v označení „modul“, budu zde nazývat modul (ve smyslu programovacího jazyka C++) „unitou“. Knihovna se skládá z unity `Globe.cpp`, která tvoří hlavní jádro v podobě jak jsem ji popsal a využívá funkce s ostatních unit: `Gsetup.cpp`, `Gload.cpp`, `Gpreload.cpp`, `Gangle.cpp`. Každá z nich má ještě hlavičkový soubor a všechny společně využívají služeb hlavičkových souborů `Gtypes.h` a `MVEInclude.h`, kde jsou nadefinované potřebné datové struktury a makra. Pro názornost se podíváme na následující obrázek.



Obr. 5 Struktura dynamické knihovny

Gangle.cpp, Gangle.h

Unita pro aritmetiku ve formátu stupňů (stupně, minuty, sekundy)

Gsetup.cpp, Gsetup.h

Unita obsahuje formulář a její funkce slouží ke komunikaci s uživatelem při volbě setup v MVE editoru.

Gpreload.cpp, Gpreload.h

Zde jsou všechny funkce, které jsou třeba pro upravení dat zadaných uživatelem (výsek modelu). Provádí se zde zpracování informací pro unitu Gload.cpp.

Gload.cpp, Gload.h

Funkce v unitě Gload provádí import bodů ze souborů, vyplnění trojúhelníkové sítě a výpočty normál.

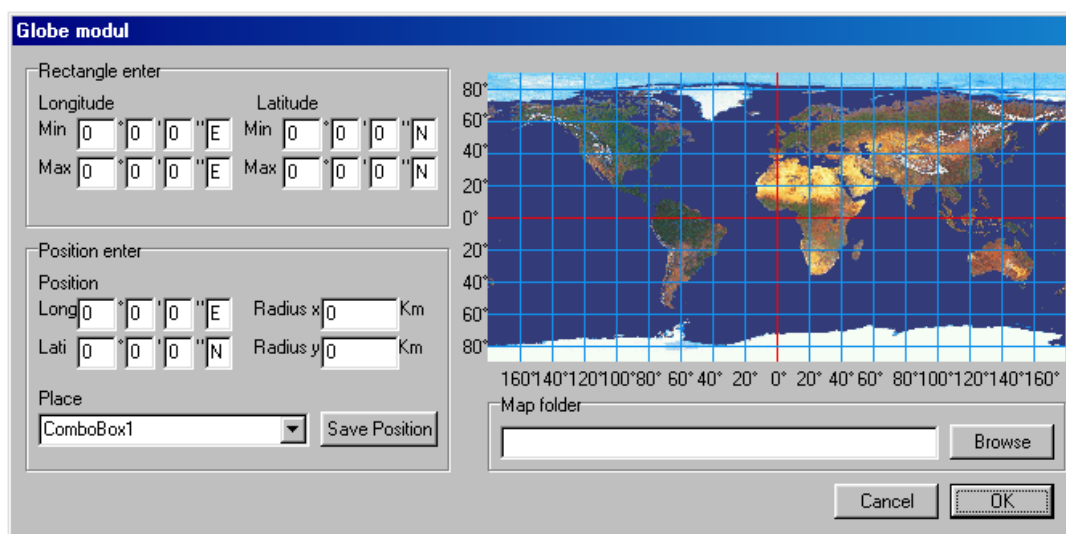
Globe.cpp

Jak bylo již uvedeno, je to hlavní unita dll knihovny.

7 Použití modulu globe loader

7.1 Modul v MVE editoru

Modul se zavádí volbou Modules v Option menu. Pokud je dynamická knihovna s modulem v adresáři modules, modul se zavede automaticky. V editoru se po zavolání položky Modules ve View menu zobrazí seznam všech dostupných modulů a jejich stručný popis. Nyní stačí potřebný modul přetahnout na pracovní plochu editoru a spojit odpovídající si vstupy s výstupy. Modul GlobeLoader má jediný výstup a tím je trojúhelníková síť. Jeho nejjednodušší funkcí je poskytnutí trojúhelníkové sítě rendereru. Jednoduše spojíme vstup modulu renderer s výstupem modulu GlobeLoader a podíváme se do Setup nabídky GlobeLoaderu. Ukáže se následující komunikační okno.



Obr. 6 Prostředí setupu GlobeLoaderu

Požadovaný výsek můžeme získat třemi různými způsoby. První možnost je graficky vyznačit výsek na mapě světa. Levým tlačítkem myši si nastavíme minimální zemskou délku a maximální šířku, pravým tlačítkem si nastavíme maximální zemskou délku a minimální zemskou šířku. Na tom jestli zadáte větší minimální šířku než maximální vůbec nezáleží, musíte však počítat s tím, že zadáte-li větší minimální zemskou délku než maximální, bude výsek počítán opačným směrem a bude obsahovat 180-ý poledník.

Druhý způsob je zadání obou rohů výseku číselně. Je to sice metoda pracnější, nicméně můžete zde zadávat i hodnoty minut a sekund, což u grafického zadání není možné. Tam totiž operujete jen v řádu stupňů.

Poslední způsob je pomocí jednoho bodu na zemském povrchu, jehož přesné souřadnice zadáte, a poloměrů v obou směrech. Představíme si to jako kdybychom zadávali střed obdélníka a poloviny stran.

Všechny tři metody jsou v konečném důsledku provázané a provedete-li změnu na kterémkoliv místě, všechny ostatní hodnoty se podle toho upraví a vykreslí.

Ještě než opustíme setup GlobeLoaderu, je nutné nastavit cestu k adresáři, ve kterém jsou všechny soubory *.dem (soubory všech plátů). Pokud víte přesně, že požadované informace budou jen na plátech, které máte v adresáři nahrané, můžete klidně GlobeLoader používat, aniž by došlo k nějaké chybě.

Spustíme-li nyní RunTime červenou šipkou v levém horním rohu editoru, začne modul načítat. Pokud se při načítání stane nestandardní situace, jako například nedostatek fyzické paměti nebo nenalezený soubor při špatně zadané cestě, modul skončí běh a vrátíte se zpět do editoru MVE.

Zde jsou na ukázkou obrázky z některých zajímavých částí naší Země.

7.2 Spojení modulu GlobeLoader a Decimation

Výškové rozdíly mezi jednotlivými body zemského povrchu jsou většinou velice malé a nadbytečnost velkého množství trojúhelníků přináší velké zatížení při vykreslování.

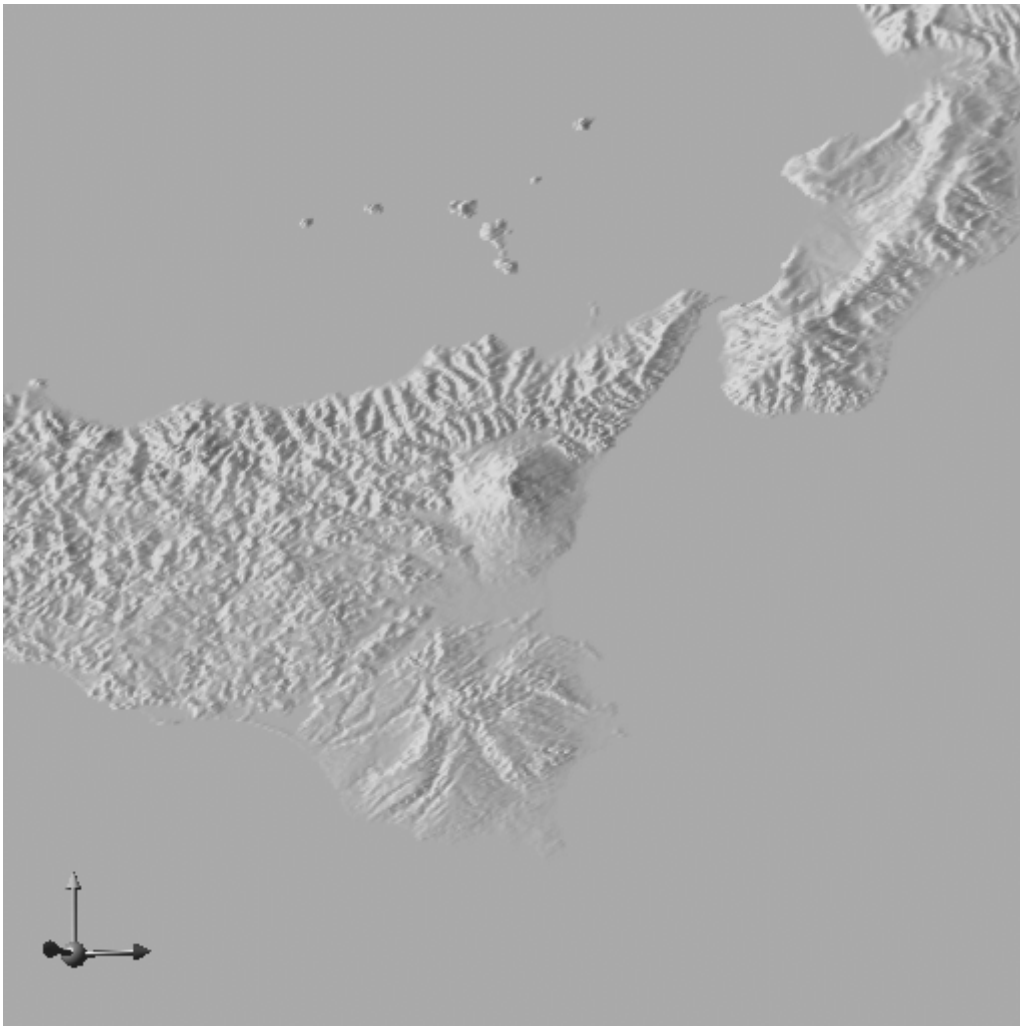
Zvláště parné je to na povrchu moří a oceánů, kde velká rovinná plocha je reprezentována mnoha trojúhelníky a redundance zde nabývá svého maxima. Proto je nutné trojúhelníkovou síť před samotným zobrazením pomocí modulu rendereru zredukovat.

Naštěstí je modul provádějící redukci trojúhelníkové sítě součástí MVE a je poměrně dobře vyřešen.

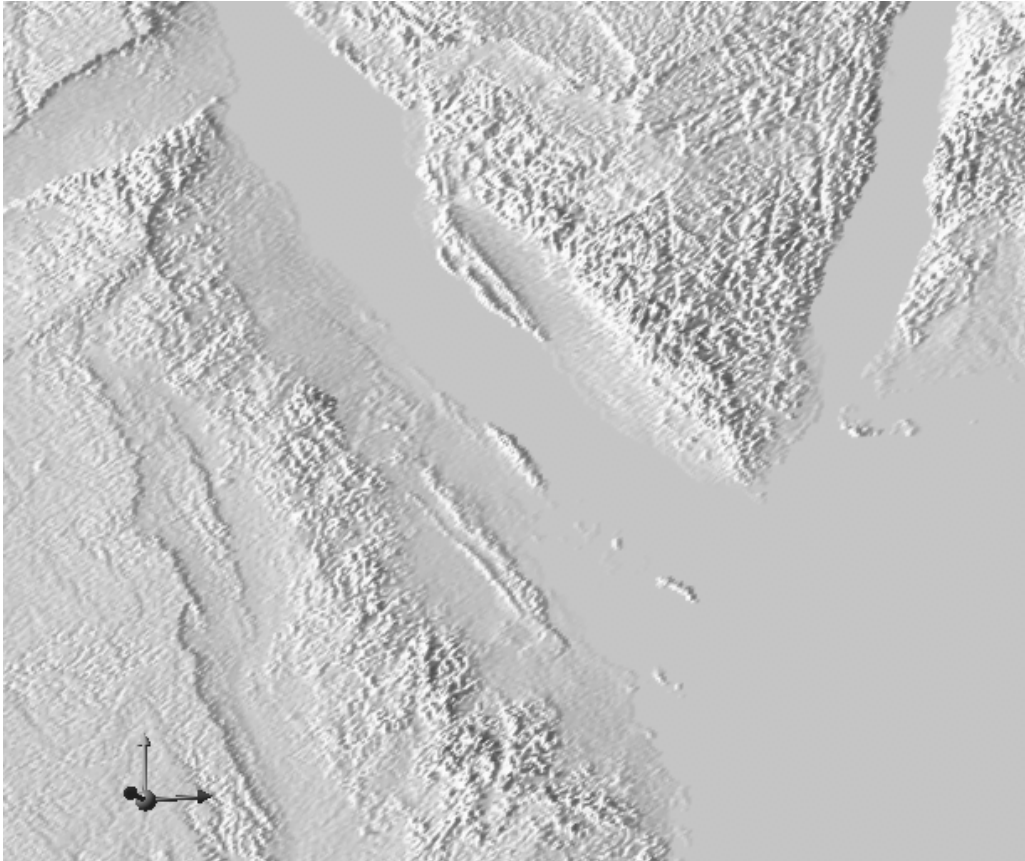
8 Výsledky

8.1 Globe loader s modulem Renderer

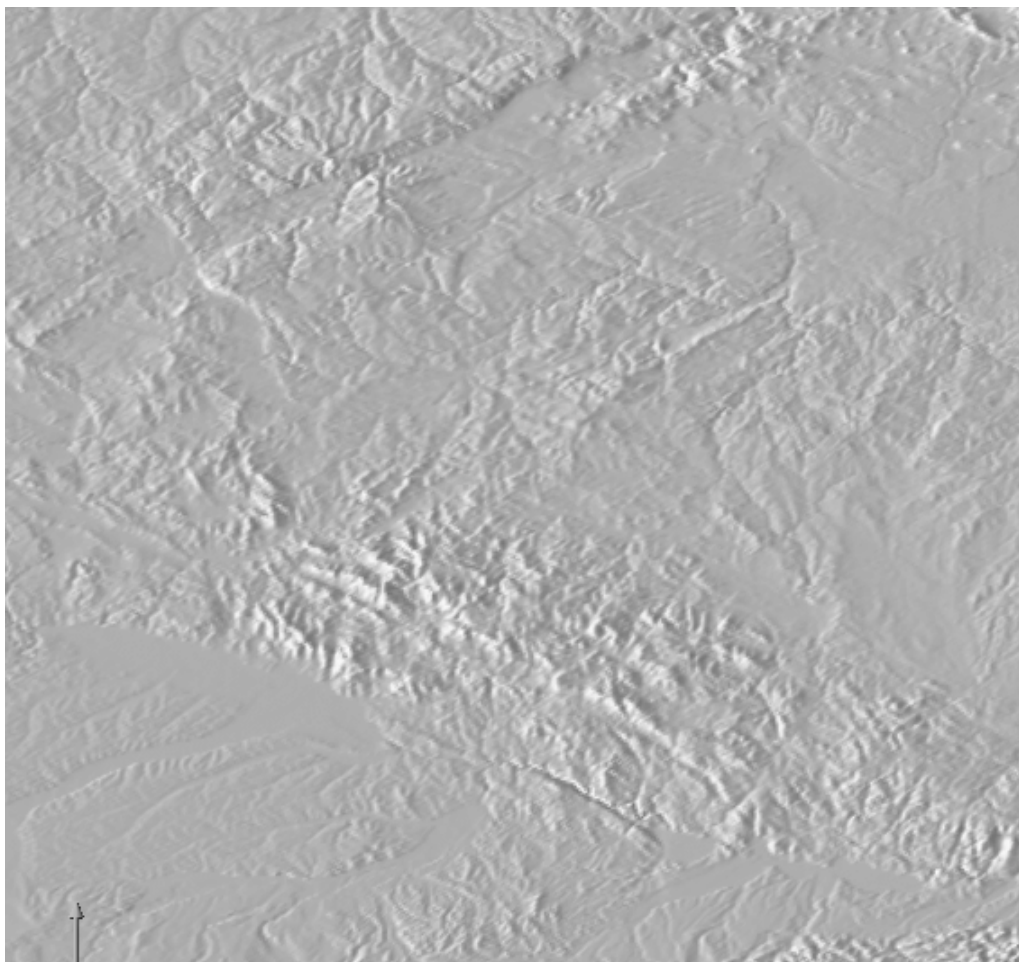
Zde jsou uvedeny některé obrázky vyrenderované v MVE, a to jak přímo z trojúhelníkové sítě povrchu, tak po redukci Schroederovou decimací s různým procentuálním zjednodušením.



Obr. 7 Pohled na Sicílii s dominantní Etnou uprostřed. Vykresleno bez redukce



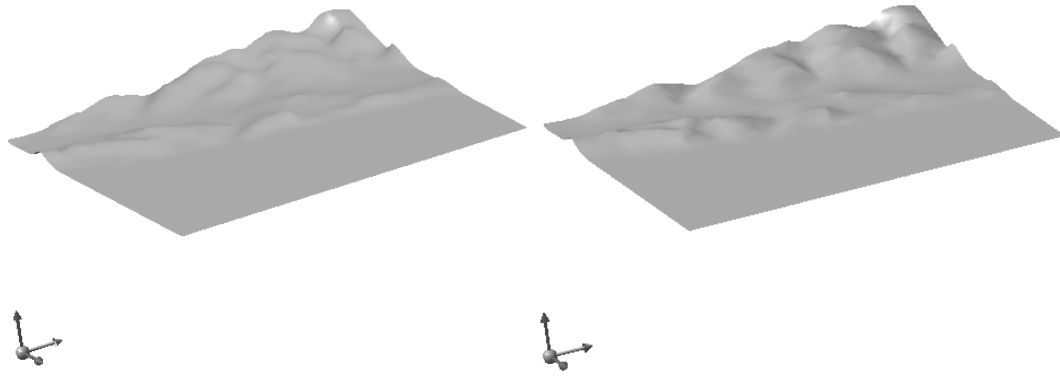
Obr. 8 Suezský záliv. (bez redukce)



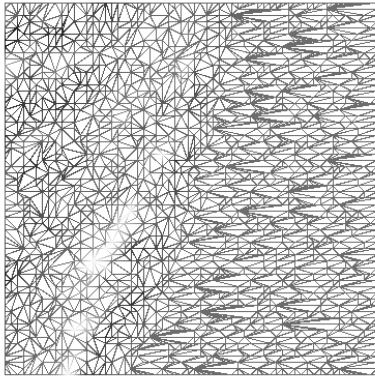
Obr. 9 Šumava a západní část naší republiky. V pravé horní části je dobře patrná Vltava. (bez redukce)

8.2 Globe loader s modulem Decimace

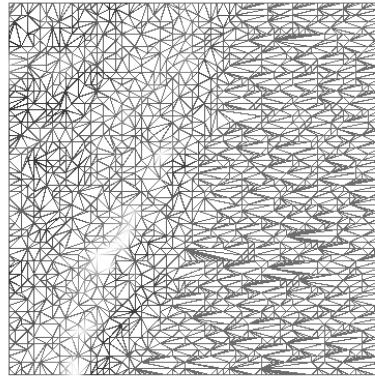
Vyprodukovaný povrch vykazuje velice velkou redundanci. Abychom vybrali vhodnou metodu decimace, nechal jsem vykreslit následující obrázky. Je na nich jedna a táž část pobřeží Tchaj-wanu, kvůli tomu, aby byly patrné výsledky decimace na vodní ploše a na relativně hornatém povrchu vedle sebe. Decimace byla testována na 50-ti procentní redukci, kdy ještě povrch nejevil příliš velké známky deformace. Výsledek



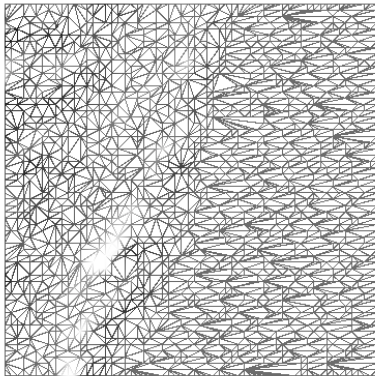
Obr. 10 Plocha před a po 50% decimaci



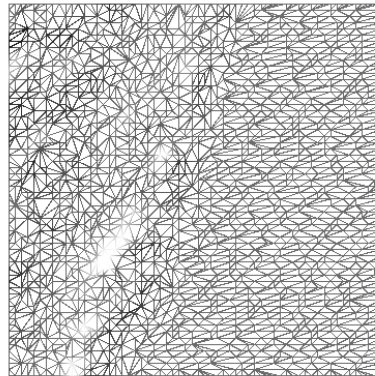
Obr. 12 Schroederova decimace



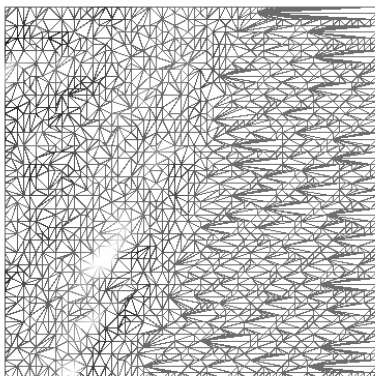
Obr. 12 Metoda průměrné normály



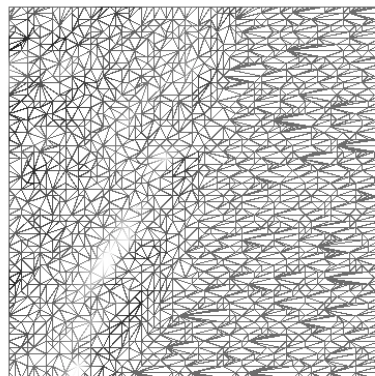
Obr. 13 Absolutní binární



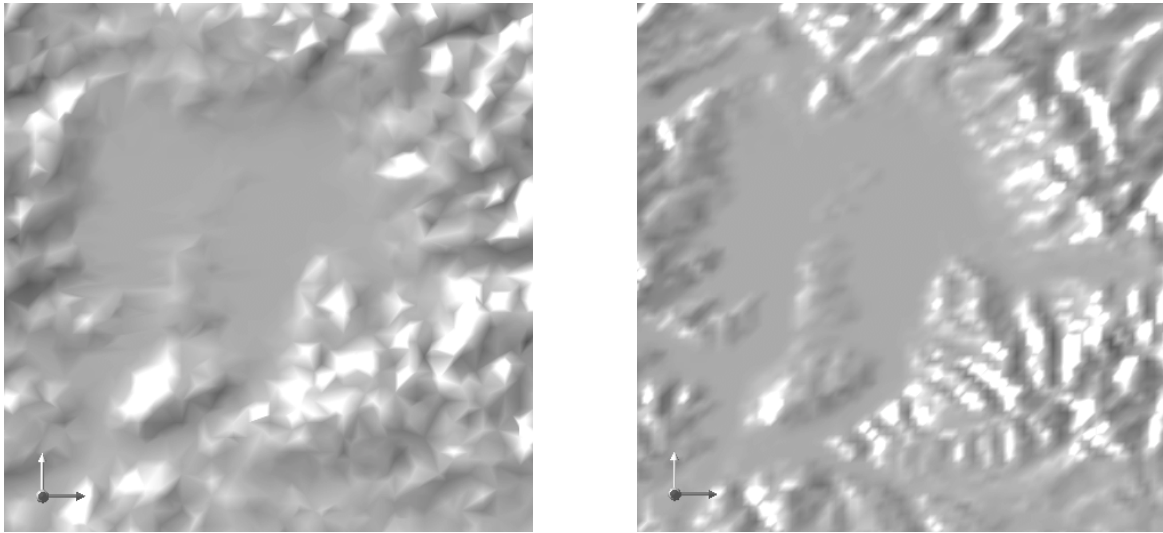
Obr. 14 Pořadí tvarů



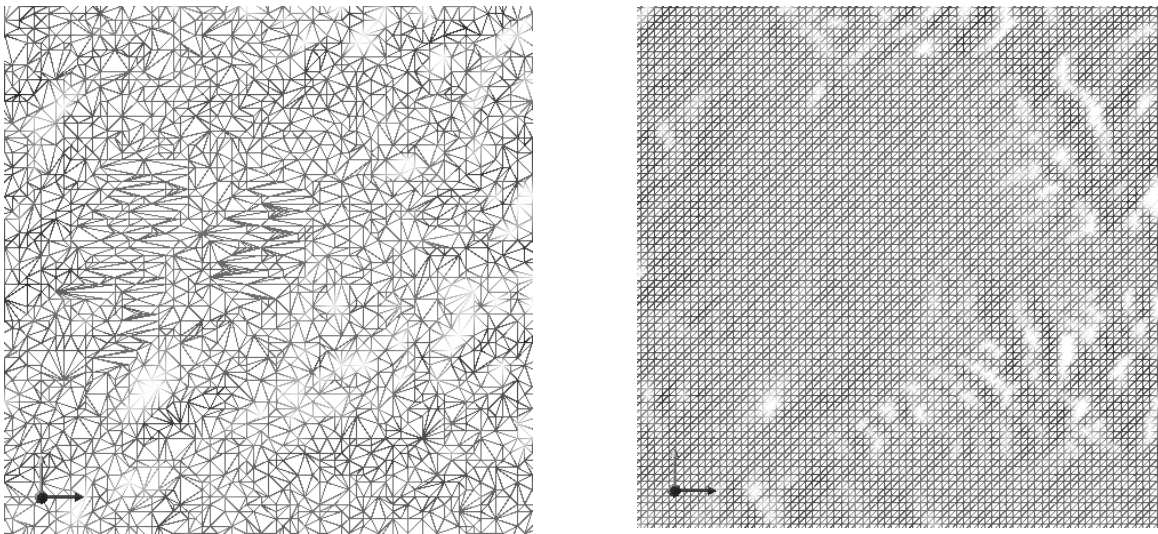
Obr. 15 Přepínání hran



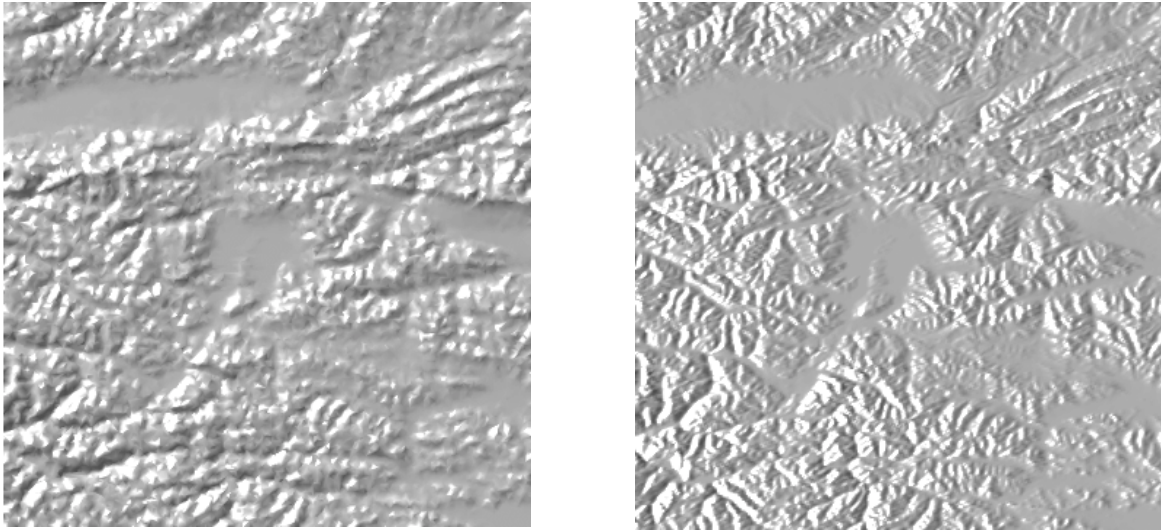
Obr. 16 Metoda náhodná



Obr. 17 Pohled na jezero Kara-kul. Vlevo výsledek Schroederovy decimace (20%), vpravo vyrenderovaná neredukovaná síť.



Obr. 18 Stejný pohled jako na obr. 17, ale povrch je znázorněn trojúhelníkovou sítí



Obr. 19 Detail jezera Kora-kul. Vlevo Schroederova decimace 40%, vpravo před redukcí

9 Poděkování

Mé díky patří panu prof. Václavu Skalovi za upozornění na tento zajímavý projekt a čas, který strávil při přípravě podkladů materiálů u svého počítače na katedře Informatiky a výpočetní techniky. Dále bych chtěl poděkovat všem lidem, kteří pracovali na projektu MVE, díky jejichž urputné práci jsem mohl využívat jimi vytvořené moduly a samotné prostředí MVE.

10 Závěr

Digitální kartografie a to především její trojrozměrná podoba je velice mladou vědní disciplínou, která v posledních letech prožívá velký rozmach. Vývoj trojrozměrných map dnes určují a omezují dva faktory. Jedním je přesnost, z jakou jsme schopni získávat data a druhým je rychlost výpočetních systémů a jejich kapacita. Samotný projekt GTOPO30 je již okolo deseti let stará záležitost, což se projevilo na vzorkování po 30-ti úhlových sekundách. Tato přesnost je již v dnešní době mnohokrát překonána a vzhledem k novým projektům institutu NASA při snímání povrchu Země se můžeme jistě v blízké době těšit na mnohem podrobnější, ale také mnohem objemnější data. Čímž se dostáváme k druhému faktoru. Dnešní počítače jsou schopné bez velkých obtíží zobrazit výřez $20^\circ \times 20^\circ$ v plném rozlišení. Pomocí algoritmů pro redukci trojúhelníkové sítě a zjednodušeným zobrazením vodní hladiny by se tento výsek dal zvětšit takřka na celou viditelnou polokouli. Otázka však je, je-li to vůbec nutné. Pro pohled z pseudoorbitu je toto rozlišení více než dostatečné vzhledem k velikosti zobrazovacího zařízení a klidně bychom se tedy spokojili s řidším vzorkováním. Pro zobrazování z lidského úhlu pohledu je mnohem důležitější, aby objem dat velmi velký, jejich vzorkování husté a systém umožňoval dostatečně rychlou komunikaci mezi pamětí s uloženými daty a pamětí pro zobrazování. Rychlost musí být tak vysoká, aby se člověk mohl běžnou rychlostí pohybovat v pseudoprostoru a systém mohl na základě informací o pohybu a poloze rychle dodat potřebnou část dat zobrazovací jednotce. To jsem ale poněkud odbočil.

Prostředí MVE je vhodnou pomůckou pro zjištění nejlepší metody použitelné pro daný problém. Práce s ním je velice jednoduchá a výsledky, kterých je možné s ním dosáhnout

jsou srovnatelné s komerčními projekty tohoto ražení a jsem rád, že jsem mohl přispět svou bakalářskou prací k rozšíření její knihovny modulů.

Modul splňuje kompatibilitu s původním souborem map, což v praxi znamená, že uživatel MVE není vázán na distribuci souboru map, ale může použít data uložená na ftp serveru v NASA. Navíc je možné pracovat jen s částí mapy pokud s jistotou víme, že výsek na ní bude ležet.

Decimace se ukázala jako velice silná zbraň a výsledky dosažené s modulem „decimation“ byly více než uspokojivé. Při 50 – 70% redukci počtu bodů a trojúhelníků jen velice málo deformuje reliéf krajiny.

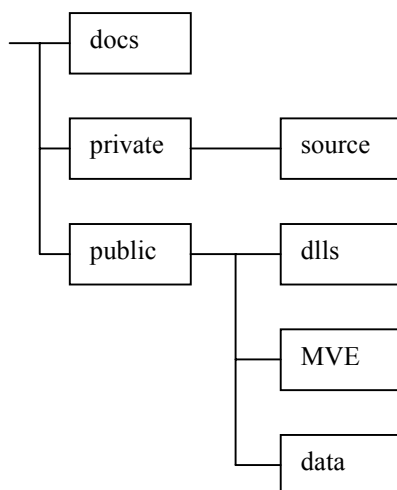
11 Literatura

[1] Žára,J.:Moderní počítačová grafika,Computer press,1998

[2] Bliss, N.B., and Olsen, L.M., 1996. Development of a 30-arc-second digital elevation model of South America. In: Pecora Thirteen, Human Interactions with the Environment - Perspectives from Space, Sioux Falls, South Dakota, August 20-22, 1996.

Příloha A

Součástí práce je CD-ROM, které je rozděleno na následující části:



Docs

Zde je uložena dokumentace elektronická podoba tohoto textu.

Private

Zde je adresář se zdrojovým kódem dll knihovny.

Source

Adresář obsahující všechny unity programu ve zdrojovém kódu i s hlavičkovými soubory a soubory popisujícími okno setupu. Pro vývoj dll knihovny bylo použito C Builder od firmy Inprice verze 4.0.

Public

Zde jsou všechny části důležité pro spuštění modulu.

Dlls

Zde je umístěna dll knihovna globe.dll

MVE

Kompletní distribuce prostředí MVE i se základními knihovnami.

Data

Kompletní popis planety Země je rozdělen do souborů obsahujících jednotlivé pláty pojmenované podle konvencí popsané v hlavní části 3. Jsou komprimované metodou zip.