

# Geometric Structures

## 0. Introduction

Martin Samuelčík

[samuelcik@sccg.sk](mailto:samuelcik@sccg.sk), [www.sccg.sk/~samuelcik](http://www.sccg.sk/~samuelcik), room I4

# Requirements

- Známká z predmetu sa udelí len na základe ohodnotenia implementačného projektu
- Budú dané 4 zadania projektov, každý študent dostane náhodne jedno zadanie na vypracovanie.
- Ak nechápete niektorým častiam zadania, treba sa ozvať na mail, spýtať na prednáške alebo prísť osobne.
- Pri implementácii sa priamo v časti kódu, ktorý rieši zadaný geometrický problém, nemôžu použiť externé zdroje ako externá knižnica, časť kódu z internetu, spolužiakov kód. Pri nájdení takýchto "externých" častí sa projekt ohodnotí Fx. (a hrozí disciplinárne konanie)
- Všetky projekty sú v 2D a musia obsahovať interaktívne ovládanie a aj vizuálnu časť, t.j. grafické zobrazenie zadaných aj výsledných prvkov.
- Po implementácii sa výsledné dielo pošle cez uploadovací systém NAIS ([http://sccg.sk/~samuelcik/\\_nais/](http://sccg.sk/~samuelcik/_nais/)) ako jeden .zip súbor s názvom **dgs\_priezvisko.zip** obsahujúci zdrojový kód, spustiteľný súbor vytvorený v Release konfigurácii pre Win platformu a podrobný návod ovládania aplikácie.
- Po odoslaní súboru už žiadne ďalšie odovzdávania žiadnou formou nie sú možné. Dobré si skontrolujte čo posielate a či je tam všetko a v správnej forme.
- Projekty musia byť odoslané od 1.1.2015 do 8.2.2015, 23:59.
- Po odovzdaní projektu bude hodnotená vizuálna stránka, funkcionálna podľa zadania, spôsob implementácie daného algoritmu, správanie sa programu pri väčšom počte vstupných údajov, ošetrenie "patologických" konfigurácií.
- Získaný počet bodov si po kontrole projektu zistíte v systéme NAIS.

# 1. Project

- **Delaunayova triangulácia + Voronoiov diagram**
- Do okna sa budú klikať body. Keď sa naklikne tretí vrchol, z daných troch vrcholov sa vyrobí a zobrazí trojuholník. Po každom ďalšom naklíknutí nového bodu je potrebné vložiť nový bod do predchádzajúcej Delaunayovej triangulácie tak, aby sme aj po vložení dostali Delaunayovu trianguláciu.
- Treba ošetriť aj prípady, keď viacero ( $>2$ ) daných bodov leží na jednej priamke.
- Nájdenu trianguláciu je potrebné zobrazit' spoločne s naklikanými bodmi a je potrebné zobrazit' aj opísanú kružnicu pre každý trojuholník v triangulácii.
- Zároveň musí projekt obsahovať aj zobrazenie Voronoiovho diagramu, pričom je možné zobrazit' oba grafy naraz rôznymi farbami alebo každý graf zvlášť a zvýrazniť vrcholy Voronoiovho diagramu.
- Voronoiov diagram sa určí ako duálny graf k Delaunayovej triangulácii.
- Po naklíknutí 1 bodu pravým tlačítkom sa určí bunka Voronoiovho diagramu, ktorá tento bod obsahuje. Bunka sa farebne zvýrazní spoločne s jej stredným určujúcim vrcholom.
- Časová zložitosť vytvorenia Delaunayovej triangulácie nesmie prekročiť zložitosť  $O(n^2)$

# 2. Project

- **Marching cubes + Quadtree**
- V okne sa bude generovať automaticky pravidelná mriežka, pričom sa bude dať zadávať rozmer mriežky v tvare  $(2^n + 1) \times (2^n + 1)$
- V každom mrežovom bode bude nejaká hodnota, defaultne 0
- Hodnota v každom mrežovom bode sa bude musieť dať zmeniť kliknutím na bod a vypísaním novej hodnoty.
- Mrežové body budú navzájom farebne odlišené podľa aktuálnej hodnoty.
- Po stlačení pravého tlačítka sa nad pravidelnou mriežkou vygeneruje izokontúra pomocou algoritmu marching cubes, pričom izohodnota sa zadá tiež používateľom.
- Nájdená izokontúra sa vykreslí v okne ako zvýraznená lomená čiara.
- Pri generovaní pomocou marching cubes sa použije quadtree na urýchlenie generovania izokontúry t.j. preskočenia tých častí mriežky, kde sa izokontúra určite nenachádza.
- Pri generovaní izokontúry pomocou marching cubes sa použije lineárna interpolácia na hraniciach buniek.
- Po kliknutí 1 bodu stredným tlačítkom do okna sa vypíše hodnota pre tento bod, ktorá sa určí bilineárnou interpoláciou v príslušnej bunke.
- Projekt bude obsahovať aj 3 ďalšie tlačítka, ktoré po stlačení vyplnia mriežku hodnotami funkcií  $x^2 + 2y^2 - 1$ ,  $x^3 - y$ ,  $\sin(x) - y^2$

# 3. Project

- **DCEL**
- V okne sa bude vykreslovať pravidelný  $n$ -uholník bez stredného bodu a zároveň nejaká jeho triangulácia, pričom  $n$  sa bude dať zadávať používateľom.
- Nad takýmto grafom sa vždy po zmene  $n$  vygeneruje reprezentácia grafu pomocou DCEL obsahujúca polhrany.
- Časť mimo  $n$ -uholníka treba brať ako ďalšiu oblasť.
- V okne sa vykreslia pri vrchoch, polhranách a oblastiach ich indexy v príslušných DCEL tabuľkách.
- Zároveň sa vypíšu tabuľky vrcholov, polhrán a oblastí aj s príslušnými susednosťami.
- Po nakliknutí na nejakú oblasť sa farebne zvýrazia susedné oblasti, pričom na ich vyhľadanie sa použije DCEL štruktúra.
- Po nakliknutí na nejaký vrchol sa farebne zvýrazia všetky vrcholy spojené s daným vrcholom hranou, pričom na ich vyhľadanie sa použije DCEL štruktúra.
- Po nakliknutí na nejakú hranu sa farebne zvýrazia všetky hrany v 2-susedstve (to sú hrany ktoré susedia s danou hranou alebo susedia so susednou hranou danej hrany), pričom na ich vyhľadanie sa použije DCEL štruktúra.

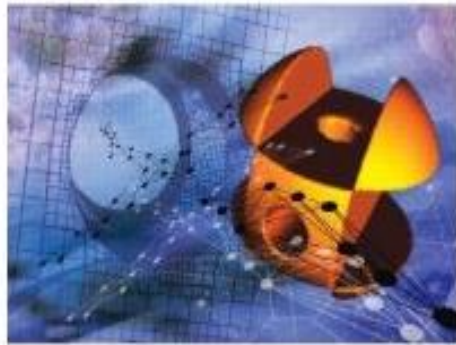
# 4. Project

- **kD-strom**
- V okne sa budú zadávať body nakliknutím ľavým tlačítkom body.
- Vždy pri pridaní nového bodu sa nad danými bodmi vytvorí vyvážený kD-strom tak, aby v listoch bol nanajvýš jeden bod.
- Hranice buniek kD-stromu sa v okne zobrazia spolu s nakliknutými bodmi.
- Po nakliknutí 1 bodu pravým tlačítkom sa použije daný kD-strom na nájdenie k najbližších bodov (číslo k sa bude zadané používateľom) z daných n bodov k nakliknutému bodu v logaritmickom čase. Nájdené najbližšie body sa zvýrazia.
- Po stlačení a držaní stredného tlačítka sa ťahaním kurzora zadá obdĺžnik. Potom sa pomocou kD-stromu nájdu všetky zadané body, ktoré v danom obdĺžniku ležia a zvýrazia sa.
- Držaním Shift klávesy a kliknutím ľavého tlačítka sa vyberá oblasť kD-stromu. Potom sa pre túto oblasť nájdu všetky susedné oblasti a vykreslia sa výraznou farbou.

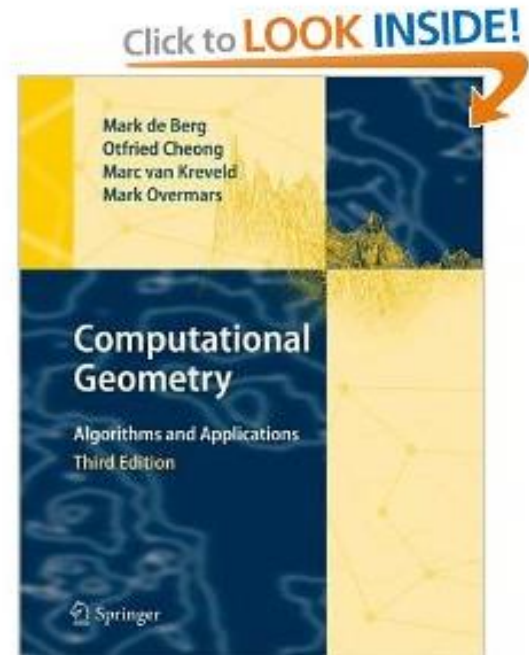
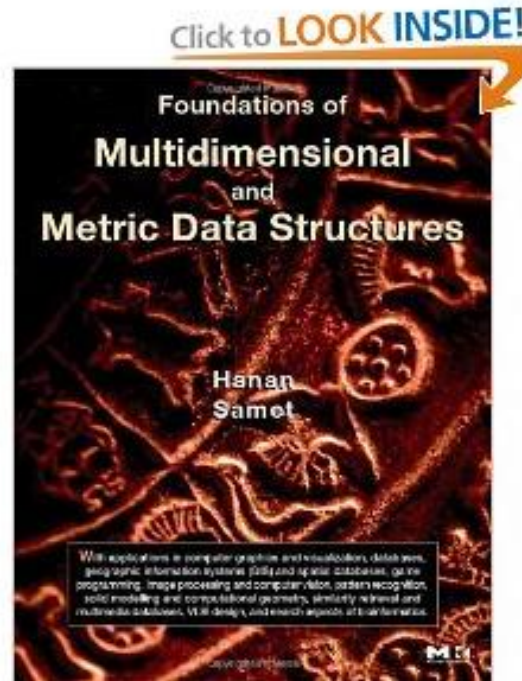
# Literature

- Elmar Langetepe, Gabriel Zachmann: *Geometric Data Structures for Computer Graphics*, AK Peters
- Hanan Samet: *Foundations of Multidimensional and Metric Data Structures*, The Morgan Kaufmann
- Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars: *Computational Geometry: Algorithms and Applications*, Springer-Verlag
- <http://dac.escet.urjc.es/rvmaster/rvmaster/asignaturas/aeda/zachmann-2003-geomdatastr.pdf>
- Donald E. Knuth: *The Art of Computer Programming*, Addison-Wesley Professional
- N. Wirth: *Algoritmy a štruktúry údajov*, Alfa
- <http://www.cgal.org/>, <http://www.google.com>

# Literature



Elmar Langetepe  
Gabriel Zachmann



Google



# Presentations

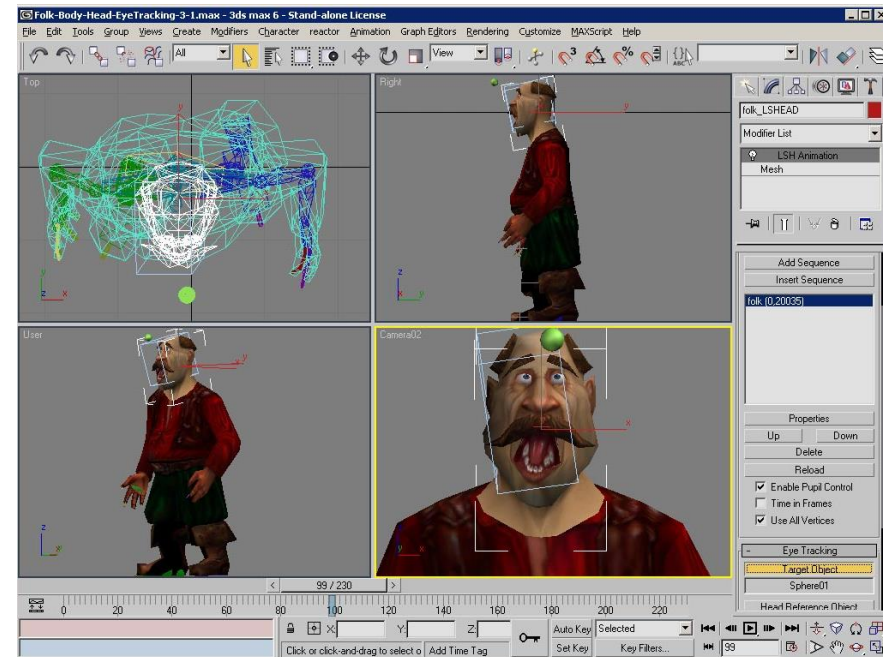
- Space search
  - Range, interval, segment trees
  - Quadtrees, Kd-trees, BSP, ...
  - Bounding volumes trees
  - Dynamical structures
- Representations
  - Manifolds representations
  - Volume data, point clouds
- Graphs
  - Proximity graphs
- Structures on GPU
- Applications, articles

# Motivation

- Geometric modeling, CAD-CAM
- Physical simulations
- Volumetric graphics
- Vizualization
- GIS systems
- Image processing, recognition
- All computer graphics applications

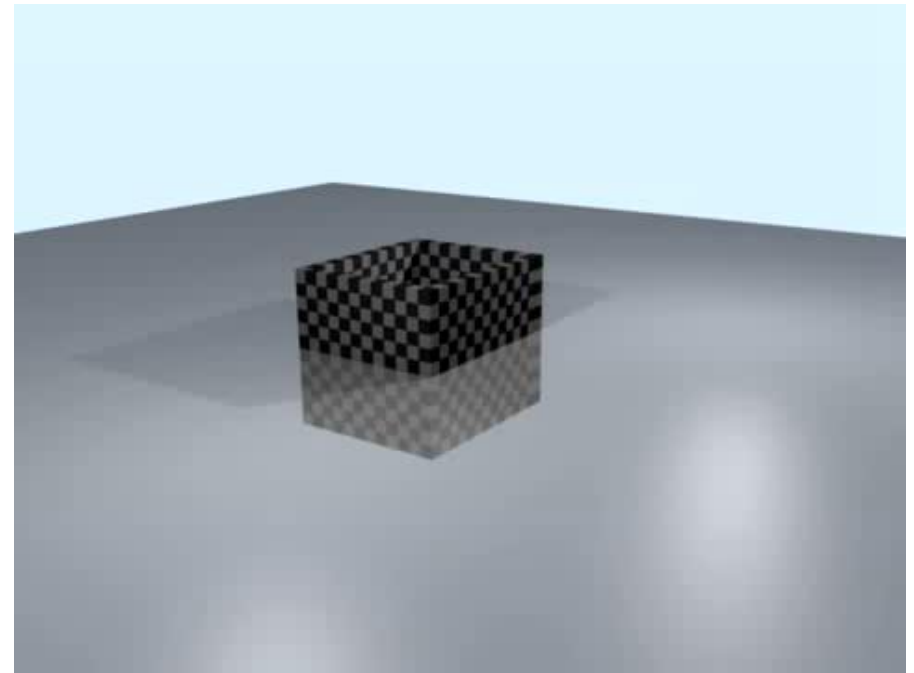
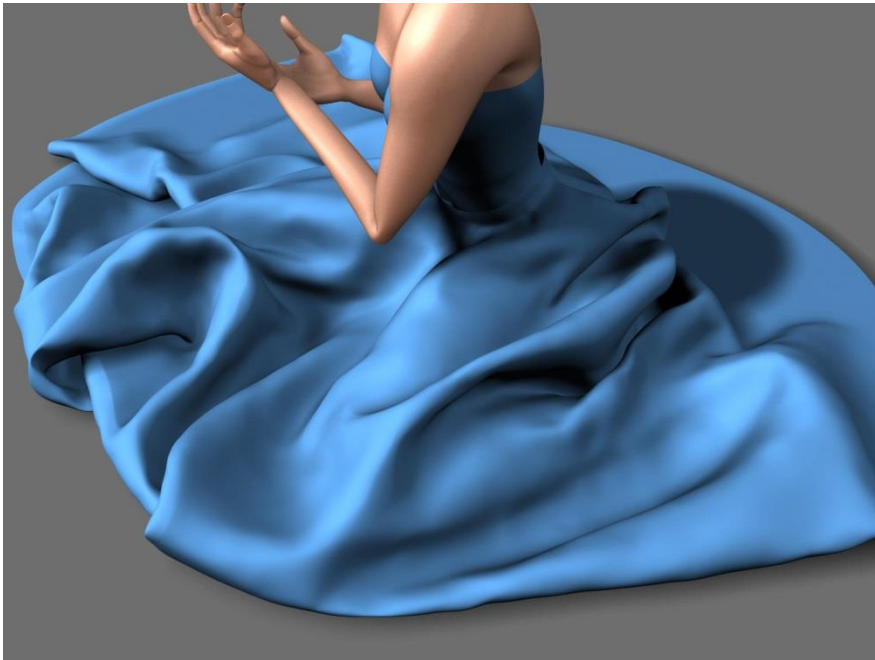
# Geometric modeling

- Using multiple representations
- Subdivision, implicit, parametric surfaces
- Local and global surface editing
- Conversions, formats



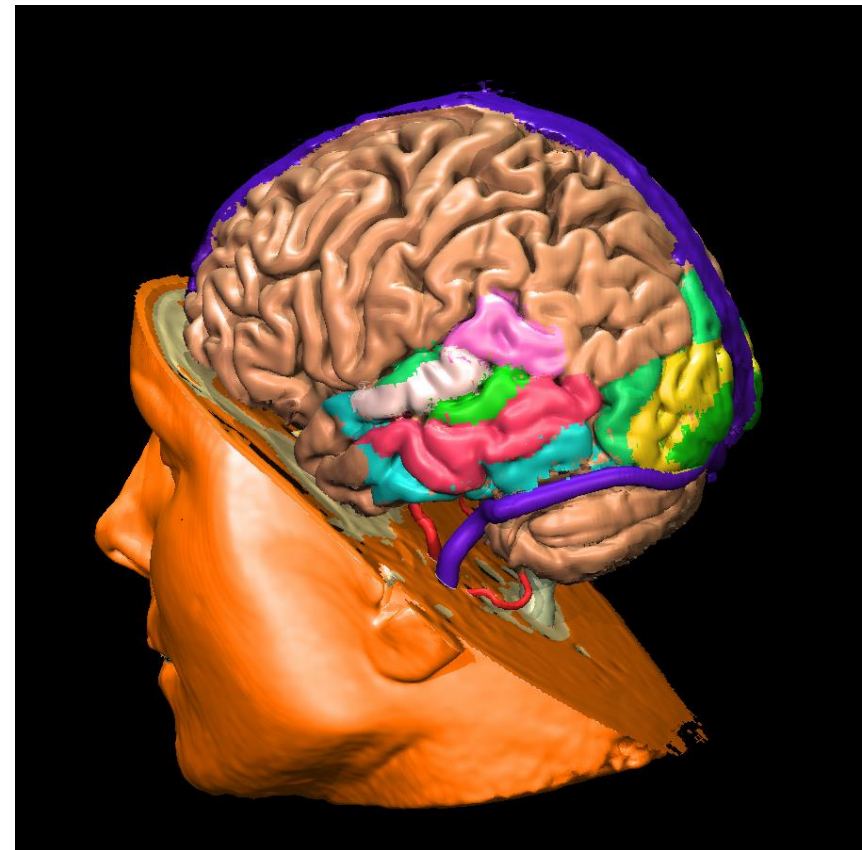
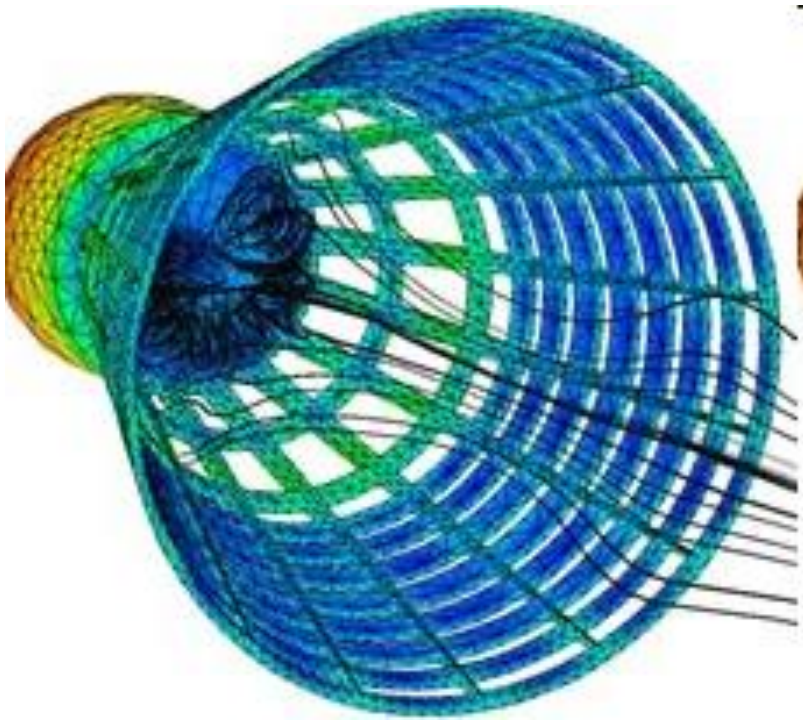
# Physical simulations

- Collision detection
- Deformations
- Physical phenomena



# Volume graphics

- Distance fields
- Voxel representations



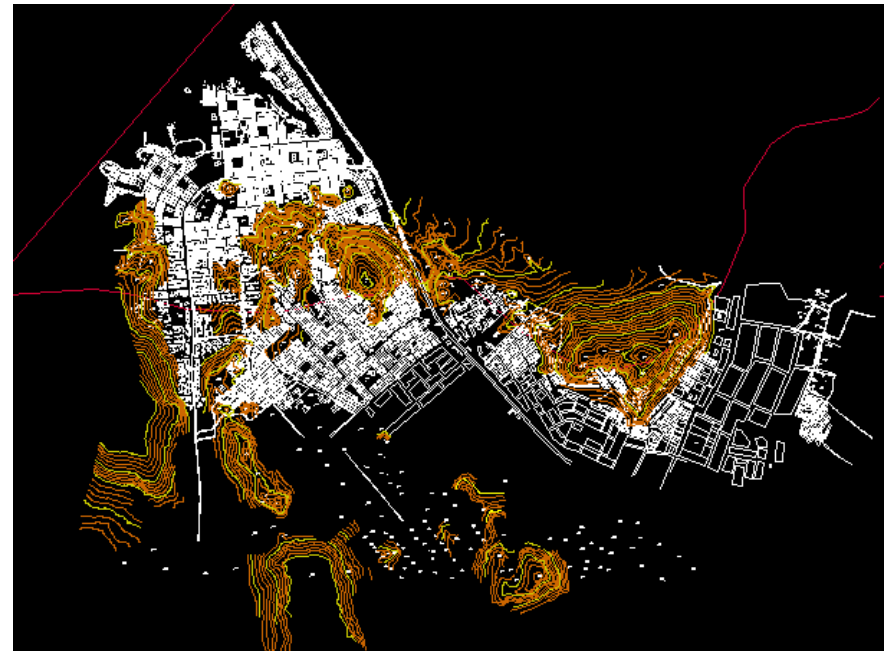
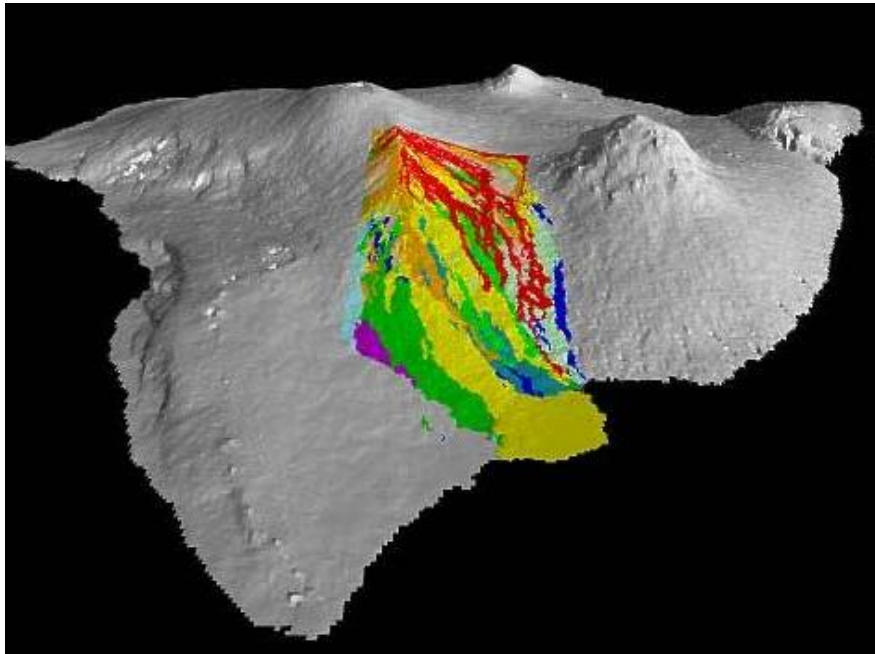
# Vizualiztion

- Raytracing, global illumination
- Visibility, shadows
- Terrain



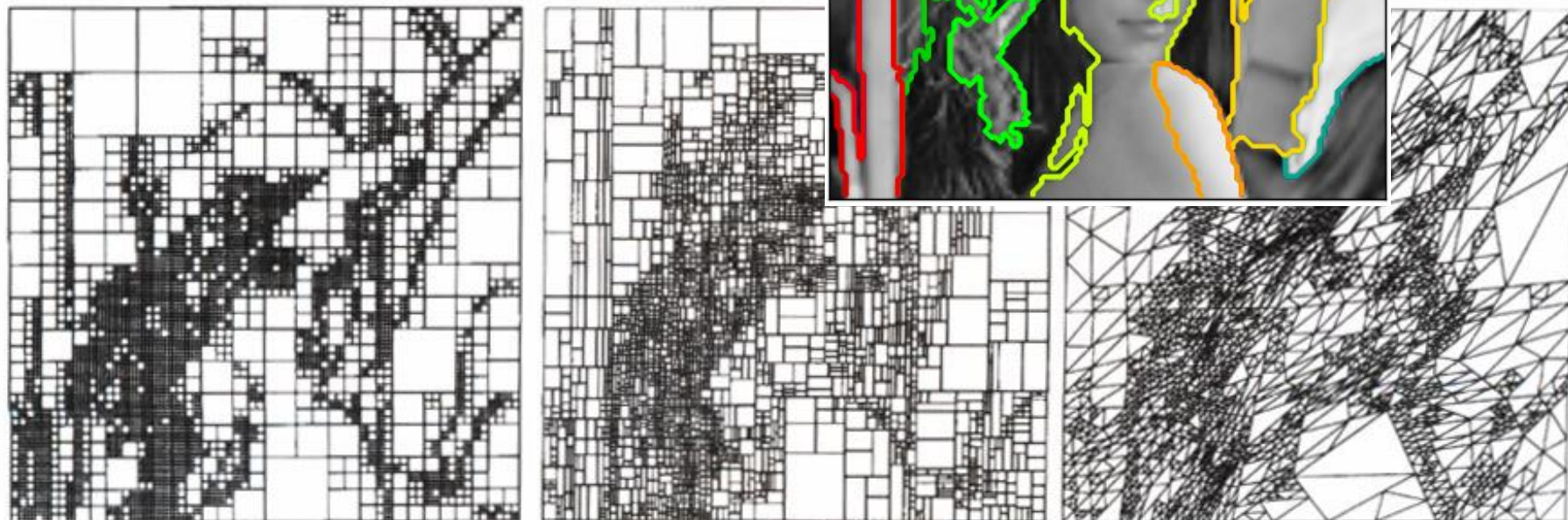
# GIS systems

- Maps processing
- Planning, simulations



# Image processing

- Segmentation
- Compression





# Algorithm

- Konečný návod ako riešiť problém s použitím daných elementárnych operácií
- Dobre definovaná procedúra, ktorá pre nejakú množinu vstupov vyprodukuje výstupnú hodnotu alebo množinu hodnôt
- Postupnosť krokov výpočtu, ktorý transformuje vstup na výstup
- Algoritmus nazývame správny, ak pre každý príklad vstupu sa zastaví so správnym výstupom

# Algorithm complexity

- Miera zložitosti  $X$  (čas, pamäť, počet aritmetických operácií...)
- Funkcia veľkosti vstupných dát udávajúca množstvo miery zložitosti  $X$  spotrebovanej algoritmom  $A$  pri riešení problému
- Veľkosť vstupu: tento pojem závisí od typu problému - počet prvkov vstupu, počet bitov, dva a viac parametrov (počet vrcholov, počet hrán)
- Rôzne typy: asymptotická zložitosť, presný počet operácií, ...

# Asymptotic complexity

- Behavior of algorithm for large input
- Using asymptotic notation
  - $O$ -notation – Asymptotic upper limit, complexity of worst case
  - $\Omega$ -notation - Asymptotic lower limit, complexity of best case
  - $\Theta$ -notation – Asymptotic upper and lower limit of complexity
  - “Small” notations

# Data structures

- $U$  – universal set – set of possible elements of data structure
- $S$  – set of data structure elements
- Inner data structure
- Operations
  - $\text{MEMBER}(x, S)$  – check if  $x \in S$  and find where it is stored(pointer)
  - $\text{INSERT}(x, S)$  – insert  $x$  into  $S$
  - $\text{DELETE}(x, S)$  – delete  $x$  from  $S$
  - $\text{MIN}(S)$  – return smallest element of  $S$
  - $\text{MAX}(S)$  – return greatest element of  $S$
  - $\text{SUCCESSOR}(S, x)$  – successor of element  $x$  in set  $S$
  - $\text{PREDECESSOR}(S, x)$  – predecessor of element  $x$  in set  $S$
  - $\text{SPLIT}(x, S)$  split set  $S$  into 2 disjunct sets  $S_1, S_2$  such that  $S_1 = \{y; y \in S, y < x\}$ ,  $S_2 = \{y; y \in S, y > x\}$
  - $\text{CONCATENATE}(S_1, S_2)$  – Concatenate two sets ( $S_1 \cup S_2$ )

# Basic structures

- List
  - Linear complexity, compactness
  - Array, linked list
- Tree structures
  - Faster searches, balanced trees
  - Binary search trees
  - AVL trees
  - Red-Black trees
  - B-trees
  - Heap
  - ...



**Otázky?**