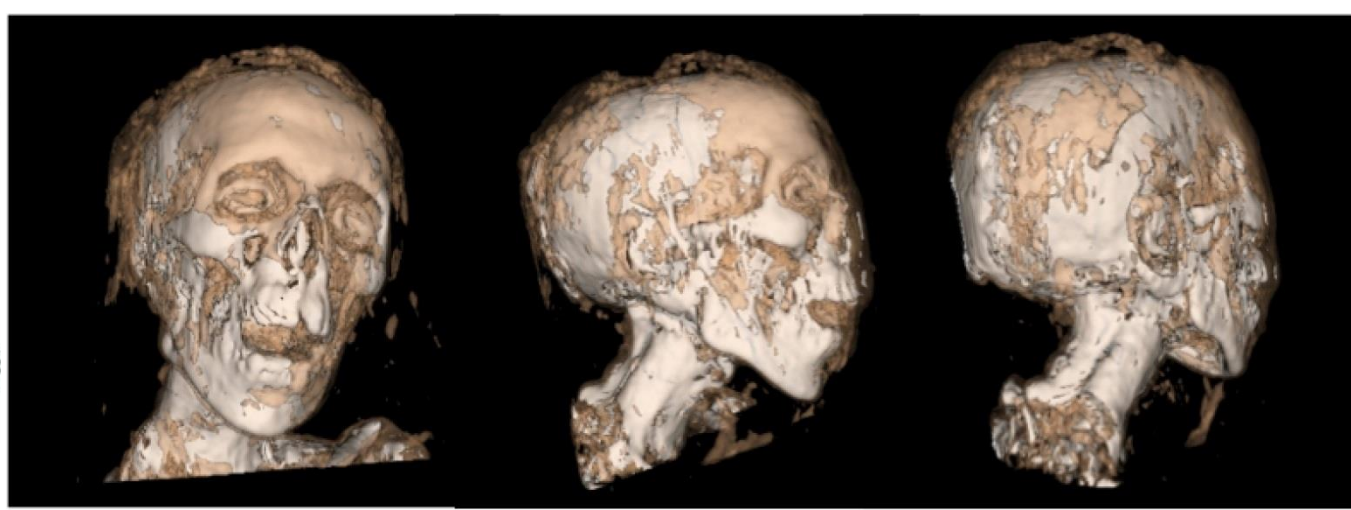
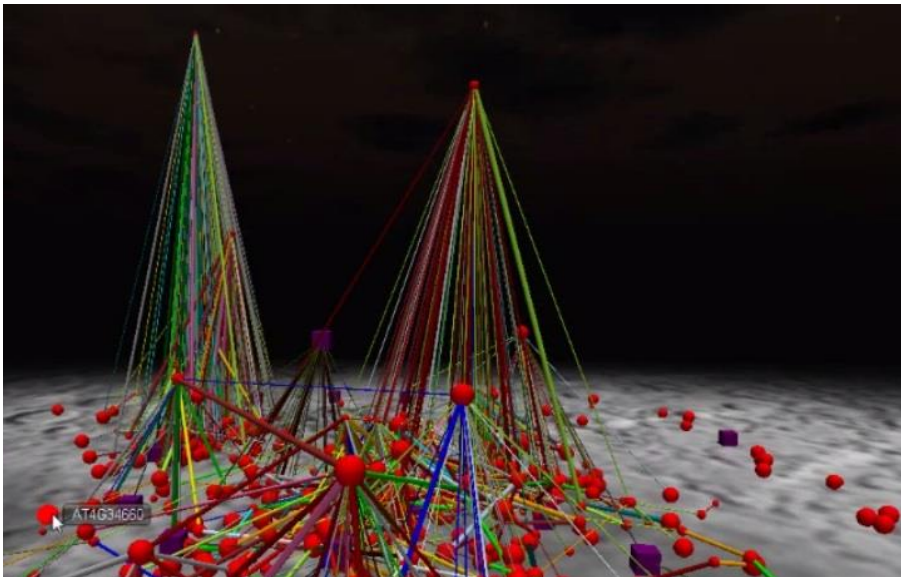


blogs.library.duke.edu

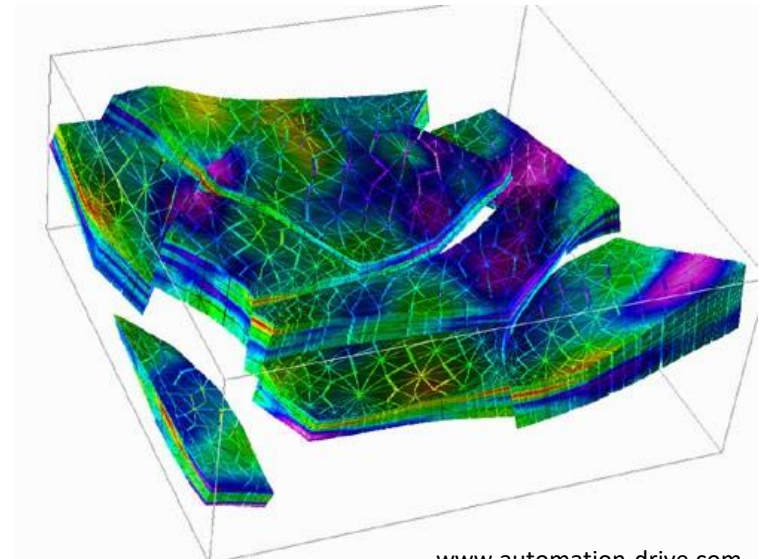


visthis.blogspot.com

4. Vizualizace prostorových dat



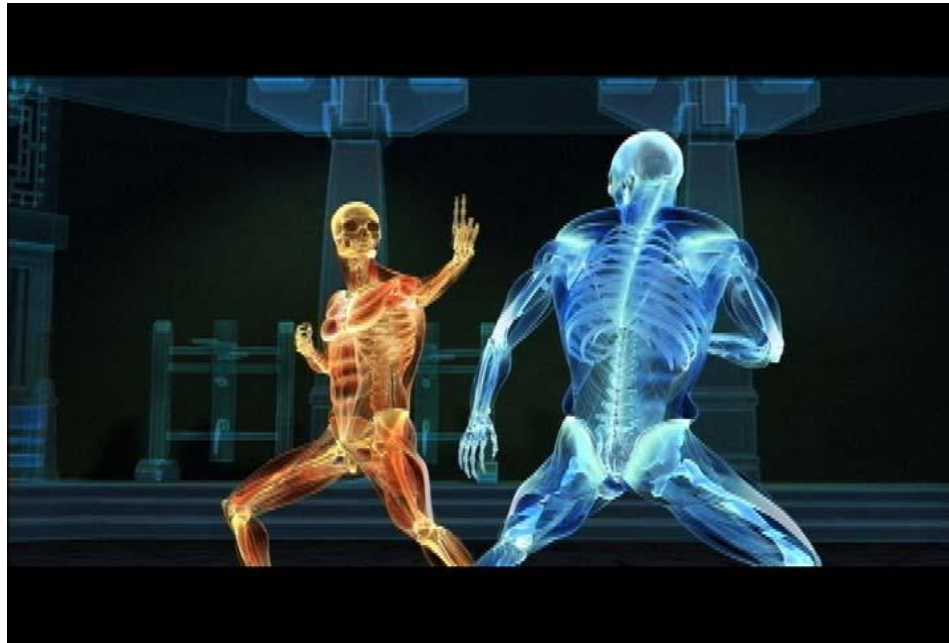
www.hypergridbusiness.com



www.automation-drive.com

Vizualizace prostorových dat

- Vstupní datové sady obsahují prostorové či časoprostorové atributy
- Zrakový systém zpracovává tato data podobně jako okolní svět



Reálný svět vs. obrazovka

- Ve skutečném světě nejsme omezeni na 2D, diskretizace, nízké rozlišení
- Na obrazovce:
 - Prozkoumání dat v různém měřítku
 - Dynamická změna kontrastu, nasvětlení, rozlišení
 - Interaktivní průzkum reálně nedostupných míst
 - Interaktivní přidávání a odstraňování částí dat

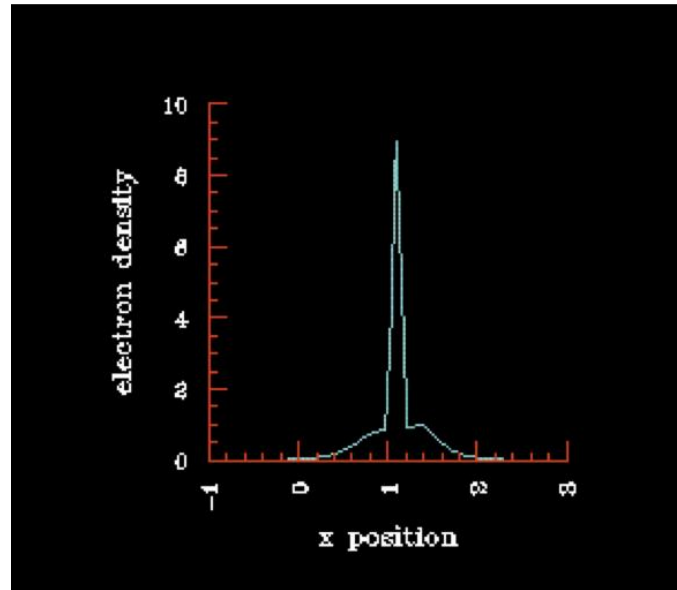
Mapování atributů

- 1. fáze:
 - Mapování prostorových atributů dat na prostorové atributy obrazovky (transformace)
- 2. fáze:
 - Mapování zbývajících atributů – barva, textura, velikost a tvar grafických entit, ...

1D data

- Sekvence 1D dat o jedné proměnné

– Graf



<http://www.opendx.org>

– Color bar

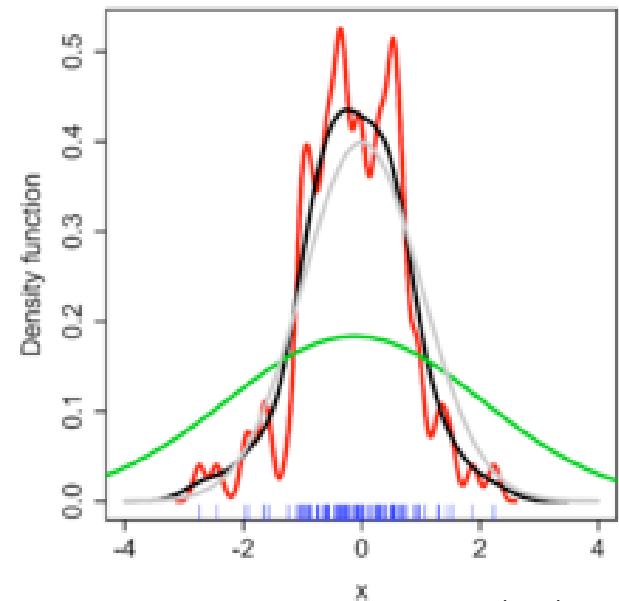
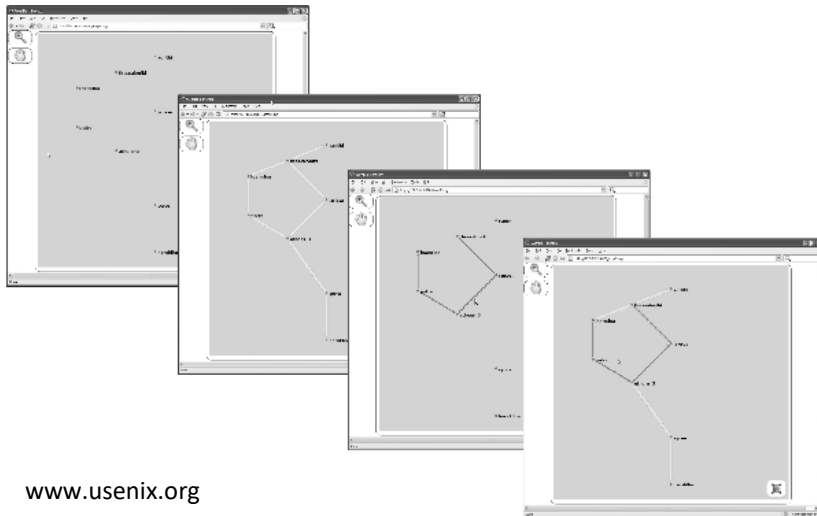


Vykreslení 1D grafu - algoritmus

- Vstup:
 - data_{\min} , data_{\max} – minimální a maximální hodnota v datech
 - $\text{data}_{\text{count}}$ – počet dat pro zobrazení
 - Obrazovka pro vizualizaci dat – obdélník (x_{\min} , y_{\min} , x_{\max} , y_{\max})

1D multivariate data

- Více proměnných nebo více hodnot pro jeden datový vstup
- Rozšíření předchozí techniky
 - Juxtapositioning
 - Superimpositioning



2D data

- Dvě prostorové dimenze – mapování prostorových atributů dat na prostorové atributy obrazovky.
- Typy vizualizace 2D dat:
 - Obrázek (image)
 - Reliéf (rubber sheet)
 - Cityscape
 - Bodový graf (scatterplot)
 - Mapa
 - Kontura, izobary

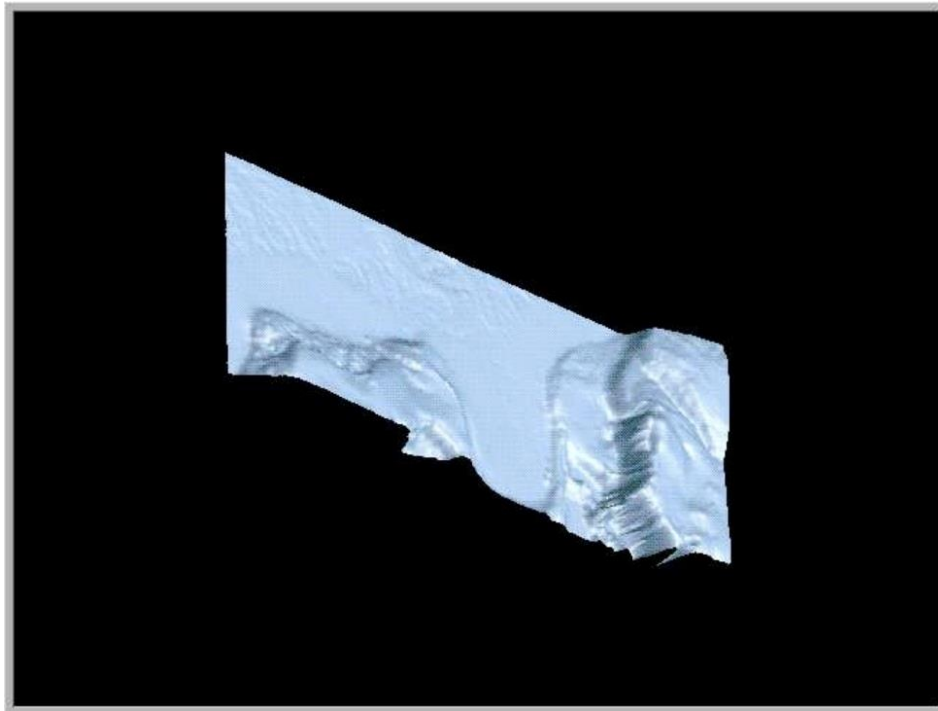
Obrázek (image)

- Datová hodnota mapována na barvu v dané pozici, barva mezilehlých pixelů interpolována



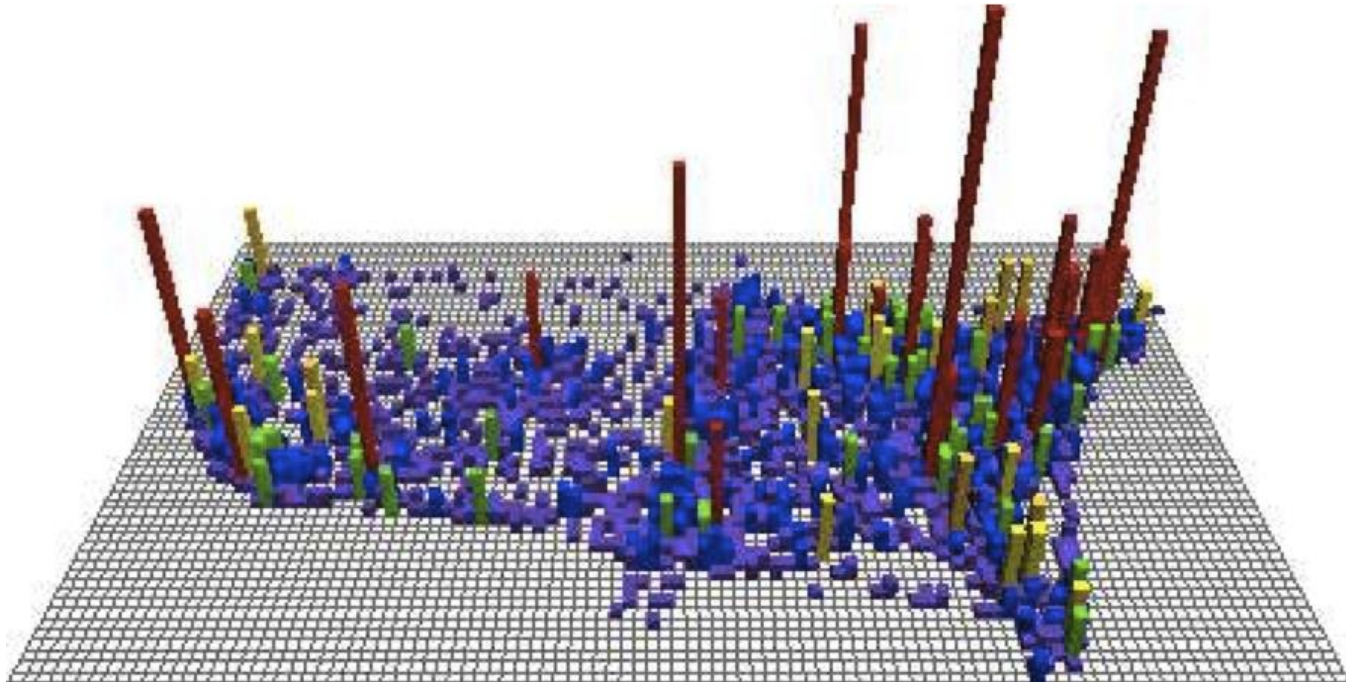
Reliéf (rubber sheet)

- Data mapována na výšku bodu v 3D prostoru + triangulace těchto bodů



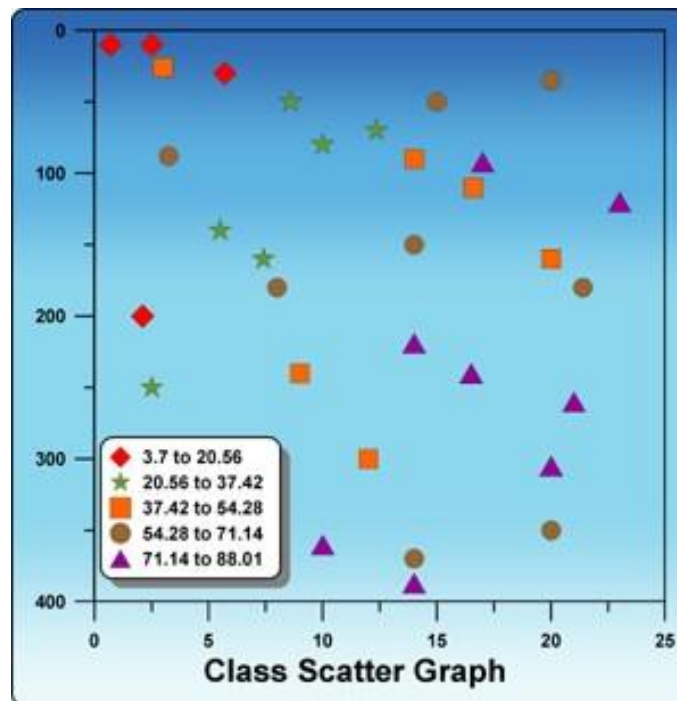
Cityscape

- Vykreslení 3D kvádrů do roviny, data mapována na jejich atributy (výška, barva, ...)



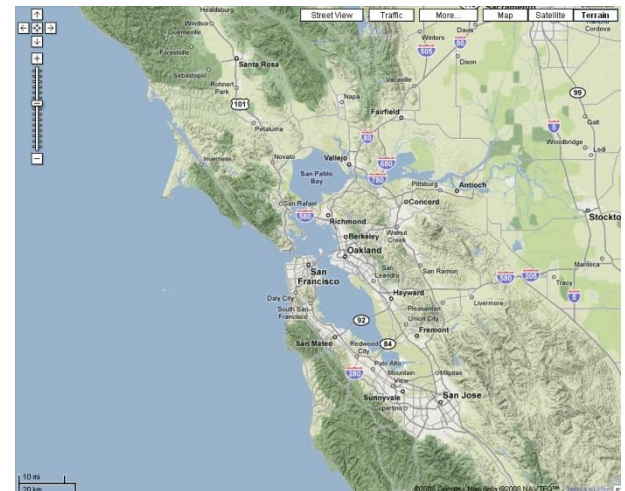
Bodový graf (scatterplot)

- Každá datová položka ovlivňuje barvu, tvar či velikost příslušné značky
- Žádná interpolace



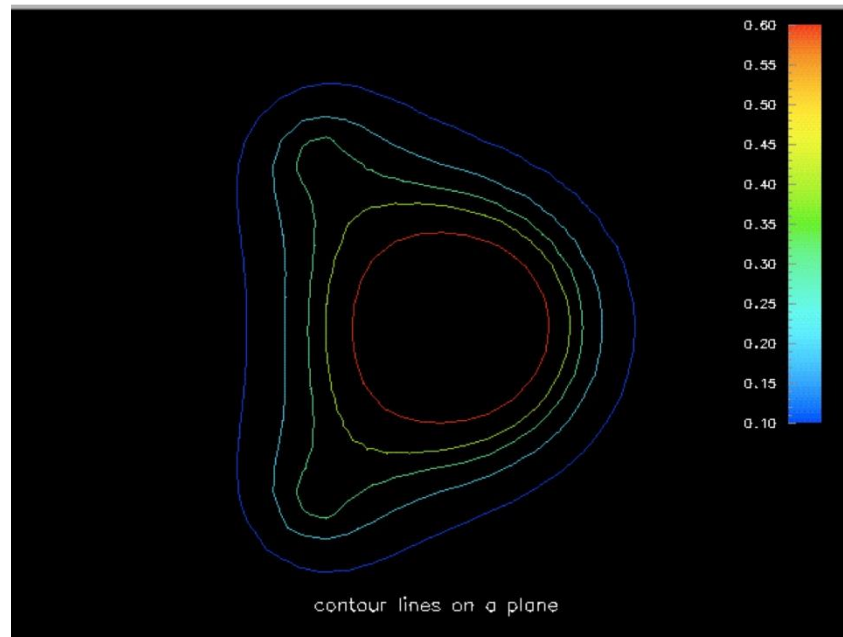
Mapa

- Lineární objekty – spojité úsečkové segmenty (řeky, silnice)
- Plošné objekty – uzavřené polygony s barvou, texturou, ... (jezera, státy)
- Bodové objekty – specifické symboly (škola, kostel)
- Označení (labels)



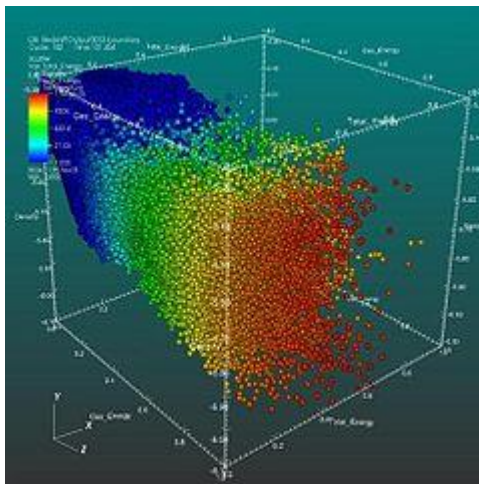
Kontury, izobary

- Hraniční informace představující spojitý jev (nadmořská výška, teplota)
- Určuje hranici mezi body s vyšší a nižší hodnotou

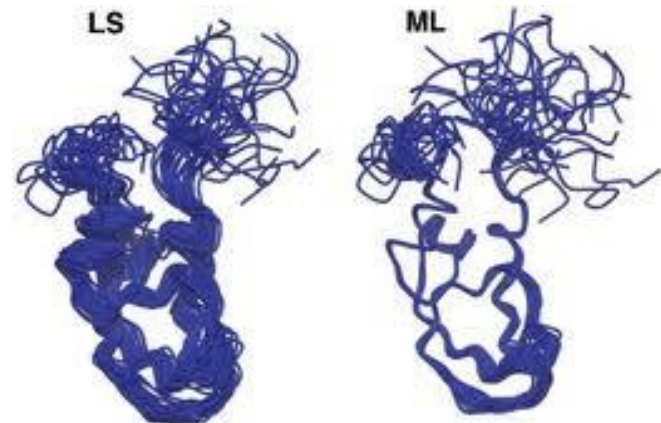


2D multivariate data

- Juxtapositioning
 - Naskládání 2D vizualizací do 3D
- Superimpositioning
 - Překládání přes sebe (např. průhledné reliéfy)
- Obě limitovány počtem proměnných



en.wikipedia.org



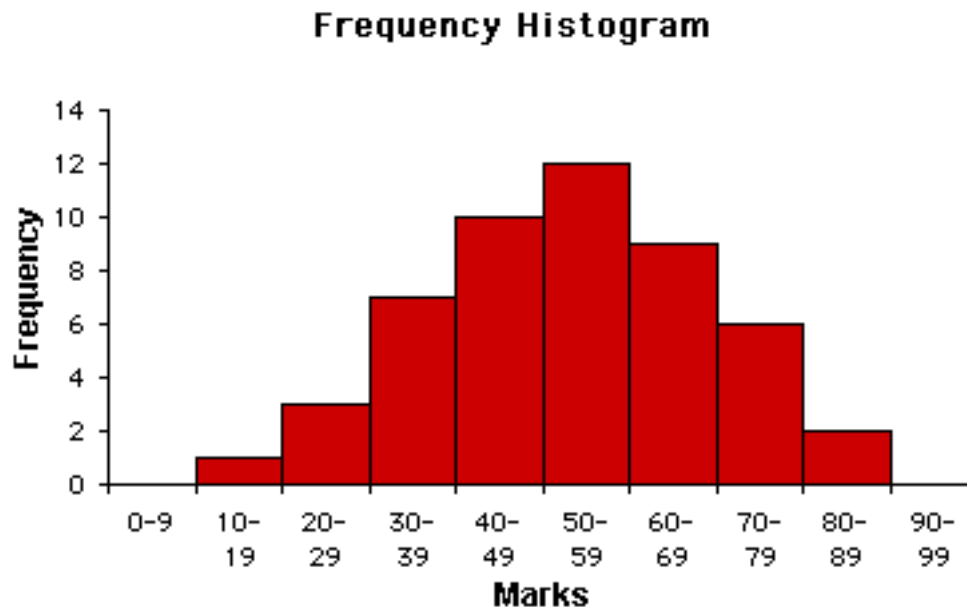
boscoh.com

Studie 2D dat

- Vizualizace podmnožin vstupních dat, projekce, sumarizace – za účelem zjednodušení
- Poté využití předchozích technik
- Techniky projekce:
 - Frekvenční histogramy
 - Slučování řádků a sloupců
 - Lineární „sondy“

Frekvenční histogramy

- Výpočet frekvence, ve které se dané hodnoty nebo podmnožiny hodnot objevují v datech
- Výsledek zobrazen v podobě sloupcového grafu



Slučování řádků a sloupců

- Lokalizace hranic oblastí zájmu a regionů s nízkou či vysokou variabilitou
- Slučování součtem, průměrem, mediánem, standardní odchylkou, maximem, minimem
- Barevné sloupce, čárové grafy, sloupcové grafy

Lineární „sondy“

- Přímka (sonda) procházející skrz vstupní data
- Využití parametrických rovnic a bilineární interpolace
- Definována pomocí dvou bodů P_1 a P_2 nebo pomocí bodu a směrového vektoru
- Parametrická rovnice přímky:

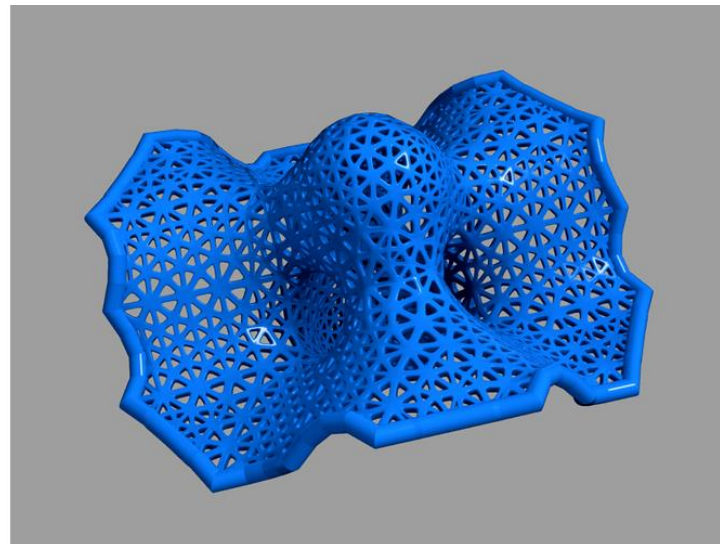
$$P(t) = P_1 + t(P_2 - P_1), \text{ kde } 0 \leq t \leq 1.0$$

3D data

- Diskrétní vzorky spojitého jevu nebo jako vrcholy, hrany a polygony
- Většinou kombinace obojího
- Základní techniky vizualizace 3D dat:
 - Vizualizace explicitních povrchů
 - Vizualizace objemových dat
 - Implicitní povrchy

Vizualizace explicitních povrchů

- Definován jako:
 - Seznam 3D vrcholů, seznam hran, seznam planárních polygonů
 - Sada parametrických rovnic definujících x , y , z souřadnic bodů, strategie jejich spojování (hrany, polygony)



Příklad

vertex[0] = (0., 0., 0.)

vertex[1] = (0., 0., 1.)

vertex[2] = (0., 1., 1.)

vertex[3] = (0., 1., 0.)

vertex[4] = (1., 0., 0.)

vertex[5] = (1., 0., 1.)

vertex[6] = (1., 1., 1.)

vertex[7] = (1., 1., 0.)

edge[0] = (0, 1)

edge[1] = (1, 2)

edge[2] = (2, 3)

edge[3] = (3, 0)

edge[4] = (0, 4)

edge[5] = (1, 5)

edge[6] = (2, 6)

edge[7] = (3, 7)

edge[8] = (4, 5)

edge[9] = (5, 6)

edge[10] = (6, 7)

edge[11] = (7, 4)

face[0] = (0, 1, 2, 3)

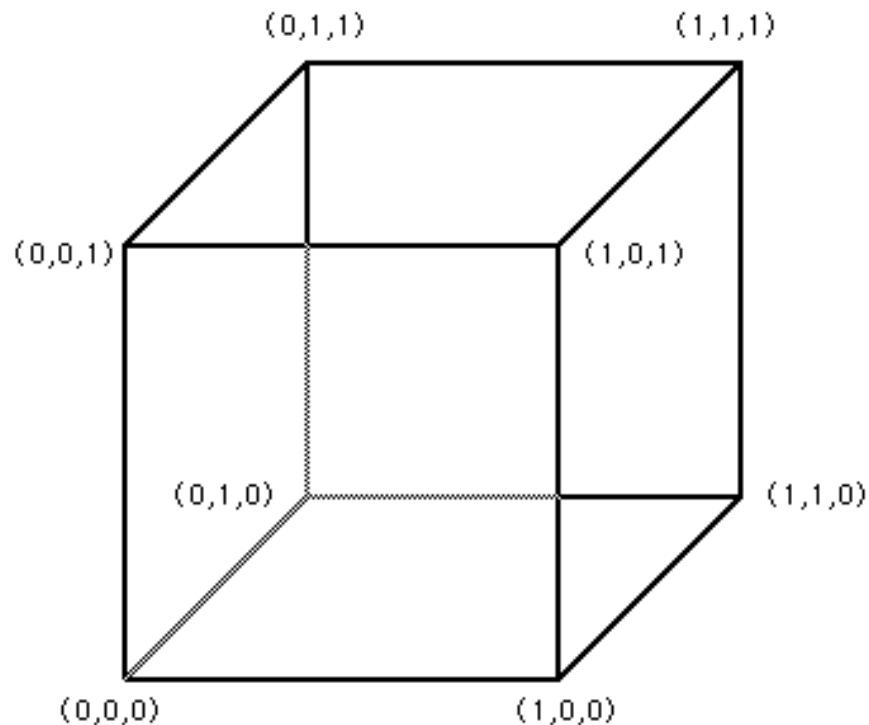
face[1] = (8, 9, 10, 11)

face[2] = (0, 5, 8, 4)

face[3] = (1, 6, 9, 5)

face[4] = (2, 7, 10, 6)

face[5] = (3, 4, 11, 7)



Příklad – jednotkový válec v ose y

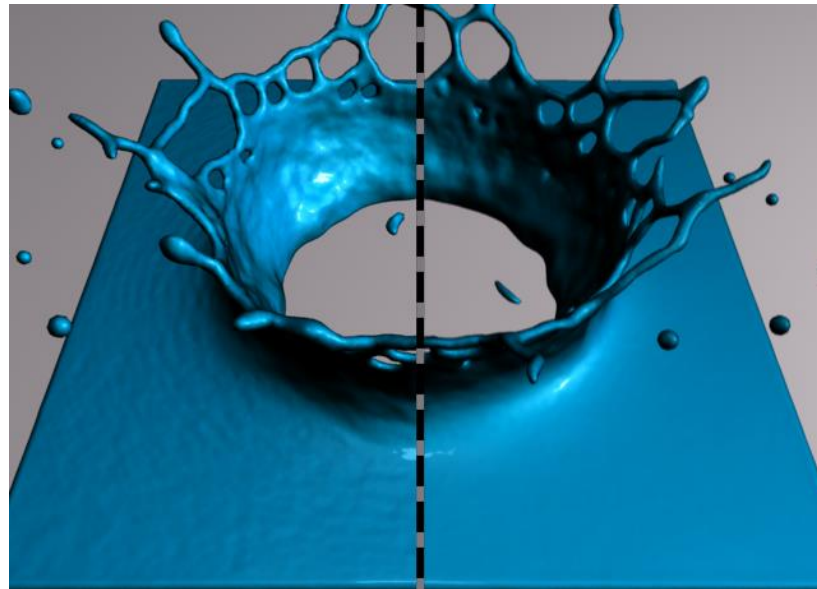
$$y = 1.0, \quad x = \cos \Theta, \quad z = \sin \Theta, \\ 0.0 \leq \Theta \leq 2\pi \quad (\text{horní podstava})$$

$$y = 0.0, \quad x = \cos \Theta, \quad z = \sin \Theta, \\ 0.0 \leq \Theta \leq 2\pi \quad (\text{dolní podstava})$$

$$y = h, \quad x = \cos \Theta, \quad z = \sin \Theta, \\ 0.0 \leq \Theta \leq 2\pi, \quad 0.0 \leq h \leq 1.0 \quad (\text{plášť})$$

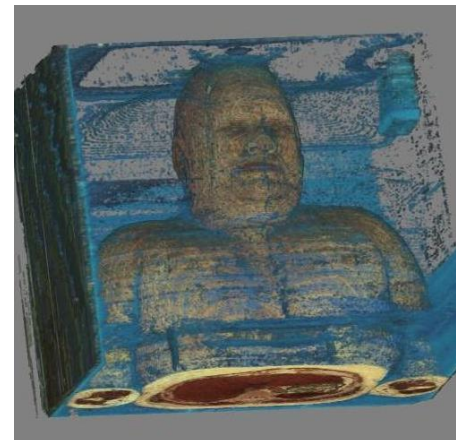
Příklady

- Vstupní data jsou asociována s:
 - vrcholy – teplota, váha uzlu
 - hranami – síla chemické vazby
 - stěnami – pokrytí oblasti mapou

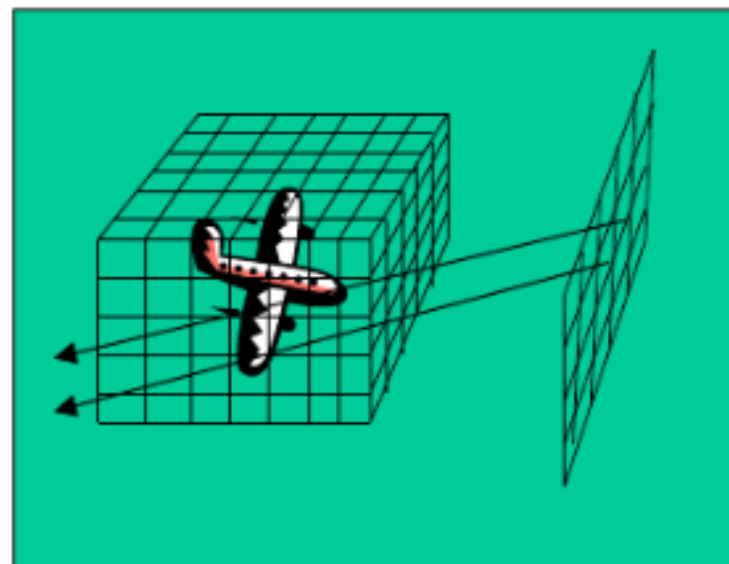


Vizualizace objemových dat

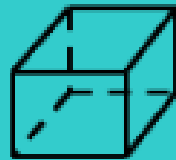
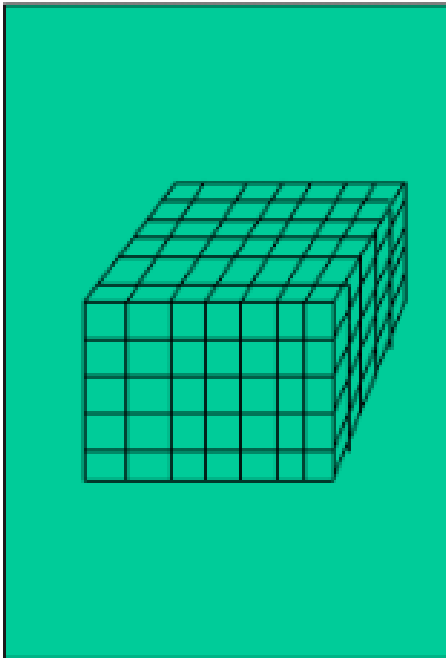
- Využití voxelů
- Kategorie:
 - Slicing („plátování“) – využití ořezávací roviny
 - Isosurfaces – generování povrchu
 - Direct volume rendering –
vrhání paprsku nebo
promítnutí voxelů na
projekční rovinu



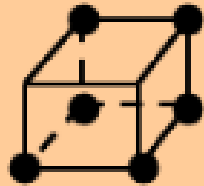
vidi.cs.ucdavis.edu



Voxel



A voxel is a cubic cell, which has a single value cover the entire cubic region



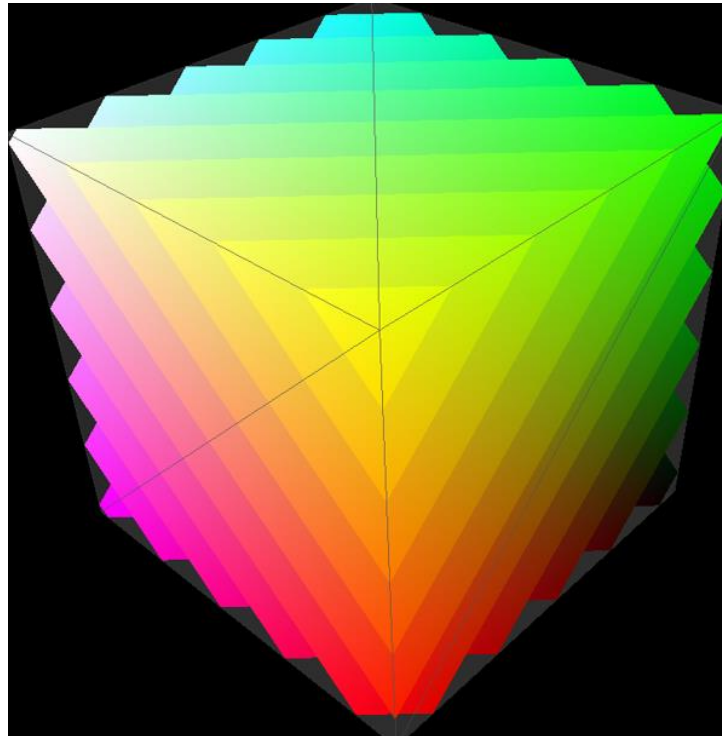
A voxel is a data point at a corner of the cubic cell
The value of a point inside the cell is determined by interpolation

Převzorkování

- Hraje významnou roli ve většině technik vizualizace objemových dat
 - Izopovrchy
 - Slicing
 - Direct volume rendering (výjimka je pouze u paralelní projekce podél hlavních os)

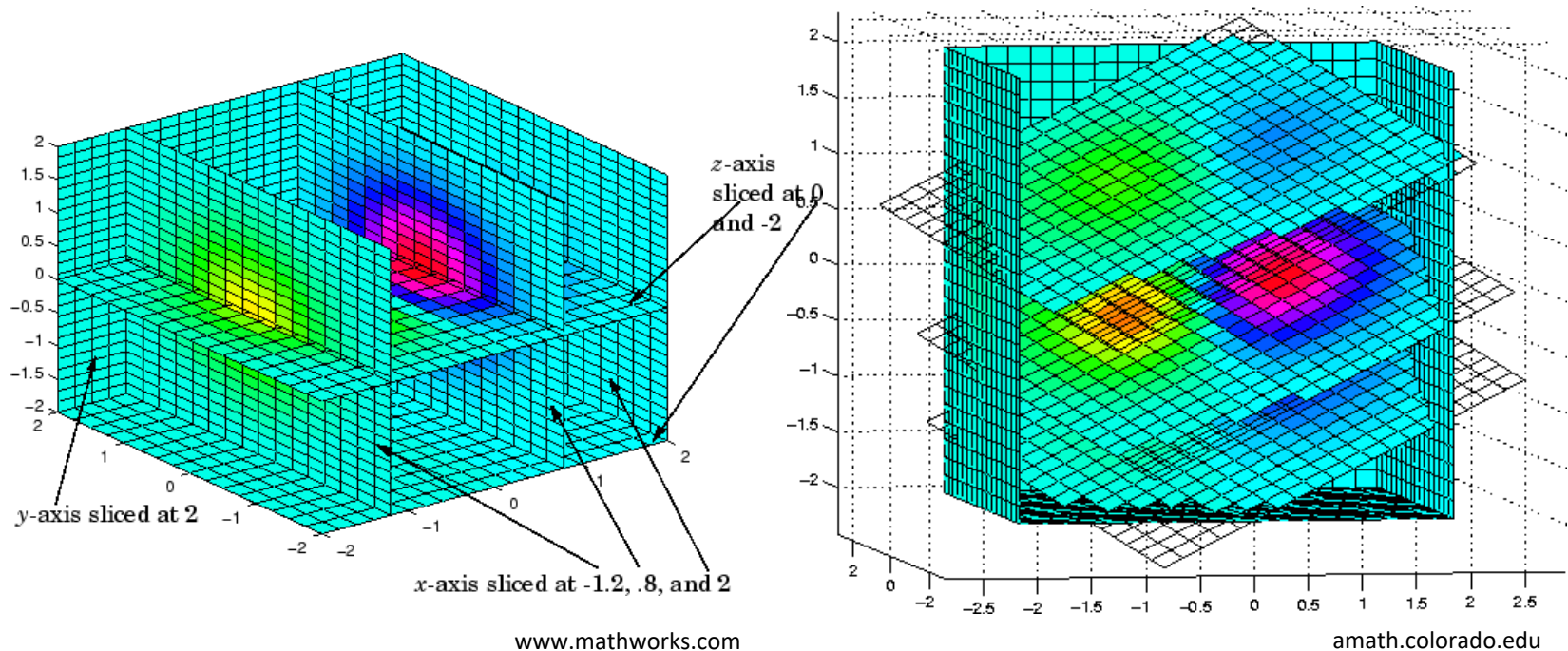
Plátování objemových dat ořezávacími rovinami

- Technika vytvoření podmnožiny vstupních dat o nižší dimenzi



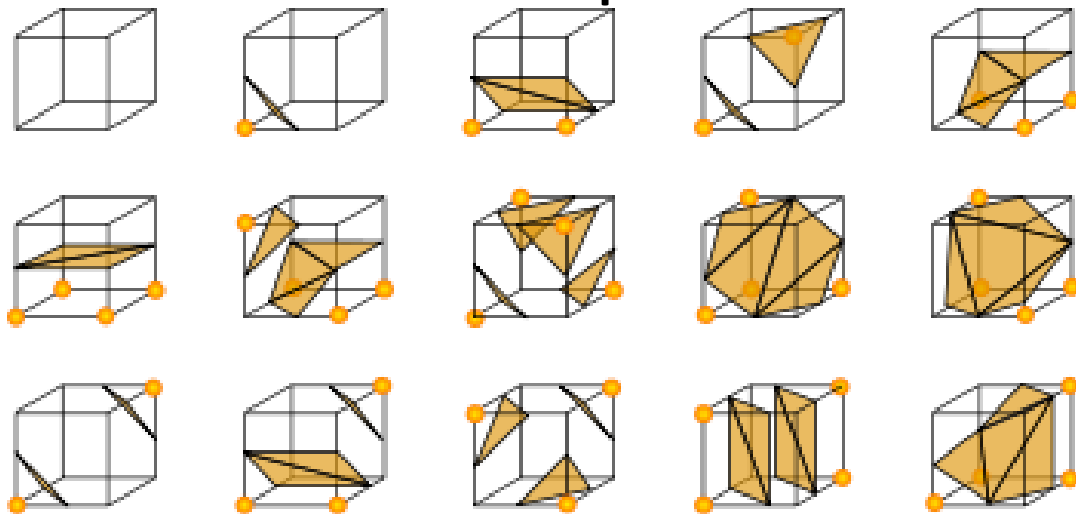
Plátování objemových dat ořezávacími rovinami

- Orientace ořezávací roviny
 - Normála roviny shodná s některou s hlavních os
 - Libovolná orientace



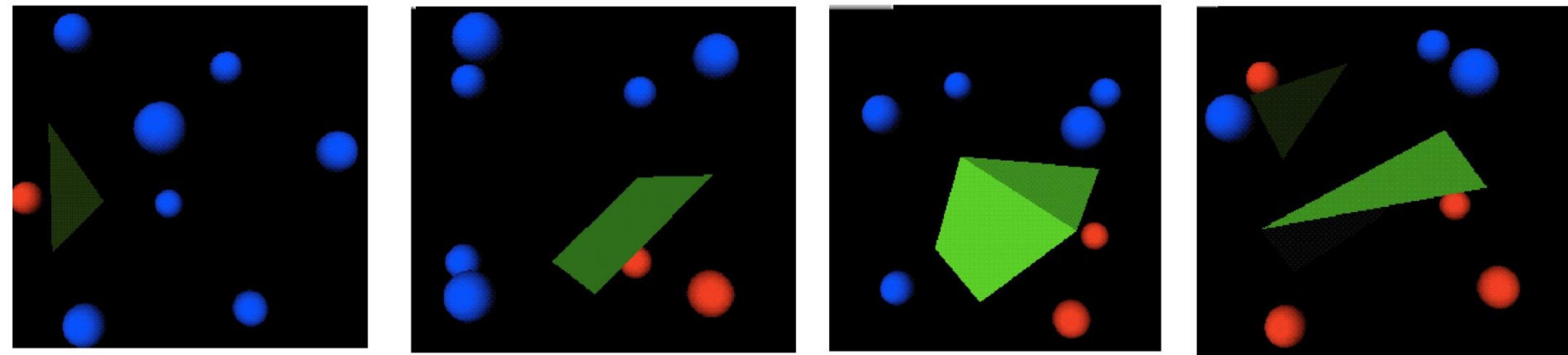
Získání izopovrchu pomocí Marching Cubes

- Algoritmus „pochodujících kostek“ – Lorensen, Cline (1987)
- Voxel = kostka s vrcholy
- Algoritmus vytváří trojúhelníky na základě příslušnosti vrcholů k izopovrchu

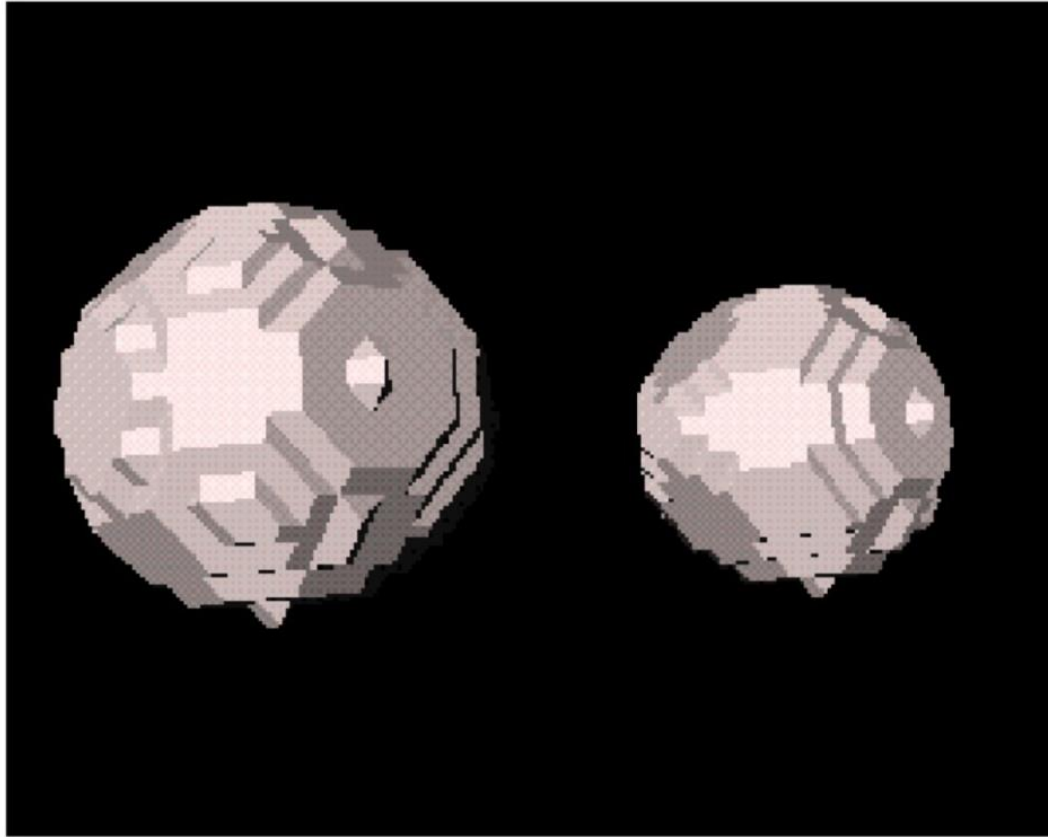


Marching Cubes - details

- 256 konfigurací, díky symetrii pouze 16 unikátních (1 = celá uvnitř, 1 = celá vně)
- Generování příslušných trojúhelníků



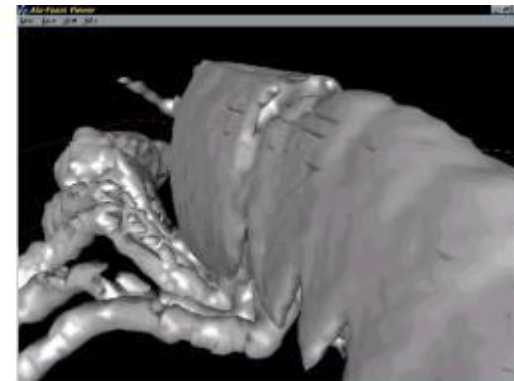
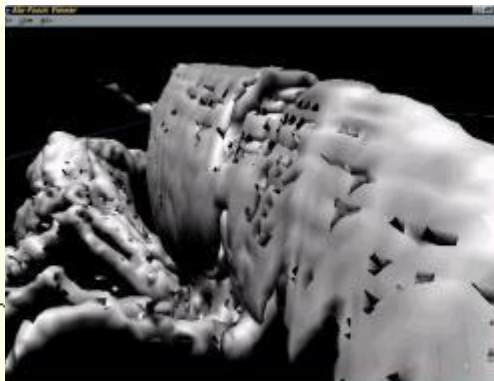
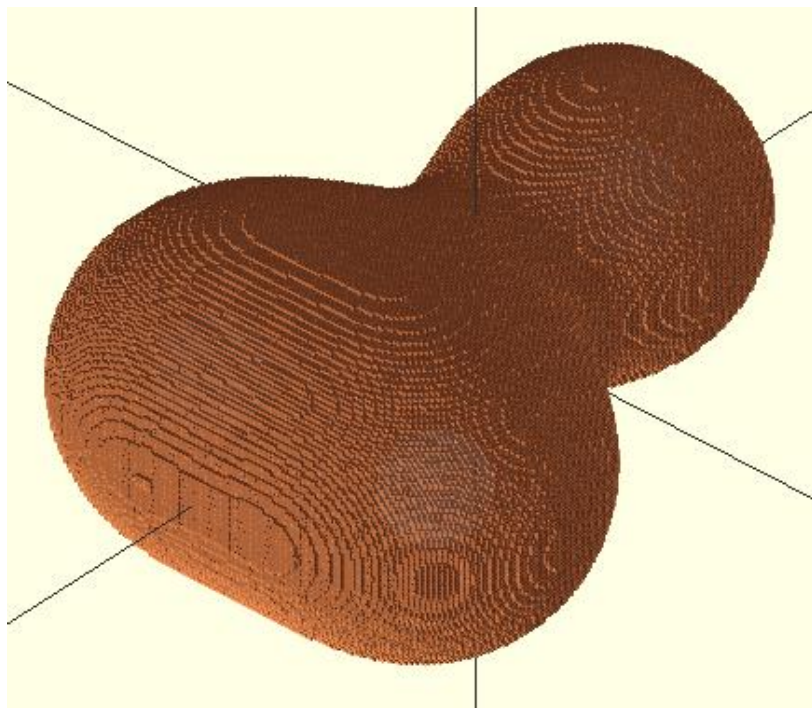
Marching Cubes - detail



(<http://www.opendx.org>)

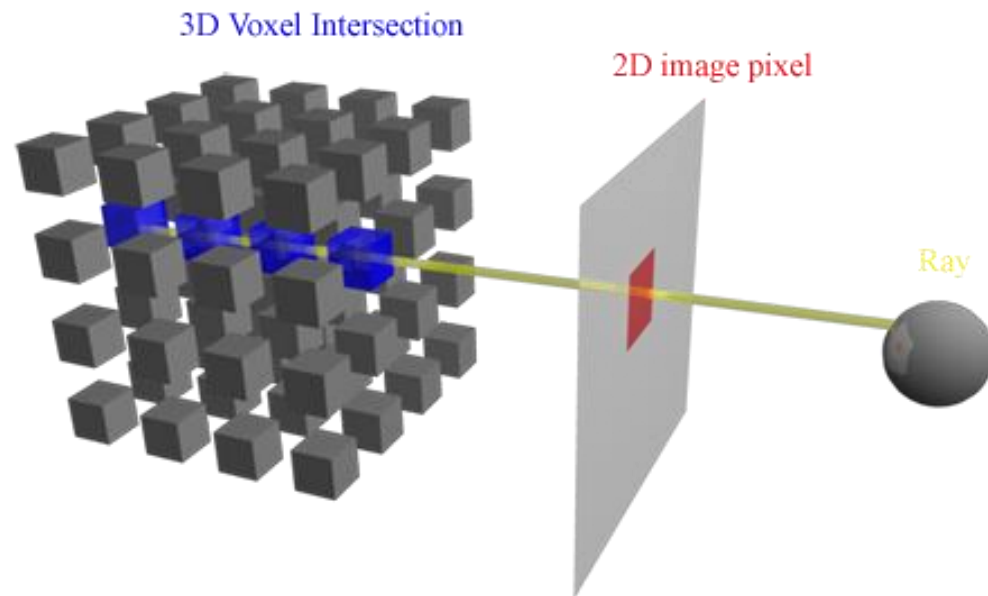
Marching Cubes - problémy

- Paměťová náročnost
- Díry v datech – nekvalitním nasnímáním



Techniky přímé vizualizace objemu

- Pixely výsledného obrazu spočteny individuálně – pomocí vrhání paprsku do scény nebo pomocí projekce voxelů
- Metody:
 - Dopředné mapování
 - Inverzní mapování (vrhání paprsku)



Dopředné mapování - problémy

- F1: Jak zacházet s pixely, které jsou ovlivněny více voxely
- F2: Jak zacházet s pixely, na něž není mapován žádný voxel
- F3: Jak se vypořádat s faktem, že voxely se obvykle promítají na pozice mezi pixely

Inverzní mapování - problémy

- I1: Jak zvolit počet bodů, které budou podél paprsku vzorkovány
- I2: Jak spočítat hodnotu v těchto bodech, které často padnou mezi voxely
- I3: Jak zkombinovat body, které paprsek na své cestě protnul

Řešení

- F2 a F3: Mapování každého voxelu na region projekční roviny, voxel pak částečně ovlivňuje hodnotu několika sousedících pixelů
- I1: Určení rozestupu mezi pixely a nastavení vzorkovací frekvence na menší hodnotu, než jsou tyto rozestupy

Řešení

- F1 a I3: Technika zvaná „compositing“
 - Každý voxel má asociovanu hodnotu průhlednosti
 - Voxel i má barvu c_i a průhlednost o_i , pak jeho příspěvek k výsledné hodnotě pixelu je:

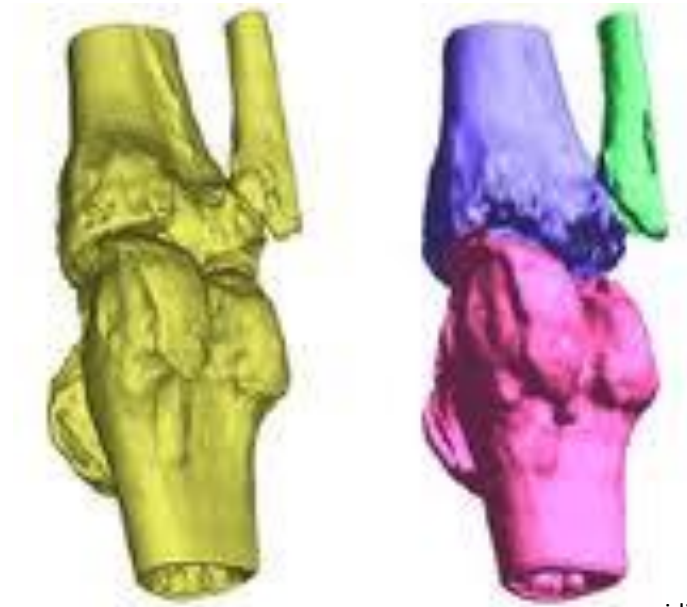
$$c_i * o_i * \prod_{j=0}^{i-1} (1 - o_j)$$

- Výsledná hodnota pixelu je pak určena jako:

$$I(x, y) = \sum_{i=0}^n c_i * o_i * \prod_{j=0}^{i-1} (1 - o_j)$$

Klasifikace

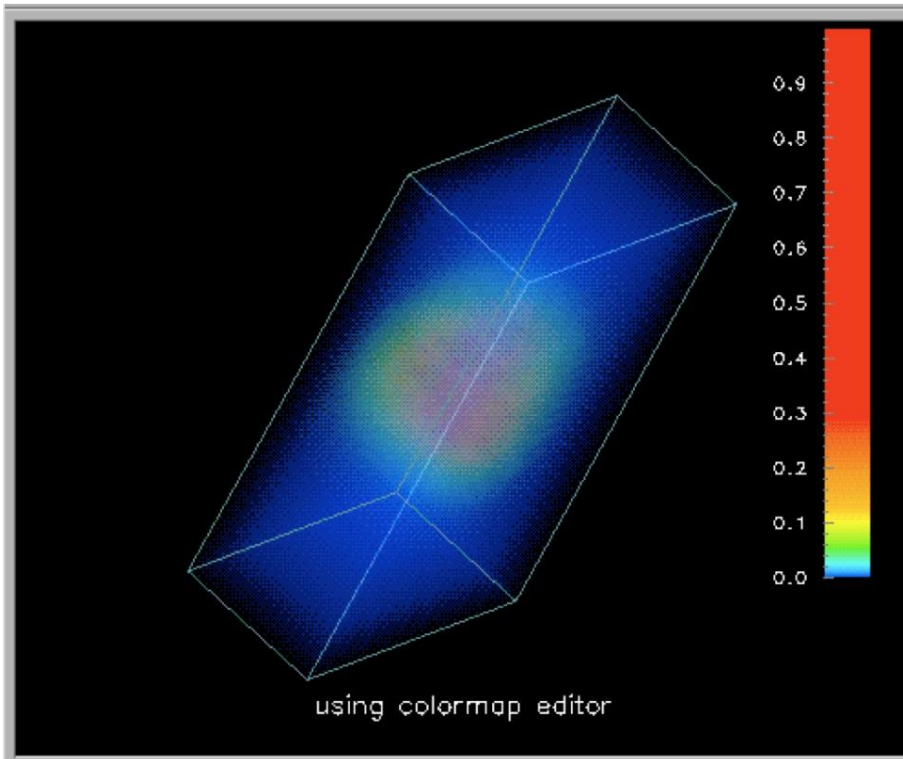
- Důležitým aspektem je určení průhlednosti a barvy přiřazené datovým hodnotám = klasifikace
- Nastavení oblastí zájmu – mohou mít jinou barvu a průhlednost



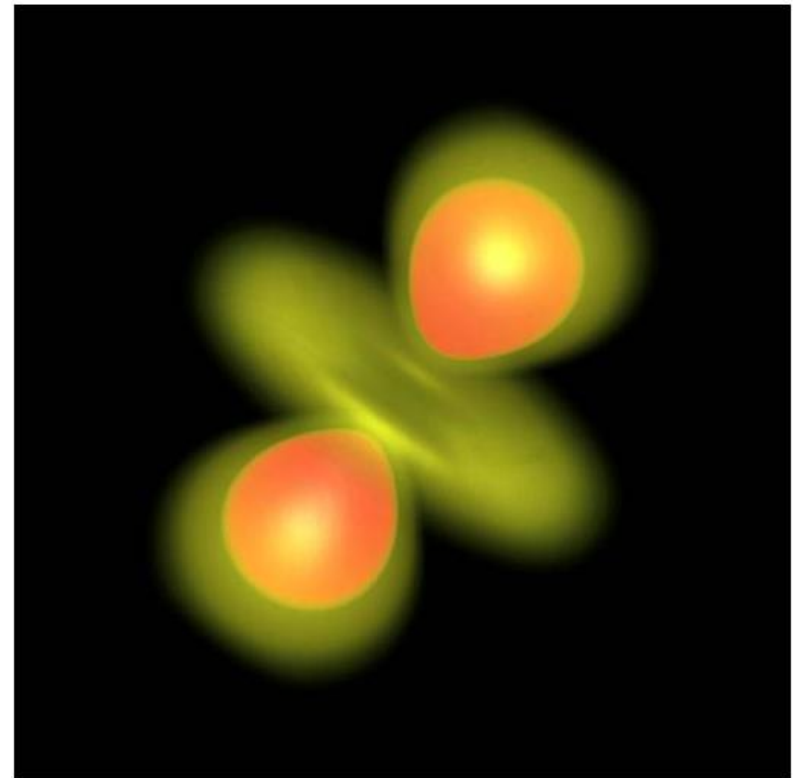
Výpočet osvětlení a stínování

- Žádné normály – nutné spočítat gradient:
 - např. pro daný voxel (v_x, v_y, v_z) je hodnota gradientu v ose x spočtena jako (v_{x-1}, v_y, v_z) a (v_{x+1}, v_y, v_z)
 - Mějme g_x označení pro x-ovou komponentu gradientu. Pak:
 - **Střední diferenční operátor**
$$V_x - V_{x-1}$$
 - **Centrální diferenční odhad gradientu**
$$V_{x+1} - V_{x-1}$$

Příklady



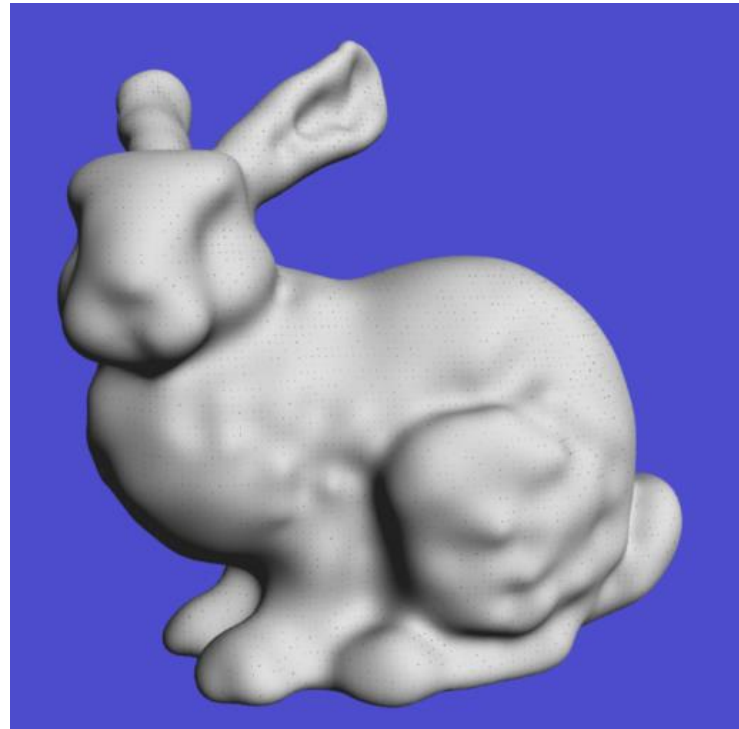
<http://www.opendx.org>



<http://old.vrvis.at/via/resources/PR-CBerger-2/index.html>

Implicitní povrchy

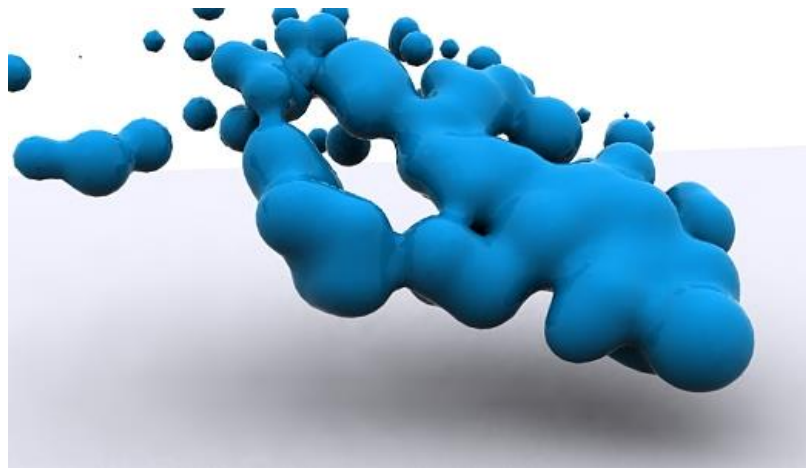
- Povrch je definován jako tzv. zero contour pro funkci o dvou nebo třech proměnných



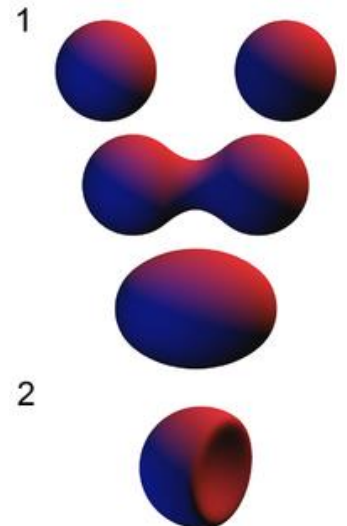
Metaballs

- Definice výpočtu použitého pro specifikaci vlivu libovolné částice na libovolný bod
- Nechť r je vzdálenost částice od daného místa v poli, b je výška Gaussovy hrbolati a a je její standardní odchylka, pak vliv částice je spočten jako:

$$b^{-ar}$$



www.pxleyes.com



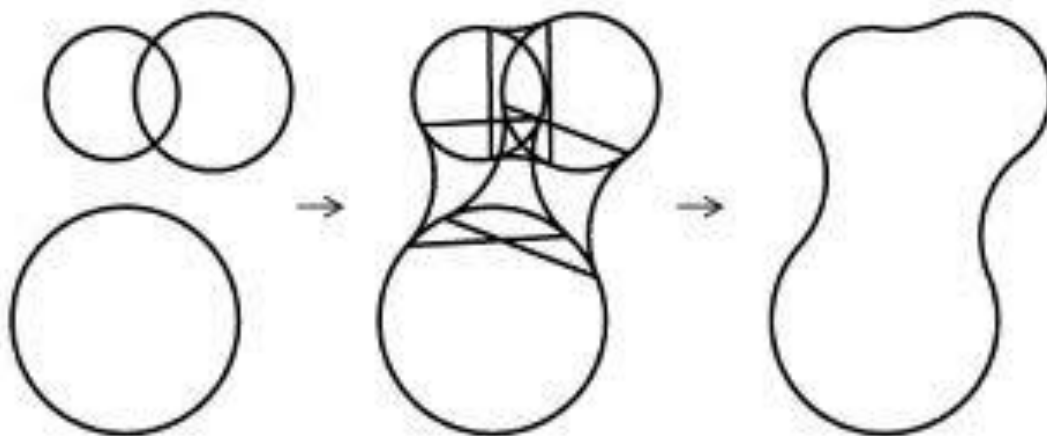
en.wikipedia.org

Metaballs

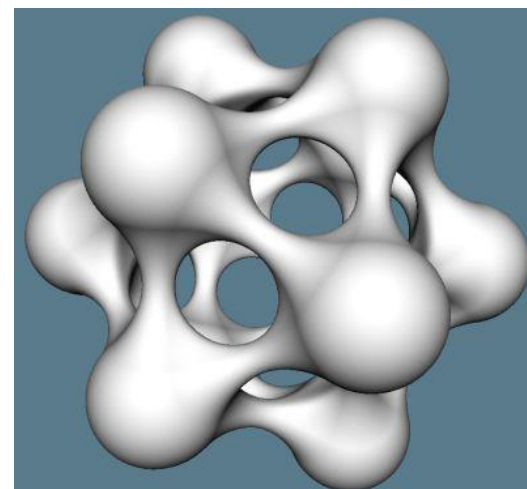
- Zavedení kubického polynomu založeného na poloměru vlivu dané částice a vzdálenosti od středu částice do dotazovaného místa
- Vliv = 1.0, pak $r = 0.0$
- Vliv = 0.0, pak $r = R$ (poloměr vlivu)

$$C(r) = 2r^3/R^3 - 3r^2/R^2 + 1$$

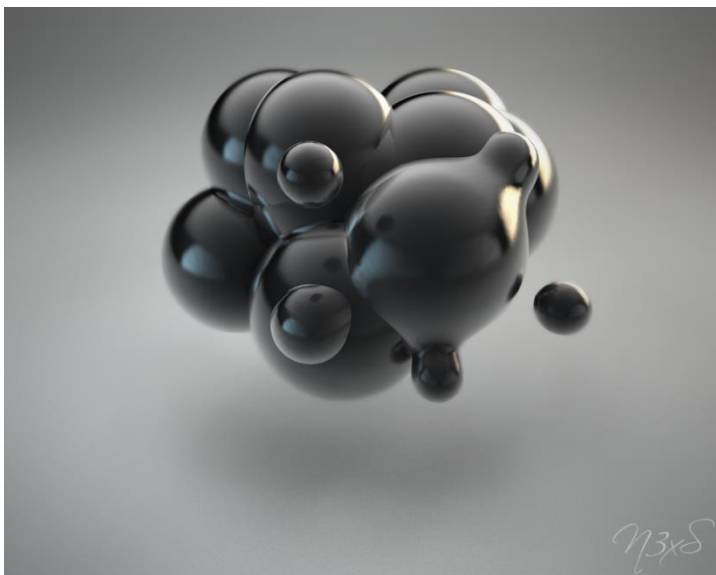
Metaballs



park12.wakwak.com

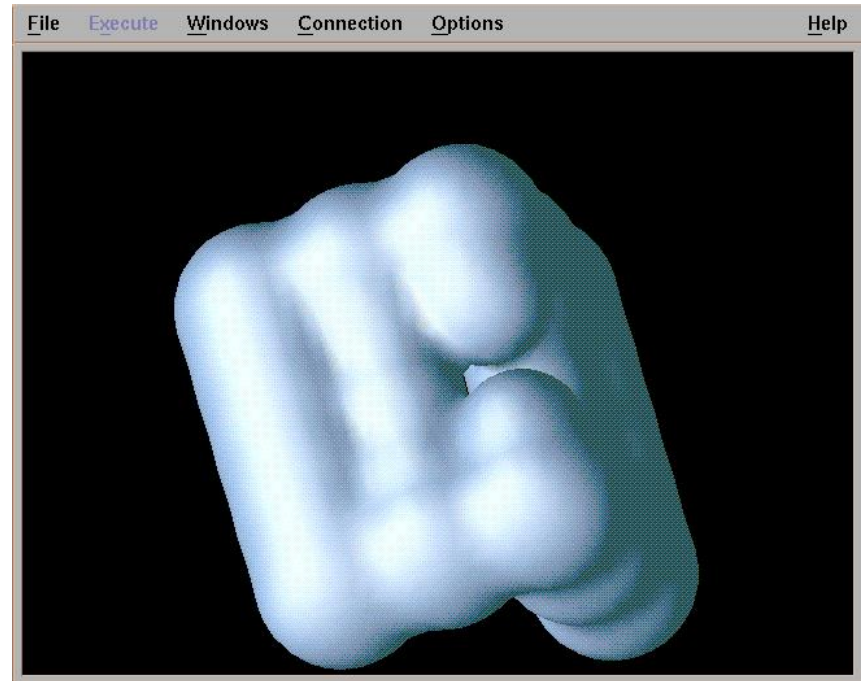
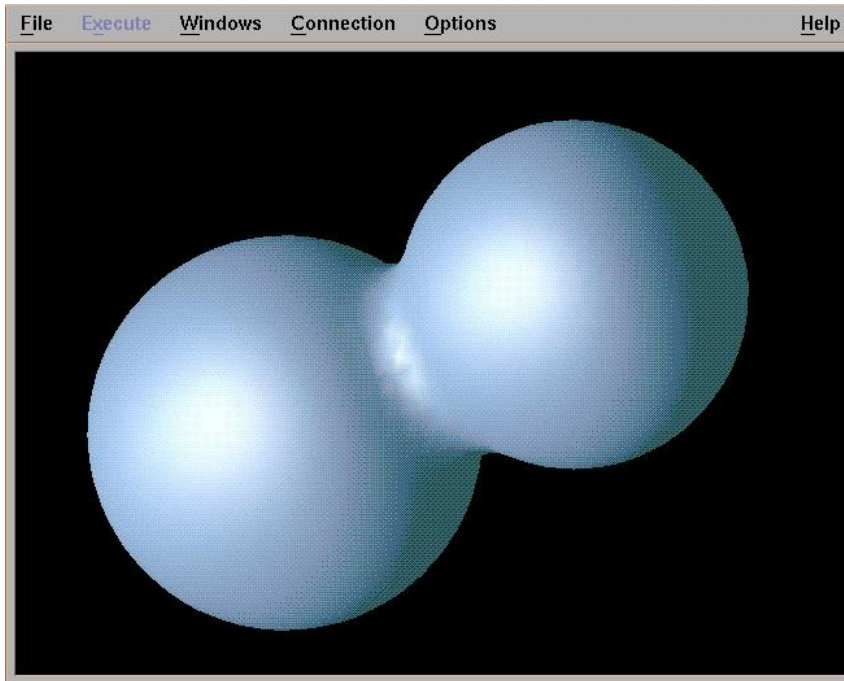


myweb.tiscali.co.uk



n3xs.deviantart.com

Implicitní povrchy



<http://www.opendx.org>

Dynamická data

- Flow visualization – metody zobrazení dynamického chování tekutin a plynů



Flow visualization

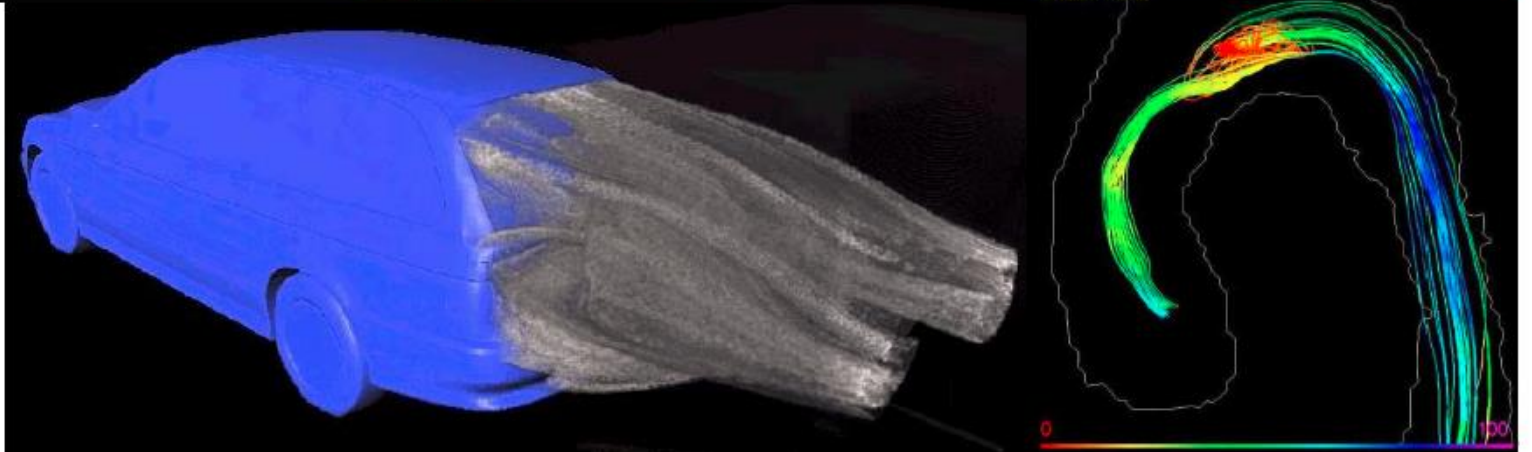
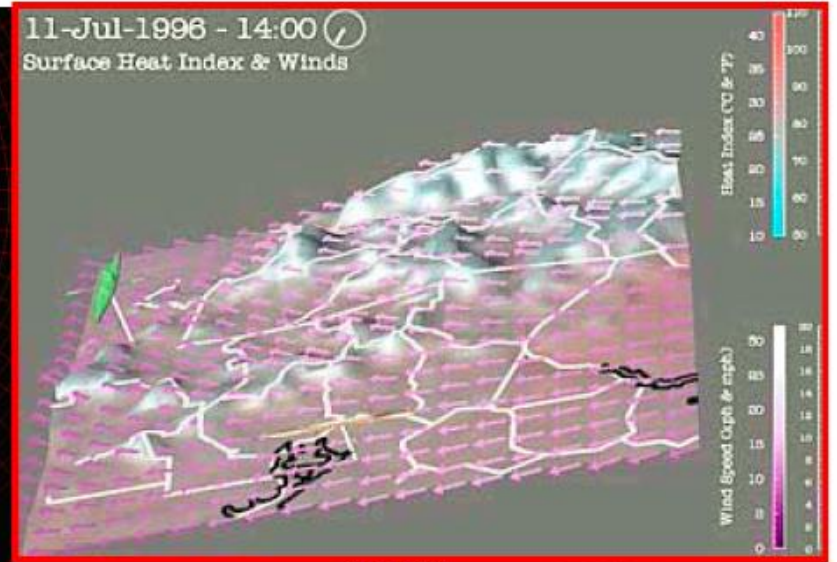
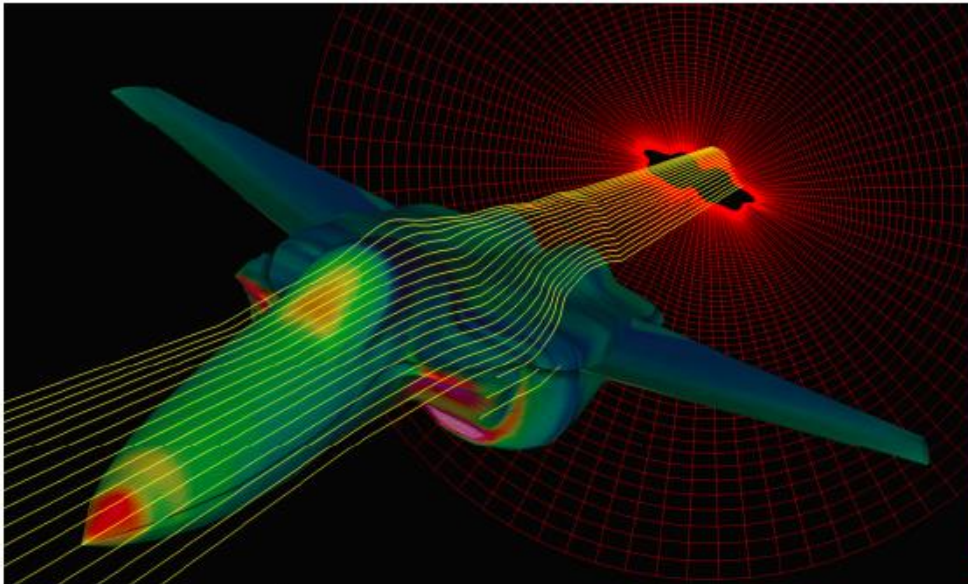
- Vizualizace změny informace
- Typicky více než 3 dimenze
- Cíle uživatele
 - Náhled na data vs. detaily
- Vstupní data:



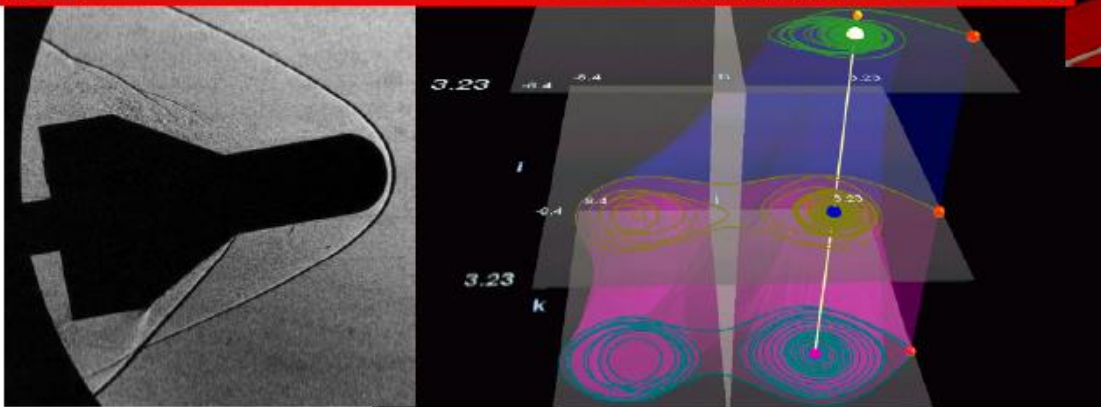
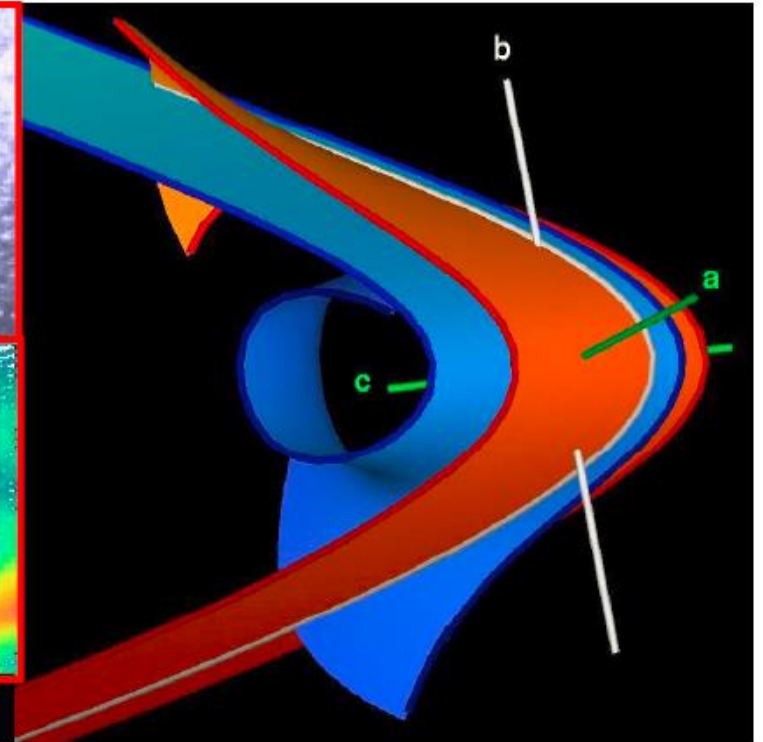
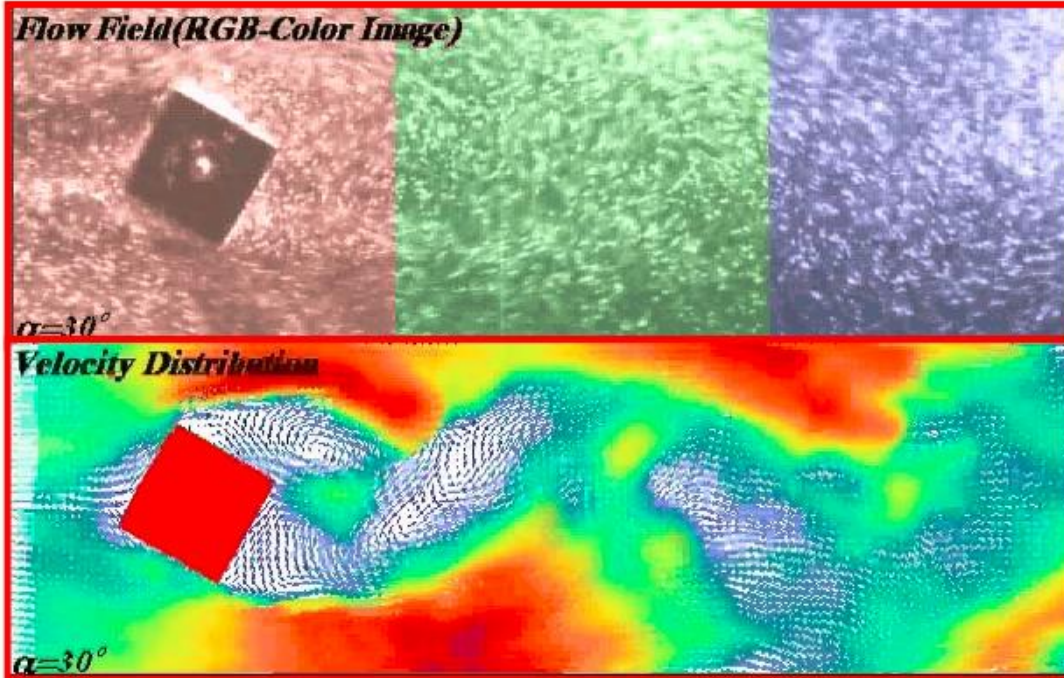
<http://en.wikipedia.org/wiki/File:Windkanal.jpg>

- Simulace – letecký, lodní, automobilový průmysl, počasí, medicína (tok krve), ...
- Měření – větrný tunel (aerodynamika)
- Modely – pomocí obyčejných diferenciálních rovnic

Příklady



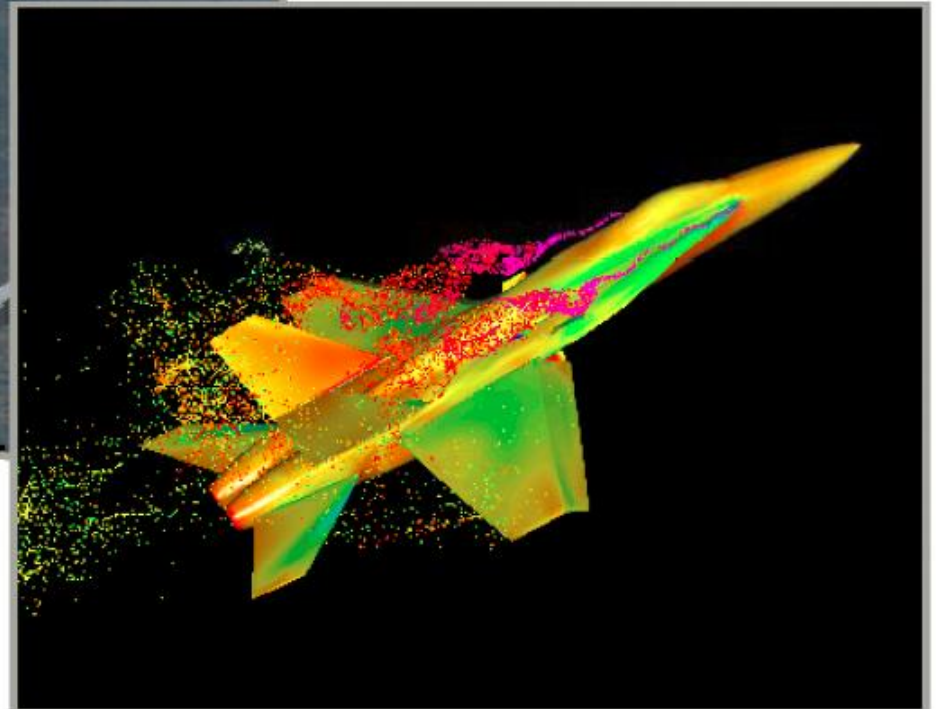
Příklady



Srovnání s reálnými experimenty



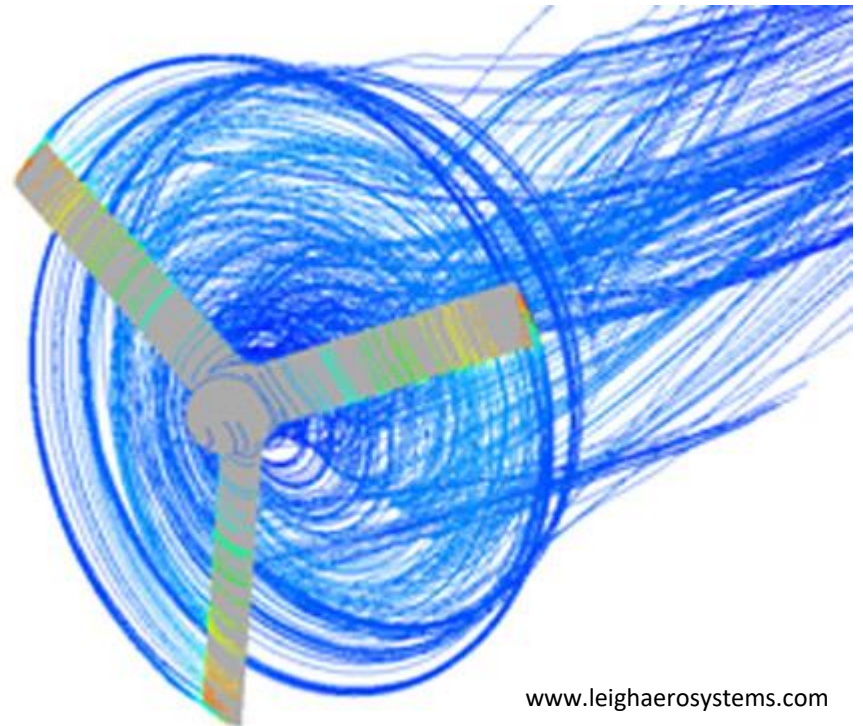
Experiment



Simulace

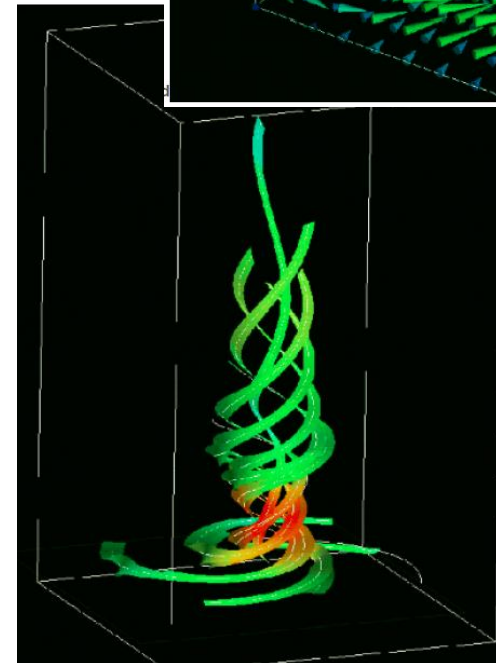
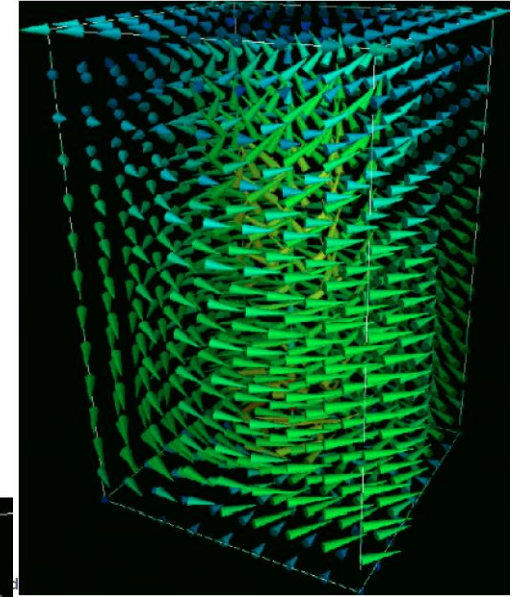
Dynamická data

- Computational Fluid Dynamics (CFD) – simulace dynamického chování tekutin pod širokou škálou podmínek
- Dvě varianty struktur generovaných dat:
 - Statické (časově nezávislé) pole
 - V čase se měnící (nestabilní) pole



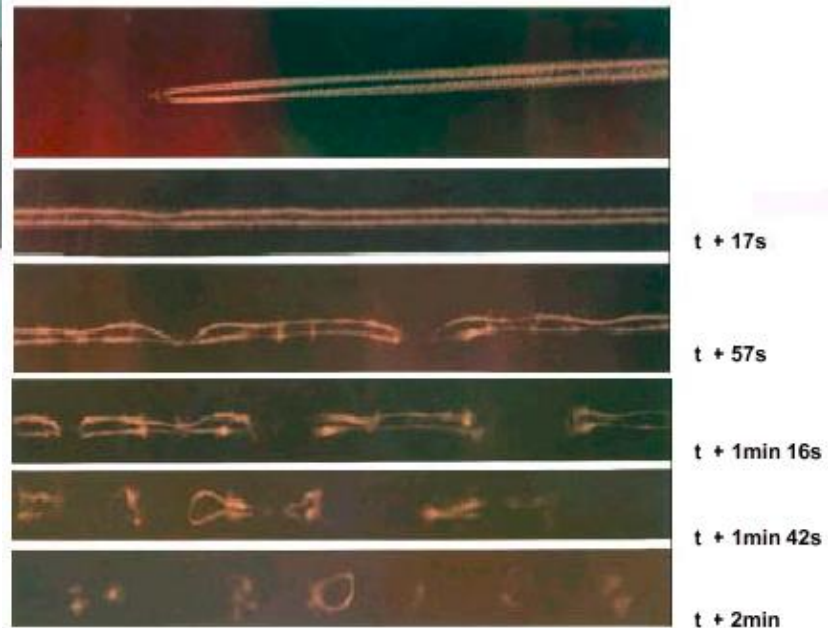
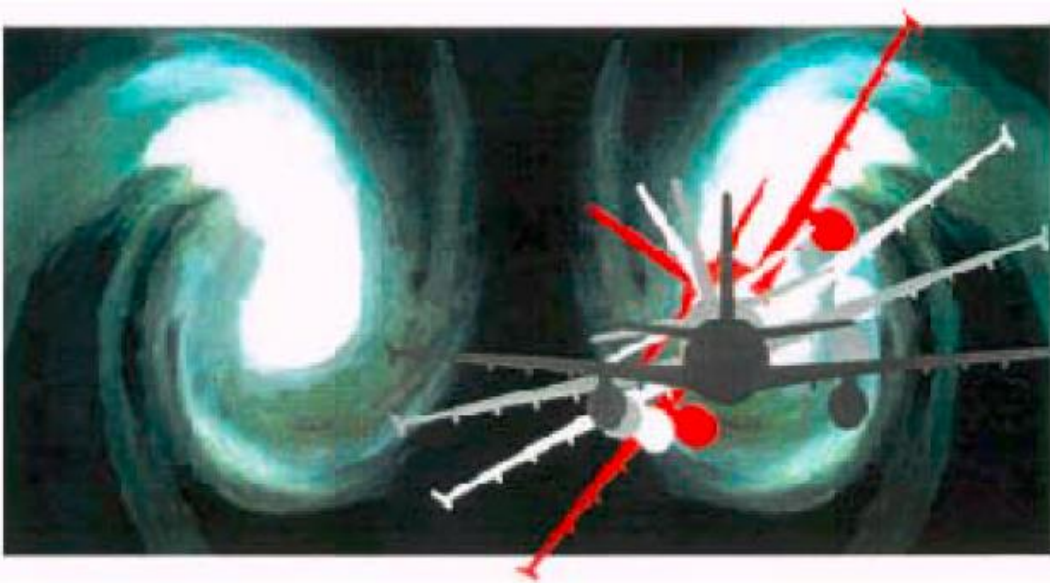
Přímá vs. nepřímá vizualizace toku

- Přímá
 - Náhled na aktuální stav toku
 - Vizualizace vektorů
- Nepřímá
 - Vizualizace vývoje toku v čase
 - Streamlines, streamsurfaces



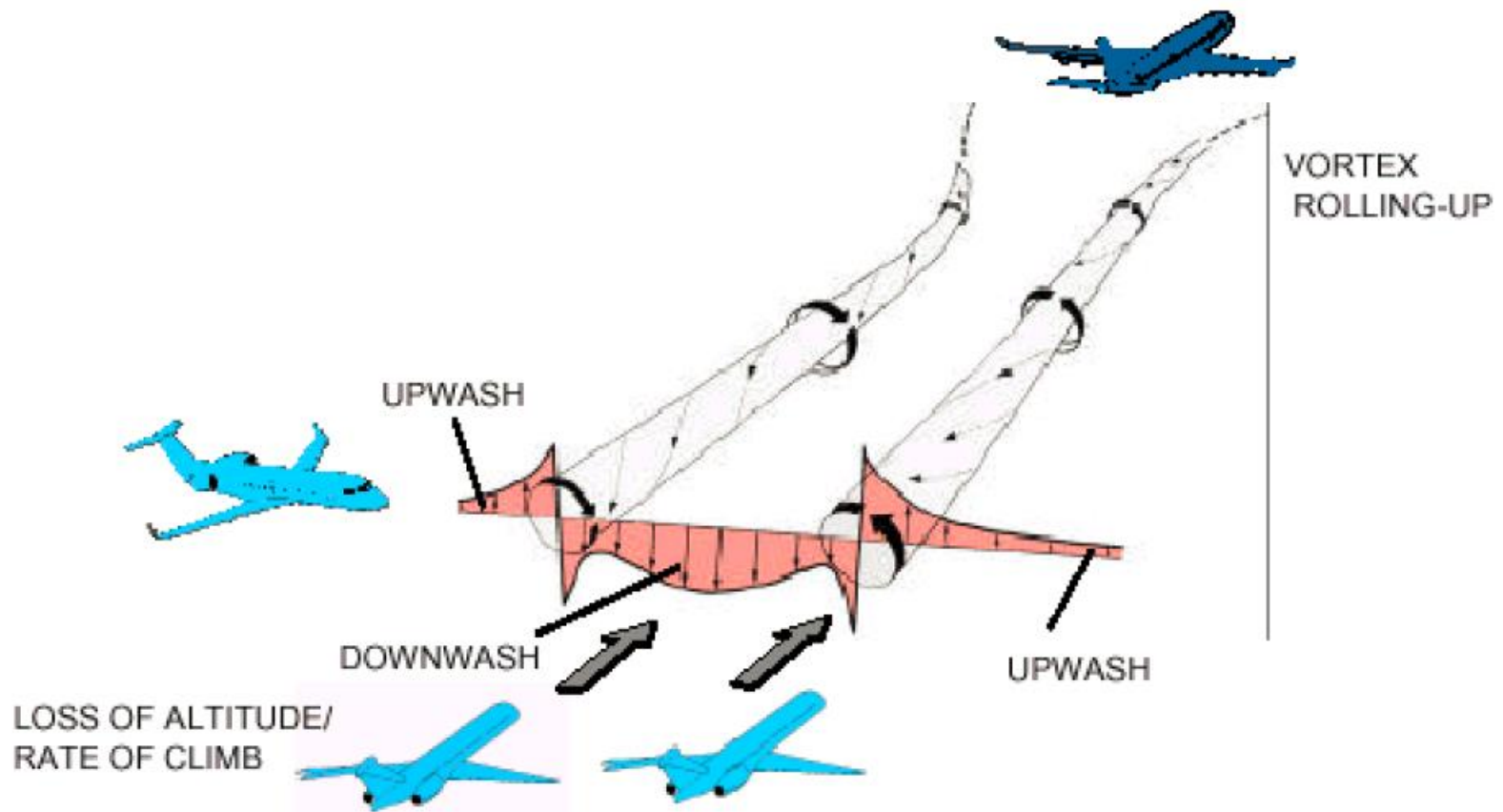
Příklad – Wind-Tip vír

- Problém: vznikající turbulence za letadlem



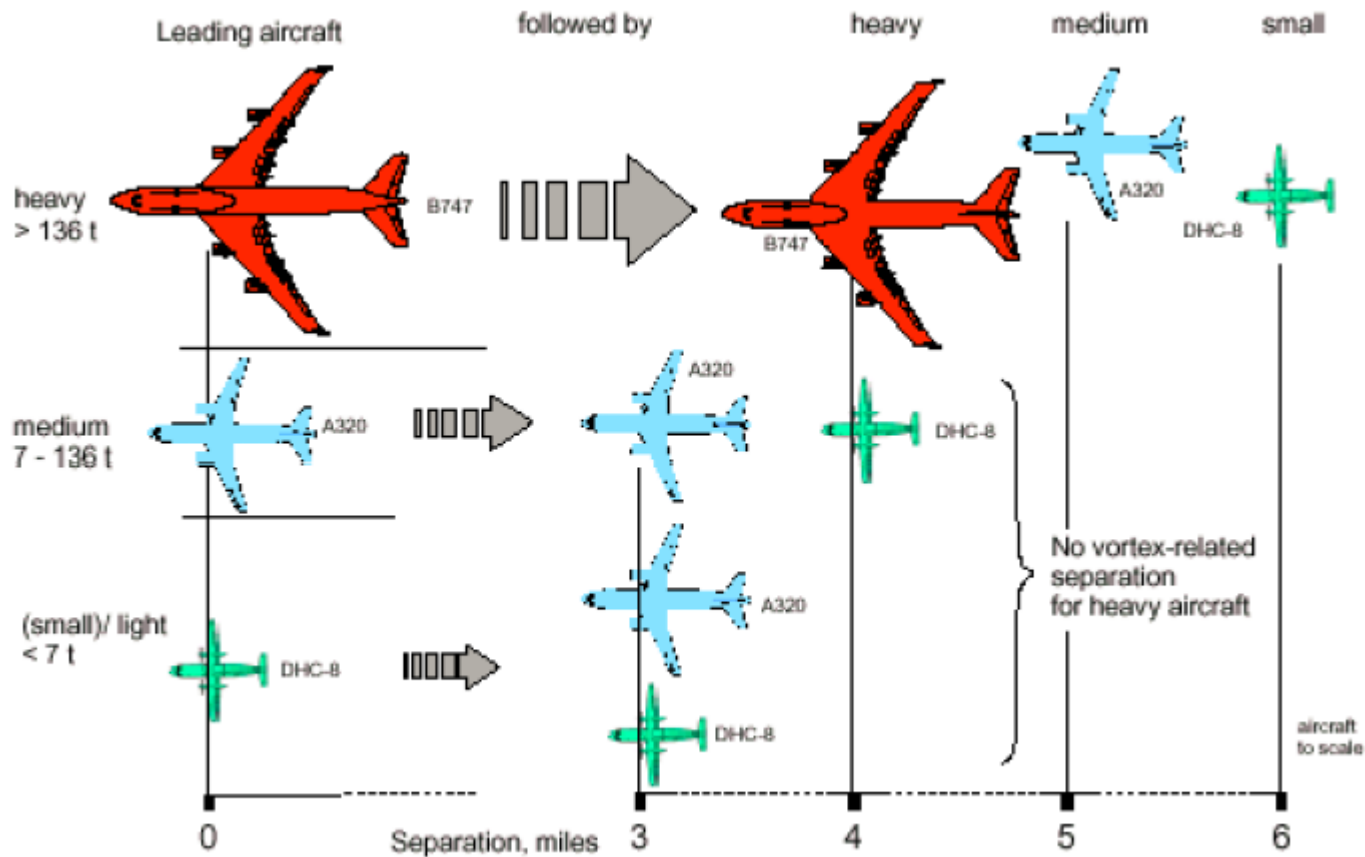
Příklad – Wind-Tip vír

- Vír je nebezpečný!



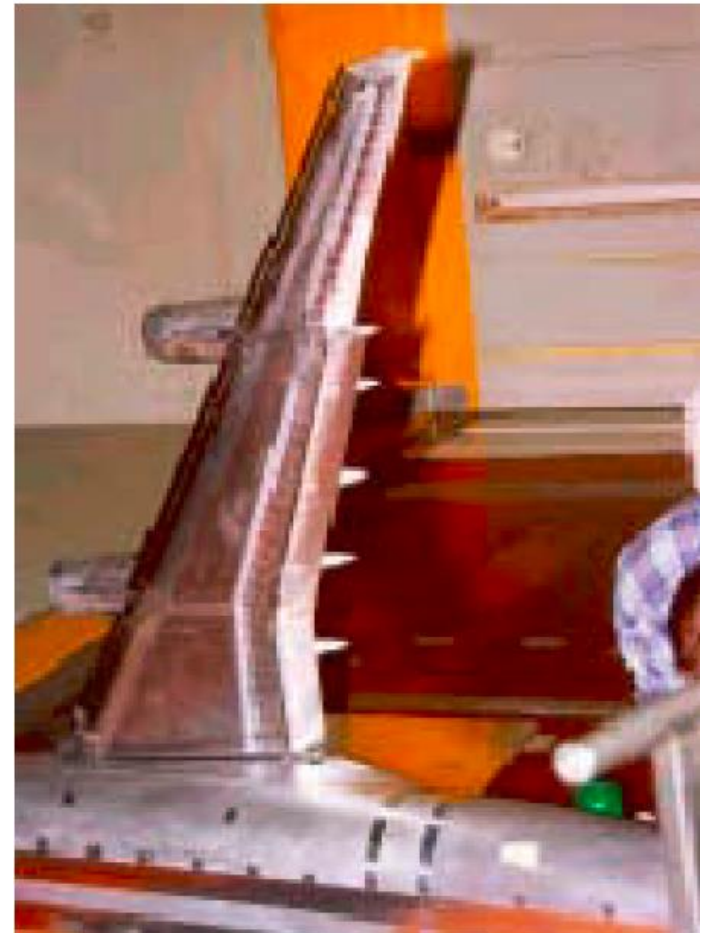
Příklad – Wind-Tip vír

- Proto nutné dodržovat určité vzdálenosti



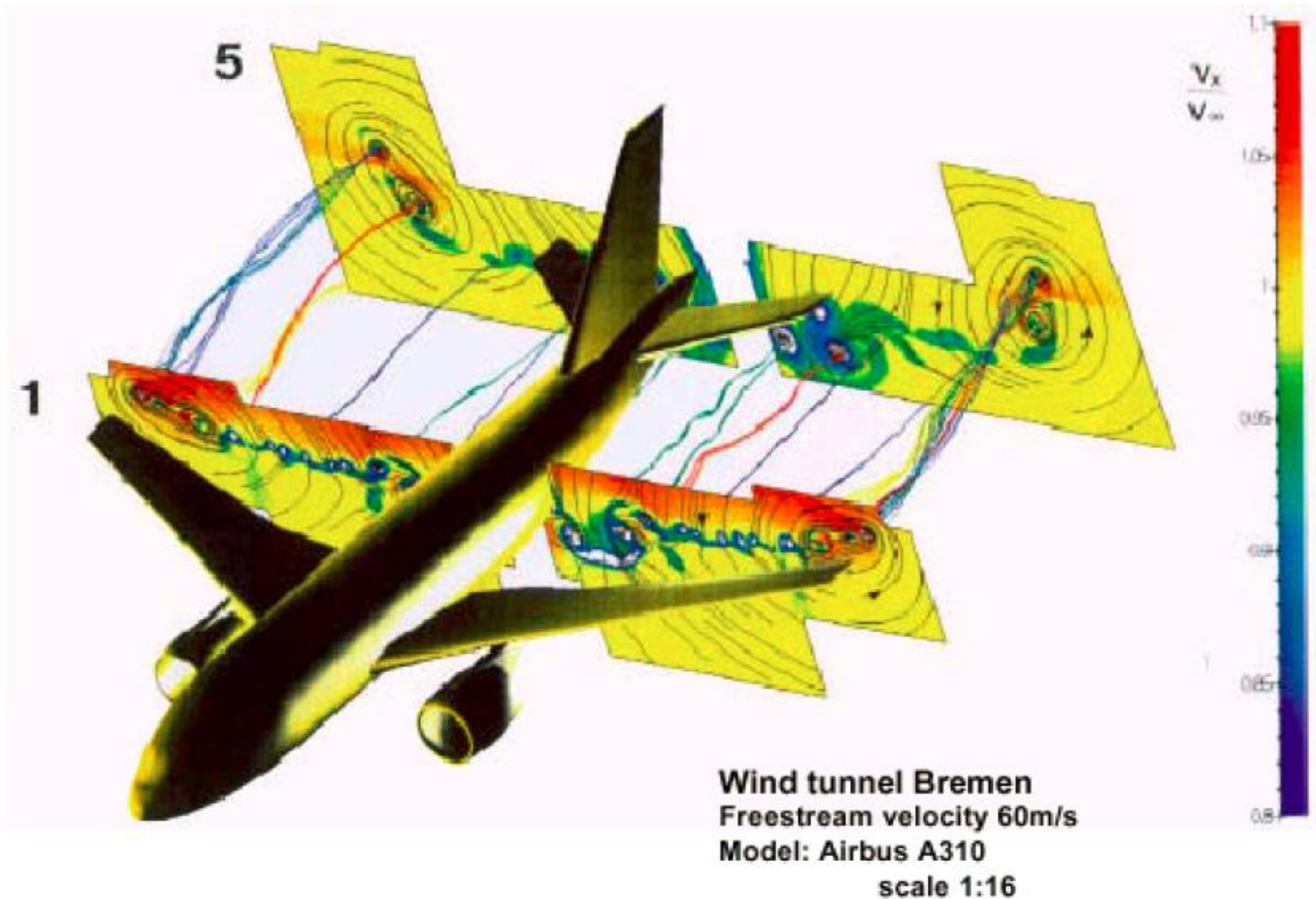
Příklad – Wind-Tip vír

- Simulace ve větrném tunelu



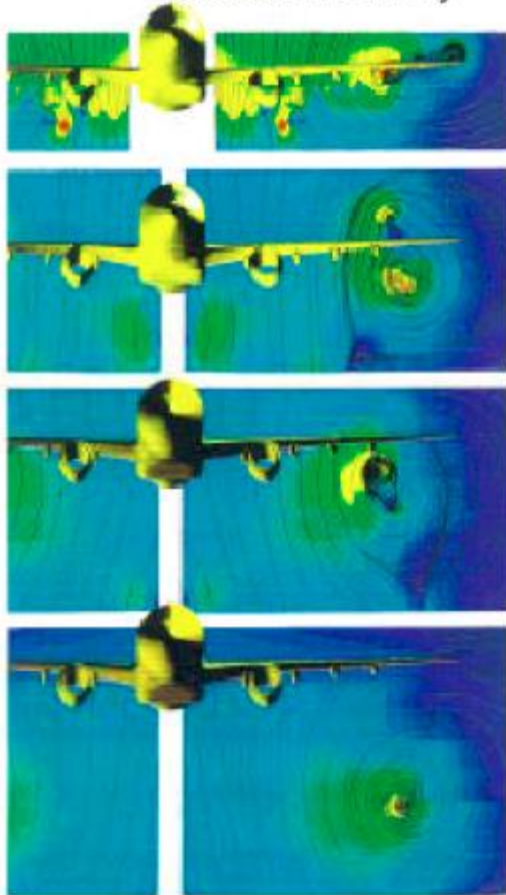
Příklad – Wind-Tip vír

- A následná vizualizace



DNW tunnel
Freestream velocity 60m/s

Crossflow velocity



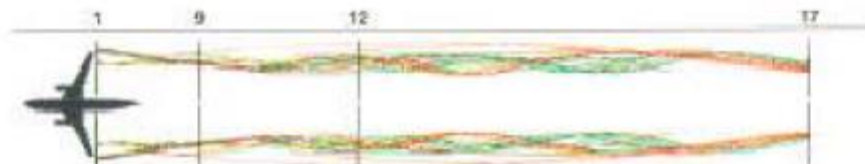
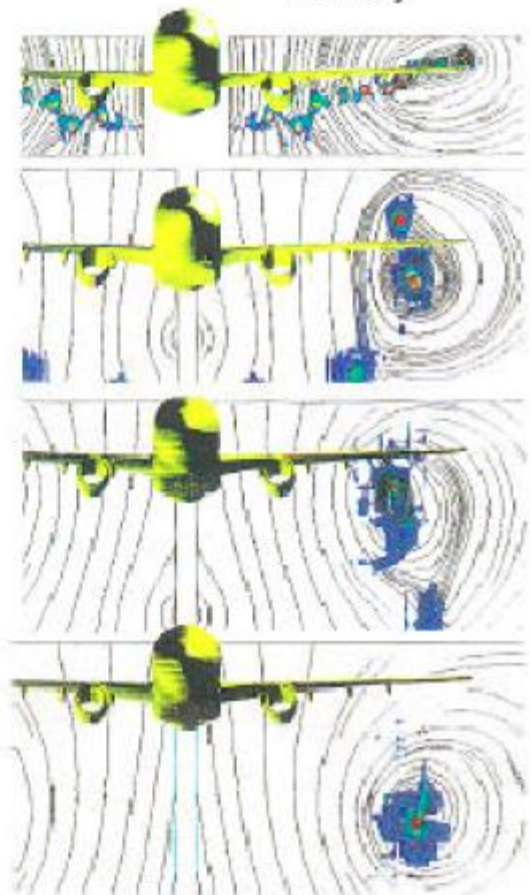
Surveying plane 1
0.03 wing spans behind wing tip

Surveying plane 9
1 wing spans behind wing

Surveying plane 12
2.5 wing spans behind wing

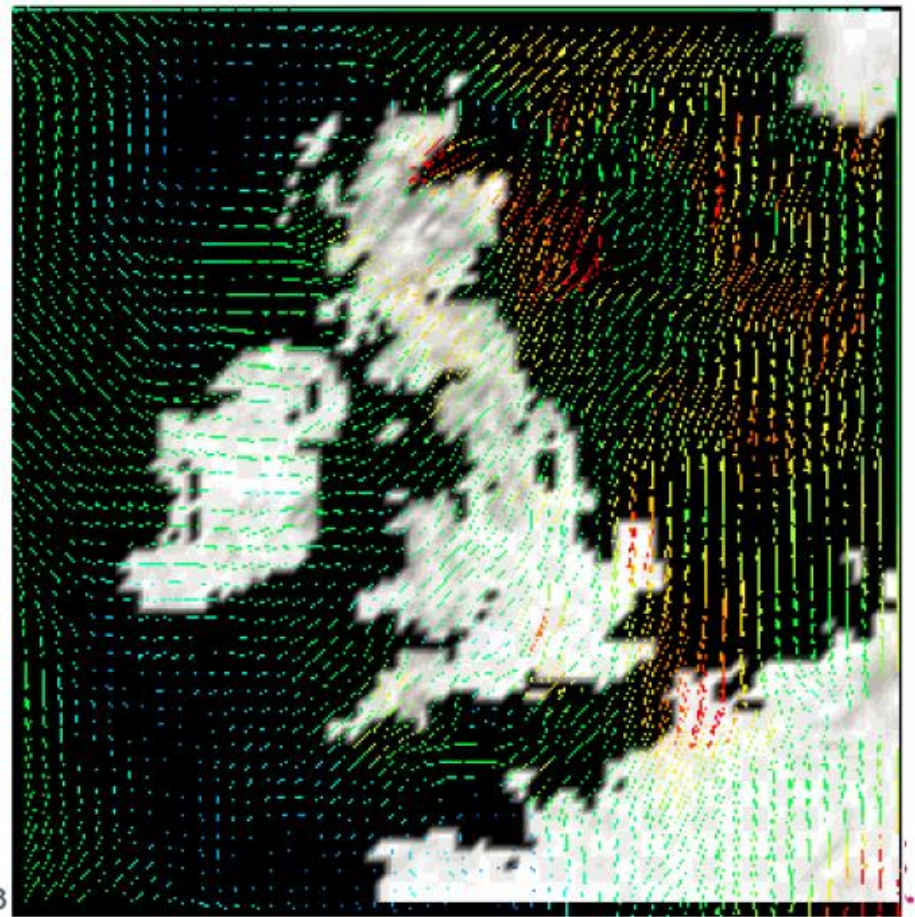
Surveying plane 17
6.8 wing spans behind wing

Vorticity



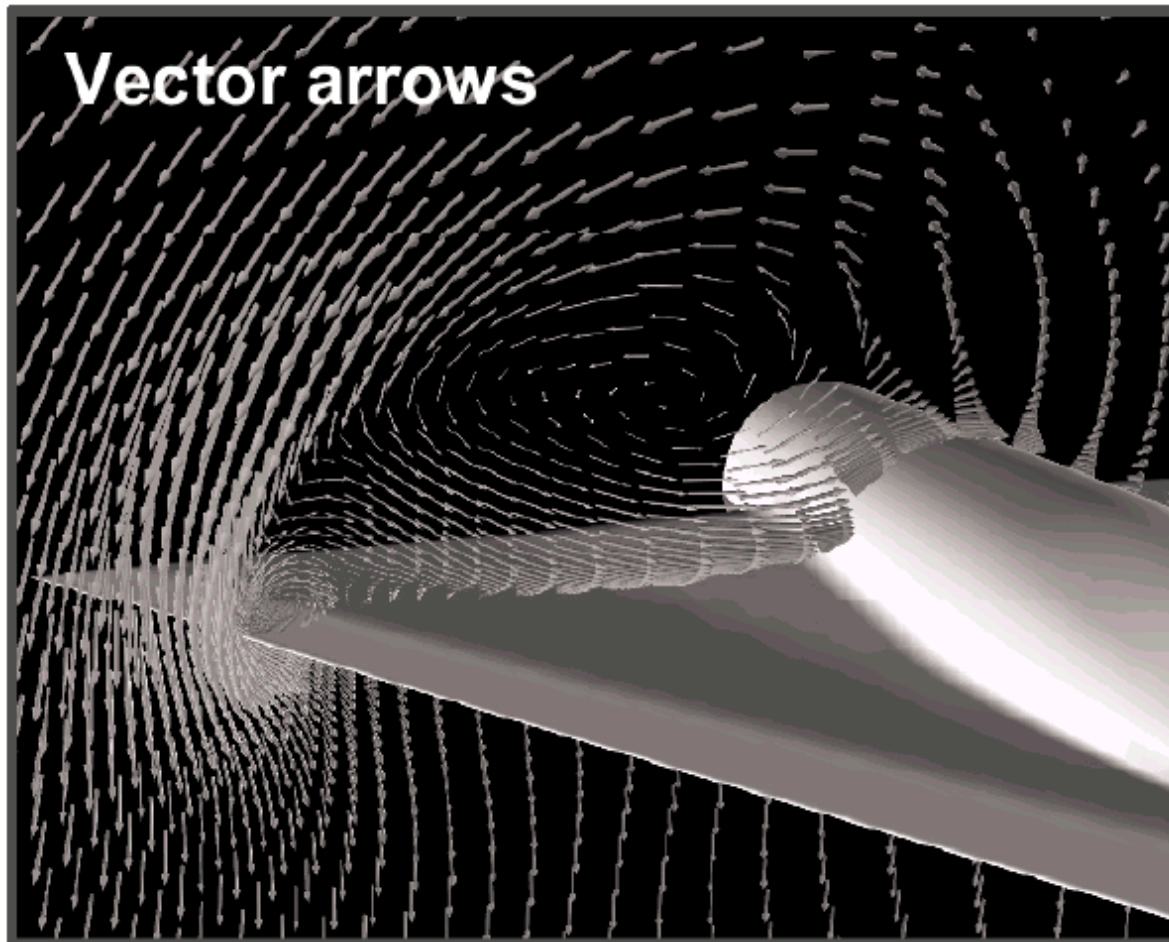
Vizualizace toku pomocí šipek

- 2D – scaling vs. obarvení šipek



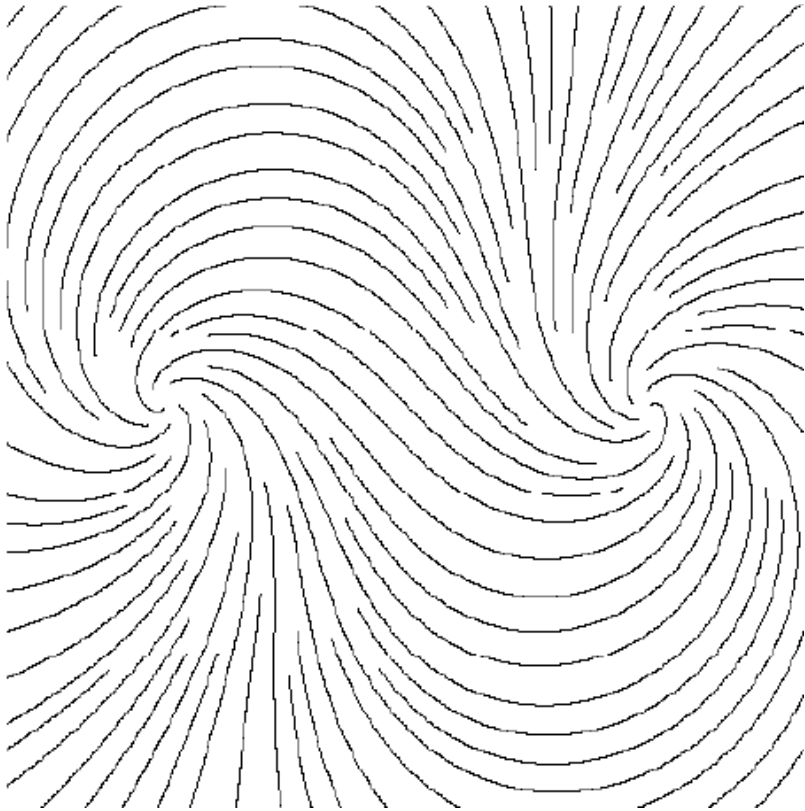
Vizualizace toku pomocí šipek

- 3D – šipky pouze v určitých „vrstvách“

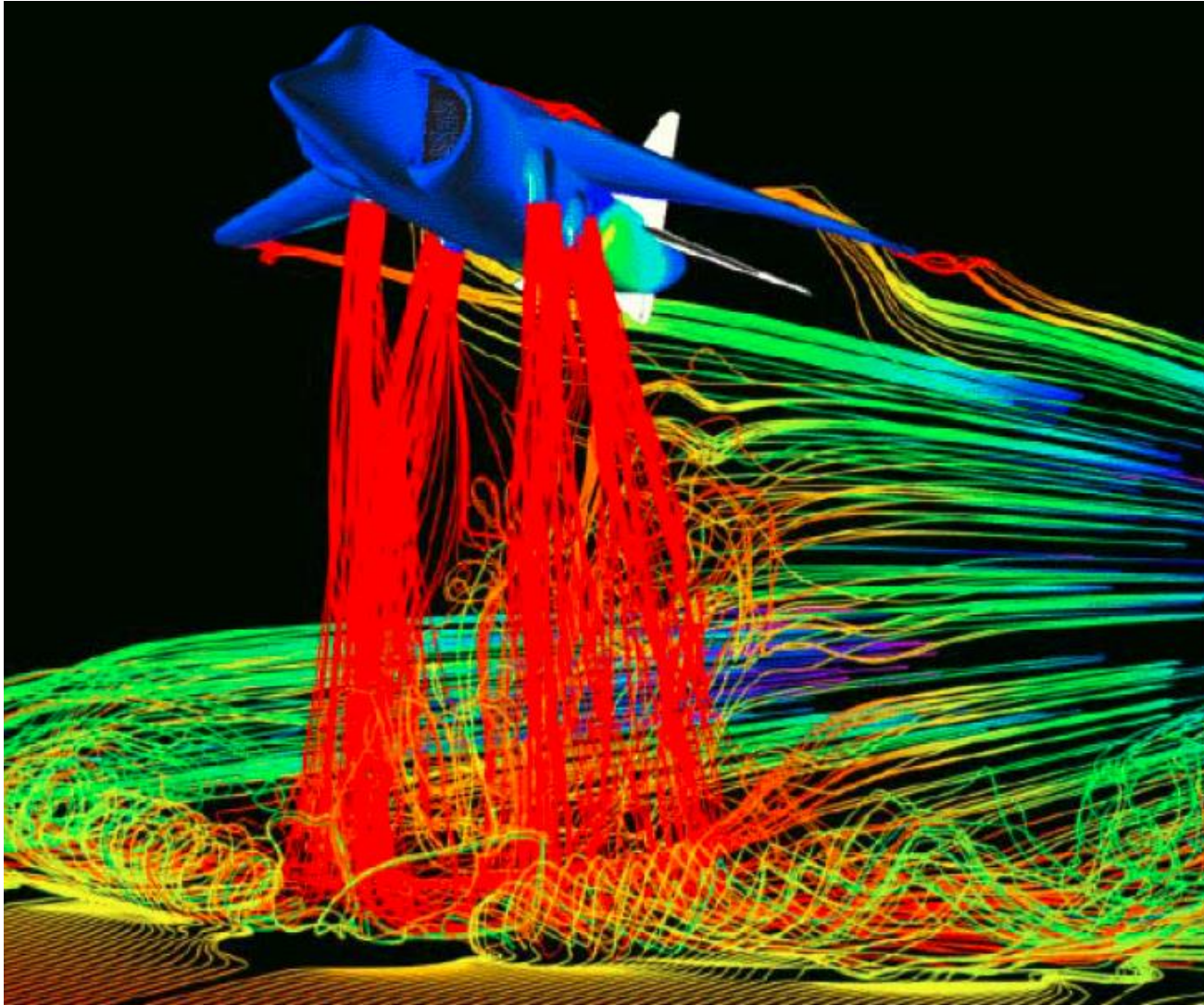


Vizualizace toku pomocí streamlines

- Streamlines = cesty jednotlivých částic v toku

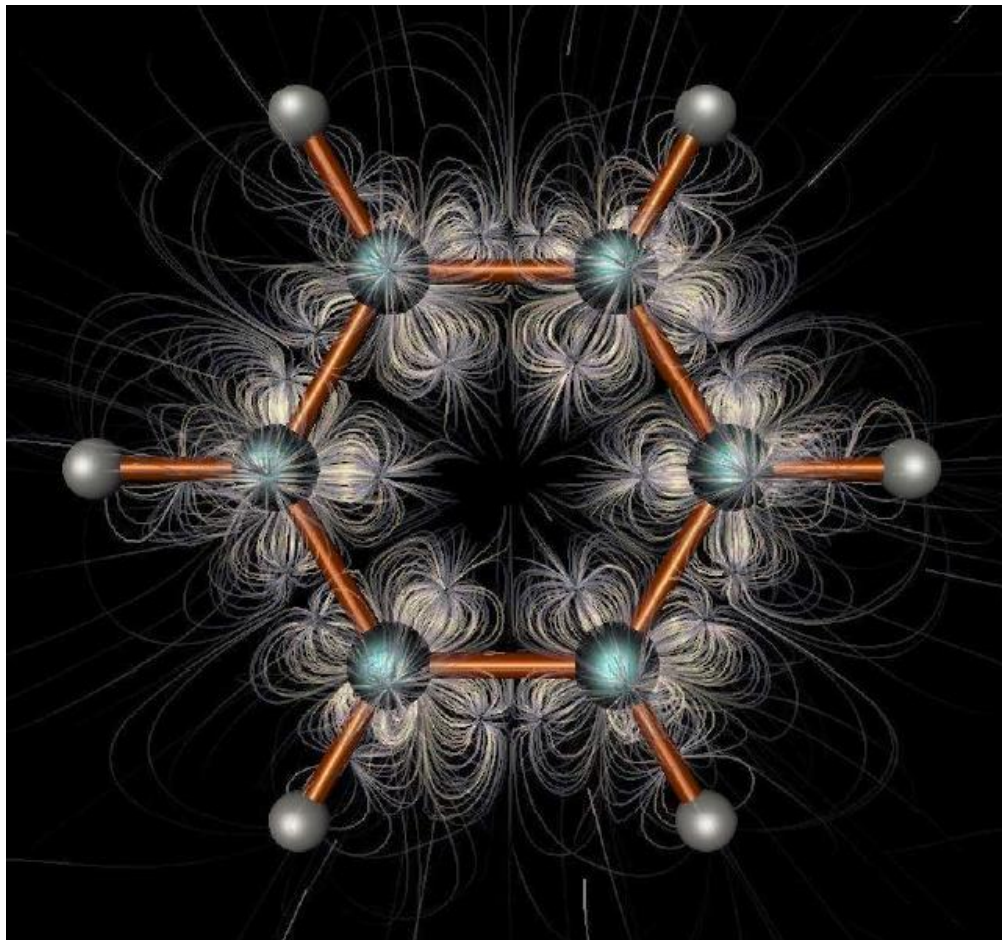


Streamlines ve 3D



„Osvětlené“ streamlines

- Pro lepší vnímání 3D prostoru



Algoritmus – umístění streamlines

- Hlavní myšlenka: streamlines by neměly být umístěny příliš blízko sebe
- Princip:
 - Parametry:
 - d_{sep} startovní vzdálenost
 - d_{test} minimální vzdálenost

Algoritmus – umístění streamlines

- Spočítej počáteční streamline, vlož do fronty
- Nastav počáteční streamline jako aktivní

WHILE není dokončeno DO

TRY získkej nový bod ve vzdálenosti d_{sep} od aktivní streamline

IF nalezeno THEN spočti novou streamline a vlož do fronty

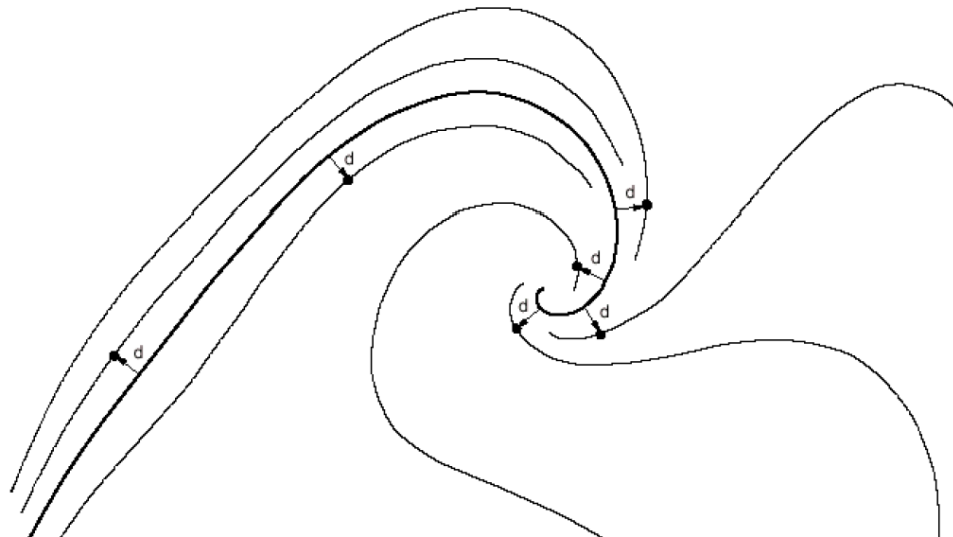
ELSE IF fronta je prázdná

THEN ukonči smyčku

ELSE další streamline ve frontě se stane aktivní

Ukončení tvorby streamlines

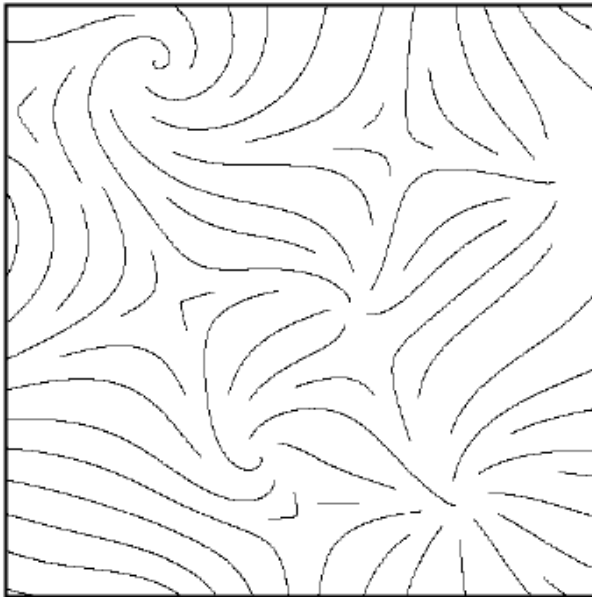
- Když je vzdálenost k sousední streamline $\leq d_{test}$
- Když streamline opustí definovanou doménu
- Když se streamline příliš přiblíží sama k sobě
- Po provedení předem definovaného počtu kroků



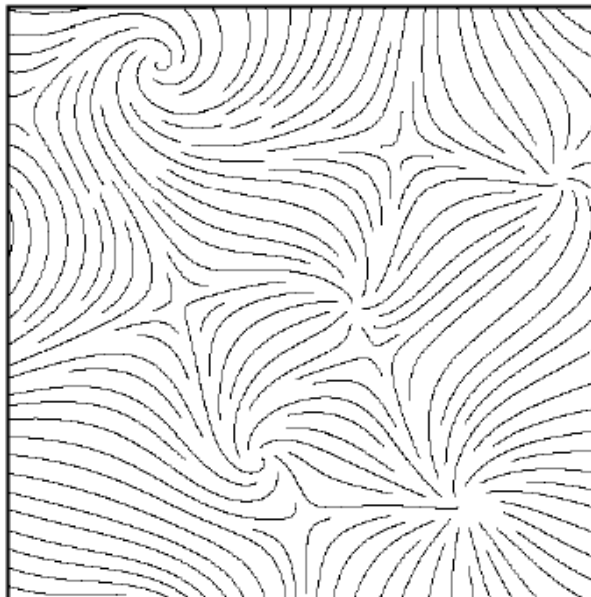
Streamlines – vliv d_{sep} na hustotu

- Relativně k šířce obrázku:

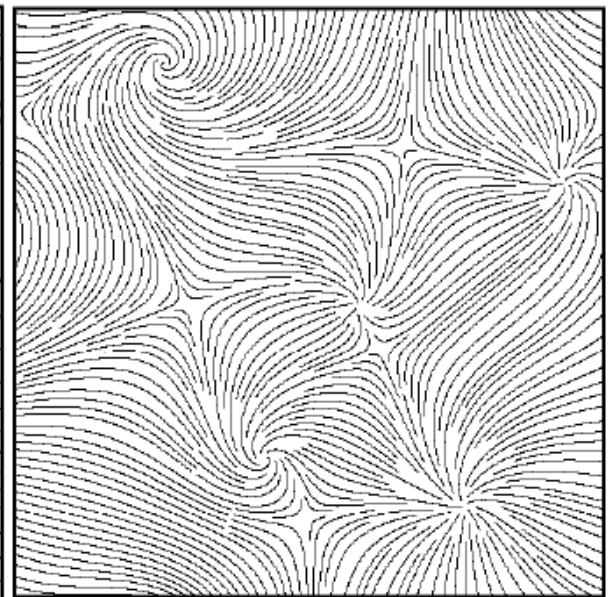
6%



3%



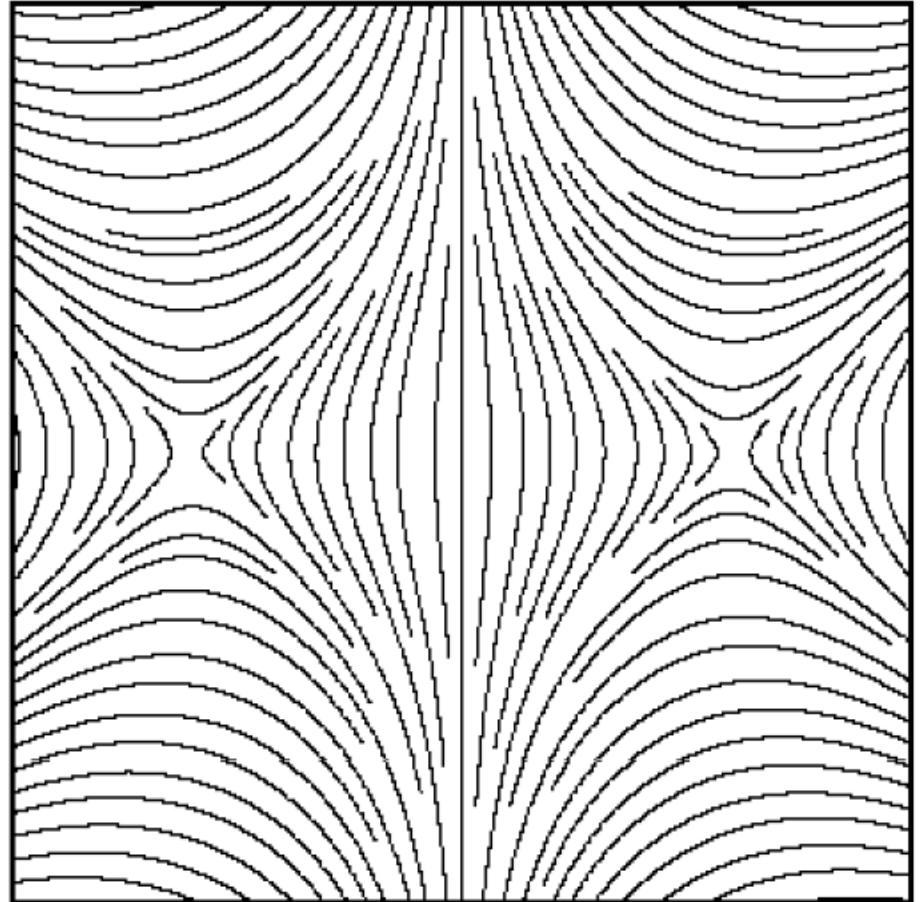
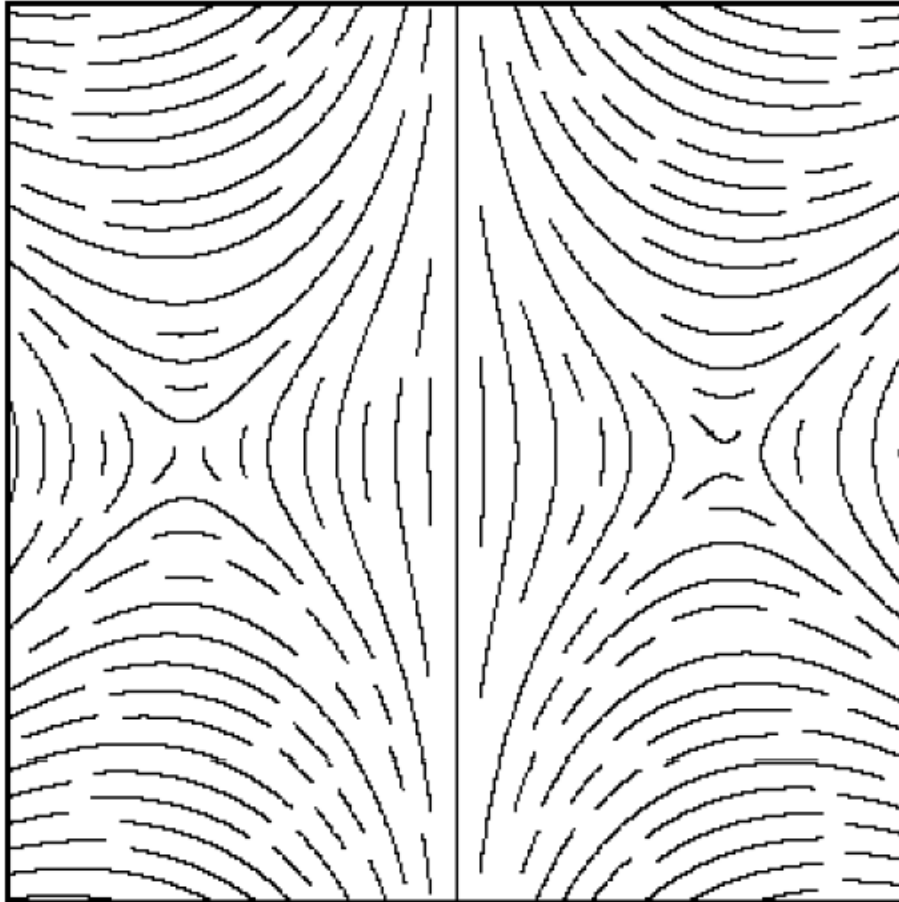
1.5%



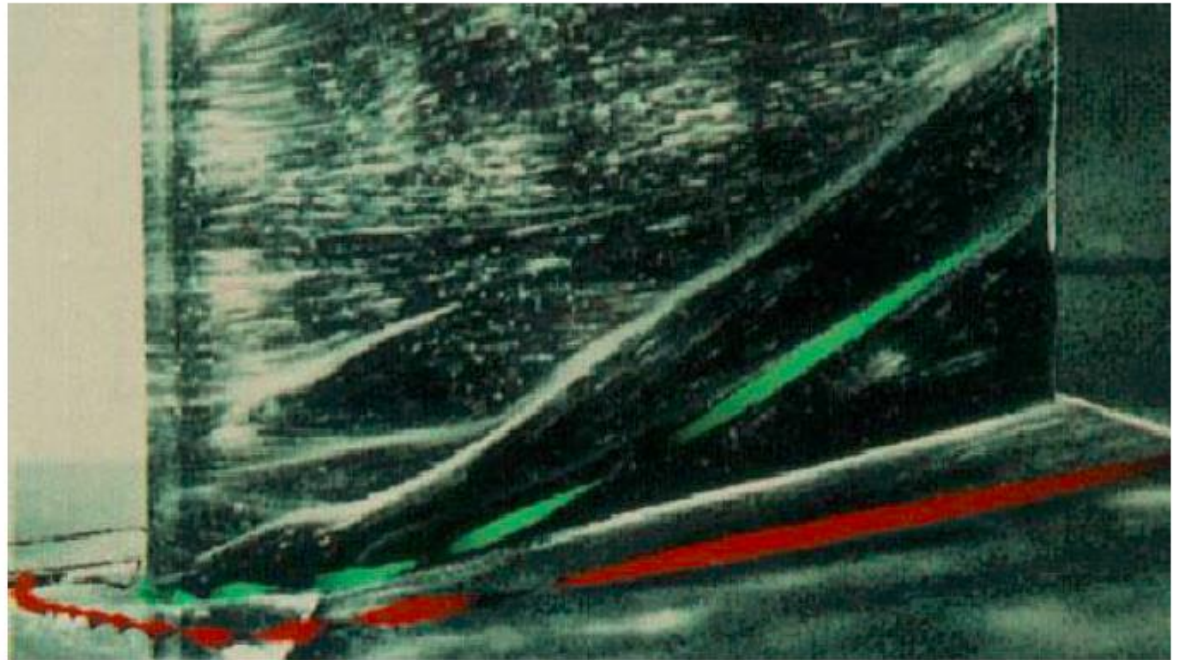
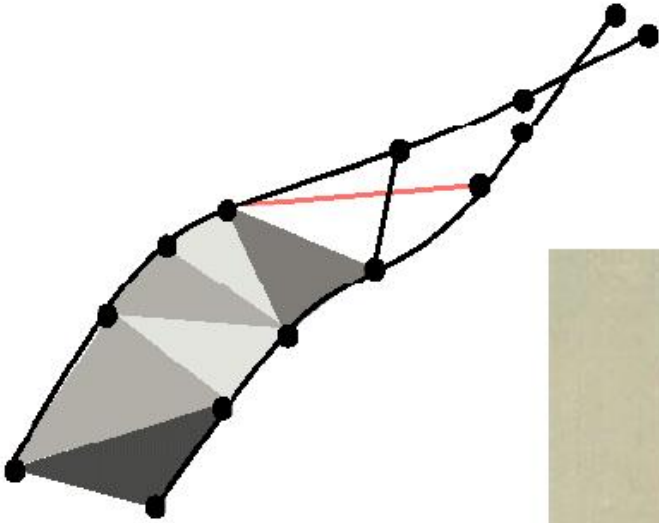
d_{sep} vs. d_{test}

$$d_{test} = 0.9 \cdot d_{sep}$$

$$d_{test} = 0.5 \cdot d_{sep}$$

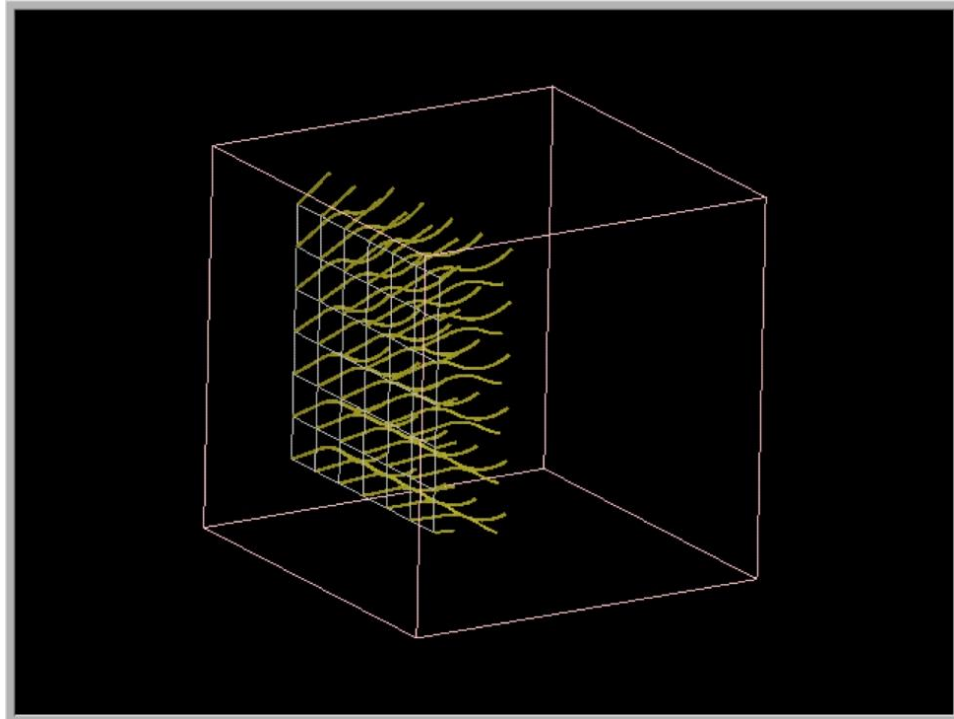


Streamribbons

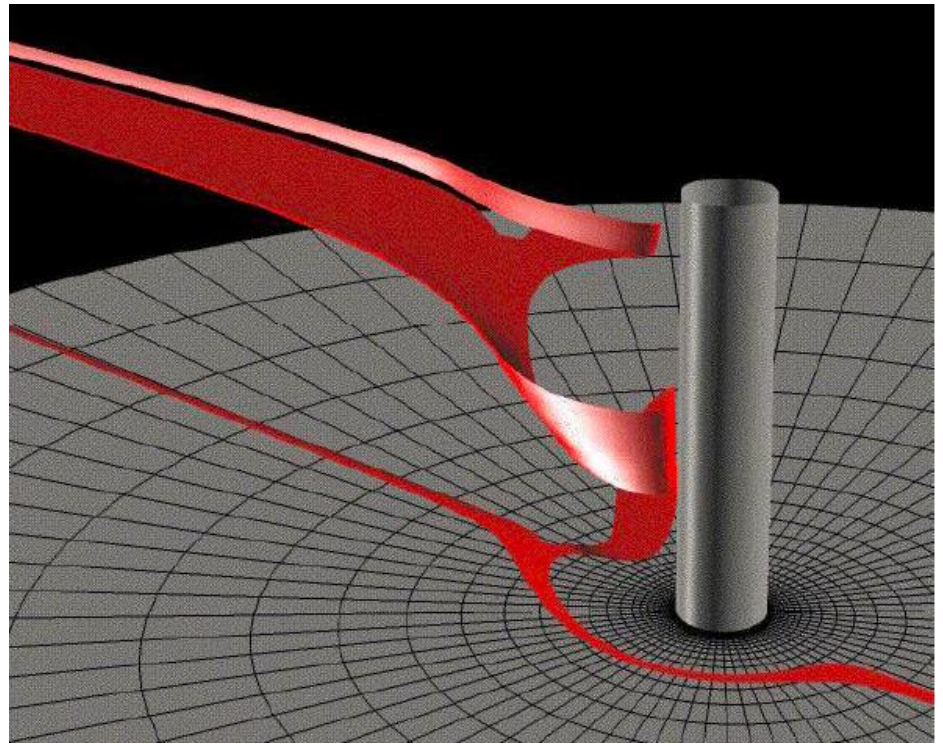
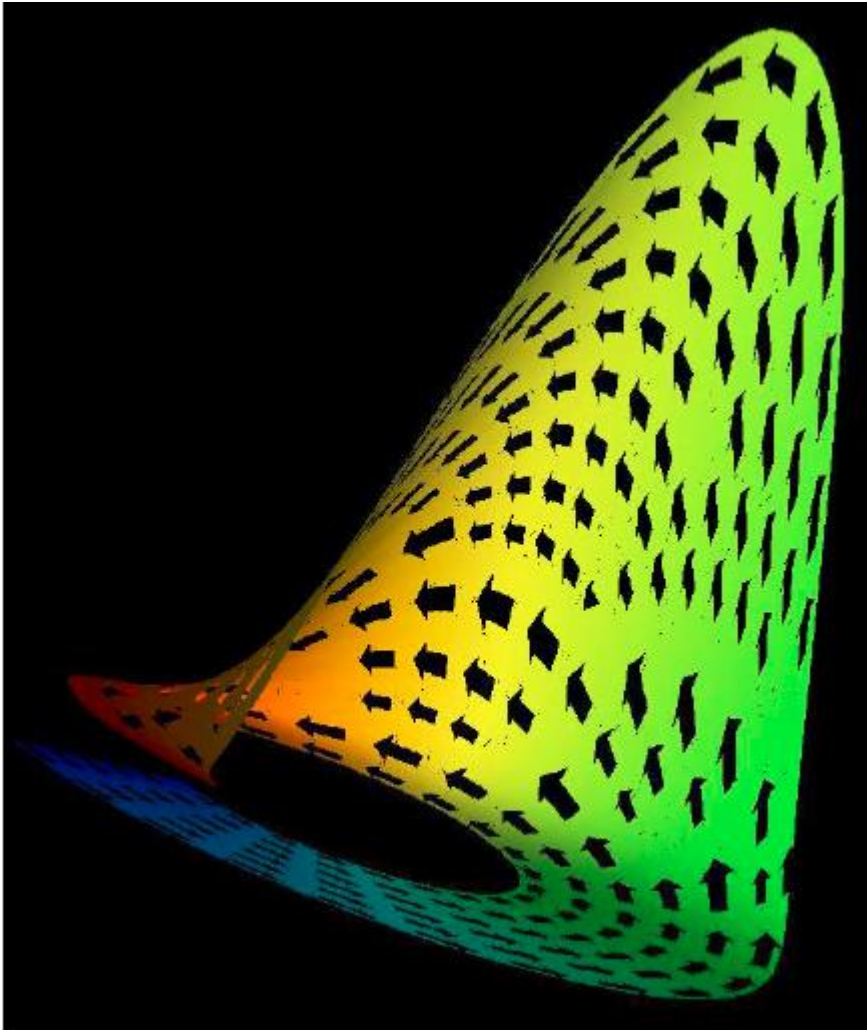


Streaklines

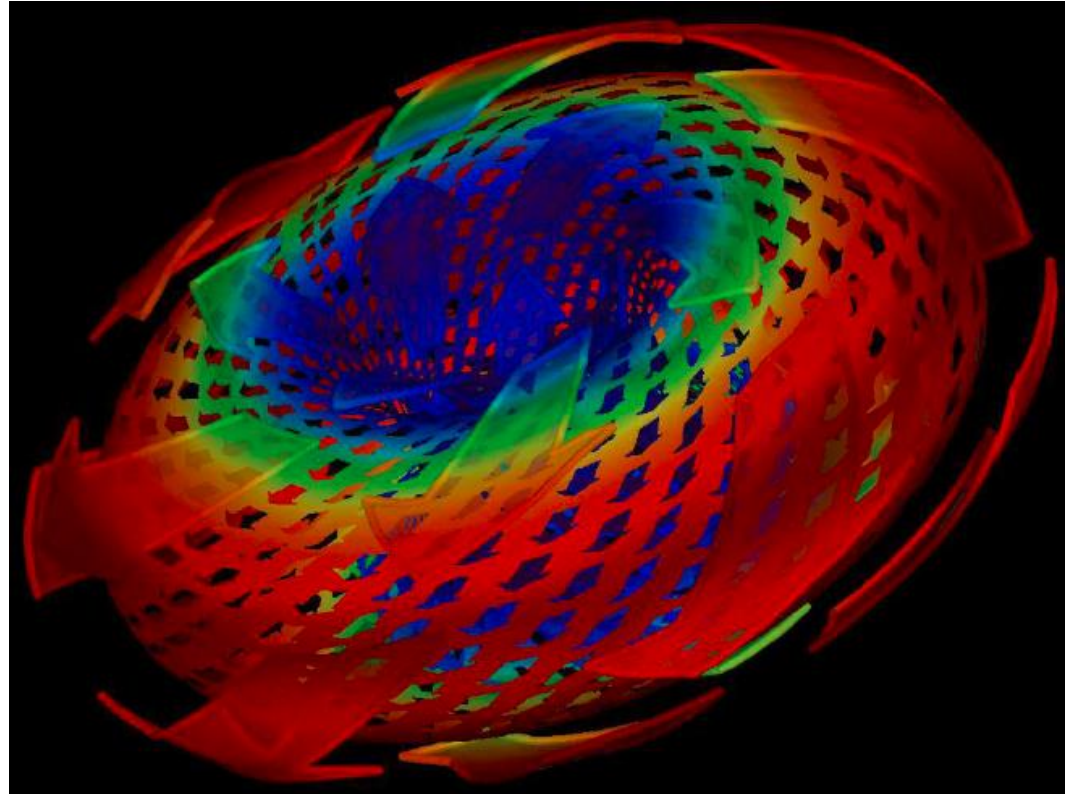
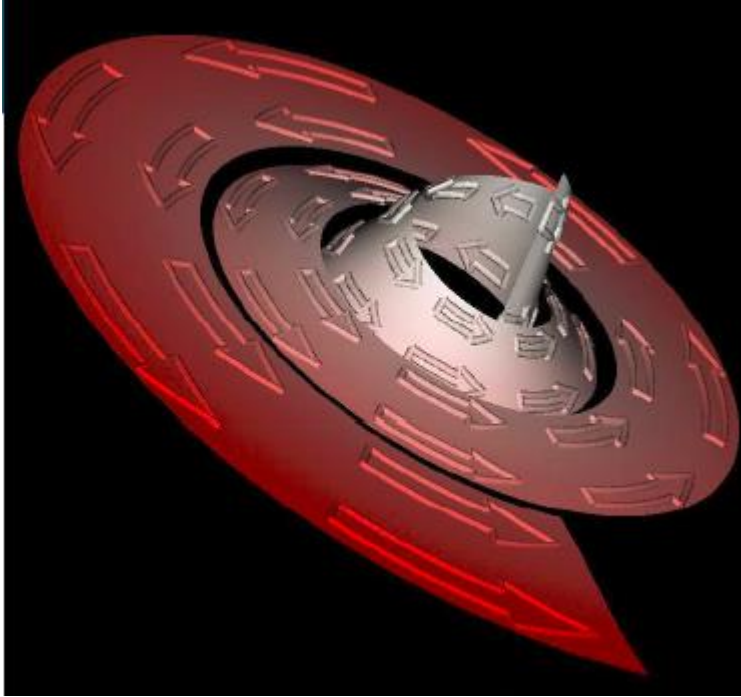
- Spojitý proud částic vyzařujících z diskrétní sady bodů a proudící skrz pole



Streamsurfaces

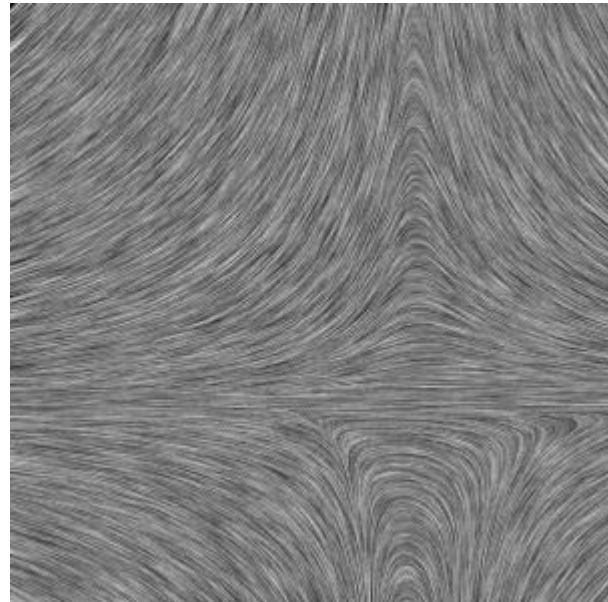


Stream Arrows



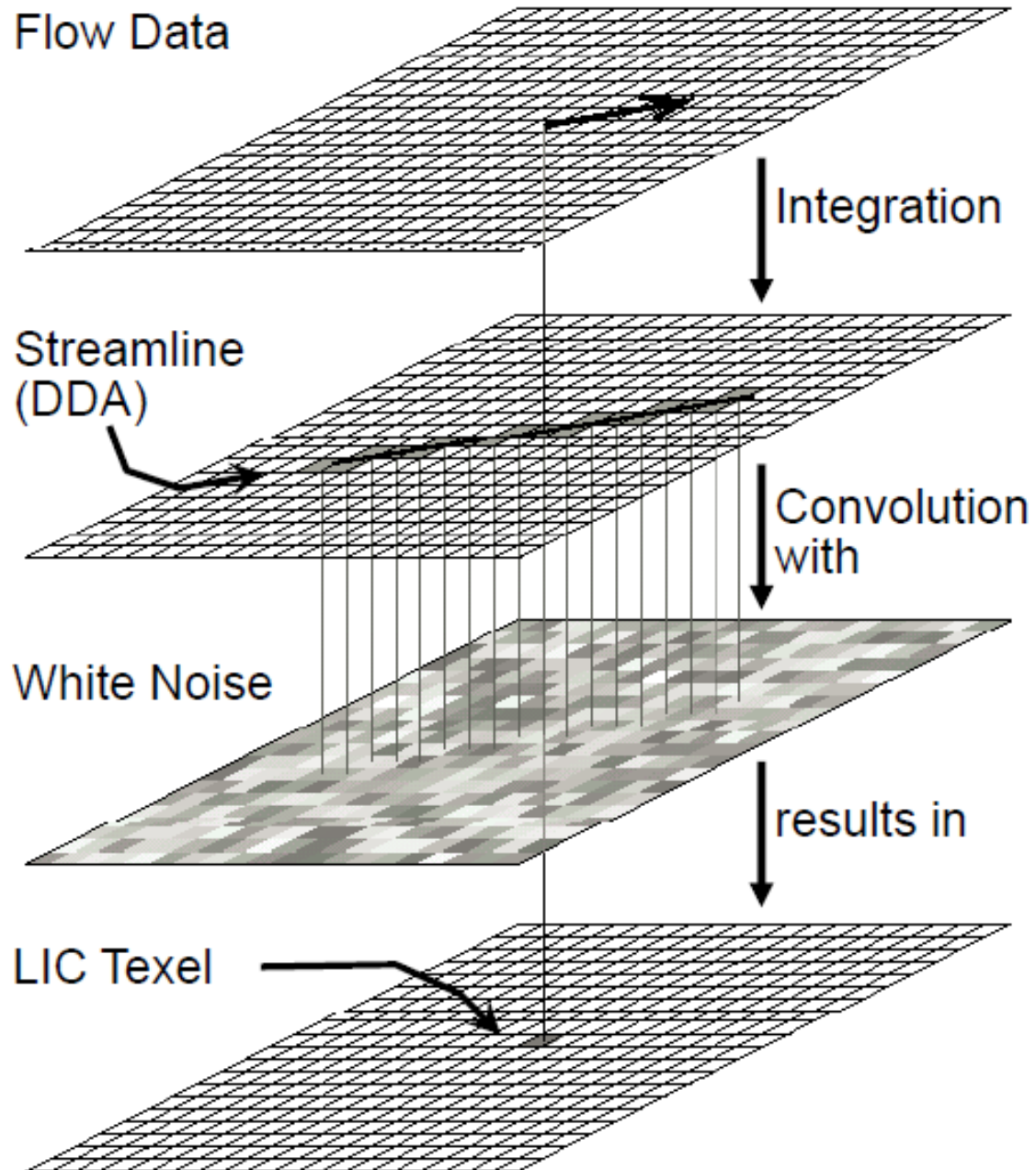
Line integral convolution (LIC)

- LIC byl navržen Cabralem a Leedomem v roce 1993
- Náhodné pole a vektorové pole stejné výšky a šířky pro generování hustého zobrazení průtokové informace

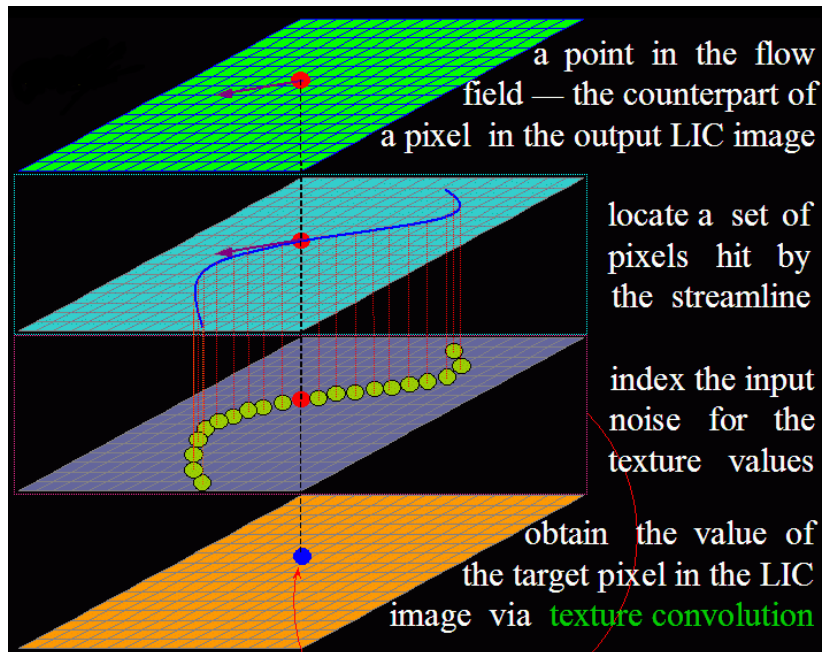
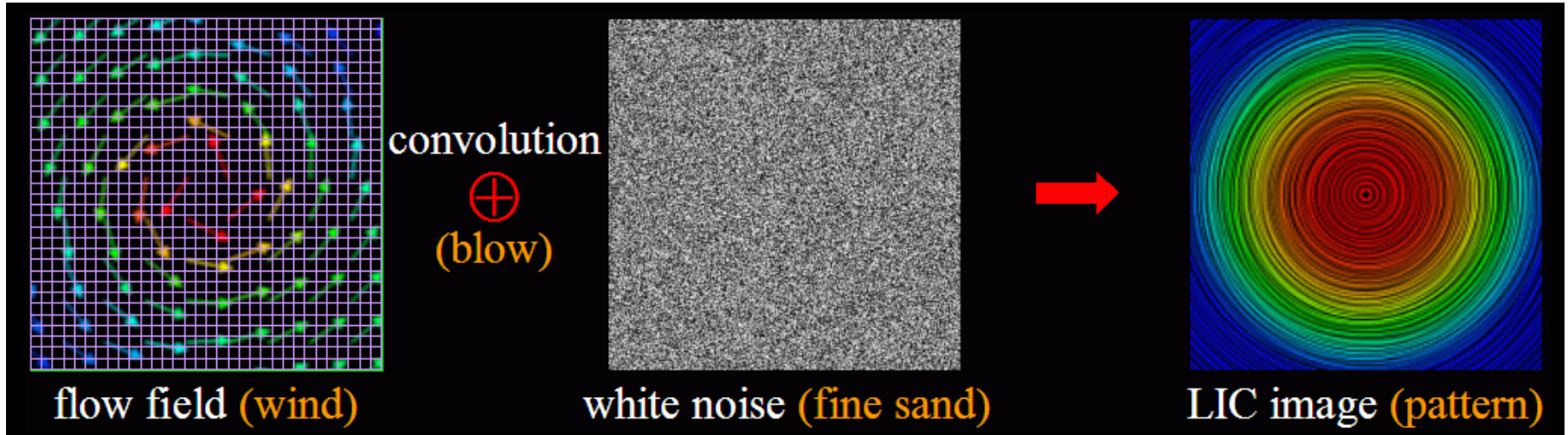


Line integral convolution (LIC)

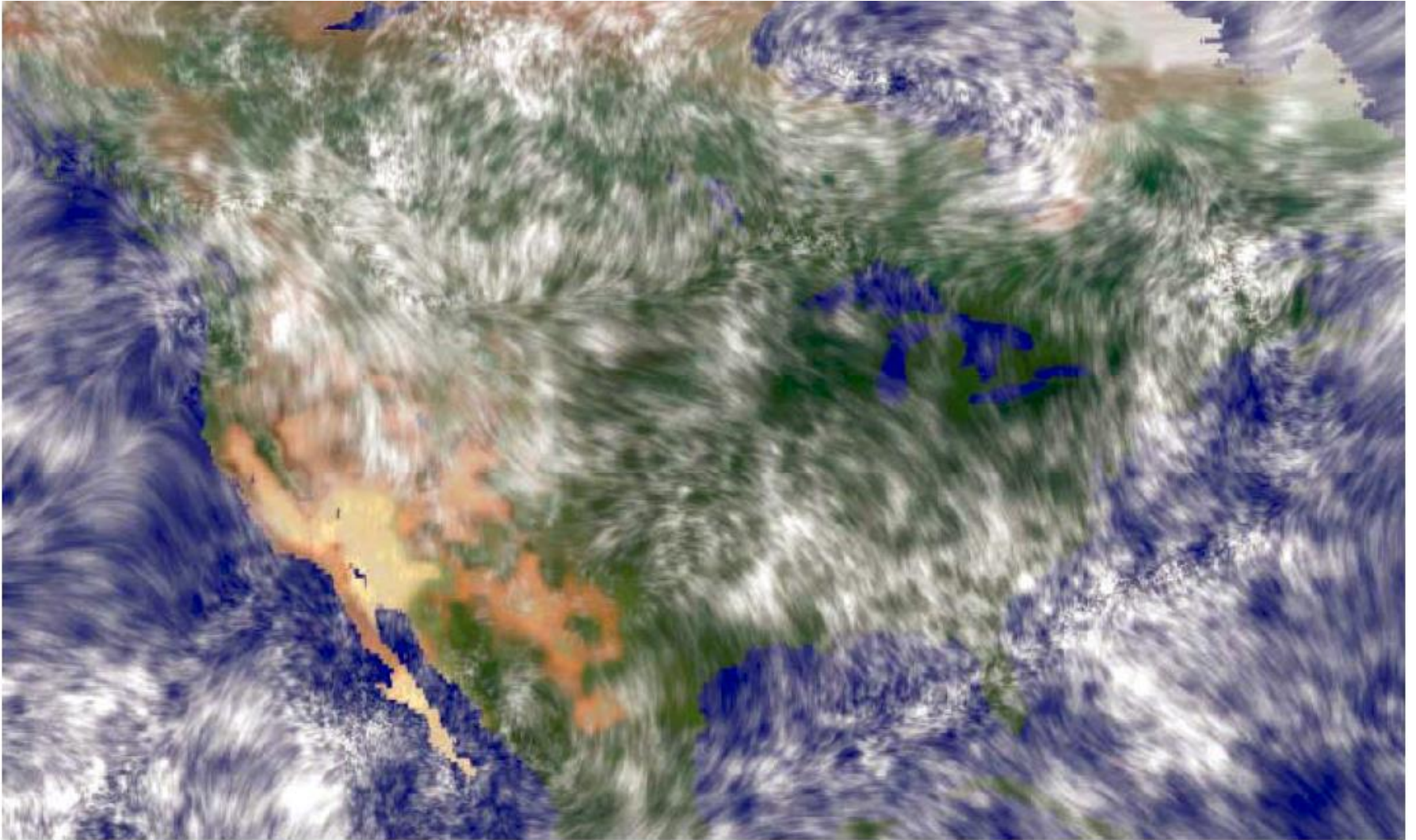
- Využívá textury pro zobrazení korelace mezi vizualizací a tokem
- Výpočet hodnoty textury
 - Náhled na streamline z daného bodu
 - Filtrace bílého šumu podél streamline



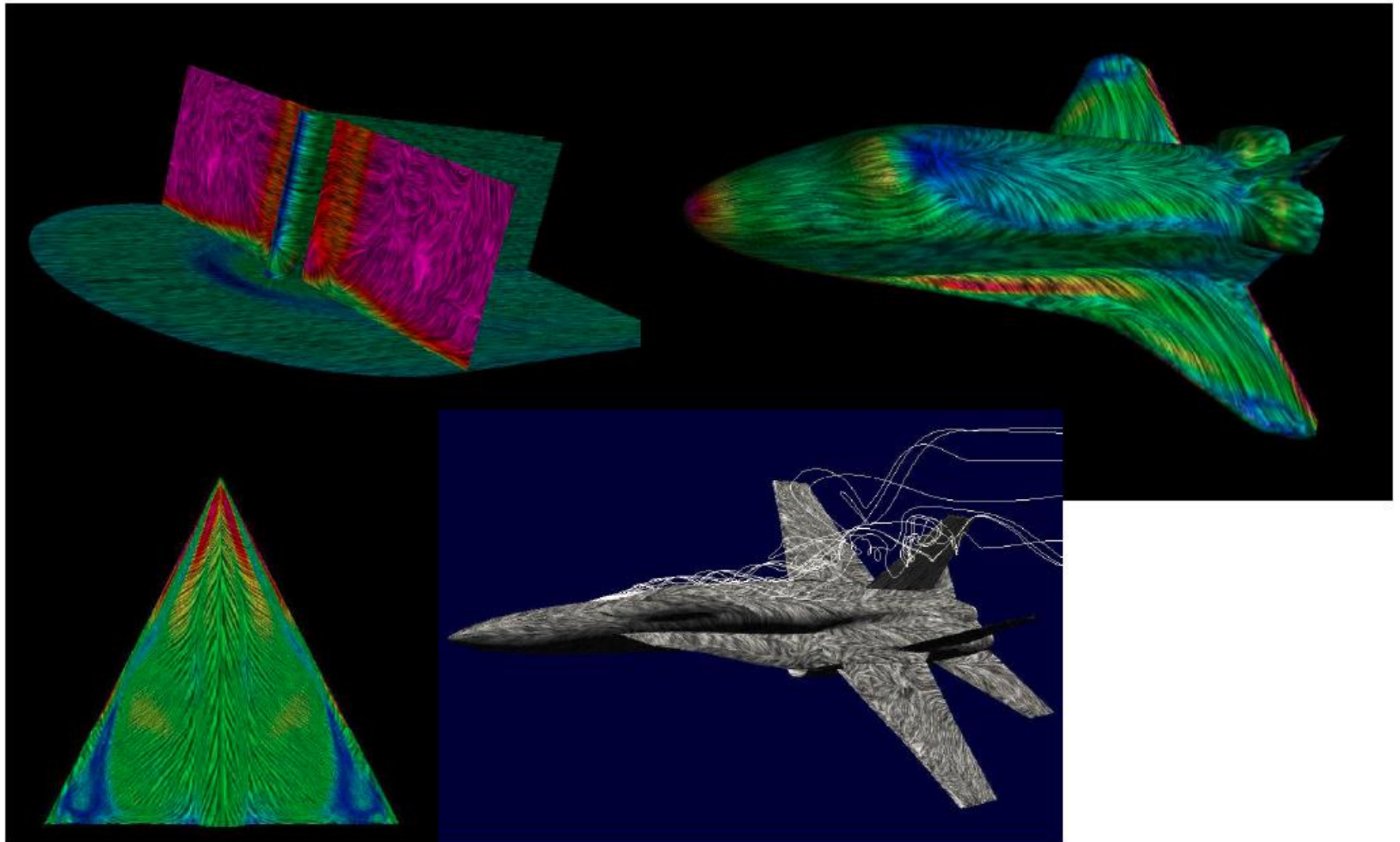
LIC



LIC příklady

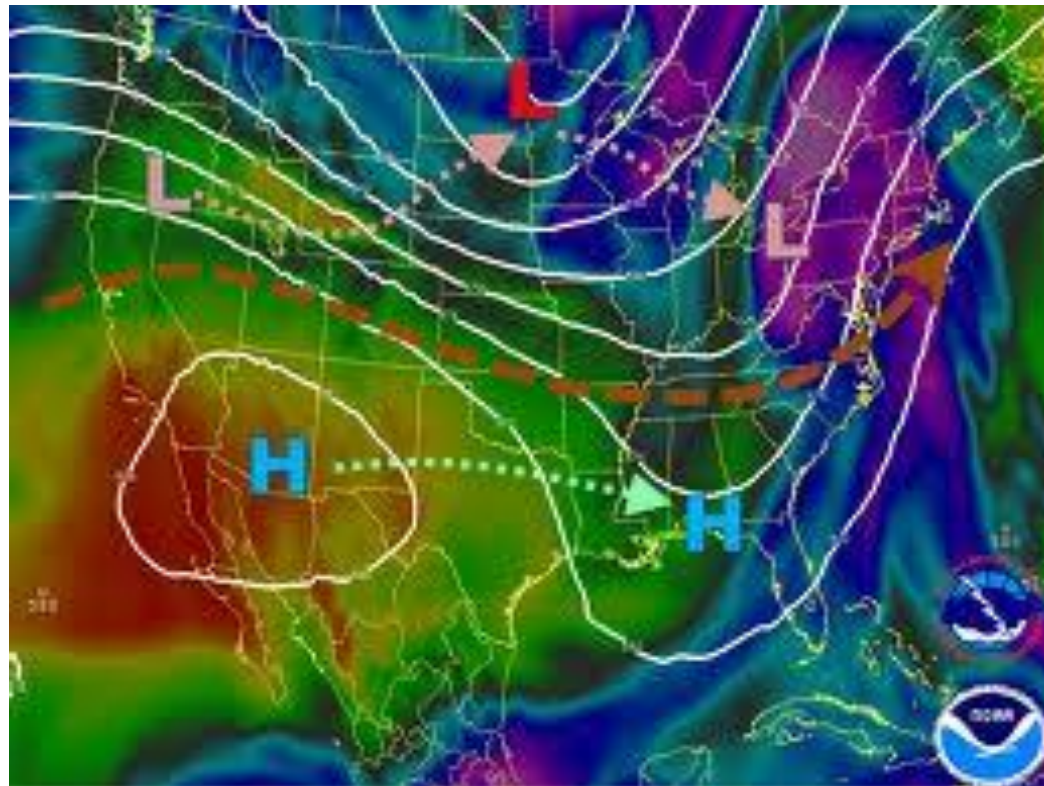


LIC – mapování na povrch



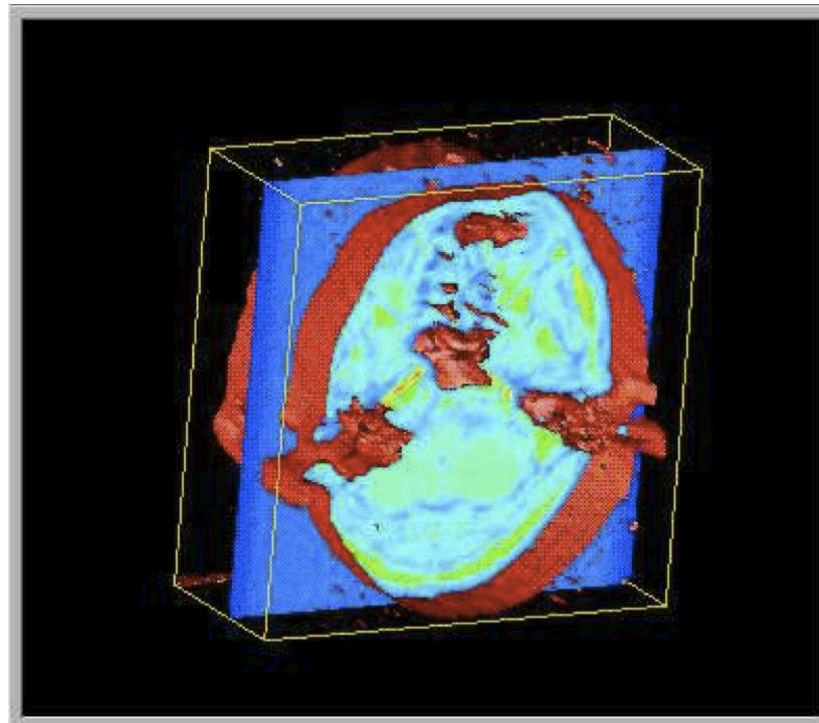
Kombinované techniky

- Kombinace technik může zvýraznit jejich silné stránky



Pláty kombinované s izopovrchem

- Izopovrch medicínských dat v kombinaci s ortogonálním plátováním
- [Video](#)

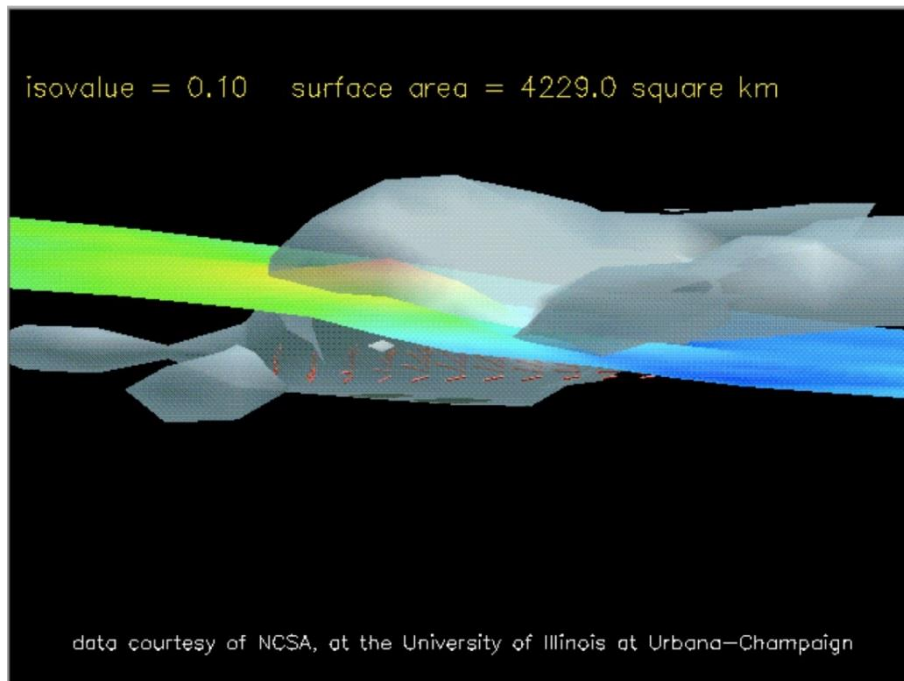


Pláty kombinované s izopovrchem

- Návrh tohoto typu vizualizace by měl vzít v úvahu:
 - Nepodporovat rychlé změny hodnot izopovrchu
 - Pozice a orientace plátu ovladatelná uživatelem
 - Pozice a orientace kamery ovladatelná uživatelem
 - Pečlivé přiřazení barvy
 - Skrytí vizualizačních komponent, případně průhlednost

Kombinace izopovrchu a piktoqramů

- Izopovrchy pro zobrazení detailů 3D povrchu, piktoqramy pro zobrazení velikosti nebo směru změny v datové množině



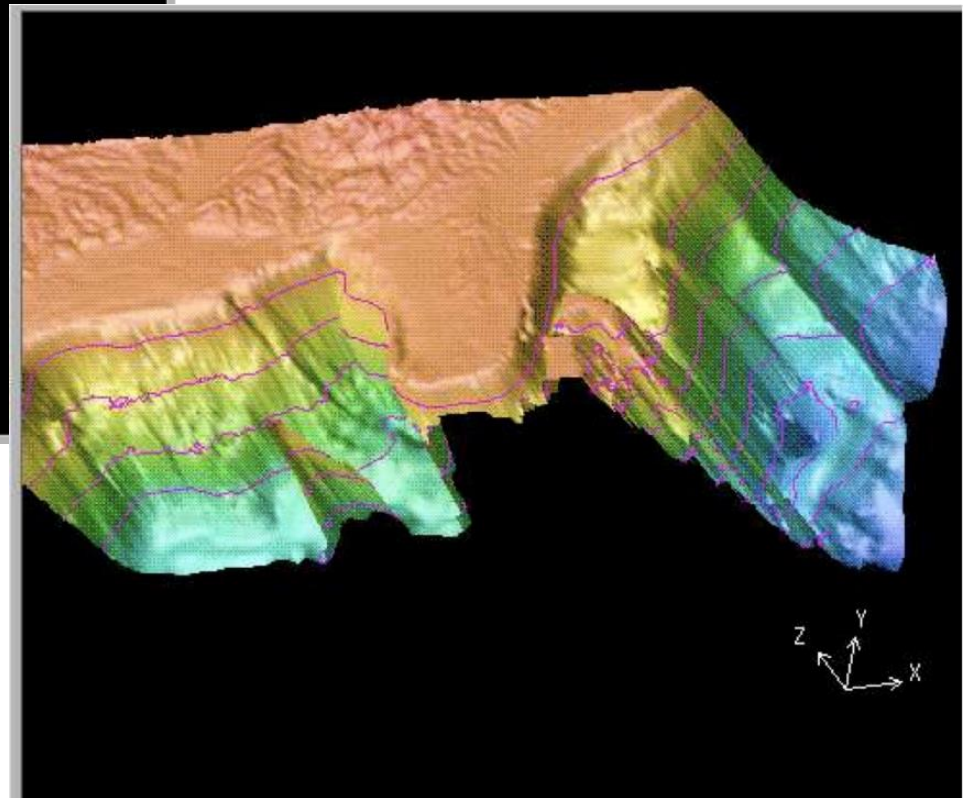
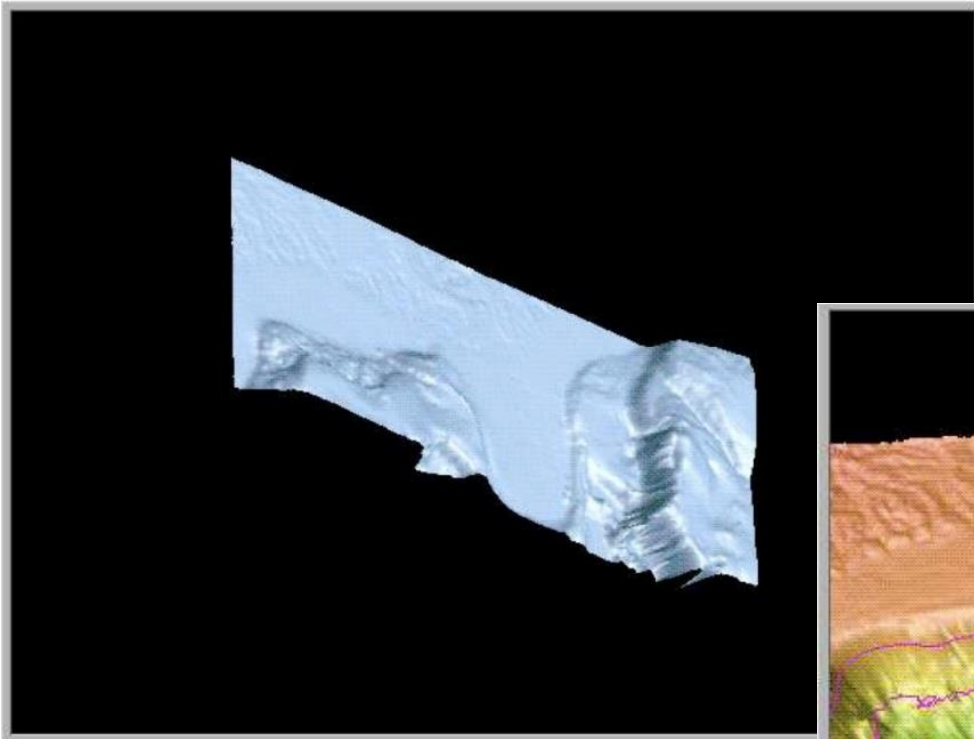
<http://www.opendx.org>



Kombinace izopovrchu a piktogramů

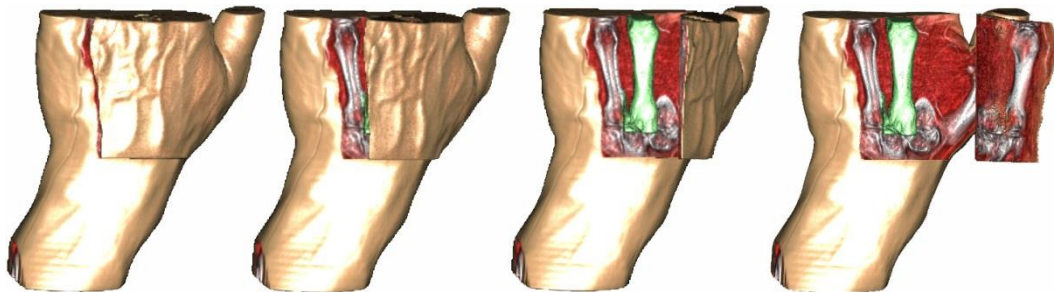
- Při návrhu vhodné zachovávat obdobná pravidla
 - Interaktivní kontrola parametrů
 - Mění se hustota glyfů
 - Mění se velikost glyfů
 - Různé barvy glyfů
 - Výpočet základní pozice glyfů

Reliéf + kontura + barva

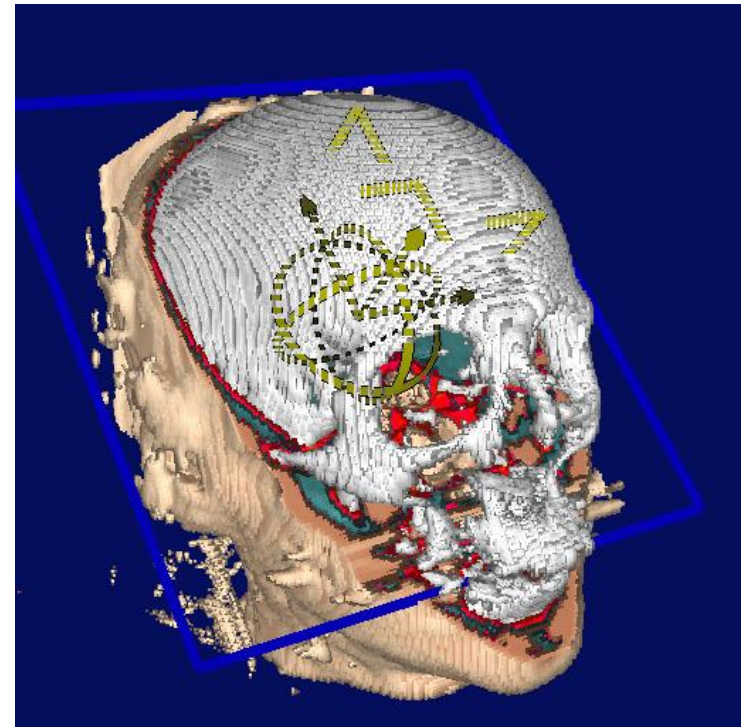


Shrnutí

- Různé techniky pro data o různých dimenzích
- Nutné porozumět kladům a záporům jednotlivých technik
- Vhodná je jejich kombinace



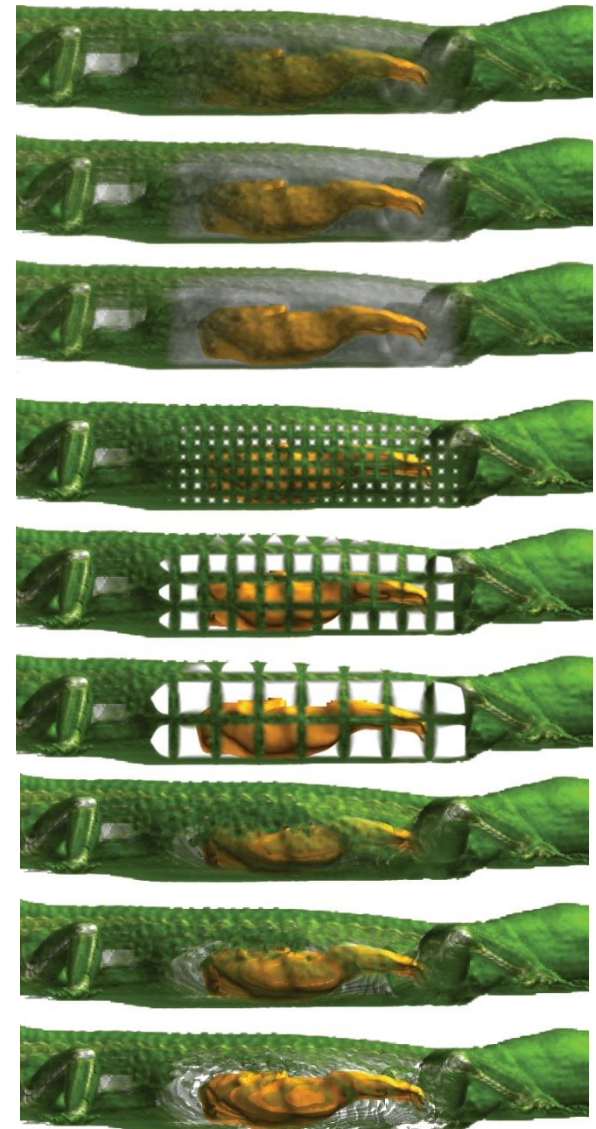
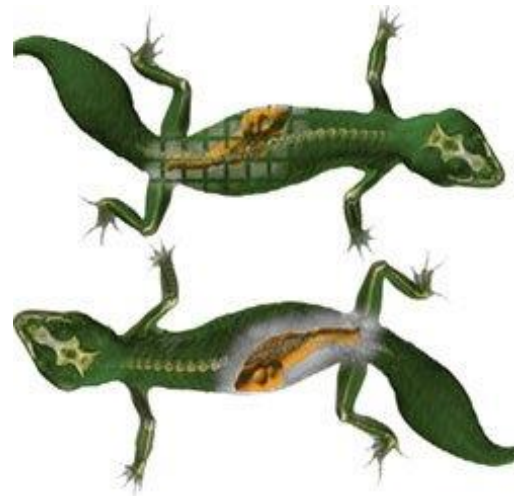
www.ii.uib.no



profs.etsmtl.ca

Příklady

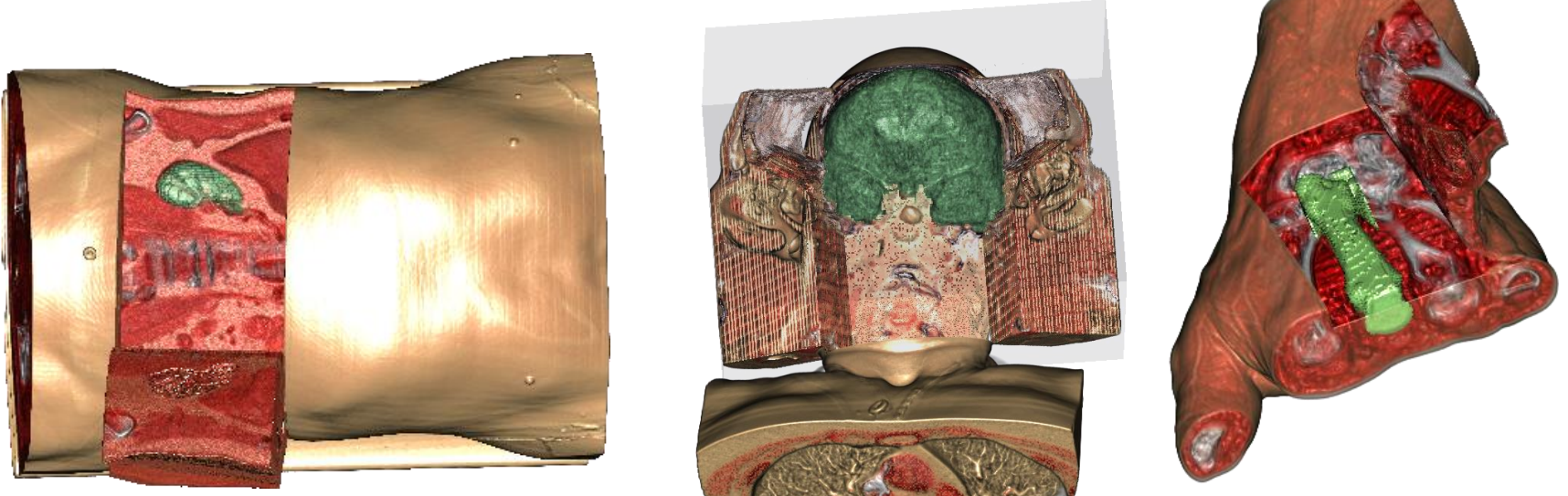
- Ivan Viola – Importance-Driven Volume Rendering



- <http://www.cg.tuwien.ac.at/research/publications/2004/Viola-2004-ImpX2/>

Příklady

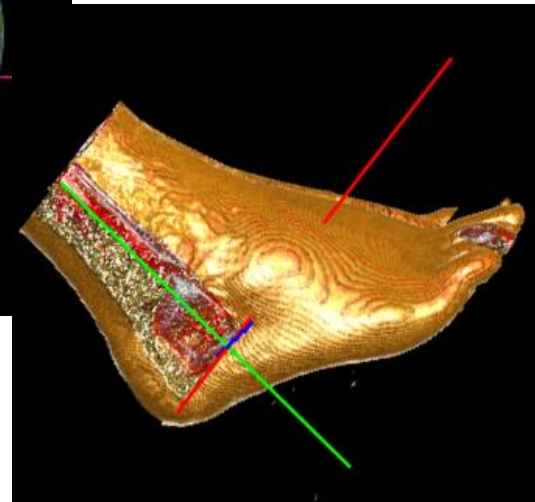
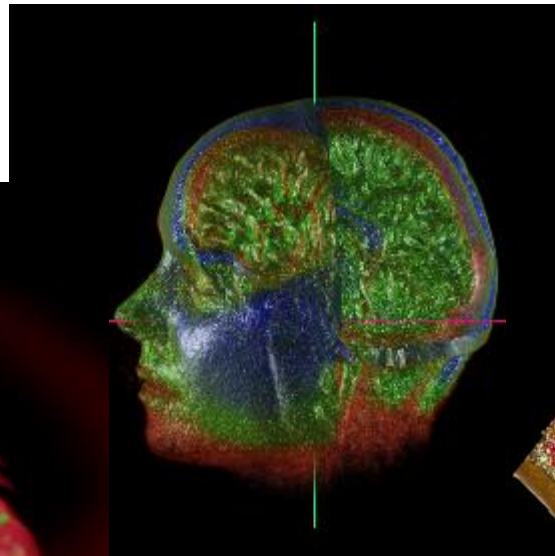
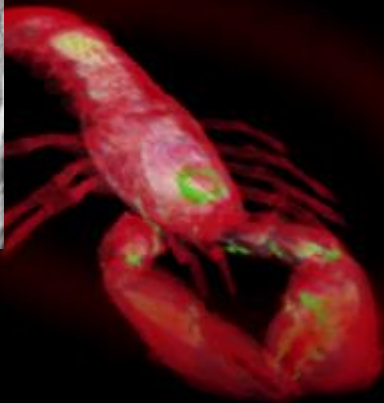
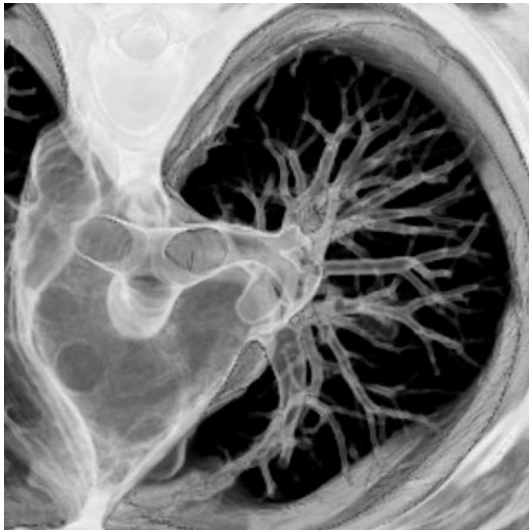
- Åsmund Birkeland - View-Dependent Peel-Away Visualization for Volumetric Data



- <http://www.iu.uib.no/vis/teaching/thesis/2008-birkeland/files/MasterThesisBirkeland2008.pdf>

Příklady

- Meißner et al., Volume Visualization and Volume Rendering Techniques, EUROGRAPHICS 2000



Videa

- 3D blood flow visualization

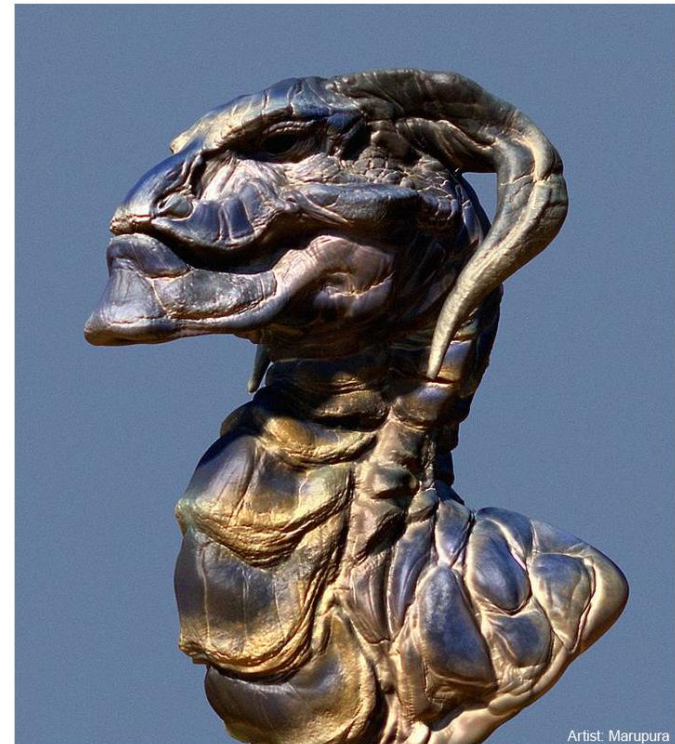
<http://www.youtube.com/watch?v=jkjbLk0nSFc>

- Flow visualization

<http://www.youtube.com/watch?v=DOUfyDHxkYQ>

Voxelové modelování

- Modelovací program 3D-Coat
 - 30-denní trial verze:
 - <http://3d-coat.com/download/>
 - Modelování na voxelech, následný převod na polygony a texturování



<http://3d-coat.com/voxel-sculpting/>