

Zobrazování a osvětlování. Vizualizace

Petr Felkel

Grafická skupina katedry počítačů ČVUT FEL
místnost E413 na Karlově náměstí
felkel@fel.cvut.cz

I. Zobrazování a osvětlování

I. Zobrazování a osvětlování

- Základní zobrazovací algoritmy
- Metody výpočtu osvětlení
- Globální osvětlovací metody

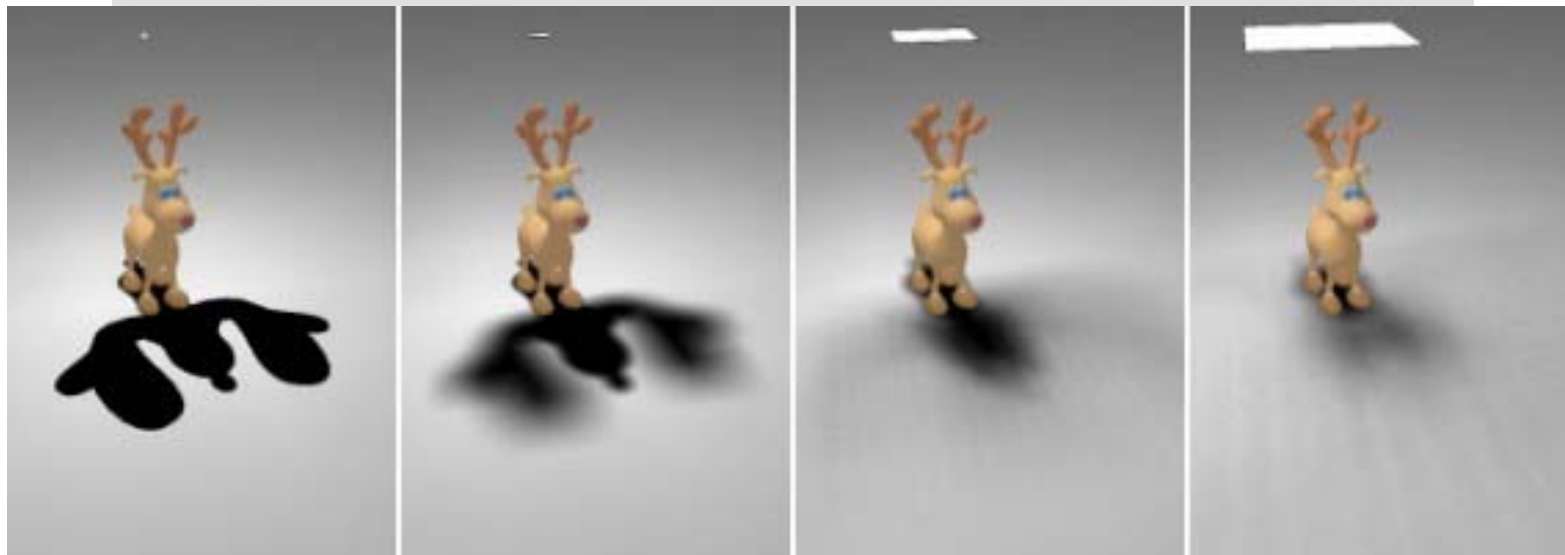
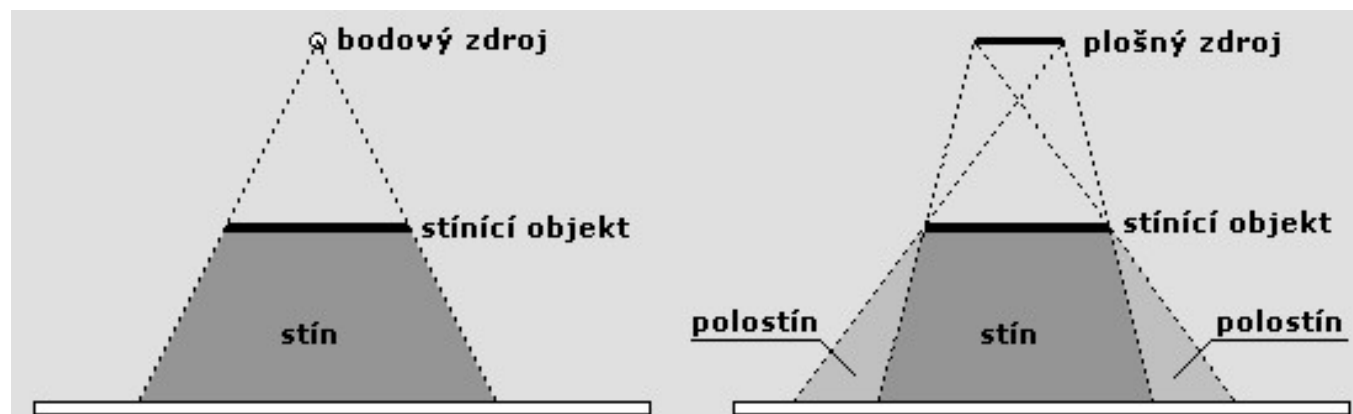
- Závisí na reprezentaci dat
- Pro hraniční reprezentaci:
- Object order – jen viditelnost + osvětlení model
 - Zpracovávají popořadě objekty
 - Malířův algoritmus (Back to Front)
 - Z-buffer (libovolné pořadí)
 - Radiozita (vzájemné osvětlení plošek)
- Image order
 - Popořadě pixely, zkoumají, co do nich padne (vysílají se paprsky z oka skrz pixely do scény)
 - Sledování paprsku (*Ray tracing*)
 - Pro objemy F-B, B-F projekce – sumace, splatting (viz dále)

- Lokální metody výpočtu osvětlení (dosud)
 - Objekty se vzájemně neosvětlují
 - Jako by každý byl sám ve scéně se všemi světly
 - Jenom se řeší vzájemná poloha kvůli viditelnosti (Zakrývají se vzájemně, ale nezakrývají světla, ani se vzájemně neosvětlují)
 - Malířův algoritmus, Z-buffer rychlé
 - (simuluje se ambientním světlem,...)
- Globální osvětlovací metody
 - Odraz od tělesa může osvětlit stěnu odvrácenou od světla (Kaustiky (prasátka) ve vodě, mlha a mraky ve vzduchu,...)
 - Problém je simulace všech optických jevů v reálném čase

- Simulují skutečný svět (fotorealismus)
- Stíny či polostíny (měkké stíny)
- Vzájemný difúzní odraz
(Osvětlení odvrácených ploch, zabarvení ploch)
- Kaustiky

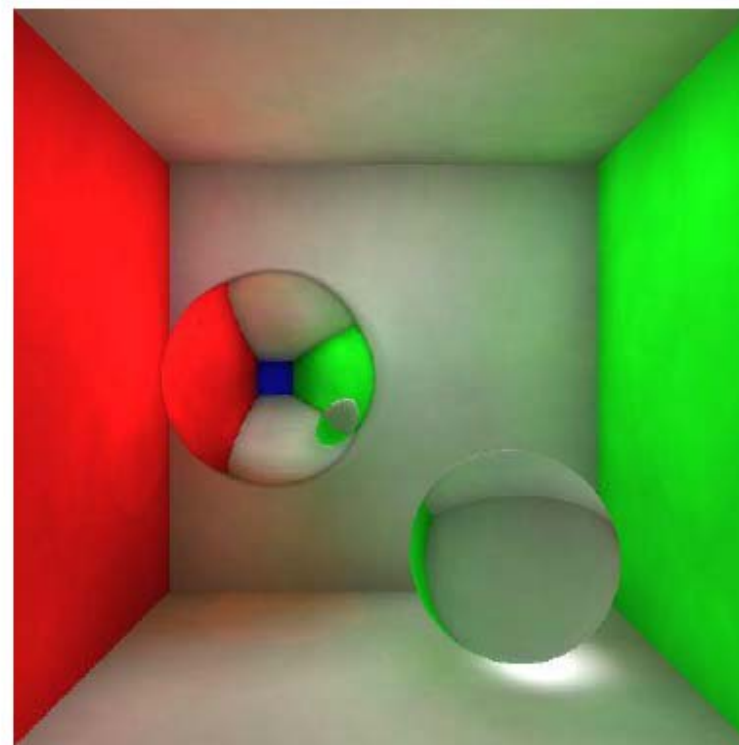
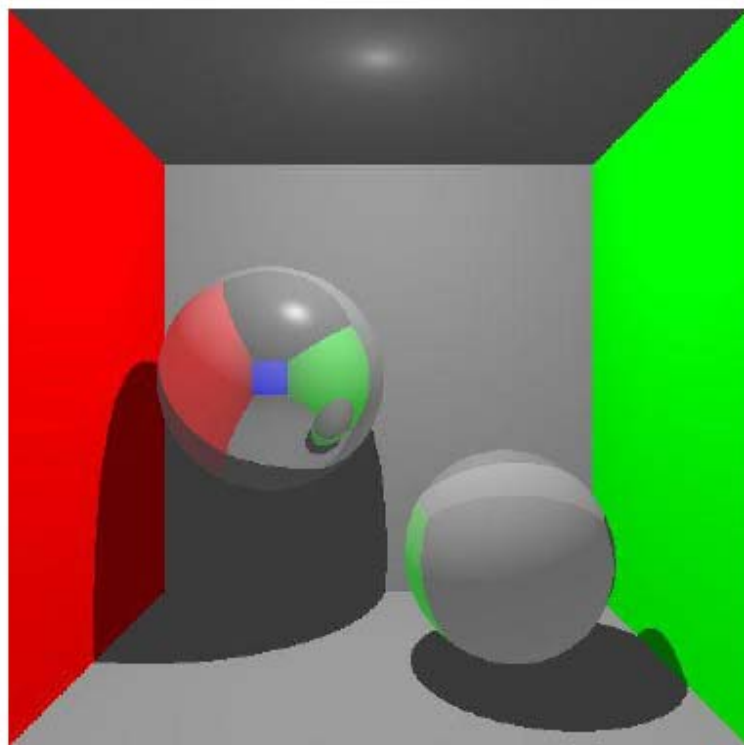
- **Pohledově nezávislé řešení (radiozita)**
- **Pohledově závislé řešení (sledování paprsku)**

Měkké stíny



DavidAmbrož, <http://www.shadowstechniques.com/>

Kaustiky a vzájemný difúzní odraz



obr. 3-1 Vlevo ukázka raytracingu, s typickými ostrými stíny, vpravo ukázka photon mappingu s měkkými stíny a kaustikou

Jiří Tuháček, diplomová práce, CGG FEL

Pohledově nezávislé řešení (radiozita)

2 kroky:

1. vypočte rozdělení světla ve scéně
 - vzájemnou viditelnost plošek
 - matice konfiguračních faktorů
 - řeší radiozitní rovnici
 - pohledově nezávislé
2. Zobrazí pohled
 - jen určí viditelnost pro polohu kamery

Pohledové závislé řešení

- Metody vycházející od pozorovatele (pixelu)
- Metody vycházející od světla
- Dvousměrové metody

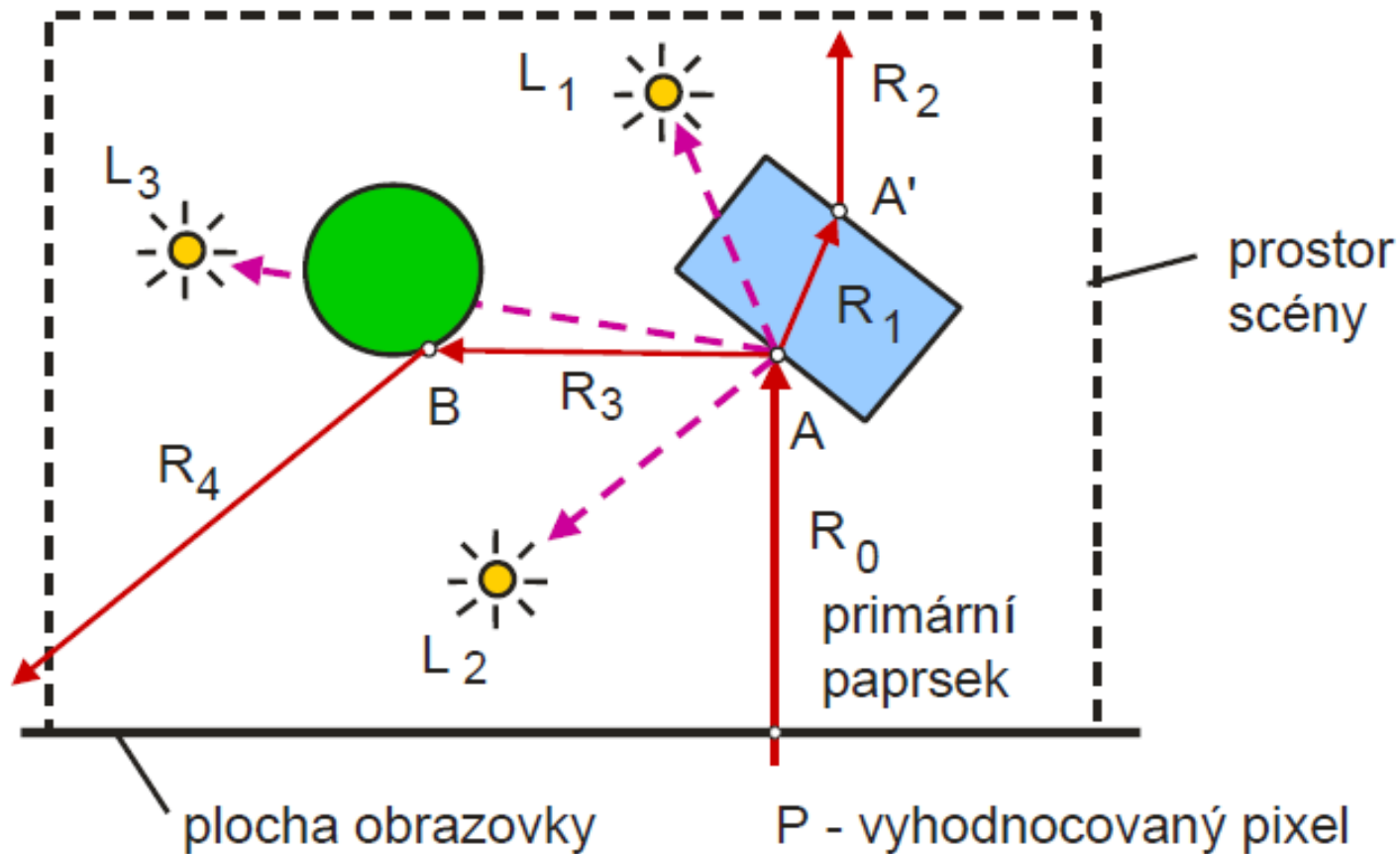
Metody vycházející **od pozorovatele** (*gathering methods*)

- Postupují proti směru světla
(z pixelů ke světlům)
- Akumulují příspěvky podél dráhy paprsku
- Zpětné sledování paprsku (*Ray tracing*) A
- Sledování cesty (*path tracing*) – BRDFB

Metody vycházející **od světelných zdrojů** (*shooting meth.*)

- (Photon tracing = Ray tracing pozadu)
- Monte Carlo sledování světla (*light tracing*) C

A. Zpětné sledování paprsku (Ray tracing)



Obrázek převzat od doc. Žáry

Zpětné sledování paprsku (Ray tracing)

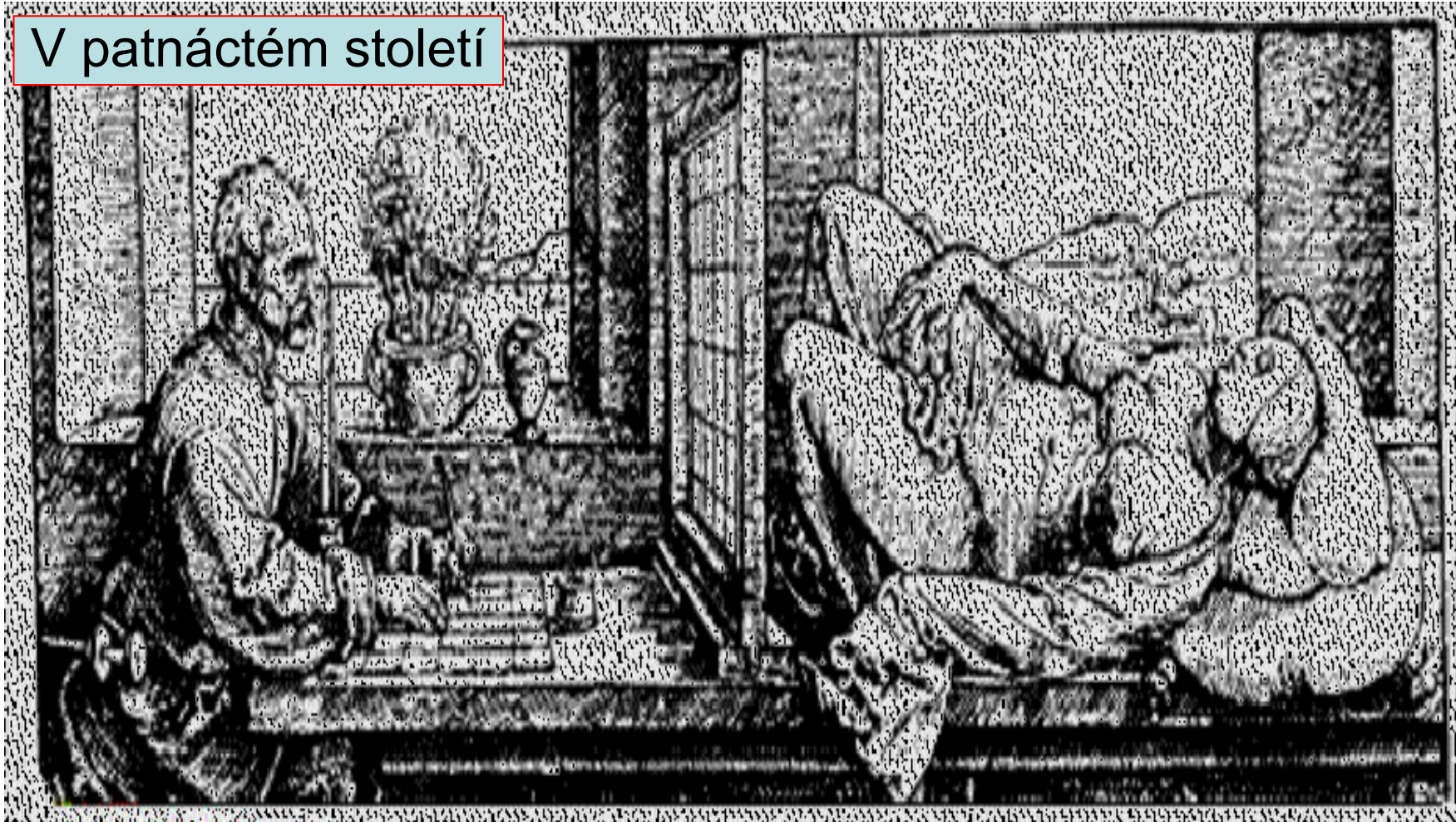
SledujPaprsek (paprsek R , hloubka rekurze H)

1. Nalezni průsečík P paprsku R s nejbližším tělesem ve scéně
2. Pokud průsečík P neexistuje *// paprsek opustil prostor scény*
přiřaď paprsku R barvu pozadí a skonči
3. Ke každému světelnému zdroji vyšli z bodu P stínový paprsek a pokud k němu paprsek dorazí, označ světelný zdroj jako nezakrytý
4. Vyhodnoť příspěvky osvětlení v bodě P od všech nezakrytých světelných zdrojů
5. Pokud hloubka H nepřekročila maximální hloubku sledování, vyšli:
 - (a) odražený paprsek RR voláním **SledujPaprsek** (RR , $H + 1$)
 - (b) lomený paprsek RT voláním **SledujPaprsek** (RT , $H + 1$)
6. Paprsku R přiřaď výslednou barvu jako součet příspěvků osvětlení, barvy odraženého paprsku RR a barvy lomeného paprsku RT

Algoritmus převzat od doc. Žáry

Zpětné sledování paprsku (Ray tracing)

V patnáctém století



<http://glasnost.itcarlow.ie/~powerk/Graphics/Notes/node12.html>

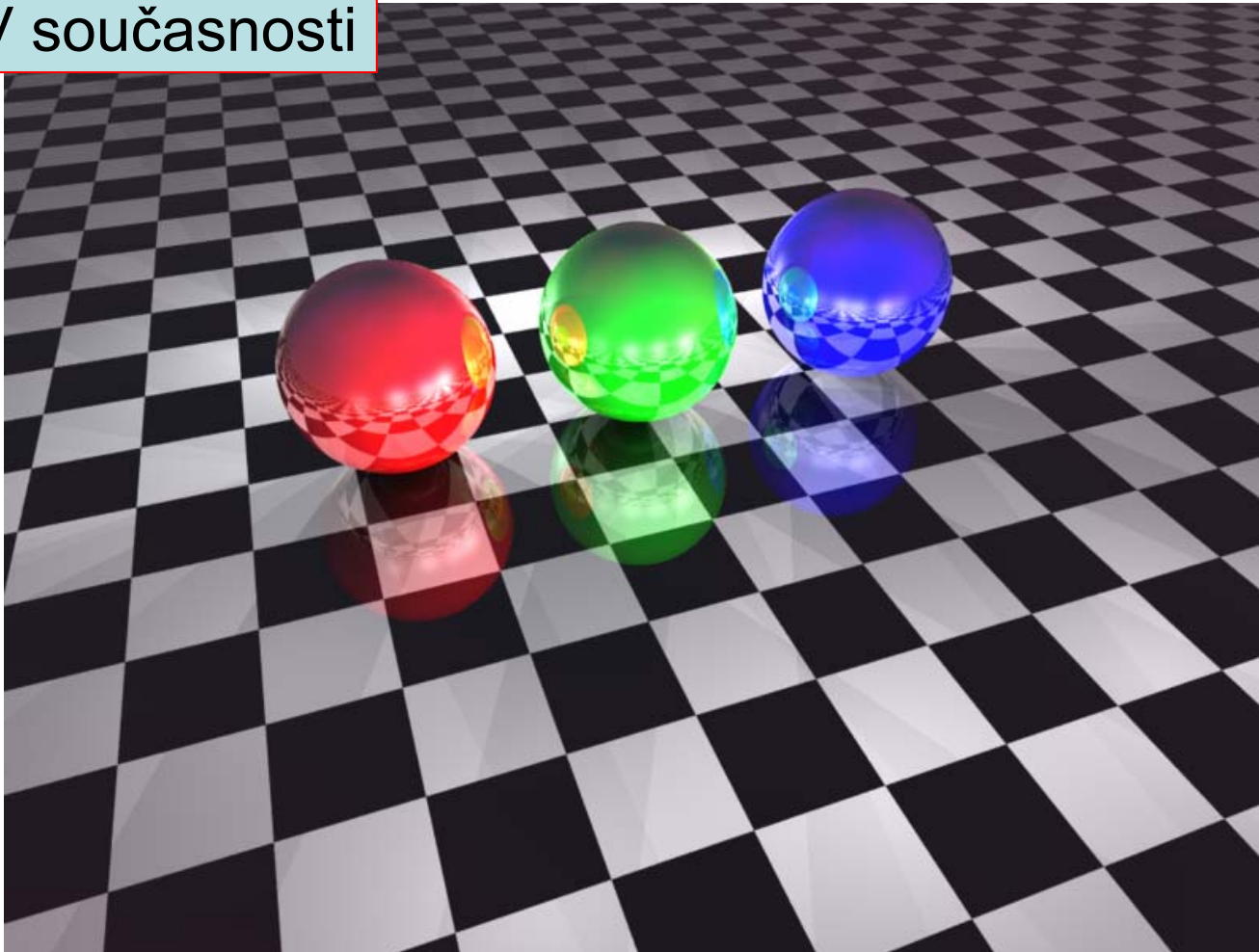
Zpětné sledování paprsku (Ray tracing)

V současnosti



Zpětné sledování paprsku (Ray tracing)

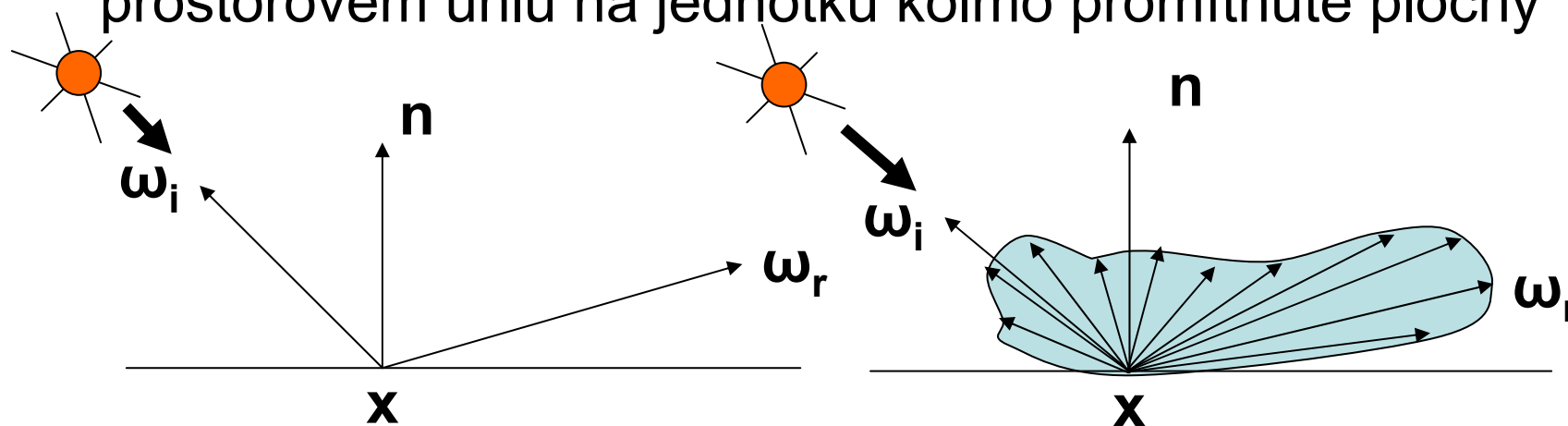
V současnosti



http://en.wikipedia.org/wiki/Image:Raytracing_reflection.png

Dvousměrová odrazová distribuční funkce (*Bidirectional reflectance distribution function*)

- Popisuje poměr **radiance odražené ve směru ω_r** k **radianci dopadající ze směru ω_i** , v určitém bodě x
- **Radiance** = přijatý či vyzářený výkon na jednotkovém prostorovém úhlu na jednotku kolmo promítnuté plochy

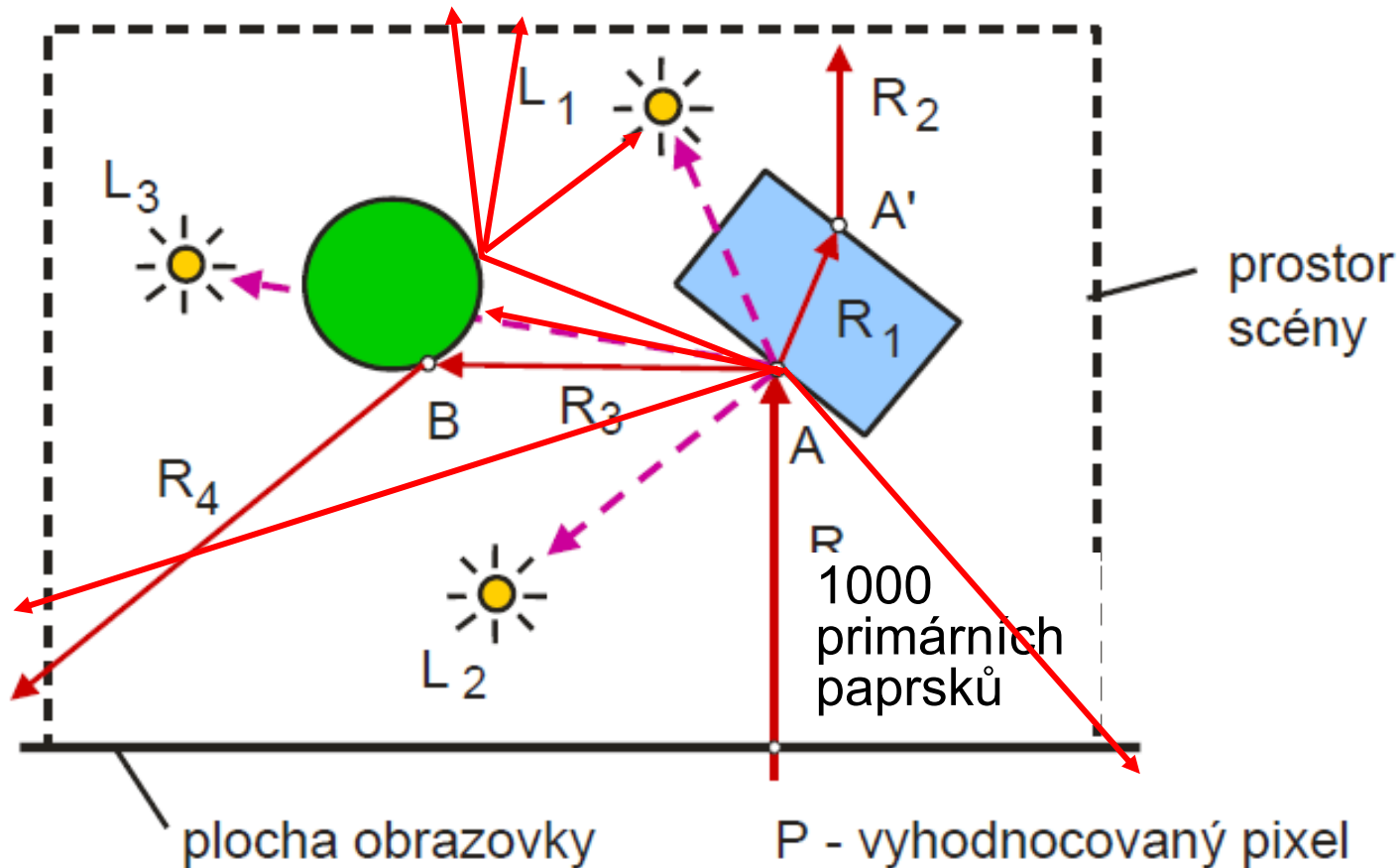


Polární souřadnice, 2 směry, každý 2 úhly => 4 parametry

B. Sledování cesty (*path tracing*) – BRDF

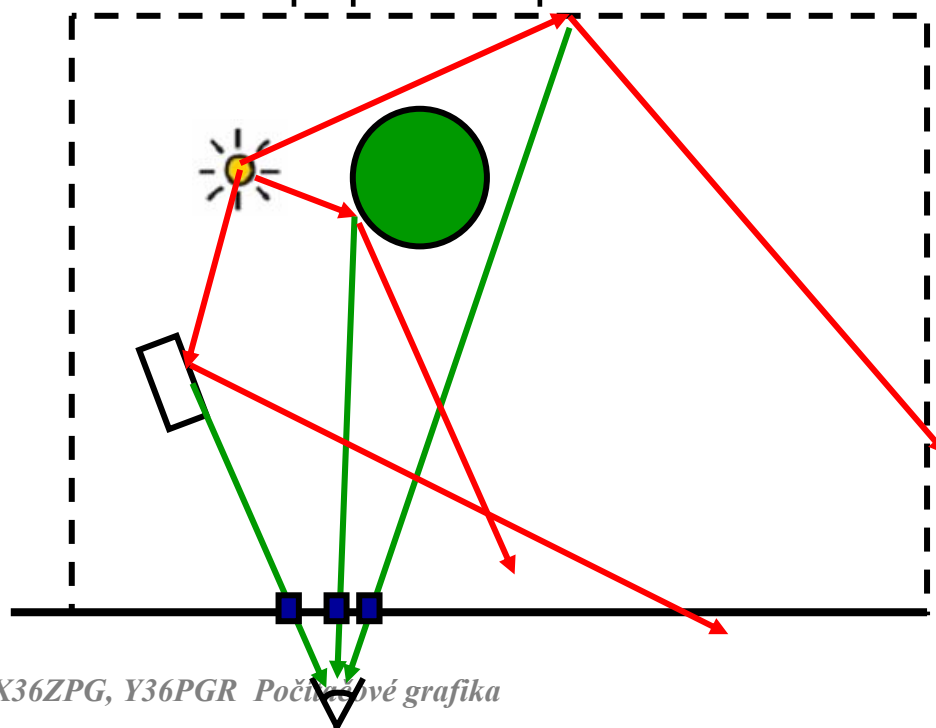
- Monte Carlo sledování cesty (od pozorovatele)
- Generuje 1000-10 000 paprsků na pixel
- Každý sleduje podobně jako ray-tracing,
ale
- Volí **náhodný úhel odrazu** a pro tento úhel vyhodnotí BRDF
- Zvládne kaustiky i difúzní odraz

B. Sledování cesty (*path tracing*) – BRDF



Monte Carlo sledování světla (*light tracing*)

- Sleduje světlo od světelných zdrojů (*Shooting method*)
- Určí se výkon odcházejícího paprsku
- Pak se paprsek rekurzivně sleduje. V místech odrazu se určí množství světla jdoucí k pozorovateli
- Průsečík paprsku k pozorovateli se zobrazovací rovinou



= příspěvek k barvě pixelu

Problémy:

- Mnoho nevyužitých paprsků
- Velký šum

Dvousměrové metody (*bidirectional methods*)

- Současně paprsky **od světla** a **od pozorovatele**

a) Dvousměrové sledování cesty (*bidirectional path tracing*)

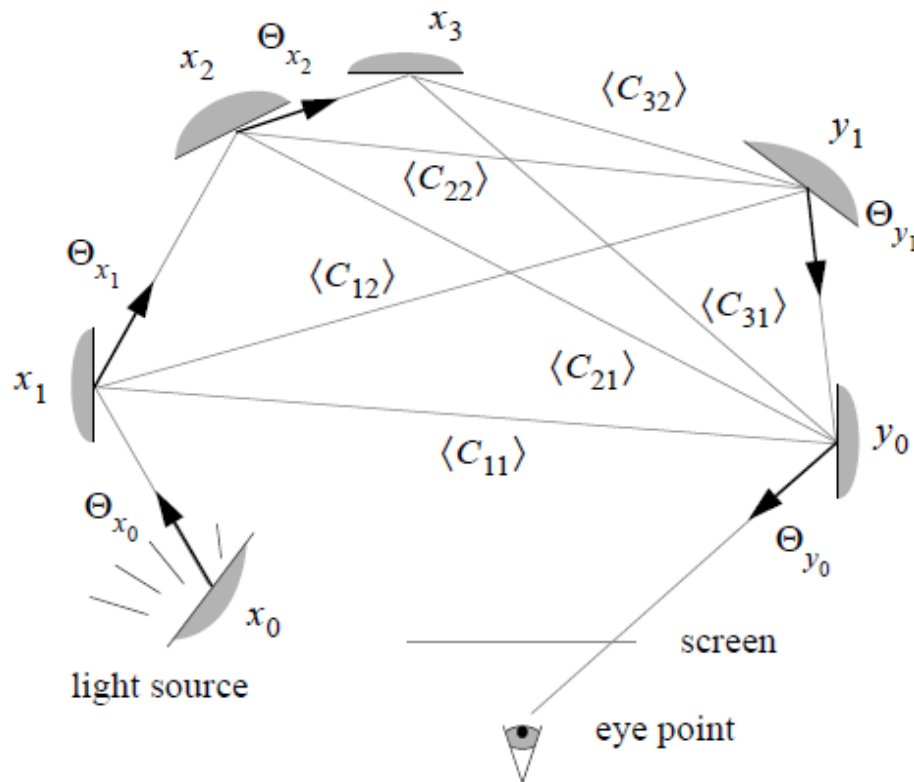
- Najdou se průsečíky se scénou (x_i) a (y_k)
- Najdou se všechny kombinace x_i a y_k (každý s každým)
- Určí se viditelnost a vzájemné příspěvky radiance

b) Fotonové mapy

1. Vystřelí *fotony* od zdrojů, uloží se do 3D fotonové mapy (fotony od všech světla)
2. Pak ray tracing
najde průsečík, sesbírá fotony v jeho okolí => tj. příspěvek od všech světla

Dvousměrové sledování cesty (*bidirectional path tracing*)

- Současně light tracing (shooting) a path tracing...



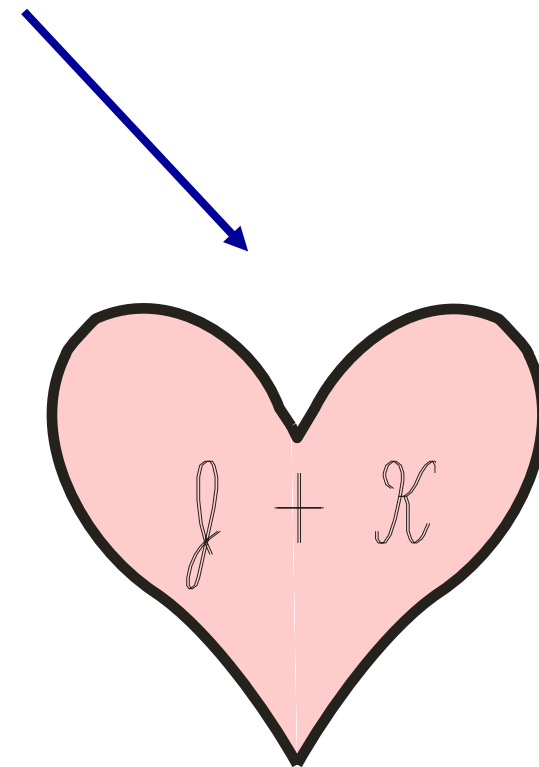
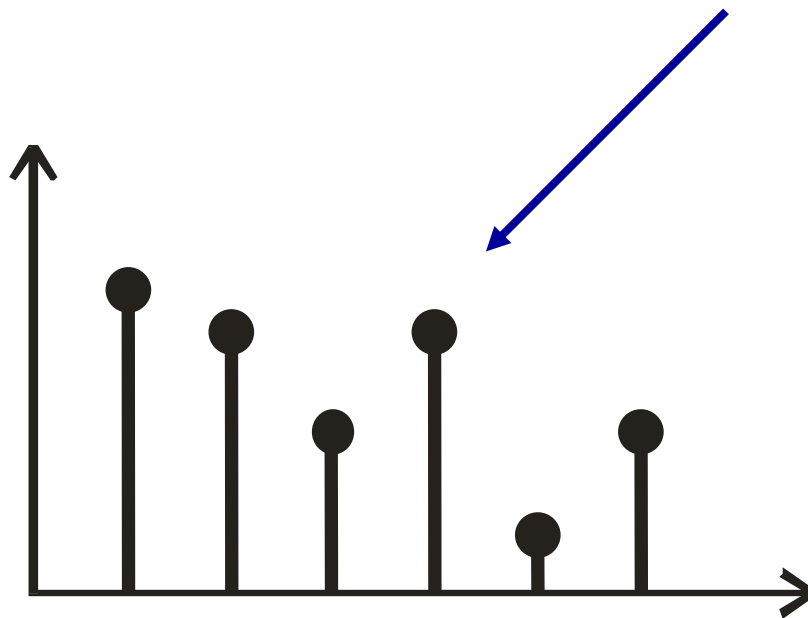
II. Vizualizace

© Felkel -> Žára -> Felkel

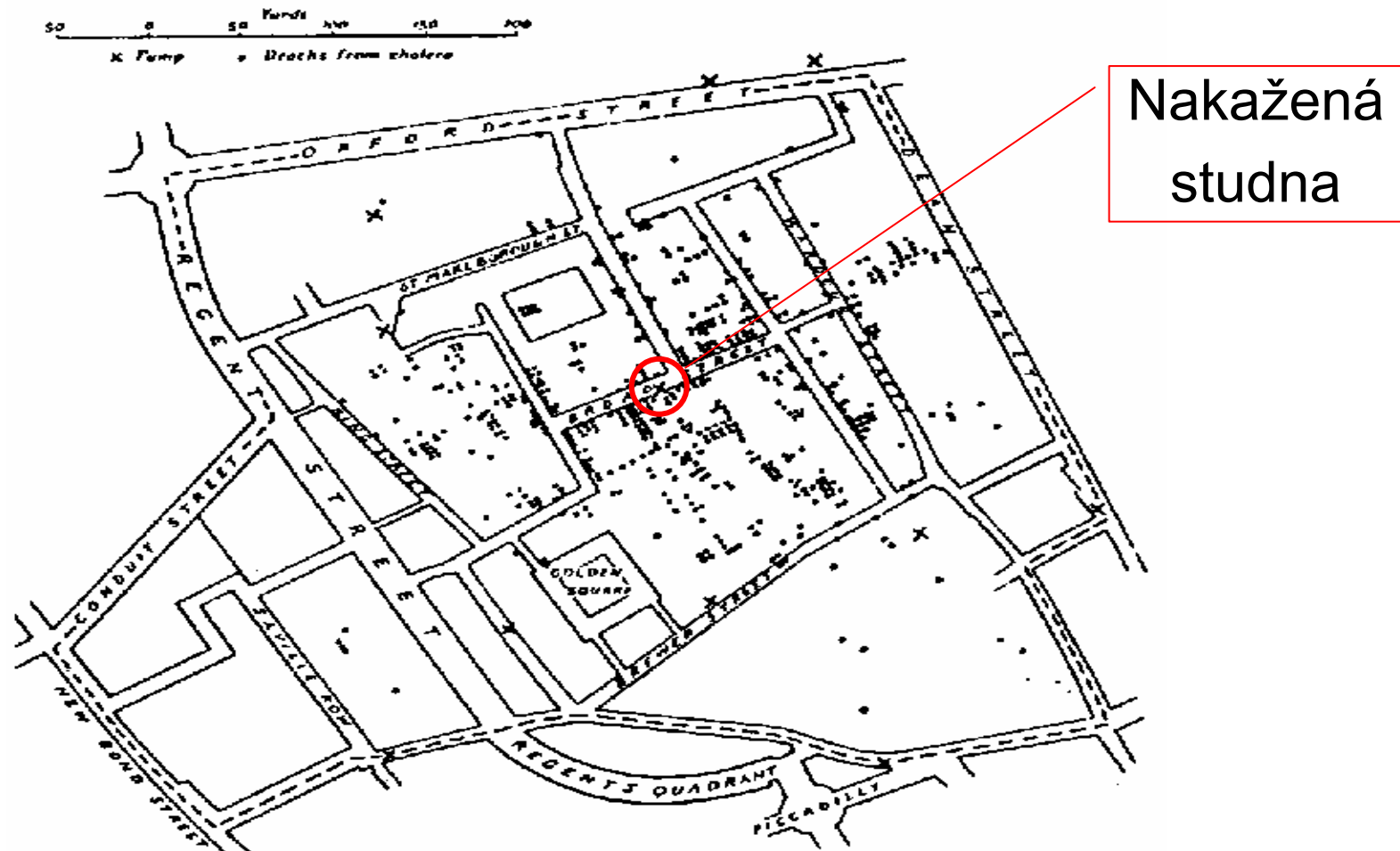
II. Vizualizace

- Co je vizualizace
(vědecká i „nevědecká“ :-)
- Proces vizualizace
- Získávání dat
- Datové typy
- Zpracování dat

= postup, vyjádření hodnot a vztahů obrazem

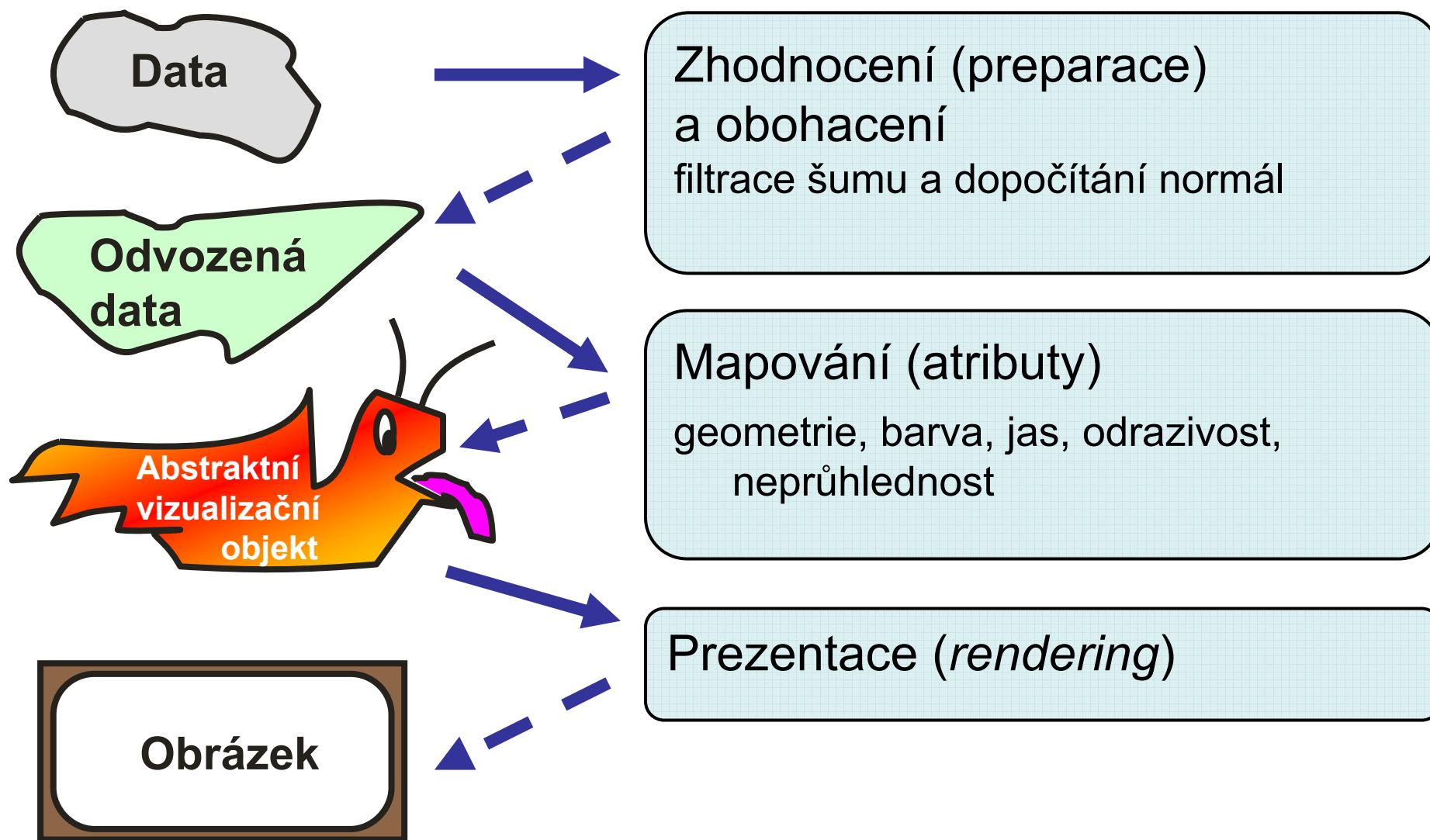


Počátky vizualizace



Cholera v Broad Street, Londýn, 1854

Vizualizační řetězec



Řetězec transformací

- konvertuje hrubá data na obraz srozumitelný člověku,
- zachovává integritu (celistvost, úplnost) informace

- **Názornost** zobrazení
 - získání co nejlepší představy o zobrazované funkci, jejím průběhu
 - cílem nemusí být krásné obrazy ale pochopení (*insight*)
 - interakce, animace (*steering*)
- **Věrnost**, pravdivost
 - vizualizace nesmí zkreslovat a způsobovat vznik nových, nepůvodních jevů (*artefakty*)
- **Rychlost** zobrazení (alespoň jednotky fps)

Pozorováním (měřením reálné situace):

Sada vzorků + interpolace

Při převzorkování vznik artefaktů

Simulací (modelováním):

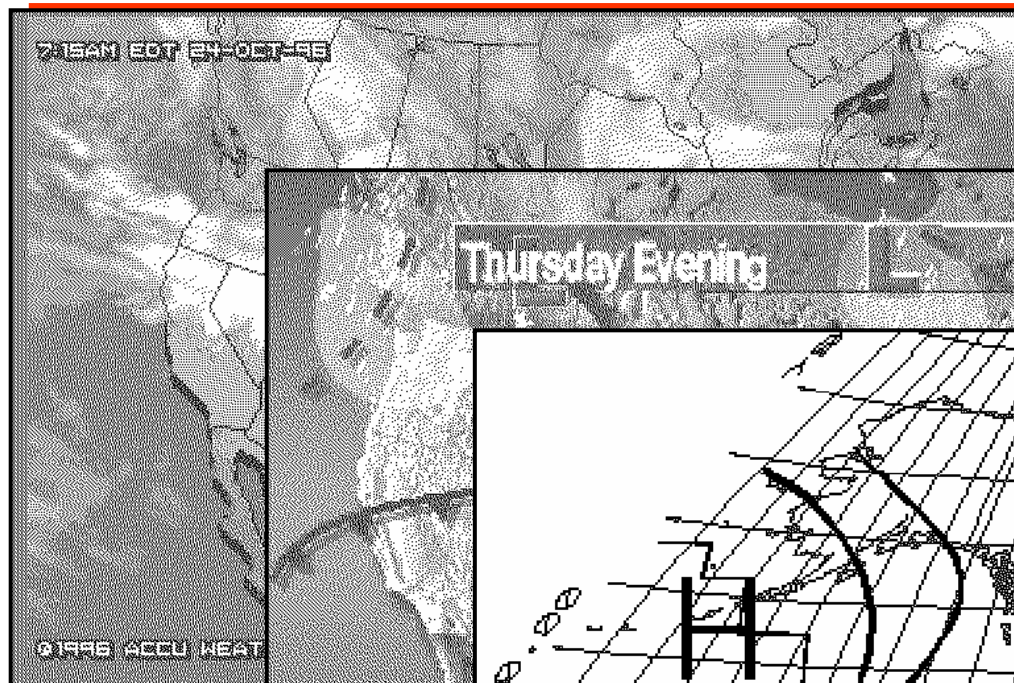
abstraktní model - fyzika, mat. popis

simulační model - realizovatelné přiblížení

simulace - experimenty

interpretace

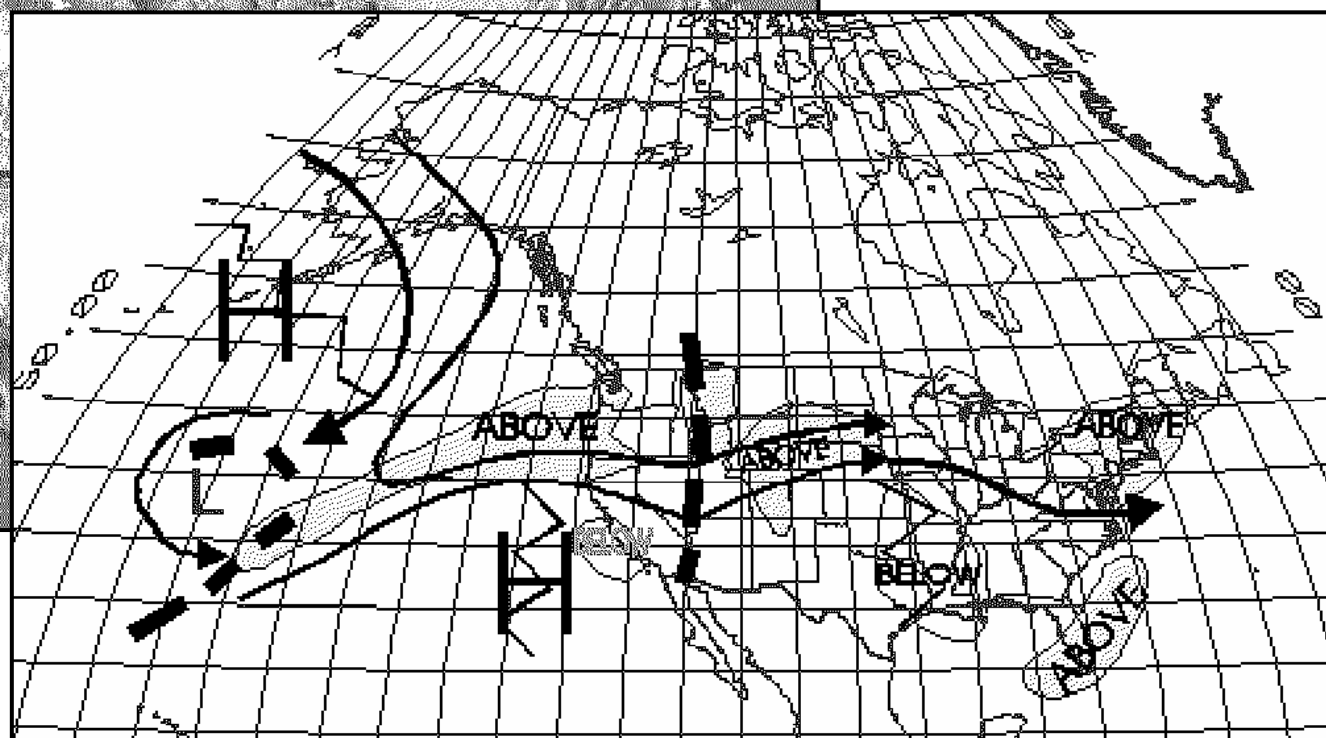
Příklad: vizualizace pozorování



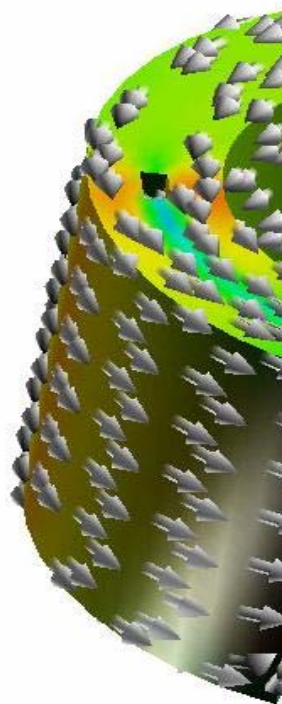
fotografie

Abstrakce: čela front

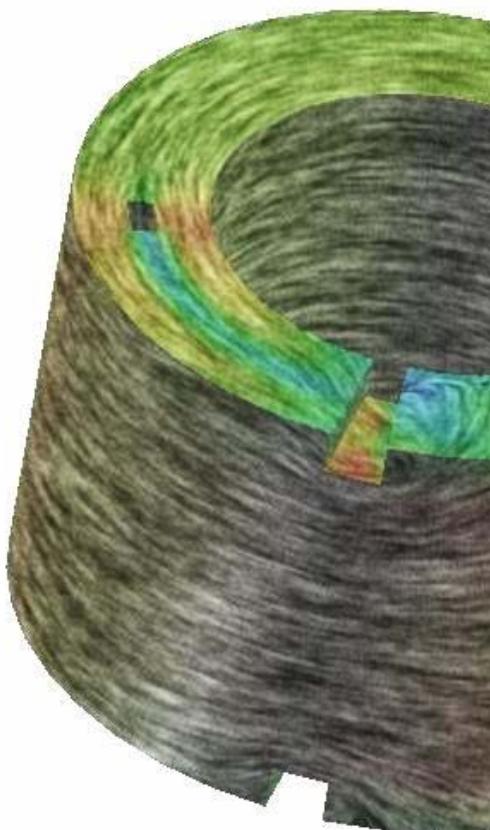
Abstrakce: další info



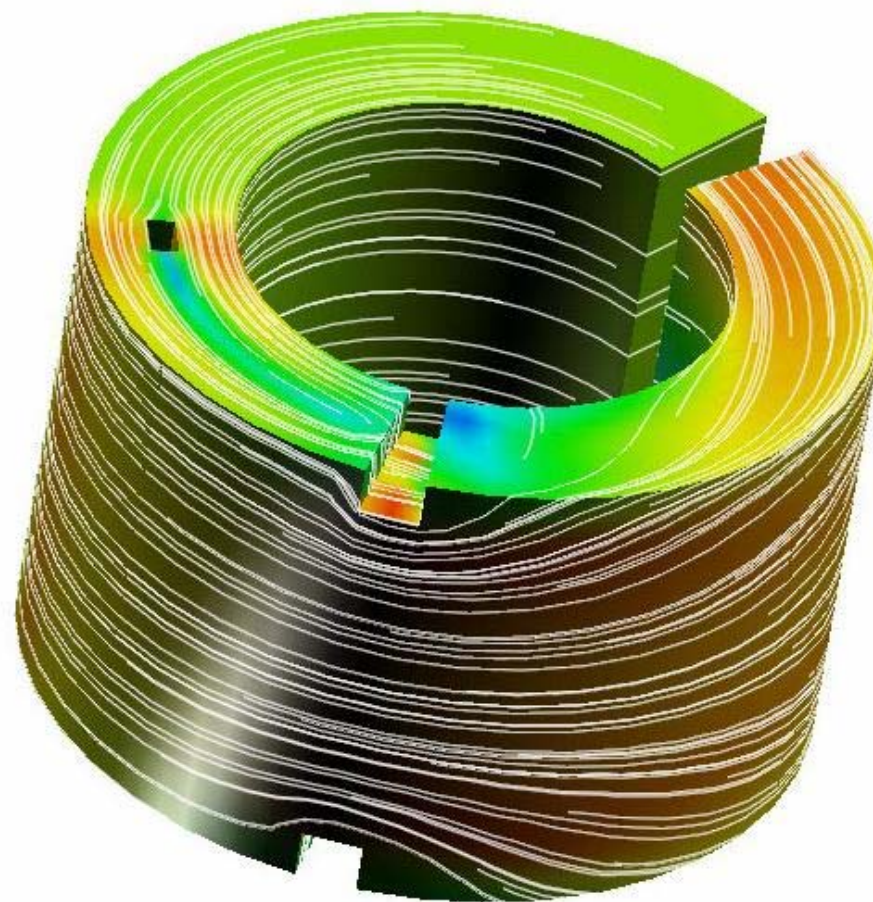
Příklad: vizualizace simulace proudění



Přímá vizualizace



Vizualizace textur



Vizualizace geometrických objektů
(*streamlines*)

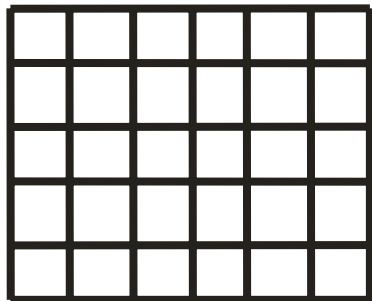
Izolované bodové vzorky (*samples*)

např. z meteorologických balónů

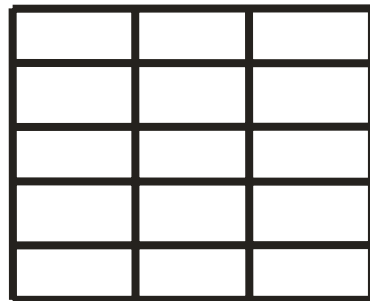
Mřížky (*lattice*)

- **Pravoúhlá mřížka** (*ortogonal, rectilinear, perimeter*)
Kartézská (*cartesian*)
Pravidelná (*regular, uniform*)
- **Strukturovaná** (*nonrectilinear, curvilinear*)
- **Nestrukturovaná** (*unstructured, pyramid*)
- **Blokově strukturovaná**
- **Hybridní**

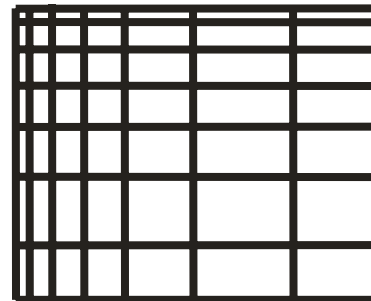
Druhy mřížek



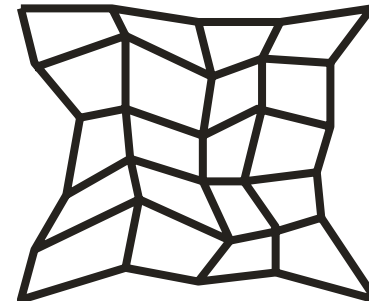
kartézská



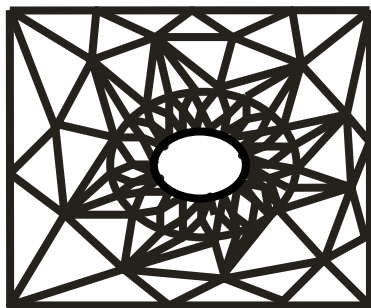
pravidelná



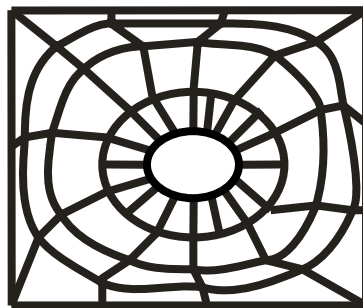
pravoúhlá



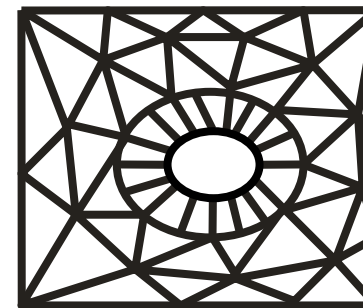
strukturovaná



nestrukturovaná



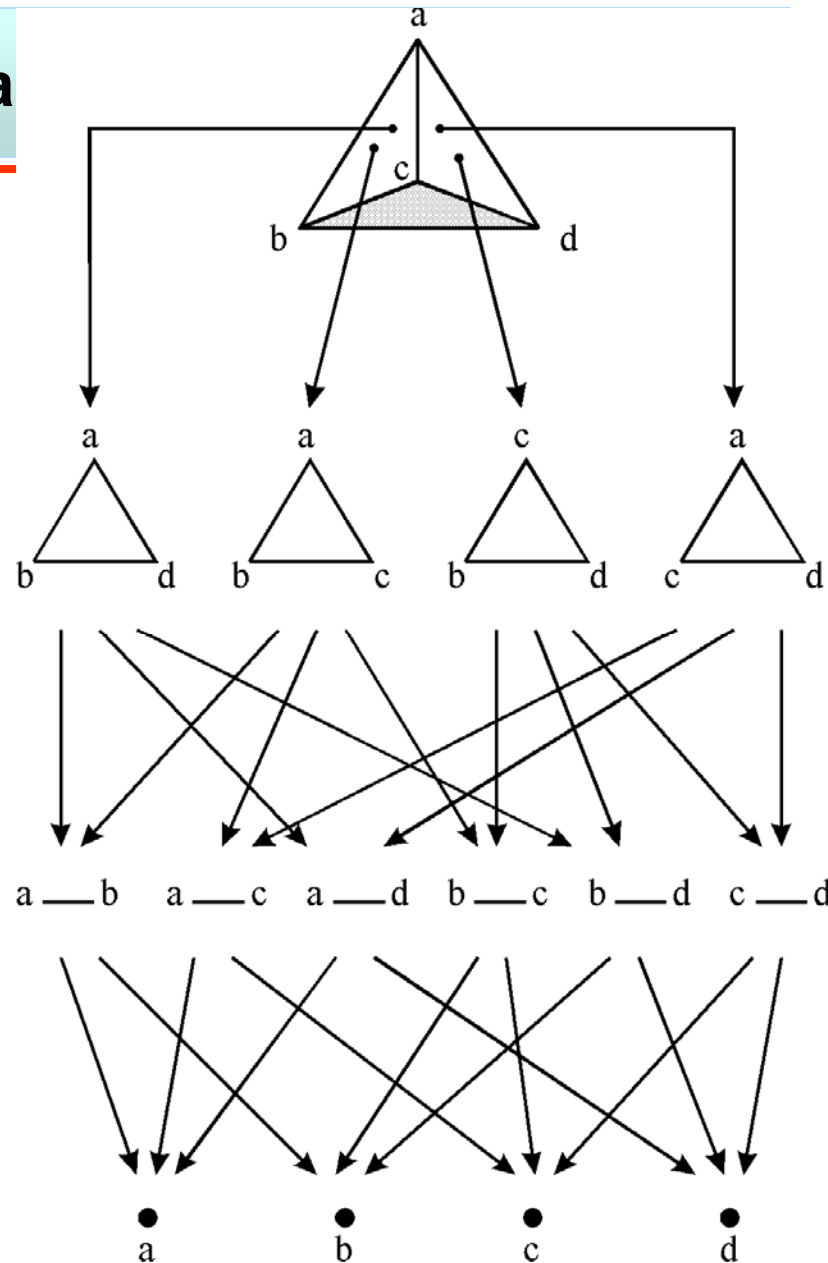
blokově
strukturovaná



hybridní

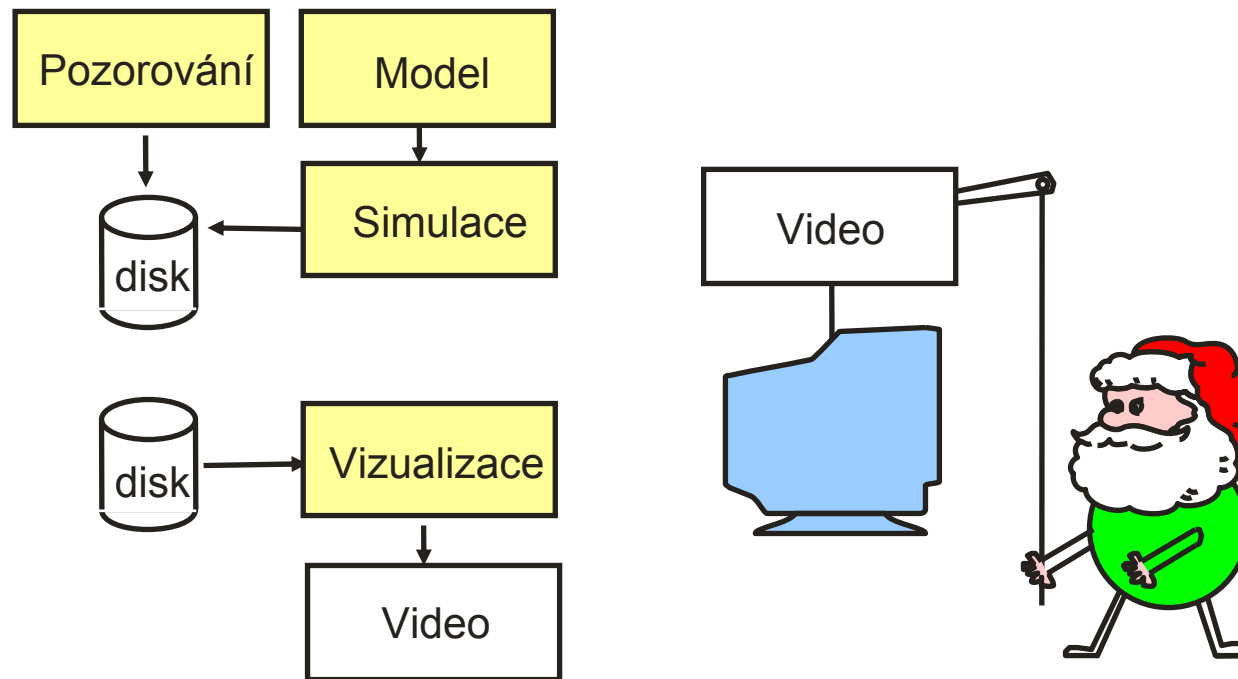
Pyramidální mřížka

VEFS reprezentace

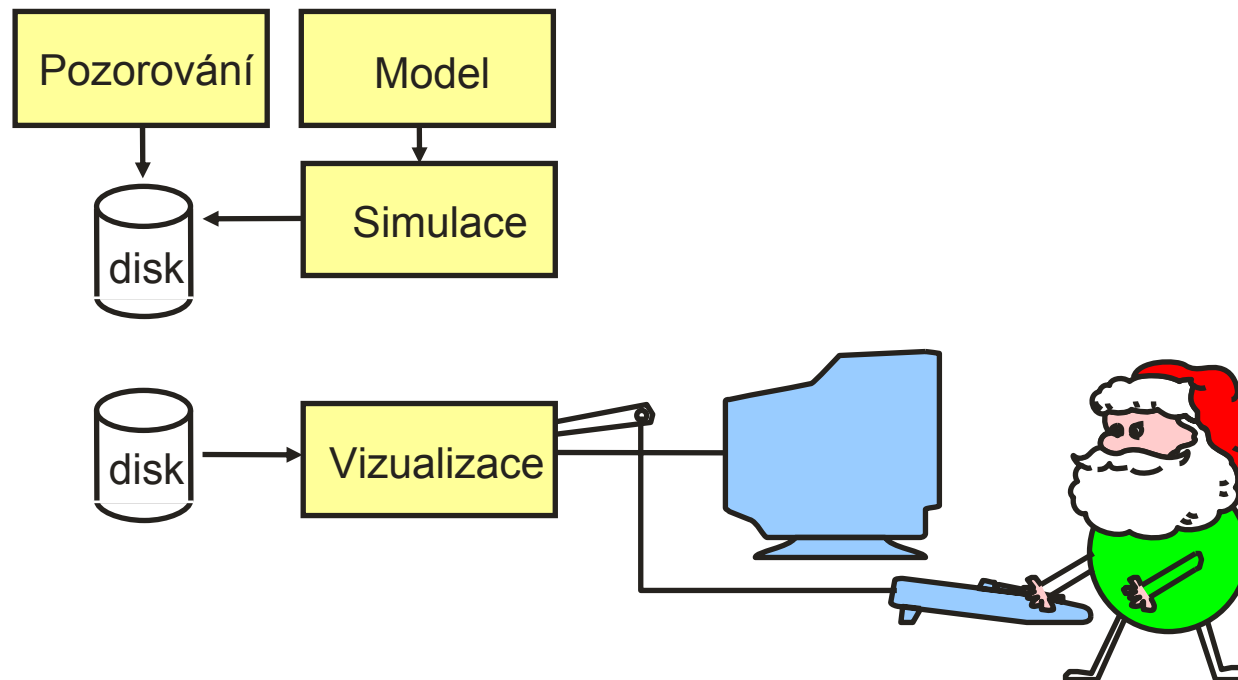


- **Následné zpracování** (*post-processing*)
 - filmový režim (*movie mode*)
 - interaktivní vizualizace
(*interactive post-processing*)
- **Sledování** (*tracking*)
- **Interaktivně řízená vizualizace**
(*interactive steering*)

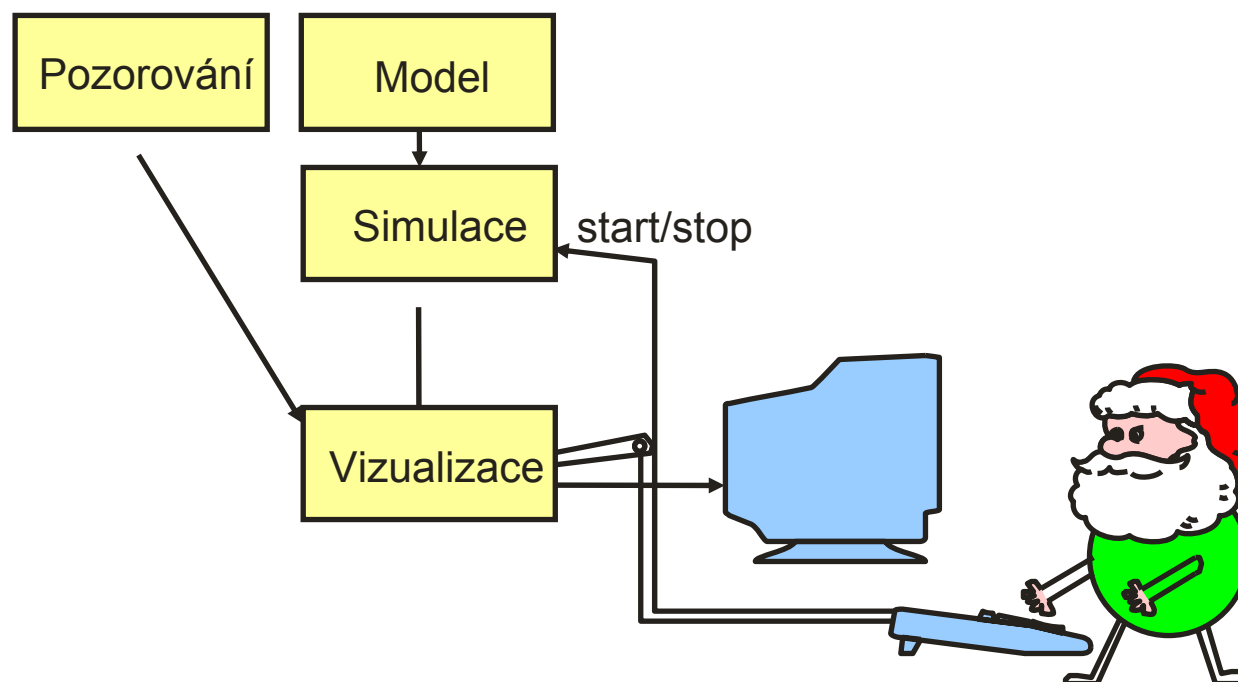
Filmový režim (*movie mode*)



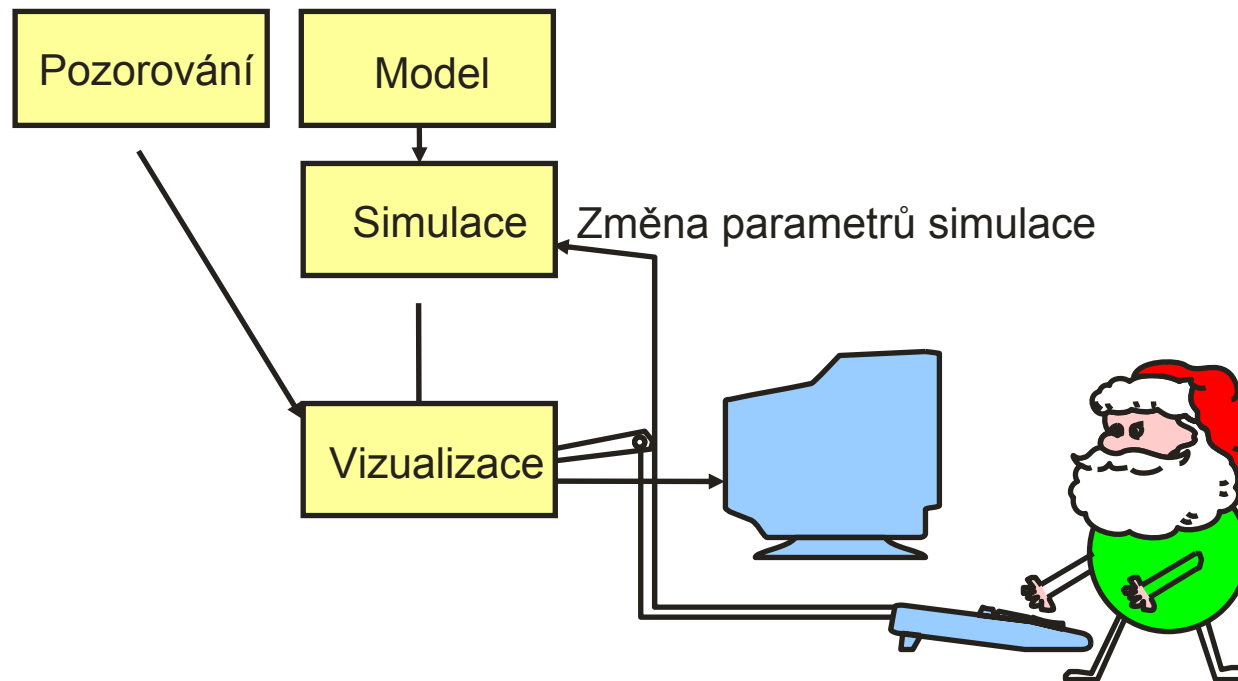
Interaktivní vizualizace (*post-processing*)



Sledování (*tracking*)

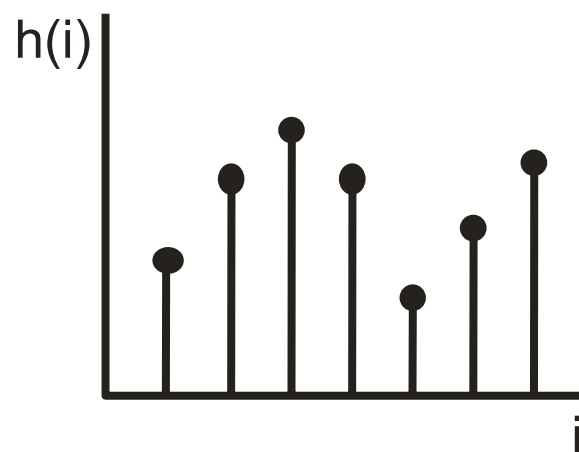


Interaktivní simulace (*interactive steering*)

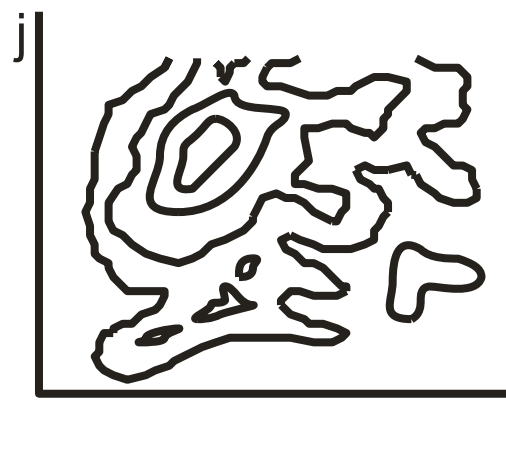


- Zhodnocení nejčastěji konvoluční filtrací:
 - Potlačení zkreslení (šumu)
 - Zdůraznění hran
 - Zvýšení kontrastu

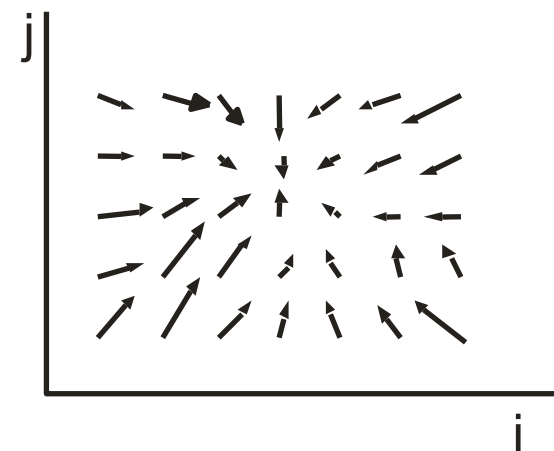
- Obohacení
 - dopočítáním normál
 Odhad symetrickou diferencí
 (a la bump textures)
 - Výpočet izočar



1D pole
skalárů



2D pole
skalárů



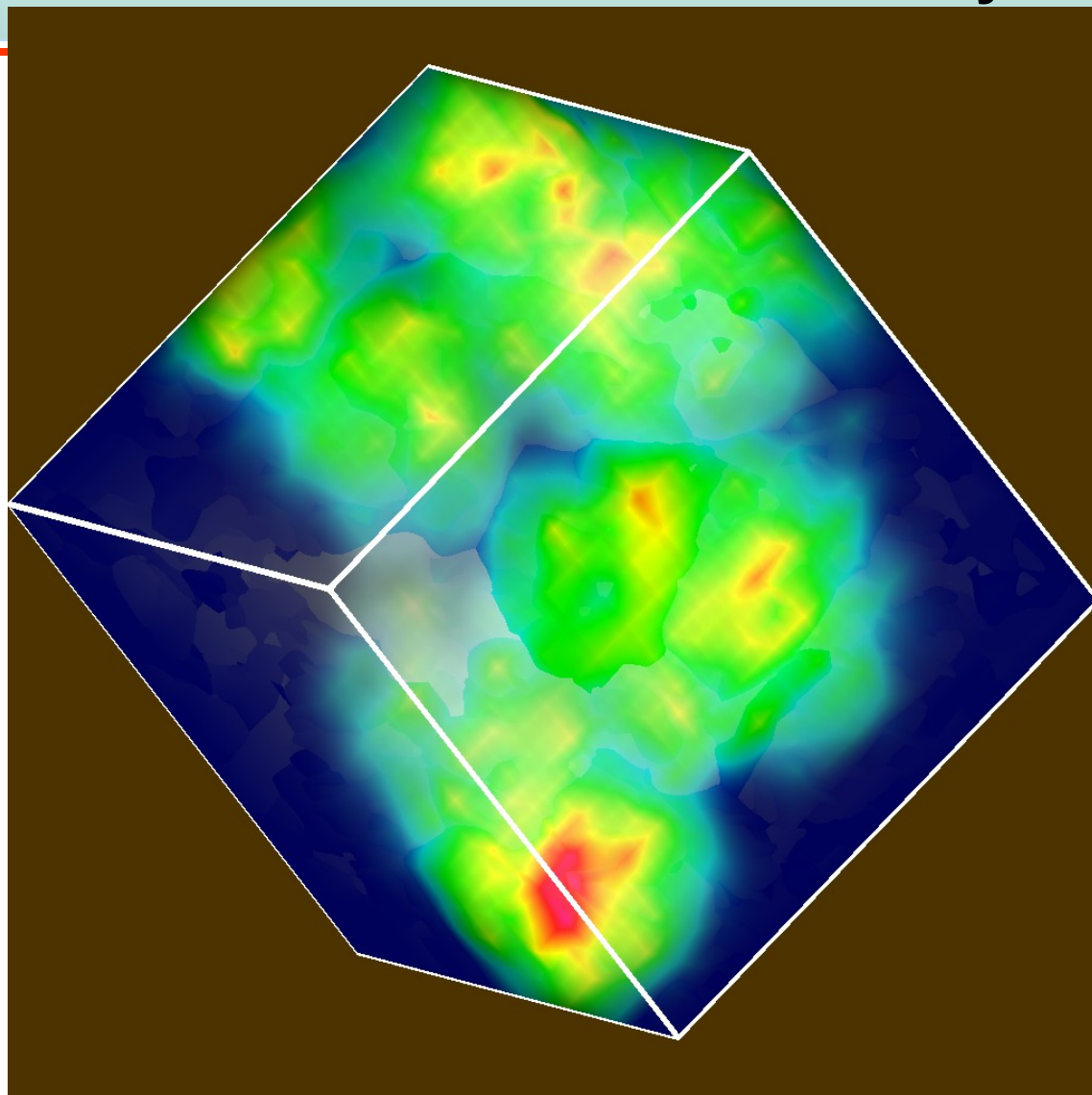
2D pole
vektorů

+

izočáry

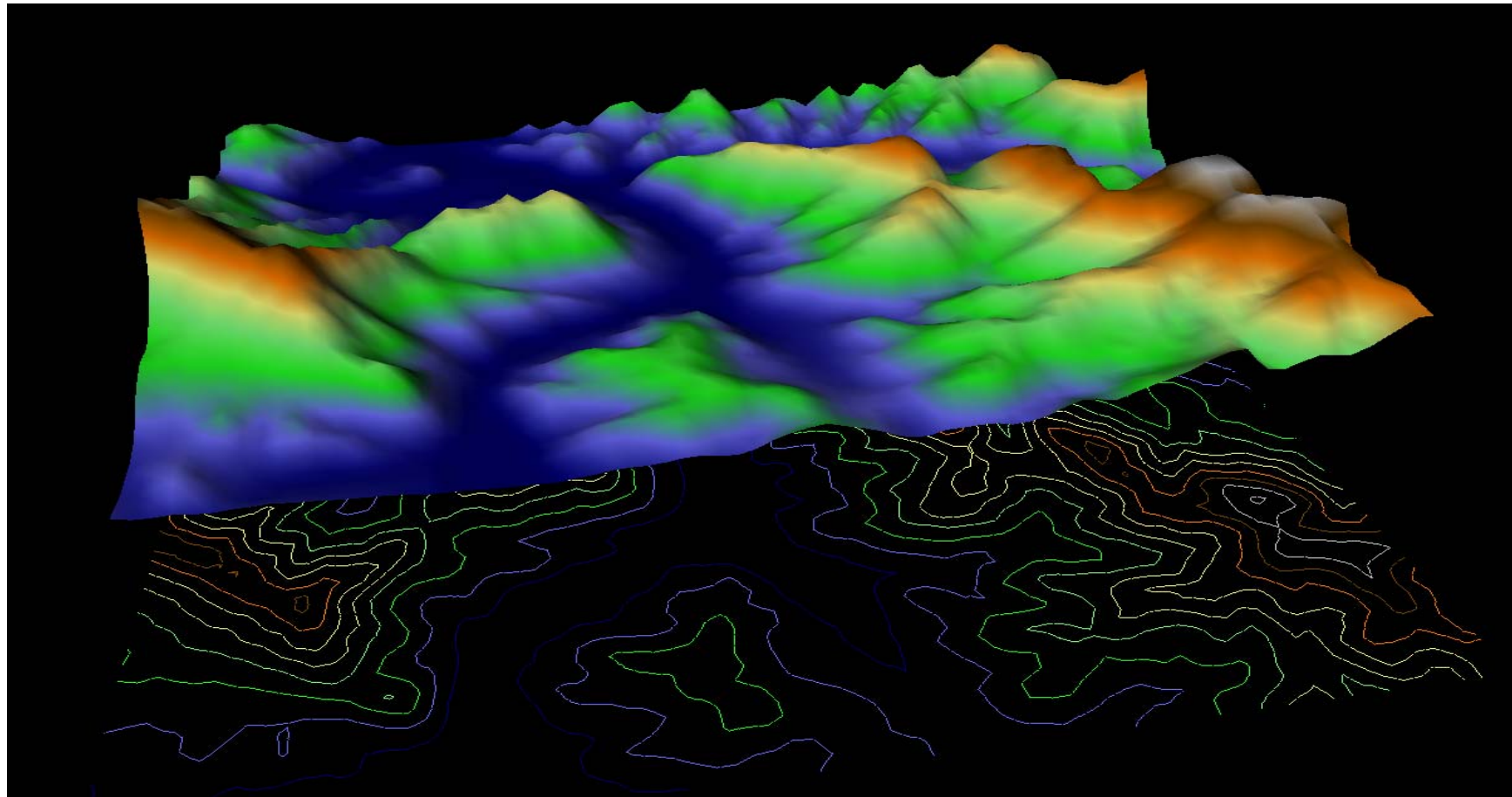
- Barva
- Tvar
- Orientace
- Velikost
- ...

Převod skalárů na barvy



vrhání
paprsku
+
trilineární
interpolace

2D pole skalárních dat

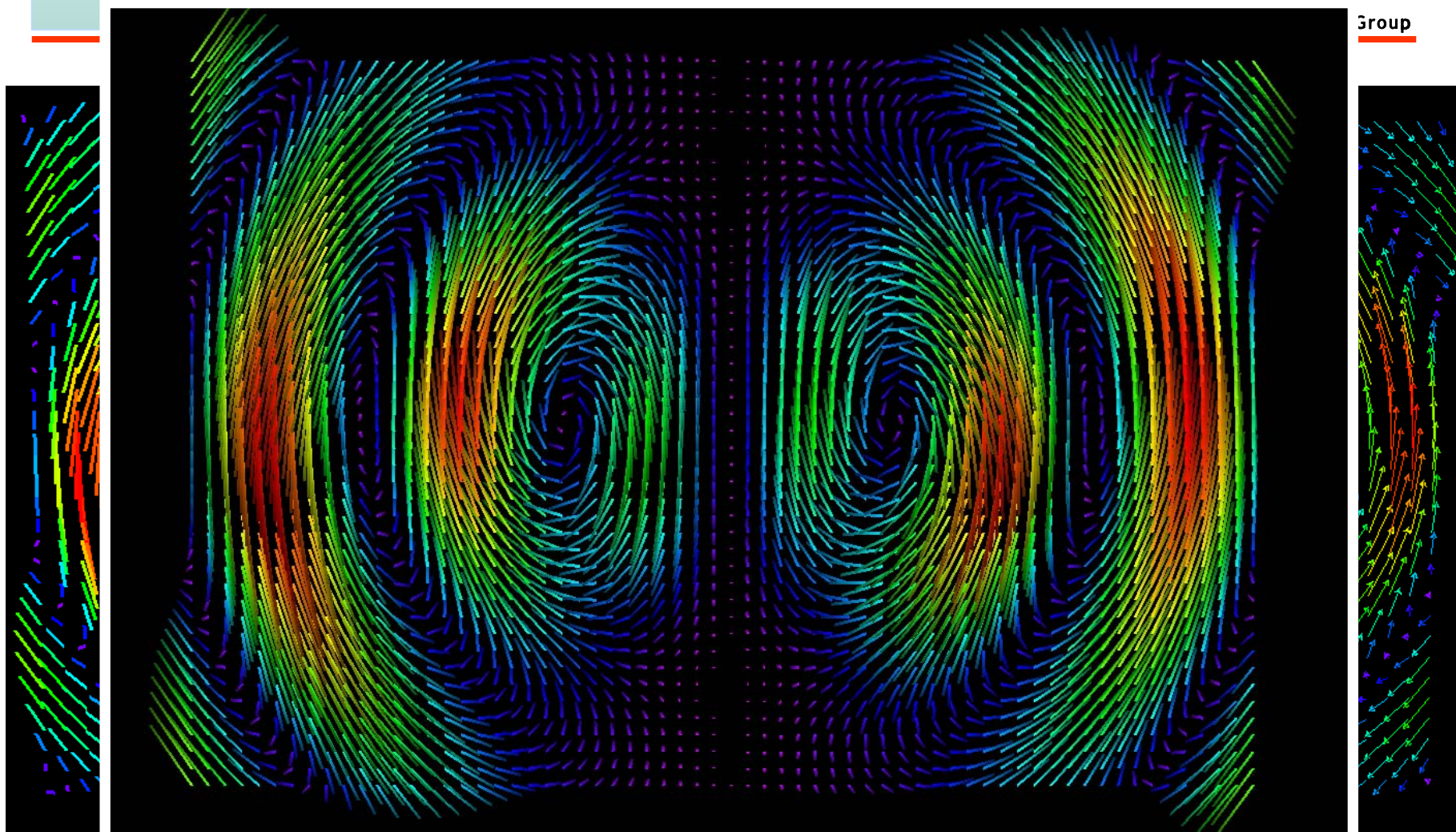


skalár => barva, výška + izočáry

2D obraz 3D pole vektorů

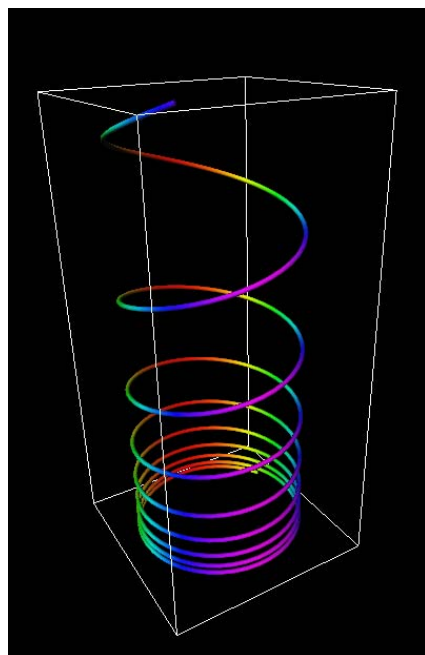


Group



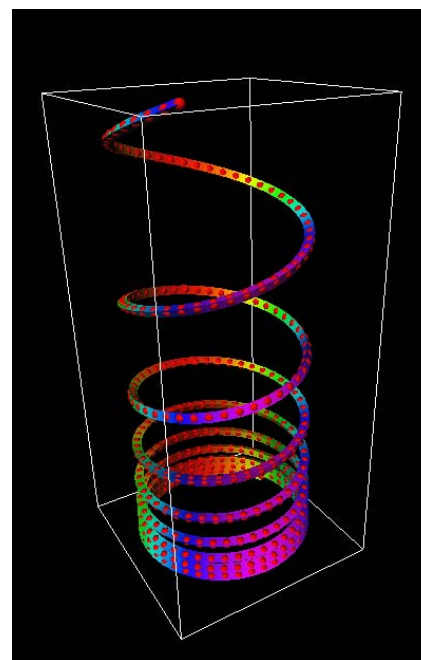
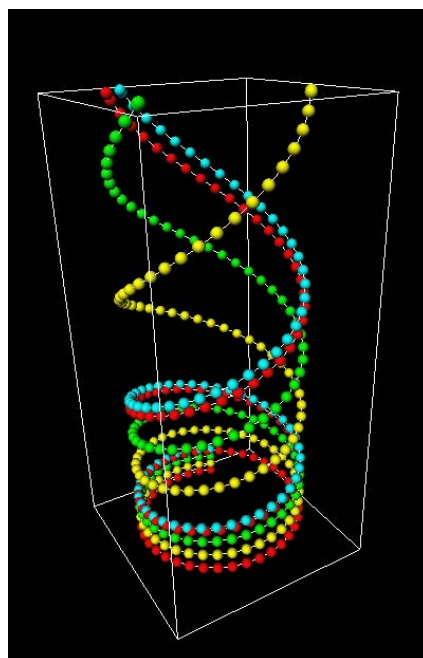
„stopy“, ztenčující se dle směru vektoru

Spirální vekt. data - *streamlines*

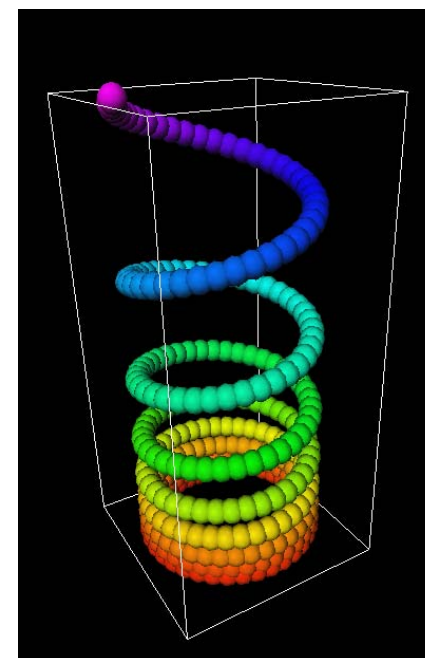


barva
~
osa x

více
křivek



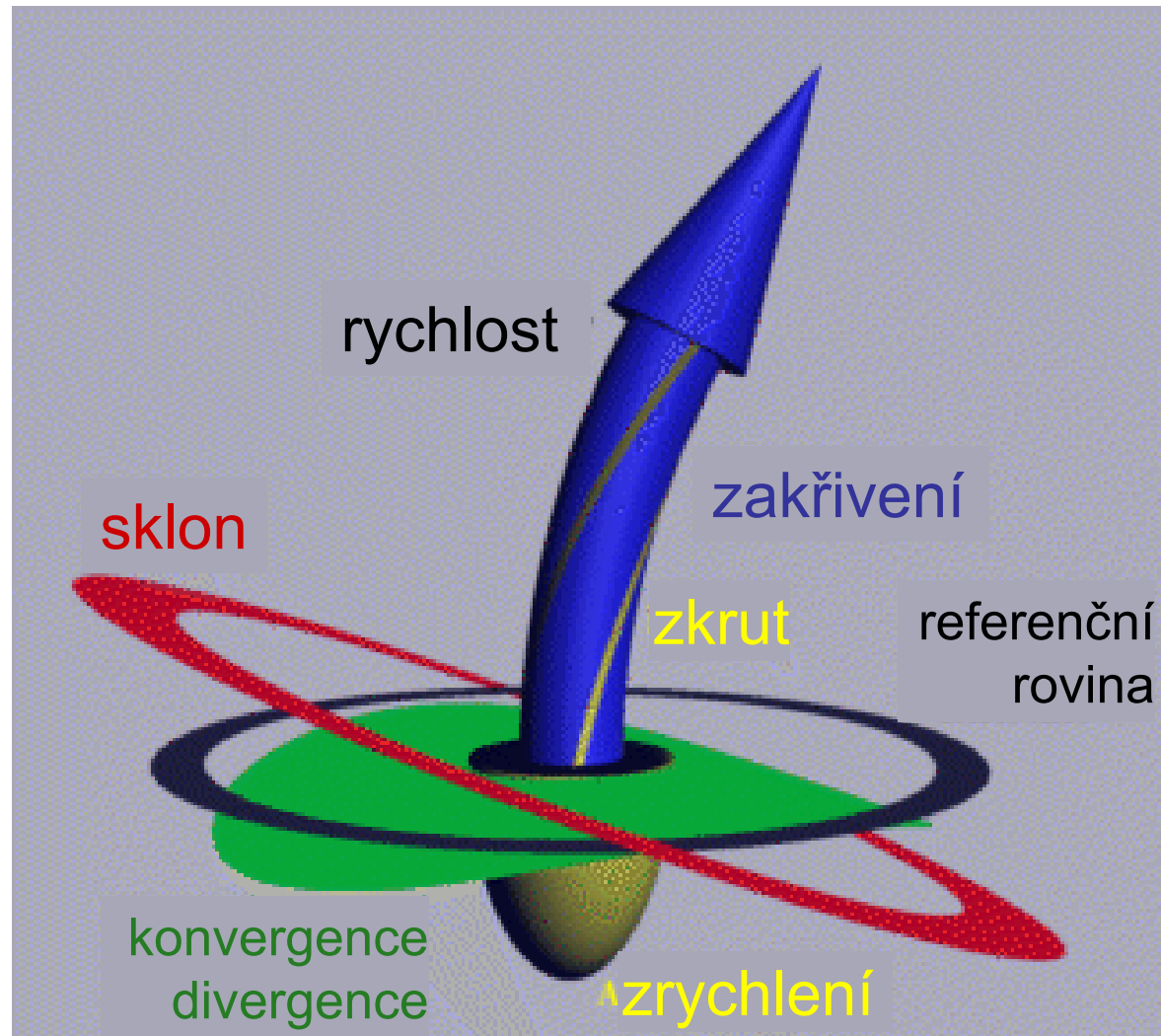
+
umístění
vzorků



pouze
vzorky

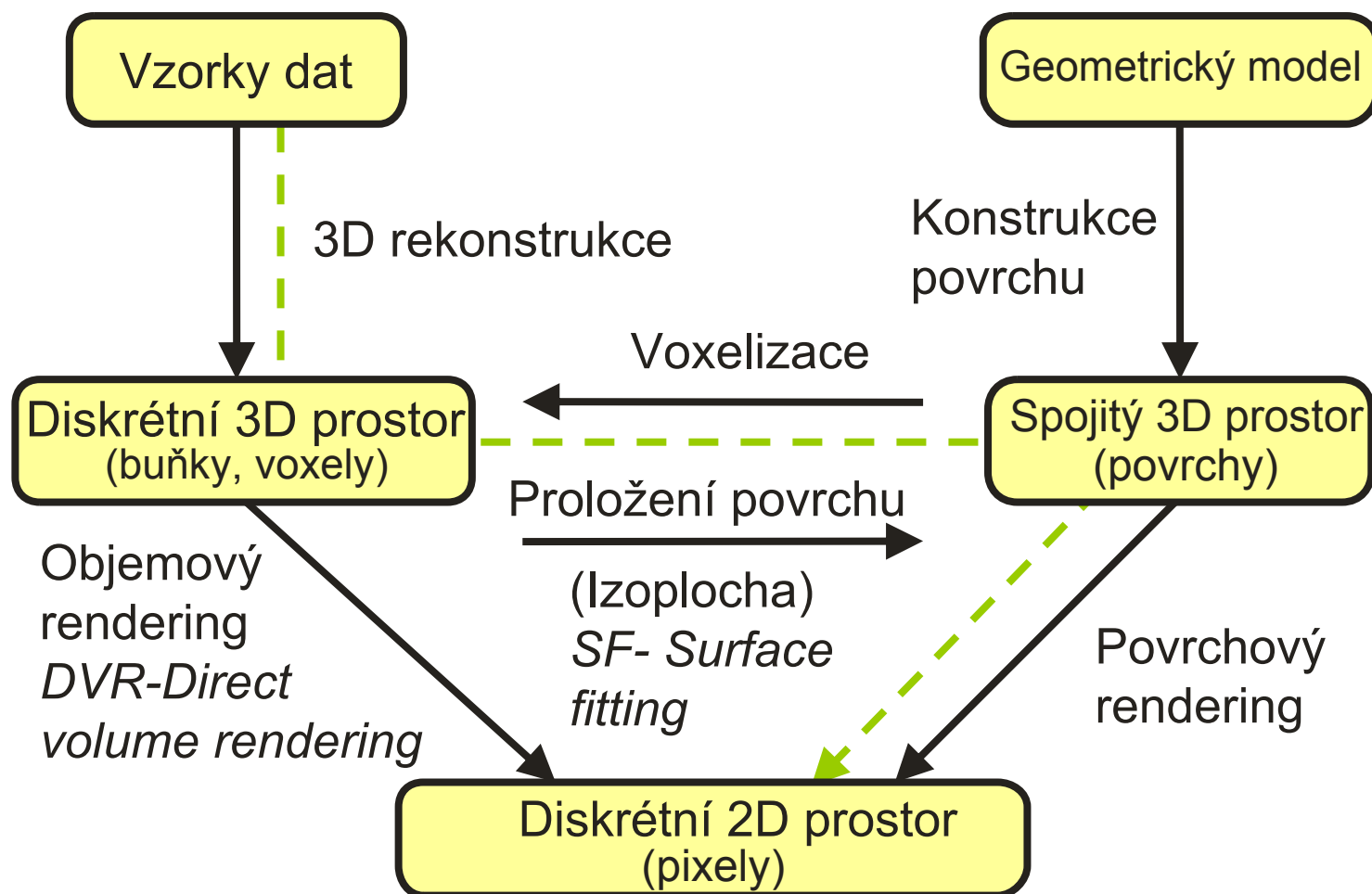
Vícerozměrné vektory

GLYF
=
ikona
s více
významy



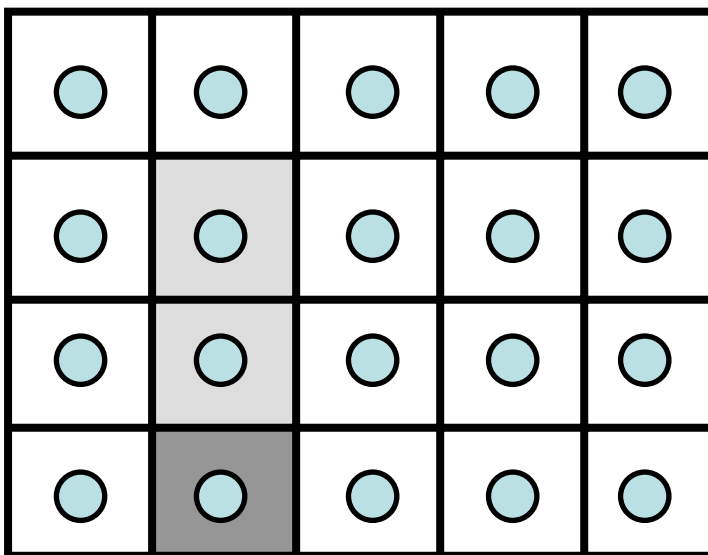
- Snímání dat
- Rekonstrukce z řezů
- Převod objem => B-Rep
- Decimace
- Přímé zobrazování objemu

3D voxelová data



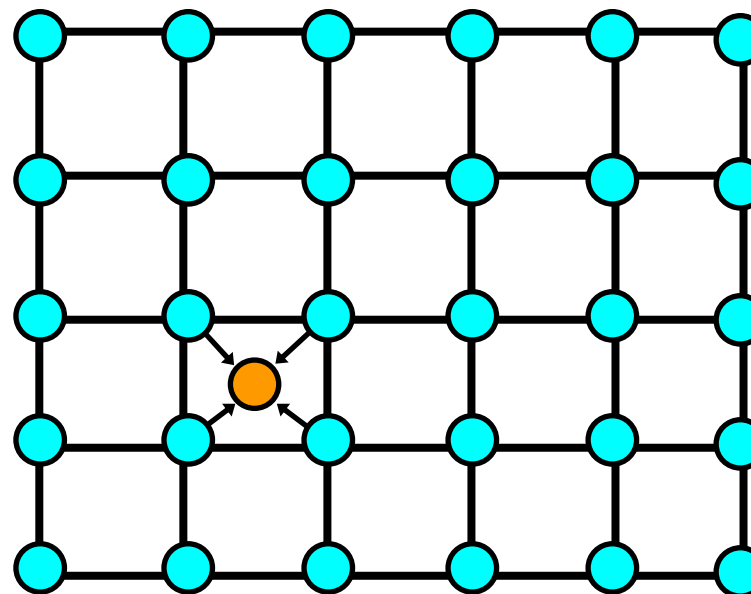
- Statická 3D data
 - $f: \mathbb{R}^3 \rightarrow \mathbb{R}^n$
 - snadno skaláry ($n=1$), resp. vektory $n \leq 3$
 - problém: v meteorologii $n \sim 30$
 - výhodné použití animací
- Dynamická 3D data (animace)
 - $f: \mathbb{R}^4 \rightarrow \mathbb{R}^n$ ($[x, y, z, t]$)
 - animace neřízená/řízená (*steering*)

Voxely nebo buňky?



VOXEL

naměřené hodnoty
jsou uprostřed



BUŇKA

naměřené hodnoty
jsou ve vrcholech

1. Pořízení (CAT, MRI)

- netriviální algoritmy
- vzdálenosti mezi řezy jiné než mezi buňkami

2. Vylepšení 2D řezů

- filtrace (šum, kontrast)
- vyrovnaní histogramu ve všech řezech

3. Vylepšení 3D

- převzorkování do uniformní mřížky

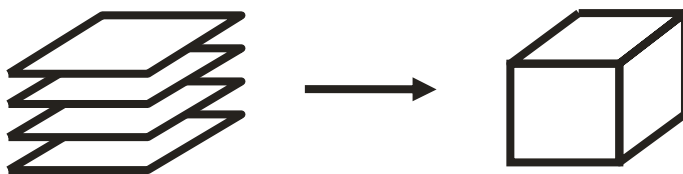
4. Klasifikace, segmentace

- druhy tkání (ručně/automaticky)

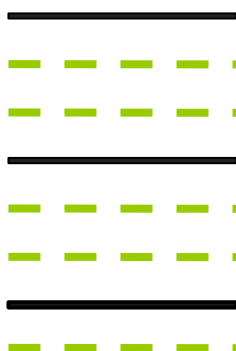
Příklad rekonstrukce

- Příklad rekonstrukce





řezy na sebe + interpolace
(regularizace)



původní
řezy

+

interpolované
řezy

Jaký skutečný tvar ?



Interpolace:



Duplikace (0.řádu)

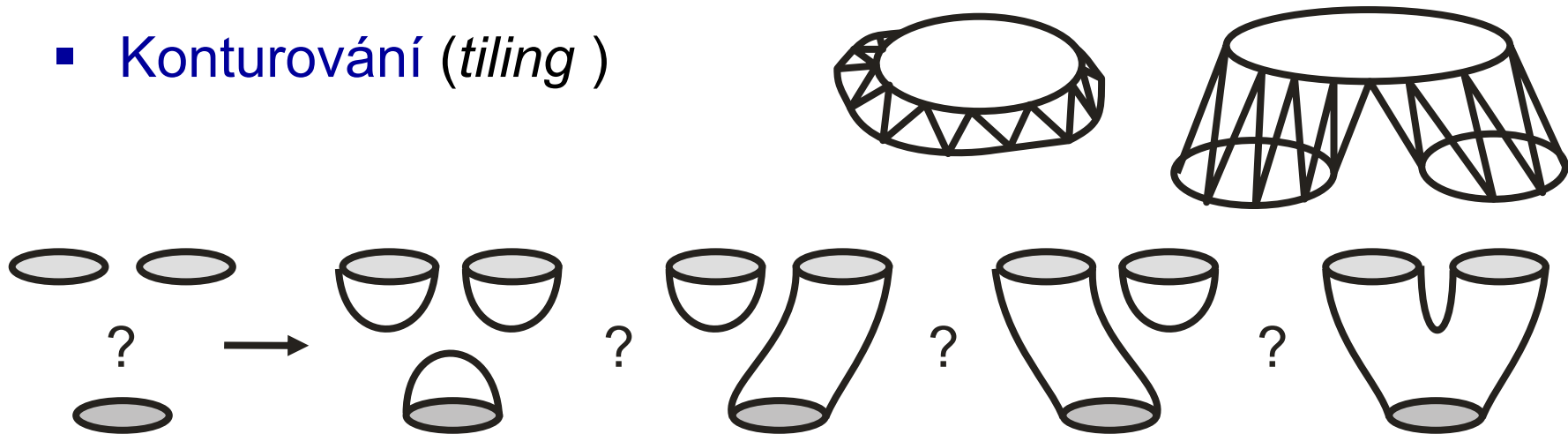


Lineární (1.řádu)

(vyšší řád - jen 1.5% rozdíl)

- **Povrchové** algoritmy
 - pomocná reprezentace (izoplochy) [POMALÉ]
surface fitting
 - klasický rendering [RYCHLÉ]
z-buffer
- **Objemové** algoritmy
 - přímé zobrazení povrchových i všech dat
[INTERAKTIVNÍ RYCHLOST]
direct volume rendering

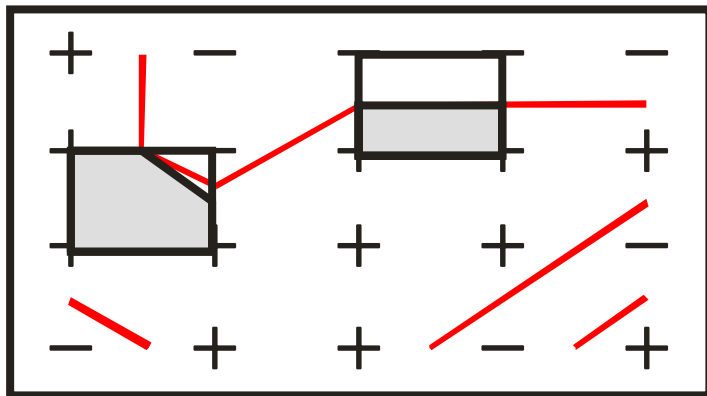
- Konturování (*tiling*)



- Pochodující kostky, čtyřstěny, ...
(*Marching cubes, tetrahedra*)
- Výsledkem vždy soubor trojúhelníků,
případně vrcholů s normálami

Pochodující kostky (princip ve 2D)

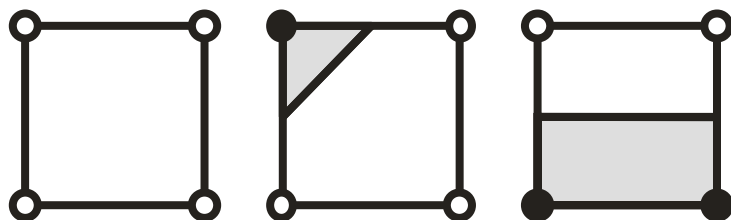
Mřížka skalárních hodnot (binární, šedotónové)



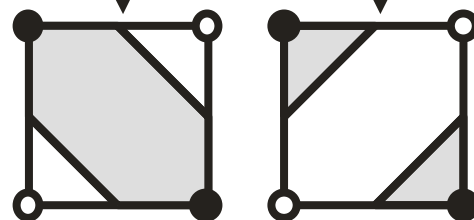
Izočára (izoplocha)

Objem vnitřní/vnější

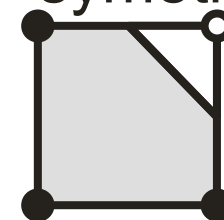
Základní konfigurace



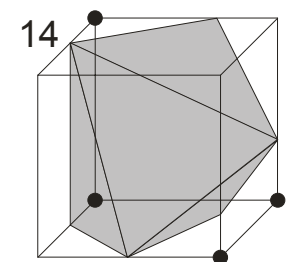
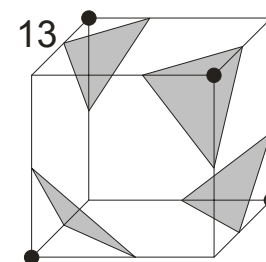
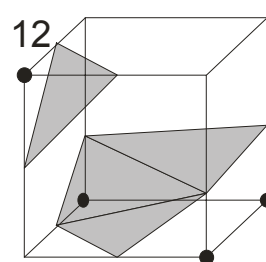
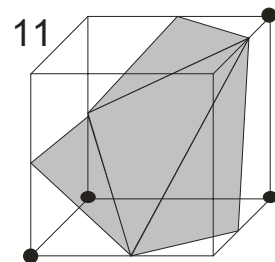
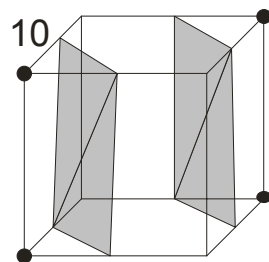
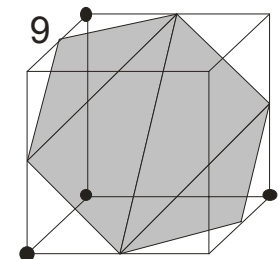
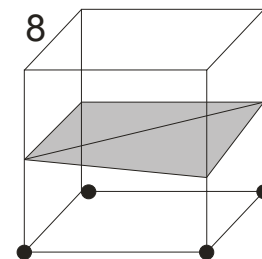
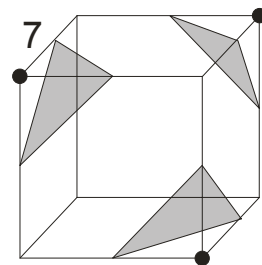
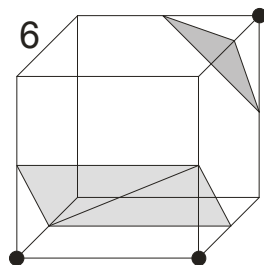
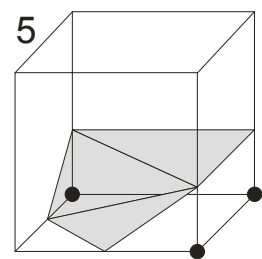
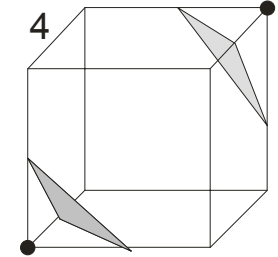
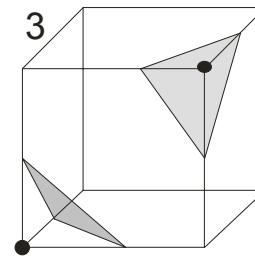
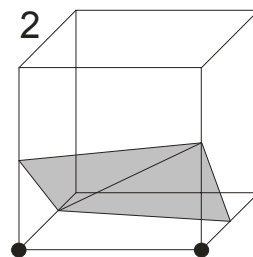
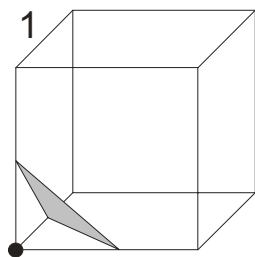
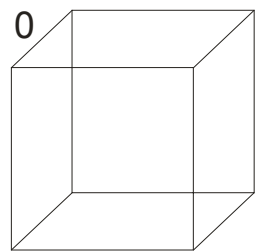
nejednoznačné



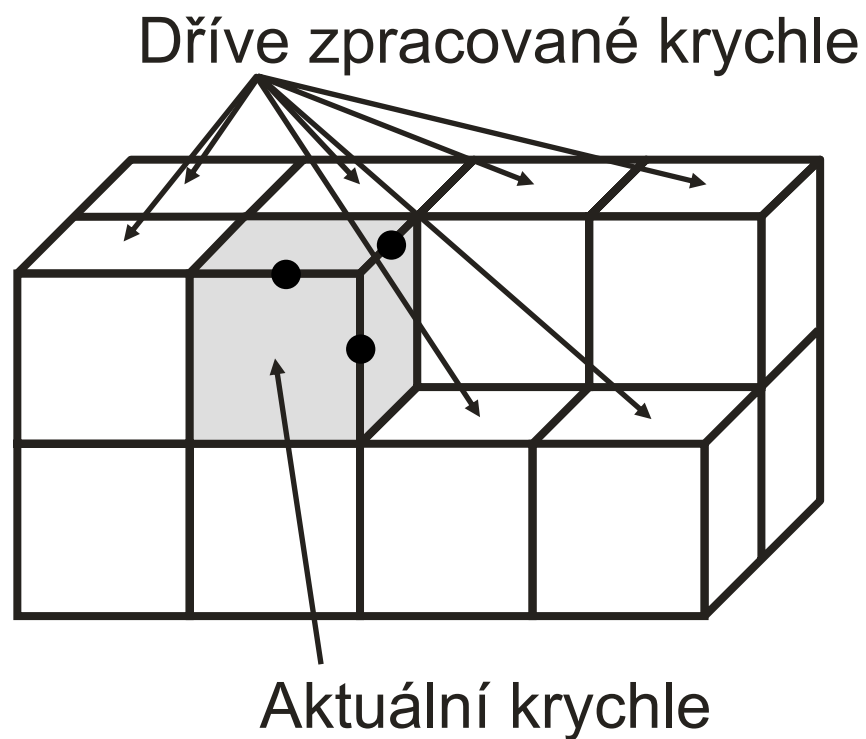
Ostatní symetrie

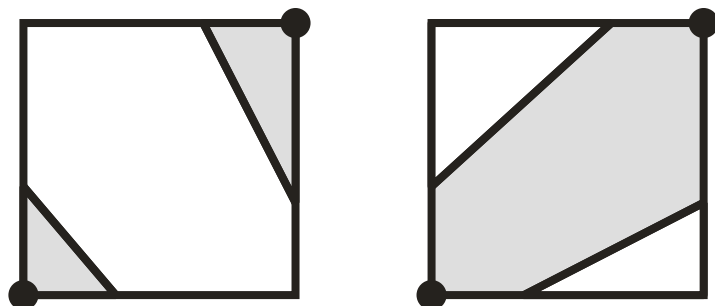


Případy ve 3D



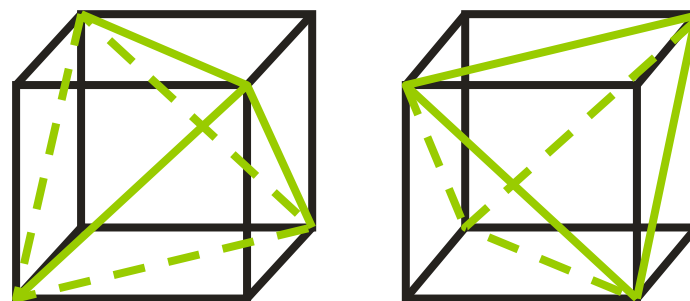
- Zpracování po řezech
- Interpolace polohy vrcholů
- Normály ve vrcholech



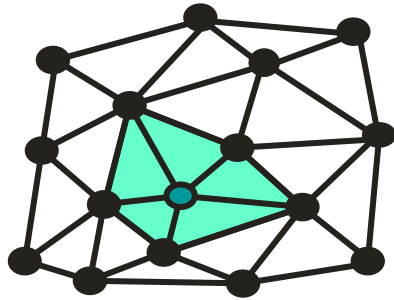


Vznik děr

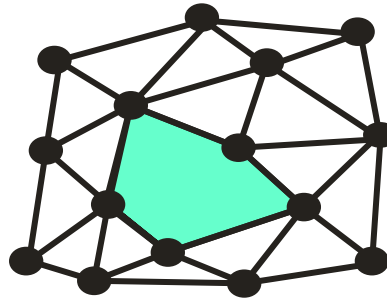
- **Kráčející čtyřstěny**
 - jednoznačné
 - 5 variant poloh
 - generují mnoho trojúhelníků



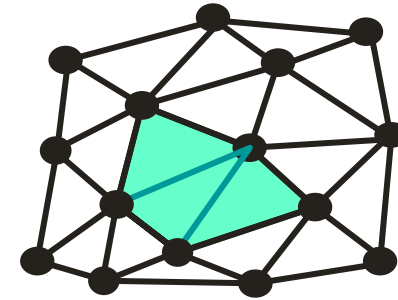
Decimace počtu trojúhelníků



test



výmaz



triangulace

Různé algoritmy

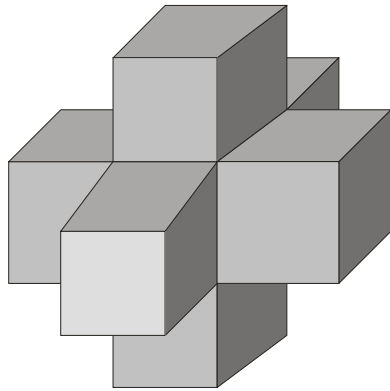
- kriteria zjednodušování
- geometrie, topologie, barvy, textury
- LOD

- Vrhání paprsku (*Ray casting*)
- Projekce (*Splatting, V-buffer*)
- Kombinace povrchové a objemové reprezentace (*Volumetric Ray-tracing*)

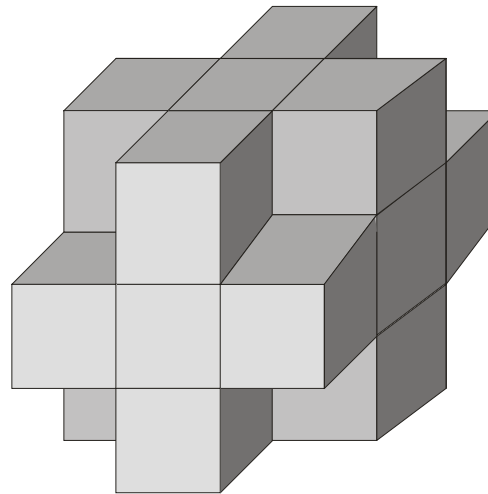
- + přímo, bez pomocné povrch. reprezentace
- + beztvaré materiály (mraky, kapaliny, plyny)
- + množinové operace
- + čas nezávisí na složitosti objektů
- prochází celý objem dat (čas, paměť, přístup)
=> postupné zjemňování (*prog. refinement*)
- nepřesné (zubaté objekty)
- výpočty závislé na pohledu

-
- Podobnost s rastrovou grafikou

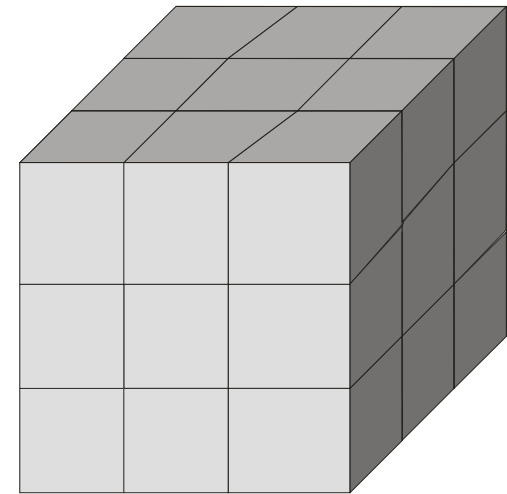
Okolí voxelu



6-okolí



18-okolí



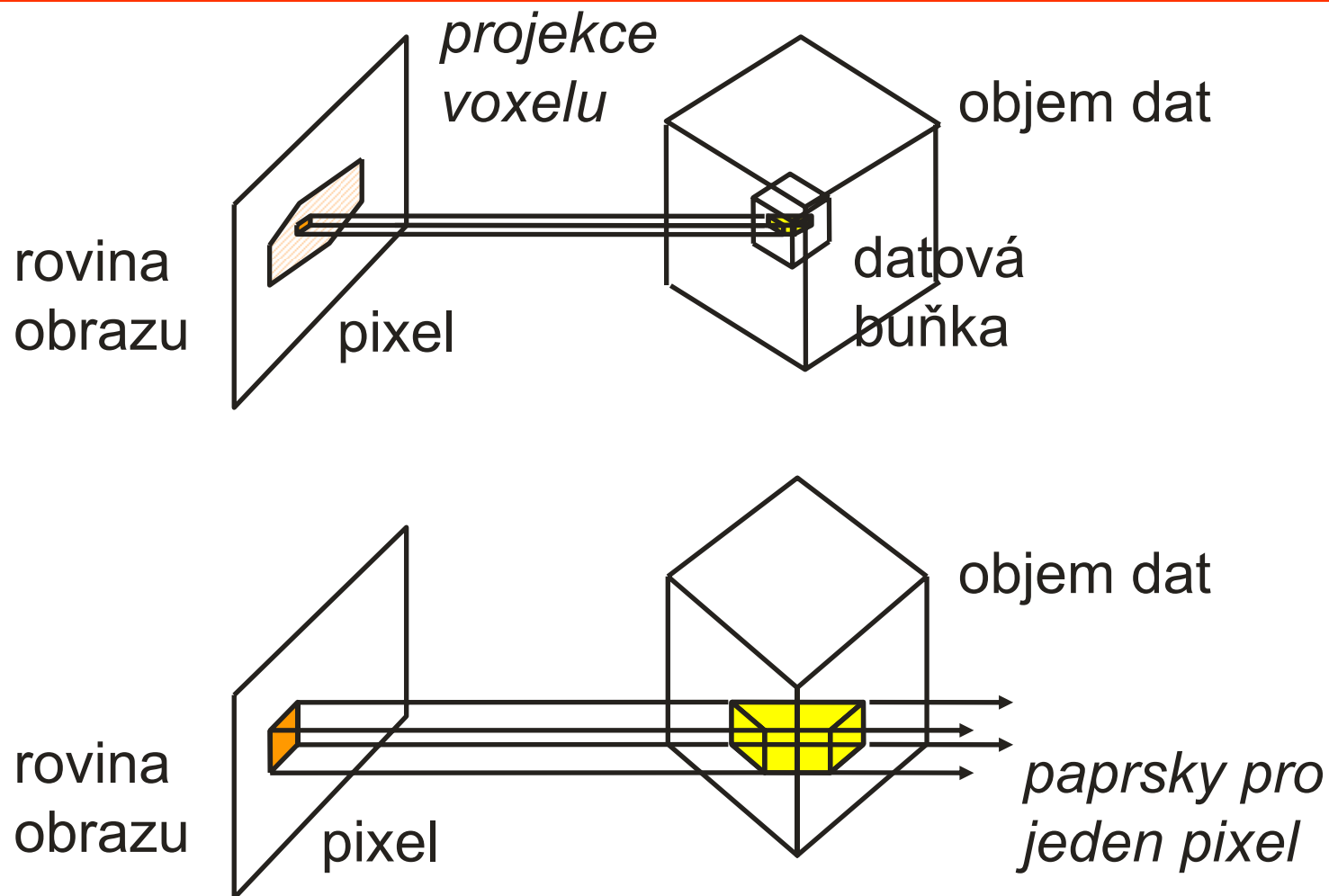
26-okolí

Různé typy paprsků (průchodů)

- V prostoru **obrazu** (*image order*)
procházíme **pixely**, vysíláme paprsky do objemu,
akumulujeme příspěvky

- V prostoru **objemu** (*object order*)
procházíme **voxely** (Front-Back nebo B-F), promítáme je
na obrazovku
 - jako celek (*splatting*)
 - pro každý pixel integrovat zvlášť (*V-buffer*)

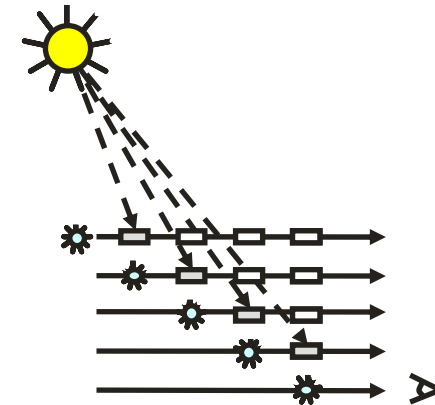
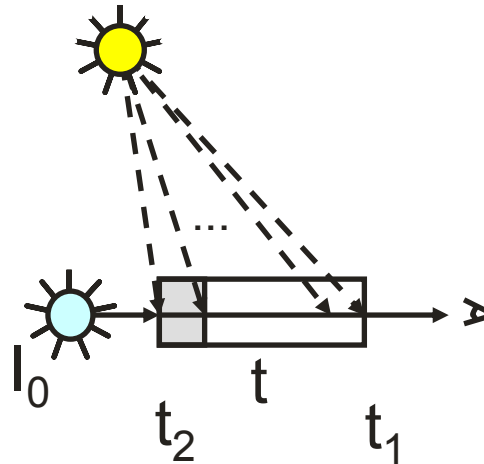
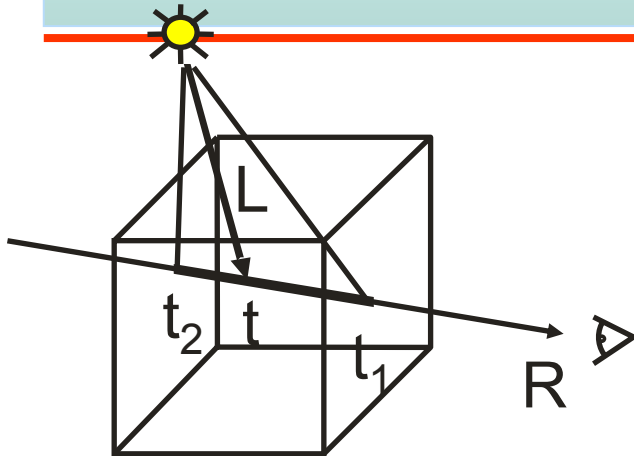
Prostor obrazu X prostor objemu



Voxely:

- Vzájemně si nezakrývají **okolní** světlo
- Navzájem světlo neodrážejí **zrcadlově**
- **Rozptylují** dopadající a **tlumí** procházející světlo (nebo dokonce **vyzařují** vlastní)

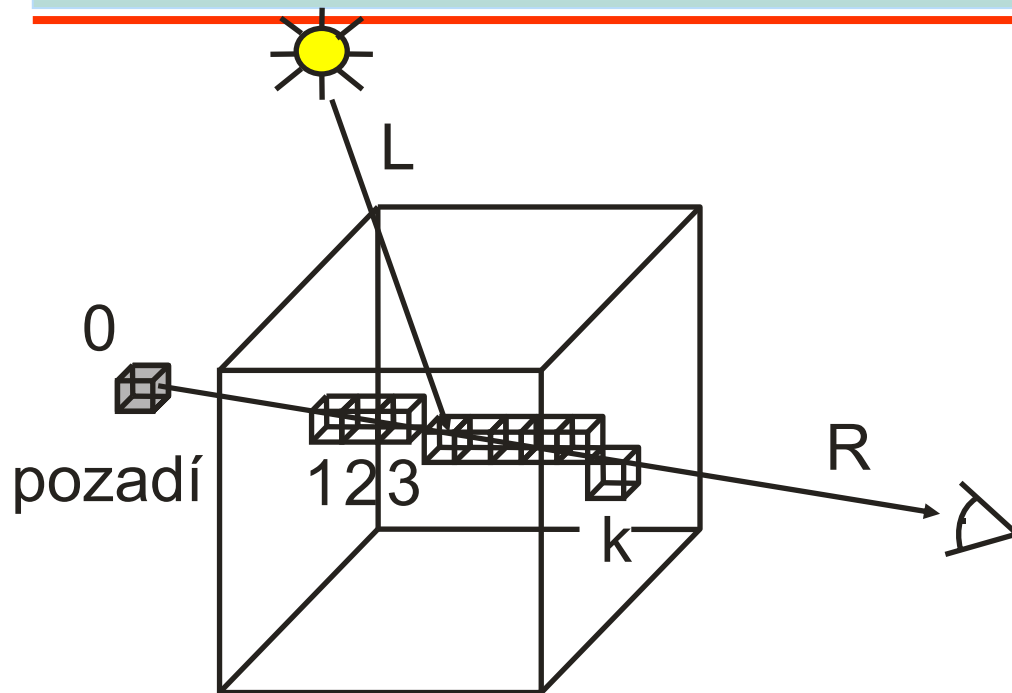
Osvětlovací model (spojitý)



$$\int_{t_1}^{t_2} \left(e^{-\tau \int_{t_1}^t D(s) ds} \right) I(t) D(t) P(\cos \theta) dt$$

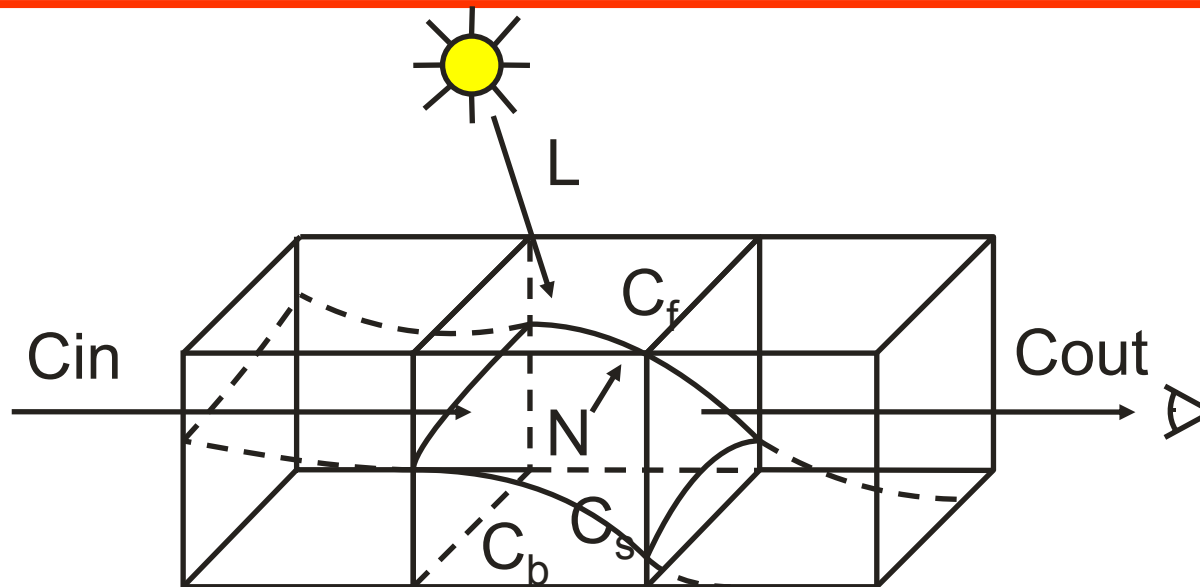
hustota **D**, útlum τ , odrazová funkce **P**, úhel mezi paprskem a světlem θ

Diskrétní osvětlovací model



$$C_R = \sum_{i=0}^K C(i) \alpha_i \prod_{j=i+1}^K (1 - \alpha_j)$$

Využití izoploch



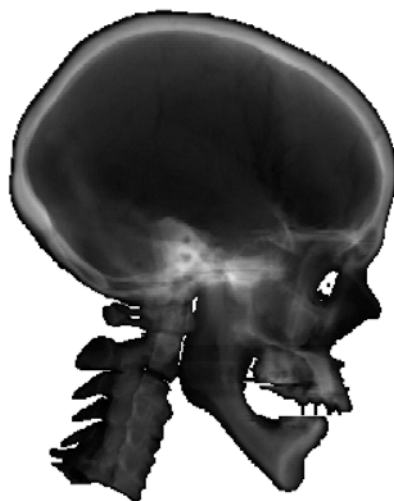
3 dále zpracovávané oblasti dané izoplochou:

- ♦ front
- ♦ surface
- ♦ back

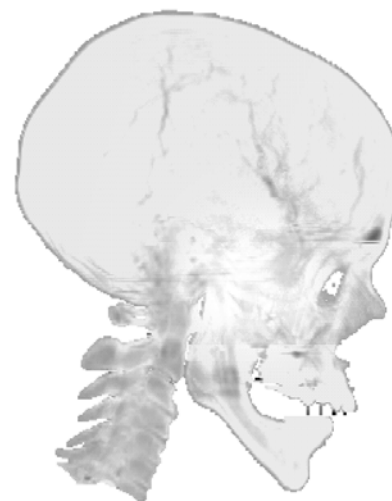
- Zobrazení **celého objemu**
 - zobrazení maxima, součtu, průměru jasu
- Vyhledání **povrchu**
 - konstantní barva dle objektu (*flat shading*)
 - jas dle vzdálenosti
- Vyhledání **povrchu** a odhad **normály**
 - odhad gradientu symetrickou diferencí



Max hodnota
podél paprsku



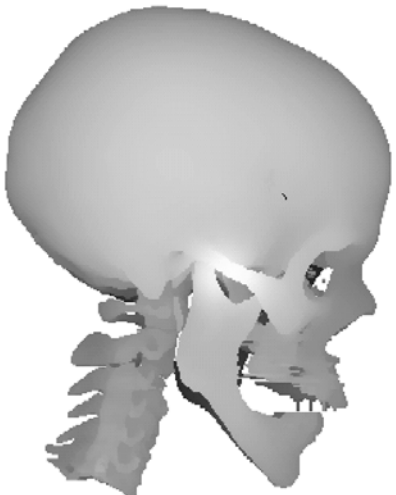
Součet hodnot
podél paprsku



Prům. hodnota
podél paprsku



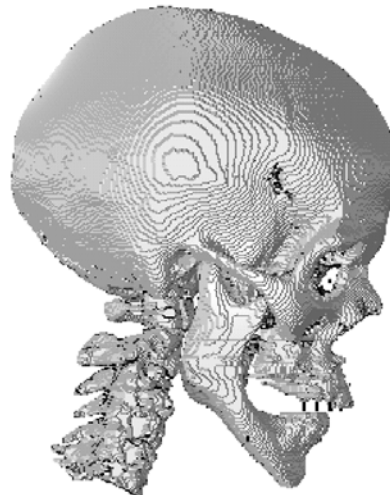
Hodnoty na
povrchu



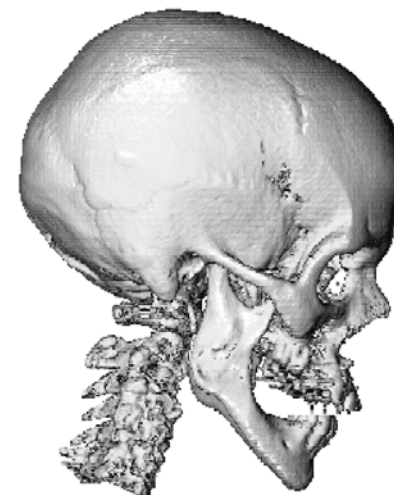
Vzdálenost
povrchu



Normály v pa-
měti hloubky

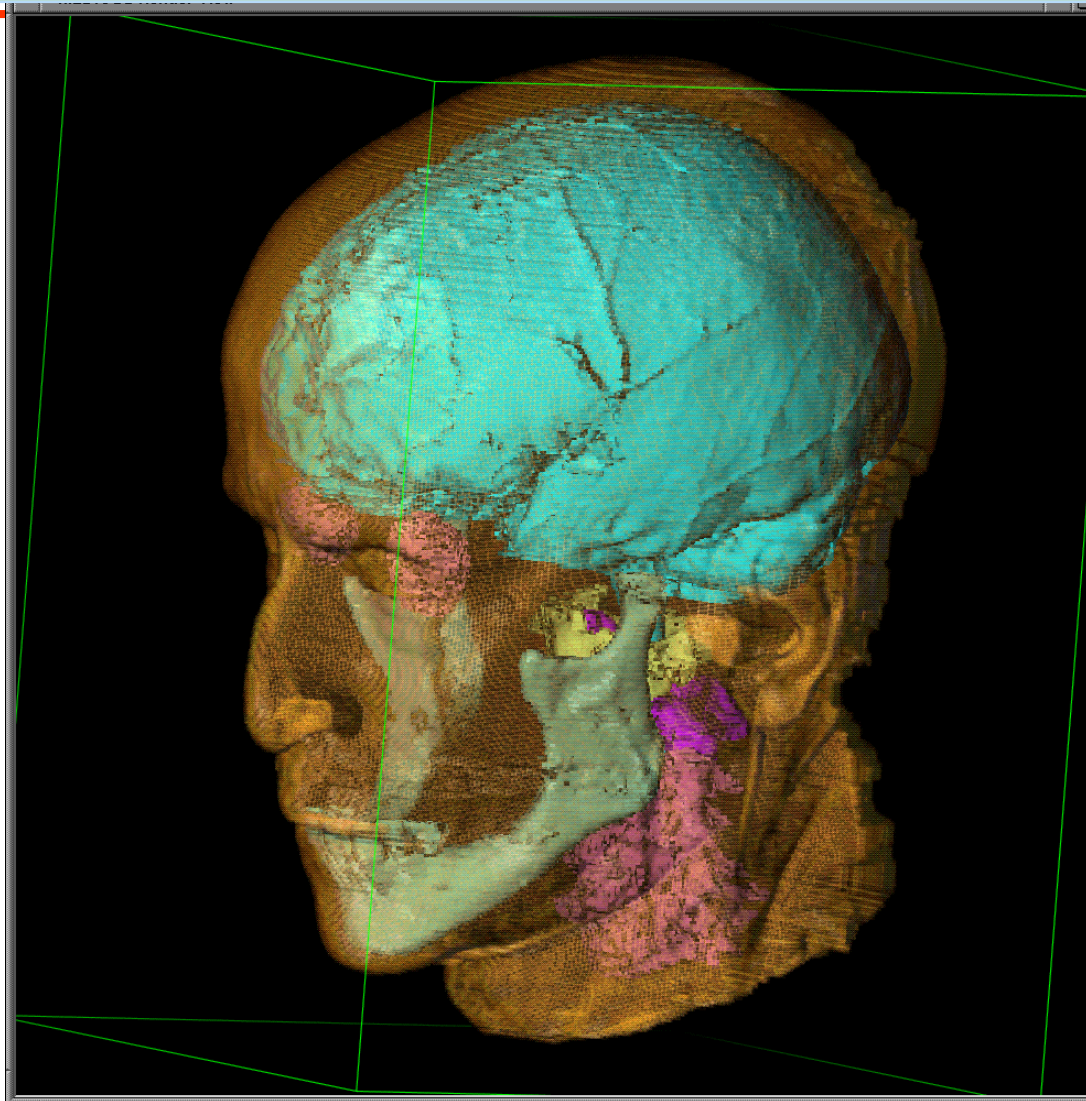


Normály v bi-
nárním objemu



Normály v šedo-
tónovém objemu

Segmentace a prezentace



[MPG] Žára, Beneš, Sochor, Felkel. *Moderní počítačová grafika*,
Computer Press, 2004
kap. 10 Světlo, str. 319-336 a
kap. 15 Globální zobrazovací metody, STR 413-455