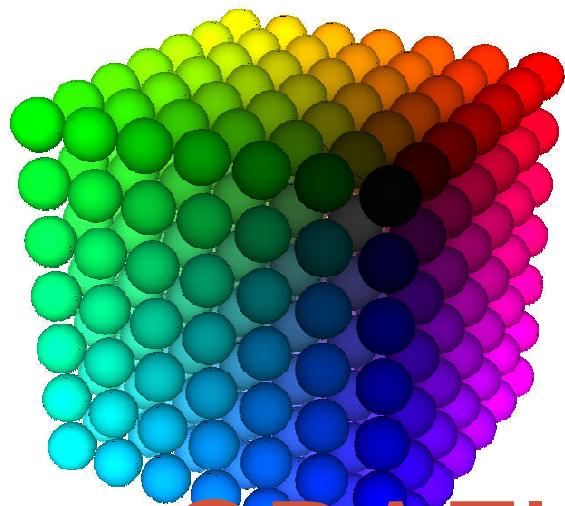
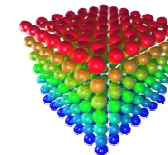


© 2014

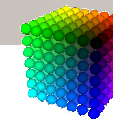


VIZUALIZÁCIA a GRAFICKÉ ARCHITEKTÚRY

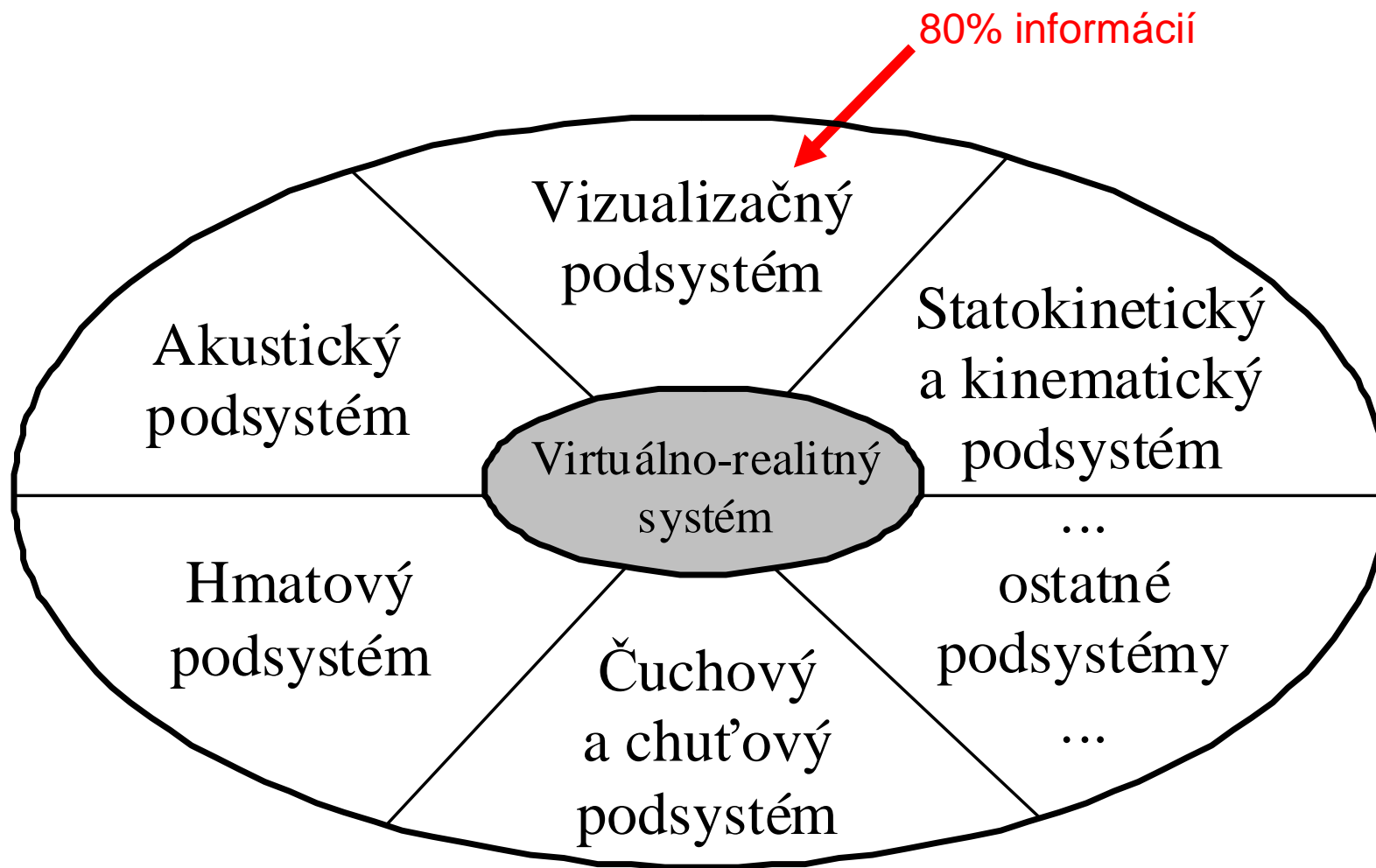
doc. Ing. Branislav Sobota, PhD.

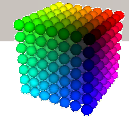
Katedra počítačov a informatiky

FEI TU Košice



Podsystemy VR systému



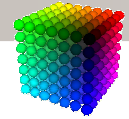


Medialita

Medialita vo forme **sprostredkovania**

Sprostredkovanie z hľadiska tejto technológie je rozšírený pojem zahrňujúci aj isté prvky prenosu zviditeľnenia (vizualizácie) do iného formátu t.j. transformáciu objektov do „mediálnej“ podoby.

Mediácia - proces transformácie údajov v rámci vytvárania resp. presunu objektov zahrňujúci aj množinu transformácií umožňujúcich transport údajov na zviditeľnenie (vizualizáciu)

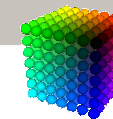


Videnie (vision)

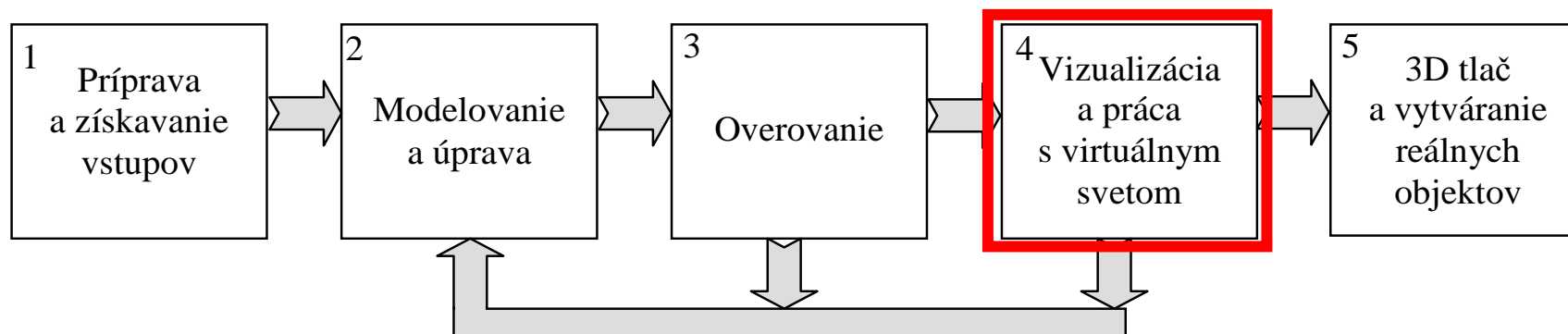
je komplementárny problém, ktorý analyzuje obrazy na vytvorenie počítačových modelov sveta

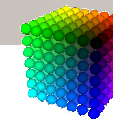
Vykresľovanie (rendering)

je proces tvorby obrazu pomocou počítačových algoritmov alebo tiež oblasť štúdia zaoberajúceho sa syntézou obrazov z modelov sveta v počítači.

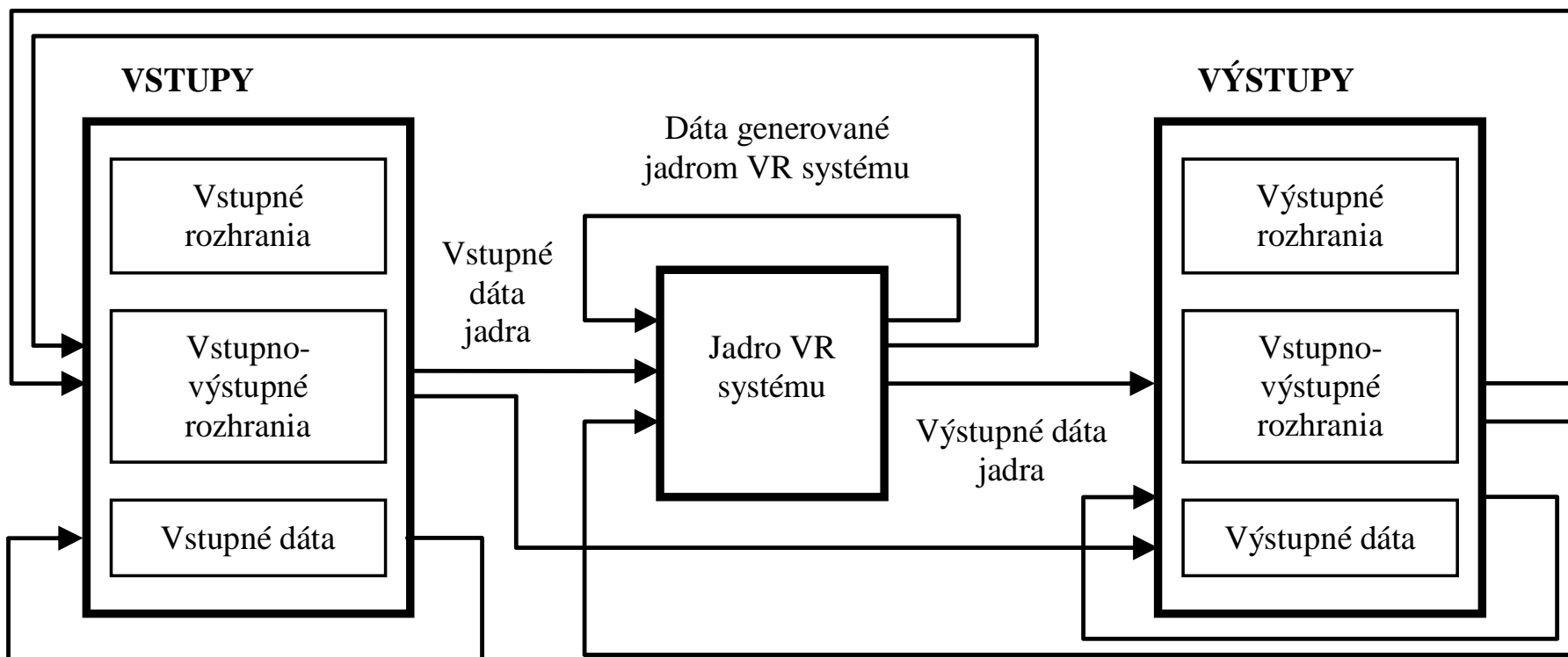
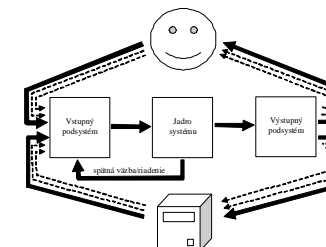


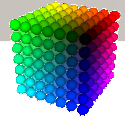
Vizualizácia a virtualizačný reťazec





Vizualizácia a VR systém



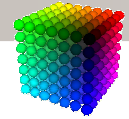


Vizualizácia

je proces transformácie popisu modelu virtuálneho sveta do výstupného obrazu na zobrazovacom zariadení



Základnou úlohou grafických architektúr je generovať grafický výstup pre pripojené zobrazovacie zariadenie

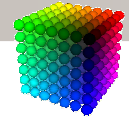


Paralelný rendering

využitie viac procesných jednotiek pre urýchlenie výpočtu scény

dôvody:

- komplexnosť scény
- kvalita výstupného obrazu
- interakcia resp. odozva v reálnom čase

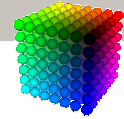


Kvalifikácia grafických výpočtových systémov

podľa

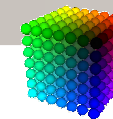
- reprezentácie, použitej na popis prostredia
- algoritmu, použitého na vytvorenie obrazov

veľa algoritmov môže byť použitých pre danú reprezentáciu a mnoho reprezentácií sa môže použiť pre jeden algoritmus

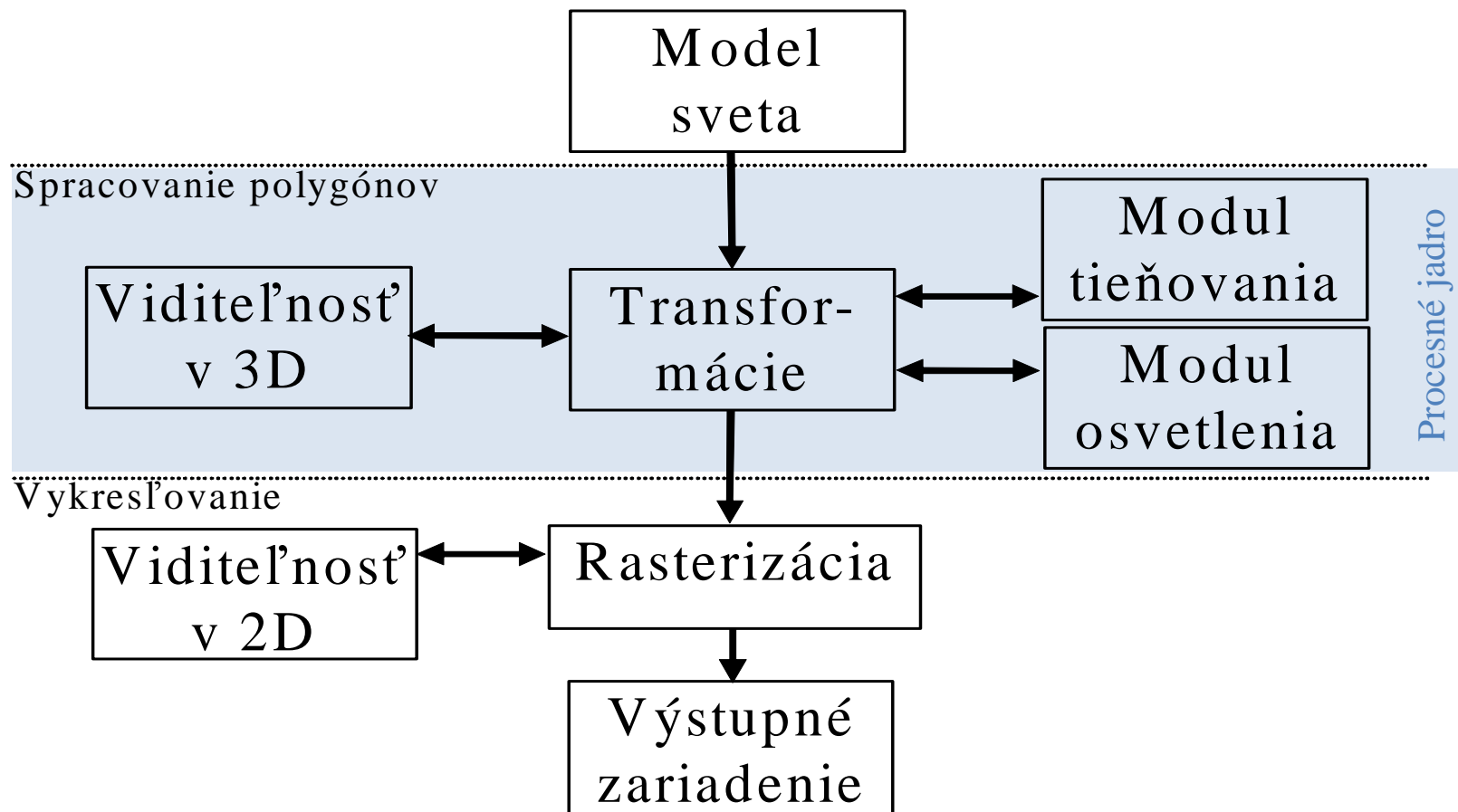


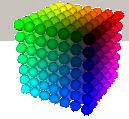
Vrstvy vizualizačného procesu

1. Definovanie/spracovanie modelu (reprezentácia, súradnicové systémy)
2. Transformácie nad objektami (geometrické)
3. Riešenie viditeľnosti
4. Tieňovanie
5. Osvetľovanie
6. Realistické zobrazovanie
7. Vykresľovanie



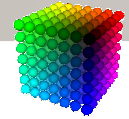
Vizualizačný podsystem





Paralelizmus v grafických architektúrach

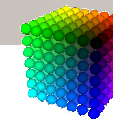
- Funkcionálny paralelizmus
- Údajový paralelizmus
- Časový paralelizmus



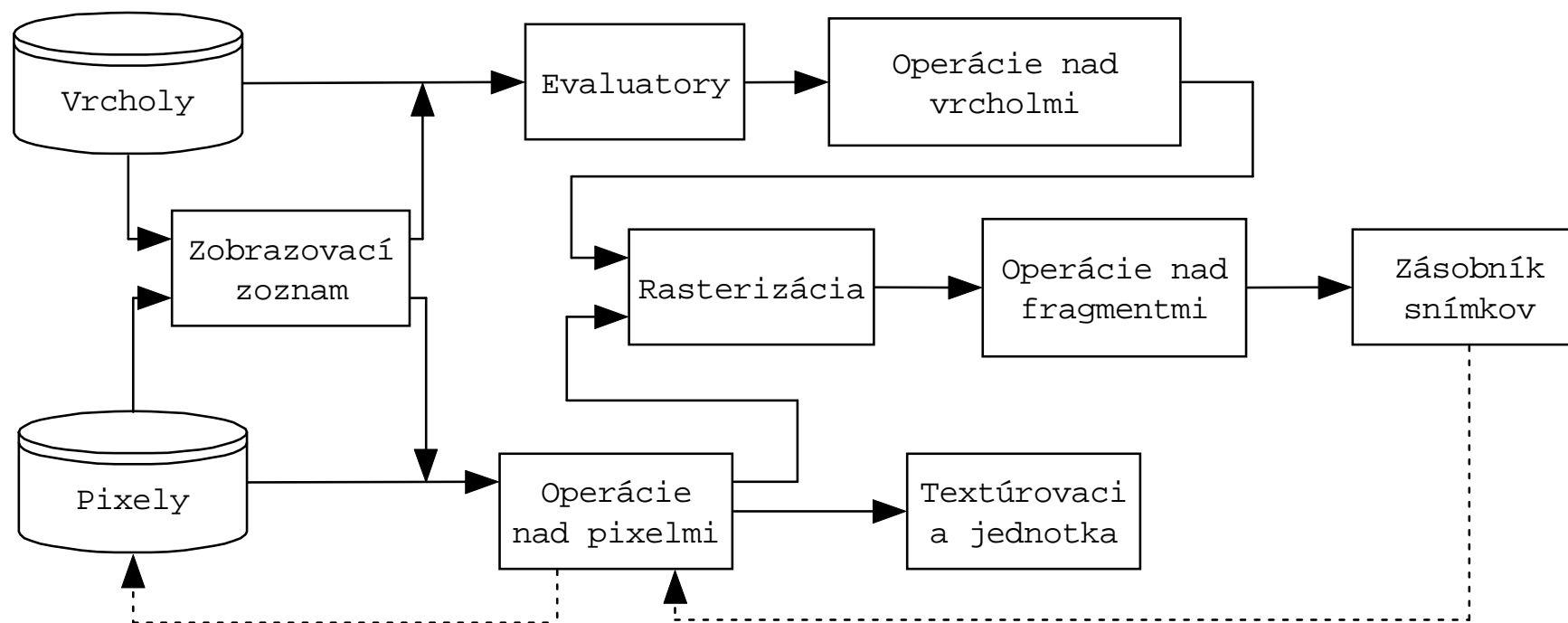
Funkcionálny paralelizmus

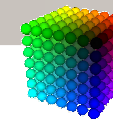
rozdelenie procesu výpočtu do niekoľkých jednoznačných funkcií, ktoré môžu byť aplikované postupne na jednotlivé údaje.

Ak jedna procesná jednotka odpovedá jednej funkcii (alebo skupine funkcií) a tok údajov je smerovaný z jednej jednotky do nasledujúcej, získame **prúdový** (zreťazený) výpočet

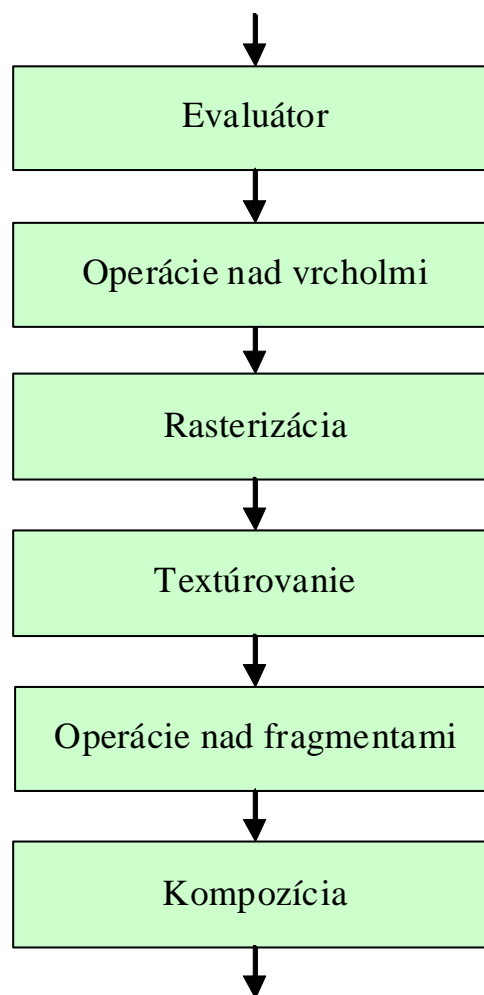


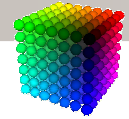
Výpočtová vizualizačná rúra





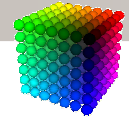
Výpočtová vizualizačná rúra (zjednodušená schéma)





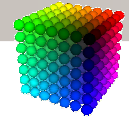
Použitie funkcionálneho paralelizmu

Vhodný pre spracovanie polygónov a plôch v aplikáciách, kde 3D geometrické primitíva sú vkladané na vstupe prúdu a hodnoty pixelov výsledného obrazu sú produkované na konci prúdu



Obmedzenia funkcionálneho paralelizmu

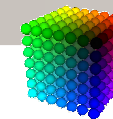
- Celková rýchlosť zreťazených procesných jednotiek je limitovaná najpomalšou z nich, preto funkčné jednotky musia byť navrhované s ohľadom na priepustnosť celého systému
- Limitovaný počet funkčných jednotiek v zreťazenom systéme



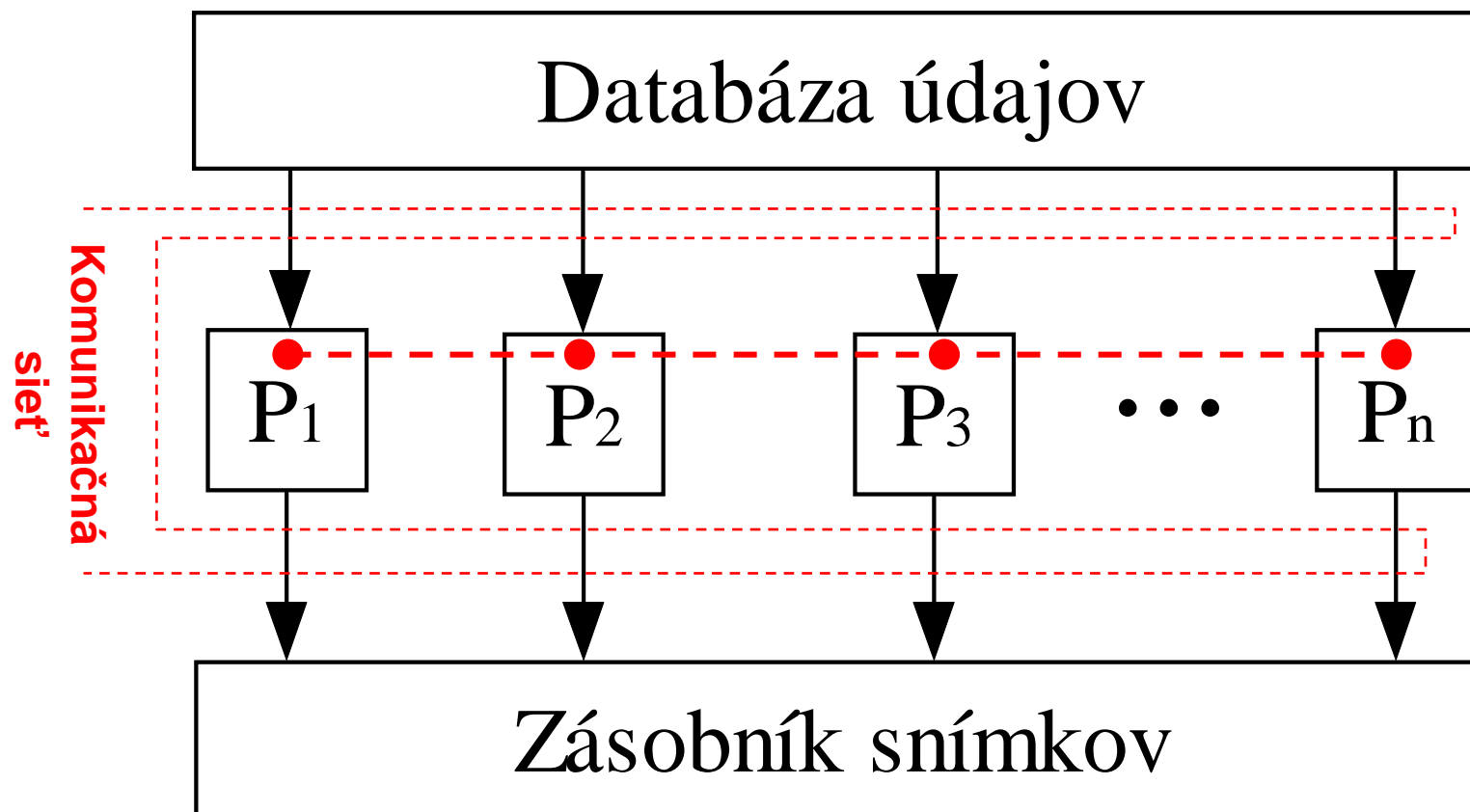
Údajový paralelizmus

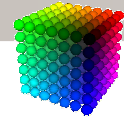
Namiesto výkonnej postupnosti zreťazených funkčných jednotiek na jednom údajovom prúde (*single data stream*), je možné rozdeliť údaje do viacerých prúdov a tie prideliť do rovnocenných procesných jednotiek.

Výsledky sa potom spájajú v zásobníku snímok (*framebuffer*)



Údajový paralelizmus





Triedy údajového paralelizmu

- Objektový paralelizmus - spracovanie objektov scény

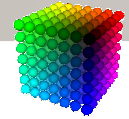
- geometrická transformácia,
- vyradovanie odvrátených primitív,
- výpočty svetelných zdrojov,
- pohľadová transformácia,
- orezanie

Transformačná
fáza

- Obrazový paralelizmus - tvorba obrazu

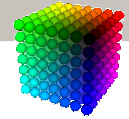
- výpočet osvetlenia,
- interpolácia,
- kompozícia,
- riešenie viditeľnosti

Rasterizačná
fáza



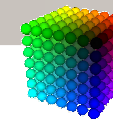
Závislosť údajového paralelizmu

- od zložitosti vstupnej scény,
- od stupňa vzorkovania,
- od rozlíšenia výstupného obrazu

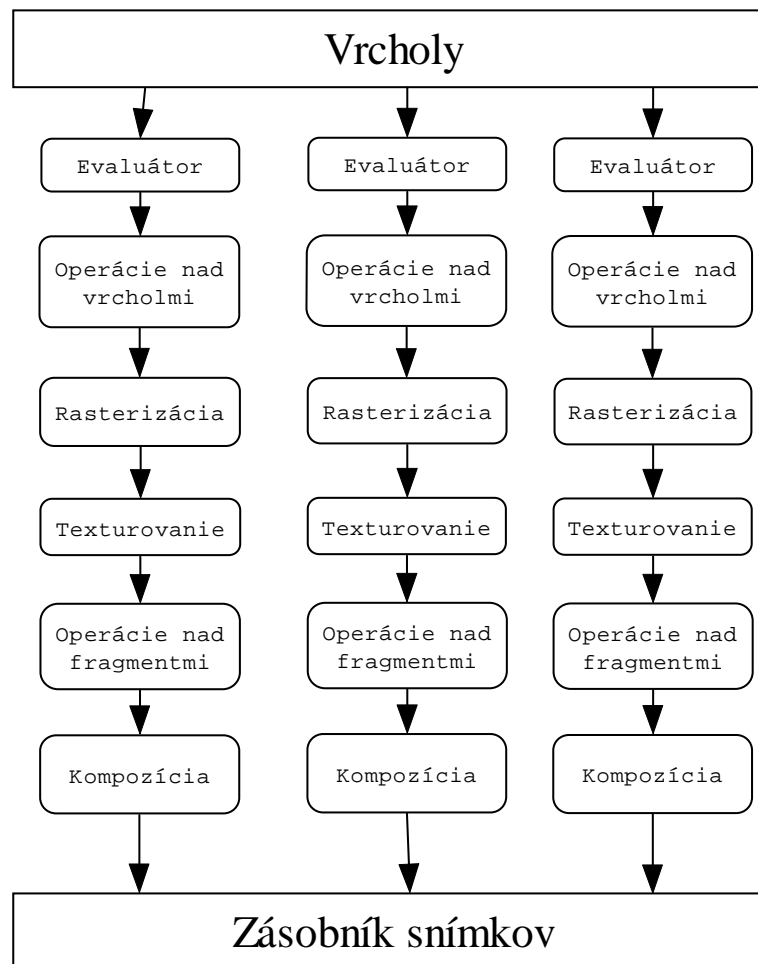


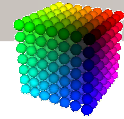
Časový paralelizmus

V animovaných aplikáciách , kde stovky alebo tisíce vysoko kvalitných obrázkov (*frames*) musia byť vypočítané pre opakované prehrávanie, čas potrebný na výpočet jednotlivých snímkov nie je taký dôležitý, ako **celkový čas potrebný na výpočet všetkých snímkov** celej animácie. V tom prípade **paralelizmus** môže byť získaný rozdelením problému **v časovej doméne**. Každý procesor si vezme na starosť určitý počet snímkov, ktoré vypočíta pomocou údajov potrebných pre tieto snímky



Hybridný prístup

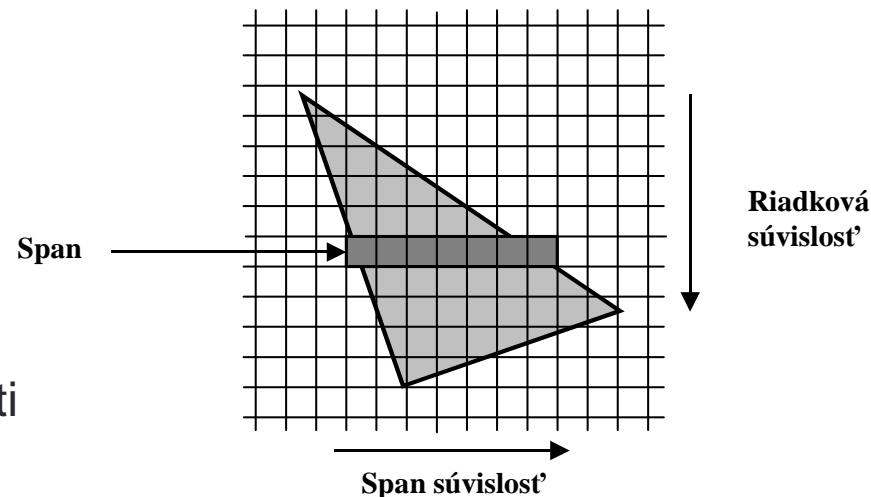


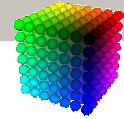


Súvislosti

- snímková súvislosť (podobnosť pixelov medzi snímkami),
- riadková súvislosť (podobnosť pixelov vertikálne medzi riadkami),
- súvislosť span (súvislosť na intervale, podobnosť pixelov v rámci riadku),
- lúčova súvislosť (v algoritmoch sledovania lúča, okolie priesečníkov lúčov)

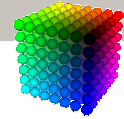
Geometrické súvislosti





Metriky pre hodnotenie výkonu grafických systémov

- Vstupná rýchlosť
- Počet spracovaných trojuholníkov (polygónov)
- Počet spracovaných pixelov
- Textúry (rozmer, typ, počet)
- Displeje (rozlíšenie, typ, počet)

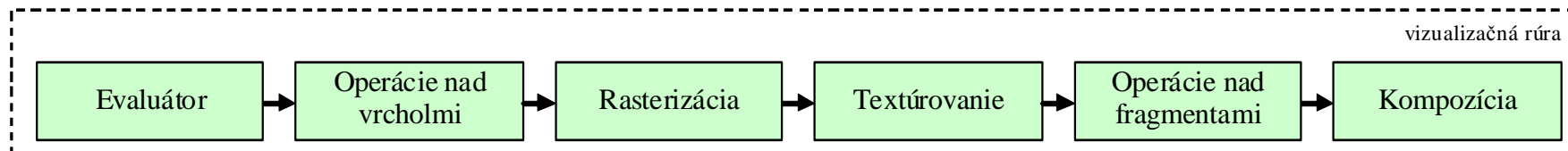


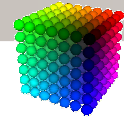
Klasifikácia grafických architektúr

podľa miesta, kde nastáva triedenie.

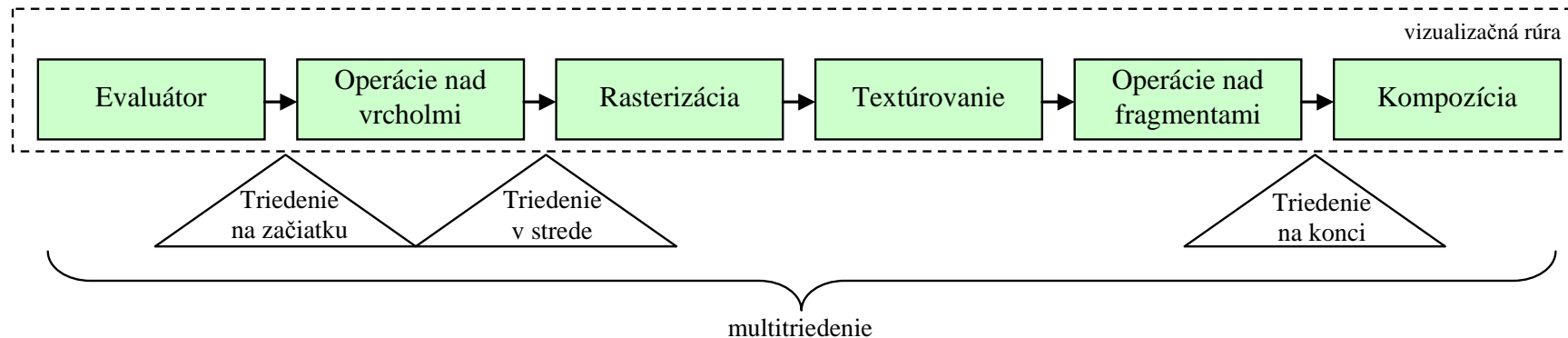
Základným predpokladom je, že architektúra sa skladá z viacerých grafických uzlov, kde každý uzol predstavuje samostatnú grafickú rúru pozostávajúcu z časti:

- Evaluátory - jednotka spracovania vstupných príkazov (skript)
- Operácie nad vrcholmi - geometrická jednotka pracujúca s vrcholmi
- Rasterizácia - rasterizačná jednotka
- Textúrovanie - textúrovacia jednotka
- Operácie nad fragmentmi - operácie na úrovni pixelov
- Kompozícia - kompozícia výstupnej snímky

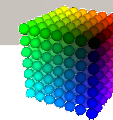




Klasifikácia grafických architektúr

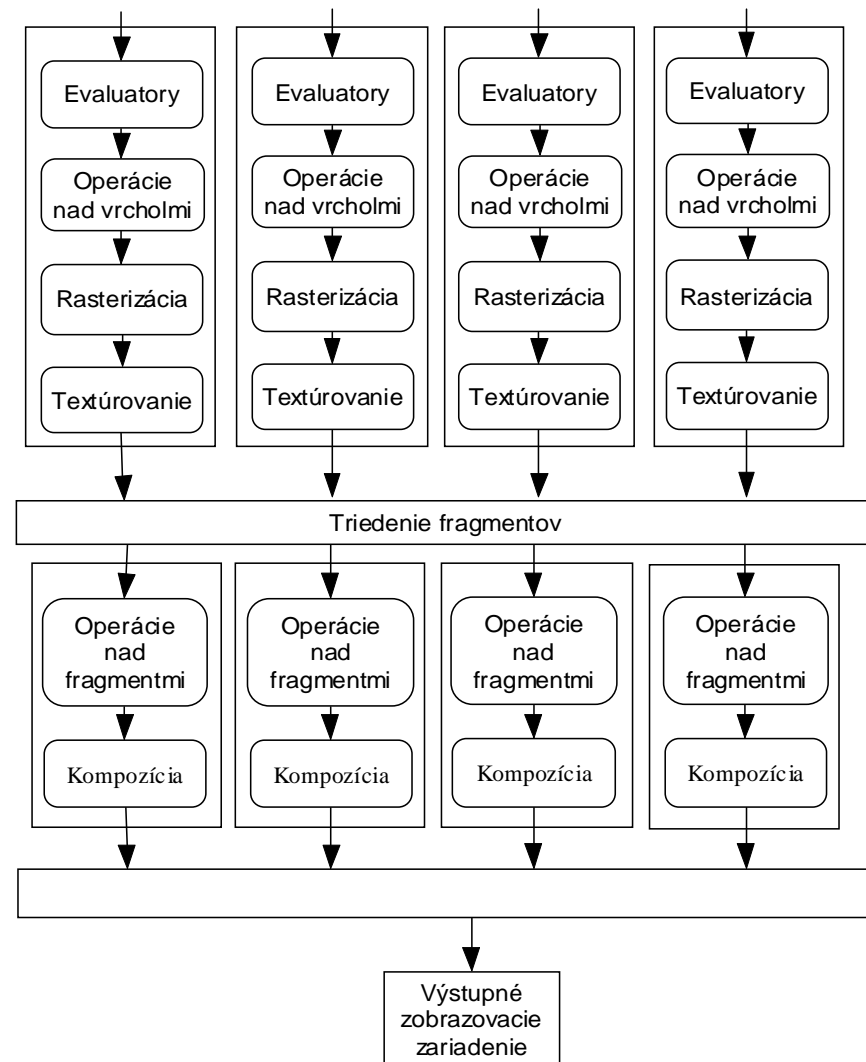


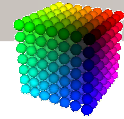
- **Architektúra s triedením na začiatku** - triedi 3D primitíva,
- **Architektúra s triedením v strede** - triedi 2D primitíva,
- **Architektúra s triedením na konci** – triedi fragmenty,
- **Architektúra s kompozíciou obrazu** - rozdeľuje vypočítané pixely až v procese zobrazovania,
- **Architektúra s viacnásobným triedením** - umožňuje komunikáciu každý s každým na všetkých úrovniach grafickej rúry s prihliadnutím na paralelnú efektívnosť



Architektúra s triedením na konci

každý procesor obsahuje takmer kompletnú renderovaciu rúru a vytvára časti obrazu po transformácii a rasterizácii príslušných častí primitív. Tieto časti obrazu sú spájané dohromady najčastejšie podľa hĺbkového triedenia každého bodu do konečného obrazu v zásobníku snímkov

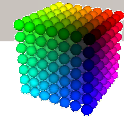




Architektúra s triedením na konci

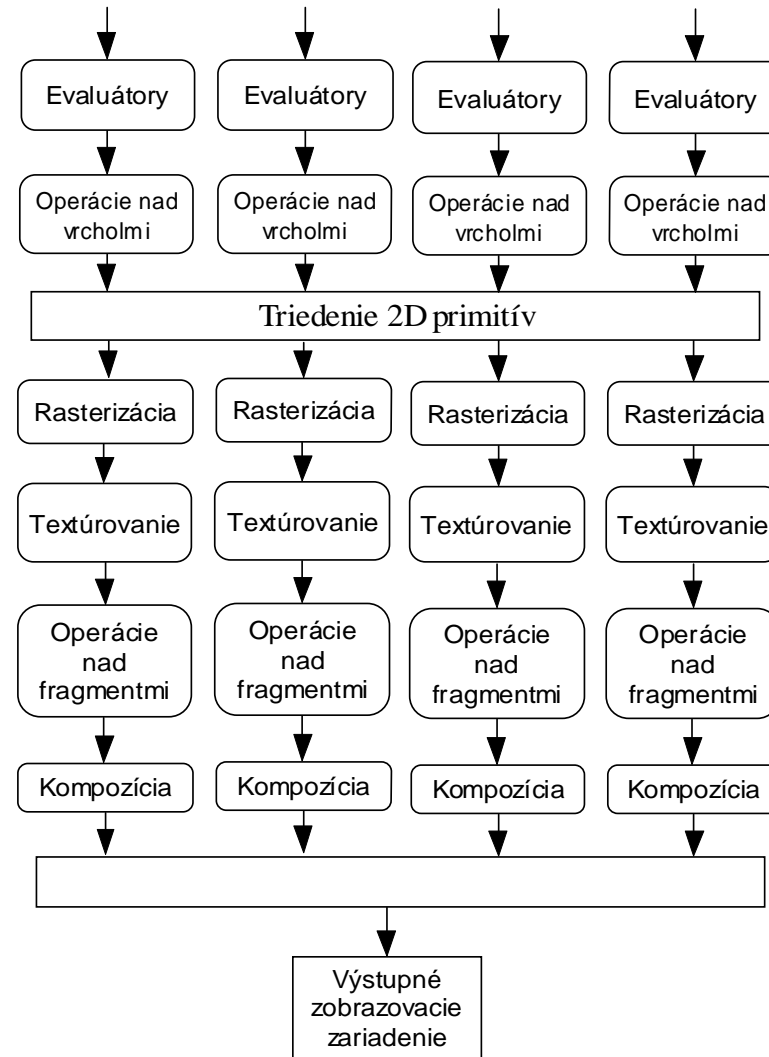
výhody/nevýhody

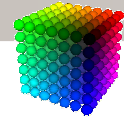
- výborná škálovateľnosť z pohľadu počtu primitív, ktoré môžu byť škálovateľné interaktívne. Stačí pridať ďalšie procesory do systému pre spracovanie väčšieho počtu primitív
- Na výstupnej strane zret'azeného spracovania je opačný problém. Tu sa nedá dobre škálovať s pohľadu počtu renderovaných pixelov. Každý vypočítaný pixel na každom procesore musí byť poslaný po spojovacej sieti a pre každý pixel sa musí posielat' dôležité **množstvo informácií** (farba, hĺbka)
- je absolútne nevhodná pre výstupy s vysokým rozlíšením



Architektúra s triedením v strede

každý uzol zodpovedá za časť zásobníka snímkov. Keď príkazy vstúpia do rúry sú konvertované na 3D primitíva, ktoré sú osvetlené a transformované do 2D zobrazovacej roviny priemetne. Každý z týchto primitív je potom triedený do príslušného rasterizačného uzla na základe rozdelenia priemetne. Potom nasleduje textúrovanie a kompozícia. Nakoniec zobrazovací systém spojí rozdelený zásobník snímkov do jednotného výstupného obrazu





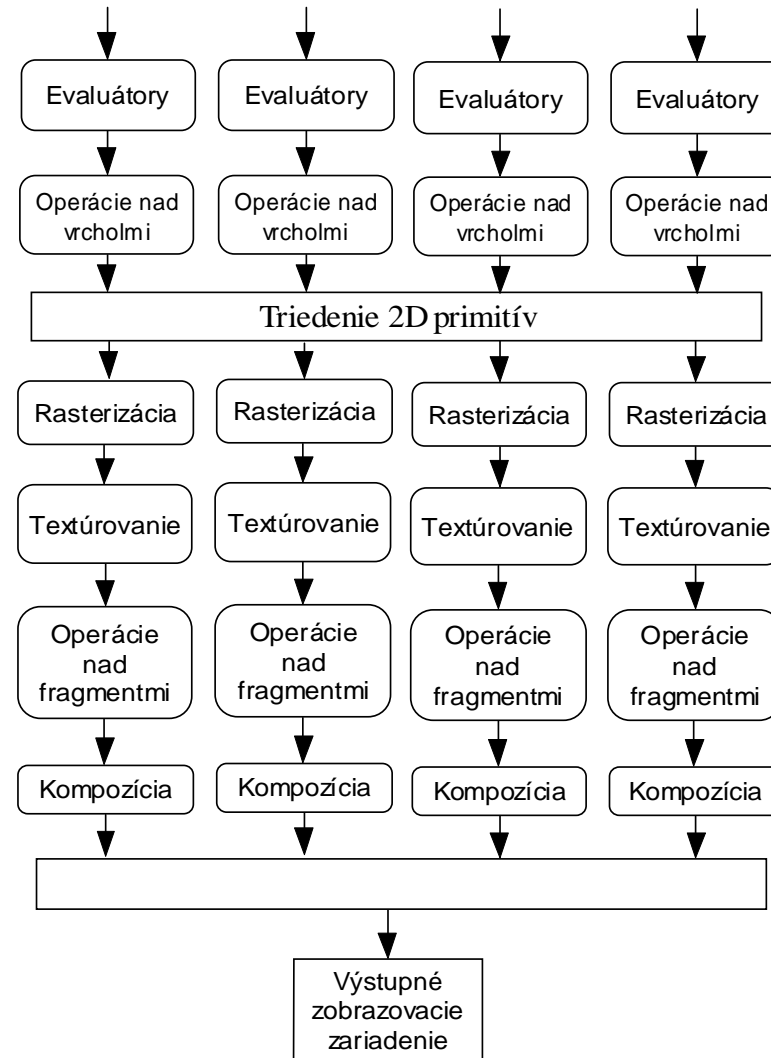
Architektúra s triedením v strede

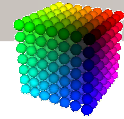
- **sada transformačných procesorov**

Na vstupe týchto procesorov využívajú rozdelenie obrazovky na ohraničené regióny pri triedení primitív. Každý procesor kompletne transformuje svoju časť pridelených primitív. Vystupujúce informácie o primitíve sú klasifikované podľa príslušného rozdelenia obrazovky do regiónov a posielané príslušným rasterizačným procesorom

- **sada rasterizačných procesorov**

Pridelené primitíva sú transformované do fragmentov, otextúrované a výsledne pixely sú odoslané do zásobníka snímokov.

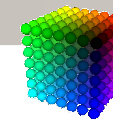




Architektúra s triedením v strede

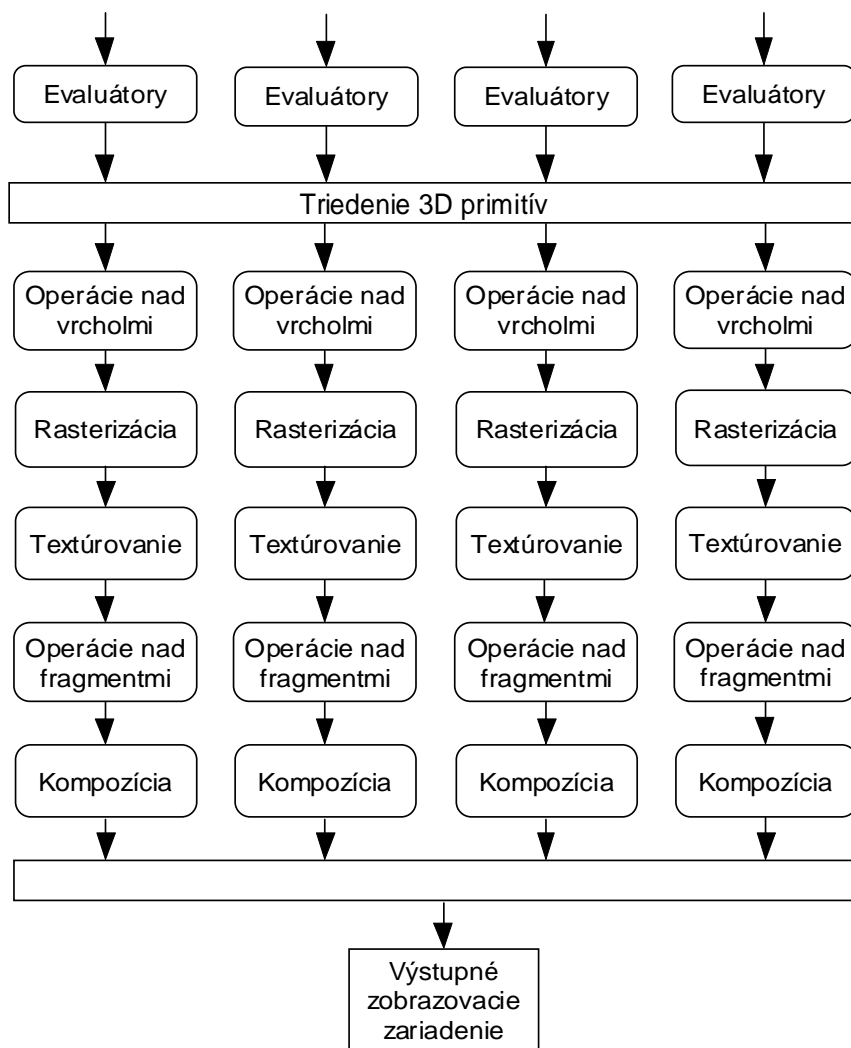
výhody/nevýhody

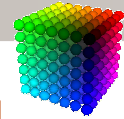
- posiela po sieti iba finálne pixely po rasterizácii do zásobníka snímkov. V porovnaní s predchádzajúcou tabuľkou je priepustnosť vo všetkých prípadoch 32-krát menšia. To znamená, že otázka priepustnosti siete nie je problémom pre architektúru s triedením v strede a je vhodným kandidátom pre aplikácie s veľkým výstupným rozlíšením
- každé zobrazené primitívum musí byť posielané z daného transformačného procesora do daného rasterizačného procesora. Je potrebná komunikačná schéma každého s každým, čo prináša obmedzenia na počet procesorov (zahltie).
 - odpamätanie zobrazených objektov, geometrické kompresné techniky, snímková súvislosť, zamedzenie opakovaného posielania primitív tým istým procesorom pre každý snímok



Architektúra s triedením na začiatku

V architektúre s triedením na začiatku príkazová jednotka interpretuje grafické príkazy a vykoná malé množstvo výpočtu, aby určila kam výsledné primitívum padne v zásobníku snímkov. Na základe toho sú primitíva poslané do príslušného grafického uzla alebo grafických uzlov, kde prejdú grafickými transformáciami a rasterizáciou. Pred vstupom do poslednej zastávky sa rozdelené oblasti zásobníka snímkov kombinujú do výsledného obrazu

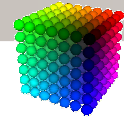




Architektúra s triedením na začiatku

výhody/nevýhody

- posiela výstup vypočítaných pixelov v jednotlivých regiónoch do výsledného zásobníka snímok
- rozdiel oproti Arch. s triedením v strede - či sa rozdeľujú primitíva podľa priestoru objektu alebo ich obrazu v rovine priemetne
- je potrebné na konci finálne pixely poslať do výsledného zásobníka snímok
- výrazné využitie snímkovej súvislosti – pri rozsiahlych dátach (počte objektov) je možné ponechať databázu objektov a nepreposielať ju
- Problém - rovnomerné využitie systému, pretože rozdelenie primitív podľa priemetu do priemetne môže byť značne nerovnomerné a záleží na tom ako sú rozdelené regióny medzi procesory
- obmedzenie algoritmami, ktoré môžu byť použité

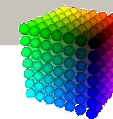


Veľkoplošné zobrazovanie

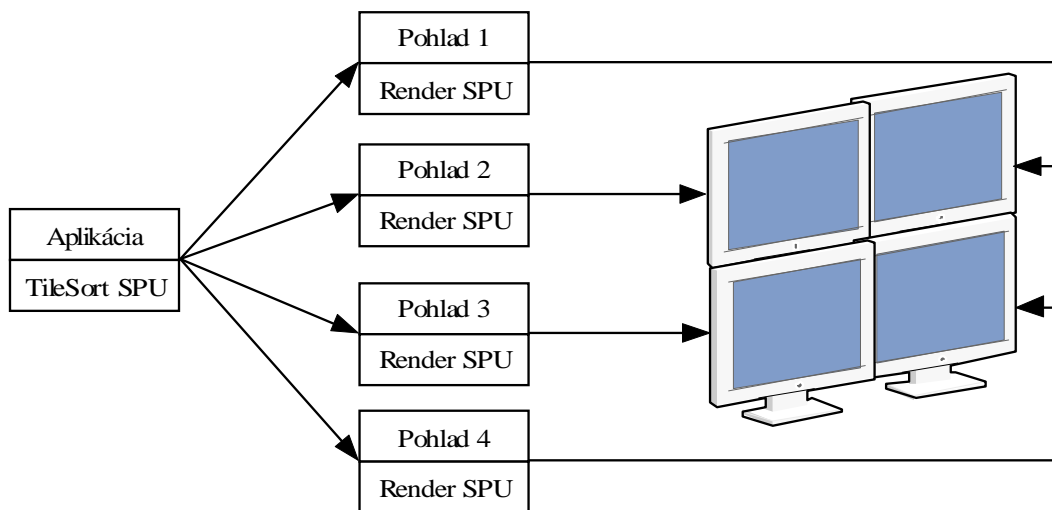


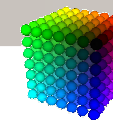
103" plazmový displej





Veľkoplošné zobrazovanie





Veľkoplošné zobrazovanie



Matrox DualHead2GO

rozlíšenia:

2048 x 768 alebo 2560 x 1024



Matrox TripleHead2GO

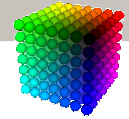
rozlíšenia:

3072 x 768 alebo 3840 x 1024

Matrox Veos

max. rozlíšenie: 5040x1080 (3ch)





Veľkoplošné zobrazovanie

AMD/ATi Eyefinity



3x1 skupina displejov (na výšku)



3x1 skupina displejov (na šírku)



3x1 skupina displejov (na šírku) + 1x Extend



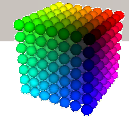
3x2 skupina displejov (na šírku)



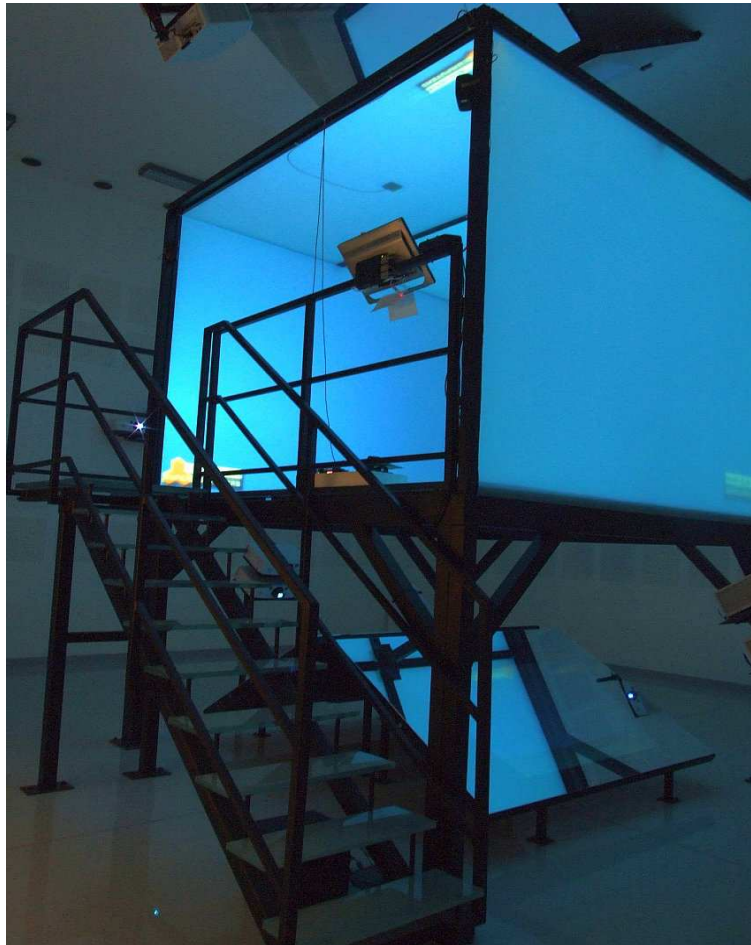
3x1 skupina displejov (na výšku) + 3x Extend



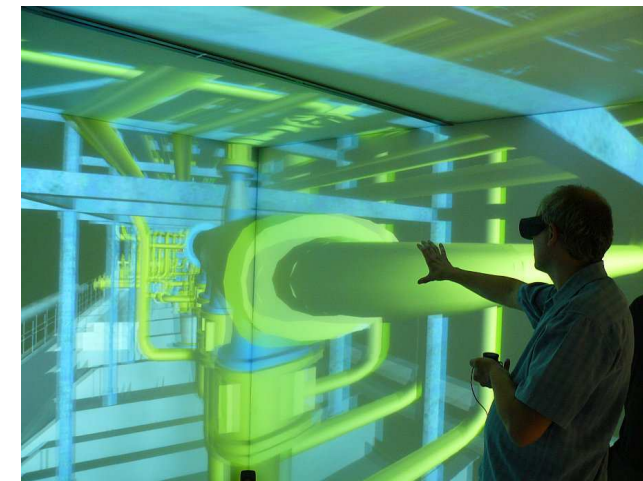
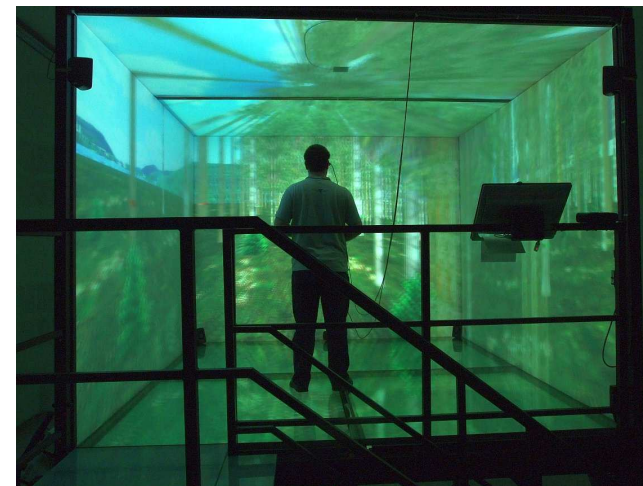
2x2 skupina displejov (na šírku) + 2x Extend

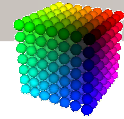


Virtuálna jaskyňa



(TU Zvolen)





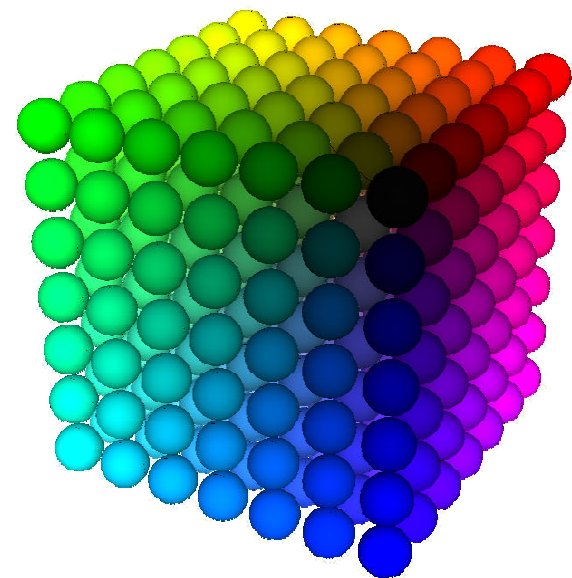
GPU/GPGPU

GPU - Graphics Processor Unit

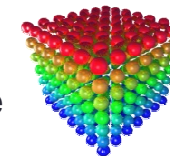
GPGPU - General Purpose Computing on Graphics Processors Units



riešenie Nvidia Tesla



© 2014 KPI FEI TU Košice



OTÁZKY ?
