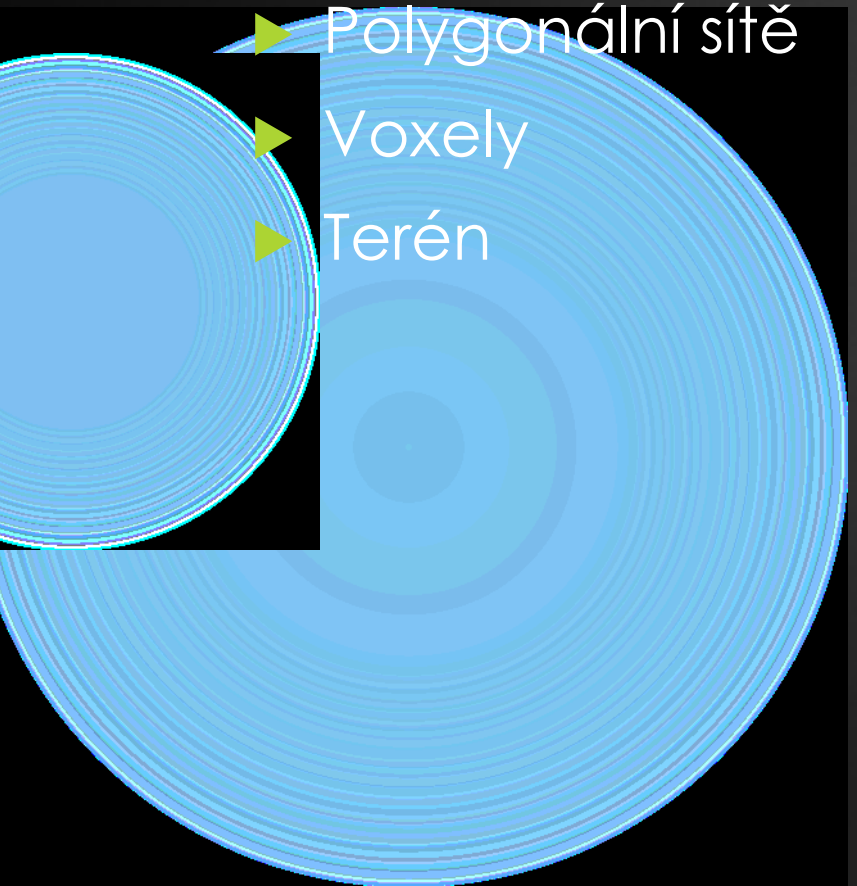
PV255 – Herní grafika

"IF IT LOOKS LIKE COMPUTER GRAPHICS, IT IS NOT GOOD COMPUTER GRAPHICS."

—JEREMY BIRN

Osnova

- Úvod
- Polygonální sítě
- Voxely
- Terén



Úvod

- Cíl herní grafiky:
  - ~~Fyzikálně přesné vykreslení~~
  - ~~Fotorealistická kvalita~~
  - Grafika, která se líbí hráčům
- Protichůdné požadavky:
  - aby to vypadalo co nejlépe
  - aby to bylo nejrychlejší
- Kompromis mezi rychlostí a kvalitou vykreslení

Úvod

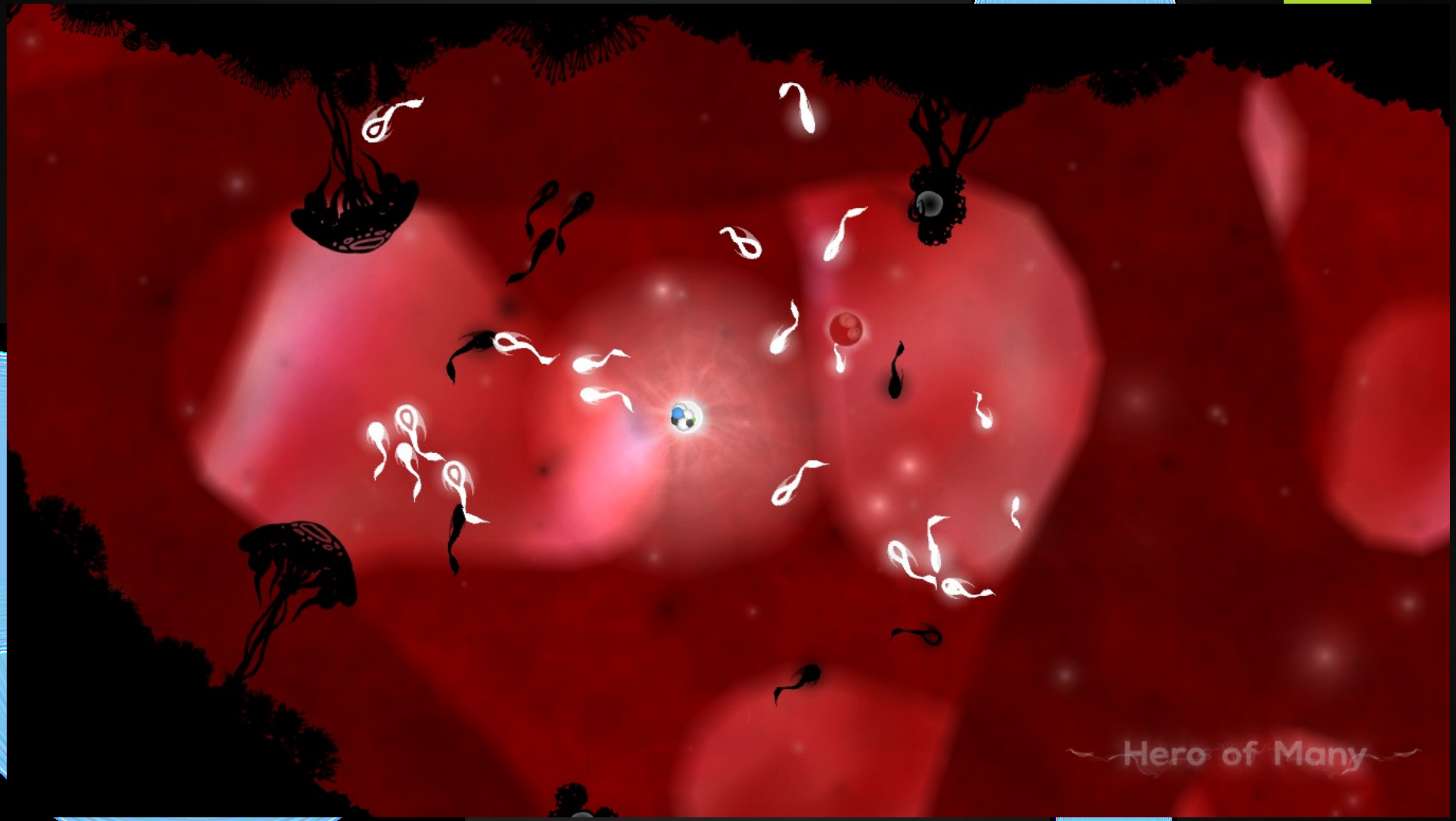
Nejen 3D, realistické vykreslení:

- 2D
- 2.5D
- Stylizace 2D / 3D
  - Siluety
  - Zvýraznění obrysů
  - Cartoon, cell shading, NPR
  - Ručně kreslená grafika

2.5D – Wolfenstein 3D (1992)

Siluety – Limbo (2010)

Siluety – Nightsky (2011)



Siluety – Hero of Many (2013)

Cartoon – Team Fortress II (2007)

Cartoon, outlines – Prince of Persia (2008)

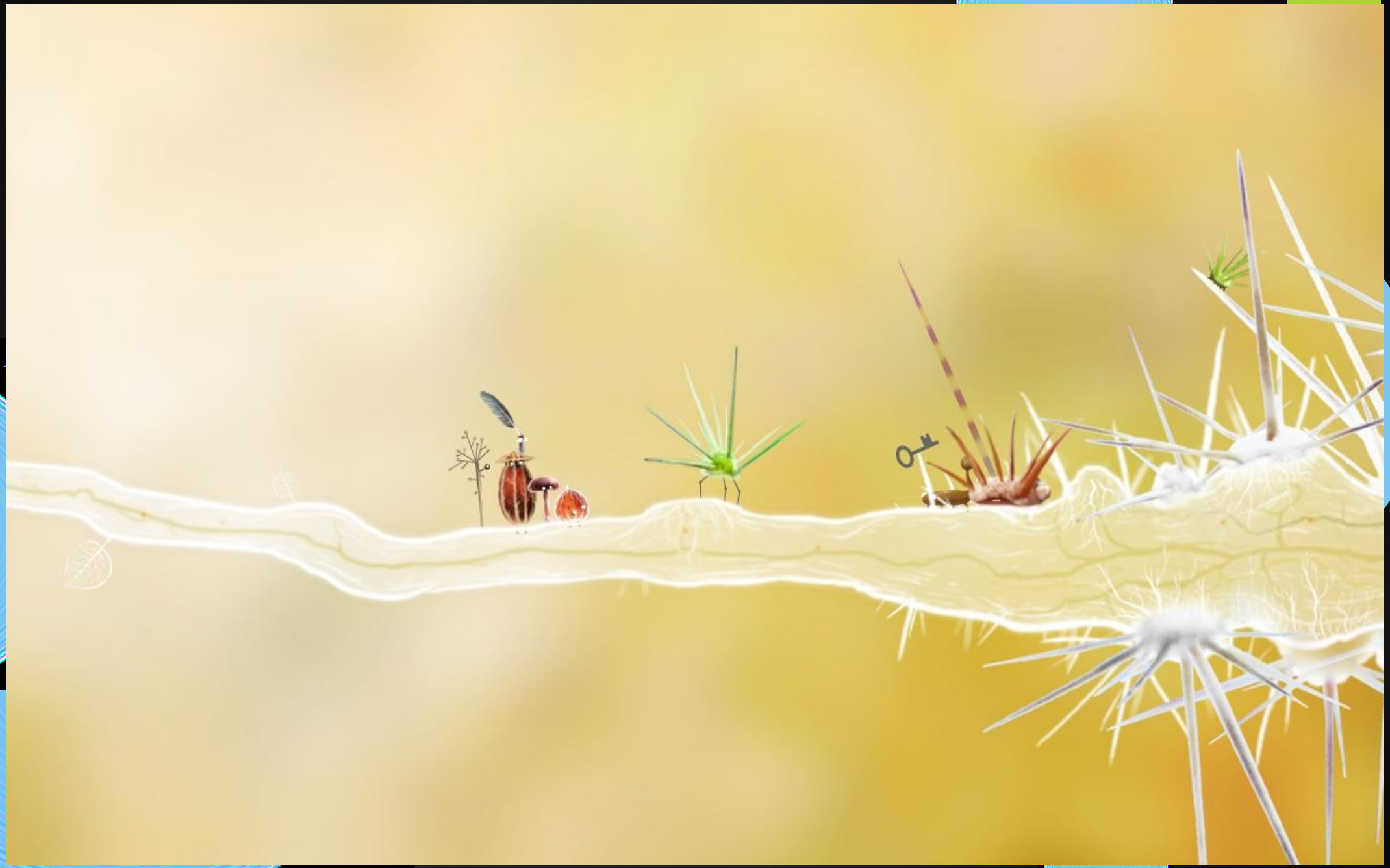
Cartoon, outlines – Dungeon Defenders (2011)

Cartoon – Bastion (2011)

Hatching – The Bridge (2013, Unity)

Ruční kresba – Inquisitor (2012)

Ruční kresba – Machinarium (2009)



Ruční kresba – Botanicula (2012)

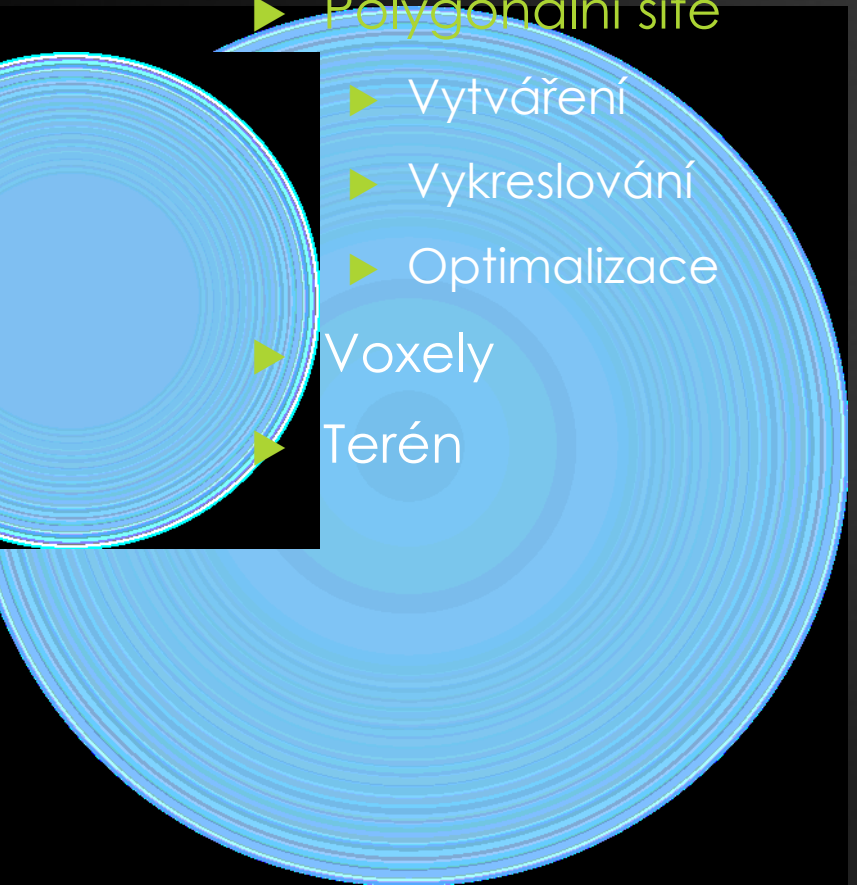
Vykreslování 3D dat

Typy vstupních dat:

- Grafická primitiva – parametrické modely
- Constructive solid geometry
- CAD
- Polygonální síť
- Voxelová mřížka
- Vzorkovaná data (CT, MRI, ...)
- Mrak bodů (point-cloud)
- Specifické struktury

Osnova

- Úvod
- Polygonální sítě
  - Vytváření
  - Vykreslování
  - Optimalizace
- Voxely
- Terén



Polygonální sítě – opakování

- „polygon soup“
  - neorganizovaná skupina polygonů
- „polygon mesh“
  - kolekce bodů, hran a plošek + informace o sousednosti
  - Bod, ploška – normála (vektor kolmý k povrchu)
  - Ploška – orientace (přední, zadní strana)
- Mesh + mapy (textury)
  - Barva
  - Bump-map (hrbolatost)
  - Normal-map
  - Paralax occusion map
  - Displacement-map
  - …

Polygonální síť – tvorba

- Ruční modelování
  - téma jiných předmětů (PB009, VV035, VV036)
  - Řada BP, DP: Tomáš Mádr – Tvorba herního charakteru
- Specifické nástroje
  - Postavy – MakeHuman, Fuse, Poser, …
  - Terén – Teragen,
  - Stromy – TreeGenerator,
- 3D skenování
  - Geriho hra (1997)
  - The Vanishing of Ethan Carter (2014)
- Procedurální
  - 3Ds Max, Blender, …
  - Unity: terén, stromy
  - Cryengine, Unreal Ungine

Polygonální síť – tvorba

Praxe: Herní charakter

- Tomb Raider (1996)
  - 230 polygonů
- Tomb Raider III (1998)
  - 300 polygonů
- TR – Angel of Darkness (2003)
  - 4 400 polygonů
- TR – Legend (2006)
  - 9 800 polygonů
- TR – Underworld (2008)
  - 32 000 polygonů
- Tomb Raider (2013)
  - 42 000 polygonů

Polygonální síť – vykreslování

Všechny polygony + všechny textury je potřeba:

1. nahrát na grafickou kartu
2. zpracovat:
   - transformovat
   - spočítat viditelnost
   - promítnout
   - rasterizovat
   - spočítat osvětlení, barvy
3. vykreslit

Polygonální síť – vykreslování

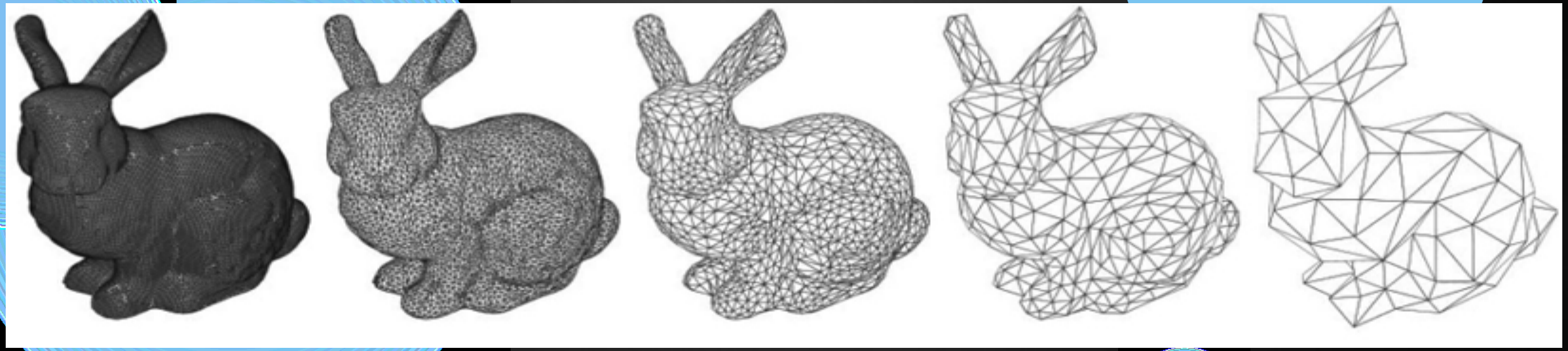
Grafické karty	Frekvence Procesoru MHz	Velikost Paměti MB	Propustnost Paměti GB/s	Zpracování Textur Mtexel/s	Passmark
RIVA 128 (1997)	100	4	1.6	100	n/a
GeForce2 MX 400 (2004)	250	64	2.65	1 000	n/a
GeForce4 MX440-SE (2009)	270	128	5.8	1 100	3
Geforce 240 (2009)	500	512	54	17 600	645
Geforce 580 (2010)	772	1 536	192	49 408	4 991
Geforce 780 (2013)	863	3 072	288	165 696	8 030
GTX Titan Z (2014)	837	12 288	672	338 000	8 015
GeForce GTX 980 Ti (2015)	1000	6 144	337	180 224	11 571

Polygonální síť – Optimalizace

- Před modelováním – nemodelujeme to, co není vidět:
  - pohled první vs. třetí osoby
  - budovy podél cest
  - motory u aut, …
- Před vykreslováním – optimalizace na výkon:
  - Level of Details
- Během vykreslování – výkon / vzhled
  - využití textur
  - teselace
  - odstřel
- Po rasterizaci – optimalizace na vzhled
  - vyhlazování (antialiasing)
  - hloubka ostrosti (Depth of Field)
  - …

Level of Details – opakování

- **Princip**: Čím větší vzdálenost objektu od kamery, tím méně detailů je vidět
- **Technika**: LoD
  - Více variant jednoho modelu s různou úrovní detailů
  - Podle vzdálenosti od kamery se vykresluje příslušná varianta modelu



- LOD na texturách = MIP mapping

LoD – tvorba

Vycházíme vždy z nejvyšší úrovně detailů, postupně model redukujeme:

- Ručně – „retopologizace“ – sami volíme, co je na modelu důležité, které hrany odstraníme
- Automaticky – algoritmy „mesh reduction“, „mesh decimation“
  - Volíme cílový počet polygonů
  - Snaha o zachování obrysů
  - 3D balíky, MeshLab, ...
- Až za běhu
  - Blending, interpolace
  - progressive mesh
  - Adaptive mesh

Diskrétní LoD

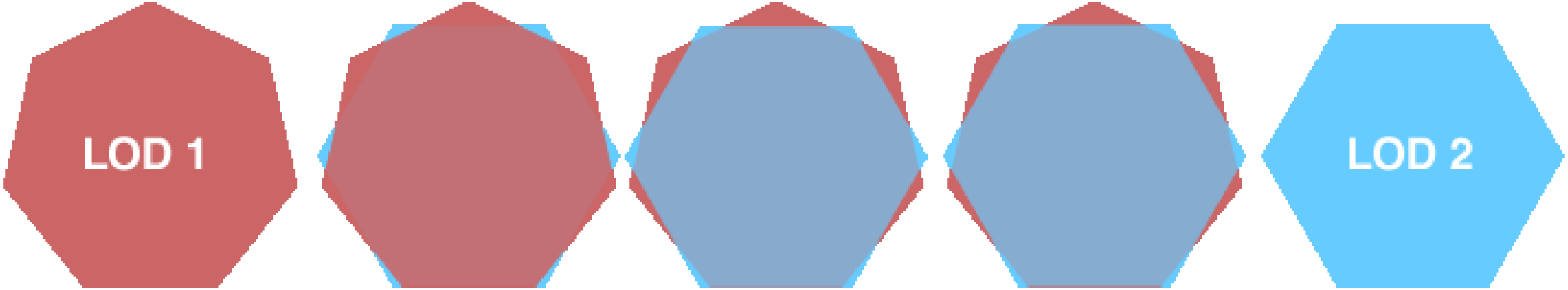
Diskrétní změna úrovně detailů (1976):

- Jednodušší varianta:
  - Na implementaci (přímá podpora v herních prostředích)
  - Na výpočet
  - Nejnižší LoD – pouze textura
- Nevýhoda: „popping“ efekt – viditelné přepínání mezi jednotlivými úrovněmi
- Videa:
  - https://www.youtube.com/watch?v=bNB-9chmRDE
  - https://www.youtube.com/watch?v=ioOKejJYlyc

Spojité LoD

Snaha vyhnout se popping efektu (1996)

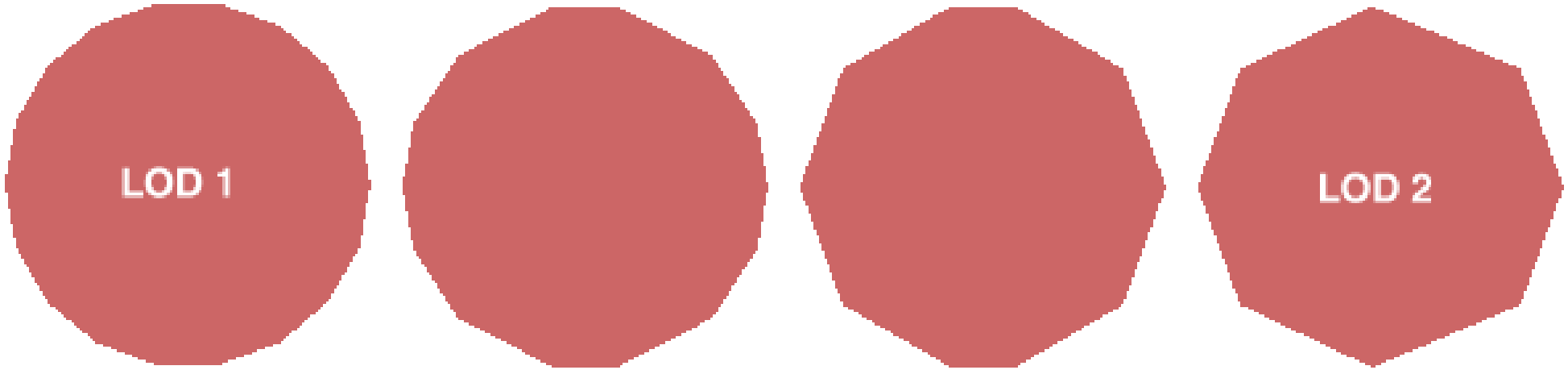
- Alpha blending – zobrazení obou úrovní současně



Spojité LoD

Snaha vyhnout se popping efektu (1996)

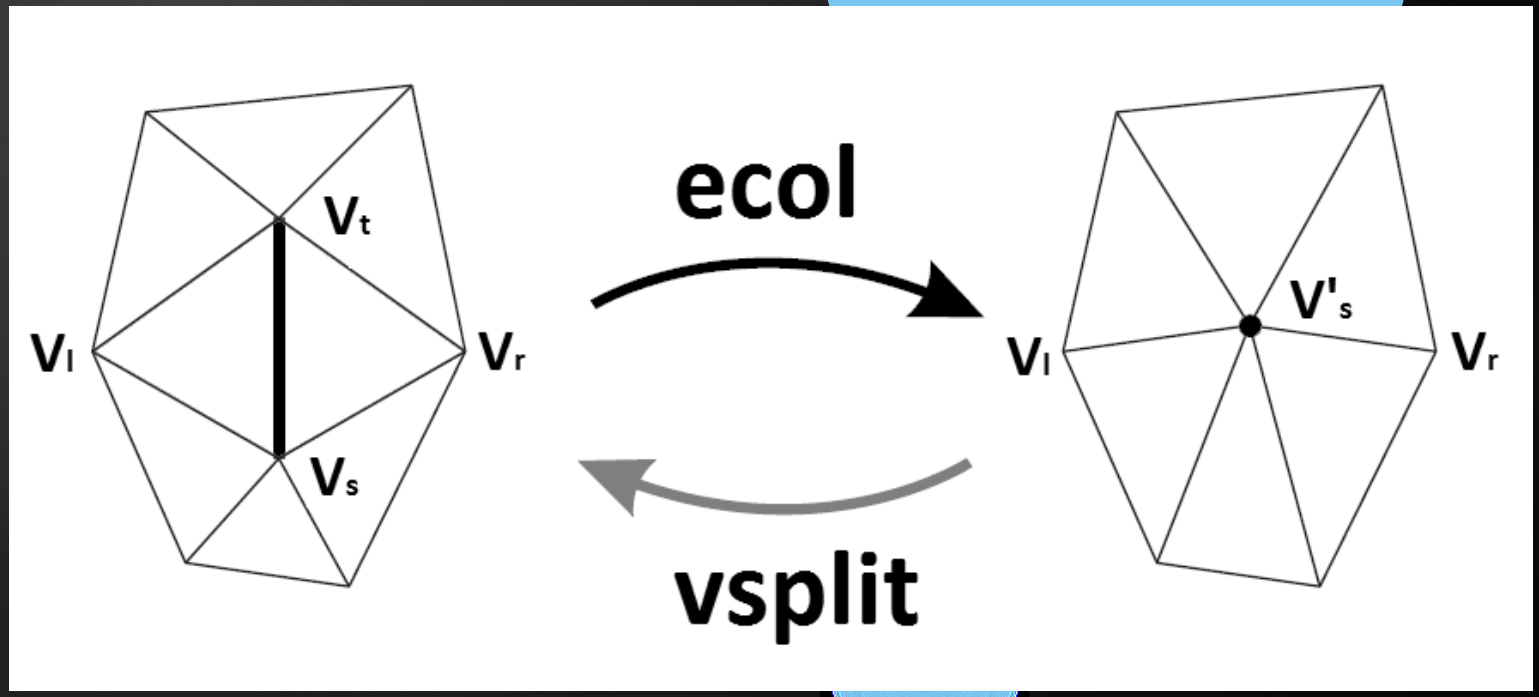
- Alpha blending – zobrazení obou úrovní současně
- Geomorphs – Interpolace mezi dvěma diskrétními LOD
  - Unreal Engine



Spojité LoD

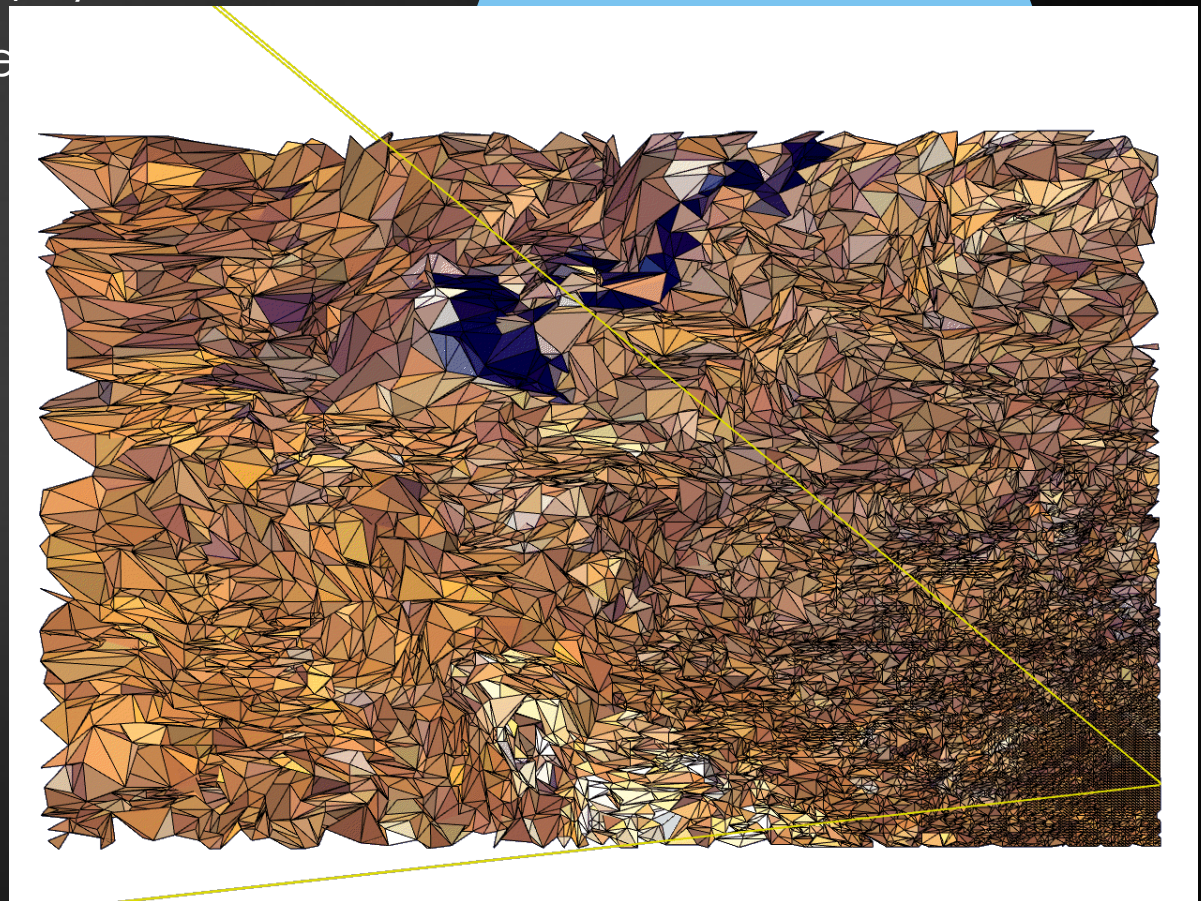
Progressive meshes – H. Hoppe (1996)

- Výpočet – postupné odebírání jednotlivých hran z nejvyšší úrovně
- Model – nejnižší úroveň + posloupnost přidávání hran (vyšší spotřeba paměti)



LoD – další varianty

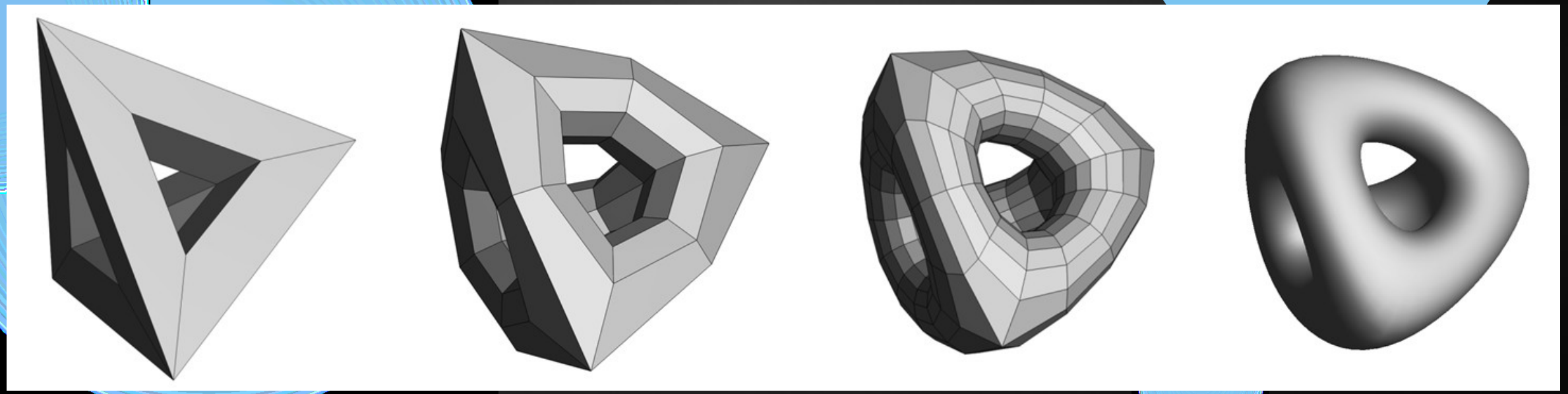
- HLOD – hierarchical LOD (2001)
  - Základem je hierarchie scény (scene-graph)
  - Vhodné pro scény s velkým počtem obje
- Pohledově závislé LOD (1997)
  - Čím blíže ke kameře,
  - tím více detailů
  - Siluety – více detailů
  - Vnitřní části – méně



Dělení povrchu (subdivision)

Dělení povrchu se zaoblením:

- Běžné v 3D modelování
- Film (Jan Pinkava, Geriho hra, 1997)
- Dělící schémata Catmull-Clark, Doo-Sabin, …



Mapování textur – opakování

- Princip: nemodelovat to, co je příliš malé nebo relativně ploché
- Technika: detaily jsou zachyceny pouze texturou, nikoli geometrií:
  - Barevná textura
  - Specular-map, bump-map, normal-map,
  - Mapy se používají při výpočtu stínování
  - Místo high-poly modelu se pro vykreslení použije low-poly model + sada textur:
    - Menší objem dat
    - Rychlejší vykreslení
    - Není závislé na vzhledu textury
  - Je potřeba znát mapování textury na 3D model (UV mapping)

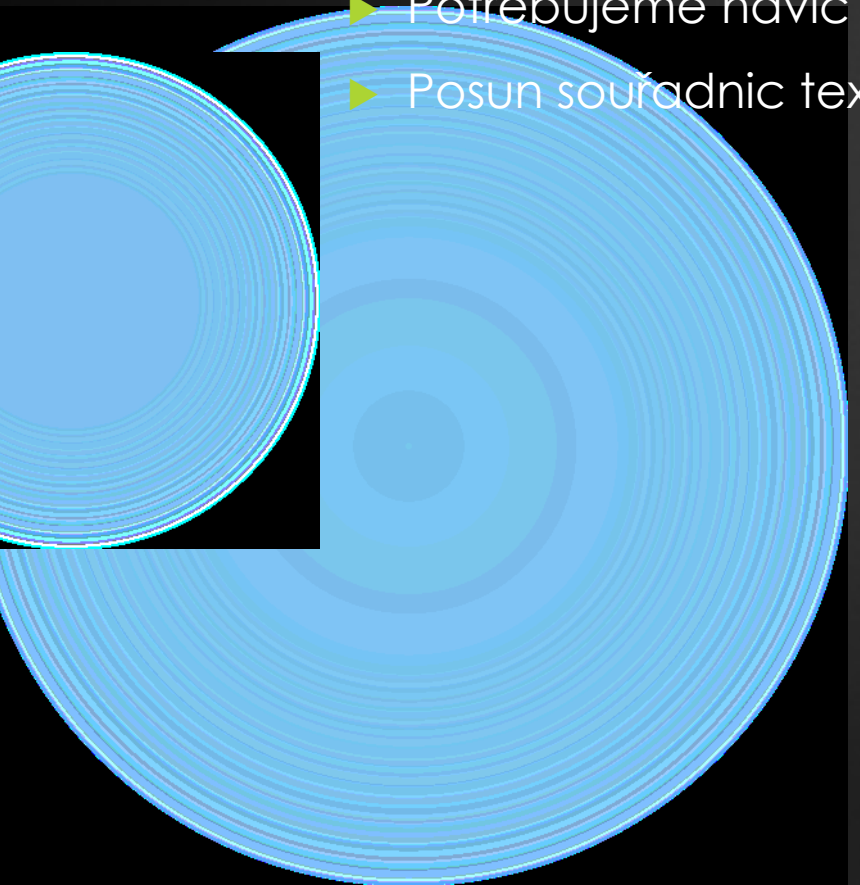
Tomáš Mádr - high-poly model:
20 000 000 polygonů

Tomáš Mádr:
low-poly model + normálová mapa:
11 500 polygonů

Tomáš Mádr: low-poly model
+ barevná, odrazivá
a normálová mapa:
11 500 polygonů

Parallax mapping

- Rozšíření techniky normal-mapping (2001)
  - Potřebujeme navíc výškovou mapu
  - Posun souřadnic textury na základě úhlu pohledu a výškové mapy



Parallax Occlusion Mapping

- Přidává sebe-zastínění, vrhání stínů
- Videa:
  - https://www.youtube.com/watch?v=UeF-kCr_vyo
  - https://www.youtube.com/watch?v=gcAsJdo7dME

Shadery – opakování

Shader = Program k řízení programovatelných částí GPU:

- Pixel/fragment shader
- Vertex shader
- Geometry shader
- Tessellation shader
- Compute shader

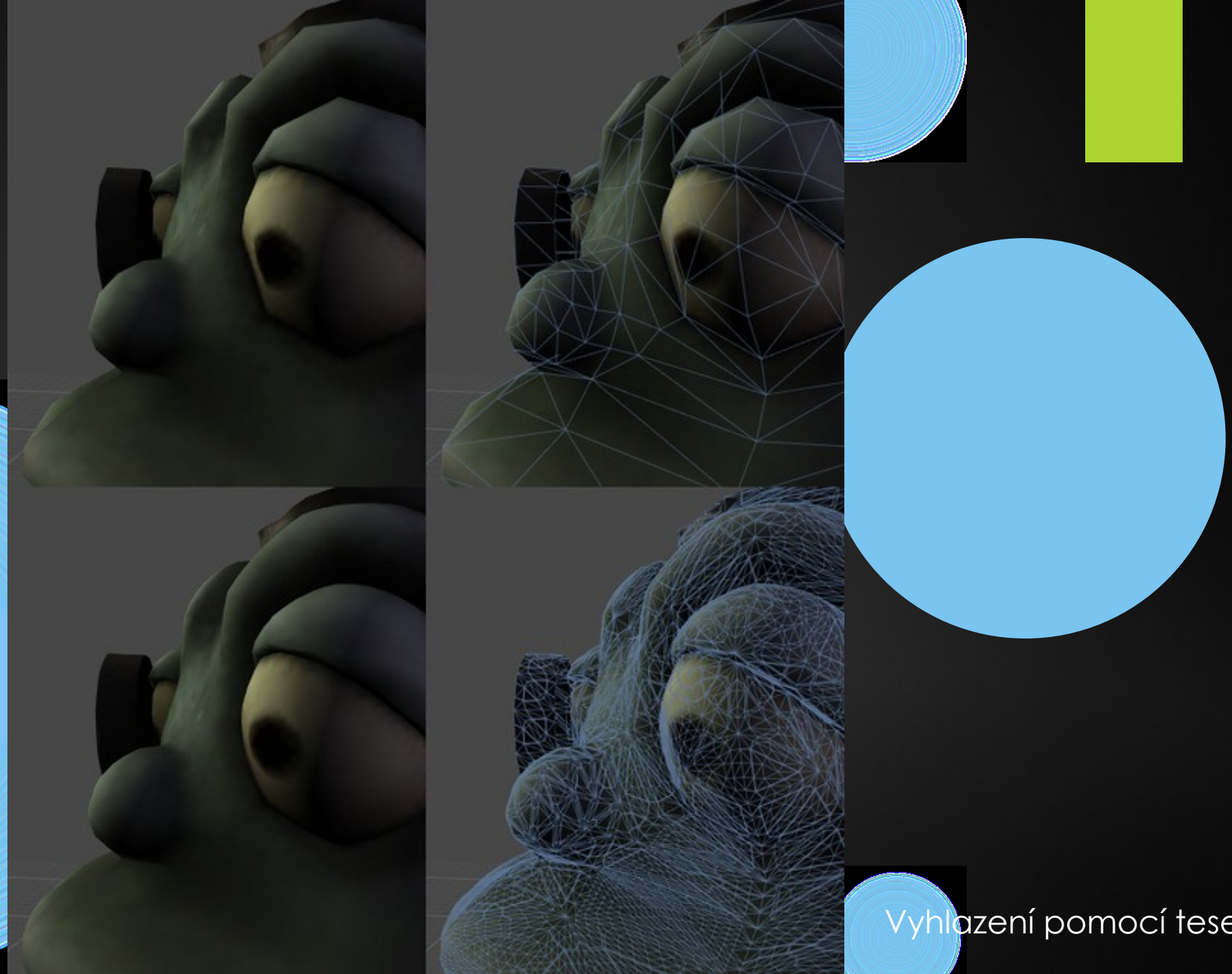
- Téma jiných přednášek a předmětů

Shadery – historie

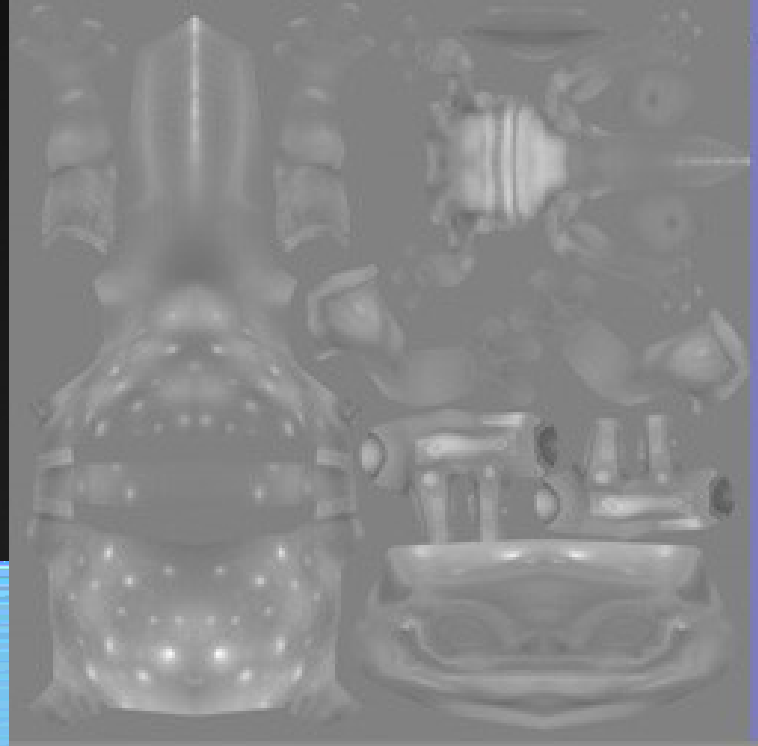
	Fragment Pipeline	Vertex Pipeline	Texture Units	Shader Processor (cuda cores)
RIVA 128 (1997)	1	0	1	na
GeForce2 (2004)	2	0	4	na
GeForce 8500 GT (2007)			8	16
GeForce 240 GT (2009)			32	96
GeForce 580 GTX (2010)			64	512
GeForce 680 GTX (2012)			128	1536
GeForce 780 GTX (2013)			192	2304
GTX TITAN Z (2014)			480	5760

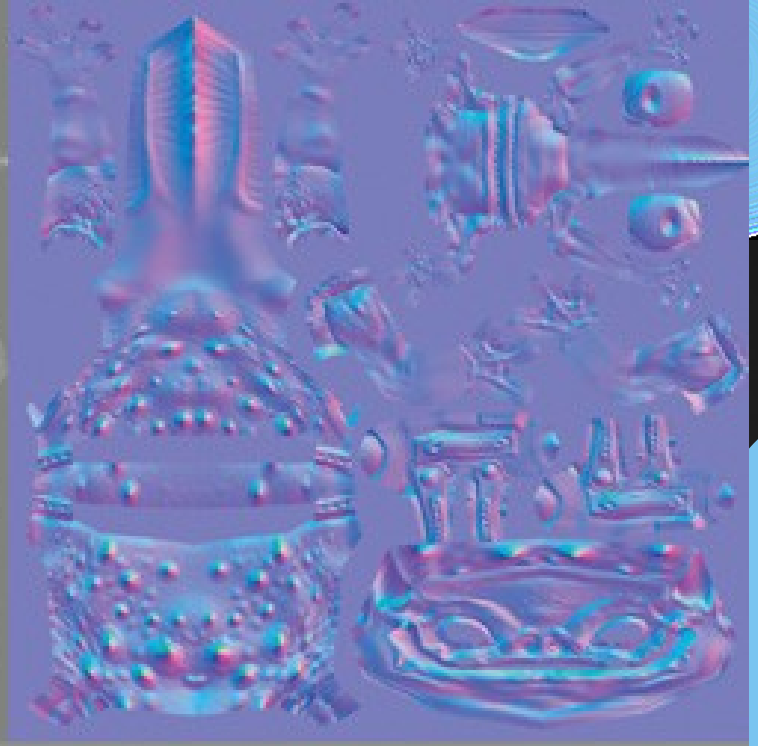
Teselace na GPU

- Teselace = dělení roviny
- Teselace na GPU – možnost dělit polygony až „na grafice“
- Teselace + posun vrcholů:
  - Vyhlazování modelů – subdivision
  - Úprava geometrie pomocí map – displacement
  - Adaptivní úprava v rámci jednoho modelu
- Hry: Alien vs. Predator, Metro 2033, Skyrim
- Video: https://youtu.be/-uavLefzDuQ?t=51s

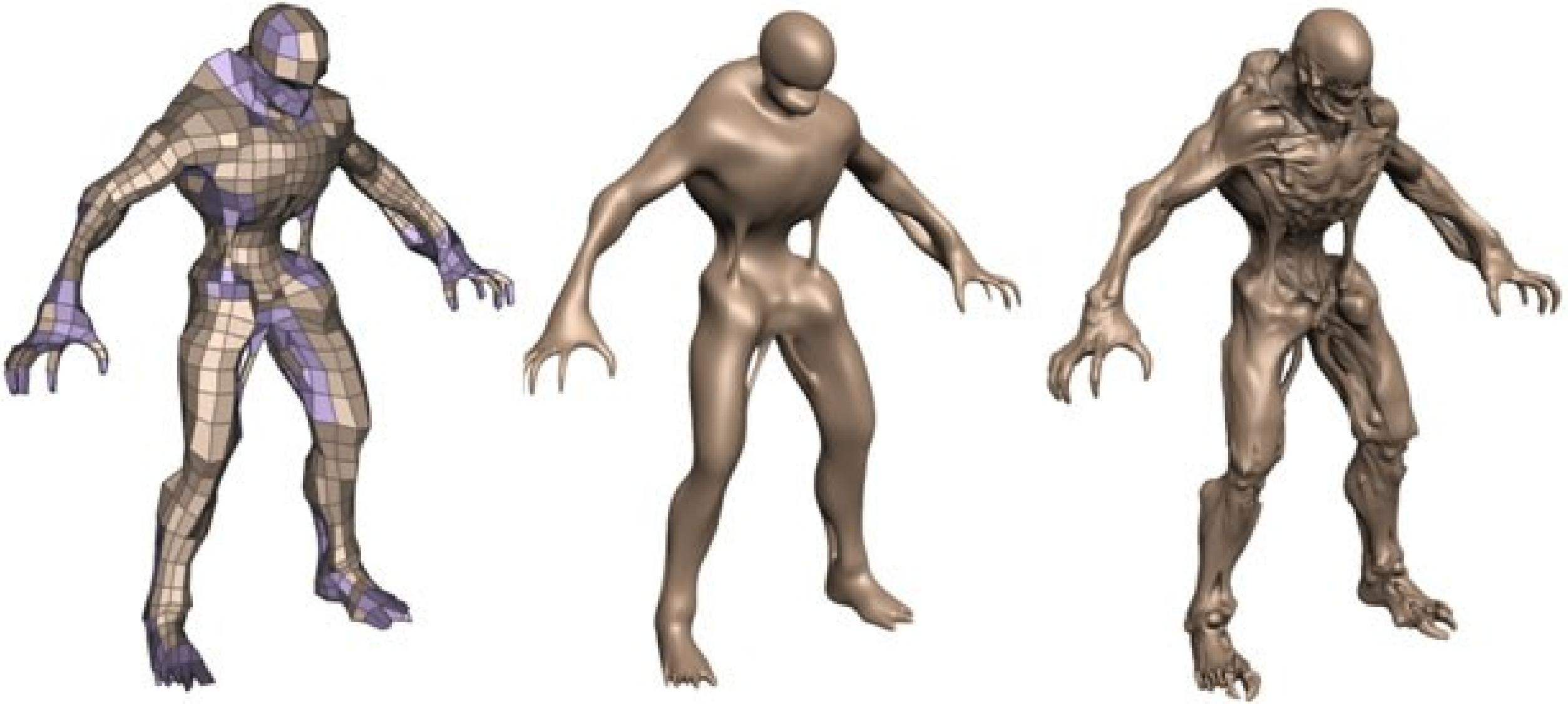


Vyhlazení pomocí teselace

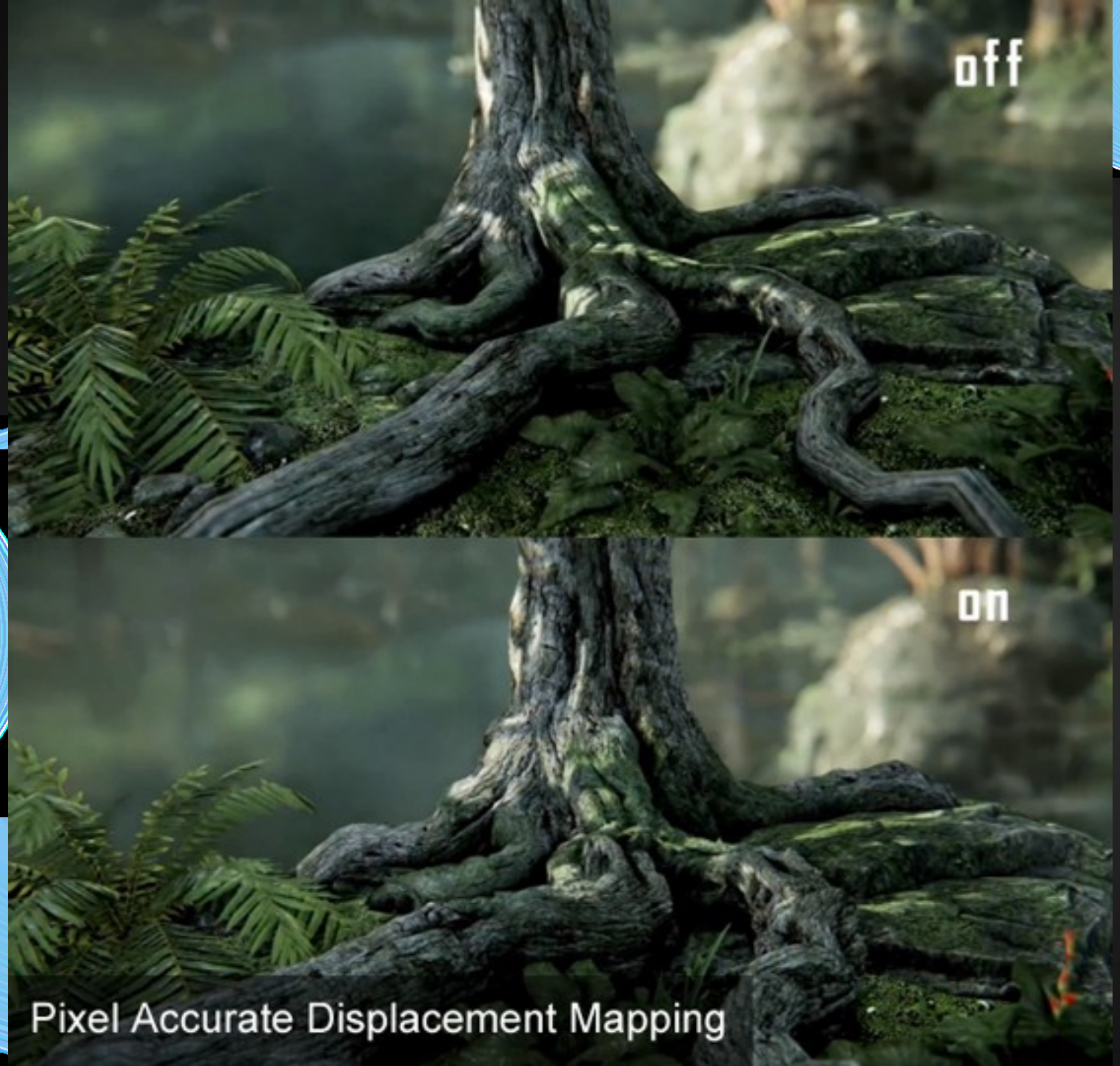




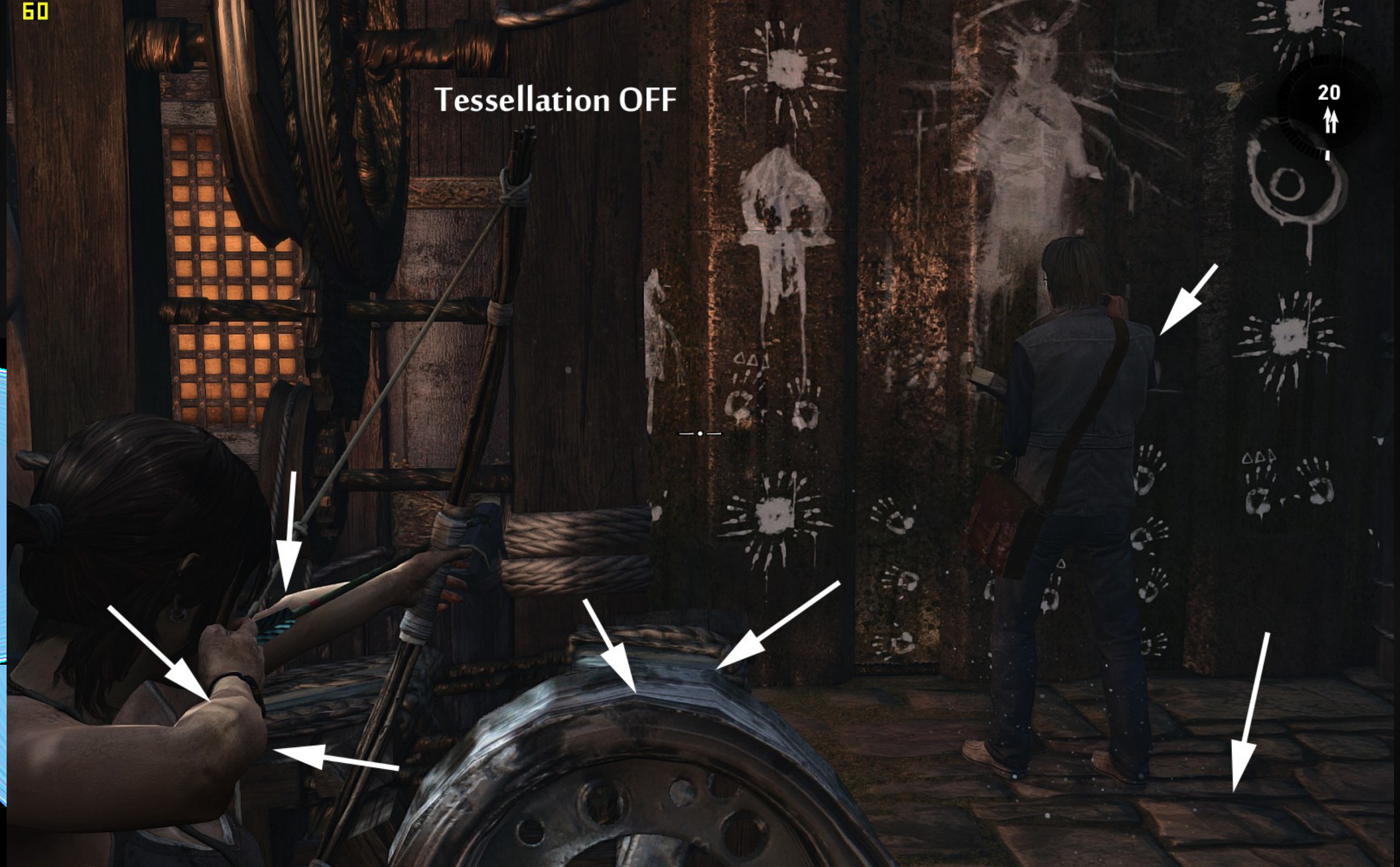
Vyhlazení
+ displacement mapping
pomocí teselace



Vyhlazení + displacement mapping pomocí teselace

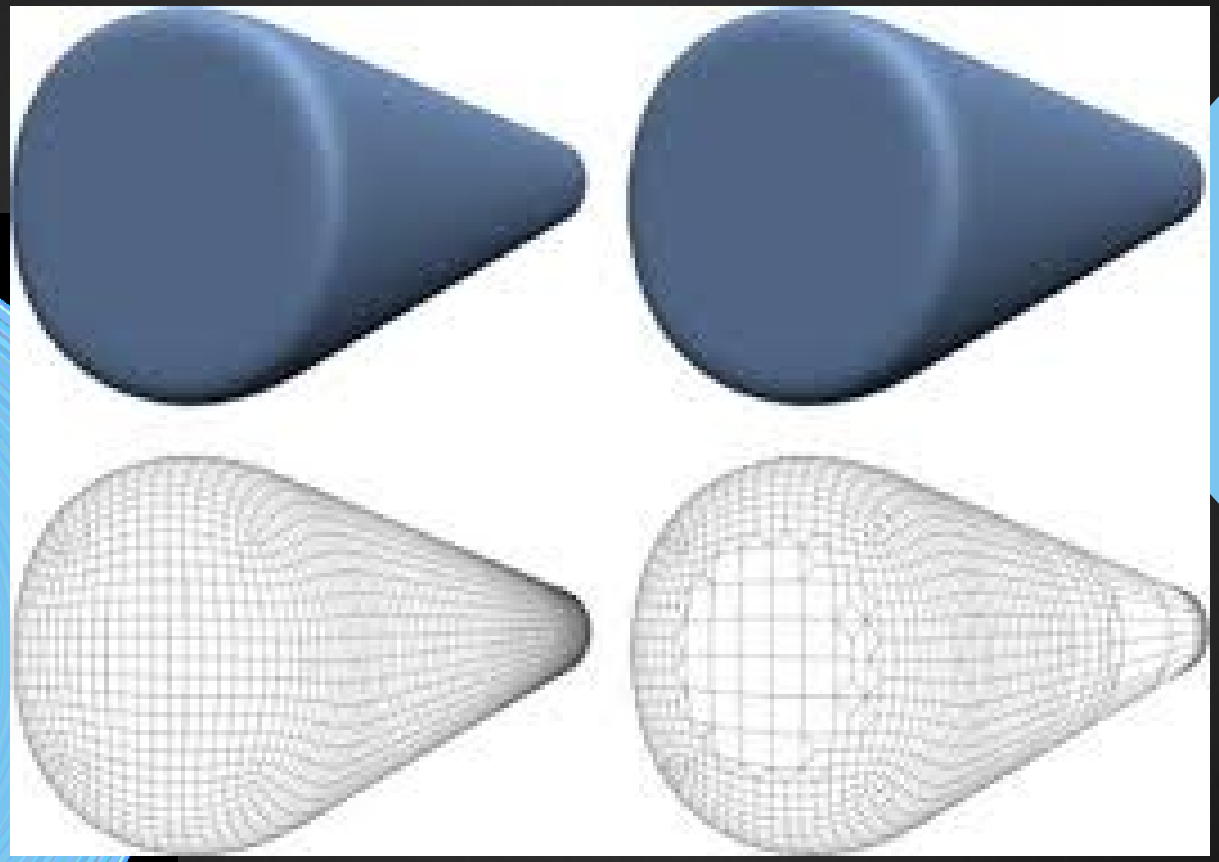


CryEngine:
displacemet pomocí teselace


60
Tessellation OFF
20

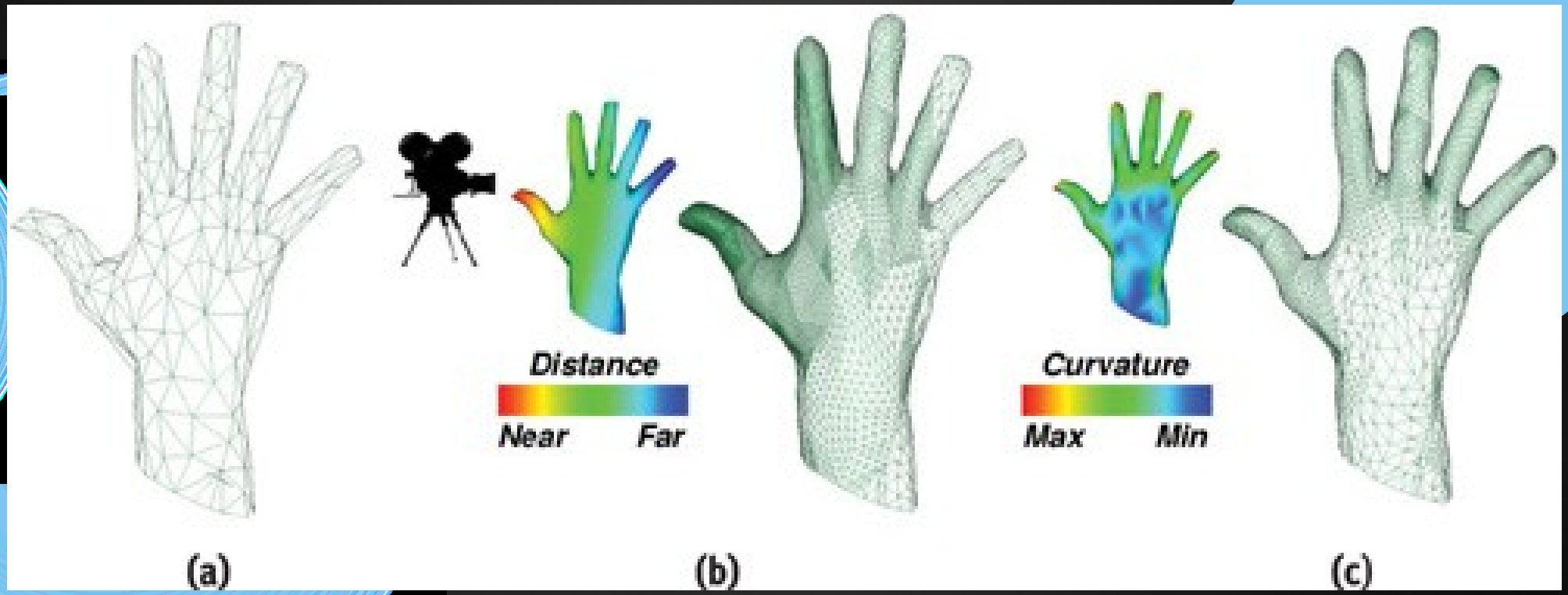

60
Tessellation ON
20

Adaptivní teselace



Klasická (vlevo) a adaptivní (vpravo) teselace (podle zakřivení)

Adaptivní teselace



Adaptivní teselace: podle vzdálenosti od kamery (b), podle zakřivení (c)

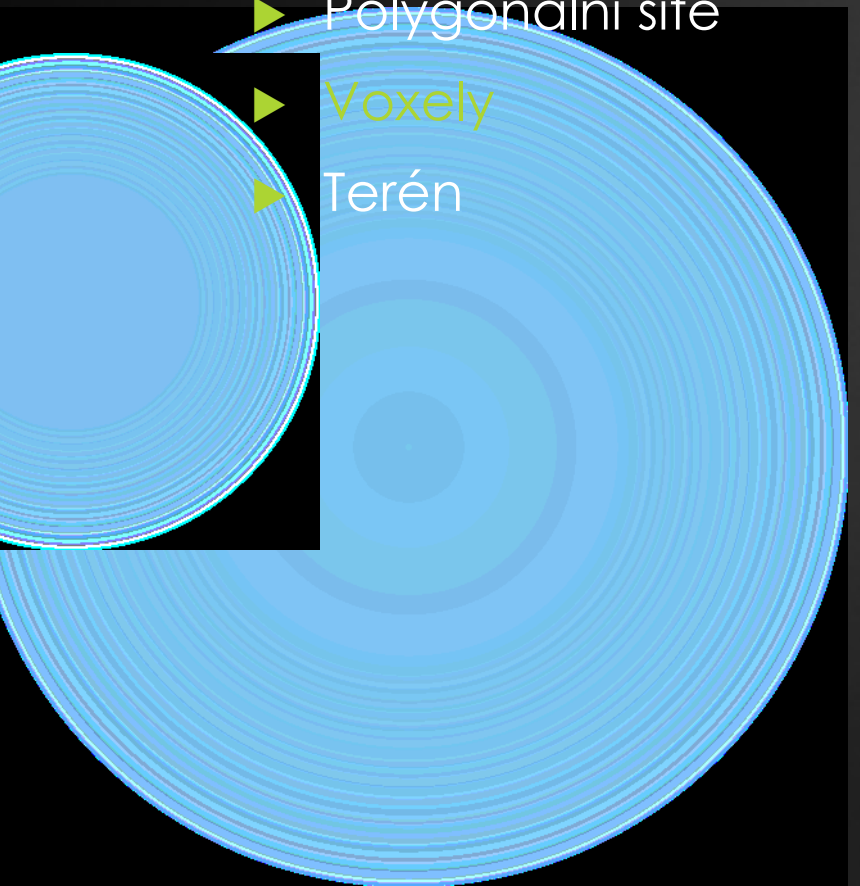
Optimalizace v prostoru obrazu

- Post-processing:
  - Vylepšení / změna vizuálního dojmu v prostředí obrazu
  - Dostupné v herních prostředích
  - Téma jiné přednášky

- Anti-aliasing
- Motion blur, camera motion blur
- Filmové zrno (grain)
- Glow (Bloom), lens flares
- Mapování barev (noční vidění)
- Mlha
- Hloubka ostrosti (DoF)
- Lens effects (vignetting, chromatic abberations, tilt shift)
- NPR – edge detection,

Osnova

- Úvod
- Polygonální sítě
- Voxely
- Terén



Hry využívající voxely

Hráč nemůže aktivně měnit voxely:

- Blade Runner (1997)
  - charaktery, předměty
- Outcast (1999)
  - nepoužívá voxely
  - Terén – vrhání parsku na výškovou mapu
- Sid Meier's Alpha Centauri (1999)
- Command & Conquer: Red Alert 2 (2000)
  - jednotky
- Crysis (Cryenige, 2007)
  - terén pomocí kombinace výškové mapy a voxelů

Hry založené na voxelech

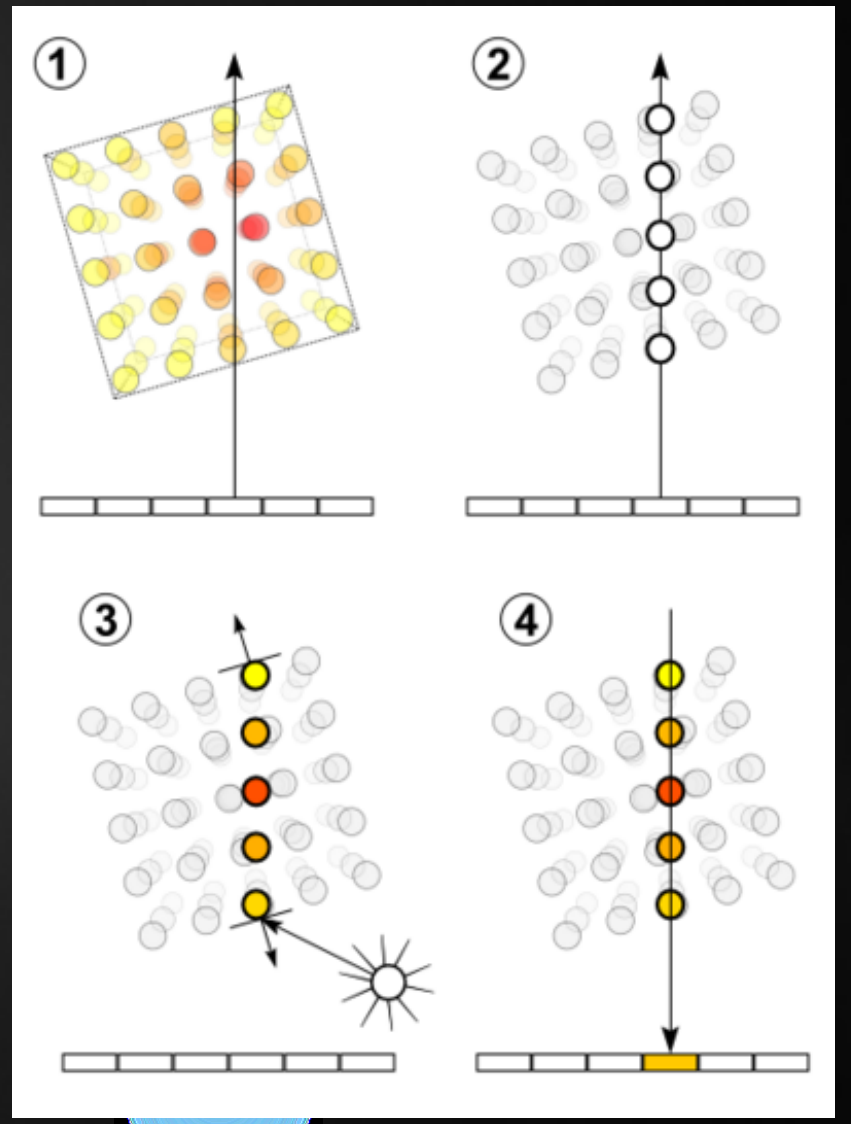
Hráč může aktivně měnit voxely:

- Worms 4 (2005)
- Voxelstein (2008)
- Minecraft (2009)
- Space Engineers (2013, early access (12. 10. 2015), indie)
- Caste Story (2013, early access (12. 10. 2015), indie)
- 7 Days to Die (2013, early access (12. 10. 2015), indie)
- Planet explorers (2014, early access (12. 10. 2015), indie)
- Planets3 (2014, alpha (12. 10. 2015), indie)
- Everquest Next, Landmark (2014-Beta (12. 10. 2015), SOE)
- Blockspace (2014, early access (12. 10. 2015))
- StarForge (2014, indie, "mostly negative")

Voxely – opakování

- Voxel = volume element
  - Souřadnice v 3D prostoru
  - Další informace (barva, materiál, fyzikální vlastnosti, …)

- Vykreslení voxelové mřížky
  - Převod na iso-plochy – marchnig cubes, ...
  - Přímé zobrazování – např. vrhání paprsku (ray-casting)



Voxely – datové struktury

- Počty voxelů:
  - Minecraft: 2.6e17 voxelů (6e7 x 6e7 horizontálně, 256 vert.)
  - Planets3: 3.2e13 voxelů na jednu planetu

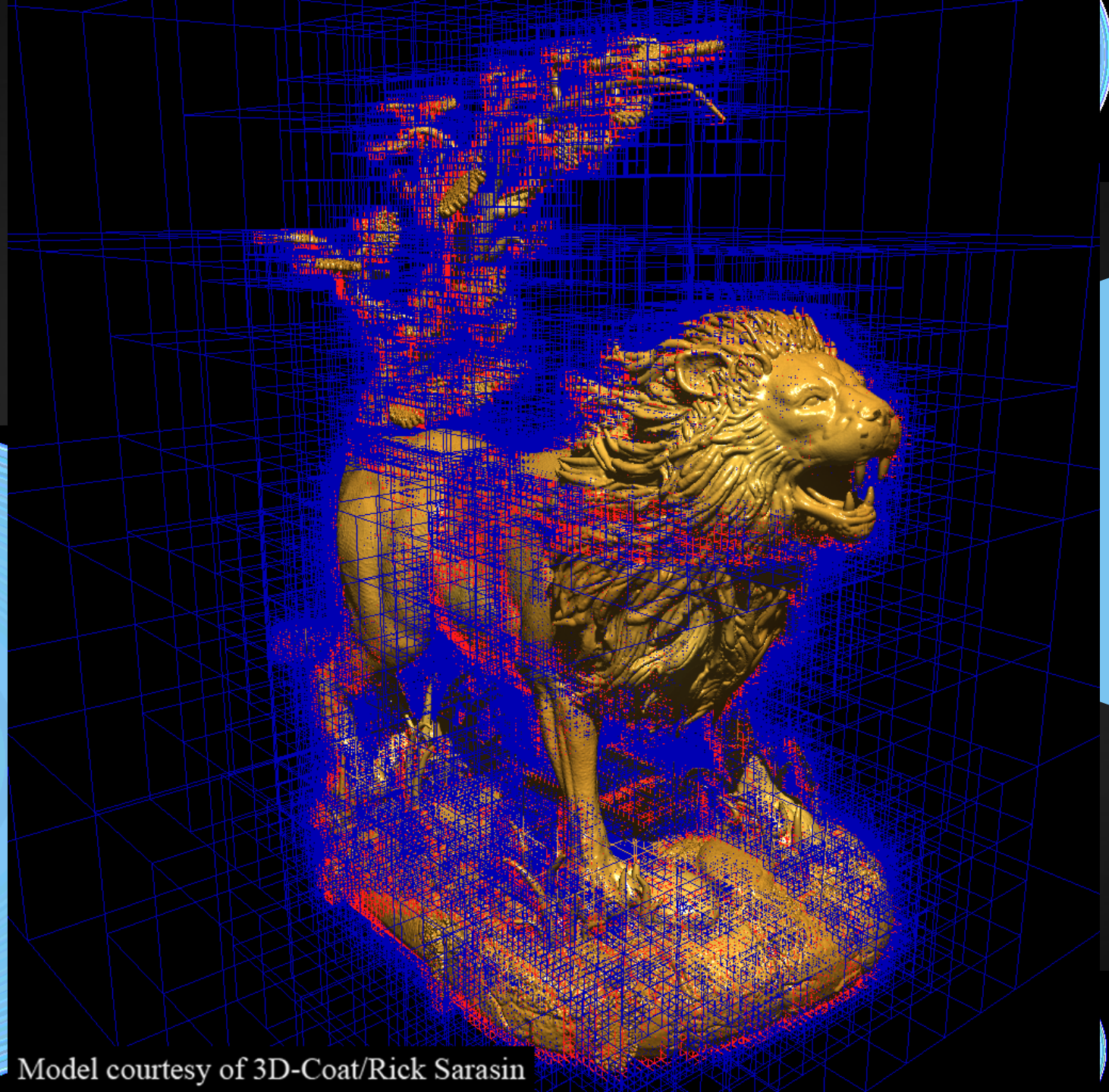
- ~~Vícerozměrné pole~~

- Chunks
  - chunk = skupina voxelů
  - Jeden blok paměti
  - Jeden „render call“ (display lists, VBO, VBA, …)

- Minecraft:
  - 1 chunk = 65 536 voxelů (16 x 16 x 256)
  - V paměti současně 25 až 1089 chunků
  - Žádná interakce s bloky, které nejsou v paměti (stromy nerostou)

Voxely – datové struktury

- Octree (1890):
  - hierarchické dělení prostoru na osminy
  - Rychlejší výpočet průchodu paprsku
  - Větší spotřeba paměti
- Sparse voxel octree (SVO):
  - Princip: Většina voxelů je prázdná
  - Technika: Dělí se pouze částečně plné voxely
  - Varianta LOD
  - Řešitelné na GPU (2010)



Model courtesy of 3D-Coat/Rick Sarasin

Voxely – techniky ukládání dat

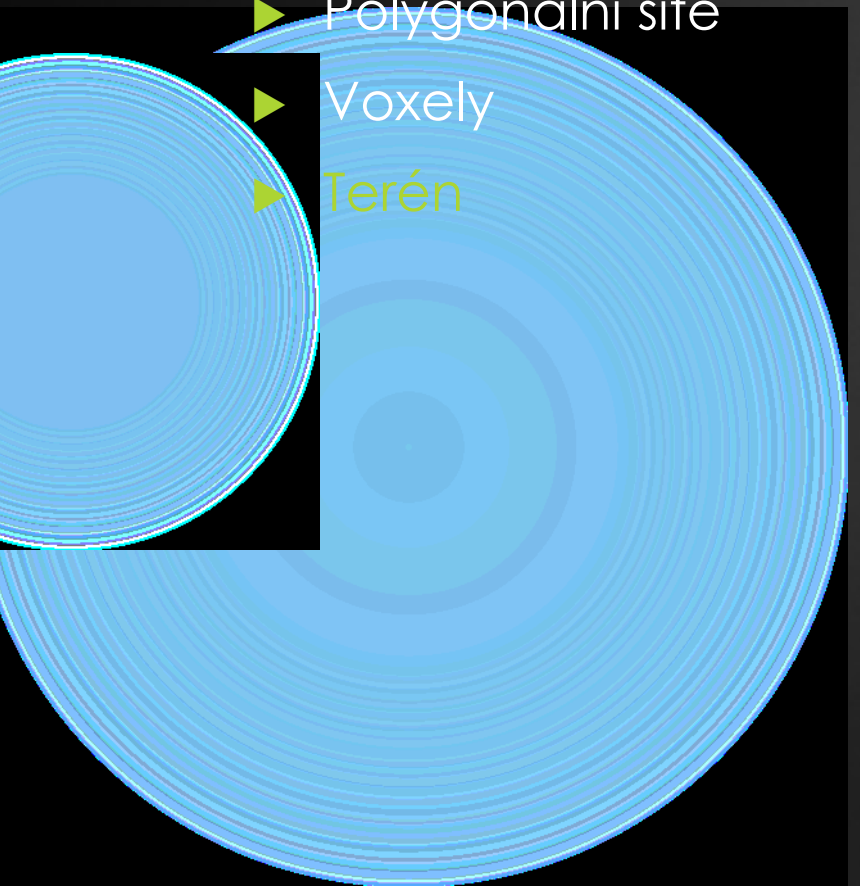
Množství dat:

- 2.6e17 * 2B ~ 500 000 000 TB

- Procedurální generování terénu
  - Ukládáme parametry generátoru, nikoli samotná data
    - Minecraft: deterministický generátor - celý terén definovaný jedním číslem
  - Plus ukládání změn

Osnova

- Úvod
- Polygonální sítě
- Voxely
- Terén



Terén – vstupní data

- Snímkování reálných dat – Digital Elevation Models (DEM)
  - ČR: ČUZK
  - Svět: USGS
  - Přesnost:
    - Horizontální: metry na pixel
    - Vertikální: centimetry na pixel
  - Ukládá se pouze nadmořská výška v každém bodě

Terén – vstupní data

- Generování
  - Fraktální geometrie
  - Brownův pohyb
  - Posun středního bodu (mid-point displacement)
  - Náhodné poruchy (random faults)
  - 2D Perlinuv šum
  - Post-processing – vodní, větrná eroze
- Nevýhoda - nelze generovat jeskyně, převisy, …

Terén – ukládání

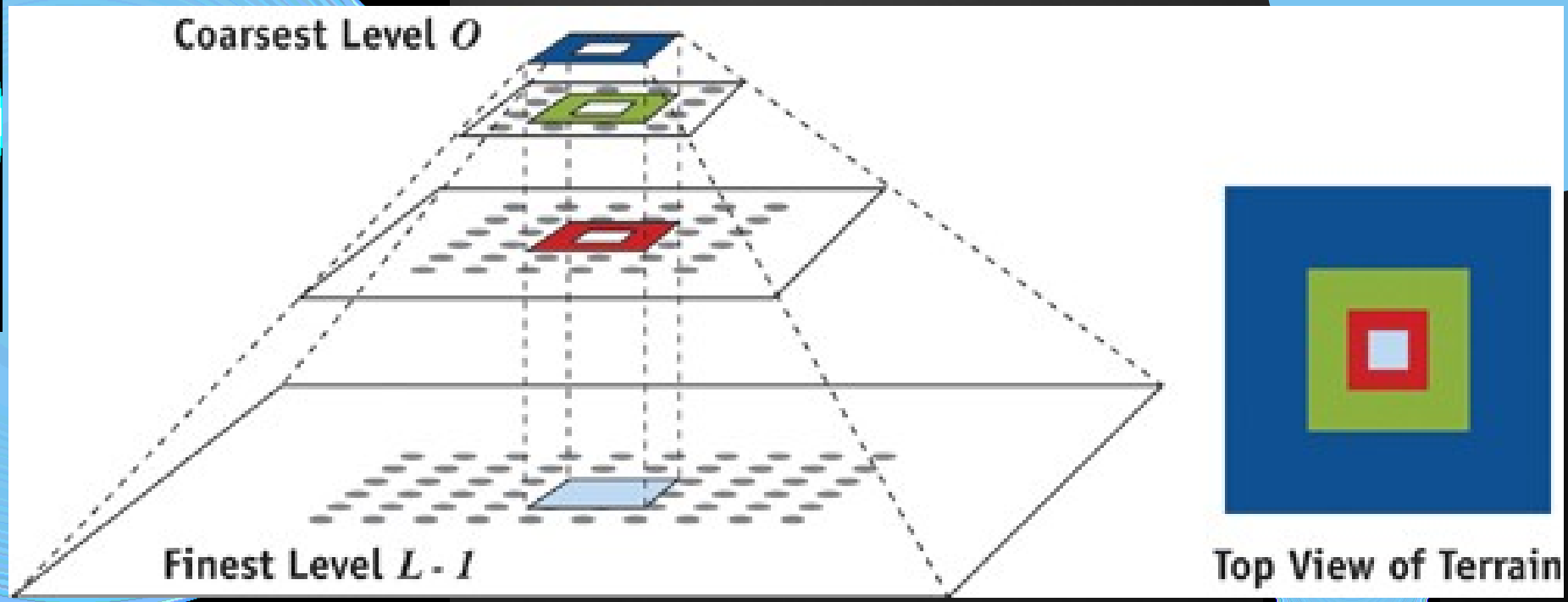
- Reálná data
  - Textury
    - Polygony jsou tvořeny až za běhu
  - Pravidelná trojúhelníková mřížka
    - Nejjednodušší
    - Nejméně efektivní
  - Nepravidelná trojúhelníková síť (TIN)
    - Je potřeba analýza vstupních dat
    - Efektivnější - méně polygonů na rovinných částech
  - Digitální kontury – vrstevnice
    - Lze snadno vytvářet triangle-strip (až za běhu)
- Generovaná data
  - Parametricky
  - Minecraft – jediné číslo pro pseudonáhodný generátor

Terén – vykreslování

- Polygonální síť
  - Patches + LoD
  - Problém – jak spojovat pláty s různou úrovní detailů
- Voxely
  - Chunks – převedení chunku na polygony
  - Minecraft – vykreslování krychlí
- Herní prostředí (Unity, Cryengine, Unreal Engine, …)
  - Vlastní optimalizace
  - Schopnost převést výškovou mapu na vnitřní reprezentaci

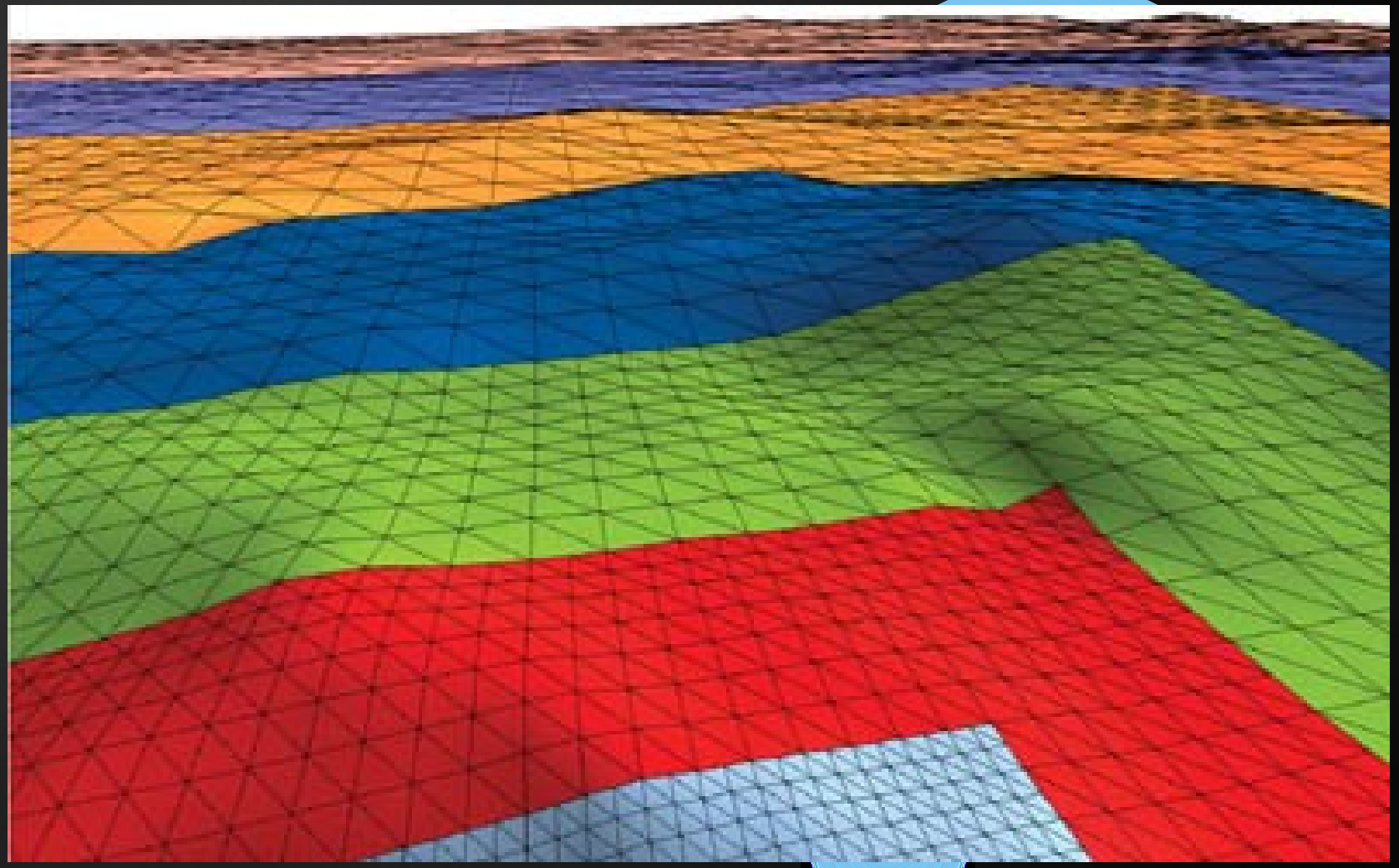
Terén – vykreslování – „Clip-maps“

- Základem je DEM, přepočtený na MIP pyramidu
- Výsledná výšková mapa:
  - se generuje v reálném čase, podle polohy kamery;
  - skládá se z různých vrstev MIP textury



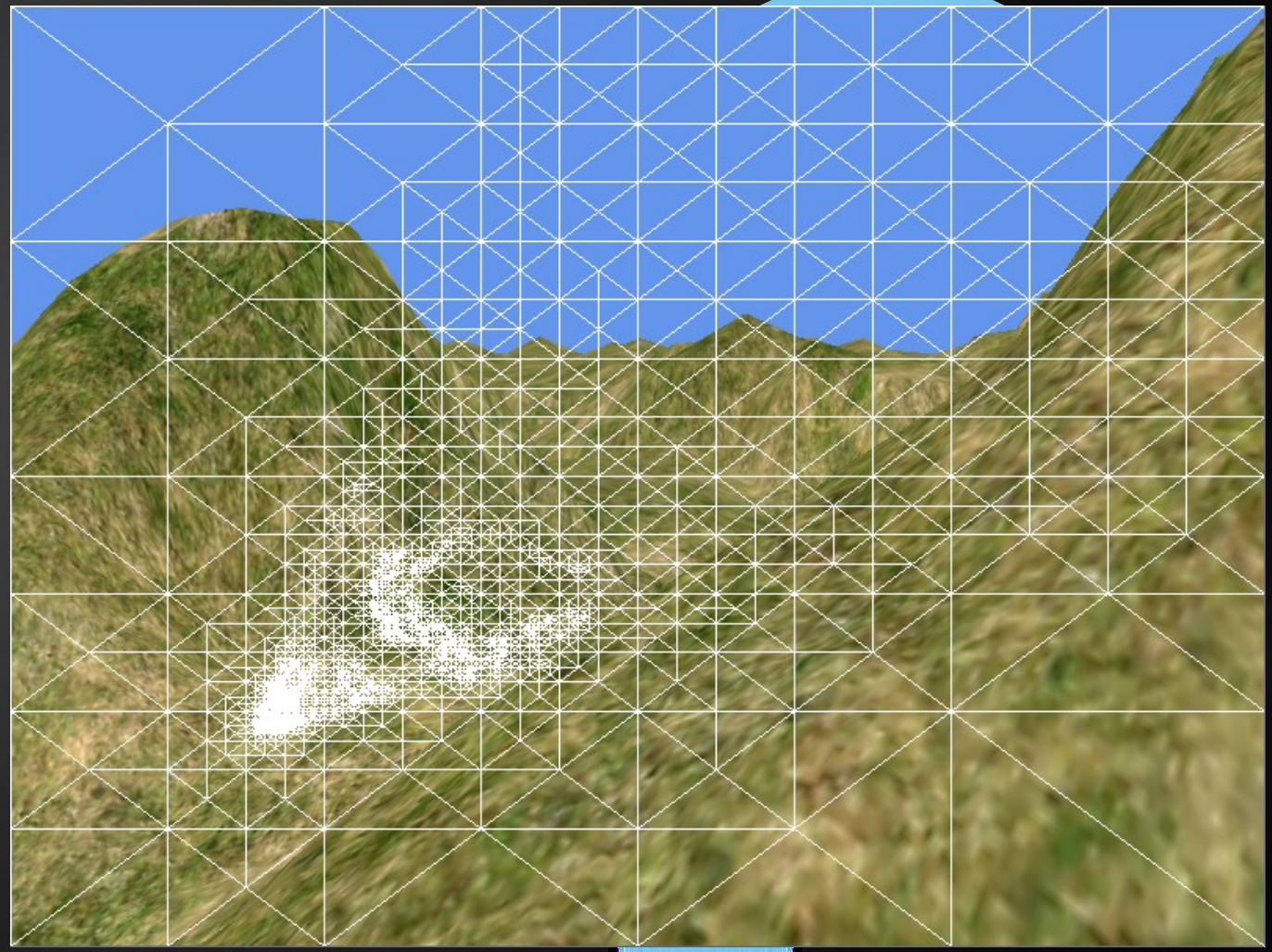
Terén – vykreslování – „Clip-maps“

- Cílem je, aby všechny trojúhelníky zabírali přibližně stejný počet pixelů



Terén – vykreslování – ROAM

- Varianta spojitého LOD pro terén
  - Binární strom – trojúhelník se dělí na dva menší
  - Pohled kamery určuje jaká hloubka stromu se použije
- Video (1:03): https://www.youtube.com/watch?v=PPjWW8uPp3o



Terén – Převisy, jeskyně, …

- Prolínání polygonálních sítí (mesh blending) – Unity
  - Kombinace optimalizovaného terénu a dalších objektů
  - Video: https://www.youtube.com/watch?v=pv8wjMGGGDM
- Kombinace polygonální sítě + voxelové mřížky – CryEngine, Unity:
  - Video: https://www.youtube.com/watch?v=PhXIkbaVaj8

Literatura

- Knihy *Game Programming Gems*
- Knihy *GPU Gems* (2004 …)
- J. D. Foley: *Computer Graphics: Principles and Practice*. (1990)
- Watt – *3D Games:Volume 1: Real-Time Rendering and Software Technology* (2000)
- D. Luebke: *Level of Detail for 3D Graphics*, 2002
- …

- T. Mádr: *Tvorba herního charakteru, BP*
- R. Tisovčík:
  - *Generation and Visualization of Terrain in Virtual Environment,* BP
  - *Cartography in Virtual Environment,* DP

Literatura

- Články:
  - Jason Mitchell, Moby Francke, and Dhabih Eng. Illustrative rendering in *Team Fortress 2*. In*Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*(NPAR '07). ACM, New York, NY, USA, 71-76. 2007. DOI=http://dx.doi.org/10.1145/1274871.1274883