Antialiasing

Róbert Bohdal

robert.bohdal@fmph.uniba.sk

https://flurry.dg.fmph.uniba.sk/webog/bohdal-vyucba

Katedra algebry a geometrie
Fakulta matematiky fyziky a informatiky
Univerzita Komenského v Bratislave

Počítačová grafika (1)
prednáška č. 10

Obsah

Základné pojmy

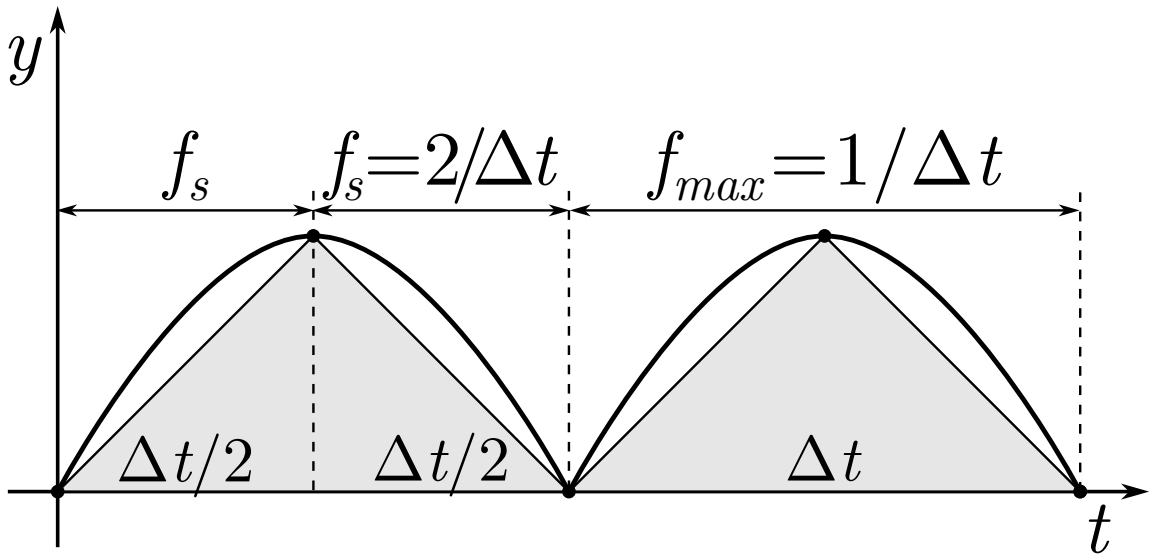
- Spojitý obraz v PG je definovaný spojitou funkciou $f(x, y)$ na definičnom obore $\langle x_{min}, y_{min}\rangle \times \langle x_{max}, y_{max}\rangle$ s oborom hodnôt v $\mathbb{R}^3$ (R,G,B), resp. v $\mathbb{R}^4$ (R,G,B,α)
- Pre zobrazenie, zaznamenanie či spracovanie obrazu je nutné spojitú funkciu diskretizovať v definičnom obore aj v obore hodnôt
- Diskretizácia v definičnom obore sa nazýva **vzorkovanie** (*sampling*)
- Diskretizácia v obore hodnôt sa nazýva **kvantovanie** (*quantization*)
- Počet vzoriek (v definičnom obore) udáva **rozlíšenie** a počet možných farieb (v obore hodnôt) udáva **farebnú hĺbku**
- Pixel s konkrétnou farbou je reprezentant jednej vybranej vzorky spojitého obrazu
- **Rekonštrukciou** obrazu/signálu rozumieme (znovu)vytvorenie spojitej funkcie, napr. pomocou interpolácie, zo vzoriek, ktoré vznikli diskretizáciou pôvodnej obrazovej funkcie

Vzorkovanie

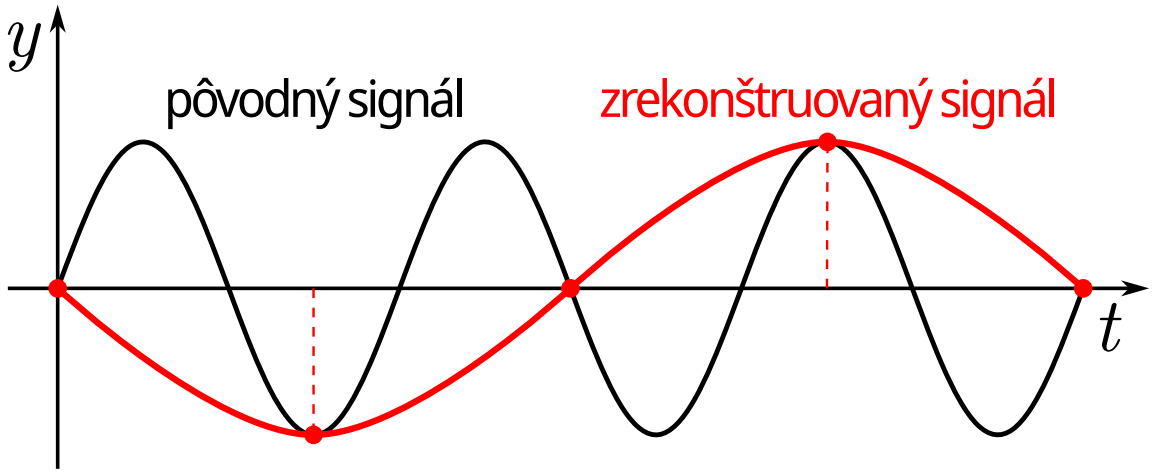
- Vzorkovanie v 1D s rovnomerným krokom (intervalom) o dĺžke Δx, so začiatkom v bode x_0: $I_i = f(x_0 + \Delta x \cdot i)$, pre $i = 0, 1, ..., n$
- Vzorkovanie v 2D: $I_{ij} = f(x_0 + \Delta x \cdot i, y_0 + \Delta y \cdot j)$, pre $i = 0, 1, ..., n$ a $j = 0, 1, ..., m$
- Vzorkovacia frekvencia v 1D je určená vzťahom $f_s = 1/\Delta x$
- V spojitom obraze je frekvencia (opakovanie určitého vzoru/signálu) neohraničená ($\Delta x \to 0$), avšak v diskrétnom obraze nekonečná nemôže byť, preto $f_s \ll \infty$
- Pri vzorkovaní dochádza k strate určitej informácie a pri veľkom kroku aj k strate „detailov"
- Chyba pri vzorkovaní obrazu s nedostatočnom frekvenciou (veľkosťou kroku Δx a Δy) sa prejavuje tzv. **aliasom**, ktorý je sprevádzaný artefaktmi
- Chyba pri kvantovaní sa sa prejavuje tzv. Machovým efektom, pri ktorom človek vníma hrany na nespojitom (skokovom) rozhraní medzi skupinami pixlov rôznych farieb

Vzorkovanie – Nyquistov limit, alias

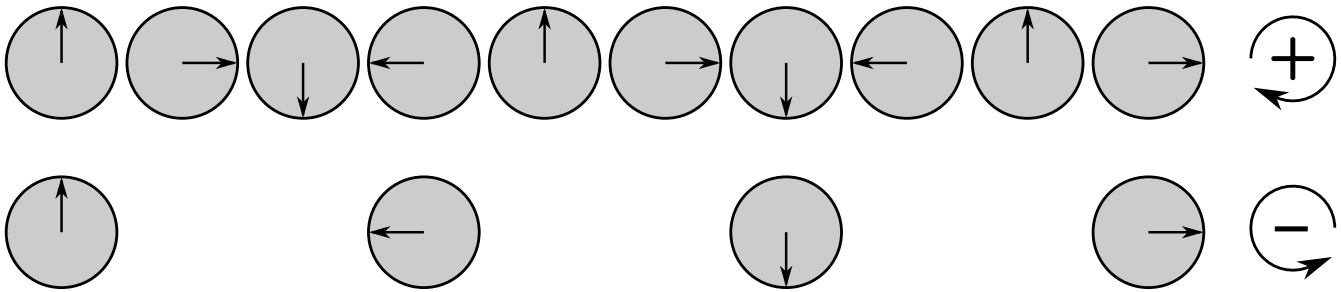
- Ak nechceme stratiť informáciu o „detailoch" signálu, musíme dodržať tzv. **Nyquistov limit**, čo znamená, že pre vzorkovaciu frekvenciu musí platiť $f_s > 2f_{max}$, kde $f_{max} = \max\{1/\Delta t_i; i = 1, \ldots, n\}$ je hodnota najvyššej frekvencie v signáli
- Alias vzniká v zrekonštruovanom signáli (obraze), ak bol vzorkovaný pod hodnotou Nyquistovho limitu, t.j. ak $f_s < 2f_{max}$
- Alias typicky vzniká pri zmenšovaní a opätovnom zväčšení obrazu bez použitia filtrovania. Napr. pri zmenšení textúry šachovnice sa stratia detaily (niektoré štvorčeky zmiznú), alebo sa posunú niektoré hrany



Vzorkovanie – alias

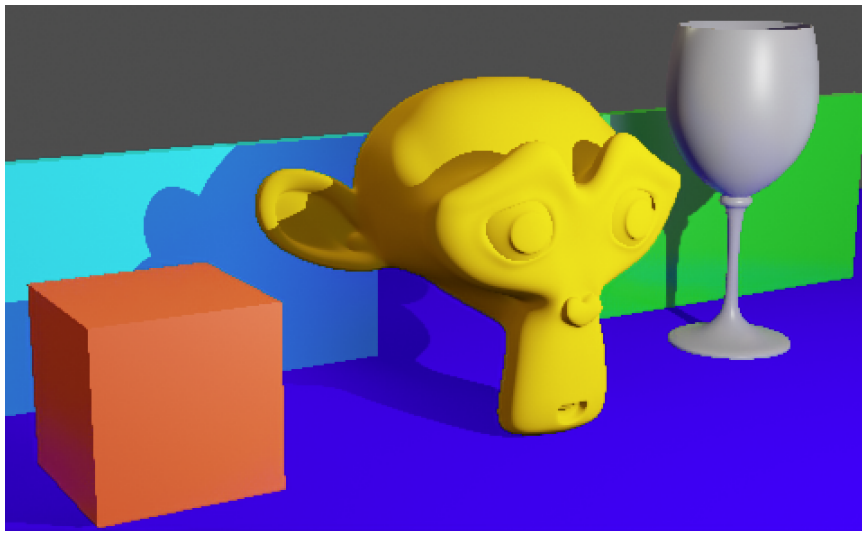


- Pri snímaní obrazu kamerou vzniká aj *temporálny* alias, ktorý sa prejavuje napr. zdanlivo opačným otáčaním kolies auta
- Aliasom (nepravým) nazývame aj zubatosť úsečiek, ktorá vzniká pri ich zobrazovaní v rastri. Nepravým ho nazývame preto, že nemá súvis so vzorkovacou frekvenciou a stratou detailov

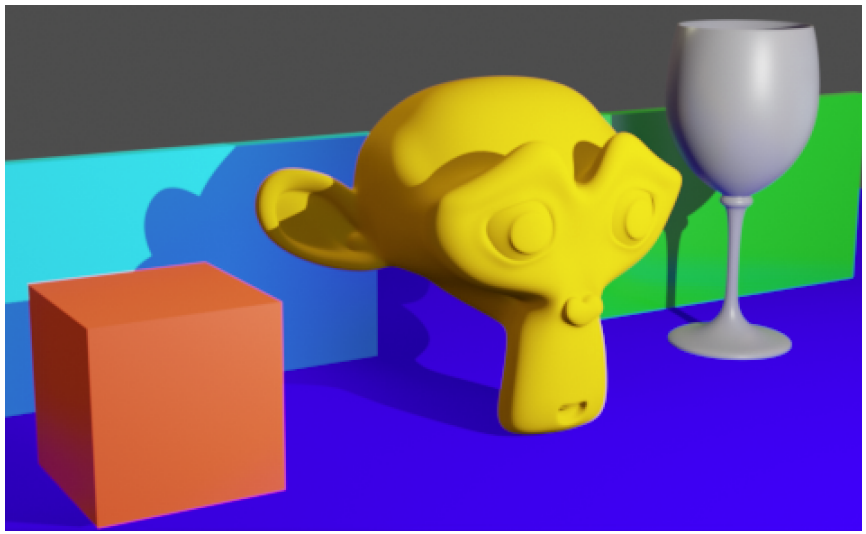


Odstraňovanie aliasu

- Pod antialiasingom rozumieme odstraňovanie aliasu
- Často sa alias neodstráni, len sa zmenší jeho viditeľnosť
- Metódy odstraňovania aliasu:
  - filtrovanie pred vzorkovaním (*prefiltering*),
  - vzorkovanie plochy (pixelu) (*area sampling*),
  - nadvzorkovanie (*supersampling*),
  - filtrovanie po vzorkovaní (*postfiltering*)



Obrázok s viditeľným aliasom



Odstránenie aliasu nadvzorkovaním

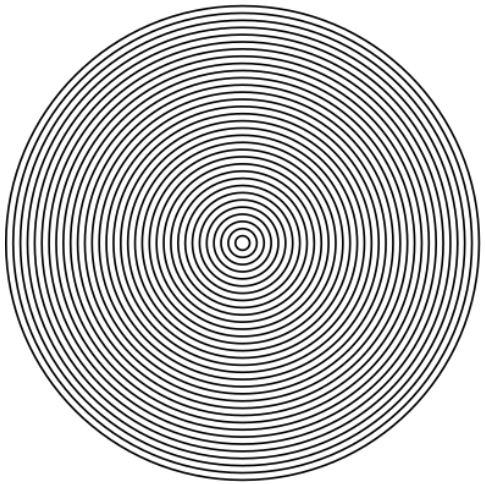
Filtrovanie pred vzorkovaním

- Filtrovanie pred vzorkovaním najčastejšie používa filter rozostrenia (*blur filter*), čím sa z obrazu odstránia vysoké frekvencie (detaily)
- Nevýhodou tejto metódy je redukcia detailov, ktorá spôsobuje vizuálne zníženie rozlíšenia obrazu

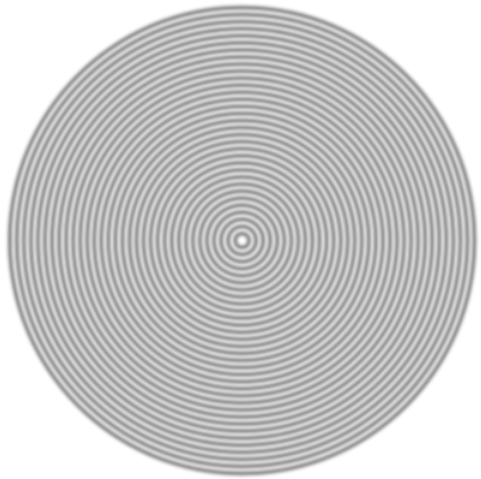
Originálny obrázok

Preškálovaný obrázok bez filtru

Preškálovaný obrázok s filtrom



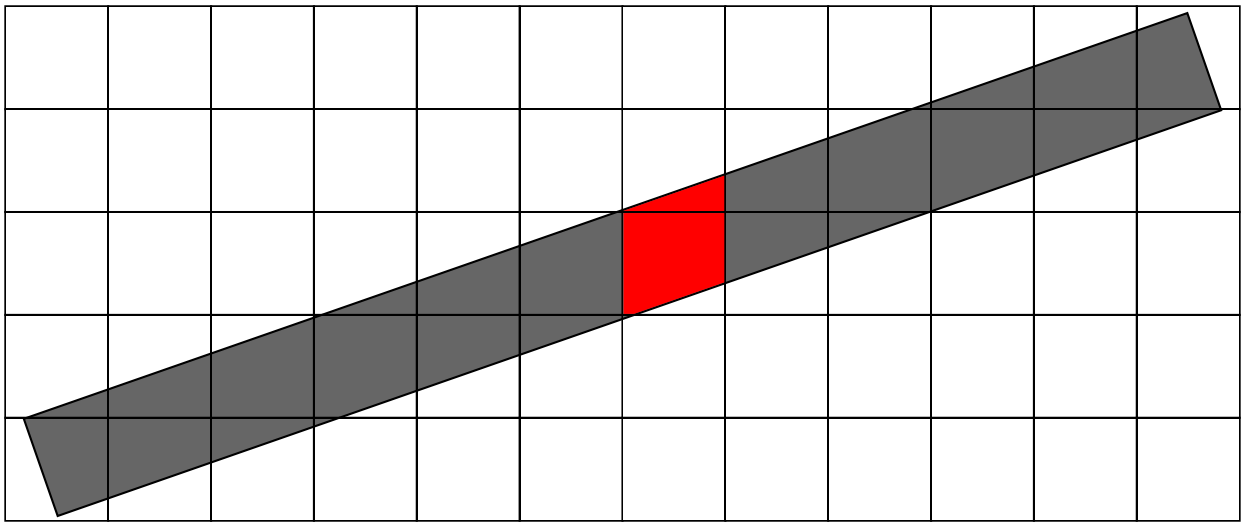
Originálny obrázok



Filtrovaný obrázok

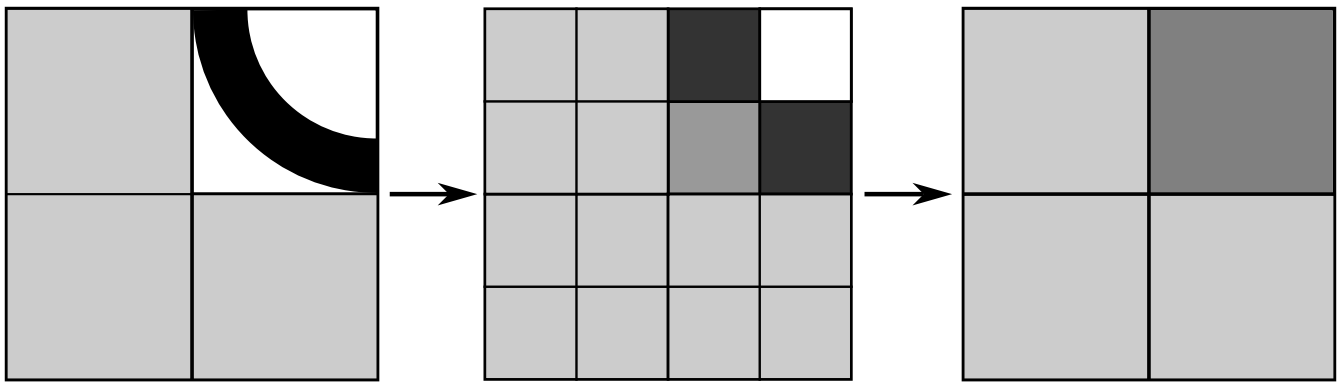
Vzorkovanie plochy

- Vzorkovanie plochy (pixlu) využíva vlastnosť, že pixel má štvorcový tvar, pričom jeho časť alebo aj celý pixel je pokrytý časťou zobrazovaného objektu
- Intenzita pixlu sa vypočíta podľa plochy, ktorú zobrazovaný objekt v pixeli zaberá, t.j. ak objekt zaberá 1/4 plochy, tak pixel bude mať 1/4 intenzity
- Výsledná farba pixlu je súčtom príspevkov všetkých tých častí objektov, ktoré majú s daným pixlom prienik a sú v pixeli viditeľné
- V praktickej realizácii sa využíva akumulačný a hĺbkový (*depth*) buffer
- Metóda je výpočtovo náročná, je nutné počítať preniky objektov s pixlami
- Pri využití aproximácie prienikov sa použijú preddefinované útvary v podobe trojuholníkov a lichobežníkov s konkrétnymi veľkosťami strán



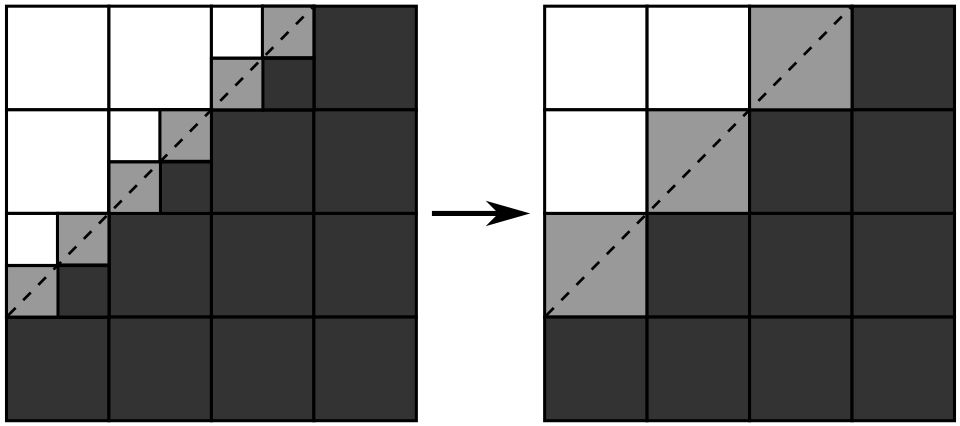
Nadvzorkovanie

- Pri mriežkovom uniformovanom nadvzorkovaní sa každý pixel rozdelí na n^2 ($n = 2, 3, ...$) subpixlov, čím sa značne zvýši pamäťová aj výpočtová náročnosť, keďže sa rasterizácia vykonáva vo väčšom (jemnejšom) rozlíšení
- Okrem štandardného mriežkového usporiadania subpixlov sa využíva aj usporiadanie vedľa seba pre 2x (vodorovne, zvislo, diagonálne), do šachovnice či n-veží pre 4x, 8x a 16x
- Pri neuniformovanom nadvzorkovaní sa pixle rozdeľujú iba tam, kde susedné pixle majú odlišnú intenzitu (farbu), napr. na hranách objektov
- Keďže sa v oboch typoch nadvzorkovania vytvára obraz vo vyššom rozlíšení, treba ho prepočítať späť na pôvodné rozlíšenie



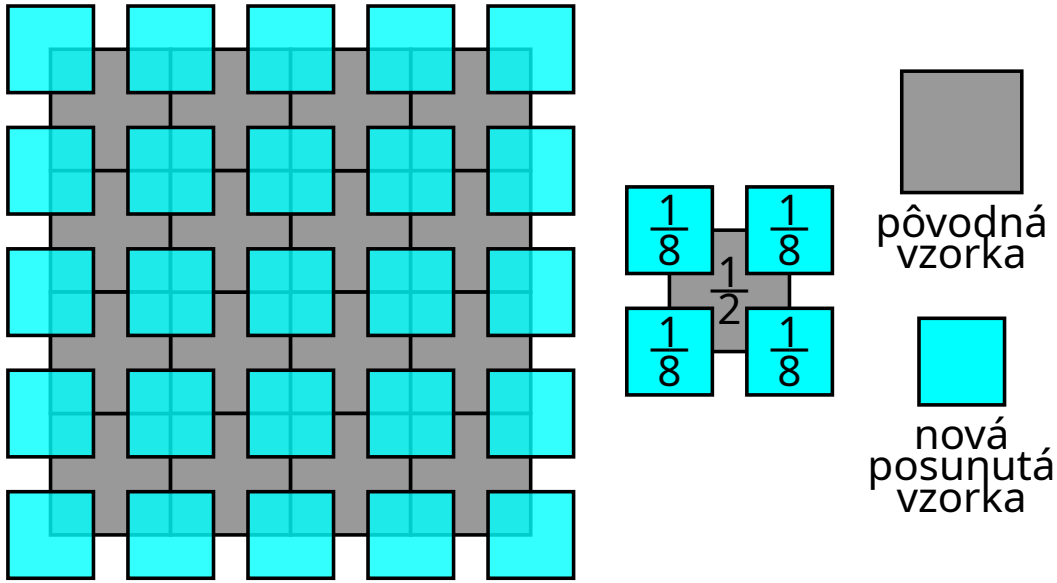
Nadvzorkovanie

- Pri prepočítaní výslednej farby pixla v nižšom rozlíšení sa používa aritmetický alebo vážený priemer zodpovedajúcich subpixlov
- Pri počítaní váženého priemeru sa používa štandardný rozostrujúci filter (napr. Gaussov), keďže však malé objekty môžu zaniknúť, používajú sa aj iné filtre
- Nadvzorkovanie obyčajne alias neodstráni, ale ho iba posúva k vyšším frekvenciám, ktoré si ľudský mozog už nevšimne



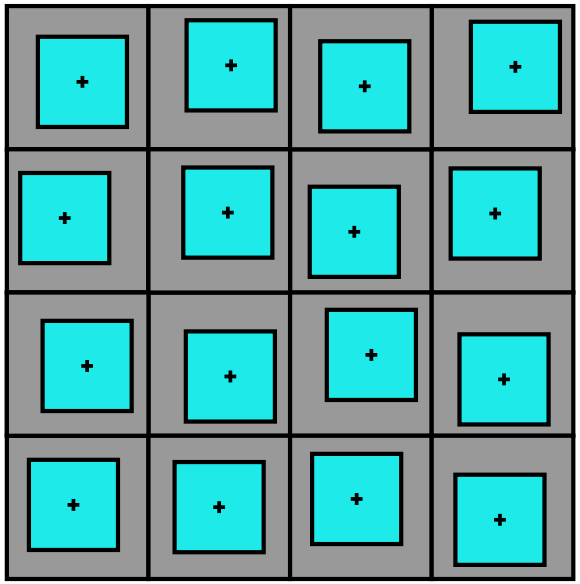
Nadvzorkovanie metódou quincunx

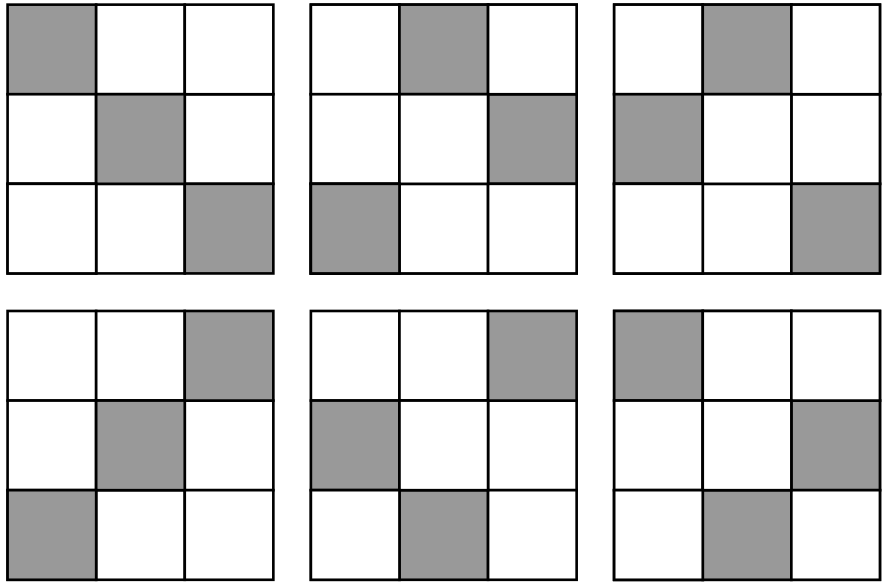
- Používa 5 „subpixlov" na 1 pixel, pričom stredový subpixel má váhu 1/2 a rohové subpixle majú váhu 1/8
- Rohové subpixle zdieľa so susednými pixlami, preto sa počítajú iba dva obrazy
- Vizuálne sa tento typ antialiasingu podobá náročnejšiemu nadvzorkovaniu 4x
- Obraz sa vykreslí (renderuje) do dvoch rastrov v štandardnom rozlíšení, pričom pri vykresľovaní do druhého rastra sa posunie kamera v priestore obrazovky o vektor (1/2, 1/2). Použitím váženého priemeru sa potom z dvoch rastrov (bitmáp) vypočíta výsledný obraz



Stochastické nadvzorkovanie

- Náhodným posunutím sa vysoké frekvencie zmenia na šum, ktorý ak je dostatočne malý, tak ho ľudský mozog nevníma, keďže svetlocitlivé bunky sú usporiadané ako vzorkovanie Poissonovho disku (čiastočne náhodne)
- Metóda roztrasením (*jittering*) využíva uniformované rozdelenie pixlov na subpixle, avšak stredy subpixlov sú posunuté o náhodný vektor s dĺžkou maximálne 1/2 veľkosti subpixla
- Metóda *n*-veží (*n*-rooks) vyberá náhodné rozloženie vzorkovaných subpixlov tak, aby sa subpixle v pixeli „neohrozovali", podobne ako veže na šachovnici



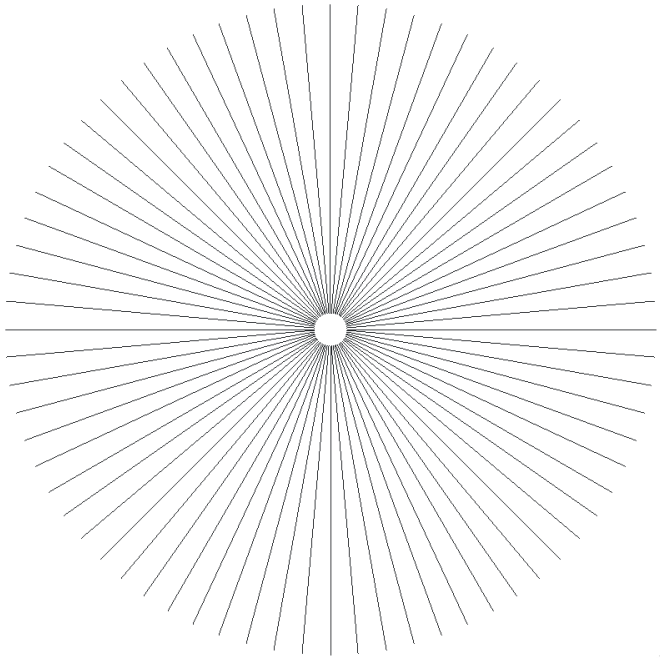


Filtrovanie po vzorkovaní

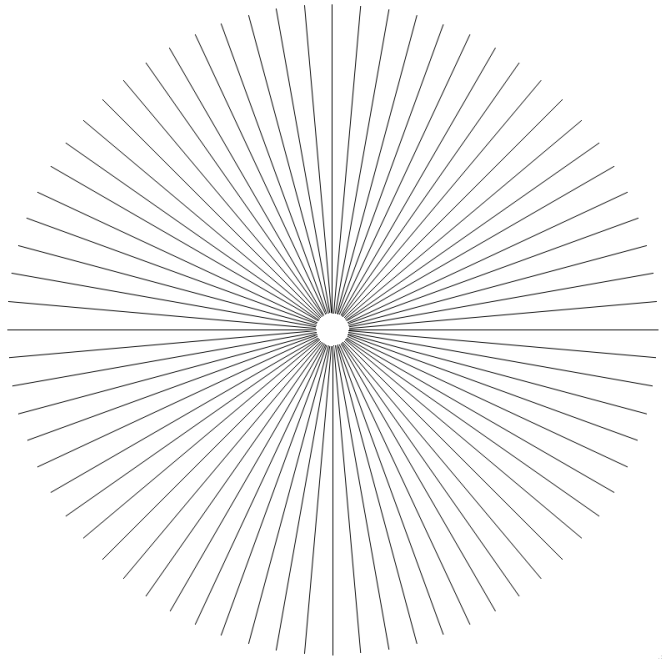
- Využívajú rôzne, najmä rozostrujúce filtre, ktoré sa použijú, až keď je obraz vyrenderovaný
- Cieľom filtrovania je vyhladiť „zubaté okraje" hrán zobrazovaných objektov
- Ako filter môže byť použitý filter s aritmetickým priemerom (box) alebo s váženým priemerom (kubický, kvadratický, Gaussov), či iné druhy filtrov (Catmull-Rom, Mitchell-Netravali, atď.), ktoré obsahujú aj záporné hodnoty váh, spôsobujúce istú formu zaostrenia niektorých pixlov obrazu, aby sa predišlo prílišnému rozostrovaniu hrán, či malých objektov

Odstraňovanie nepravého aliasu pre úsečky a krivky

- Pri klasickom vykresľovaní kriviek do rastra vzniká známa zubatosť a moiré
- Môže sa použiť nadvzorkovanie, pričom sa do jemnejšieho rastra vykresľuje n násobne hrubšia čiara. Výsledná intenzita sa potom vypočíta podľa počtu vykreslených subpixlov
- Pri jednofarebnom pozadí je rýchlejšie a pamäťovo výhodnejšie použiť napr. modifikáciu Bresenhamovho algoritmu, pri ktorej sa okrem štandardne vykresľovaných pixlov počítajú aj relevantné susedné pixle



Úsečky s viditeľným aliasom



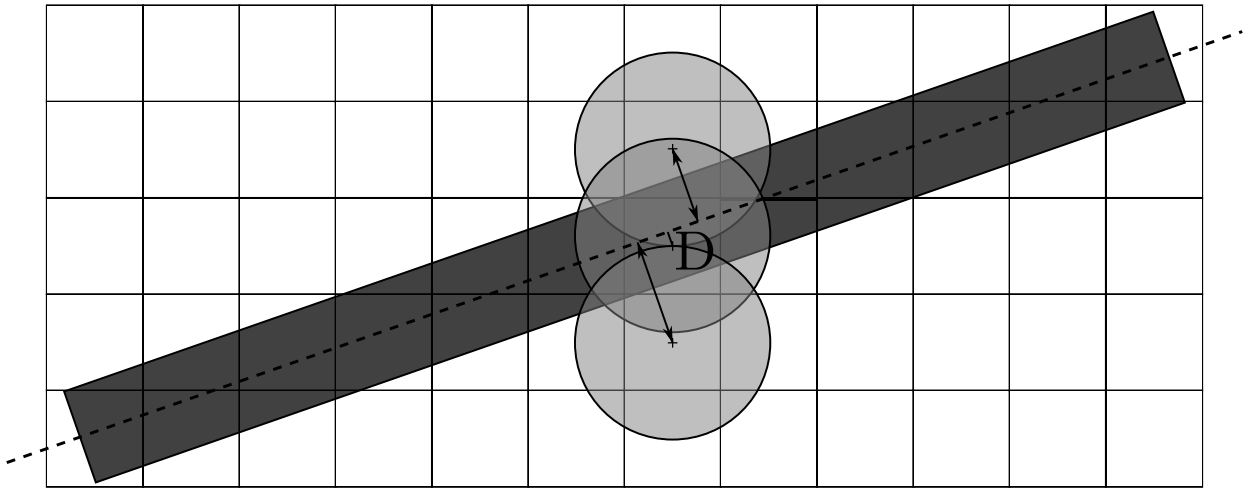
Úsečky vykreslené s antialiasingom

Metódy vzorkovania plochy

- Nevážené vzorkovanie plochy využíva namiesto úsečky obdĺžnik s hrúbkou 1 pixlu a výsledná intenzita farby vykresľovaného pixlu je určená plochou časti obdĺžnika, ktorá pokrýva daný pixel
- Vážené vzorkovanie plochy dávajú väčšiu váhu oblastiam, ktoré sú bližšie k úsečke. Ten to druh vzorkovania využíva buď vhodnú váhovú funkciu (napr. vzdialenosť k úsečke) alebo nejaký druh filtra (napr. kužeľový)

Metóda Gupta-Sproull pre kreslenie úsečky

- Gupta a Sproull použili pri kreslení úsečky vyhladzovací filter s kužeľovou funkciou rozloženia bodov o polomere jedného pixlu, pričom jeho konvolúcia s priamkou je závislá iba od vzdialenosti kužeľa k priamke
- Intenzita zobrazovaného pixlu je úmerná objemu časti kužeľa, ktorý prekrýva rastrovanú úsečku
- Tento objem je vypočítaný pomocou vzdialenosti D stredu pixlu ku priamke reprezentujúcej danú úsečku
- Vzdialenosť D sa v algoritme vykresľovania úsečky, kvôli efektivite, počíta inkrementálne
- Pre vykreslenie bodu úsečky používa 3 pixle, pre hrubšie úsečky viac pixlov

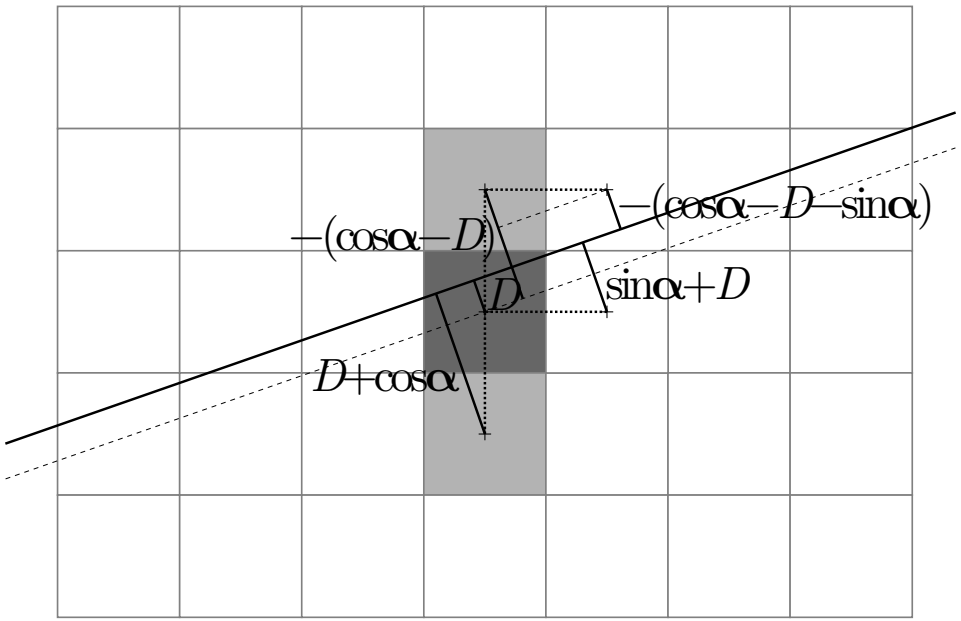


Chenova varianta metódy Gupta-Sproull pre 1. oktant

$$\cos\alpha = \frac{\Delta x}{\sqrt{\Delta x^2+\Delta y^2}},\ \sin\alpha = \frac{\Delta y}{\sqrt{\Delta x^2+\Delta y^2}}$$

$$D_{curr}(y-1) = D + \cos\alpha$$
$$D_{curr}(y+1) = -(\cos\alpha - D) = D - \cos\alpha$$

$$D_{next}(y) = D + \sin\alpha$$
$$D_{next}(y+1) = -(\cos\alpha - D - \sin\alpha) = D + \sin\alpha - \cos\alpha$$



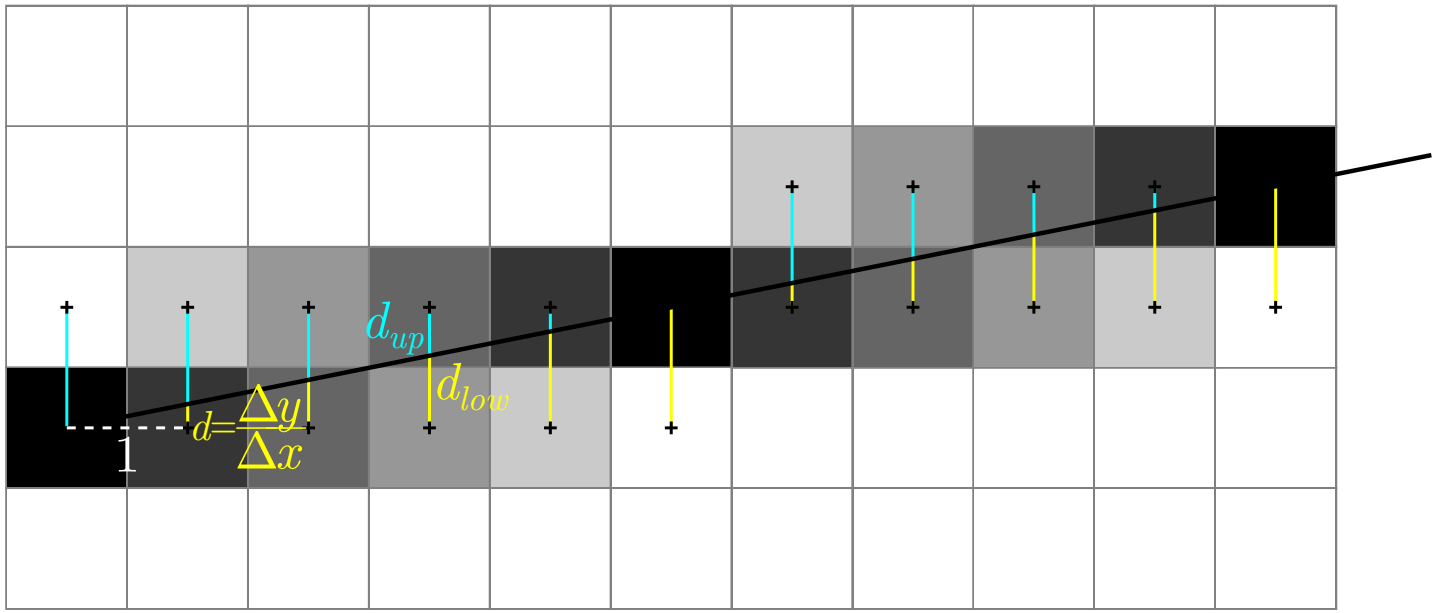
Chenova varianta algoritmu Gupta-Sproull pre 1. oktant

```
procedure draw_line(ax, ay, bx, by: integer);
x, y, deltax, deltay, d: integer;
D, length, sin, cos: real;
begin
  x := xa; y := ya;
  deltax := xb - xa;
  deltay := yb - ya;
  d := 2 * deltay - deltax;
  D := 0;
  length := sqrt(deltax * deltax + deltay * deltay);
  sin := deltay / length;
  cos := deltax / length;
  while x <= xb do begin
    drawpixeli(x, y - 1, D + cos);
    drawpixeli(x, y, D);
    drawpixeli(x, y + 1, D - cos);
    x := x + 1;
    if d <= 0 then begin
      D := D + sin;
      d := d + 2 * deltay;
    end
    else begin
      D := D + sin - cos;
      d := d + 2 * (deltay - deltax);
      y := y + 1;
    end
  end
end
```

Wuova metóda pre kreslenie úsečky

- Jednoduchšia a efektívnejšia metóda ako Gupta-Sproull
- Založená na úprave Bresenhamovho alebo DDA algoritmu kreslenia úsečky
- Vykresľuje dva pixle, jeden pod a druhý nad „geometrickým" obrazom úsečky
- Intenzita pixlu je daná jeho y-ovou vzdialenosťou od úsečky

$$y = mx + b = \frac{\Delta y}{\Delta x}x + b$$
$$d_{next}(y) = d + \frac{\Delta y}{\Delta x}$$
$$d_{low} = d,\ d_{up} = 1 - d_{low}$$



Bresenhamov variant Wuovho algoritmu pre úsečku v 1. oktante

```
procedure draw_line(ax, ay, bx, by: integer);
x, y, deltax, deltay, dy: integer;
low: real;
begin
  x := xa; y := ya;
  deltax := xb - xa;
  deltay := yb - ya;
  dy := 0;
  while x <= xb do begin
    low := dy / deltax;
    drawpixeli(x, y, low);
    drawpixeli(x, y + 1, 1 - low);
    dy := dy + deltay;
    if dy > deltax then begin
      dy := dy - deltax;
      y := y + 1;
    end
    x := x + 1;
  end
end
```

Antialiasing (vyhladzovanie) v počítačových hrách

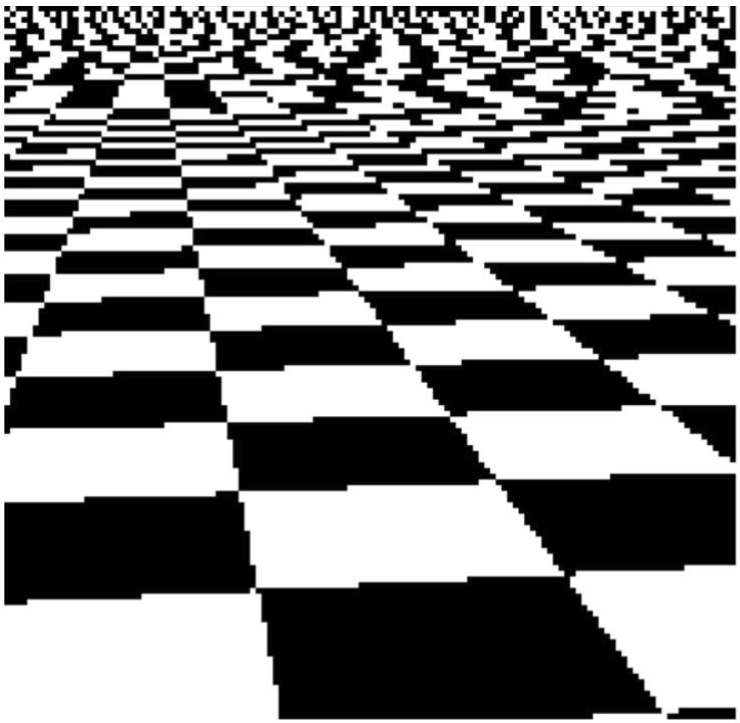
- Existuje viacero rôznych techník vyhladzovania, avšak využívajú sa najmä dva prístupy, včítane ich kombinovania a rôznych modifikácií:
  - nadvzorkovanie – počítajú obraz vo vyššom rozlíšení a potom ho prepočítajú na pôvodné rozlíšenie s využitím priemeru
  - filtrovanie – na vypočítaný obraz sa aplikuje vhodný filter, ktorý mierne rozostruje okolie hrán
- Vyhladzovanie sa vykonáva na celom zobrazovanom obraze, t.j. antialiasing je celoobrazovkový (FSAA – *Full Screen Anti-Aliasing*)

Typy antialiasingu v počítačových hrách

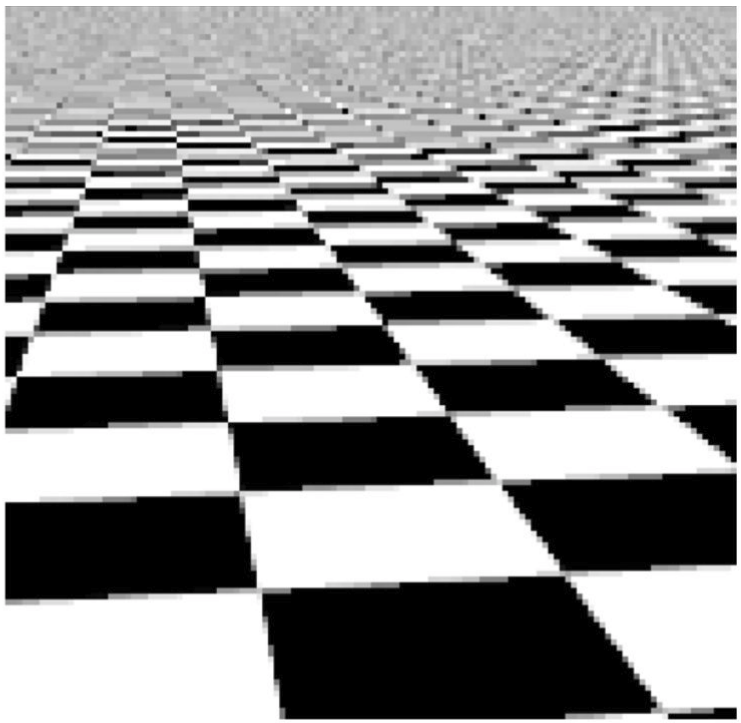
- Supersample Anti-aliasing (SSAA)
- Multisample Anti-aliasing (MSAA)
- Fast Approximate Anti-aliasing (FXAA)
- Temporal Anti-aliasing (TAA)
- Enhanced Subpixel Morphological Anti-aliasing (SMAA)
- Deep Learning Super-Sampling (DLSS)

Supersample (nadvzorkovací) antialiasing – SSAA

- Základna forma antialiasingu, pri ktorej sa renderuje obraz s vyšším rozlíšením
- Napr. pri SSAA 4x (2×2) sa na jeden pixel použijú 4 subpixle, vypočíta sa obraz pre jemnejšie rozlíšenie a potom sa váženým priemerom vypočíta zo 4 pixlov hodnota výsledného zobrazovaného pixlu
- SSAA dáva pri väčšom počte subpixelov najlepšie výsledky vyhladzovania, ale je to výpočtovo najnáročnejšia metóda



single sampling



32x32 supersampling

Multisample (viacvzorkový) antialiasing – MSAA

- Objekty sa vykresľujú ako polygonálna sieť, pričom MSAA prebieha počas rasterizácie (pri rasterizácii trojuholníkov na pixely)
- Na rozdiel od SSAA je výpočet farby a testovanie viditeľnosti oddelené. Pixel shader sa typicky spúšťa iba raz pre každý pixel (v jeho strede), zatiaľ čo depth test sa vykonáva pre každú subpixelovú vzorku
- Rasterizér generuje masku pokrytia (*coverage mask*), ktorá určuje, ktoré subpixely daného pixelu sú pokryté práve rasterizovaným trojuholníkom
- Výsledná farba, získaná z pixel shaderu, sa skopíruje do tých subpixelových vzoriek, ktoré sú v maske pokrytia (t.j. daný subpixel leží v práve vykresľovanom trojuholníku) a zároveň prešli hĺbkovým testom (sú viditeľné)
- Na konci vykresľovania sa subpixelové vzorky spriemerujú (podvzorkovanie), čím vznikne vizuálne hladká hrana na okrajoch objektov
- Táto metóda je výpočtovo menej náročná než SSAA, no má vysoké nároky na pamäť (VRAM), keďže buffer pre hĺbku aj farbu musia mať rozlíšenie na úrovni subpixelov

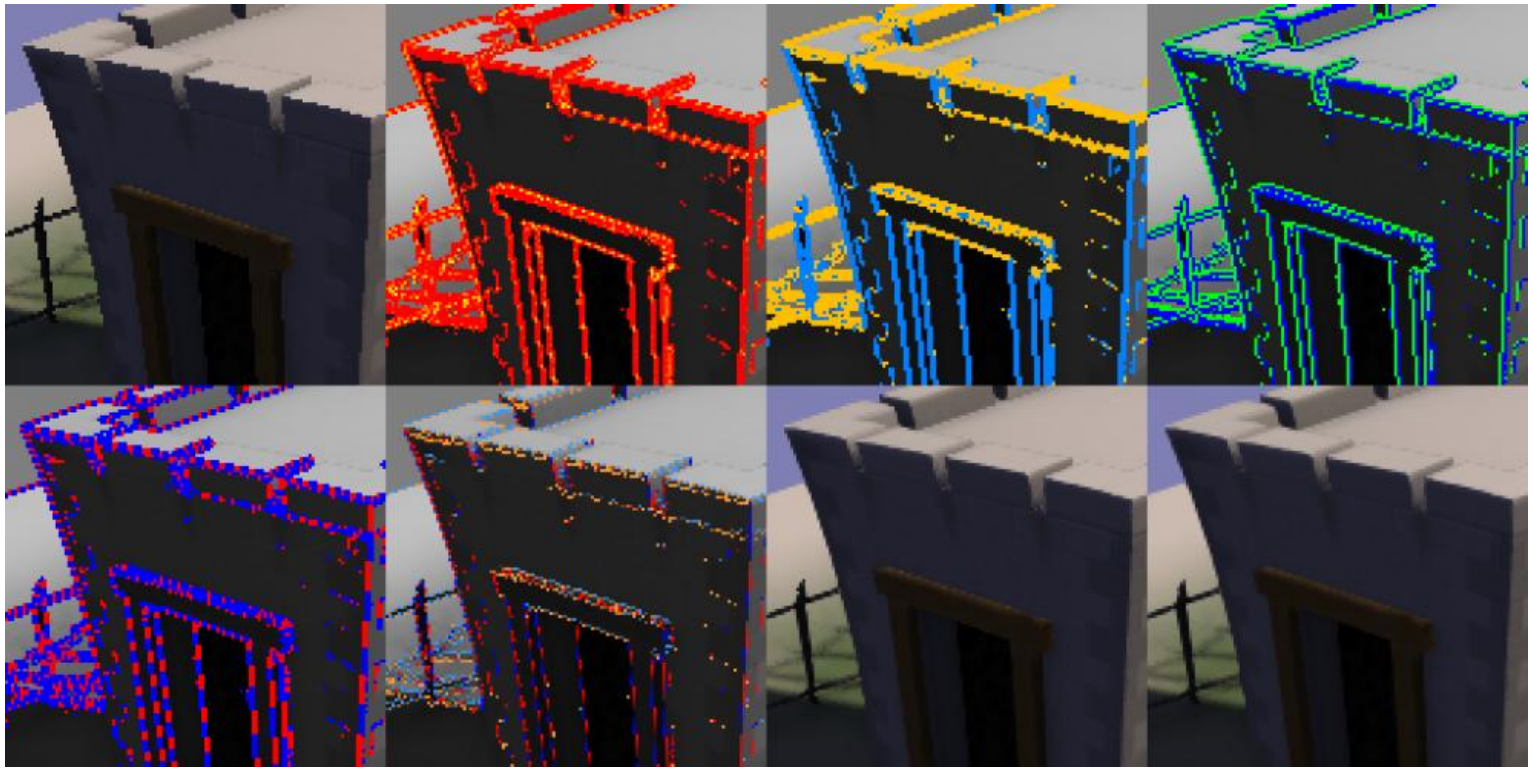
Fast approximate (aproximačný) antialiasing – FXAA

- Bol vyvinutý firmou Nvidia a predstavuje typické filtrovanie obrazu, ktoré je vykonávané ako postproces, až po výpočte pixlov obrazu
- Pri detekcii hrán porovnáva intenzitu susedných pixlov
- Tento druh AA je oveľa menej náročný ako SSAA a MSAA, avšak na úkor kvality obrazu

Algoritmus pozostáva z nasledujúcich krokov:

1. RGB hodnoty pixlov obrazu sa skonvertujú na hodnoty intenzít
2. Z 3×3 okolia sa odhadne rozsah intenzít; ak je pod prahom pixel sa nefiltruje
3. Vypočíta sa *edge amount* pre vertikálne/horizontálne smery z vážených hodnôt lineárnych kombinácií intenzít v okolí 3×3, a určí sa, či ide o horizontálnu alebo vertikálnu hranu
4. Vyberie sa dvojica susedných pixlov s najväčším kontrastom, ktorá leží kolmo na zistený smer hrany. Tento pár slúži na určenie, ako sa bude hrana vyhladzovať
5. Pozdĺž zistenej hrany sa v oboch smeroch (dopredu aj dozadu) postupne kontrolujú body, kým sa nedosiahne koniec hrany, t.j. miesto, kde sa priemerná intenzita vybranej dvojice (kolmej na smer hrany) výrazne zmení
6. Vypočíta sub-pixelový posun kolmo na hranu (na základe polohy pixelu medzi začiatkom a koncom hrany), ktorý určuje, odkiaľ sa bude obraz vzorkovať na vyhladenie hrany
7. Obraz sa znovu vzorkuje na posunutej pozícii a aplikuje sa dolno-priepustný filter
8. Pôvodná farba pixelu sa zmieša s filtrovanou, aby sa znížil alias

Fast approximate (aproximačný) antialiasing



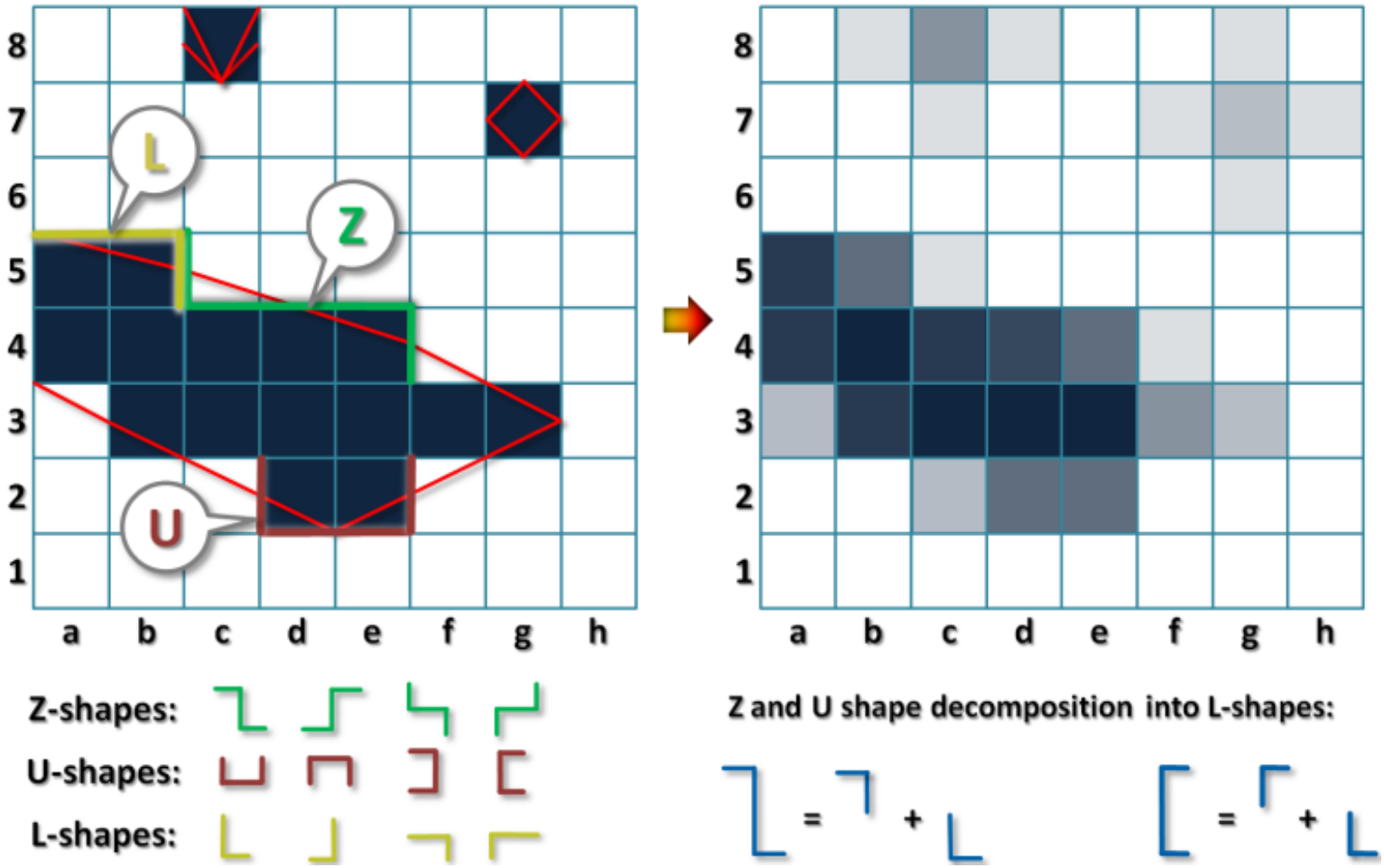
Temporal (časový) antialiasing – TAA

- Ide o techniku vyhladzovania hrán realizovanú v rámci postprocesu
- Využíva fakt, že farby objektov sa medzi susednými snímkami (*frames*) zvyčajne veľmi nemenia
- Ak scéna nie je statická a kamera alebo objekty sa hýbu, susedné snímky zobrazujú ten istý objekt, len mierne posunutý
- Algoritmus používa predchádzajúcu snímku (a niekedy aj viac minulých) spolu s aktuálnou snímkou na výpočet novej hodnoty každého pixelu. Hodnoty sa kombinujú pomocou váženého priemeru (filter)
- Pri nesprávnom nastavení alebo rýchlom pohybe môže dochádzať k vzniku duchov (*ghosting artefacts*) na hranách objektov
- Pri implementácií TAA sa používajú vektory pohybu (*motion vectors*), aby sa vedelo, kam sa zhluky pixlov presunuli medzi snímkami

Enhanced subpixel morphological antialiasing – SMAA

- Predstavuje kombináciu viacerých metód antialiasingu
- V niečom je podobný FXAA, ale používa robustnejšiu klasifikáciu hrán – nielen zvislé a vodorovné, ale aj šikmé hrany
- Rozširuje štandardný morfologický antialiasing (MLAA), ktorý vo výslednom obraze hľadá hranice objektov v ich pôvodnej (nerasterizovanej) forme. MLAA v nich potom nachádza špecifické vzory (tvary), ktoré využije pri vytvorení vhodného váženého priemeru okolitých pixlov
- Oproti FXAA, nadvzorkovanie metódou MSAA kombinuje s TSAA (voliteľne), aby sa obmedzilo „blikanie" pixlov na hranách zobrazovaných objektov počas animácie
- SMAA poskytuje lepší pomer kvalita/výkon než FXAA, najmä v statických scénach

Morphological (morfologický) antialiasing – MLAA



Deep Learning supersampling – DLSS

- Metóda bola vyvinutá spoločnosťou Nvidia pre jej grafické karty radu RTX
- Využíva neurónovú sieť na rekonštrukciu obrazu, čím zobrazuje vylepšený obraz alebo obraz vo vyššom rozlíšení
- Funguje podobne ako TAA, ale netestuje každý pixel v každej snímke. Namiesto toho vzorkuje rôzne pixely v rôznych snímkach a používa pixely z minulých snímok na doplnenie chýbajúcich detailov v aktuálnej snímke
- DLSS 3.0 pridáva vkladanie novej snímky medzi dve nasledujúce snímky, aby sa medzi nimi vytvoril plynulejší prechod. Využíva sa pri tom tzv. *Optical Flow Accelerator*
- Vo verzii DLSS 3.5 bola pridaná metóda rekonštrukcie lúčov založená na AI, ktorá sa používa na vylepšenie „zašumených" renderovaných obrázkov, ktoré často vznikajú po použití ray tracingu s nedostatočným počtom lúčov na pixel