

Tekstury

Opracowanie:

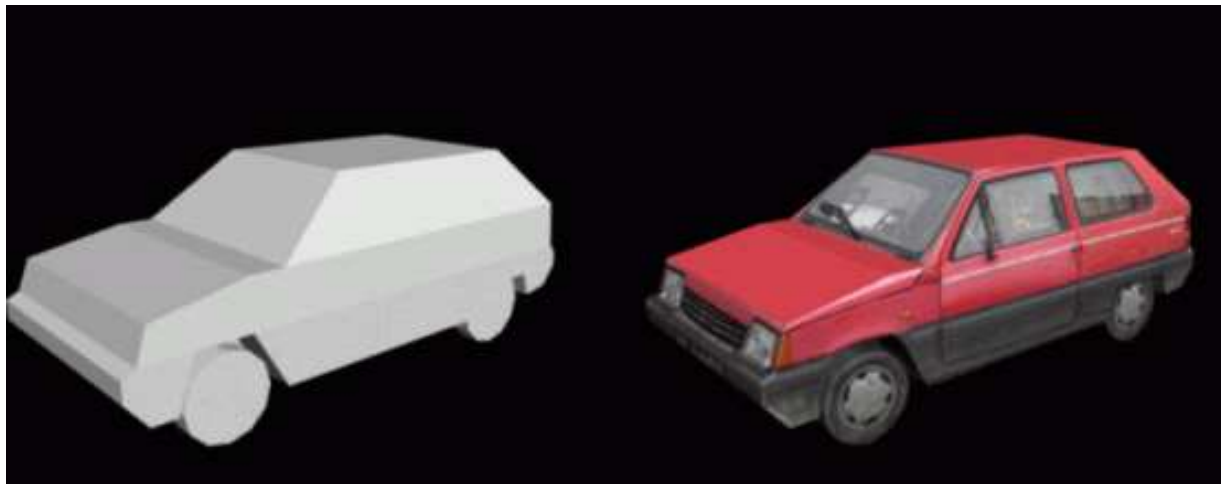
dr inż. Grzegorz Szwoch

Politechnika Gdańska

Katedra Systemów Multimedialnych

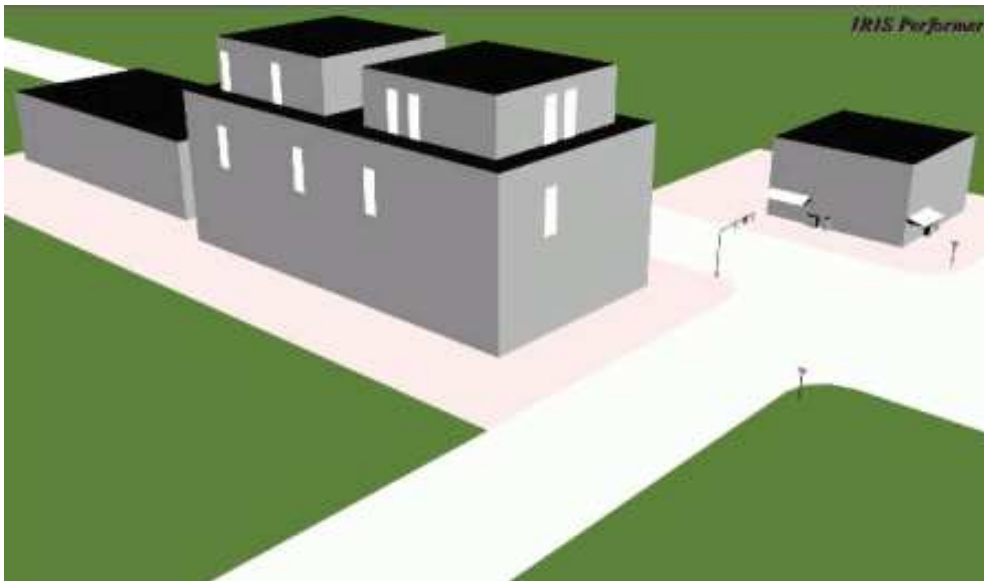
Tekstura

- **Tekstura** (*texture*) – obraz rastrowy (mapa bitowa, *bitmap*) nakładany na powierzchnię modelu obiektu.
- Pozwala uwidocznić szczegóły obiektu.
- Wartości pobrane z tekstury są uwzględniane w procesie cieniowania.



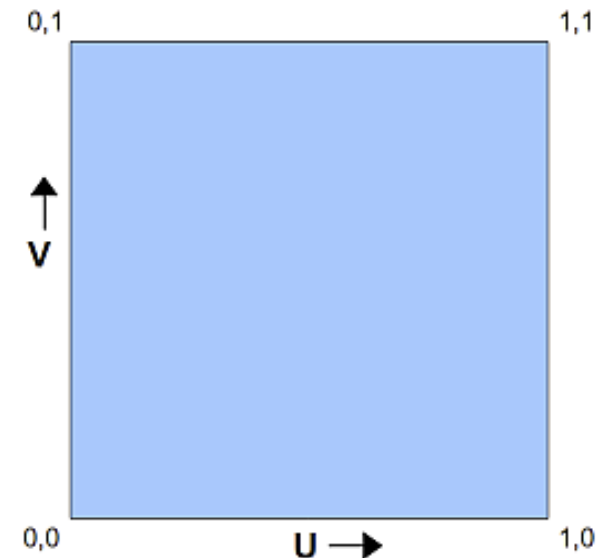
Tekstury

Przykład - scena bez teksturowania
(tylko cieniowanie) oraz z nałożeniem tekstur



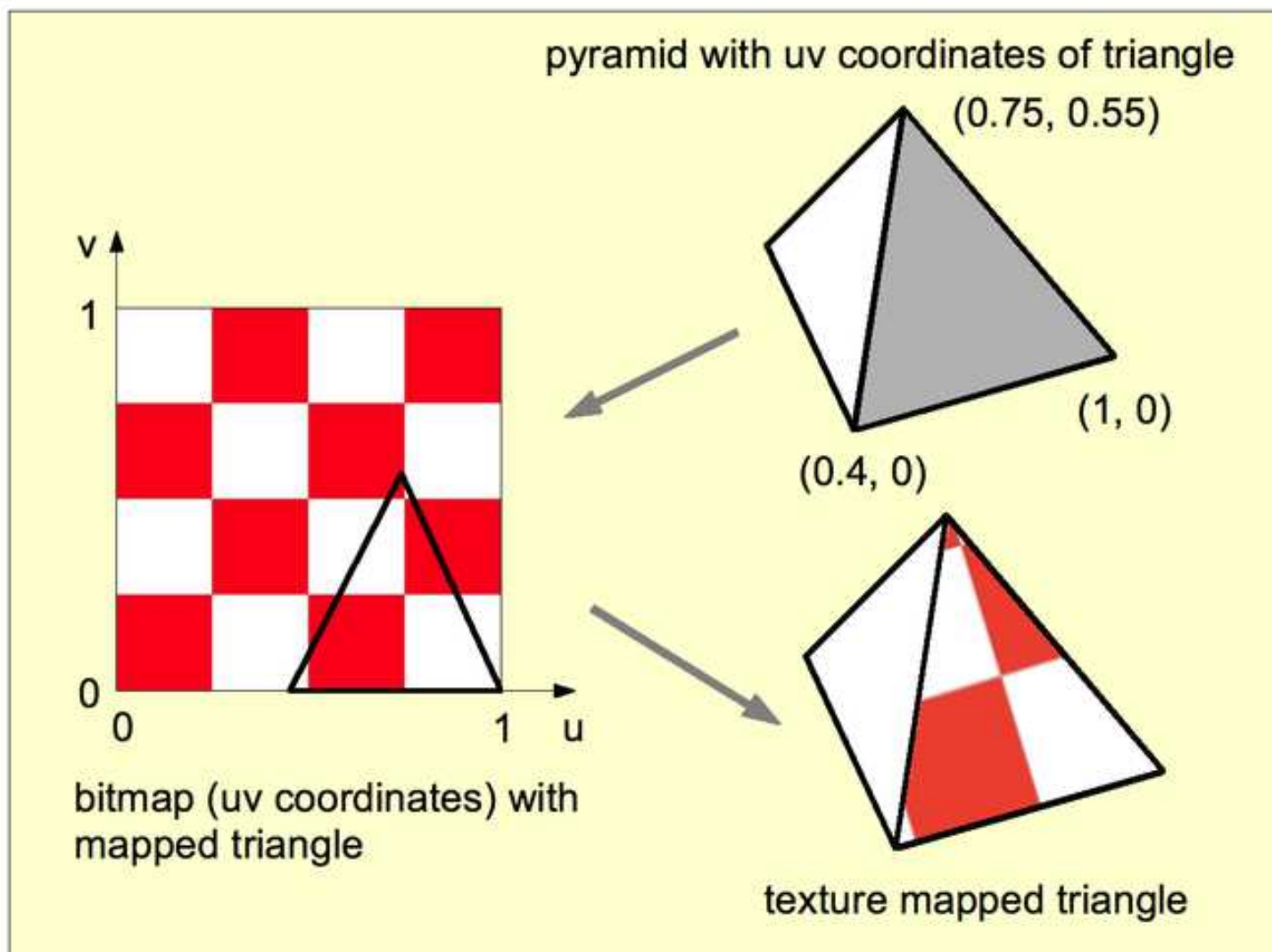
Odwzorowanie tekstury

- Teksel (*texel*) – piksel tekstury.
- Współrzędne **UV** tekstury – unormowane do zakresu od 0 do 1.
- **Odwzorowanie tekstury** (*texture mapping*) – przypisanie współrzędnych UV tekstury do współrzędnych XYZ modelu.
- W przypadku siatki trójkątowej: wsp. UV zapisywane są w werteksach siatki.
- Inna nazwa: **mapowanie UV**.



Odwzorowanie tekstury

Przykład:



Odwzorowanie tekstury

Podczas modelowania obiektu, grafik „maluje” teksturę na rozłożonej siatce obiektu i tworzy mapowanie współrzędnych.



Odwzorowanie tekstury

Można stosować współrzędne UV spoza zakresu $(0, 1)$. Znaczenie zależy od wybranego trybu.

Np. 1.2 może oznaczać to samo, co:

- 1.0 (ograniczenie),
- 0.2 (powtórzenie, $1.2 - 1.0$)
- 0.8 (odbicie, $1.0 - (1.2 - 1.0)$)

Próbkowanie tekstury

- Współrzędne UV tekstury są zapisane w wierzchołkach siatki modelu.
- Rasteryzer **interpoluje** je dla wybranego fragmentu wewnątrz trójkąta.
- Tekstura jest **próbkowana** w miejscu wskazanym przez interpolowane współrzędne.
- Wykonuje to obiekt nazywany **samplerem**.
- Pobrana z tekstury barwa jest stosowana w procesie cieniowania – łączona z barwą światła i obiektu.

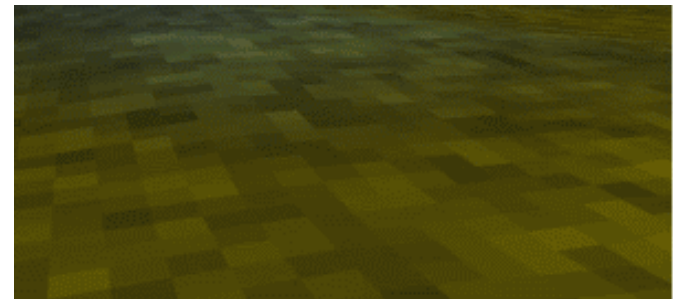
Próbkowanie tekstury

Przykład:

- Tekstura o wymiarach 512x512 pikseli.
- Z interpolacji wychodzą współrzędne $UV = (0.34, 0.78)$.
- Przeliczenie na współrzędne pikseli: $(174.08, 399.36)$.
- Jak zinterpretować „ułamkowe” piksele?
- Najprostsze podejście: zaokrąglić $(174, 399)$.

Próbkowanie punktowe

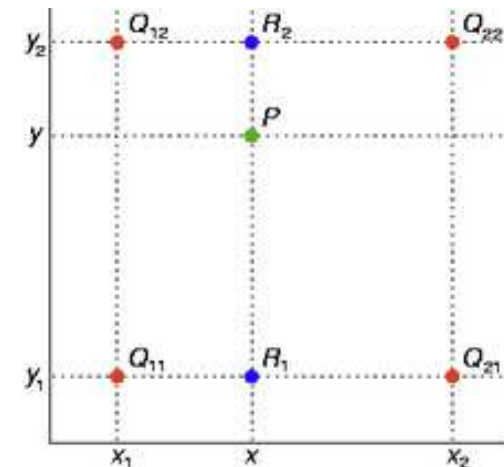
- Najprostsze podejście – zaokrąglenie do najbliższego teksela – **próbkowanie punktowe** (*point sampling*).
- Inaczej: **metoda najbliższego sąsiada** (*nearest neighbour*).
- Wada: może się zdarzyć, że wiele fragmentów spróbkuje ten sam teksel.
- Efekt: zblokowanie pikseli („pikseloza”).



Filtracja dwuliniowa

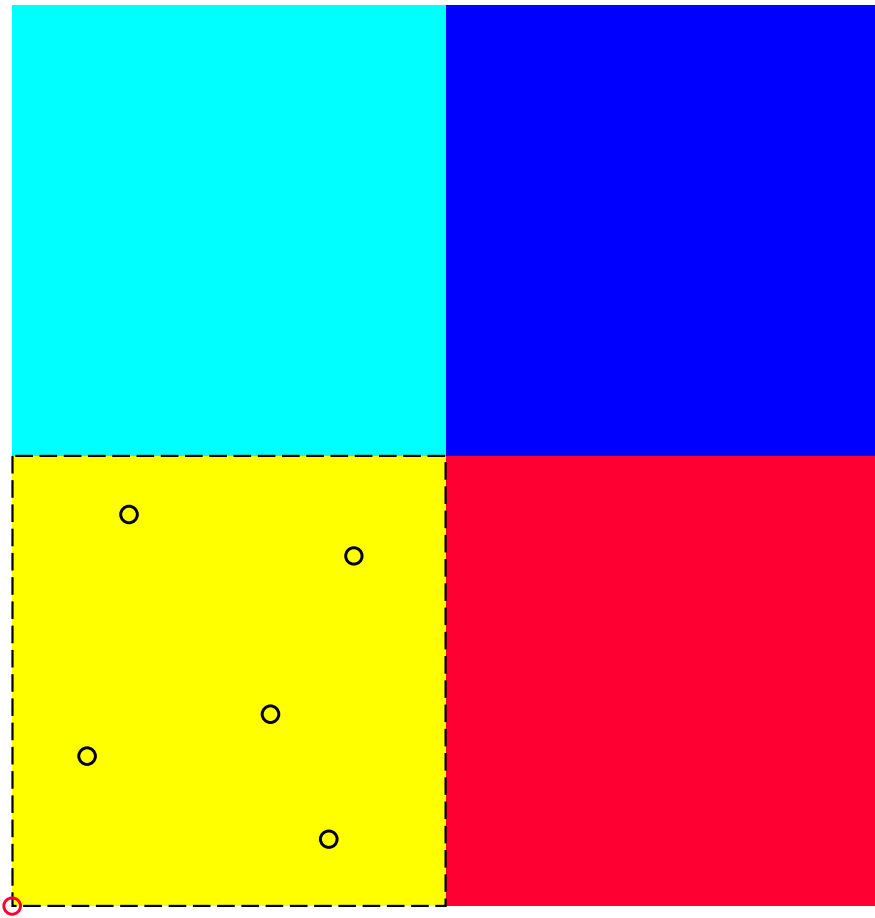
Lepsze podejście:

- próbkowanie 4 najbliższych teksele,
- interpolacja dwuliniowa, w zależności od odległości od punktu próbkowania.
- **Filtracja dwuliniowa** tekstury (*bilinear filtering*) wygładza przejścia między barwami.
- Przy małej rozdzielczości może spowodować rozmycie tekstury.

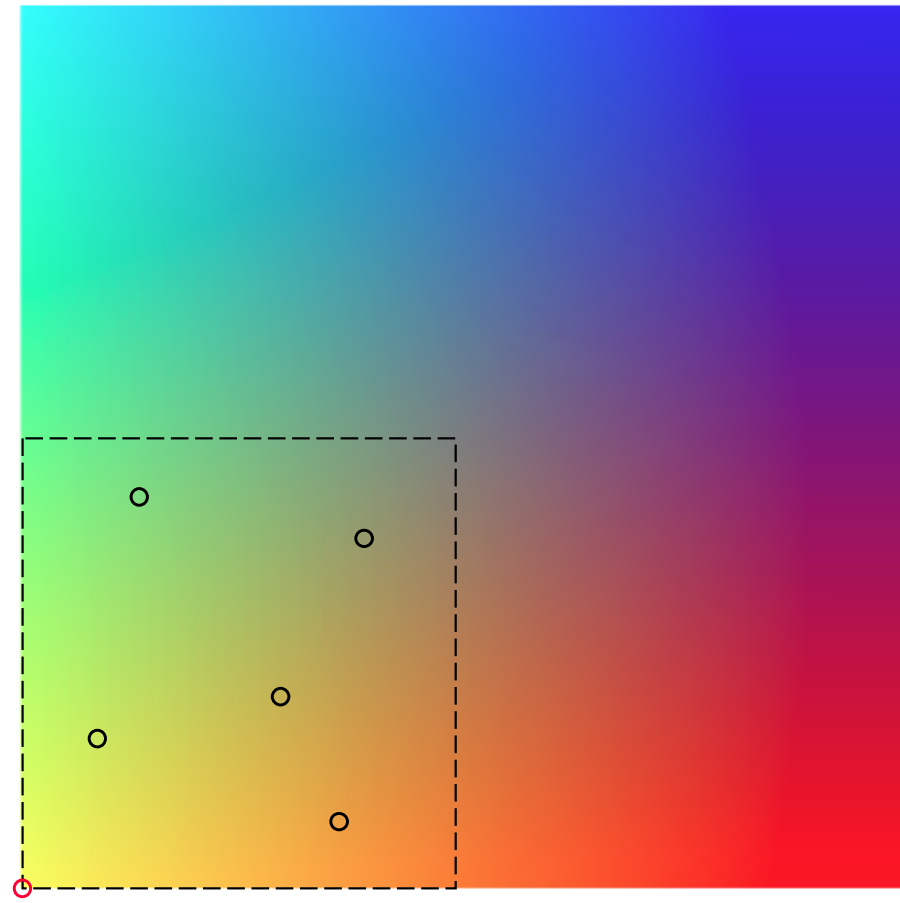


Filtracja dwuliniowa

Próbkowanie punktowe

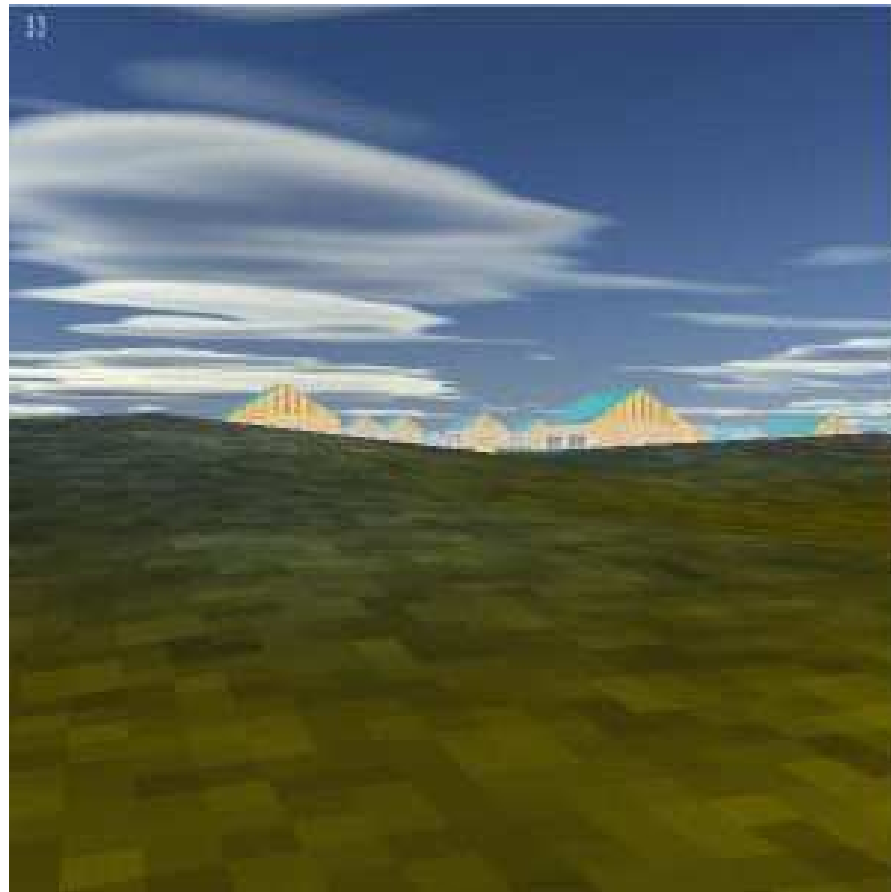


Filtracja dwuliniowa

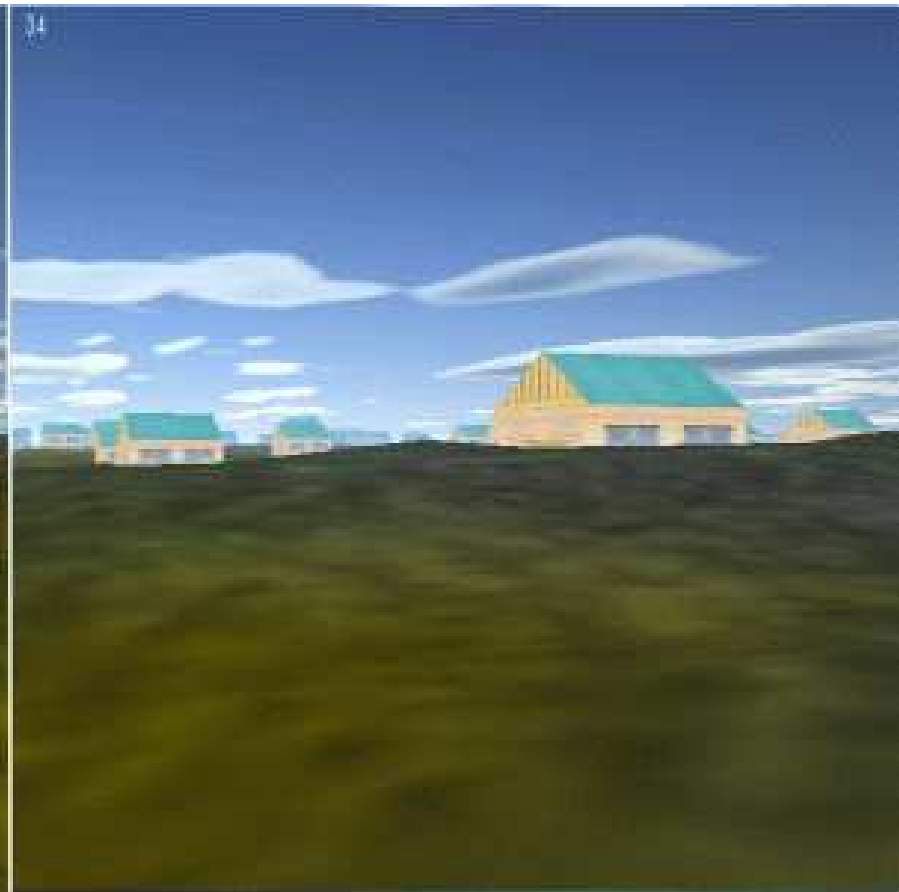


Filtrowanie tekstur

Próbkowanie punktowe



Filtracja dwuliniowa



MIP mapping

Problem rozdzielczości tekstury:

- obiekt blisko kamery – potrzebna tekstura o dużej rozdzielczości,
- obiekt daleko od kamery – potrzebna tekstura o małej rozdzielczości,
- zbyt mała rozdzielczość – tekstura „rozmazana”,
- zbyt duża rozdzielczość – utrata szczegółów.



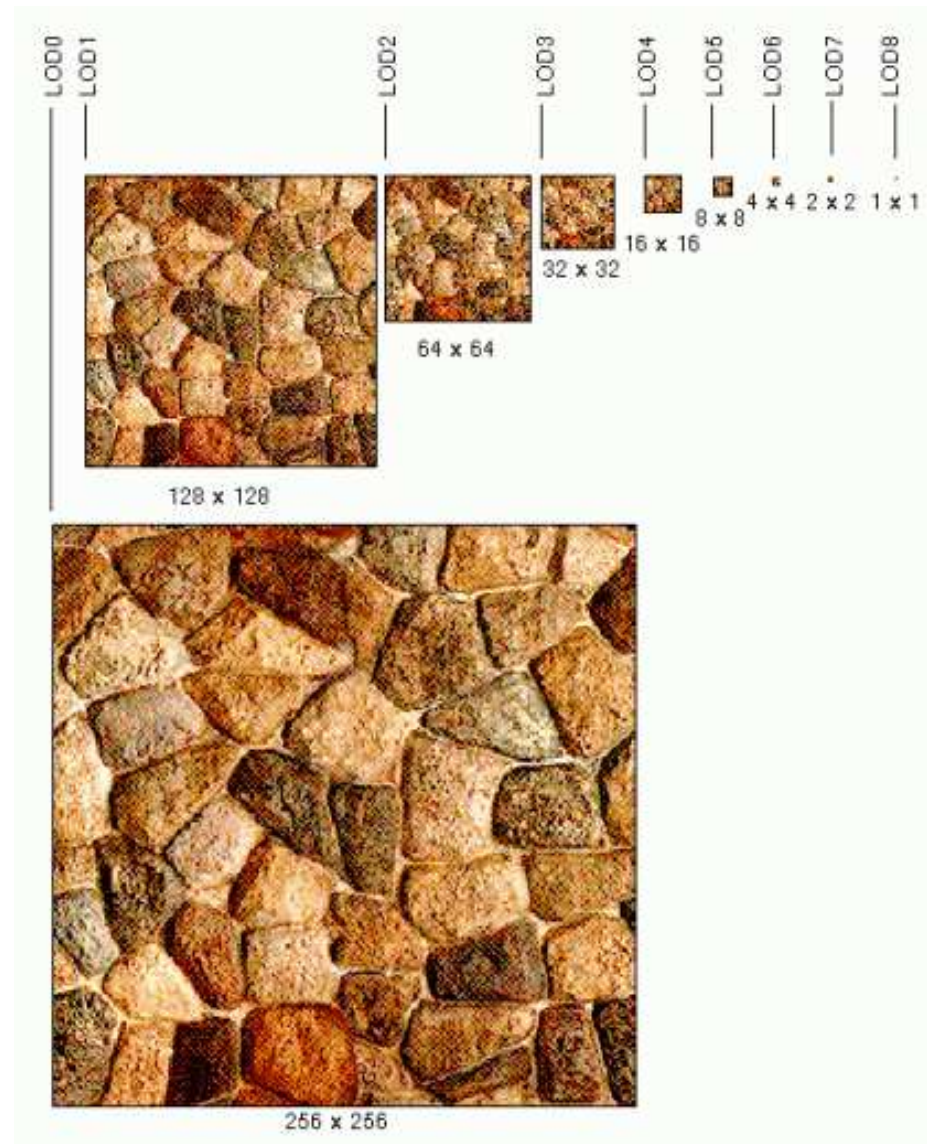
MIP mapping

- **MIP mapping** – tworzonych jest wiele wersji (**mipmap**) tej samej tekstury o różnych rozdzielczościach (różnych poziomach MIP) – automatycznie lub ręcznie.
- Wybierana jest mipmapa o rozmiarach najlepiej pasujących do pokrywanego obszaru.
- Unikamy zniekształceń tekstur.
- Potrzebujemy więcej pamięci.
- Rasteryzer sam dobiera właściwy poziom.

MIP mapping

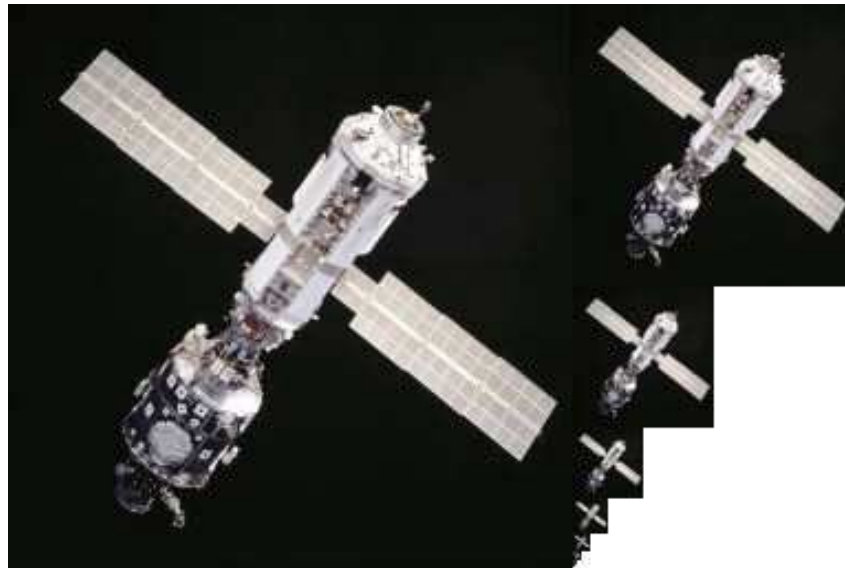


Tekstura powiększona (bez MIP)



MIP mapping

- W praktyce, system skaluje załadowaną teksturę „w dół”, więc należy podać teksturę o maksymalnej rozdzielczości.
- Mipmapy są przechowywane w pamięci w układzie pozwalającym na łatwe próbkowanie tekstele.



MIP mapping

- MIP mapping w połączeniu z filtracją dwuliniową:
 - wybór „najbliższej” mipmapy,
 - filtrowanie dwuliniowe z tej tekstury.
- Dobry rezultat, o ile kamera się nie rusza.
- Przy ruchach kamery – zbliżaniu i oddalaniu się od obiektu – może być zauważalne „przeskakiwanie” między mipmapami.

Filtracja trójliniowa

Filtracja trójliniowa (*trilinear filtering*):

- wybór dwóch najbliższych mipmap – większej i mniejszej niż potrzebna,
- próbkowanie 4 teksele z każdej mipmapy,
- interpolacja barwy w trzech wymiarach.

Daje lepsze wyniki niż dwuliniowa przy ruchu kamery. Przy statycznej kamerze różnica jest niezauważalna.

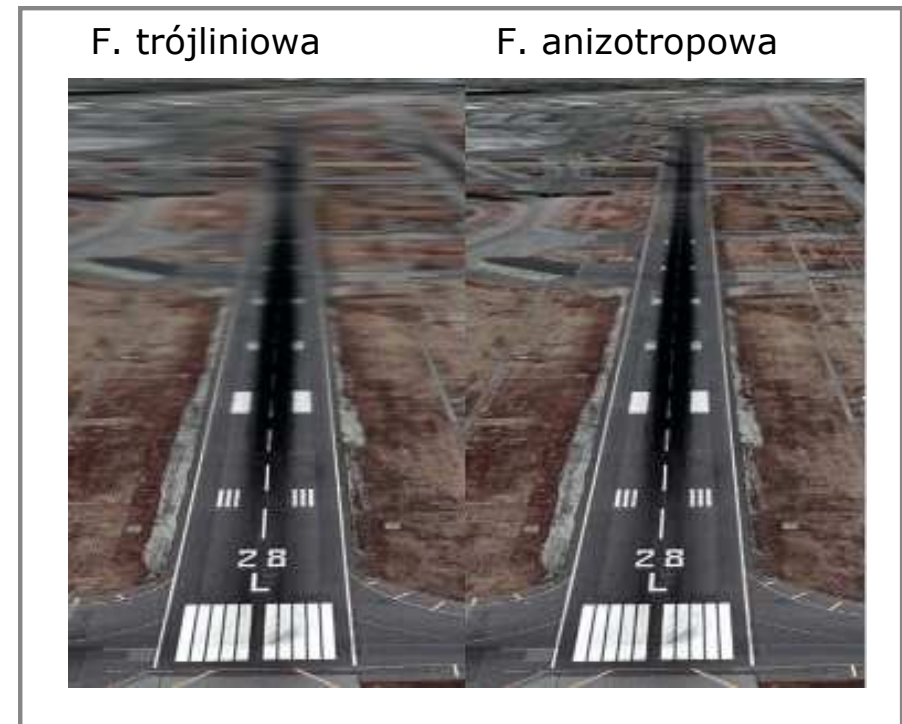
Wymaga większej liczby operacji.

Filtracja anizotropowa

- Filtracja dwu- i trójliniowa: problem przy pokrywaniu dużych powierzchni ułożonych pod dużym kątem względem obrazu (podłóże, ściany).
- Blisko kamery – właściwa rozdzielczość.
- Dalsze plany – za mała rozdzielczość, tekstura ulega rozmyciu.
- Potrzebna jest tekstura o „zmiennej rozdzielczości” zależnie od głębi obrazu.

Filtracja anizotropowa

- „Anizotropowy” – zależny od kierunku.
- **Filtracja anizotropowa** (*anisotropic filtering*) tworzy „mipmapy” przeskalowane niejednorodnie.
- Liczba pobieranych próbek tekstury zależy od ułożenia powierzchni – nie zakłada się, że piksel „jest kwadratowy”.
- Algorytm bardzo obciążający pamięć i procesor GPU.



Filtracja anizotropowa

Przykład mapy anizotropowej tekstur
- kopie skalowane
niejednorodnie.



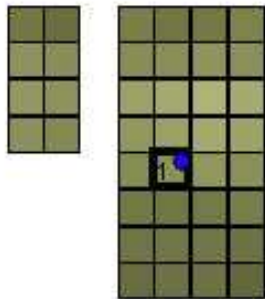
Filtracja anizotropowa

- Rząd filtracji (*anisotropic order*)
 - maksymalny współczynnik skalowania.Parametr często dostępny w opcjach gier.
- Np. współczynnik 8 oznacza, że maksymalny poziom skalowania wynosi 8:1.
- Wyższy poziom może poprawić ostrość szczegółów w dalszych planach, ale zwiększa czas obliczeń.

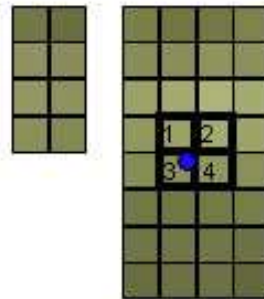
Porównanie metod filtracji



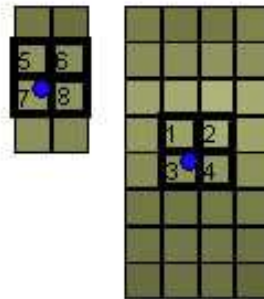
Point Sampling



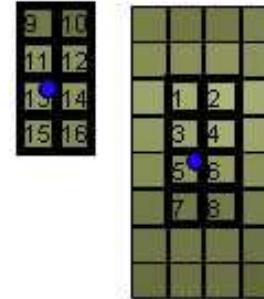
Bilinear Filtering



Trilinear Filtering



Anisotropic Filtering
(2x trilinear)

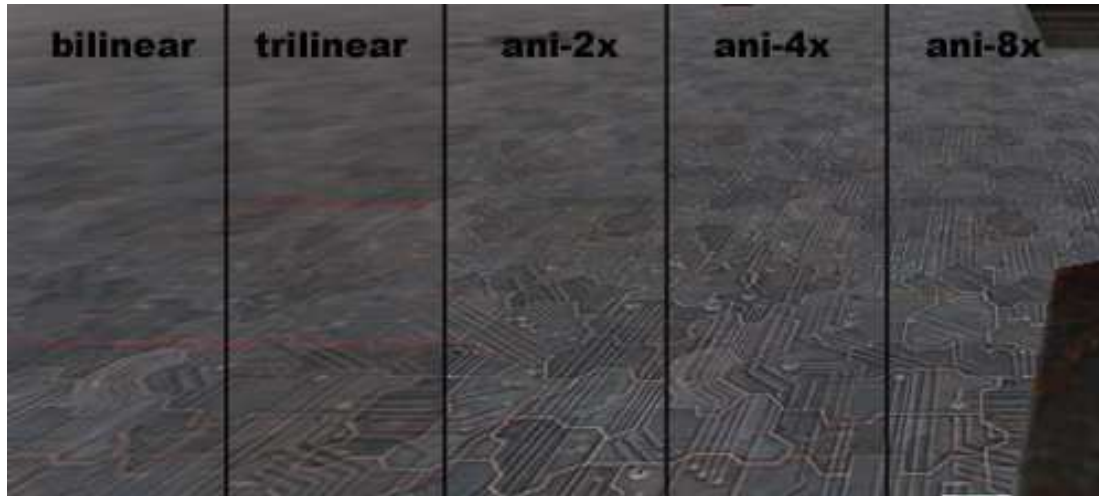


• Pixel Co-ordinates

□ Texel

■ # Sampled Texel

Porównanie metod filtracji



Tekstury a cieniowanie

Wykorzystanie tekstur w procesie cieniowania:
barwa fragmentu jest iloczynem:

- barwy światła (kierunkowego, otoczenia),
- barwy obiektu (zwykle biała),
- współczynnika oświetlenia ($\cos \alpha$),
- barwy spróbkowanej z tekstury, zazwyczaj po filtracji.

Można ustawić barwę obiektu na inną niż biała
– wtedy zabarwimy teksturę.

Tekstury a cieniowanie

Przykład obliczeń (źródło kierunkowe):

- barwa światła kierunkowego: $(0.9, 0.9, 0.8)$
- barwa obiektu: $(1, 1, 1)$
- obliczenie cieniowania Lamberta:
 $\theta = 30^\circ, \cos(\theta) = 0.866$
- barwa pobrana z tekstury: $(0.1, 0.3, 1.0)$
- wynikowa barwa fragmentu:
 $(0.9, 0.9, 0.8) \cdot (1, 1, 1) \cdot 0.866$
 $\cdot (0.1, 0.3, 1.0) = (0.08, 0.23, 0.69)$

Specjalne typy tekstur

- Wielotekstury (*multitextures*) – więcej niż jedna tekstura przypisana do danego obiektu.
- Tekstury dynamiczne – obliczane w trakcie renderingu i nakładane na obiekt (np. mapy światła, odbicia zwierciadlane).
- Mapy głębokości (nierówności) – informacje o głębi tekstury.
- Tekstury tła (*skybox, cube*).
- Tekstury 3D – z zapisaną informacją o głębi tekstury.

Podsumowanie - opis wierzchołka w modelu

Jeden wierzchołek obiektu:

- współrzędne wierzchołka (x, y, z)
- współrzędne wektora normalnego $[x, y, z]$
- barwa odbitego światła otoczenia A
(zwykle biała: $1,1,1$)
- barwa odbitego światła kierunkowego D
(zwykle biała: $1,1,1$)
- materiał – barwa odblasku S (r, g, b)
- współrzędne tekstury (u, v)