



Politechnika Łódzka

Instytut Informatyki

Instrukcja do laboratorium

Fotorealistyczna Grafika Komputerowa

Laboratorium IV

Teksturowanie

Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej
dr inż. Piotr Napieralski
mgr inż. Krzysztof Guzek

Łódź, 06.02.2009



Instytut Informatyki

90-924 Łódź, ul. Wólczajska 215, budynek B9

tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

Nakładanie tekstury jest jednym z najważniejszych zadań grafiki fotorealistycznej. Pozwala zwiększyć stopień realizmu generowanych obrazów, dodając pewne cechy obiektom, które bez tego wyglądałyby nienaturalnie. Jednocześnie ponieważ jest to nakładanie – odwzorowanie obrazu na powierzchni materiału, jest to technika szybsza i efektywniejsza niż realizacja rzeczywistego definiowania właściwości powierzchni obiektu. Nie jest to tylko nałożenie obrazu – zdjęcia, ale może to być także zmiana pewnych postrzeganych właściwości materiałowych. Możliwa jest także zmiana pewnych cech związanych z otoczeniem – tekstura jest także stosowana do przyspieszenia lub uproszczenia wyznaczania oświetlenia. Zalety stosowania tekstur doceniliśmy już wielokrotnie na innych zajęciach. Postaramy się teraz dodać możliwość teksturowania do tworzonego na laboratorium systemu renderującego.

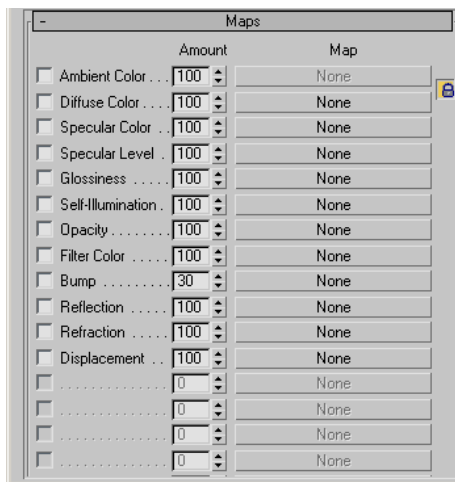


Rysunek 1 Niewątpliwe zalety teksturowania (lewy rysunek – brak, prawy z teksturami)

Nakładanie (mapowanie) tekstury może obejmować szeroki zestaw zagadnień. Formalnie należałoby więc definiować mapowanie jako pewną wektorową funkcję wielu zmiennych. Liczba zmiennych jest zależna od właściwości odwzorowania. [8in]

1.1 IMPLEMENTACJA TEKSTUR

W skrócie tekstura ma za zadanie zwrócić na wyjściu kolor RGB w momencie przecięcia obiektu teksturowanego przez promień. Odpowiednio dwu wymiarowa tekstura zwraca kolor z nałożonego obrazu, tekstura proceduralna oblicza kolor na podstawie zdefiniowanej procedury. Użycie tekstury powoduje modulację odpowiednich współczynników materiału. Definiując teksturę podobnie jak w maxie określamy na jaki współczynnik czy parametr materiału ma ona wpływać. Tekstura powinna być więc powiązana z materiałem.



Rysunek 2 Teksturowanie w programie 3ds Max

Funkcja zwracająca wartość koloru tekstury powinna używać parametrów powiązanych z układem współrzędnym tekstury (u,v). Najprostszy rodzaj tekstury powinien zwrócić określony kolor, niezależny od przeciętego punktu.

Program 1 Definicja tekstury

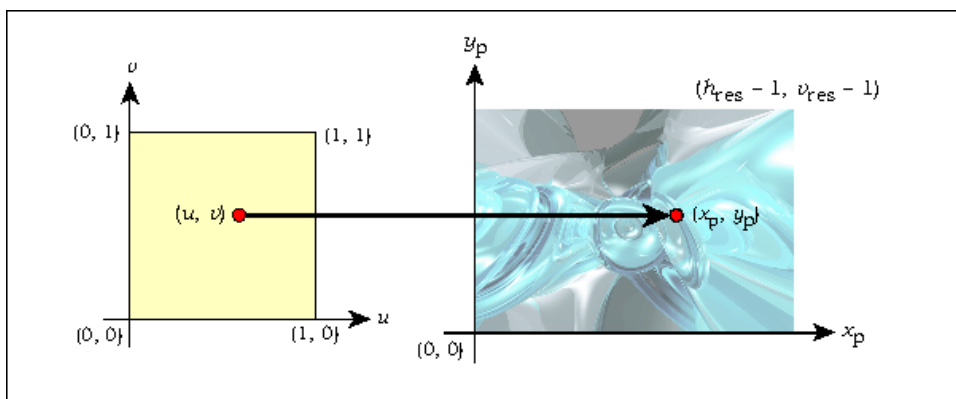
```
public class Texture
{
    public int Width;
    public int Height;
    public Intensity[,] ColorMap;
    (.....) ..
```

Odpowiednio `colormap[x, y]` zawiera piksele pobrane z obrazu tekstury

Obraz zapisany jest w przestrzeni dwuwymiarowej, mapowany jest natomiast na obiekt opisany w trójwymiarowym układzie współrzędnych. Procedura mapowania będzie inna dla każdego rodzaju powierzchni. Przestrzeń obrazu tekstury opisana jest za pomocą znormalizowanego układu współrzędnych tekstury $(u,v) \in [0,1] \times [0,1]$. Tego rodzaju zapis jest standardem dla grafiki komputerowej i nosi nazwę uvmapping-u. Jeśli obraz jest w rozdzielczości (h_{res}, v_{res}) można użyć następującego zapisu dla współrzędnych teksli (piksel tekstury) $(x_p, y_p) \in [0, h_{res} - 1] \times [0, v_{res} - 1]$:

$$x_p = (h_{res} - 1)u,$$

$$y_p = (v_{res} - 1)v,$$



Rysunek 3 współrzędne tekstury (u,v) odpowiadają współrzędnym teksli (x_p, y_p) na obrazie tekstury

Współrzędne x_p, y_p zapisane są jako liczby rzeczywiste. Rysunek powyżej przedstawia związek pomiędzy (u,v) a (x_p, y_p) . Po obliczeniu współrzędnych piksela w punkcie przecięcia, możemy wyciągnąć kolor piksela z obrazu tekstury i wykorzystać go do obliczenia koloru (cieniowania) badanego punktu.

Przyjrzyjmy się teraz sposobom mapowania ze względu na teksturowaną powierzchnię.

1.1.1 MAPOWANIE PROSTOKĄTNE

Najprostszy przykład teksturowania odwzorowuje prostokątną teksturę $(x, z) \in [-1, +1] \times [-1, +1]$ na płaszczyźnie (x, z) . Mając punkt na płaszczyźnie $(x, 0, z)$, współrzędne u, v mogą zostać obliczone:

$$u = (z + 1) / 2,$$

$$v = (x - 1) / 2.$$

1.1.2 MAPOWANIE CYLINDRYCZNE

Niech punkt przecięcia z cylindrem wynosi p i będzie na niego nałożona odpowiednia wartość obrazu tekstury. Obliczonemu przecięciu odpowiadać będzie punkt na obrazie we współrzędnych cylindrycznych $\phi \in [0, 2\pi)$ oraz $y \in [-1, +1]$. Jeśli współrzędne przecięcia są odpowiednio (x, y, z) , to kąt promienia wodzącego ϕ będzie obliczony na podstawie następującego równania:

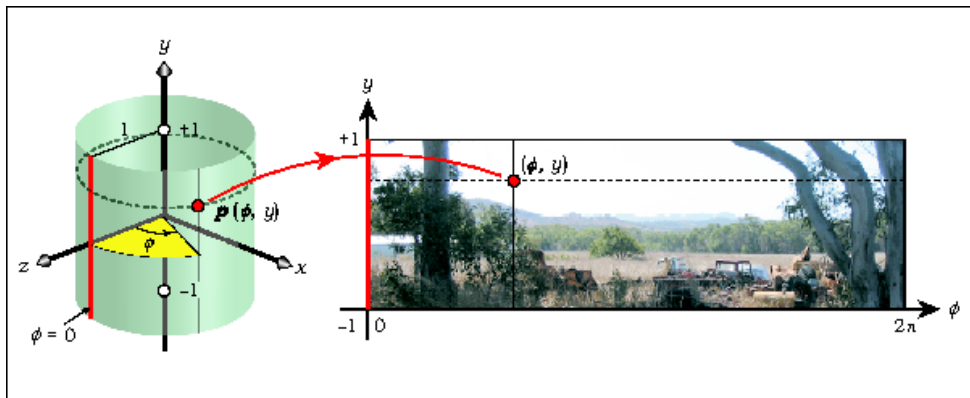
$$\phi = \tan^{-1}(x / z)$$

Współrzędne tekstury będą odpowiednio:

$$u = \phi / 2\pi,$$

$$v = (y + 1) / 2.$$

Poniższy rysunek ilustruje rozważany przypadek:

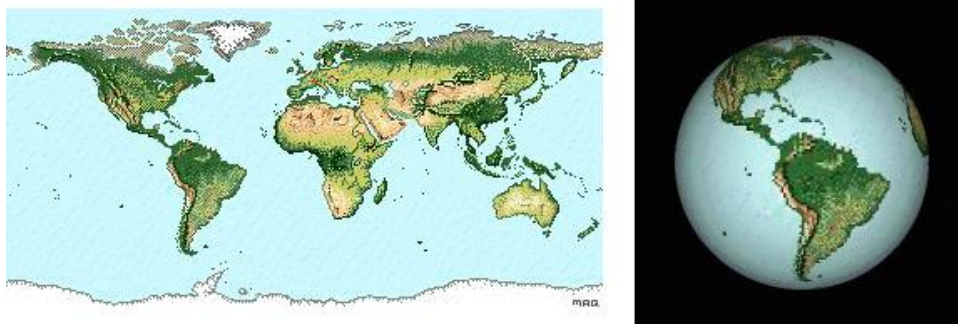


Rysunek 4 Mapowanie cylindryczne

Mapowanie cylindryczne i prostokątne sprowadza się do prostego „rozłożenia” bryły trójwymiarowej na dwa wymiary i przypisanie odpowiadającej zmiennej przecięcia zmienną z powierzchni tekstury.

1.1.3 MAPOWANIE SFERYCZNE

Najprostsza bryła do implementacji przecięcia nie jest już tak prosta w przypadku teksturowania. Jeśli chcielibyśmy za pomocą kartki owinąć walec czy sześcian dokonalibyśmy tego bez specjalnych trudności. W przypadku sfery ...



Rysunek 5 Teksturowanie sfery

Mając obliczony punkt przecięcia o współrzędnych (x,y,z) należy znaleźć odpowiednie współrzędne sferyczne ϕ oraz θ , przy założeniu że promień takiej sfery jest $r=1$. Współrzędna ϕ będzie taka sama jak w przypadku walca, należy zastanowić się nad wartością θ . Korzystając z właściwości biegunowego układu współrzędnych:

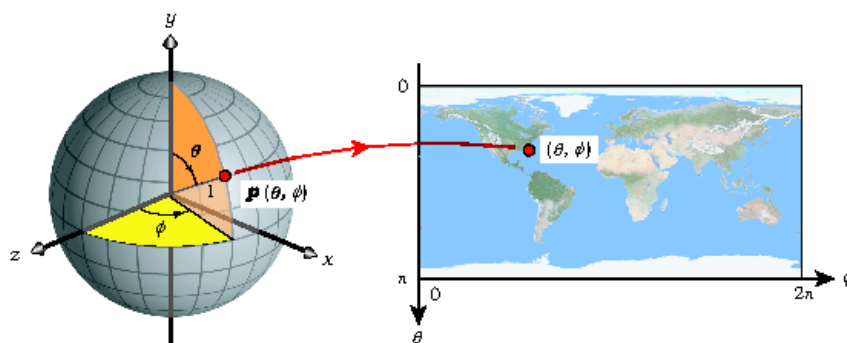
$$\theta = \cos^{-1}(y),$$

Gdzie $\cos^{-1}(y)$ może zostać zapisany jako $a \cos(y)$. $a \cos(y) \in [0, \pi]$, należy jednak pamiętać że $\theta=0$ gdy $y=1$. Odpowiednio parametry (u,v) będą obliczone :

$$u = \phi / 2\pi,$$

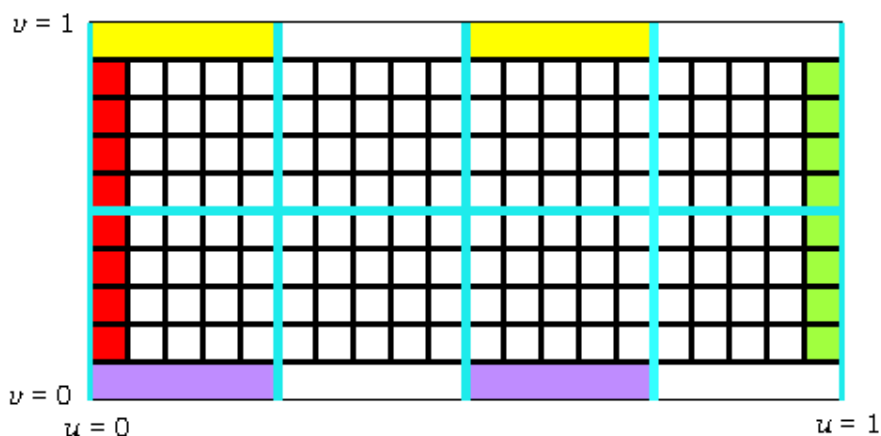
$$v = 1 - \theta / \pi.$$

Przyjrzyjmy się jak wyglądają relacje dla współrzędnych sfery i tekstury na podstawie punktu przecięcia p:



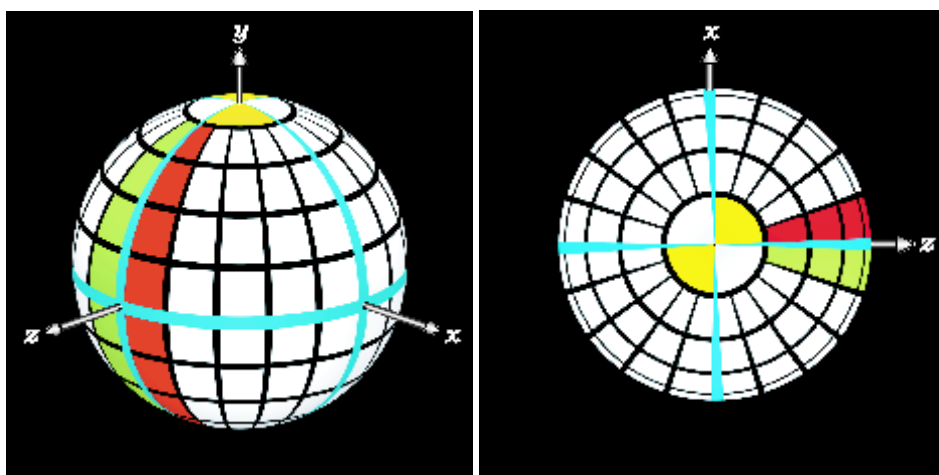
Rysunek 6 Relacja między punktem p na sferze jednostkowej a obrazie tekstury

Jeśli $\phi \in [0, 2\pi]$ oraz $\theta \in [0, \pi]$, parametr przestrzeni w punkcie płaszczyzny (θ, ϕ) będzie miał współczynnik proporcji (aspect ratio) taki jak na rysunku powyżej. Oznacza to że obraz pokrywający sferę powinien mieć taki sam współczynnik kształtu. Obrazy o innym współczynniku powinny przed mapowaniem zostać odpowiednio przeskalowane.



Rysunek 7 Siatka przeznaczona do pokrycia sfery

Rysunek powyżej użyty zostanie do pokrycia sfery teksturą. Został tak dobrany by pokazać zniekształcenia które powstaną po nałożeniu go na obiekt. Poniższy rysunek przedstawia zniekształcenia jakie powstaną na mapowanej teksturze



Rysunek 8 Rezultat mapowania siatki na sferze

Zniekształcenia występują wzdłuż równoleżników. W biegunie północnym i południowym ($y=\pm 1$) gdzie linie $v=0$ oraz $v=1$ zbiegają się w punkcie. Przyjrzyjmy się w jaki sposób obszar żółty został zniekształcony na biegunie ($y=\pm 1$). Czerwony i zielony rząd ilustruje sposób odwzorowania dwóch skrajnych obszarów obrazu. Zostają one zmapowane na krzywej powstałej z przecięcia sfery z płaszczyzną (y,z) gdzie $z > 0$.

Poniżej znajduje się przykładowy kod w języku C++ dla funkcji mapowania sferycznego.

Program 2 Pobieranie współrzędnych tekstury dla mapowania sferycznego

```

SphericalMap::get_texel_coordinates(const Point3D& local_hit_point,
                                   const int xres,
                                   const int yres,
                                   int& row,
                                   int& column) const {

    // wpierw obliczamy kąty theta i phi

    float theta = acos(local_hit_point.y);
    float phi    = atan2(local_hit_point.x, local_hit_point.z);
    if (phi < 0.0)
        phi += TWO_PI;

    // następnie, konwertujemy theta i phi do (u, v) w [0, 1] X [0, 1]

    float u = phi * invTWO_PI;
    float v = 1.0 - theta * invPI;

    // na końcu, przekształcamy u oraz v w współrzędnych texela

    column = (int) ((xres - 1) * u); // kolumna jest poziomo
    row    = (int) ((yres - 1) * v); // wiersz jest pionowo
}

```

2 ZADANIE

Należy rozbudować aplikację o następujące elementy:

1. Definicję klasy tekstury.
2. Odwołanie do tekstury wewnątrz klasy materiału.
3. Implementację mapowania prostokątnego i sferycznego tekstury przy obliczaniu cieniowania.

3 LITERATURA

3.1 BIBLIOGRAFIA

- [1] **Ashdown Ian, P. Eng., LC, FIES.** *Photometry and Radiometry A Tour Guide for Computer Graphics Enthusiasts.* : John Wiley & Sons in 1994
- [2] **Woźniak Władysław Artur.** *Radiometria i Fotometria.* : Instytut Fizyki Politechniki Wrocławskiej 2004
- [3] **Wynn Chris.** *An Introduction to BRDF-Based Lighting* : NVIDIA Corporation 2000
- [4] **Rusinkiewicz Szymon.** *A Survey of BRDF Representation for Computer Graphics* : CS348c, Winter 1997
- [5] **Glassner, Andrew S.** *Space Subdivision for Fast Ray Tracing* : IEEE Computer Graphics & Applications, March 1988, volume 8, number 2, pp. 60-70
- [6] **Marlon John.** *Focus On Photon Mapping* : Premier Press 2003 ISBN 1-1-59200-008-8
- [7] **Shirley Peter, R. Keith Morley.** *Realistic Ray Tracing: Second Edition* : AK Peters; 2nd edition (July 2003) ISBN-13: 978-1568811987
- [8] **Whitted Turner.** *An improved illumination model for shaded display* : Communications of the ACM archive. Volume 23 , Issue 6 (June 1980)
- [9] **Appel Arthur.** *The notion of quantitative invisibility and the machine rendering of solids* : Proceedings of ACM National Conference 1967
- [10] **Sunday Dan.** *Intersections of Rays, Segments, Planes and Triangles in 3D* : softSurfer 2006
- [11] **Maciej Falski** *Przegląd modeli oświetlenia w grafice komputerowej:* Praca magisterska Uniwersytet Wrocławski Wydział Matematyki i Informatyki Instytut Informatyki 2004
- [12] **Christophe Schlick.** *An inexpensive BRDF model for physically-based rendering* : Computer Graphics Forum 1994
- [13] **Kevin Suffern.** *Ray Tracing from the Ground Up* : ISBN: 978-1-56881-272-4 A K Peters, Ltd. Year: 2007

3.2 STRONY INTERNETOWE

- [1in] **Dr inż. Władysław Artur Woźniak.** *Strona informacyjna Instytut Fizyki .* : <http://www.if.pwr.wroc.pl/~wozniak/>

- [2in] **The Australian National University Faculty of Engineering and Information Technology (FEIT)** . *Global Illumination Models Physically Based Illumination, Ray Tracing and Radiosity*
: <http://escience.anu.edu.au/lecture/cg/GlobalIllumination/printNotes.en.html>
- [3in] **Wikipedia**. *Wolna encyklopedia internetowa* . : <http://pl.wikipedia.org/>
- [4in] **Softsurfer**. *List of Algorithm Titles in the softSurfer Archive*. . :
http://softsurfer.com/algorithm_archive.htm
- [5in] **Henrik Wann Jensen**. *Strona informacyjna Henrika Wann Jensena*. . :
<http://www.gk.dtu.dk/~hwj>
- [6in] **Jiajun Zhu** CS 645 Computer Graphics <http://www.cs.virginia.edu/~jz8p/>
- [7in] **Webster's New Millennium™ Dictionary of English, Preview Edition (v 0.9.7)**
Copyright © 2003-2009 Dictionary.com, LLC <http://dictionary.reference.com>
- [8in] Dariusz Sawicki **GKIW Moduł 10 - Dążenie do realizmu**, <http://wazniak.mimuw.edu.pl>