Podstawy Grafiki Komputerowej

Wykład 3: Transformacje

Andrzej Łukaszewski

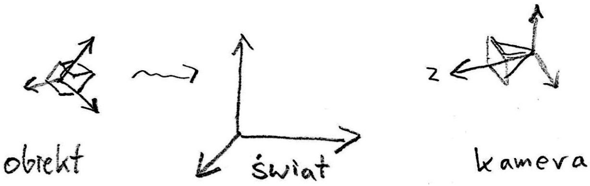
Pracownia Grafiki Komputerowej
(LIGHT — Laboratory of Imaging and GrapHic Techniques)

Instytut Informatyki, Uniwersytet Wrocławski

21 października 2019

Transformacje: motywacje

- Transformacje wykonywane przez GPU musimy zdefiniować sami w shaderze
- Ważna efektywność: wykonywane dla wielu wierzchołków
- Różne układy współrzędnych: lokalny obiektu, świata, obserwatora



- Potrzebne nie tylko przekształcenia liniowe
- Zrozumienie i intuicje geometryczne dla algebry
- Kwaterniony dla rotacji

Transformacje: preliminaria

- zapis wektorów kolumnowy (inna wersja: wiersze)
- dualność zapisów u^T
- kolejność i dopuszczalność operacji !
- iloczyn skalarny:

$$u \cdot v = u^T v =< u, v >= |u| \cdot |v| \cdot cos\alpha$$

- obliczanie rzutu wektora
- norma wektora $|u|^2 =< u, v >$
- macierze przekształceń liniowych 2D/3D (2x2 lub 3x3)

$$\begin{bmatrix} U_x & V_x & W_x \\ U_v & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} U & V & W \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix}$$

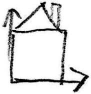
Transformacje: preliminaria

- Wyznacznik jako pole/objętość
- Jak sprawdzić czy punkt leży na płaszczyźnie wyznaczonej przez trzy punkty U, V, W ?
- Jeśli punkt (x, y, z) leży na płaszczyźnie UVW to objętość rozpiętego wielościanu musi być równa 0:

$$
det \begin{bmatrix} x - U_x & x - V_x & x - W_x \\ y - U_v & y - V_y & y - W_y \\ z - U_z & z - V_z & z - W_z \end{bmatrix} = 0
$$

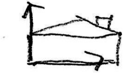
- Otrzymujemy też równanie płaszczyzny przechodzącej przez trzy punkty.

Transformacje: przekształcenia liniowe 2D



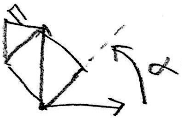
- $\leftarrow$ oryginał

- skalowanie $S(s) = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$

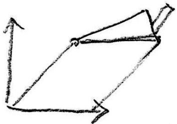


- skalowanie niejednorodne
  $$S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

- odbicie $Mirror_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$



- obroty/rotacje $R(\alpha) = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix}$



- ścięcie $Shear(a) = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$
- translacja ???

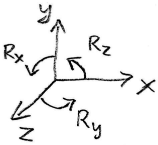
Transformacje: obroty 3D

- rotacje wokół osi x, y, z np.

$$R_z(\alpha) = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix}$$

$$R_y(\alpha) = \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) \\ 0 & 1 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) \end{bmatrix}$$



- Reguła: dodatnie obroty o 90st. $x \rightarrow y \rightarrow z \rightarrow x$

Transformacje: dowolne obroty 3D

- można otrzymać składając trzy obroty wokół osi układu
- to samo to obrót wokół 1 dowolnej osi: macierz ortonormalna o wyznaczniku 1
- czyli dla ortogonalnych i unormowanych U, V, W macierz

$$R = \begin{bmatrix} \\ U & V & W \\ \\ \end{bmatrix}$$

- macierz odwrotna: transponowana

$$\begin{bmatrix} U^T \\ V^T \\ W^T \end{bmatrix} \begin{bmatrix} \\ U \\ \\ \end{bmatrix} = \begin{bmatrix} U \cdot U \\ V \cdot U \\ W \cdot U \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

SVD — singular value decomposition

Dla dodatniej i symetrycznej macierzy $A^T A$ możemy znaleźć n wektorów i wartości własnych konstruując rozkład: $A^T A = VSV^T$

Dowolną macierz można rozłożyć na iloczyn $A = USV^T$, tak że macierze U i V są ortonormalne, a macierz S diagonalna.

Inaczej: dowolne przekształcenie liniowe można rozłożyć na dwa obroty i skalowanie dla 2D:

$$M = R_1 S(s_x, s_y) R_2$$

stąd mamy sposób na liczenie przekształcenia odwrotnego:

$$M^{-1} = R_2^T S(1/s_x, 1/s_y) R_1^T$$

Transformacje: prawo i lewoskrętny układ

Dwie wersje układów są lustrzanymi odbiciami.

Obroty dodatnie zależnie od definicji macierzy ale standardowo zgodne z zasadą prawej/lewej ręki.

Iloczyn wektorowy wektorów u i v jest to wektor prostopadły do obu o długości: $|u \times v| = |u| \cdot |v| \cdot \sin(\alpha)$.
Reguły $x \times y = z$, $y \times z = x$, $z \times x = y$ a odwrotnie z minusem: $y \times x = -z$

$$u \times v = (y_u z_v - z_u y_v, z_u x_v - x_u z_v, x_u y_v - y_u x_v)$$

zastosowanie: tworzenie wektorów lokalnej bazy

Transformacje: wektora normalnego

- przy przekształceniu wierzchołków obiektu przez macierz M shear wektor normalny przekształcony przez M będzie zły.
- jeśli M jest macierz przekształcenia dla wierzchołków to jaka jest macierz N przekształcającą wektory normalne n ?
- M przekształca wektor t styczny do powierzchni poprawnie.

$$t_M = Mt$$

$$n_N = Nn$$

- wektor normalny jest prostopadły do stycznych: $n^T t = 0$, $n_N^T t_M = 0$, stąd mamy :

$$0 = n^T t = n^T M^{-1} M t = (n^T M^{-1}) t_M = ((M^{-1})^T n)^T t_M$$

- Ponieważ zachodzi to dla dla każdego wektora stycznego to mamy macierz przekształcenia dla wektorów normalnych: $N = (M^{-1})^T$ (dla obrotów po prostu M)

Transformacje: jednorodny układ współrz.

- jak rozróżniać punkty i wektory (poprawność dodawania) ?
- dodajemy dodatkową współrzędną jednorodną: 0/1
- mamy gratis reprezentacje translacji (nie jest przekształceniem liniowym) w postaci macierzy dla 3D po dodaniu mamy 4 współrzędne:

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

- translacja dla wektorów bez zmian !
- przekształcenia liniowe w jednorodnym układzie:

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformacje: jednorodny układ współrz.

- możemy reprezentować wszystkie **przekształcenia afiniczne** (liniowe + translacje)
- zastosowania: obrót w 2D wokół dowolnego punktu: $T(x,y)R(\alpha)T(-x,-y)$ lub analogicznie w 3D wokół dowolnej prostej
- Złożenie przekształceń liniowych i translacji w przestrzeni jednorodnej powstaje przez wymnożenie macierzy. Otrzymujemy ogólnie macierz postaci:

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  czyli z odseparowaną częścią liniową i translacją.
- dolny wiersz nie zawsze trzeba pamiętać dla p. afinicznych chyba że wykorzystujemy funkcje mnożenia macierzy.

- Jednorodny układ współrzędnych polega na utożsamianiu punktów po przeskalowaniu tzn $(x, y, z, w) = (sx, sy, sz, sw)$ dla $s \neq 0$.
- Kanoniczna reprezentacja punktu zawiera 0 lub 1 na dodatkowej współrzędnej jednorodnej.
- Dla 2D przestrzeń rzutowa RP^2 to zbiór prostych w przestrzeni R^3 albo półsfera ze sklejonym brzegiem — uwaga nieorientowana !!!
- Czy nie przypomina nam to perspektywy ?

Transformacje: rzuty

- rzut równoległy zachowuje równoległość prostych, może zachować odległości w niektórych kierunkach np. izometryczny; zastosowanie w systemach CAD, typowe rzuty w kierunkach x,y,z.
- rzut perspektywiczny naturalnie oddaje perspektywę (dalsze obiekty mniejsze ale nie zachowuje równoległości prostych),
  odpowiada temu jak postrzegamy świat, temu jak obraz jest tworzony w aparatach fotograficznych i kamerach.
- kanoniczna postać gdy kierunek rzutu to oś z, środek układu współrzędnych.
- rzut równoległy

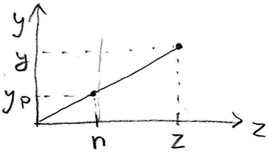
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \to \begin{bmatrix} x \\ y \end{bmatrix}$$

Rzut perspektywiczny:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x/s \\ y/s \end{bmatrix}$$

Zakładamy, że płaszczyzna rzutowania jest $z = n$, z tw. Talesa:

$$\frac{y}{z} = \frac{y_P}{n}$$



czyli

$$y_P = \frac{y}{z/n}$$

W układzie jednorodnym odpowiada temu macierz:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ z/n \end{bmatrix} = \begin{bmatrix} \frac{x}{z/n} \\ \frac{y}{z/n} \\ n \\ 1 \end{bmatrix}$$

Jak nie stracić informacji o odległości ?

Inna macierz:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{n+f}{n} & -f \\ 0 & 0 & 1/n & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z\frac{n+f}{n} - f \\ z/n \end{bmatrix} = \begin{bmatrix} \frac{x}{z/n} \\ \frac{y}{z/n} \\ n + f - \frac{nf}{z} \\ 1 \end{bmatrix}$$

Funkcja $F(z) = n + f - \frac{nf}{z}$ spełnia $F(n) = n$, $F(f) = f$ i jest monotoniczna na tym przedziale więc zachowuje porządek względem głębokości.

Potok grafiki:

- zmiana współrzędnych obiektu na współrzędne świata
- transformacja do współrzędnych kamery (z lub $-z$ to kierunek patrzenia)
- clipping do prostopadłościanu/ostrosłupa widzenia
- przekształcenie do współrzędnych ekranu
- rasteryzacja

Jak działa przekształcenie perspektywiczne?

Podział perspektywiczny: sprowadzenie współrzędnych jednorodnych do kanonicznej postaci).

Transformacje: model kamery

Specyfikacja rzutu: równoległy lub perspektywiczny.

1. PW — Punkt Widzenia
2. KP — Kierunek Patrzenia
3. KW — Kierunek Wertykalny (UP)
4. Piramida widzenia: kąty / odległość rzutni
5. Przednia i tylna płaszczyzna obcinająca *n* i *f* (odległości)