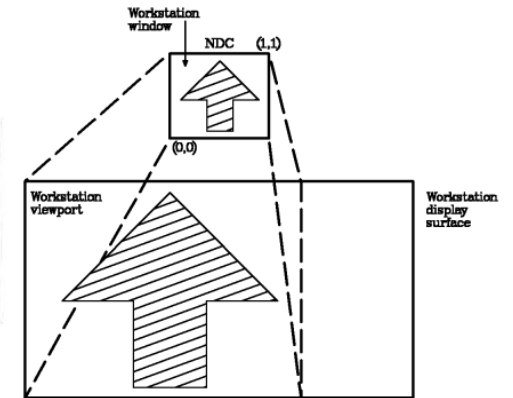


Obr. 1.8 Základná štruktúra interaktívnej grafickej aplikácie



3D grafika zachováva schému z Obr. 1.8 z Ružického, najjednoduchšie premietanie iba vynechá hĺbku,

(x,y,z) v scéne sa zobrazí na (x,y) . Ak by sa ale dva body premietli rovnako, tak uvidíme bližší z nich.

Na pravom obrázku si šípku predstavme ako 3D objekt, no do záberu sa premieta nám jeho neviditeľná, odvrátená strana s menej šrafovaním.

Zdroj: <http://pic.dhe.ibm.com/infocenter/zos/v1r13/topic/com.ibm.zos.r13.admk100/admk1a0036.htm> . Copyright IBM Corporation 1990, 2012

Zopakovali sme si dva známe obrázky, no pridali sme k nim 3D. **Akou maticou by ste zriadili priemet $(x,y,z) \rightarrow (x,y)$?** (Ružický, s. 142)

Ružický, E. et al. 2012. *Počítačová grafika a spracovanie obrazu*. <http://www.sccg.sk/~ferko/PGASO2012-bookmarks.pdf>

Ak by sa ale dva body premietli rovnako, tak uvidíme bližší z nich (**problém viditeľnosti v 3D**).

Algoritmus z-buffer sa používa najčastejšie na riešenie v praxi.

Objekty najčastejšie modelujeme trojuholníkmi a najčastejšia triangulácia roviny alebo telesa býva

Delaunayova. Ak treba, zopakujme si ju za 5 minút.

Voronoi, 2 min

<https://www.youtube.com/watch?v=7eCrHAV6sYY>

Delaunay, 3 min

<https://www.youtube.com/watch?v=bmaUtG4CbOs>

Objekty najčastejšie modelujeme OTEXTUROVANÝMI a OSVETLENÝMI trojuholníkmi.

Najčastejší osvetľovací model navrhol Phong (vietnamské priezvisko).

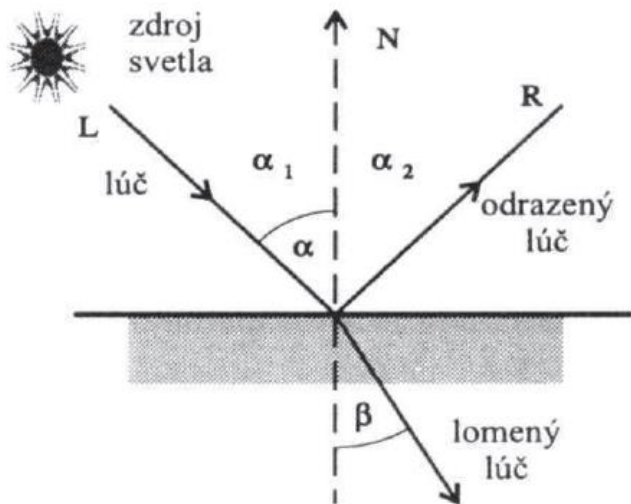
Phong, 3 min

<https://www.youtube.com/watch?v=DFMFPsDOUa8>

Phong, 12 min

<https://www.youtube.com/watch?v=kmAGV7y7rto>

Najprv si ale pripomenieme postup modelovania telies na guli, body trojuholníkov nameriame alebo vypočítame. V týchto bodoch budeme vyčísl'ovať farbu/osvetlenie. K farbe objektu súčtom pridáme denné svetlo (ambient) a prinásobíme difúzne osvetlenie lúčom v danom bode. Vektory označujeme L (light), N (normal).



Najjednoduchší z modelov je Bouknightov z roku 1970. Používa iba difúznu a ambientnú zložku. **Ambientná zložka** reprezentuje okolité svetlo, ktoré vzniká mnohonásobnými odrazmi a rozptylmi a spôsobuje, že odvrátené povrchy nie sú úplne čierne. Základný vzorec pre intenzitu (1) pôvodne zahŕňal iba jeden zdroj svetla umiestnený v nekonečnej vzdialenosti od osvetľovaného bodu v smere pozorovateľa:

$$I(\lambda) = K_a + K_d(\mathbf{N} \cdot \mathbf{L}) \quad (K_a + K_d) < 1 \quad (1)$$

Konštantu k_a kopírujeme z textúry, ak je sivá, alebo trojmo po zložkách R, G, B, jedno alebo tri čísla. K_a nastavíme.

10.3 Modelovanie 3D objektov

V kapitole 4 "Krivky a plochy" sme popísali možnosti vytvárania plôch, ktoré však nutne nemuseli uzatvárať objem. V mnohých aplikáciach však potrebujeme vytvoriť skutočné trojrozmerné teleso, aby sme preň mohli zisťovať iné technické a fyzikálne charakteristiky, napríklad ťažisko. Pri metóde konečných prvkov treba teleso rozložiť na základné stavebnicové prvky ako sú kocky alebo štvorsteny, prípadne hranicu rozložiť na trojuholníky alebo štvoruholníky. Musíme vedieť pre objem ohraničený hranicou zistiť, či hranica je korektne definovaná. Často sa stretávame s aproximáciou telesa mnohostenom, i keď z matematického hľadiska sa dá hovoriť o interpolácii jeho časti (napr. pri rotačných a translačných telesách). V aplikačných softveroch sa vypracovali rôzne techniky reprezentácie a modelovania telies. V tejto časti priblížime základné techniky modelovania objektov.

10.3.1 Hraničná reprezentácia

Pri hraničnej reprezentácii sa 3D objekt reprezentuje plochou, ktorá tvorí hranicu objektu. Môžeme ju vytvoriť viacerými spôsobmi: rotačne, translačne, analyticky, splajnovo, objemovo, CSG stromom prípadne inými. Pri definovaní niektorých objektov v priestore používame aproximáciu mnohostenom pre ich vykresľovanie. Podrobnejšie o hraničnej reprezentácii mnohostenom, čo podrobnejšie rozoberieme v časti 10.4.

Rotačné teleso

Rotačná plocha vznikne rotáciou krivky okolo priamky, ktorú nazývame *osou rotačnej plochy*. Pre zjednodušenie zápisu zvolíme sústavu súradníc tak, aby sme mali zadanú os rotácie totožnú s osou z a krivka bola zadaná v rovine xz , inak uskutočníme transformáciu v priestore. Nech parametrické vyjadrenie krivky je:

Metódy modelovania a zobrazovania 3D objektov	141
10.1. Úvod	141
10.2. Premietanie	141
10.3. Modelovanie 3D objektov	149
10.4. Kódovanie mnohostenov	154
10.5. Metódy zobrazovania	158
Určenie viditeľného povrchu	161
11.1. Úvod	161
11.2. Zjednodušenie stredového premietania a obálky	161
11.3. Algoritmus využívajúci z-bufer	163

Video – Chapter 1 for sphere texturing using stereographic projection (13 min. * 9 kapitol) http://www.dimensions-math.org/Dim_tour_E.htm

Slovenske titulky <https://www.youtube.com/embed/6cpTEPT5j0A?list=PL3C690048E1531DC7>

$$x = f(t), \quad z = g(t).$$

Každý bod na rotačnej ploche je určený voľbou t a parametrom otočenia u . Parametrické rovnice rotačnej plochy môžeme ľahko odvodiť:

$$x = f(t) \cdot \cos(u),$$

$$y = f(t) \cdot \sin(u),$$

$$z = g(t),$$

kde parameter t je z intervalu zo zadania krivky a parameter u je z intervalu $\langle 0, 2\pi \rangle$. Ak plocha vytvára v priestore teleso, potom ho nazývame **rotačným telesom**. Napríklad keď je definujúca krivka uzavretá (napr. torus) alebo krivka sa začína a končí na osi z (napr. guľa). Pretože guľa môže byť tiež vytvorená rotačnou plochou, prevzali sa pojmy ako meridián a rovnobežkové kružnice pre všeobecnú rotačnú plochu. Guľová plocha je zadaná nasledujúcimi rovnicami :

$$x = r \cdot \cos(t) \cdot \cos(u),$$

$$y = r \cdot \cos(t) \cdot \sin(u),$$

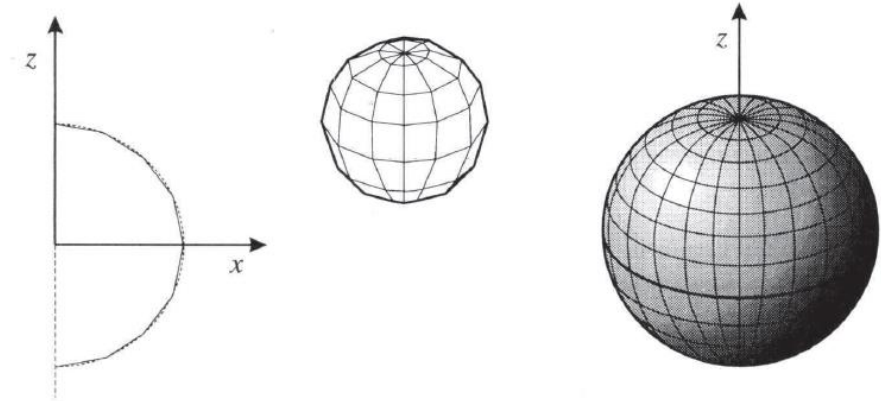
$$z = r \cdot \sin(t),$$

kde $t \in \langle -\pi/2, \pi/2 \rangle$ a $u \in \langle 0, 2\pi \rangle$.

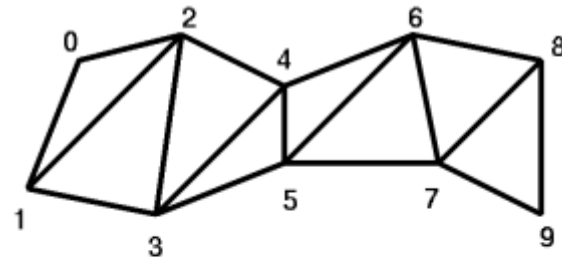
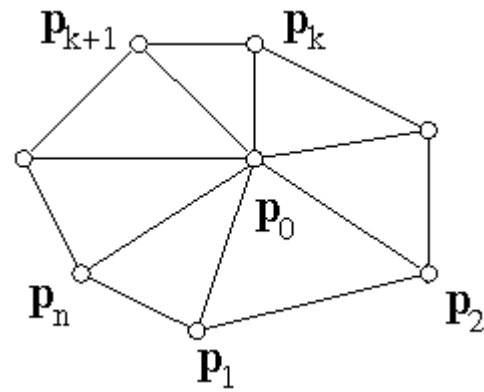
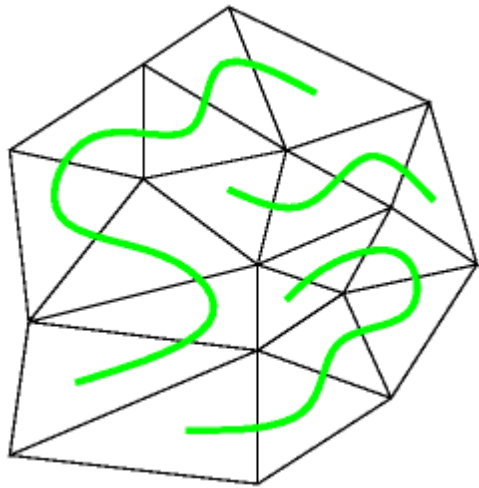
Na plôšky (faces) sa modelovanie štvoruholníkmi (quadrilaterals) hodi, no na renderovanie sa napokon štvoruholníky zvyknú predeliť na trojuholníky, aby sa na grafickú kartu dali posielat plôšky nie jednotlivo, ale PARALELNE. Rozlišujeme dva typy súborov trojuholníkov: vejár (fan) a pás/pruh (stripe), v slovenčine sa už dávnejšie zaužívalo príbuzné slovo striptíz (strip tease), no stripifikáciu v bežnom živote poznáme pri lúpaní jablka alebo šúpaní zemiakov. Vejáre a pásy zrýchľujú prenos dát, prvý trojuholník preniesieme celý (tri vrcholy 0, 1, 2), no na každý ďalší stačí preniesť iba súradnice jediného vrchola v 2D alebo 3D (2-3 reálne alebo celé čísla). Úlohu možno riešiť ručne, no optimálny algoritmus by mal nájsť globálne minimum. Taká triangulácia (iná ako DT) sa volá hamiltonovská a je na obrázku vpravo dole. Metaforicky ju nazvali Onion Peeling, šúpanie cibule. Odkaz na článok je pod obrázkom a čiarkovaná lomená čiara indikuje JEDINÝ stripe, na rozdiel od troch zelených vľavo hore.

Lineárne interpolujeme krivku v rovine xz výberom n bodov (napríklad s prírastkom dt pre parameter t). Zvolíme si prírastok du pre rotáciu tak, aby sme aproximovali rotáciu m otočeniami od 0 po 2π . Týmto vytvoríme na rotačnej ploche sieť u - a t -kriviek, ktoré lineárne interpolujú meridiány a rovnobežkové kružnice (pozri obr.10.8).

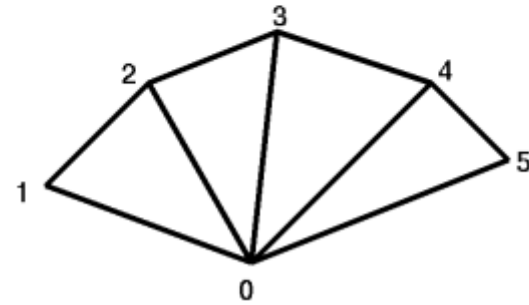
Ak krivka prechádza cez os z , potom je výhodné vybrať tento bod krivky. Po rotácii interpolujúce steny s vrcholom na osi z vytvárajú trojuholníky a ostatné steny zo siete vytvárajú štvoruholníky.



Obr. 10.8 Zobrazenie hranice gule pomocou rotačnej plochy



triangle "strip"
 vertices: 10 triangles: 8
 vertices per triangle: 1.25



triangle "fan"
 vertices: 6 triangles: 4
 vertices per triangle: 1.5

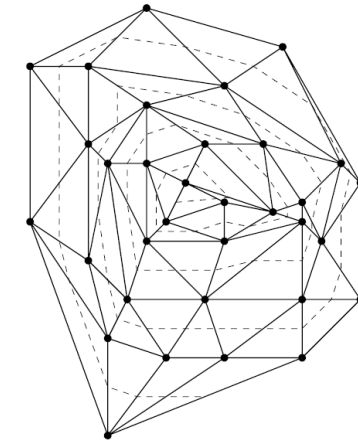
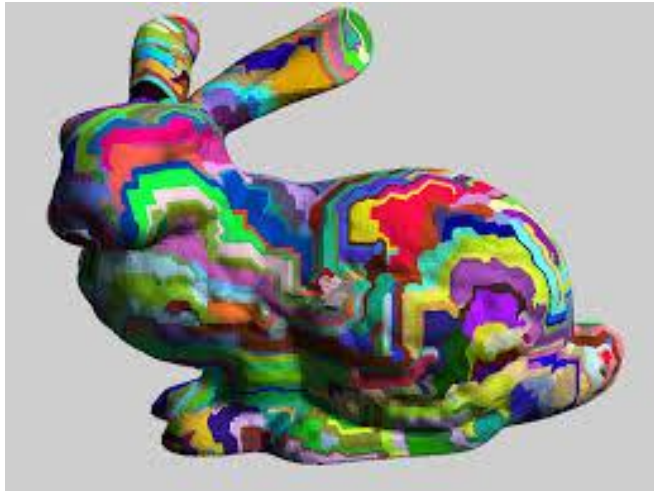


Figure 11: Hamiltonian triangulation via onion method.

http://wscg.zcu.cz/wscg2004/Papers_2004_Full/G07.pdf <https://www.cs.cmu.edu/~kmcrae/Projects/StripePatterns/> <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.3639&rep=rep1&type=pdf>

Video <https://www.youtube.com/watch?v=H3GauH8IOx4>

13.2.3 Phongov model

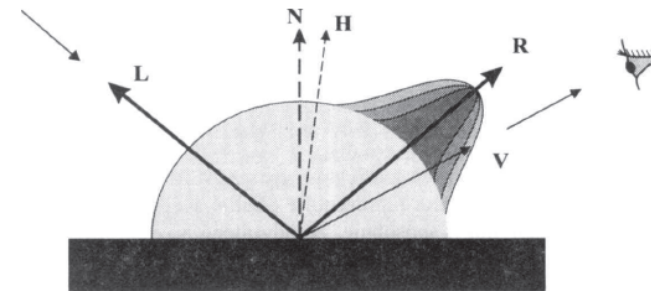
Klasickým modelom, najpoužívanejším v počítačovej grafike, je Phongov lokálny osvetľovací model. Okrem ambientnej a difúznej zložky zahrňuje i zložku zrkadlovú. Jej charakteristickou vlastnosťou je smerovosť a je príčinou vzniku odleskov.

Odlesky môžeme pozorovať ako svetlé miesta na zobrazovanom objekte. Sú odrazom svetelných zdrojov alebo iných svietiacich objektov. Empiricky môžeme tento jav modelovať funkciou, ktorá dosahuje maximum v zrkadlovom smere od vektora svetelného zdroja a rýchlo klesá v smeroch odkláňajúcich sa od zrkadlového smeru. Základný vzťah, ktorý odvodil Phong má tvar:

$$I(\lambda) = K_a(\lambda) \cdot I_a(\lambda) + K_d(\lambda) \cdot I_L(\lambda) \cdot (\mathbf{N} \cdot \mathbf{L}) + K_s \cdot I_L(\lambda) \cdot (\mathbf{R} \cdot \mathbf{V})^k, \quad (3)$$

K_s je koeficient zrkadlového odrazu a k je miera drsnosti povrchu.

Zrkadlová zložka je nezávislá na vlnovej dĺžke, a tak farba odleskov závisí iba na farbe zdroja svetla. Vektor \mathbf{R} je symetrický k vektoru \mathbf{L} podľa normály a možno ho vyjadriť zo vzťahu $\mathbf{R} = 2(\mathbf{L} \cdot \mathbf{N})\mathbf{N} - \mathbf{L}$. Intenzita nadobúda maximum vtedy, ak smer dopadu lúča svetla a smer pozorovateľa spĺňajú zákon odrazu. Toto maximum je tým výraznejšie, čím je parameter k väčší. S rastúcim k sa odlesky na telese stávajú menšie a ostrejšie, pre dokonalé zrkadlo parameter k je nekonečno (obr. 13.4). V prípade $\mathbf{V} \cdot \mathbf{R} < 0$, je pozorovateľ odvrátený od zrkadlovej zložky a tento súčin považujeme za nulový.



Obr. 13.4 Intenzita svetla v závislosti od parametra k

Osvetľovacie modely a tieňovanie	193
13.1. Úvod	193
13.2. Empirické osvetľovacie modely	194
13.3. Model svetla	197
13.4. Osvetlenie povrchu	199
13.5. Lokálne fyzikálne modely	201
13.6. Tieňovanie	203

Video Blinn-Phong <https://www.youtube.com/watch?v=SE6U7HG36uU>

VIDITEĽNOSŤ

Ak by sa ale dva body premietli do toho istého pixla, tak uvidíme bližší z nich (problém viditeľnosti v 3D).

Algoritmus z-buffer sa používa najčastejšie na riešenie v praxi.

2D e-learning K. Vojtova <http://www.st.fmph.uniba.sk/~vojtova5/PG/index.html>

obsahuje aj videa Viditeľnosť Z. Černeková, viacpohľadovo sa pozrieme na z-buffer

<http://www.st.fmph.uniba.sk/~vojtova5/PG/Viditelnost.html>

Ak treba, zopakujme si výklad za pár minút.

z-buffer, 2 min https://www.youtube.com/watch?v=yhwg_O5HBwQ

z-buffer, 7 min <https://www.youtube.com/watch?v=johvsXKB8mk>

11.3 Algoritmus využívajúci z-bufer

Tento algoritmus je jedným z najjednoduchších algoritmov pracujúcich v priestore obrazu. Počas jeho práce sa udržiava najmenšia z-súradnica jednotlivých obrazových bodov (z-bufer) a tiež bufer ich farieb pre zobrazenie. Na začiatku z-bufer naplníme maximálnymi hodnotami a bufer farieb inicializujeme farbou pozadia. Každú stenu transformujeme do rastrovej formy a vypočítanú z-súradnicu porovnáme s odpovedajúcim bodom z-bufra. Ak je hodnota menšia t.j. bod je bližšie k pozorovateľovi, potom sa bod vykreslí a hĺbková súradnica sa odloží do z-bufra.

Algoritmus z-bufer sa dá zapísať do týchto krokov:

Z-bufer algoritmus

1. Pre každý obrazový bod (x, y) inicializujeme z-bufer a bitovú mapu t.j.

$$Z_{buf}(x, y) = z_{max} \quad \text{and} \quad M_{bit}(x, y) = backgr$$

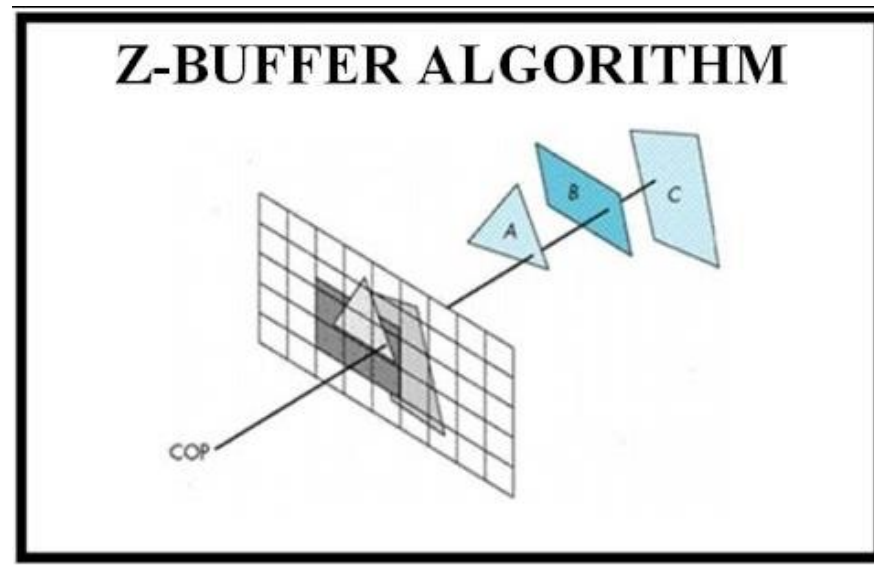
2. Pre každú stenu P určíme v priestore mnohoúhelník a urobíme jeho rozklad do rastrovej formy. Vypočítame z súradnic (t.j.hĺbku) pre každý obrazový bod (x, y) .

if $z(x, y) < Z_{buf}(x, y)$ **then begin**

$$Z_{buf}(x, y) = z(x, y);$$

$$M_{bit}(x, y) = colour(P);$$

end



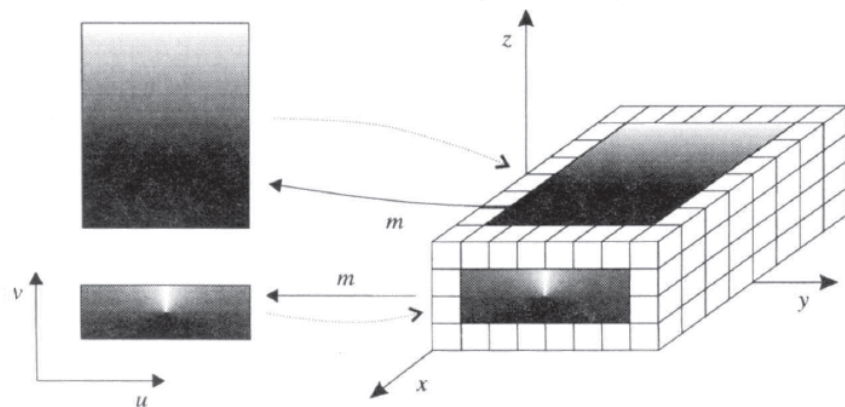
TEXTUROVANIE

Z matematického hľadiska definujeme **všeobecnú textúru** ako zobrazenie rovinatej oblasti do modulovaného priestoru, ktorým môže byť priestor farieb alebo úrovni šedej

$$t : D_t \rightarrow M, \text{ kde } D_t \subseteq \mathbb{R}^2 \text{ a } M \subseteq \mathbb{R} (\mathbb{R}^3).$$

Ak máme zadaný tvar objektu, tak pomocou inverzného mapovania budeme zobrazovať pre každý bod povrchu bod z oblasti textúry

$$m : D_m \rightarrow D_t, \text{ kde } D_m \text{ je oblasť na povrchu objektu.}$$



Obr. 15.1 Mapovanie textúry (texture mapping)

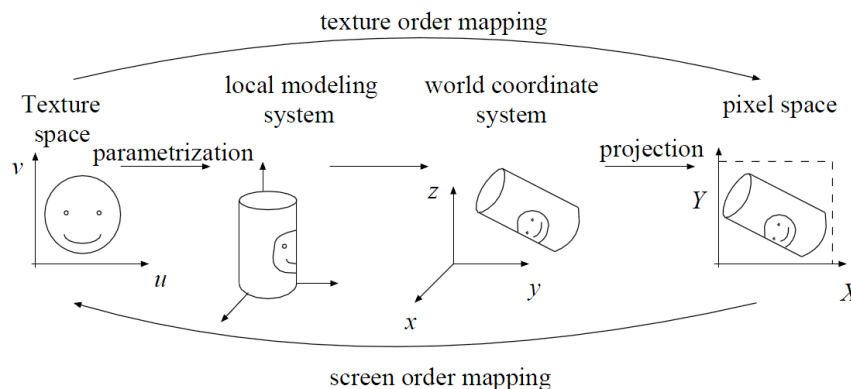


Figure 12.1: Survey of texture mapping

Výklad z Ružického a schéma z učebnice Szirmay-Kalos. Nevyhnutnou etapou pred grafickým zobrazením je **geometrické modelovanie**. V počítačovej grafike sa používa 5 až 7 súradnicových systémov, ak nerátame ďalšie, ktoré treba pre multimédiá. Nasleduje charakteristika 5 súradnicových systémov pre 2D a 3D grafiku podľa ISO noriem CGRM, PREMO, PHIGS:

- modelové (pre objekty, lokálne súradnice pre každý objekt)
- svetové (scéna, kde sa objekty kombinujú)
- normalizované, interné pre grafický systém
- pohľadu, kamery
- zariadenia (DC, device coordinates)

Šiesty súradnicový systém sa označuje TLC (text local coordinates), v ktorom návrhári fontu reprezentujú krivky pre true type fonty. Tieto súradnice sú obdobou modelových pre grafický výstupný prvok text. Siedmy súradnicový systém sa v prípade potreby používa na textúrovanie povrchu objektov a pre tieto súradnice sa zaužívalo označenie (u, v). Okrem toho sa používajú aj súradnice na vyjadrenie polohy svetelného zdroja pre výpočet viditeľnosti či osvetlenia, alebo sa môže zaviesť pre potreby interakcie rozdelenie obrazovej alebo pracovnej plochy do nejakých vhodných častí. Existujú aj ďalšie, tzv. multimediálne súradnice (napr. na vyjadrenie polohy záznamu v excelovskej tabuľke, spreadsheet). Pri každom súradnicovom systéme treba zvoliť počiatok, osi, orientáciu a pomer strán. Po zavedení súradníc možno už geometriu scény a jej transformácie vhodne vyjadriť pomocou lineárnej alebo nelineárnej algebry a maticového vyjadrenia afinných transformácií a projekcií.

Video VECTARY ilustruje tvorbu 3D obsahu https://www.youtube.com/watch?v=pc_iqT4GEmo