MODELOVANIE A REPREZENTÁCIA TROJROZMERNÝCH OBJEKTOV

Existuje mnoho štruktúr a metód slúžiacich na reprezentáciu trojrozmerných objektov v počítačovej grafike. Voľba závisí od viacerých faktorov. Dôležitý je *tvar a typ samotného objektu*, ktorý chceme reprezentovať, a potom aj *účel* reprezentácie. Pokiaľ nám ide o vytvorenie pekného obrázku, spravidla stačí približná aproximácia objektu a častá voľba bude pravdepodobne reprezentácia mnohouholníkovou sieťou. Ak ide o návrh súčiastky a podobne, približná mnohouholníková sieť nie je uspokojivá a radšej sa prikloníme k presným modelom ako je CSG reprezentácia (napríklad pre stojnícky priemysel) alebo reprezentácia pomocou splajnových plôch (pri modelovaní karosérií áut alebo plášťov lietadiel). V neposlednom rade svoju úlohu zohráva aj *forma vstupných údajov.* Typickým príkladom je spracovanie a vizualizácia medicínskych dát, keď už vstup je reprezentovaný voxlami a teda nejaké ďalšie špekulácie o inej reprezentácii strácajú opodstatnenie – je na užívateľovi (lekárovi), aby z rastrovej štruktúry prečítal potrebnú informáciu.

Reprezentáciu objektu treba zvoliť jednak na strane užívateľa za účelom vytvárania nových a modifikácie už existujúcich objektov, a jednak na strane ukladania dát vnútri grafického systému. Tieto dve reprezentácie môžu ale aj nemusia byť zhodné. Napríklad je nereálne chcieť po užívateľovi, aby zadal objekt reprezentovaný mohutnou polygonálnou sieťou.

Najpoužívanejšie reprezentácie objektov (zoradené podľa popularity) sú nasledovné:

- mnohouholníkové siete
- slpajny
- CSG reprezentácie
- reprezentácie prerozdeľovaním priestoru
  - oktálne stromy
  - BSP stromy
- implicitné reprezentácie

1. Mnohouholníkové siete (polygonal meshes)

Ide o takzvanú hraničnú reprezentaciu, čo znamená, že reprezentujeme hranicu objektu. Reprezentácia mnohouholníkovými sieťami je jednoznačne mainstreamom v súčasnej počítačovej grafike. Je veľmi jednoduchá a nevyžaduje si hlbšiu analýzu objektov – povrch sa len aproximuje množstvom mnohouholníkov. Na druhej strane je zrejmé, že ide o veľmi neefektívny spôsob ukladania informácií o objekte: čím väčšiu presnosť požadujeme, tým mohutnejšia je množina dát, ktorú treba uložiť, a taktiež strácame akúkoľvek štrukturálnu informáciu (symetria, hladkosť,...). Paradoxne, algoritmy spracúvajúce takto reprezentované objekty sú veľmi rýchle. Dôvodom je práve spomínaná popularita sietí, ktorá podnietila vývoj špeciálnych algoritmov, ich optimalizáciu a dokonca hardverovú pordporu práve pre sieťami reprezentované objekty. Preto nie je zriedkavosťou, že aj iné reprezentácie sa niekedy napríklad pre účely vizualizácie konvertujú do mnohouholníkových sietí.

Často sa mnohouholníkové siete používajú hybridne, napríklad v kombinácii s fotografickou textúrou. Najmä v interakvívnych aplikáciách (hry, virtuálna realita) sa v záujme rýchlosti mnohokrát uspokojíme s veľmi hrubou aproximáciou objektu, na ktorú potom mapujeme textúru. Takto sa textúra stáva súčasťou modelu a nie je len atribútom pri vizualizácii. Tiež tieňovacie algoritmy väčšinou predpokladajú, že objekt reprezentovaný mnohouholníkovou sieťou, teda plochou s hranami, je v skutočnosti hladký, a tak sa pri tieňovaní ostré hrany vyhladzujú.

1.1. **Základné pojmy.**

Definícia 1.1. *Mnohouholníková sieť* je súbor *vrcholov* (t.j. konečnej množiny bodov), *hrán* (t.j. úsečiek, ktorých koncové body sú vrcholy) a *stien* (t.j. konvexných rovinných mnohouholníkov, ktorých hranicu tvoria hrany), pričom sú splnené podmienky

- každý vrchol je koncovým bodom aspoň jednej hrany,
- každá hrana je časťou hranice aspoň jednej steny,
- ak majú dve steny neprázdny prienik, je ním prvok siete (vrchol, hrana).

Najčastejšie sa používajú *trojuholníkové siete*, teda také, kde všetky steny sú trojuholníkmi. Niektoré grafické systémy dokonca podporujú výlučne takéto siete. Nie je to žiadne principiálne obmedzenie, pretože každý mnohouholník sa dá triangulovať, dokovca bez pridávania nových vrcholov. Navyše pri trojuholníkových sieťach nenastáva žiaden problém s planaritou stien – tri vrcholy vždy ležia v rovine, čo však už nemusí byť pravda, ak je stena určená viacerými vrcholmi. Výchylky z roviny môžu nastať napríklad ako dôsledok zaokrúhľovania.

Definícia 1.2. *Hranicou* siete je zjednotenie hrán, ktoré patria iba jednej stene. Mnohouholníková sieť je *uzavretá*, ak nemá hranicu.

Definícia 1.3. Mnohouholníková sieť sa nazýva *súvislá*, ak je súvislý graf, ktorý dostaneme, keď steny siete nahradíme vrcholmi grafu, a dva vrcholy grafu budú spojené hranou vtedy, ak príslušné steny mali spoločnú hranu siete.

Definícia 1.4. Mnohouholníková sieť sa nazýva *varieta* (angl. *manifold*), ak každý vnútorný bod siete má okolie topologicky izomorfné s okolím bodu v rovine. (Okolím bodu siete myslíme prienik jeho okolia v priestore so sieťou).

Ak je sieť varietou, tak každná hrana tvorí hranicu nanajvýš dvoch stien. (Tiež dostávame podmienku pre vrcholy, ale už je zložitejšie neformálne ju popísať.)

Definícia 1.5. *Orientáciu* mnohouholníka volíme vymenovaním jeho vrcholov pozdĺž hranice. Dva susedné mnohouholníky, čiže také, ktoré majú spoločnú hranu, sú orientované *konzistentne*, ak sú v každom z nich vrcholy pozdĺž spoločnej hrany vymenované v opačnom poradí. Mnohouholníková sieť, ktorá je varietou, sa nazýva *orientovateľná*, ak môžme v každom mnohouholníku zvoliť orientáciu tak, že všetky dvojice susedných mnohouholníkov sú orientované konzistentne.

1.2. **Dátové štruktúry.** Najjednoduchšie možno mnohouholníkovú sieť reprezentovať vymenovaním jednotlivých prvkov. Mali by sme teda tabuľku alebo zoznam vrcholov, kde pre každý vrchol by boli zapísané jeho súradnice, ďalej tabuľku hrán, v ktorej by sme pre každú hranu uložili smerníky, indexy alebo mená jej koncových vrcholov, a napokon tabuľku stien, ktorá by pre každú stenu obsahovala hrany hranice (smerníky, mená,...).

Pre mnohé účely je táto reprezentácia výhodná. Obsahuje všetky informácie potrebné na popísanie siete a nekladie žiadne obmedzenia na typ a vlastnosti (orientovateľnosť, varietovosť a podobne). Avšak modifikácia siete vyžadujúca zložitejšie zmeny než len zmeny súradníc jednotlivých vrcholov je ťažkopádna, pretože reprezentácia neposkytuje dostatočne pohodlný prístup k okoliam: ak chceme zistiť, s ktorými hranami alebo stenami je niektorý vrchol incidentný, musíme prehladať príslušné zoznamy.

1.2.1. *Okrídlené hrany (winged edges).* Ide o zoznamy (tabuľky) vrcholov, hrán a stien, pričom podstatná informácia o sieti je uložená v zozname hrán, zvyšné dve štruktúry sú pomocné. Každú hranu si zorientujeme a do šturtúry o nej uložíme nasledujúce informácie:

- začiatočný vrchol (smerník naň alebo iný identifikátor)
- koncový vrchol
- ľavá stena
- pravá stena
- ľavý traverz:
  - hrana ľavej steny incidentná so začiatočným vrcholom

  - hrana ľavej steny incidentná s koncovým vrcholom
- pravý traverz:
  - hrana pravej steny incidentná so začiatočným vrcholom
  - hrana pravej steny incidentná s koncovým vrcholom

V prípade statickej štruktúry samozrejme ukladáme aj jedinečný identifikátor hrany. Každá hrana je teda v tabuľke uložená ako orientovaná úsečka, táto orientácia však nijak nesúvisí s orientáciou siete, slúži len na zoradenie dát vrámci štruktúry pre hranu.

Zoznam vrcholov pre každý vrchol obsahuje smerník na hranu siete, ktorá tento vrchol obsahuje. V prípade, že je takých hrán viacero, môže ukazovať na ktorúkoľvek z nich. Podobne v zozname stien je pre každú stenu uložený smerník na niektorú z hrán jej hranice.

1.2.2. *DCEL (double connected edge list).* Pomocou tejto štruktúry môžme reprezentovať iba orientovateľné plochy. Najprv si celú mnohouholníkovú sieť orientujeme. Ďalej každú vnútornú hranu nahradíme dvojicou *polhrán* – budú to dve navzájom opačne orientované úsečky, ktoré by boli po „zabudnutí" orientácie obe totožné s pôvodnou hranou. Každá polhrana potom prislúcha iba jednej stene, a síce tej, kde jej orientácia naznačuje poradie menovania vrcholov v stene. Hrany na hranici siete nahradíme polhranami príslušnej orientácie.

Podobne ako v predchádzajúcom prípade máme zoznam polhrán, ktorý obsahuje celú informáciu o šturktúre siete. Predpokladajme, že steny siete sme si orientovali proti smeru chodu hodinových ručičiek. Potom pre každú polhranu sú v zozname zaznačené

- smerník na začiatočný vrchol
- smerník na stenu naľavo od polhrany (stena, ku ktorej polhrana prislúcha)
- smerník na opačnú polhranu
- smerník na polhranu vychádzajúcu z koncového vrchola, tvoriacu pokračovanie hranice ľavej steny.

Okrem zoznamu polhrán máme aj pomocné štruktúry: zoznam vrcholov a zoznam stien. Zoznam vrcholov pre každý vrchol obsahuje smerník na niektorú polhranu, ktorá z neho vychádza, a zoznam stien zas pre každú stenu jednu z polhrán tvoriacich jej hranicu.

Obe štrukrúry, okrídlené hrany aj polhrany, poskytujú rýchly prístup k okoliam, čo je v prípade obrovského množstva vrcholov veľká výhoda. Preto umožňujú efektívnu reprezentáciu dynamických sietí, keď sa v čase mení nielen poloha jednotlivých vrcholov, ale aj topológia siete. Na druhej strane sú komplikovanejšie než na úvod spomenutá jednoduchá reprezentácia siete. Pri niektorých aplikáciách sa výhody poskytované okrídlenými hranami a polhranami vôbec nevyužijú, napríklad ak sa sieť vytvorí len kvôli vizualizácii objektu reprezentovaného pôvodne iným spôsobom. Vtedy je samozrejme výhodnejšie siahnuť po jednoduchších reprezentáciách.

Tiež si treba uvedomiť, že spomínané štruktúry kladú aj obmedzenia na sieť, ktorú je možné reprezentovať: v oboch prípadoch sa štuktúry používajú len pre orientovateľné variety. Ide však len o základné dátové štruktúry, ktoré sa pre potreby aplikácie dajú modifikovať, aby sa dali reprezentovať aj niektoré nevarietové povrchy.

1.3. **Získavanie dát pre mnohouholníkovú sieť.** Vkladať dáta pre túto reprezentáciu priamo je cesta schodná len pre veľmi jednoduché objekty. Je zrejmé, že pomocou sietí sa objekty reprezentujú predovšetkým vnútri systému, v žiadnom prípade nejde o reprezentáciu vhodnú pre komunikáciu s užívateľom. Často sa polygonálne siete získajú prevodom z iných typov reprezentacií, ako si neskôr ukážeme. Tu si uvedieme niektoré špeciálne konštrukcie.

1.3.1. *Využitie parametrizácie.* Všimnime si najprv, že parametrizovaná plocha sa veľmi ľahko aproximuje mnohouholníkovou sieťou. Pod *parametrizovanou plochou* rozumieme vektorovú funkciu o dvoch premenných

$$f\colon D \to \mathbb{R}^3 \quad (D \subset \mathbb{R}^2), \quad (u,v) \mapsto (f_x(u,v), f_y(u,v), f_z(u,v)),$$

a predpokladajme navyše, že táto táto funkcia je injektívna, teda že výsledná plocha nemá žiadne samoprieseky. Rozdeľme definičný obor D na obdĺžniky: napríklad ak D je súčin intervalov

$\langle u_{min}, u_{max}\rangle \times \langle v_{min}, v_{max}\rangle$, rozdelíme každý interval na niekoľko podintervalov s hranicami

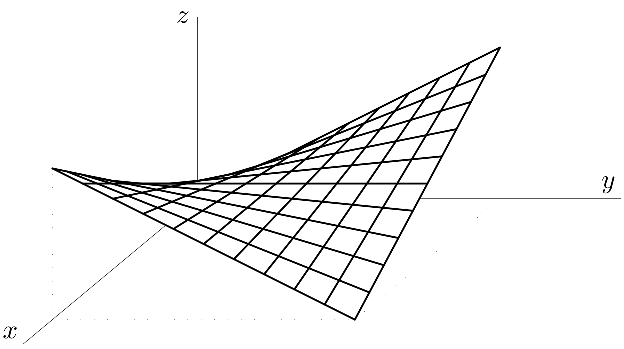
$$u_{min} = u_0 < u_1 < \cdots < u_n = u_{max},$$
$$v_{min} = v_0 < v_1 < \cdots < v_m = v_{max}.$$

Získame tak sieť obdĺžnikov určených vrcholmi $(u_{i-1}, v_{j-1}), (u_i, v_{j-1}), (u_i, v_j), (u_{i-1}, v_j)$. Obrazy vrcholov týchto obdĺžnikov budú potom tvoriť vrcholy „štvoruholníkov" sieťovej reprezentácie plochy. Často je vhodné tieto „štvoruholníky" ešte rozdeliť na dva trojuholníky, keďže vo všeobecnosti obrazy štyroch vrcholov jednotlivých obdĺžnikov nemusia ležať v jednej rovine.

Príklad 1.6. Reprezentujme mnohouholníkovou sieťou plochu parametrizovanú nasledovne:

$$x = u, \quad y = v, \quad , z = \frac{u+v}{2} - uv,$$

kde $u \in \langle 0, 1\rangle, v \in \langle 0, 1\rangle$.



OBR. 1. Parametrizovaná plocha

Intervaly v smere u aj v rozdelíme, čím získame sieť obdĺžnikov. Vypočítame súradnice obrazov všetkých vrcholov, a tak máme súradnice vrcholov mnohouholníkovej siete. Ostáva už len vrcholy pospájať do mnohouholníkov. V príklade na obrázku je plocha vykreslená pomocou „štvrouholníkov", treba si však uvedomiť, že tieto útvary nie sú rovinné! Preto ak by sme chceli aplikovať na náš objekt zložitejšie algoritmy, je vhodné každý „štvoruholník" rozdeliť najprv na trojuholníky.

Spôsobov, ako získať parametrizovanú plochu, je mnoho, spomeňme si aspoň tie najjednoduchšie.

1.3.2. *Translačné plochy.* Plochu dostaneme tak, že nejakú zvolenú krivku posúvame v priestore, výsledná plocha je potom zjednotenie všetkých trajektórií všetkých bodov krivky. Krivku môžme posúvať v priestore po priamke, alebo všeobecnejšie po ďalšej krivke. Nech krivka C (krivka, ktorú chceme posúvať) je daná parametricky

$$x = c_x(u), \quad y = c_y(u), \quad z = c_z(u), \tag{1}$$

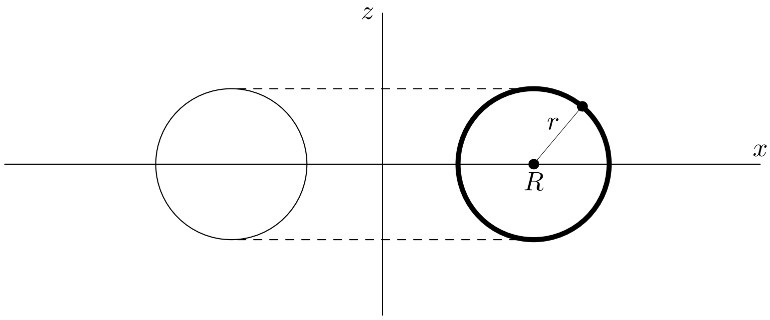
kde u je parameter pohybujúci sa v intervale $\langle u_0, u_1\rangle$. Trajektóriou posúvania nech je krivka T, tiež zadaná parametricky:

$$x = t_x(v), \quad y = t_y(v), \quad z = t_z(v), \quad v \in \langle v_0, v_1\rangle,$$

čiže vektor $(t_x(v) - t_x(v_0),\ t_y(v) - t_y(v_0),\ t_z(v) - t_z(v_0))$ je vektorom posunutia v „čase" v. Výsledná plocha je teda parametrizovaná

$$\begin{aligned}
x &= f_x(u, v) = c_x(u) + t_x(v) - t_x(v_0)\\
y &= f_y(u, v) = c_y(u) + t_y(v) - t_y(v_0)\\
z &= f_z(u, v) = c_z(u) + t_z(v) - t_z(v_0), \quad u \in \langle u_0, u_1\rangle, v \in \langle v_0, v_1\rangle.
\end{aligned}$$

1.3.3. *Rotačné plochy.* Pri konštrukcii *rotačnej plochy* z krivky (1) popisujeme plochu, ktorá vznikne jej rotáciou okolo niektorej osi. Namiesto všeobecných vzorcov si uveďme konkrétny príklad: Popíšeme torus ako rotačnú plochu, keď kružnicu v rovine xz rotujeme okolo osi z, viď obrázok. Kružnica v rovine xz je parametrizovaná krivka



Obr. 2. Konštrukcia torusu

$$x = R + r\cos u, \quad y = 0, \quad z = r\sin u,$$

kde r je polomer kružnice, R je vzdialenosť jej stredu od začiatku súradníc, a u je parameter, ktorý sa pohybuje v intervale $\langle 0, 2\pi)$. Každý bod tejto krivky chceme rotovať okolo osi z. Dosadíme teda parametrizáciu krivky do rovníc rotácie a dostávame parametrizáciu plochy:

$$\begin{aligned} x &= (R + r\cos u)\cos v \\ y &= (R + r\cos u)\sin v \\ z &= r\sin u, \qquad\qquad \text{kde } u, v \in \langle 0, 2\pi\rangle\,. \end{aligned}$$

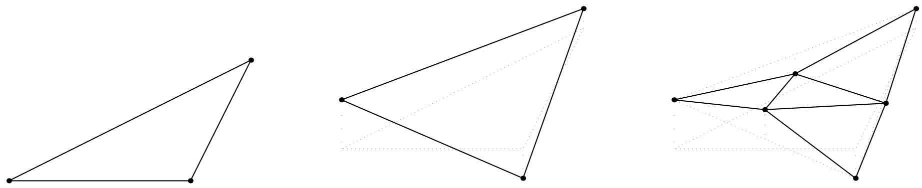
1.3.4. *Šablónovanie (sweeping).* Pri popise plochy metódou šablónovania je východiskom rovinná krivka, ktorou nejakým spôsobom pohybujeme v priestore pozdĺž zadanej trajektórie spravidla tak, že rovina obsahujúca krivku sa je vždy kolmá na momentálny smer pohybu. Výsledná plocha je zjednotením trajektórií všetkých bodov krivky. Pri takomto pohybe stále ostáva ešte jeden stupeň voľnosti, a síce otočenie v rovine krivky okolo jej prieniku s trajektóriou Taktiež sa spravidla pripúšťa, že krivka počas pohybu spojito mení svoj tvar.

Šablónovanie je teda pomerne všeobecný spôsob popisu plochy. Niekedy sa šablónovanie samo považuje za spôsob reprezentácie. Inokedy sa šablónovanie môže použiť ako reprezentácia pre užívateľské rozhranie, ktorá sa následne vnútri systému prevedie na mnohouholníkovú sieť.

1.3.5. *Procedurálne modelovanie.* Z mnohých špeciálnych metód na generovanie objektov si spomeňme ešte využitie tzv. Brownovho pohybu na generovanie terénu. Cieľom je získať mnohouholníkovú sieť, ktorá čo najdôveryhodnejšie predstavuje nejaký členitý terén.

Východiskom je napríklad rovná horizonlálna plocha v tvare trojuholníka. V prvej iterácii algoritmu náhodne zmeníme súradnice vrcholov tohto trojuholníka. V ďalšom kroku každú stranu trojuholníka rozdelíme novým vrcholom na dve časti, a tým sa pôvodný trojuholník rozdelí na štyri menšie. Následne náhodne zmeníme súradnice nových vrcholov. Iteratívne pokračujeme v zjemňovaní siete a v náhodnom posúvaní vrcholov. Namiesto trojuholníka môžme vychádzať aj z obdĺžnika, ktorý v každom kroku prerozdelíme na štyri časti.

Metóda, ktorú používame pri posúvaní vrcholov, už závisí od aplikácie. Ak chceme terén rovnomerne členitý, používame stale ten istý generátor náhodných čísel. Treba ešte doriešiť, ktorým smerom sa budú vrcholy posúvať. Najjednoduchšie by bolo meniť vždy len výšku bodu a zvyšné dve súradnice ponechať (viď obrázok). Takto vygenerovaný terén však nemusí byť veľmi uspokojivý, napríklad nikdy tak nenamodelujeme skalné previsy. Pre tie by bolo vhodné, keby sme nový bod ležiaci na strane trojuholníka posúvali v smere kolmom na pôvodnú plochu. Vtedy však stojíme pred problémom, že keď bod leží na spoločnej strane dvoch susedných trojuholníkov,



Obr. 3. Generovanie terénu

tak máme dva rôzne smery, ktorými by sme chceli bod posunúť. (Pre viac ilustrácií pozri napr. `http://local.wasp.uwa.edu.au/~pbourke/fractals/planets/`).

1.3.6. *Rekonštrukcia reálnych dát.* Často sa snažíme dáta pre reprezentáciu objektu získať nie umelým modelovaním, ale akýmsi kopírovaním skutočného objektu. K tomu si môžme pomôcť rôznymi 3D skenermi, ktoré zosnímajú vložený objekt a ako výstup nám dajú súradnice množstva bodov z povrchu objektu. Potom už len zostáva pospájať tieto body hranami do mnohouholníkovej siete (metódy v diskrétnej geometrii) a prípadne, ak niektorá časť objektu je zosnímaná slabšie, vhodne tieto diery vyplniť dodatočnými bodmi (metódy v geometrickom modelovaní).

Iný spôsob je rekonštrukcia trojrozmerných dát z dvojrozmerných snímkov (fotogrametria, počítačové videnie). Napríklad máme dva priemety (dve fotografie, z rôznych strán) jedného objektu. Užívateľ identifikuje niekoľko kľúčových bodov na týchto snímkach, napríklad niektoré rohy krabice, a potom sa už automaticky snažíme zrekonštruovať celý objekt. V prípade, že máme k dispozícii viacero priemetov objektu, snažíme sa samozrejme o vyšší stupeň automatizácie. Opačne, môžme tiež chcieť zrekonštruovať objekt, z ktorého máme k dispozícii iba jeden priemet. Vtedy si musíme pomôcť rôznymi dodatočnými informáciami (vieme niečo o symetrii objektu, o osvetlení a podobne).

1.4. **Modifikácia mnohouholníkových sietí.** Ak chceme pozmeniť objekt reprezentovaný mnohouholníkovou sieťou, znamená to, že v lepšom prípade treba zmeniť súradnice obrovského množstva vrcholov, v horšom prípade aj topológiu siete (spôsob, ako sú vrcholy pospájané hranami a stenami). Takéto modifikácie si vyžadujú istú dávku samostatnosti a tvorivosti v každom konkrétnom prípade.

Najjednoduchšia modifikácia objektu je použitie afinných transformácií. Je jednoduché objekt posunúť, otočiť, škálovať a podobne, stačí príslušnú afinnú transformáciu aplikovať na všetky vrcholy mnohouholníkovej siete. Pochopiteľne, často je tento nástroj nedostatočný. Ak chceme previesť na objekte nejakú inú deformáciu, musíme si navrhnúť vhodné transformačné rovnice.

Príklad 1.7. Majme namodelovaný kváder s rozmermi napríklad výška 5 cm, šírka 1 cm a hrúbka 0.2 cm tak, že zvislá z-os prechádza stredom kvádra. Chceme ho zdeformovať do skrutky, čiže budeme ho otáčať okolo z-osi tak, že čím sa bod kvádra nachádza vyššie (má väčšiu z-súradnicu), tým väčší uhol otočenia použijeme. Máme tak rovnice pre skrútenie

$$\begin{aligned} x' &= x\cos(\varphi(z)) - y\sin(\varphi(z)) \\ y' &= x\sin(\varphi(z)) + y\cos(\varphi(z)) \\ z' &= z, \end{aligned}$$

kde $\varphi(z)$ je nejaká rastúca funkcia, napríklad $\varphi(z) = z$, (vstup je v centimetroch, výsledok v radiánoch). Treba už len reprezentovať kváder dostatočne hustou mnohouholníkovou sieťou a aplikovať transformačné rovnice na každý vrchol.

Podobne sa dá jednoducho aplikovať škálovanie, keď chceme objekt zúžiť v smeroch x a y, ale len v niektorej časti. Pomôžeme si rovnicami pre škálovanie, len škálovací faktor (podobne ako uhol v predchádzajúcom príklade) nebude konštantný, ale bude závisieť od výšky bodu.

KAGDM FMFI UK Bratislava
E-mail address: `jana.pilnikova@fmph.uniba.sk`