RENDERING OF WATER SURFACE

Marek Kopecký

Bachelor Degree Programme (3), FIT BUT

E-mail: xkopec36@stud.fit.vutbr.cz

Supervised by: Adam Herout

E-mail: herout@fit.vutbr.cz

Abstract: This paper deals with rendering of water, which is a key part of many computer games and simulations. The resulting application runs fluently and is powered by XNA Framework. The simulation includes reflection. Foam is rendered on the bank of water. Water waves are computed using the FFT method.

Keywords: rendering, water, reflection, Fresnel term, foam, FFT

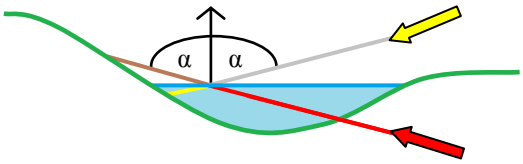
1. ÚVOD

Cílem tohoto příspěvku je popsat vytváření grafické simulace vody v reálném čase. Voda má vlny, propouští a odráží světlo z okolního prostředí. Na přelomu terénu a vody je v grafické simulaci vykreslována jednoduchá pěna. Podobné simulace jsou v různých počítačových hrách.

Aplikace je psána v jazyku C#, k vykreslování je použit Framework XNA.

2. ODRAZ A LOM SVĚTLA

XNA Framework využívá k vykreslení scény mimo jiné dva údaje. Pozici kamery ve virtuálním světě a bod, na který se kamera dívá. Výsledný vykreslený obraz je tedy výstup virtuální kamery.



Obrázek 1: Schéma odrazu a lomu.

Na obrázku 1 představuje žlutá šipka kameru zobrazující scénu. Na tuto kameru dopadá světlo (šedá čára). Na vodní hladině se však mísí [4] barva terénu pod vodní hladinou (žlutá čára) a barva terénu odraženého od vodní hladiny (hnědá čára). Úhel dopadu světla je roven úhlu odrazu. K zachycení barvy odraženého terénu používám kameru znázorněnou červenou šipkou. Tato kamera je umístěna pod vodou ve stejné vzdálenosti a ve stejném úhlu od vodní hladiny jako kamera zobrazující scénu. Předměty se pro ni vykreslují pouze nad vodní hladinou. Paprsek světla, který projde vodou, by měl správě lomit svůj směr. Pro zjednodušení však lom světla neimplementuji a předpokládám, že paprsek světla projde vodou bez lomu.

2.1. FRESNEL TERM

Každý pixel (zobrazený bod) vody se bude skládat z barvy předmětů odražených (viz obrázek 2 vlevo) a barvy předmětů pod vodou (viz obrázek 2 uprostřed). Je však třeba zjistit, jakou intenzitou bude každá barevná složka zastoupena. Když se do vody díváme kolmo, vidíme pouze barvu pod vodou. Se zvětšujícím se úhlem pohledu k normále vody (viz obrázek 3 vpravo) se zvětšuje zastoupení barvy odrazu.

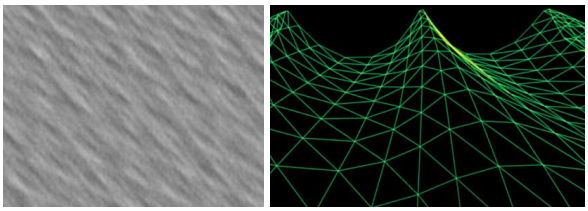
Obrázek 2: Odraz světla (vlevo), propuštění světla (uprostřed), Fresnel term (vpravo).

Indikátor sklonu [3] pohledu kamery k vodě je skalární součin *s* mezi vektorem pohledu kamery *e* a normálou vody *n* (platí $s = e \cdot n$). Čím je výsledek skalárního součinu *s* menší, tím více má na barvě vody *c* zastoupení barva odrazu *o* a tím méně má na barvě vody *c* zastoupení barva *p*, která projde vodou. Tento postup (viz rovnice 1) je jednoduchá aproximace metody Fresnel term [5], která slouží k přesnému výpočtu poměru barev. V mé simulaci je Fresnel term počítán na GPU v pixel shaderu.

$$c = o \cdot (1 - s) + p \cdot s \qquad (1)$$

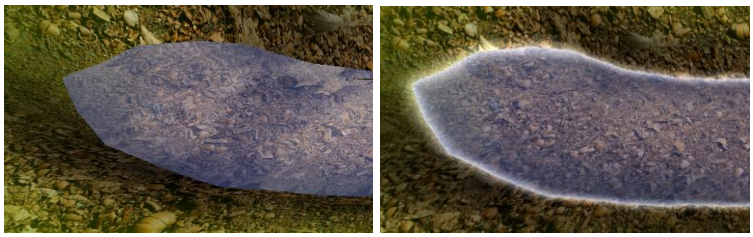
3. VLNY

Na venkovních vodních plochách (např. oceán, jezero) se objevují vlny. Aplikace, která slouží k vykreslování vody, by tedy vlny měla vykreslovat. Existuje hodně algoritmů, jak vlny vytvářet. Většina algoritmů vytvoří výškovou mapu, na základě které se vytvoří 3D model vodní hladiny. Čím je bod na výškové mapě světlejší, tím výš je umístěna vodní hladina v daném bodě. Vodní hladina se pak skládá ze sítě polygonů. Tento postup je znázorněn na obrázku 3.



Obrázek 3: Ukázka výškové mapy vlny (vlevo) a sítě vodní hladiny (vpravo) [1].

Ve své práci jsem se rozhodl pro podobný přístup. Program vytváří výškovou mapu, kterou následně převádí na bump mapu. Mapuje ji na dva trojúhelníky. Bump mapa je textura, která se používá k vytvoření zdání 3D prostoru. Tato iluze je vytvářena světlem, na každý pixel trojúhelníku působí světlo jinak. Tento postup zaručuje, že na grafickou kartu jsou kladeny daleko menší výpočetní nároky. Nevykresluje totiž síť trojúhelníků (viz obrázek 3 vpravo), ale vykreslí pouze dva trojúhelníky představující čtverec. Bump mapa je aplikována na vodní povrch po nanesení barvy odraženého světla a barvy propuštěného světla. Aplikování bump mapy se děje v pixel shaderu na GPU.



Obrázek 4: Voda bez pěny (vlevo) a voda s pěnou (vpravo).

Můj postup je znázorněn na obrázku 4. Skutečnost, že vodní hladina je všude stejně vysoká, je očividná na přelomu vody a terénu (viz obrázek 4 vlevo). Proto jsem ještě dodatečně implementoval jednoduché vykreslování pěny (viz obrázek 4 vpravo), které tento problém zamaskuje. Výsledkem je tedy voda se zdáním skutečných vln, ale pozorovatel pravděpodobně nepozná, zdali se jedná o zdání, nebo skutečné 3D vlny. Vykreslení pěny se prování v pixel shaderu formou post-procesingu.

3.1. MATEMATICKÝ POPIS VLN

Pro správné zobrazení vln je nejdříve třeba vytvořit výškovou mapu vln (viz obrázek 3 vlevo). Pro její tvorbu existuje řada algoritmů. Algoritmy [2] jsou většinou založeny na rychlé Fourierově transformaci (FFT).

Ve své práci jsem se rozhodl použít statický model vln. Oceánografická literatura je často používá pro simulace v kombinaci s experimentálním pozorováním. Výška vlny je zde považována za náhodnou proměnnou horizontální polohy a času (viz rovnice 2). Další zajímavostí je, že u statických modelů je možné výškovou mapu vyjádřit jako součet sinus a cosinus funkcí. Součet se provádí pomocí FFT.

$$h(p,t) = \sum_k \tilde{h}(\mathbf{k},t)\, e^{i\mathbf{k}\cdot p} \qquad (2)$$

V rovnici (2) je p horizontální poloha, platí $p = (x, z)$. Dále t představuje čas, $\boldsymbol{k}$ je vektor se složkami $\mathbf{k} = (k_x, k_z)$, $k_x = 2\pi n / L_x$, $k_z = 2\pi m / L_z$, pro n platí $-N/2 \leq n < N/2$0, pro m platí $-M/2 \leq m < M/2$0. FFT generuje výškovou mapu v diskrétním bodě $\mathbf{x} = (nL_x / N, mL_z / M)$. $\tilde{h}(k, t)$ představuje počáteční výšku amplitud Fourierových složek, určuje strukturu povrchu [2].

V mé simulaci byla použita inverzní diskrétní Fourierova transformace pro výpočet výškové mapy vln. Výšková mapa je počítána z Phillipsova spektra [2]. Tento výpočet probíhá na CPU.

4. ZÁVĚR

V tomto článku bylo představeno komplexní řešení vodní simulace. Testování jsem provedl na notebooku s procesorem AMD Athlon(tm) X2 Dual-Core QL-62, pamětí DDR2 4GB a grafickou kartou NVIDIA GeForce 9600M GT. Při testování bylo vykreslováno 60 snímků za sekundu při rozlišení 1680x1050. Vykreslení vodní plochy trvá v průměru 4,5 ms. Důraz byl kladen na vlny a jejich matematický popis. Součástí simulace je odraz světla. Pro jednoduchost světlo vodou projde, ale neláme se. Simulace vody je použita v počítačové hře, kde společně s ostatními herními prvky vytváří dojem reálného prostředí. Dalším pokračováním projektu by mohlo být přidání vodopádů nebo vody tříštící se o skálu.

PODĚKOVÁNÍ

Výsledná aplikace je počítačová hra, která v sobě zahrnuje řadu herních prvků. Některé jsem implementoval já (např. terén a vodu). Na implementaci hry se mnou spolupracoval spolužák Jakub Dražka. Design herního menu a 3D modelů provádí Lucie Zemanová (VUT FSI). Tvorbu 3D modelů zajišťuje Jan Dražka. Výsledná aplikace vzniká pro soutěž Imagine Cup, kterou pořádá firma Microsoft. Všem výše jmenovaným bych chtěl tímto poděkovat za spolupráci.

REFERENCE

[1] Finch, M.: Chapter 1. Effective Water Simulation from Physical Models. GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics. Addison-Wesley Professional, 2004, ISBN 0-321-22832-4

[2] Tessendorf, J.: Simulating Ocean Water. SIGGRAPH 2001 Course Notes, 2001

[3] Grootjans, R.: XNA 3.0 Game Programming Recipes: A Problem-Solution Approach. Apress, Berkely, CA, USA, 2009, ISBN 978-1-4302-1855-5

[4] J. Fuka, B. Havelka: Optika, SPN 1961

[5] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F.: Computer Graphics – Principles and Practice (2nd Ed.). Addison-Wesley, 1990, ISBN 0-201-12110-7