**Metody Renderování Vodních Jevů v Reálném Čase**

**Real-time Water Rendering Techniques**

Jan Tobola

Aplikovaná informatika, 2. ročník

[jan.tobola@uhk.cz](mailto:jan.tobola@uhk.cz), [www.uhk.cz](http://www.uhk.cz)

Obsah

[Úvod 3](#_Toc325372770)

[Fyzikální jevy 3](#_Toc325372771)

[Odrazivost vodní hladiny 3](#_Toc325372772)

[Propustnost a lom světla 3](#_Toc325372773)

[Fresnelův efekt 4](#_Toc325372774)

[Kaustika 4](#_Toc325372775)

[Disperze a efekt hlubiny 4](#_Toc325372776)

[Vlnový model 5](#_Toc325372777)

[Height fields 5](#_Toc325372778)

[Perlinův šum 5](#_Toc325372779)

[Fast Fourier Transformation (FFT) 6](#_Toc325372780)

[Navier-Stokesovy rovnice (NSE) 6](#_Toc325372781)

[Částicové systémy 6](#_Toc325372782)

[Jiné algoritmy 6](#_Toc325372783)

[Závěr 7](#_Toc325372784)

[Odkazy 8](#_Toc325372785)

# Úvod

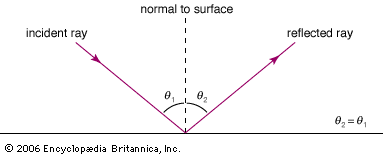
Se vzrůstajícím výpočetním výkonem dnešních počítačů je výzva vytvářet simulace a animace vodních jevů v reálném čase, počítané především na grafických kartách, které se stále více snaží vyvolat realistický dojem. Existuje několik grafických algoritmů a metod, které tento dojem navyšují, nicméně některé fyzikální aspekty jsou natolik výpočetně náročné, že je zatím není možné počítat v reálném čase.

# Fyzikální jevy

Je zde několik základních i složitých fyzikálních jevů, které nesmí být u renderování vodní hladiny opomenuty.

## Odrazivost vodní hladiny

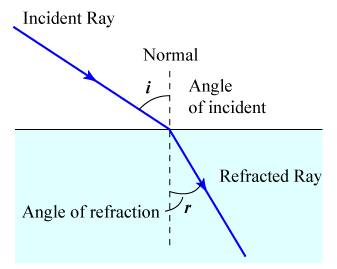
Jedním z nejhlavnějších a nejjednodušších jevů, který se zde vyskytuje, je odrazivost vodní hladiny *(reflection)*, kde dochází k odrážení okolního světla. Paprsky okolního světla dopadají na hladinu pod určitým úhlem a pod stejným úhlem také hladinu opouštějí, tedy platí zde pravidlo, že úhel dopadu se rovná úhlu odrazu jako je znázorněno na obrázku 1.



Obr. 1 Popis odrazu dopadajícího paprsku

## Propustnost a lom světla

Dalším velice důležitým jevem je propustnost a lom světla *(refraction)*, kdy se světlo procházející vodní hladinou zlomí o určitý úhel díky přechodu světla z opticky řidšího do opticky hustšího prostředí či naopak. O jaký úhel se mají paprsky po přechodu z jednotlivých prostředí zlomit, určuje index lomu, který je pro vodu definován hodnotou 1,33.



Obr. 2 Popis lomu paprsku při přechodu mezi dvěma prostředími. *Zdroj: one-school.net*

## Fresnelův efekt

Kombinaci předešlých dvou jevů závisející na pozorovacím úhlu pozorovatele popisuje tzv. Fresnelův efekt *(Fresnel term, Fresnel effect)*. Pokud je pozorovací úhel k hladině spíše kolmý, voda se tak stává pro pozorovatele více průhledná a méně odrazivá, zatímco v případě velmi ostrého pozorovacího úhlu se voda bude tvářit jako jedno velké zrcadlo a průhledná nebude téměř vůbec.

## Kaustika

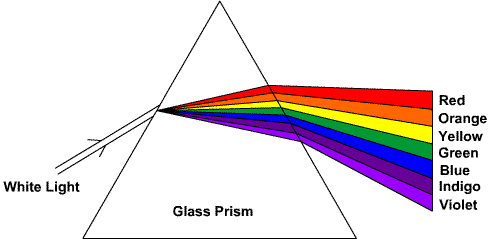
V neposlední řadě velmi důležitou součástí vody je efekt kaustiky *(caustic)*. Kaustika je jakási obálka světelných paprsků zlomených přechodem mezi různými prostředími nebo odražených jiným předmětem. V tomto případě se jedná o promítnutý obraz této obálky na dně určitého prostředí, v němž se voda nachází. Přesný výpočet kaustiky v reálném čase je zcela nemožný, proto se efekt kaustiky nahrazuje potažením dna animovanou texturou přibližně odpovídající pohybu vln.

## Disperze a efekt hlubiny

Jsou zde i další jevy jako je efekt hlubiny nebo disperze *(dispersion)*, tyto jevy jsou samozřejmě nedílnou součástí skutečného vodního prostředí, avšak lidskému mozku již nedávají hlavní signál, že s vyrenderovanou vodní hladinou není něco v pořádku, proto jsou tyto jevy mnohdy v počítačových hrách a animacích přehlíženy na úkor snížení nároků na výpočetní výkon.

Se zvětšující se hloubkou resp. masou vody se hladina zabarvuje zpravidla do tmavé modré barvy a stupeň zabarvení hladiny závisející na vzdálenosti ode dna je většinou logaritmický.

Disperze je cizí výraz pro rozklad bílého světla na jednotlivé barvy spektra v závislosti na různé vlnové délce jednotlivých barevných složek. Tento efekt se u renderování vody vyskytuje velmi zřídka.



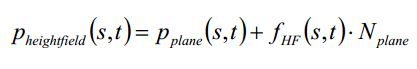
Obr. 3 Rozklad bílého světla na barvy spektra. *Zdroj: school-for-champions.com*

# Vlnový model

To jak vznikají a jak se pohybují vlny na hladině, udává vlnový model *(vawe model)*. Je několik možných způsobů jak vlnový model implementovat. Nejznámější a nejvíce používané techniky jsou pomocí *height fields*, známé také jako *heightmap* za použití Perlinova šumu *(Perlin noise)* nebo *Fast Furier Transofrmations*. Oba tyto algoritmy generují velmi přirozený a věruhodný pohyb vln a vzhledem k jejich malé výpočetní náročnosti se hodí i na renderování velkých ploch jako třeba oceány. Dalším možným způsobem je aplikace Navier-Stokesových rovnic (Navier-Stokes Equations - NSE), které vypadají pro simulování vody velmi realisticky.

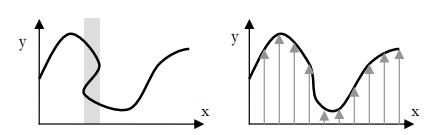
## Height fields

Povrch vodní hladiny může být reprezentován pomocí height field. Jedná se o funkci o dvou proměnných, která vrací hodnotu výšky pro zadaný bod v dvourozměrném prostoru. Rovnice na obrázku 4 znázorňuje, jak height field může být použito k přenesení roviny do třírozměrného prostoru.



Obr. 4 z [1]

Použití této metody získává svá určitá omezení ve srovnání se skutečnou vodní hladinou. Vzhledem k faktu, že tato funkce vrací pouze jednu hodnotu výšky pro zadaný bod, je tudíž nemožné generovat lámané vlny, kdy jedna část hladiny se v určitý okamžik překrývá s jinou, jak je demonstrováno na obrázku 5.

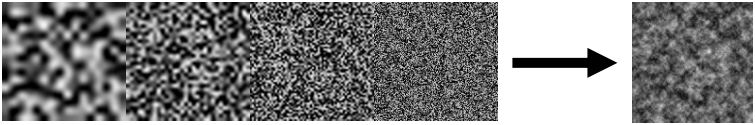


Obr. 5 z [1] Ukázka omezení height fields.

Stejně tak má aplikace height fields hned několik výhod. Tato metoda je snadná na použití a je jednoduché uchovávat její data, nejčastěji se data uchovávají v podobě textury *(heightmap)*. Renderování vodní hladiny za pomoci heightmapy je také známo pod pojmem *displacement-mapping*.

## Perlinův šum

K tomu, aby mohl být vytvořen Perlinův šum je potřeba tzv. bílý šum. Perlinův šum není až tolik sám o sobě zajimavý. Více zajímavý a důležitý je fraktální šum *(fractal noise)*, který se získá součtem několika těchto šumových funkcí s různou frekvencí a amplitudou, tyto jednotlivé funkce, které se skládají dohromady, jsou nazývány oktávy *(octaves)*. Frekvence následující oktávy je dvojnásobkem frekvence oktávy předcházející. Skládání jednotlivých oktáv ve 2D znázorňuje obrázek 6. Počet z kolika oktáv bude výsledný fraktální šum složen, je libovolný. Plynulých a realisticky vypadajících přechodů vln lze dosáhnout interpolací jednotlivých bodů v čase. Perlinův šum se dále v praxi využívá ke generování dalších náhodných přirozeně vypadajících obrazců (např. oblaka, dřevo).



Obr. 6 z [1] Součet čtyř Perlinových šumových funkcí v jeden fraktální šum.

## Fast Fourier Transformation (FFT)

FFT algoritmus dosahuje nejlepších výsledků a byl i několikrát použit komerčně (ve filmech Titanic a Vodní svět). FFT používá statistický model z pozorování skutečných moří a oceánů, popisující chování vln v závislosti na povětrnostních podmínkách, který vyvinuli oceánografové. Tato metoda je více popsána v [2].

## Navier-Stokesovy rovnice (NSE)

Tento algoritmus je definován sadou diferenciálních rovnic, které popisují pohyb nestlačitelných kapalin. NSE jsou velice komplexní a pro účely renderování vody v reálném čase jsou zjednodušovány. Přestože tato metoda dosahuje velice realistických výsledků, je nepraktické použít ji v simulacích rozsáhlých ploch vzhledem k její výpočetní náročnosti. Často se NSE používá v kombinaci s jiným vlnovým modelem, pak se NSE stará pouze o reakci vody v případě kolize s jinými objekty.

# Částicové systémy

Téma částicových systémů *(particle systems)* je kapitolou sama o sobě. Pohybem jednotlivých částic a jejich neustálé působení na okolní částice, s vhodně nastavenými vlastnostmi, lze docílit velmi přesných simulací (např. tekoucí vody). V závislosti na dostupné výpočetní síle lze v reálném čase počítat kolizi jen určitého počtu částic. Částicové systémy mají využití i v jiných oblastech počítačové grafiky.

# Jiné algoritmy

Jsou používané i jiné algoritmy, využívající některé z předešlých metod, které jsou různě optimalizované a jsou specifické pro určitý druh simulací. Jedním z příkladu může být *projected grid*, který je specifický pro renderování otevřených moří nebo oceánů, princip tohoto algoritmu je popsán v [1]. Další metoda, která zvyšuje realistický dojem, se nazývá *Vertex Texture Displacement* [3].

# Závěr

Jednotlivé zmiňované metody se dnes v praxi hojně využívají, ať už v oblasti počítačových her nebo vědeckých simulací. Výpočetní výkon se stále navyšuje a vyvíjejí se nové metody v oblasti počítačové grafiky s cílem dosáhnout co nejrealističtějších podob renderovaných objektů.

# Odkazy

[1] Claes Johanson, (2004), Real-time Water Rendering, Introducing the projected grid concept

[2] Lasse Staff Jensen & Robert Golias, (2001), Deep-Water Animation and Rendering

[3] NVIDIA, (2009), GPU Gems 2: Chapter 18. Using Vertex Texture Displacement for Realistic Water Rendering