

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

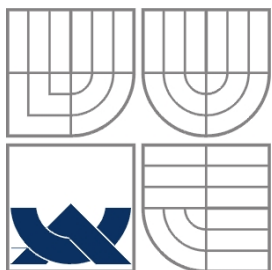
POKROČILÁ SIMULACE A VIZUALIZACE KAPALINY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

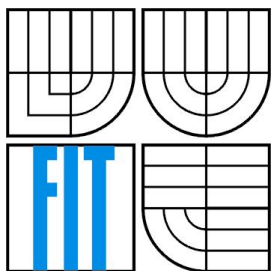
AUTOR PRÁCE
AUTHOR

Obr Jakub, Bc.

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ADVANCED SIMULATION AND VISUALIZATION OF A FLUID

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Obr Jakub, Bc.

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Jan Navrátil

BRNO 2011

Abstrakt

Tato práce se zaměřuje na fyzikálně založenou simulaci kapaliny a její následnou fotorealistickou vizualizaci. Zabývá se realizací metody Smooth Particle Hydrodynamics pro viskoelastické kapaliny a jejím rozšířením pro interakci více kapalin. Dále se věnuje problému hranic u SPH a implementací techniky neviditelných hraničních částic. Pro vizualizaci je popisována metoda sledování paprsku s rozšířením o absorpci a obarvení světla u průhledných materiálů. V souvislosti s ní je také probírán problém nekonečných totálních odrazů a jsou nabídnuty způsoby jeho řešení. Pro získání povrchu kapaliny je využívána metoda Marching Cubes, která je rozebírána z pohledu metody sledování paprsku.

Abstract

This thesis concentrates on physically based simulation of fluids followed by its photorealistic visualization. It describes one form of Smooth Particle Hydrodynamics methods for viscoelastic fluid simulation and includes its extension for multiple interacting fluids. It also deals with SPH boundary problem and investigates its solution by fixed boundary particles. For visualization of fluids there is a method of Ray Tracing described in detail and it's extended with light absorption in transparent materials. In connection with this method there is also discussed a problem of infinite total reflections and some solution techniques are offered. To extract the surface of the fluid there is used a Marching cubes method and its discussed in terms of Ray Tracing.

Klíčová slova

simulace kapaliny, vizualizace kapaliny, Smooth Particles Hydrodynamics, SPH, neviditelné hraniční částice, interakce více kapalin, Ray Tracing, totální odrazy, vizualizace transparentního objektu, Marching Cubes, problém interpolace normál

Keywords

liquid simulation, liquid visualization, Smooth Particles Hydrodynamics, SPH, fixed boundary particles, multiple fluids interaction, Ray Tracing, total reflections, transparent object visualization, Marching Cubes, normal interpolation problem

Citace

Jakub Obr: Pokročilá simulace a vizualizace kapaliny, diplomová práce, Brno, FIT VUT v Brně, 2011

Pokročilá simulace a vizualizace kapaliny

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jana Navrátila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Obr
11.1.2011

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Janu Navrátilovi za odbornou pomoc a také za jeho trpělivost a ochotu při pomoci s řešením projektu.

© Jakub Obr, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Simulace kapaliny.....	3
2.1 Navier-Stokesovy rovnice.....	3
2.2 Metody simulace kapaliny.....	4
2.3 Smooth Particle Hydrodynamics.....	5
3 Vizualizace kapaliny.....	16
3.1 Optické jevy.....	16
3.2 Metody vizualizace.....	18
3.3 Metoda Marching Cubes.....	22
4 Implementace simulace a vizualizace kapaliny.....	24
4.1 Simulační část.....	24
4.2 Vizualizační část.....	27
4.3 Zpracování na více počítačích.....	32
5 Závěr.....	33
Literatura.....	34
Ostatní zdroje.....	35
Seznam příloh.....	36

1 Úvod

Simulace určitého děje či situace a následná vizualizace nasimulovaných dat jsou nedílnou součástí mnoha vědních oborů a ve velké míře se s nimi setkáváme i v oblasti zábavního průmyslu. Konkrétně simulace a vizualizace kapaliny se využívá například pro předpověď šíření vodních toků při povodních, ale také při tvorbě digitálních efektů pro film nebo videohru.

Podle oblasti využití jsou na simulaci a vizualizaci kapaliny kladeny různé požadavky a je třeba pro ně zvolit metodu realizace, která tyto požadavky splňuje. Nástroj pro simulaci protržení přehrady bude vyžadovat co nejvyšší fyzikální přesnost simulace, ale pro vizualizaci si vystačí se symbolickým zastoupením kapaliny a prostředí. Ve filmovém průmyslu naopak stačí, aby kapalina pouze vytvářela dojem realistického chování a není důležité zachování vysoké fyzikální přesnosti. Při vizualizaci je však vyžadována co nejvyšší fotorealističnost, které se také dosahuje přesnější fyzikální simulací optických jevů reálného světa. Pro tvorbu efektů do filmu je také nutné mít možnost aplikovat na kapalinu externí virtuální síly, tedy kontrolovat její chování. V oblasti videoher je situace o něco náročnější, protože na rozdíl od předchozích dvou se simulace i vizualizace provádí v reálném čase a optické jevy se proto různými postupy na akcelerovaném grafickém systému jen napodobují.

Tato práce je zaměřena na realistickou simulaci a následnou fotorealistickou vizualizaci kapaliny neprováděnou v reálném čase. Zabývá se fyzikálně založenou metodou Smooth Particle Hydrodynamics (SPH) ve formě pro simulaci visko-elastických látek a jejím rozšířením pro interakci více kapalin. Také porovnává metody pro zpracování hranic SPH a navrhuje postupy pro práci s metodou neviditelných hraničních částic. Pro vizualizaci kapaliny podrobně popisuje metodu sledování paprsku a věnuje se odstranění problému nekonečných totálních odrazů, které u ní mohou vznikat. Dále se zabývá jejím rozšířením o absorpci a obarvení světla průchodem přes průhledný materiál a zavedením plošných zdrojů a neostrého odrazu či lomu z distributivní metody sledování paprsku. Věnuje se také metodě pro extrakci povrchu kapaliny Marching Cubes a s ní souvisejícímu problému interpolace normál.

Hlavní text práce je rozdělen do tří kapitol, z nichž dvě jsou teoretické a jedna praktická. První teoretickou kapitolou je kapitola 2 a zabývá se simulační částí práce. Seznámí čtenáře s používanými metodami pro simulaci kapaliny a podrobněji se věnuje jedné z nich, metodě Smooth Particle Hydrodynamics, která se díky schopnosti pracovat v reálném čase stala v poslední době velice oblíbenou. Značná část kapitoly se pak soustředí na problém hranic, který u této metody vzniká.

Kapitola 3, Vizualizace kapaliny, je tou druhou teoretickou částí a kromě obecného popisu metod vizualizace opět jednu z nich rozebírá podrobněji. Vybrána byla, také velice populární, metoda sledování paprsku, jelikož ji lze použít jako základ jiných vizualizačních metody. Část této kapitoly se zároveň věnuje optickým jevům souvisejících s kapalinou a metodě k vytvoření povrchové reprezentace kapaliny v podobě polygonálního modelu.

Poslední kapitola nahlíží na simulaci a vizualizaci z praktického hlediska. Zabývá se problémy, se kterými je možné se při implementaci simulátoru a rendereru kapaliny setkat, a na většinu z nich nabízí vhodné řešení.

V závěru práce jsou shrnuty získané poznatky a probrány výsledné schopnosti vytvořené aplikace. Také je diskutováno další možné pokračování této práce. Popis ovládání aplikace lze pak nalézt v příloze 1.

2 Simulace kapaliny

Kapaliny a plyny obecně nemají schopnost udržet si svůj tvar kvůli vzájemné pohyblivosti jednotlivých částic [16]. Tato vlastnost, tekutost, pak společně označuje kapaliny a plyny jako tekutiny.

Důležitou vlastností tekutin je také viskozita, nebo-li vazkost, a projevuje se jako její odpor proti změně tvaru, pokud tekutina není v klidu. Toto způsobuje viskózní síla, která snižuje vzájemné rozdíly rychlostí v proudící tekutině.

Pro popis pohybu tekutin našli, nezávisle na sobě v polovině 19. století, Claude-Louis Navier a George Gabriel Stokes takzvané Navier-Stokesovy rovnice.

2.1 Navier-Stokesovy rovnice

Vektorové diferenciální Navier-Stokesovy rovnice jsou určeny pro popis proudění nestlačitelné Newtonské tekutiny¹. Vznikly aplikací druhého Newtonova zákona na pohyb tekutiny zároveň s využitím předpokladu, že napětí kapaliny je součtem viskózní a tlakové složky [34].

Tyto rovnice nepracují s pozicemi, ale s rychlostmi a jejich výsledkem je tzv. pole rychlostí, nebo také proudové pole, které popisuje rychlost tekutiny na daném místě a v daném čase. Jakmile je pole vypočítáno, mohou být určeny i ostatní požadované hodnoty, jako rychlost toku nebo třecí síly.

Navier-Stokesovy rovnice sestávají ze dvou rovnic a jejich řešením v čase je možné simulovat chování kapaliny. První (2.1) slouží k zajištění zachování hmoty a zaručí nám tak nestlačitelnost kapaliny.

$$\nabla \cdot u = 0 \quad (2.1)$$

u..... pole rychlostí kapaliny
 ∇ gradient (operátor nabla) [21]

Druhá rovnice (2.2) dává do vztahu pole rychlostí a tlaku pomocí zákona o zachování hybnosti a modeluje tak změny pole rychlostí v čase jako následek viskozity, hustoty, tlaku, gravitace a proudění [9].

$$u_t = \nu \nabla \cdot (\nabla u) - (u \cdot \nabla) u - \frac{1}{\rho} \nabla p + g \quad (2.2)$$

u..... pole rychlostí
 ∇ gradient (operátor nabla)
 ν viskozita
 ρ hustota
p..... tlak
g..... gravitace

Analytický výpočet Navier-Stokesových rovnic je možný jen v určitých speciálních případech a dodnes nebylo matematicky dokázáno, že v trojrozměrném prostoru vždy existuje jejich řešení, které neobsahuje singularitu². Protože tyto rovnice popisují fyzikální chování mnoha dějů vědeckého i ekonomického charakteru a mohou tak posloužit například při návrhu letadel, předpovědi počasí nebo při analýze znečištění, nadace Clayova matematického institutu je označila jako jeden ze sedmi nejdůležitějších nevyřešených matematických problémů a za jejich vyřešení vypsala odměnu jeden milion dolarů [34].

1 Newtonská tekutina je viskózní látka, která se řídí Newtonovým zákonem viskozity [14]

2 Singularita je obecný název bodu, ve kterém daný matematický objekt není definován, nebo kde se objekt nechová v jistém smyslu rozumně (např. není diferencovatelný)

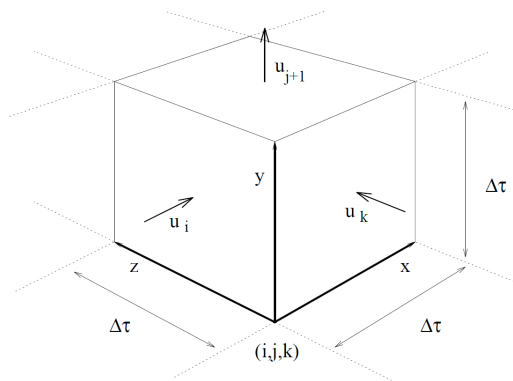
2.2 Metody simulace kapaliny

Pro numerické řešení Navier-Stokesových rovnic je nutné simulovaný objekt diskretizovat. Podle způsobu provedení se používají dva hlavní přístupy, Eulerův a Lagrangeův, a jeden hybridní, Semi-Lagrangeův.

2.2.1 Eulerův přístup

U Eulerova přístupu jsou vzorkovací body umístěny do pravidelné statické mřížky a v těchto bodech jsou uloženy vlastnosti látky na dané pozici. Simulační výpočet pak mění hodnoty vlastností v jednotlivých buňkách mřížky a kapalina se tím v prostředí pohybuje. Na obrázku 2.1 je ukázka buňky z tzv. MAC (Marker-and-Cell) mřížky. Ta uchovává skalární vlastnosti látky, jako je tlak či teplota, ve středu buňky a vektory rychlosti má oproti hlavní mřížce posunuty do stěn buněk, kde jsou přepočítány na velikost vektoru kolmému na danou stěnu v jejím středu. Tento způsob reprezentace je výhodný proto, že máme přímo k dispozici rychlost toku, kterým tekutina buňkou proudí. Nevýhodou naopak je, že není k dispozici explicitní informace o jejím povrchu.

Eulerův přístup pro kapaliny si získal na oblibě po uvedení prací dvojice Foster, N. a Metaxas, D. [10, 11], kteří Navier-Stokesovy rovnice řeší na pravidelné voxelové mřížce a na všechny diferenciální rovnice používají metodu konečných rozdílů [24].



Obrázek 2.1: Buňka mřížky se znázorněnými třemi vektory rychlosti, definovaných kolmo na stěny krychle [9].

2.2.2 Lagrangeův přístup

Lagrangeův přístup definuje simulovaný objekt jako skupinu bodů, nebo také částic, které s časem mění svou polohu. Vlastnosti objektu jsou uloženy v těchto bodech a pohybují se tedy s nimi. Zároveň mezi body mohou dynamicky vznikat a zanikat vazby a je možné tak simulovat tuhé, vazké i ideálně tekuté látky a lze vytvářet trvalé deformace způsobené aplikací dostatečných sil na objekt.

Jednou z metod Lagrangeova částicového přístupu pro simulaci kapaliny je metoda Smooth Particle Hydrodynamics, a protože tuto metodu používám v aplikaci patřící k této práci, je jí věnována samostatná podkapitola 2.3.

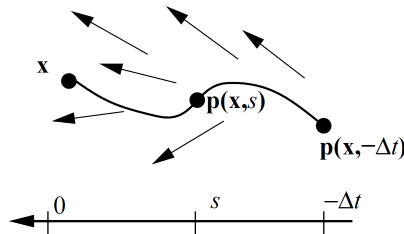
2.2.3 Semi-Lagrangeův přístup

Kombinaci obou předchozích přístupů poprvé do své práce vymyslel Jos Stam [30] a nechal tak vzniknout simulaci Semi-Lagrangeovým přístupem. Jeho simulátor dýmu pracoval v reálném čase, jelikož při tomto přístupu zůstává simulace stabilní i při použití většího simulačního kroku.

Jeho základem je pole rychlostí a pole tlaků uložené v pravidelné mřížce stejně jako u Eulerova přístupu. Při řešení Navier-Stokesových rovnic se však od toho přístupu liší v aplikaci advekčního kroku na sebe sama, při němž využívá sledování pohybu částic po tomto poli. K získání hodnoty rychlosti v určitém bodě v novém čase $t+\Delta t$ se stopuje pozpátku pohyb tohoto bodu polem rychlostí

po dobu Δt a novou hodnotou rychlosti bude hodnota částice z vystopované předchozí pozice. Náčrt této zpětné advekce je na obrázku 2.2

Výsledkem Semi-Lagrangeova přístupu je bezpodmínečná stabilita a tím vhodnost použití v reálném čase. Simulace je ale na druhou stranu nedostatečně přesná pro využití například ve stavebnictví a proudění v tekutině má tendenci k příliš rychlému útlumu oproti skutečnosti [30].



Obrázek 2.2: K vyřešení advekčního kroku u Semi-Lagrangeova přístupu se stopuje každý bod pole zpět v čase. Nová rychlost v bodě x je tedy rychlost, jež tato částice měla před časem Δt , tedy na pozici $p(x, -\Delta t)$ [30].

2.3 Smooth Particle Hydrodynamics

Smooth Particle Hydrodynamics (SPH) je metoda původně vyvinuta k řešení astrofyzikálních problémů v otevřeném trojrozměrném prostoru. Aplikuje se v mnoha oblastech inženýrství a kromě modelování vzniku galaxií a hvězd nebo u simulace nestlačitelné tekutiny najde své uplatnění například i v balistice při simulaci průniku objekty za vysokých rychlostech.

SPH využívá částic Lagrangeova přístupu k reprezentaci spojitého média a k transformaci spojitých parciálních rovnic určujících pohyb částic na diskrétní parciální rovnice [29]. Prostor je nerovnoměrně těmito částicemi navzorkován a uchovávají se v nich jeho vlastnosti, jako je hustota hmoty nebo rychlost. Částice jsou v průběhu simulace sledovány a tyto vlastnosti lze pak vyhodnotit v jakémkoliv místě prostoru pomocí radiálně symetrických vyhlazovacích jader [5].

Předpokladem metody SPH je možnost aproximovat hodnotu v prostoru váženým součtem hodnot z pomocné domény, která je tvořena sledovanými částicemi. Váhy jsou určeny jádrovou funkcí, což je v podstatě interpolační funkce udávající příspěvek hodnoty do místa v prostoru. Hodnotu a gradient v libovolném místě prostoru lze pak určit rovnicemi (2.3) a (2.4), které jsou již zapsány v diskretizované podobě [29].

$$A(r) \approx \sum_{j=1}^N A(r_j) \frac{m_j}{\rho_j} W(r-r_j, h) \quad (2.3)$$

$$\nabla A(r) \approx \sum_{j=1}^N A(r_j) \frac{m_j}{\rho_j} \nabla W(r-r_j, h) \quad (2.4)$$

- r pozice hledané hodnoty v prostoru
- r_j pozice částice z pomocné domény
- N počet částic
- $A(x)$ hodnota prostoru v bodě x
- m_j hmotnost částice j
- ρ_j hustota částice j
- $W(x,y)$ jádrová funkce ve vzdálenosti x a s maximální vyhlazovací vzdáleností y
- h vyhlazovací vzdálenost
- ∇ gradient (operátor nabra)

2.3.1 SPH při simulaci visko-elastické kapaliny

Simulaci kapaliny pomocí částic používají ve své práci Clavet a spol. [5]. Využili základní model SPH a jeho rozšířením vytvořili speciální přístup nazvaný double density relaxation, neboli ustálení dvojí hustoty. Při něm působení mezi částicemi závisí na dvou odlišných výpočtech hustoty v okolí částice a kromě základní hustoty se užívá tzv. blízké hustoty, která bude popsána dále v této kapitole. Ještě je na začátku důležité poznamenat, že hustota zde neznačí fyzikální veličinu, ale pouze vyčíslení vztahu částice ke svému okolí.

Ustálení dvojí hustoty

Nestlačitelnost se u SPH dosahuje udržováním konstantní klidové hustoty. Pro částici i Clavet a spol. počítají hustotu pomocí (2.5), kde jádro $(1 - r_{ij}/h)^2$ uznali experimentováním jako nejvhodnější. Z této hustoty pak získávají pseudotlak jako odchylku od klidové hustoty (2.6).

$$\rho_i = \sum_{j \in N(i)} (1 - r_{ij}/h)^2 \quad (2.5)$$

ρ_ihustota částice i
 h interakční poloměr (nebo také vyhlazovací vzdálenost)
 $N(i)$množina okolních částic částice i v dosahu interakční vzdálenosti h
 r_{ij} vzdálenost mezi částicí i a j

$$P_i = k(\rho_i - \rho_0) \quad (2.6)$$

P_ipseudotlak v místě částice i
 k parametr tuhosti
 ρ_0klidová hustota

Blízká hustota, a z ní vypočítávaný blízký tlak, slouží k zamezení shlukování částic. Pokud by nebyla použita, částice by mohly dosáhnout klidové hustoty těsným přitažením malého množství okolních částic a kapalina by se tak rozdělila do nespojitých na sobě nezávislých shluků.

Jádro v rovnici (2.7) pro výpočet blízké hustoty bylo opět získáno experimentálně a je obdobné jádru pro obyčejnou hustotu. Na rozdíl od ní je však kubické a má tedy strmější průběh. Aby blízký tlak zabránil shlukování částic, musí vytvářet odpudivé síly na blízké částice a je proto počítán z blízké hustoty přímo pomocí (2.8).

$$\rho_i^{near} = \sum_{j \in N(i)} (1 - r_{ij}/h)^3 \quad (2.7)$$

ρ_i^{near}blízká hustota částice i

$$P_i^{near} = k^{near} \rho_i^{near} \quad (2.8)$$

P_i^{near}blízký pseudotlak v místě částice i
 k^{near}parametr tuhosti pro blízký tlak

Výpočet posunu částic k zachování nestlačitelnosti je tvořen hlavním cyklem přes všechny částice i , do kterého jsou vnořeny dva cykly procházející všechny její okolní částice j . První vnořený cyklus určí pomocí (2.5) a (2.7) hustotu a blízkou hustotu částice i a v druhém cyklu se vypočítá posun, který se aplikuje vždy na obě částice k uplatnění zákona akce a reakce, přičemž na částici j se aplikuje přímo a na částici i ve znegovaném tvaru.

Pro výpočet posunu slouží rovnice (2.9). Činitel $(1 - r_{ij}/h)$ upravuje jeho velikost závislou na tlaku P_i podle vzdálenosti mezi částicemi a podobně činitel $(1 - r_{ij}/h)^2$ upravuje míru odpudivé síly závislé na blízkém tlaku P_i^{near} . Jednotkový vektor \vec{r}_{ij} pak udává posunu směr. Aby se zachovala nezávislost na velikosti časového kroku, je v rovnici zahrnut kvadratický činitel Δt^2 , protože jsou síly

$$\vec{D}_{ij} = \Delta t^2 [P_i(1 - r_{ij}/h) + P_i^{near}(1 - r_{ij}/h)^2] \vec{r}_{ij} \quad (2.9)$$

\vec{D}_{ij} vektor posunu mezi částicemi i a j
 Δt velikost časového kroku
 \vec{r}_{ij} jednotkový vektor směřující od částice i k částici j

k získání posunu aplikovány dvakrát. To je způsobeno použitým integračním schématem Prediction-Relaxation Scheme (Schéma predikce a ustálení), při kterém se jednotlivé působící síly aplikují sekvenčně. Tím je možné lépe reflektovat právě probíhající změny a dosahuje se vyšší stability simulace. Bližší popis Schématu predikce a ustálení lze nalézt v příloze v článku Claveta a spol. [5].

Viskozita

K dosažení viskozity se na částice aplikují impulzy, které tlumí jejich rychlost, pokud se k sobě přibližují. Velikost tohoto impulzu se počítá pomocí (2.11) a závisí na vzdálenosti částic, určené pomocí lineárního jádra $(1 - r_{ij}/h)$, a na činiteli $(\sigma + \beta u^2)$, který zajišťuje lineární a kvadratickou závislost viskozity na vzájemné radiální rychlosti částic. Lineární člen σ slouží k nastavení viskozity u vysoce viskózních látek a kvadratický člen β zabraňuje vzájemnému pronikání částic. Vzájemná radiální rychlost u je velikost rychlosti, se kterou se částice k sobě přibližují, a získá se pomocí (2.10) jako průmět relativní rychlosti částic do směru mezi nimi.

$$u = (v_i - v_j) \cdot \vec{r}_{ij} \quad (2.10)$$

u vzájemná radiální rychlost částic i a j
 v_i vektor rychlosti částice i
 v_j vektor rychlosti částice j

$$I = \Delta t (1 - r_{ij}/h) (\sigma u + \beta u^2) \vec{r}_{ij} \quad (2.11)$$

I impulz viskózní síly aplikovaný na částice i a j
 σ lineární člen viskozity
 β kvadratický člen viskozity

Elasticita a plasticita

Zavedením pružin mezi částicemi Clavet a spol. docilují u simulované látky elasticity a plasticity. Jejich působení je rozděleno do dvou fází, kdy při první dochází k úpravě klidové délky pružin a při druhé k započtení jejich vlivu.

Úprava klidové délky pružin závisí na konstantě plasticity α , určující rychlost formování délky pružin, a na činiteli poddajnosti γ , který slouží pomocí (2.12) k získání mezní deformace d . Hodnota této mezní deformace určuje vzdálenost částic od klidové vzdálenosti pružin, při níž ještě nedochází k jejich trvalým změnám. Právě následkem jejího překročení jsou pružiny lineárně s časem prodlužovány podle rovnice (2.13) nebo zkracovány podle (2.14).

Vzájemný posun částic vlivem pružin se ve druhé fázi vypočítá pomocí rovnice (2.15) a rozloží se rovnoměrně mezi obě částice. Kromě upravené klidové délky závisí posun ještě na konstantě

$$d = \gamma L_{ij} \quad (2.12)$$

d mezní deformace pružiny mezi částicemi i a j
 γ činitel poddajnosti pružin
 L_{ij} klidová vzdálenost pružiny mezi částicemi i a j

$$L_{ij} = L_{ij} + \Delta t \alpha (r_{ij} - L_{ij} - d) \quad (2.13)$$

$$L_{ij} = L_{ij} - \Delta t \alpha (L_{ij} - d - r_{ij}) \quad (2.14)$$

α konstanta plasticity

$$D = \Delta t^2 k^{\text{spring}} (1 - L_{ij}/h) (L_{ij} - r_{ij}) \vec{r}_{ij} \quad (2.15)$$

D..... vzájemný posun mezi částicemi i a j
 k^{spring} konstanta tuhosti pružin

tuhosti pružin k^{spring} , která určuje rychlost jejich návratu do klidového stavu. Koeficient $(1 - L_{ij}/h)$ pak slouží k pozvolnému útlumu sil vyvolaných dlouhými pružinami.

Aby kapalina s pružinami neztratila svou tekutost, dochází v první fázi k vytváření pružin dynamicky. Pokud se k sobě volné částice přiblíží více jak na interakční vzdálenost, je mezi nimi pružina vytvořena s klidovou hodnotou nastavenou na velikost této interakční vzdálenosti. Pokud se naopak vzdálí více jak na interakční vzdálenost, pružina zaniká.

Při simulaci kapalin s velmi nízkou viskozitou není nutné pružiny používat, ale u vysoce viskózních látek, jako například u medu, je jejich aplikace nezbytná. Samostatné viskózní síly totiž nedokážou tok částic dostatečně zpomalit a látka nezanechává proti gravitaci stopy původního tvaru.

Pro kompletní algoritmy a podrobný popis visko-elasticity v SPH odkazují na 4. a 5. kapitolu původní práce Claveta a spol. [5].

2.3.2 Kolize s objekty

Při pohybu částic v určitém prostředí je nezbytné řešit jejich kolize s okolními objekty. Clavet a spol. [5] jako reakci na kolize aplikují na částice impulzy podle modelu mokrého tření s nulovou restitucí. Při něm se rychlost částice rozděluje na kolmou a tečnou komponentu k místu kolize a výsledný impulz ruší její kolmou část a tlumí tečnou. Výpočet kolmé a tečné komponenty rychlosti ukazují rovnice (2.16) a (2.17), samotného impulzu pak rovnice (2.18). Míra útlumu je dána parametrem tření μ , které může nabývat hodnot od 0, kdy není aplikováno žádné tření, do 1, kdy naopak nedochází ke klouzání částic po povrchu objektu. Jelikož model mokrého tření s nulovou restitucí vyžaduje aktuální rychlost částice vypočítanou z předchozí a současné pozice, řeší se reakce na kolize až na konci celého simulačního cyklu.

$$\vec{v}_{normal} = -(\vec{v} \cdot \vec{n}) \vec{n} \quad (2.16)$$

\vec{v}_{normal} normálová komponenta rychlosti částice v místě kolize
 \vec{v} vektor aktuální rychlosti částice
 \vec{n} jednotkový normálový vektor povrchu objektu v místě kolize

$$\vec{v}_{tangent} = \vec{v} - \vec{v}_{normal} \quad (2.17)$$

$\vec{v}_{tangent}$ tečná komponenta rychlosti částice v místě kolize

$$I = -(\vec{v}_{normal} + \mu \vec{v}_{tangent}) \quad (2.18)$$

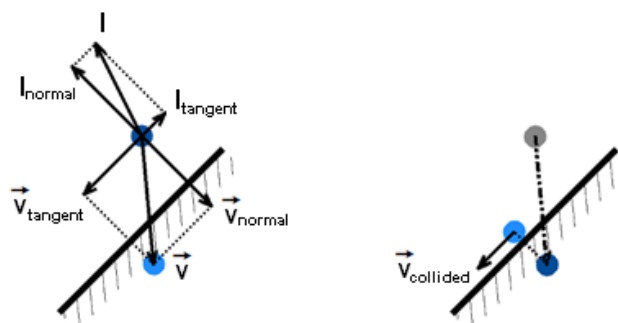
I..... impulz aplikovaný na kolidující částici
 μ parametr tření

$$\vec{v}_{collided} = \vec{v} - (1 - \mu)(\vec{v} \cdot \vec{n}) \vec{n} + \mu \vec{v} \quad (2.19)$$

Kolize s tuhými tělesy

Kolize s tuhými tělesy, nebo objekty, můžeme rozdělit na dva případy. Kolize s nehybným tuhým tělesem, při které se pozice ani rychlost objektu jejím vlivem nemění, a kolize s volným tuhým tělesem, u které dochází k úpravě polohy, rychlosti a úhlové rychlosti objektu podle sil působících částic.

Pokud je detekován průnik částice do nehybného tuhého tělesa, musí se z něj nejprve odstranit. Při použití přiměřené velikosti časového kroku není nutné zjišťovat přesný bod vstupu částice a k jejímu odstranění z objektu jí stačí posunout po normále povrchu o hloubku penetrace, aniž by tím docházelo k zásadní změně v chování částic.



Obrázek 2.3: Aplikace impulsu kolize na částici. V levé části obrázku jsou stavy částice před a po proniknutí do objektu, v pravé části je po vyjmutí částice z povrchy a aplikace impulsu. Světle modrý bod značí aktuální stav částice, tmavě modrý stav předcházející.

Následně se upraví jejich rychlost podle modelu mokrého tření s nulovou restitucí. Přepočtem rovnic (2.16) až (2.18) lze rychlost částice po kolizi získat přímo pomocí (2.19). Princip zpracování částice před kolizí a po ní je znázorněn na obrázku 2.3.

U kolize se volnými tuhými tělesy je situace mnohem komplikovanější. Clavet a spol. k tomuto účelu vymysleli třífázovou metodu interakce částic a těles. V první fázi akumuluje sílu a točivý moment způsobený všemi kolidujícími částicemi. V druhé podle naakumulovaných sil upraví postavení a rychlost objektu. Nakonec přizpůsobí pozice částic, aby se nenacházely uvnitř objektu.

Zavedení volných objektů vyžaduje vytvoření systému pro řešení mechaniky tuhých těles, což přesahuje rámec této práce. Proto pro více informací odkazují na 6. kapitulu práce Claveta a spol. [5] a také na práci Guendelmana a spol. [12], která je v ní k systému tuhých těles doporučována.

Hraniční podmínky

Aby bylo možné udržet částice kapaliny v požadovaném prostoru, je třeba reagovat na jeho hraniční podmínky. K řešení těchto podmínek můžeme přistupovat stejně jako ke kolizi s nehybnými objekty a při dodržení dvou zásad je můžeme nahradit kolizním systémem.

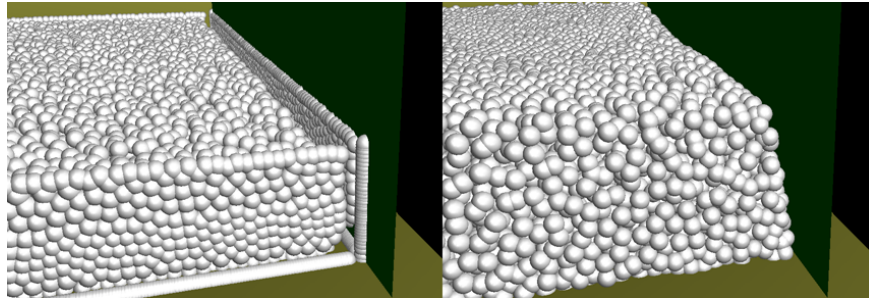
První zásada je velice intuitivní. Musíme tento prostor těsně ohradit objekty, aby přes ně částice nemohly uniknout. Při použití obecně konvexního tvaru hraničního prostoru lze k tomuto účelu vytvořit speciální objekt nekonečné zdi, u níž lze skalárním součinem jednoduše testovat, zda se částice nachází před nebo za ní.

Druhá zásada je podmíněna použitým způsobem práce s částicemi a je užitečná jen za předpokladu, že částice nesmí opustit určitou oblast. To může být dáno například úmyslně neošetřenými hranicemi pole, do kterého částice ukládáme. V takovém případě je nutné vzít v potaz maximální velikost časového kroku simulace a maximální předpokládanou rychlost pohybu částice a vypočítat z nich vzdálenost, o kterou mohou částice proniknout do objektu před započtením vlivu kolize. O tuto vzdálenost pak musí být hraniční objekty posunuty od skutečných požadovaných hranic.

Určení maximální předpokládané rychlosti je ovšem problém, protože by v ideálním případě měla být nekonečná. Z toho důvodu je vhodnější nastavit posun hranic experimentálně. Při simulaci ale mohou nastat určité anomálie, nebo spíše neočekávané situace, které vedou k vystřelení částic vysokou rychlostí, a tudíž nechávat mezní podmínky úplně neošetřené nedoporučuji.

2.3.3 Zpracování hranic pro SPH

Při simulaci s pomocí SPH vzniká problém s určováním hodnot spojitého média blízko hranic překážek. Zde jádrová funkce zasahuje i do oblasti překážky, ve které se však nemohou částice nacházet. Vážený součet hustoty částic na tomto místě proto není korektní a dochází k jejich shlukování u hranic objektů. Příklad takového hromadění částic můžeme vidět na obrázku 2.4.



Obrázek 2.4: Shlukování částic – V levé části obrázku není řešeno zpracování hranic a částice se shlukují u stěn a v rozích. V pravé části jsou při výpočtu použity neviditelné hraniční částice.

Další problém s hranicemi vzniká u vodorovné podlahy, tedy přesněji u plochy kolmé na aplikovanou gravitační sílu. Pokud se všechny částice dostanou do jedné, absolutně shodné hladiny na povrchu této plochy a následně je nahustíme zmenšením volného prostoru kolem nich, již nedojde k nárůstu objemu kapaliny do výšky. Toto je zaviněno použitou metodou zachování hmoty přes udržování konstantní hustoty částic. Při ní dochází k úpravě jejich poloh pouze po drahách mezi sousedními částicemi. Nemohou se proto již dostat na jinou úroveň, než na které se současně všechny nachází. Vzhledem ke způsobu řešení kolize vytlačení na povrch objektu se částice, při dostatečném volném prostoru, dostanou do jedné hladiny v okamžiku, kdy dojde k jejich rozprostření po podložce.

Na oba tyto problémy existuje hned několik možných řešení a následující podkapitoly popisují ty nejznámější.

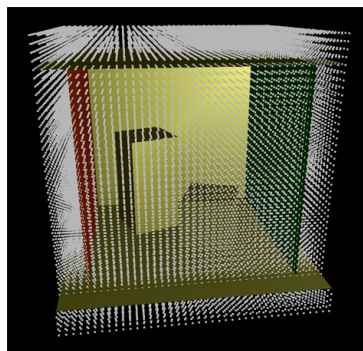
Neviditelné hraniční částice

Tato metoda je zmiňována v mnoha pracích zabývajících se SPH. Důvodem je její jednoduchý princip a snadné začlenění do jakékoliv podoby algoritmu SPH.

Její podstata spočívá v obsazení prostoru za hranicemi a uvnitř překážek statickými částicemi, které se nezobrazují a přebírají vlastnosti simulované kapaliny. Při určování hustoty se pak u hranic započítává jejich příspěvek a odstraní se tím prázdná oblast radiálního jádra. Algoritmus k udržení konstantní hustoty z kapitoly 4.2 práce Claveta a spol. rozšířený o výpočet vlivu hraničních částic pro statické objekty je sepsaný v sekci Algoritmus 2.1. Tmavší barva textu značí upravené úseky a světlejší ty původní. U algoritmu je pak důležité si povšimnout, že při aplikaci posunů od hraničních částic statických objektů se jejich velikost již nedělí a k pozici částice kapaliny se připočítávají celé kvůli zachování zákona akce a reakce. Ukázka neviditelných částic použitých ve scéně v aplikaci je na obrázku 2.5.

Jednoduchost v nasazení této metody vyvažuje několik nově vzniklých problémů, které se musí řešit. Prvním z nich je umístění a rozestup částic.

Hraniční částice by měly poskytovat objektu konstantní hustotu v celém jeho objemu. V zásadě však stačí jimi oblast tělesa vyplnit do hloubky určené interakční vzdáleností částic. Pro lepší



Obrázek 2.5: Neviditelné hraniční částice vygenerované kolem stěn scény

Algoritmus 2.1:

<pre> pro každou částici i ρ = 0 ρ^{near} = 0 // vypočti hustotu a blízkou hustotu pro každou částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 ρ = ρ + (1 - q)² ρ^{near} = ρ^{near} + (1 - q)³ // připočti hustotu hraničních částic pro každou hraniční částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 ρ = ρ + (1 - q)² ρ^{near} = ρ^{near} + (1 - q)³ // vypočti tlak a blízký tlak P = k(ρ - ρ₀) P^{near} = k^{near} ρ^{near} dx = 0 pro každou částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 // aplikuj posuny D = Δt² (P (1 - q) + P^{near} (1 - q)²) r̄_{ij} x_j = x_j + D/2 dx = dx - D/2 // připočti posuny hraničních částic pro každou hraniční částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 // aplikuj posuny D = Δt² (P (1 - q) + P^{near} (1 - q)²) r̄_{ij} dx = dx - D x_i = x_i + dx </pre>	<pre> // spočítej posuny samotných hraničních částic pro každou hraniční částici i ρ = 0 ρ^{near} = 0 // vypočti hustotu a blízkou hustotu pro každou částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 ρ = ρ + (1 - q)² ρ^{near} = ρ^{near} + (1 - q)³ // připočti hustotu hraničních částic pro každou hraniční částici j ∈ sousedi(i) pokud j ≠ i q = r_{ij}/h pokud q < 1 ρ = ρ + (1 - q)² ρ^{near} = ρ^{near} + (1 - q)³ // vypočti tlak a blízký tlak P = k(ρ - ρ₀) P^{near} = k^{near} ρ^{near} dx = 0 pro každou částici j ∈ sousedi(i) q = r_{ij}/h pokud q < 1 // aplikuj posuny D = Δt² (P (1 - q) + P^{near} (1 - q)²) r̄_{ij} x_j = x_j + D </pre>
--	---

Legenda:

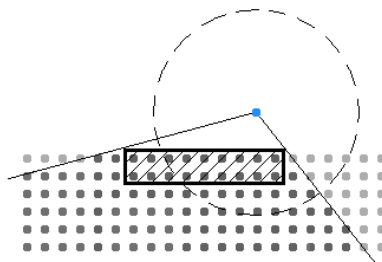
sousedi(*i*) částice sousedící s částicí *i* do vzdálenosti interakčního poloměru *h*

výsledky je vhodné tuto vzdálenost zvýšit o konstantu kvůli penetraci částic do objektů vlivem externích sil, jako je gravitace. Velikost této konstanty se určí podobně jako posun hraničních objektů popisovaný v podkapitole Hraniční podmínky kapitoly 2.3.2.

K zajištění konstantní hustoty se neviditelné částice umísťují do pravidelné struktury. Nejčastěji se využívá pravoúhlá mřížka, která poskytuje jejich dostatečné konstantní rozložení. Protože je ale prostor vzorkován spojitou radiální jádrovou funkcí, je tato konstantnost při nepravidelném pohybu po povrchu snížena a částice kapaliny jsou tak pravidelností mřížky ovlivňovány. Zlepšení lze dosáhnout umístováním bodů zároveň do vrcholů i středů buněk mřížky. Zvýší se tím ovšem hustota částic a musí se proto upravit jejich rozestup.

Za účelem zachování jednoduchosti této metody je rozestup buněk nejvhodnější získat experimentálně, čímž se odprostíme od způsobu výpočtu hustoty a použité vážené jádrové funkce. Experimentování můžeme provést naplněním kvádřové nádoby částicemi kapaliny a inicializací rozestupu na libovolnou hodnotu, při níž dochází ke shlukování částic u stěn a v rozích nádoby. Poté vždy snižujeme rozestup o konstantní hodnotu epsilon, dokud nezmizí mezera mezi vnitřními částicemi a částicemi podél hranic nádoby. Pokud jsme již dosáhli požadované přesnosti, můžeme nyní experimentování ukončit. V opačném případě vrátíme hodnotu rozestupu o krok epsilon zpět, zmenšíme velikost kroku na polovinu a pokračujeme v testování od fáze zvyšování hodnoty rozestupu. Po každé změně rozestupu hraničních částic je nutné vždy nechat simulaci spuštěnou dostatečně dlouho, aby měla kapalina čas na ustálení.

Hraniční částice není vhodné umísťovat na samotném povrchu, ale posunout je od něj o jejich vzájemný rozestup. Důvodem je použitý kolizní systém, který reaguje na částice až po jejich průniku



Obrázek 2.6: Neviditelné hraniční částice pro jednu stěnu tenké překážky. Modrou barvou je zakreslena částice kapaliny. Světle šedé jsou pro kapalinu viditelné hraniční částice a tmavě šedé jsou kryté překážkou.

do objektu a vyrovnává je na jeho hranici. Pokud bychom je vytvářeli již od hranic, částice kapaliny by nad povrchem pluly a nebyly by na ně aplikovány třecí síly. V takovém případě by bylo nutné kolizní systém upravit, aby na částice působil ještě před vniknutím do objektu.

Dalším problémem, který je s neviditelnými hraničními částicemi třeba řešit, jsou tenké překážky. U nich jádrová funkce zasahuje do překážky i do oblasti za ní a vyžaduje si proto speciální zpracování, které může být následující: Do výpočtu hustoty částice se započítají jen ty okolní částice kapaliny, které mají na počítanou částici přímou viditelnost a nenachází se tedy mezi nimi tenká překážka. Hraniční částice máme pro tento objekt vygenerovány pro každou jeho stěnu zvlášť do hloubky určené interakční vzdáleností, stejně jako u obyčejných překážek. Na rozdíl od nich se navíc o tuto vzdálenost generují do šířky a do výšky i mimo oblast za plochou stěny. Pro částici se poté vyberou hraniční částice podle stěny, ke které jsou nejbližší a započítají se z nich jen ty, na něž nemá přímou viditelnost, aby se vyrovnal pouze skrytý objem. Jelikož se v tuto chvíli mohou částice nacházet již v objektu, musí se na to při kontrole přímé viditelnosti brát ohled a dočasně při ní částici posunout mimo něj. Příklad takto vygenerovaných částic pro jednu stěnu tenké překážky znázorňuje obrázek 2.6.

Zavedením neviditelných hraničních částic dojde s jejich počtem ke kvadratickému zvýšení časové složitosti a s počtem částic kapaliny k lineárnímu zvýšení časové složitosti simulace. Zároveň může být problematické jejich stanovení u komplikovanějších objektů. I přes značné množství nevýhod však nalezneme tuto metodu díky své jednoduchosti uplatnění pro nenáročnou nebo testovací scény.

Zrcadlené neviditelné hraniční částice

Ghost particles, jak se tato metoda také označuje, využívá neviditelné hraniční částice podobně jako metoda předchozí. Na rozdíl od ní ale nemají vůči objektu pevnou pozici.

Pokud se v interakční vzdálenosti od hranice nachází částice kapaliny, vytvoří se souměrně od hranice její zrcadlová kopie. Tato neviditelná částice má stejné vlastnosti jako částice kapaliny, její rychlost je však hranicí zrcadlově obrácena. Navíc, když se částice kapaliny nachází blízko rohu dvou hranic, generuje se ještě jedna symetricky k tomuto rohu. V trojrozměrném prostoru tedy u vrcholu krychle vznikne sedm zrcadlených částic – tři od stěn, tři od rohů vždy mezi dvěma stěnami a jedna od rohu všech tří stěn.

Mimo zrcadlených částic tato metoda vyžaduje ještě neviditelné částice s nulovou rychlostí umístěné na hranicích objektů. Ty bývají vytvořeny s objektem předem staticky, ačkoliv existují modifikace tohoto algoritmu, kdy se na hranici vytváří dynamicky mezi zrcadlenou a původní částicí. Také je nutné zapojit hraniční částice do výpočtu viskozity, aby se uplatnil efekt jejich rychlostí, což u předchozí metody nebylo nutné.

Začlenění zrcadlených neviditelných částic by mělo vést k lineárnímu zvýšení časové složitosti v závislosti na počtu částic. Podle testů Donga a Jianga[7] je ale tato metoda méně stabilní než statické neviditelné hraniční částice, což je v jejich práci odůvodněno přenosem chyb z částic kapaliny do zrcadlených kopií.

Odpudivé hraniční síly

Na hranicích objektů je možné aplikovat na částice odpudivé hraniční síly a tím zamezit shlukování. Jejich použití vychází z předpokladu, že na hranicích objektů působí od hraničních částic odpudivé centrální síly, což jsou síly, jejichž velikost závisí pouze na vzdálenosti. Stanovení těchto sil zapsaných v rovnici (2.20) je vyvozeno z výpočtu sil mezi molekulami ve formě tzv. Lennard-Jonesova potenciálu, což původně představil ve své práci Monaghan, J.J.[22]. Za konstanty p_1 a p_2 jsou pro většinu simulací doporučeny hodnoty $p_1=4$, $p_2=2$ a výchozí vzdálenost mezi částicemi r_0 určuje mezní vzdálenost, do které je síla ještě aplikována. Koeficient D se volí podle fyzikálních vlastností simulované úlohy a například při simulaci problému protržení přehrady závisí na hloubce vody H a určuje se rovnicí $D=5gH$, kde g zastupuje gravitační sílu.

$$f(r) = \begin{cases} D \left[\left(\frac{r_0}{r} \right)^{p_1} - \left(\frac{r_0}{r} \right)^{p_2} \right] \frac{r}{r^2} & , \text{pokud } r \leq r_0 \\ 0 & , \text{pokud } r > r_0 \end{cases} \quad (2.20)$$

$f(r)$ odpudivá síla na jednotku hmoty částice
 r vzdálenost částice od hranice
 r_0 výchozí vzdálenost částic (nebo také mezní vzdálenost síly)
 p_1, p_2 konstanty simulace
 D koeficient závislý na simulované úloze

Pro více informací o výpočtu odpudivých sil s využitím Lennard-Jonesova potenciálu lze kromě Monaghanovi práce [22] nalézt i mnoho dalších, které se jimi zabývají při řešení hraničních podmínek [36], nebo z ní vycházejí a rozšiřují ji [37].

Zajímavé řešení odpudivých sil k vyrovnání hraničních podmínek nabízí ve své práci dvojice Song, B. a Dong, L. [29]. Jelikož přesné stanovení odpudivých sil podle Monaghanova přístupu způsobem, aby částice nepronikaly do objektů, může být pro složitější povrchy náročné, vytvořili metodu EFS (Effective Force Surface), nebo-li metodu Účinného silového povrchu.

Při této metodě se vychází z předpokladu, že rozložení hraničních částic na povrchu objektu je rovnoměrné a působí tedy na částici kapaliny společnou odpudivou silou, která je na povrch kolmá. K určení síly pak využívají znalosti hustoty hraničních částic na jednotku povrchu a výpočtu obsahu plochy účinného silového povrchu. Tato plocha ve své podstatě značí obsah povrchu zasažený vyhlazovacím jádrem. U rovinné plochy se například počítá podle rovnice (2.21) jako obsah podstavy kulové úseče vzniklé průnikem této plochy se sférickým vyhlazovacím jádrem.

$$S = \pi (h^2 - r^2) \quad (2.21)$$

S velikost plochy účinného silového povrchu
 h interakční vzdálenost
 r vzdálenost částice od povrchu

Jelikož hraničních síly nesmí již odpuzovat částice, které povrchem prostoupily na jeho opačnou stranu, využívá se k zabránění pronikání částic do objektů několika virtuálních hraničních vrstev. Čím je vrstva hlouběji, tím větší odpudivou sílu na částici vyvíjí. Vzdálenosti mezi jednotlivými vrstvami zároveň umožňují simulovat tvrdost povrchu, kdy se při kratších vzdálenostech částice okamžitě odrazí zpět, napodobující tvrdý povrch, a naopak při větších vzdálenostech dojde k postupnému návratu částice, v napodobení povrchu měkkého. Opět pro podrobnější informace odkazují na původní materiál [29].

Odpudivé hraniční síly obecně přináší elegantní řešení hraničních podmínek s menším nárůstem časové složitosti oproti předchozím metodám. Jejich implementace je však v porovnání s nimi složitější vlivem vyšší matematické komplexnosti.

2.3.4 Interakce více kapalin

Začleněním interakce dvou kapalin do výpočtu hustoty se zabývá ve své práci v kapitole 3.2.12 Marian, J. [20]. Vzájemného působení dosahuje simulací třech vlastností – hustoty, difuzivity a odpudivosti. Hustota je zavedena v podobě hmoty částic m . Difuzivita Φ určuje vzájemné pronikání částic kapalin mezi sebou a při hodnotách různých od 1 mají částice odlišných kapalin tendenci do sebe pronikat. Podle toho, zda je Φ menší nebo větší jak 1, snaží se proniknout buď první kapalina do druhé nebo naopak. Poslední vlastnost, odpudivost, je označena ξ a v závislosti na hodnotě větší nebo menší jak 0 se mezi sebou částice různých kapalin odpuzují nebo přitahují.

Takto nadefinované hodnoty vlastností ovšem určují vztah vždy mezi dvěma kapalinami a pro interakci více kapalin proto není vhodné je určovat přímo. K tomu je nutné zavést nové vlastnosti hmoty částic m_i , difuzivity Φ_i a velikosti odpudivé síly ξ'_i , které budou určovat hustotu, difuzivitu a odpudivost i -té kapaliny.

Původní hodnota hmoty, kterou lze opět použít v algoritmu popsáném Marianem, se získá podílem hmot interagujících kapalin podle rovnice (2.22). Přesněji, vzhledem ke zmíněnému algoritmu, vydělením hmoty částice první kapaliny hmotou částice kapaliny druhé. Zvýšení hmoty částice konkrétní kapaliny takto intuitivně vyznačuje její vyšší hmotnost a přeneseně i hustotu.

Vzájemné difuzivity se dosahuje obdobným způsobem podle rovnice (2.23). Zde je ovšem jejich poměr obrácen, aby kapalina s vyšší hodnotnou difuzivity pronikala do kapaliny s nižší hodnotou.

Odlišná situace nastává u odpudivosti, kde bylo vhodné pro lepší představu novou vlastnost odlišit již názvem. Velikost této vlastnosti určuje míru určité pseudo-síly ξ'_i , kterou kapalina působí na částice ostatních kapalin. Znaménko pak určuje její směr. Vzájemná odpudivost mezi konkrétními kapalinami se následně získá podle (2.24) součtem těchto ad hoc vlastností.

$$m = \frac{m_i}{m_j} \quad (2.22)$$

m vzájemná hmoty částic mezi kapalinou i a kapalinou j

m_i hmoty částice kapaliny i

m_j hmoty částice kapaliny j

$$\Phi = \frac{\Phi_j}{\Phi_i} \quad (2.23)$$

Φ vzájemná difuzivita mezi kapalinou i a kapalinou j

Φ_i difuzivita kapaliny i

Φ_j difuzivita kapaliny j

$$\xi = \xi'_i + \xi'_j \quad (2.24)$$

ξ vzájemná odpudivost kapalin

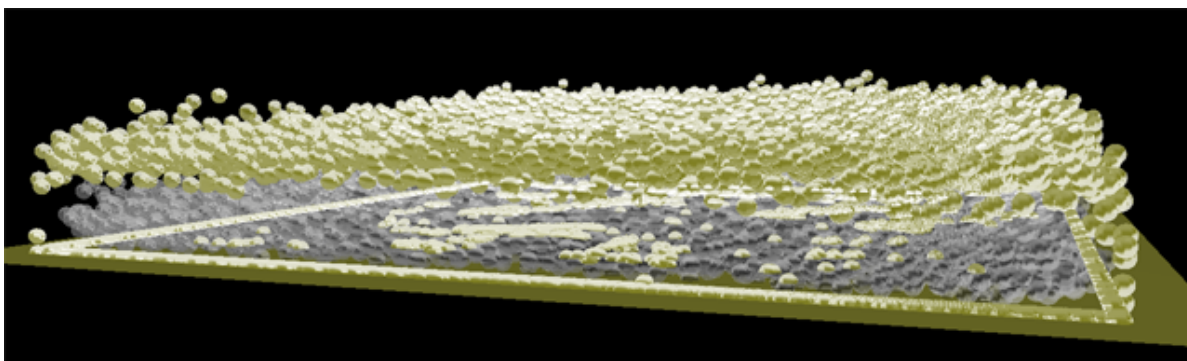
ξ'_i odpudivá pseudo-síla kapaliny i

ξ'_j odpudivá pseudo-síla kapaliny j

Nově vytvořené vlastnosti lze již svázat s jednotlivými kapalinami a algoritmus Mariana jednoduše rozšířit pro interakci více než dvou kapalin.

Neviditelné hraniční částice při interakci více kapalin

Jak bylo zmíněno v kapitole 2.3.3, hraniční částice přebírají vlastnosti počítané kapaliny, aby na rozhraních objektů korektně vyrovnávaly vyhlazovací jádrovou funkci. Při simulaci více kapalin je však takovéto přizpůsobování nepřijatelné, jelikož jsou na částice různých kapalin, v závislosti na jejich vlastnostech, vyvíjeny relativně odlišné síly. Lehká kapalina přelítá těžkou má pak tendenci držet se u hranic. To je zapříčiněno vlivem sil od hmoty těžké kapaliny, které lehkou kapalinu stlačují do shluků. Působením hraničních částic se následně tyto shluky drží u podložky a zároveň svým vyšším nahuštěním odpuzují částice těžké kapaliny. Lehká kapalina tedy není nikdy kompletně



Obrázek 2.7: Shlukování vlivem neviditelných hraničních částic při simulaci více kapalin

vytlačena nad těžkou a i při jejich promíchání dochází k uvíznutí shluků převážně v rozích objektů, kde jsou nejvíce ovlivněny silami od hraničních částic. Na obrázku 2.7 je ukázka shlukování částic v rozích a na podložce po slití lehké a těžké kapaliny. Lehká kapalina je znázorněna žlutými matnými a těžká šedými průhlednými kuličkami.

Tímto nedostatkem ovšem netrpí například metoda Ghost Particles, jelikož doména zrcadlených částic je společná pro všechny kapaliny. Na lehčí částice tedy působí těžká kapalina i z podložky a vytlačuje ji nad sebe. Proto je vhodné tyto dvě metody při interakci kapalin zkombinovat. Neviditelným hraničním částicím se zachová pravidelné uspořádání, jejich vlastnosti se však již nebudou nastavovat podle aktuálně počítané kapaliny, ale před začátkem výpočtu hustoty každá hraniční částice převezme vlastnosti kapaliny, jejíž částice je k ní nejbližší. Vznikne tím podobný efekt jako při zrcadlení částic, ale bez nutnosti je dynamicky vytvářet.

Při tomto způsobu řešení sice již nedochází ke shlukování různých kapalin, ovšem částice mohou u povrchu stále lnout o něco déle, než by měly. Tuto metodu proto lze ještě vylepšit zavedením skutečného zrcadlení vlastností namísto pouhého kopírování od nejbližší kapaliny. Nakonec tedy můžeme takovýmto způsobem rozlišit dvě nové hybridní metody – kopírované statické neviditelné hraniční částice a zrcadlené statické neviditelné hraniční částice.

3 Vizualizace kapaliny

K získání pozorovatelného obrazu kapaliny je nutné provést její zobrazení vhodnou metodou vizualizace. Těchto metod existuje několik a jelikož se všechny ve své podstatě snaží o určitý stupeň abstrakce reálného světa, je první kapitola věnována optickým jevům, se kterými se následně při jejich popisu budeme moci setkat.

3.1 Optické jevy

3.1.1 Odraz

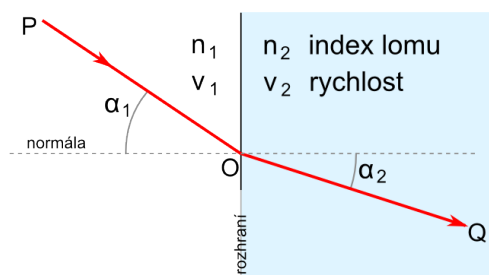
Odraz paprsku světla může vzniknout při kontaktu s rozhraním dvou prostředí, kdy se paprsek odráží zpět do prostředí, ze kterého na rozhraní narazil. Odraz se řídí zákonem odrazu, který říká, že odražený paprsek leží v rovině dopadu a úhel odrazu se rovná úhlu dopadu [13].

Podle materiálu, ze kterého se paprsek odráží, pak může docházet k zrcadlovému a nebo k difuznímu odrazu, což je ovlivněno mikroskopickou členitostí povrchu materiálu.

3.1.2 Lom

Pokud se světlo dostane na rozhraní dvou prostředí, jimiž může procházet, a za předpokladu, že nenastává totální odraz, je část světelné energie lomena do druhého prostředí a zbylá část je odražena zpět podle zákona odrazu. Úhel, pod kterým se paprsek světla láme, závisí na indexech lomu obou prostředí a řídí se Snellovým zákonem. Rovnice (3.1) je jednou z formulací Snellova zákona a na obrázku 3.1 je pak lom paprsku vyobrazen i s popisky k členům rovnice [13].

$$\sin \alpha_2 = \frac{n_1}{n_2} \sin \alpha_1 \quad (3.1)$$



Obrázek 3.1: Snellův zákon při lomu paprsku [41].

Totální odraz

Při průchodu paprsku do opticky řidšího prostředí může nastat, že je sinus úhlu lomu roven 1. V takovém případě je pak samotný úhel lomu 90° a paprsek je vychylován do plochy rozhraní. Úhel dopadu, který způsobuje lom paprsku do pravého úhlu od normály, se nazývá mezní úhel a je to největší úhel dopadu, při kterém ještě dochází k lomu do druhého prostředí. Při vyšším úhlu již vzniká tzv. totální odraz, kdy se veškerá energie paprsku od rozhraní odráží a do druhého prostředí tudíž neproniká.

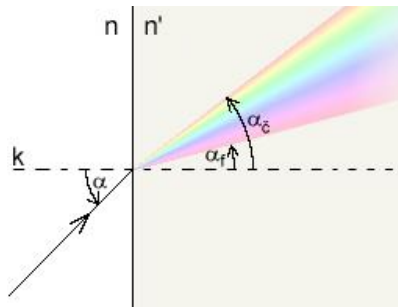
Fresnelovy rovnice

Pokud dochází při lomu světla zároveň i k odrazu, musí se jeho energie mezi lomený a odražený paprsek rozdělit. Poměr dělení závisí na indexech lomu obou prostředí, na úhlu dopadu a na směru

polarizace světla. Fresnelovy rovnice slouží k výpočtu tohoto poměru pro paprsky polarizovány kolmo a paralelně k rovině dopadu. Protože se v reálném světě většinou vyskytuje přibližně stejné množství kolmo a paralelně polarizovaných paprsků, počítají se hodnoty pro oba případy a výsledný poměr je pak určen jejich aritmetickým průměrem [35].

Disperze

Disperze je jev, při kterém závisí index lomu na vlnové délce paprsku a nastává při lomu do libovolného hmotného disperzního prostředí. Při disperzi dochází k rozkladu světla na jeho jednotlivé složky a proto se tomuto jevu říká také chromatická disperze [26]. Ilustrace jevu disperze při průchodu paprsku rozhraním je na obrázku 3.2.



Obrázek 3.2: Disperze světla lomem [39].

Absorpce světla

Průchodem paprsku světla prostředím dochází k útlumu jeho intenzity vlivem absorpce materiálu. U látky se stejnorodým rozložením absorpčního znečištění lze intenzitu po průchodu spočítat pomocí Lambertova-Beerova zákona zapsaného ve tvaru (3.2). Hodnota molárního absorpčního koeficientu je v něm závislá na vlnové délce světla, a proto lze výpočet v této formě aplikovat pouze na monochromatické záření[31]. Rovnice (3.3) je pak jen upravený tvar zákona pro výpočet úbytku energie.

$$I = I_0 e^{-\epsilon c d} \quad (3.2)$$

I intenzita paprsku po průchodu látkou
 I_0 intenzita vstupujícího paprsku
 d vzdálenost průchodu paprsku látkou
 α molární absorpční koeficient
 c molární koncentrace

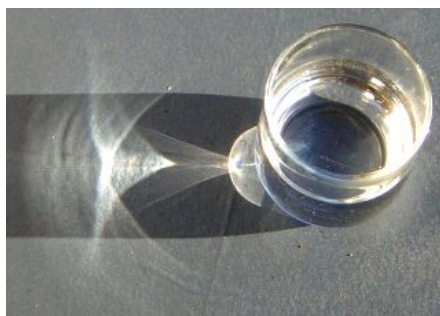
$$\frac{I}{I_0} = e^{-\epsilon c d} \quad (3.3)$$

Kaustiky

Kaustika je v optice ohraničená oblast světelných paprsků odražených či lomených zakřiveným povrchem nebo objektem a nebo také projekce těchto paprsků na jiný povrch [4]. Jedná se tedy o oblast křivky nebo plochy, která je vlivem vyšší koncentrace dopadajících paprsků zesvětlena. Na obrázku 3.3 je ukázka kaustik vznikajících průchodem paprsků přes sklenici s vodou.

3.1.3 Ambient occlusion

Zastínění okolím, nebo také ambient occlusion, vzniká v reálném prostředí v místech, kde je snížena přímá viditelnost na ostatní osvětlené objekty vlivem nějaké překážky nebo umístěním. Důvodem je světlo, které se ve scéně šíří vlivem nejen zrcadlových, ale i difuzních odrazů. Tento druh nepřímého osvětlení se souhrnně označuje jako ambientní a v důsledku mnohonásobných odrazů proniká i za překážky. Do míst, která jsou z různých směrů více blokována, se ho však dostane méně, a jeví se



Obrázek 3.3: Kaustiky způsobeny sklenicí s vodou [40].

proto tmavší. Příkladem zastínění okolím, se kterým se běžně setkáváme, jsou rohy místností, nebo obecně veškeré vnitřní rohy objektů.

3.2 Metody vizualizace

Metod vizualizace je několik a liší mezi sebou rychlostí tvorby výstupu, jeho kvalitou a realističností. Konkrétní volba metody pak závisí na požadavcích, které na aplikaci máme, protože žádná z nich není dostatečně univerzální. Jiná metoda je potřeba například pro vizualizaci strojírenské aplikace v reálném čase a jiná pro tvorbu fotorealistických snímků do filmu.

Metody se podle rychlosti dělí na reálné, které pracují v reálném čase a jsou schopny vytvořit alespoň 25 snímků za vteřinu, nebo offline, které potřebují k vytvoření výstupu nezanedbatelný čas.

Hledisko kvality určuje míru a druh šumu ve výsledném snímku a jedná se spíše o hodnocení vizuálního dojmu z výstupu. Bez šumu můžeme považovat pouze reálný svět, jelikož dnes neexistují techniky, které by do obrazu nezanášely určitý druh šumu. Například i fotografie obsahují šum způsobený velikostí zrněk stříbra ve filmu.

Podle realističnosti lze rozlišit metody na fotograficky přesné, fotorealistické a nerealistické. Za fotograficky přesné se považují fotografie reálného světa. Fotorealistické metody se snaží o vytvoření výstupu co nejvíce podobnému fotografii. Nerealistické metody jsou potom jejich opakem.

3.2.1 OpenGL, DirectX

Tato hardwarově akcelerovaná grafická API spadají do kategorie nerealistické vizualizace v reálném čase. Pro modelování scény podporují pouze základní primitiva typu bod, úsečka, trojúhelník a polygon a k osvětlení používají bodové zdroje světla. Označení polygonu za základní primitivum je třeba brát s jistým nadhledem, jelikož je u většiny grafického hardwaru vykreslován pomocí trojúhelníků.

Primitiva se vytvářejí specifikací jejich vrcholů a je možné na ně aplikovat transformace typu translace, rotace a zvětšení. Dále lze upravit jejich vzhled nastavením konstantní barvy nebo pomocí Phongova materiálu. Konstantní barva se použije ke stínování primitiva při vypnutém osvětlení. Pokud je osvětlení zapnuto, je při stínování aplikován Phongův osvětlovací model společně s nastaveným materiálem. V tomto případě je ovšem nutné mít u jednotlivých vrcholů definovány i normálové vektory povrchu, které jsou k výpočtu osvětlení nezbytné. Při stínování ploch lze v obou případech volit mezi plochým (jednoduchým) stínováním, Goraudovým stínováním nebo potažením vlastní texturou, která nahrazuje vyšší členitost povrchu objektu. Téměř všechny grafické karty dnes také podporují tzv. bump mapping, při němž se během stínování upravuje poloha normálového vektoru podle speciální textury s informacemi o hrboлатosti povrchu.

Kromě pevně definovaného osvětlovacího modelu je v OpenGL i v DirectX ještě omezení v počtu světelných zdrojů ve scéně. Lze jich současně použít maximálně osm a přestože existují grafické karty, které podporují světla více, není možné se na toto rozšíření spoléhat. Pokud bychom tedy chtěli v zobrazované scéně více zdrojů světla, je vhodné to vyřešit implementací v aplikaci.

Důležitým rysem těchto rozhraní je také neexistence vztahů mezi vkládanými primitivy na scéně. Mnoho optických jevů však tyto vazby vyžaduje a nelze proto přímo zobrazovat například stíny, lom či odraz světla. Existují však techniky, s jejichž pomocí lze některé jevy emulovat. Například zrcadlový povrch je možné vytvořit kopií a převrácením objektů za plochou odrazu a ostrých stínů se dosahuje vícenásobným renderováním scény při různě nastaveném osvětlení. Pro průhledné objekty bez lomu světla již rozhraní základní podporu obsahují v podobě míchání alfa kanálů. To způsobuje prolínání barev zakrývajících se objektů v závislosti na hodnotě alfa složky jejich barvy. Aby však zobrazení průhledných objektů proběhlo korektně, musí se geometrie vytvářet odzadu směrem k pozorovateli.

Jelikož modelování probíhá na velice základní úrovni, je pro pohodlnou práci s komplexními objekty vhodné využít správce scén. Za tímto účelem je v DirectX k dispozici tzv. retained mód, ve kterém je možné udržovat objekty v hierarchické struktuře grafu scény. Kvůli svým omezením však přestal být vyvíjen a pro obě rozhraní je proto nutné využít buď produktů třetích stran, nebo si vlastního správce scény a modelů vytvořit sami.

Vzhledem ke svým celkovým omezením nejsou tato rozhraní vhodná k tvorbě fotorealistického obsahu. Jejich primární zaměření je však na reálné zpracování, kde mají své využití i pro vizualizaci kapaliny. V oblastech technického charakteru, ve kterých je kladen důraz na simulační část, lze například částice kapaliny zobrazovat pomocí bodů a její povrch naznačit bez pokročilých optických jevů pomocí průhledných trojúhelníků. V reálném čase je pak možné procházet scénou a sledovat průběh simulace z potřebných úhlů. Zobrazení kapaliny je v současné době také důležité pro herní průmysl, v němž se k vykreslování používají výhradně tato rozhraní. V tomto případě jsou kladeny vysoké nároky na vizuální stránku výstupu při zachování rychlosti zobrazování. Zdánlivě realistické kvality se proto dosahuje napodobováním optických jevů za použití různých triků. Díky shaderům a vysokému paralelnímu výkonu grafických kartách lze dnes proto ve hrách spatřit například i disperzi světla nebo kaustiky.

3.2.2 Ray Tracing

Metoda sledování paprsku, neboli ray tracing, se zařazuje mezi fotorealistické metody nepracující v reálném čase. Vzhledem však k výkonu dnešních grafických karet, které umožňují využít svou víceprocesorovou architekturu pro obecné vysoce paralelizovatelné výpočty, se situace začíná měnit. Důkazem toho je engine pro interaktivní ray tracing od společnosti nVidia nazvaný OptiX [25], který má u procedurálně definované scény dosahovat snímkové frekvence i více jak 100fps. Během renderování trojúhelníkových modelů se ovšem, podle testů Ludvigsena a Elstera[19], počet snímků pohyboval od 20 do 67fps při použití grafické karty určené pro profesionální účely (konkrétně nVidia Quadro FX 5800). Takovýto rozsah snímků za vteřinu je pro plynulé interaktivní aplikace stále nedostačující. Než si tedy budeme moci na běžném hardwaru zahrát hru využívající k zobrazování výhradně metodu sledování paprsku, budeme si muset ještě nějakou dobu počkat.

Ray tracing zobrazuje scénu po pixelech a využívá lokálních vazeb mezi objekty k tvorbě stínů, k odrazu světla od reflexních povrchů nebo k lomu skrz průhledné objekty. Umožňuje použít libovolný počet světelných zdrojů, které jsou v jeho základním provedení bodové a vrhají proto ostré stíny. U průhledných objektů lze navíc aplikovat útlum intenzity světla vlivem absorpce materiálu. Postrádá však interakci na globální úrovni a nepodporuje proto ve scéně nepřímé osvětlení. Dalšími jeho nedostatky je vznik prostorového a časového aliasingu a absence plošných světelných zdrojů.

Ve své podstatě ray tracing simuluje reálný pohyb světelného paprsku ve scéně. Původně napodoboval skutečný svět vrháním mnoha paprsků ze zdrojů světla a některé z nich následně dopadly na zobrazovací plochu. Tento způsob je však velice neefektivní, jelikož v reálném světě je množství paprsků, které po odrazech ve scéně skončí v oku pozorovatele, zanedbatelné v porovnání s počtem paprsků vyslaných zdroji světla. Proto dnes pojem ray tracing značí metodu nazvanou Backwards ray tracing, neboli zpětná metoda sledování paprsku či zpětný ray tracing. Při ní jsou paprsky vysílány do scény pozpátku, tedy z oka pozorovatele přes jednotlivé body zobrazovací plochy. Během cesty paprsku se počítá barevný příspěvek od zasažených předmětů, který je po ukončení průchodu vrácen do zobrazovaného bodu.

Jelikož aplikace k této práci využívá pro vizualizaci kapaliny právě ray tracing, je v následující podkapitole popsán podrobněji.

Průchod paprsků scénou

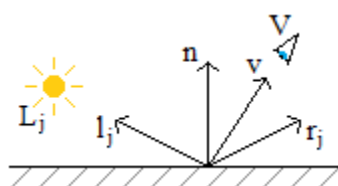
Při ray tracingu se můžeme setkat celkem se čtyřmi druhy paprsků – primární, stínový, reflexní a transparentní, přičemž od každého se očekává, že po vyslání vrátí ze scény získanou barvu.

Samotné sledování začíná vysláním primárního paprsku z virtuálního oka přes bod na zobrazovací ploše do scény. Pokud během své cesty scénou nezasáhne žádný předmět, vrátí jako svůj produkt barvu pozadí a přejde se na vyslání další primárního paprsku přes jiný bod. Pokud ovšem ale paprsek zasáhne nějaký objekt, je třeba získat v místě střetu barvu objektu.

Nejprve se vypočítá vliv přímého osvětlení posláním stínových paprsků z místa intersekce do všech světél. Pokud stínový paprsek ještě před světlem protne nějaký objekt, je dané světlo zastíněno a jeho příspěvek není do přímého osvětlení bodu započítán. Barvu přímého osvětlení lze pak počítat například jednoduše pomocí Phongova osvětlovacího modelu (3.4) [33]. Nejsme však vázáni jen na něj a můžeme použít například i náročnější model BRDF, Bidirectional reflectance distribution function, který je kromě úhlu osvětlení či pohledu od normály závislý ještě na směru pohledu.

$$C_D = \sum_{j \in N} \left\{ k_a L_{ja} + \frac{1}{a + bd + cd^2} [k_d L_{jd} (n \cdot l_j) + k_s L_{js} (v \cdot r_j)^\alpha] \right\} \quad (3.4)$$

- C_Dvýsledná barva přímého osvětlení Phongova modelu
- Nmnožina všech nezastíněných světél
- k_a, k_d, k_sambientní, difúzní a spekulární barevná složka materiálu
- L_{ja}, L_{jd}, L_{js}ambientní, difúzní a spekulární barevná složka světla j
- njednotkový vektor kolmý k povrchu v bodě střetu
- l_jjednotkový vektor směřující z místa střetu ke světlu j
- vjednotkový vektor směřující z místa střetu k pozorovateli (nebo také k počátku střetnutého paprsku)
- r_jjednotkový vektor udávající směr odraženého světla ze zdroje j
- αlesklost materiálu
- dvzdálenost bodu střetu od světla
- avelikost konstantního útlumu světla
- bvelikost lineárního útlumu světla se vzdáleností
- cvelikost kvadratického útlumu světla se vzdáleností



Obrázek 3.4: Znázornění vektorů k rovnici Phongova modelu (3.2) se zdrojem světla L_j a pozorovatelem V .

$$T = [\eta_r (N \cdot I) - \sqrt{1 - \eta_r^2 (1 - (N \cdot I)^2)}] N - \eta_r I \quad (3.5)$$

- T vektor směru transparentního paprsku
- η_rrelativní index lomu (poměr indexu lomu prostředí, ze kterého paprsek přichází, k indexu lomu prostředí, do kterého se láme)
- Njednotková normála v bodě střetu
- Ijednotkový vektor směřující z místa střetu k počátku dopadajícího paprsku

Máme-li získanou hodnotu přímého osvětlení, následuje vyslání sekundárních paprsků. Je-li objekt průhledný, v závislosti na jeho indexu lomu a na velikosti úhlu dopadu se část paprsku láme do objektu a část se od něj odráží. Směr transparentního paprsku se určí pomocí (3.5), jehož odvození naleznete v [6]. Je-li výsledek odmocniny v rovnici imaginární číslo, dochází k totálnímu odrazu a v tomto případě je energie reflexního paprsku 100% a transparentní paprsek není vůbec vytvořen. Pokud však k totálnímu odrazu nedochází, musí se energie lomeného a odraženého paprsku spočítat pomocí Fresnelových rovnic. Jelikož tato energie závisí ještě na polarizaci dopadajícího světla, použije se při tzv. *polarizaci s*, kdy jsou paprsky polarizovány kolmo k rovině dopadu, Fresnelova rovnice (3.6), a při *polarizaci p*, kdy jsou paprsky polarizovány rovnoběžně s rovinou dopadu, naopak rovnice (3.7). V reálném světě se většinou vyskytuje stejné množství paprsků obou druhů polarizace, a proto se energie pro odražený paprsek počítá jako jejich aritmetický průměr (3.8). Energie transparentního paprsku je pak jednoduše zbytek do 100%, tedy zde do hodnoty 1 (3.9).

$$E_{Rs} = \left[\frac{\eta_1 \cos \alpha_1 - \eta_2 \sqrt{1 - (\eta_r \sin \alpha_1)^2}}{\eta_1 \cos \alpha_1 + \eta_2 \sqrt{1 - (\eta_r \sin \alpha_1)^2}} \right]^2 \quad (3.6)$$

$$E_{Rp} = \left[\frac{\eta_1 \sqrt{1 - (\eta_r \sin \alpha_1)^2} - \eta_2 \cos \alpha_1}{\eta_1 \sqrt{1 - (\eta_r \sin \alpha_1)^2} + \eta_2 \cos \alpha_1} \right]^2 \quad (3.7)$$

$$E_R = \frac{E_{Rs} + E_{Rp}}{2} \quad (3.8)$$

$$E_T = 1 - E_R \quad (3.9)$$

- E_{rs} energie odraženého paprsku s polarizací *s*
- E_{rp} energie odraženého paprsku s polarizací *p*
- E_R průměrná energie odražených paprsku s polarizací *r* a *s*
- E_T energie lomeného paprsku
- η_1 index lomu prostředí, ze kterého paprsek přichází
- η_2 index lomu prostředí, do kterého se paprsek láme
- η_r relativní index lomu
- α_1 úhel dopadu (značení je shodné s obrázkem 3.1)

Směr odraženého paprsku se řídí zákonem odrazu a odvození jeho vektorového výpočtu (3.10) lze opět nalézt v [6]. Objekt s neprůhledným reflexivním povrchem pak musí mít ještě ve svém materiálu explicitně zadáno množství energie odraženého paprsku.

$$R = V - 2(V \cdot N)N \quad (3.10)$$

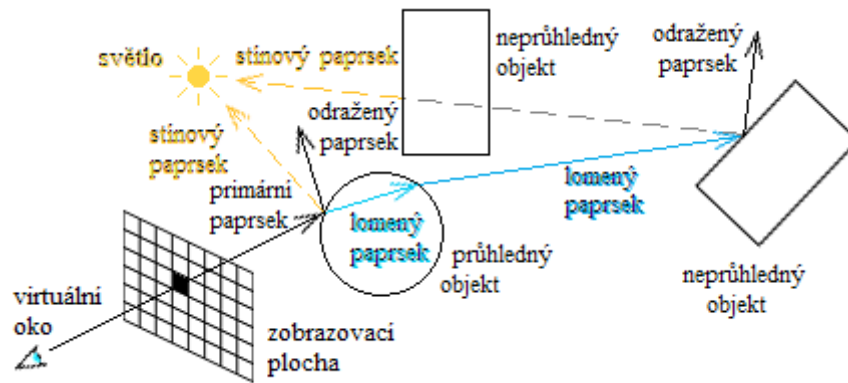
- R vektor směru odraženého paprsku
- Njednotkový normálový vektor v bodě střetu
- Vjednotkový vektor směřující od počátku dopadajícího paprsku do místa střetu

$$C = C_D + E_R C_R + E_T C_T \quad (3.11)$$

- C výsledná barva střetnutého bodu
- C_D barva přímého osvětlení
- E_R energie odraženého paprsku
- C_R barva vrácená odraženým paprskem
- E_T energie lomeného paprsku
- C_T barva vrácená lomeným paprskem

Zpracování sekundárních paprsků probíhá stejně jako paprsků primárních a jejich vysláním tedy vzniká v algoritmu rekurze. Proto se ray tracing také někdy označuje jako rekurzivní ray tracing. Podmínkou pro zastavení rekurzivního zpracování pak může být minimální energie paprsku nebo maximální hloubka rekurze.

Výsledná barva vrácená paprskem od zasaženého předmětu se nakonec získá sečtením barvy z přímého osvětlení a barev vrácených sekundárními paprsky (3.11) a výpočet pokračuje paprskem přes další bod na zobrazovací ploše. Příklad paprsku procházejícího scénou je znázorněn na obrázku 3.5.



Obrázek 3.5: Ukázka průchodu paprsku scénou při ray tracingu.

Distributivní ray tracing

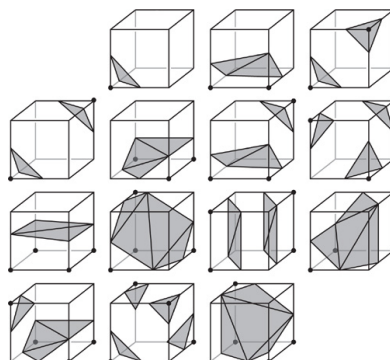
Distributivní ray tracing, nebo také stochastický, rozšiřuje základní ray tracing vysláním vždy několika paprsků náhodně vychýlených místo jednoho. Tím odstraní prostorový aliasing a nahradí ho šumem, který je pro lidské oko na pohled mnohem příjemnější.

Distributivní ray tracing však hlavně umožňuje použití plošných zdrojů světla a s nimi spojené měkké stíny, dále pak neostrý odraz a lom, motion blur, neboli rozmazání pohybem, efekt hloubky ostrosti a disperzi. Záleží ovšem na konkrétní aplikaci, které všechny jevy bude implementovat.

3.3 Metoda Marching Cubes

Pro účely simulace je kapalina uložena ve formě pole nebo částic. Potřebujeme-li ji však zobrazit pomocí OpenGL nebo ray tracerem optimalizovaným na renderování trojúhelníků, musíme tuto reprezentaci převést na polygonální. Právě k tomu je určena metoda Marching Cubes, která umí z implicitně zadaného povrchu vytvořit hraniční reprezentaci s užitím trojúhelníků.

Algoritmus rozdělí prostor, ve kterém se kapalina může nacházet, pravidelnou mřížkou a v jednotlivých vrcholech určí hodnotu implicitní funkce částic. Ta má vně kapaliny kladnou hodnotu úměrnou vzdálenosti od jejího povrchu, uvnitř hodnotu zápornou a na povrchu nulovou.



Obrázek 3.6: Zobrazení 14 základních krychlí algoritmu Marching Cubes [38].

Každé buňce mřížky pak Marching Cubes přiřadí krychli s předpočítanými trojúhelníky aproximujícími průběh povrchu podle uspořádání vnitřních a vnějších vrcholů. Ukázka několika předpočítaných krychlí je na obrázku 3.6. Bourke [2] například používá pro označení jednotlivých průběhů povrchu osmibitové číslo, u kterého hodnota každého bitu náleží jednomu z osmi vrcholů buňky a je nastavena na jedničku, pokud je vrchol uvnitř nebo na povrchu kapaliny, a na nulu, pokud je mimo ní. Toto číslo následně slouží jako index do tabulky trojúhelníků, ve kterých jsou uchovány jejich předpočítané body pro reprezentaci povrchu dané buňky. Vrcholy trojúhelníků jsou uloženy ve formě indexů hran, na nichž se vrcholy nacházejí.

Přesná pozice vrcholu na hraně se určí interpolací hodnot implicitních funkcí bodů této hrany. Po vytvoření normál trojúhelníků vektorovým součinem jeho dvou hran bychom již v tomto kroku měli, v závislosti na použité velikosti buňky, poměrně kvalitní reprezentaci povrchu objektu. Chceme-li však dosáhnout ještě hladšího průběhu, provádí se navíc výpočet normál vrcholů jako interpolace normál trojúhelníků majících tento vrchol společný.

Velice dobře a podrobně je algoritmus popsán v práci Mariana, J. [20], na kterou případné zájemce o bližší informace odkazují.

4 Implementace simulace a vizualizace kapaliny

Součástí této práce je simulátor kapaliny, který ke své činnosti využívá výše popsané metody. Tato kapitola popisuje některé jeho implementační detaily a zejména se zaměřuje na problémy, které bylo nutné během tvorby řešit.

Při implementaci lze vytvořit simulaci a vizualizaci jako samostatné celky, je však výhodné postavit je na vzájemných znalostí použitých metod. U simulátorů bez informace o povrchu kapaliny to například bývá právě algoritmus pro jeho rekonstrukci, který úzce spolupracuje s oběma úseky. Tohoto principu využívá i vytvořený simulátor.

Simulační část aplikace je založena na metodě SPH podle práce Claveta a spol.. K řešení hranic byly použity kopírované neviditelné hraniční částice a k extrahování povrchu kapaliny z částic posloužila metoda Marching Cubes. Vizualizace je k dispozici dvojího typu. K zobrazení aktuálního stavu se využívá OpenGL a výstup ve vysoké kvalitě probíhá pomocí distributivního ray tracingu. Aplikace není zaměřena na reálnodobé zpracování a proto při její tvorbě nebyly implementovány různé optimalizace. Na úseky, které jsou vhodné ke zrychlení, je však upozorněno a jsou případně diskutovány dostupné způsoby vylepšení.

Aby při práci s aplikací nedocházelo k zablokování uživatelského rozhraní vlivem náročného výpočtu, musí být jeho provádění rozděleno na dílčí fáze. Nejkritičtějšími místy zpracování je renderování výstupu pomocí ray tracingu a jeho rozdělení na výpočet po jednotlivých pixelech snímku by mělo být dostačující. Při použití supersamplingu je ale nutné počítat s kvadratickým nárůstem časové složitosti se zvyšováním jeho hodnoty. V aplikaci bylo proto použito rozdělení po řádcích subpixelů v jednom pixelu. K odstranění závislosti na nastavení aplikace je však vhodné dělení po stabilním počet subpixelů.

Simulační krok je v aplikaci prováděn v celku a proto u slabších počítačů dochází k zablokování jejího ovládání i při nižším počtu částic. K implementaci se nabízí základní rozdělení na dílčí fáze výpočtu SPH. Srovnání doby trvání jednotlivých úseků v tabulce 4.1 však ukazuje, že přes 97% času je rozloženo mezi výpočet pružin a ustálení dvojí hustoty. Aby tedy mělo rozdělení smysl, musí se provést na úrovni částic uvnitř těchto funkcí.

Časy měření byly získány průměrem z 40 sousledných časových kroků pro obě skupiny částic zvlášť. Jeho výsledky jsou však pouze orientační pro získání přehledu o rozdělení výpočetního času a nelze v něm porovnávat hodnoty skupin mezi sebou, jelikož jsou ovlivněny aktuálním počtem okolních částic, které jsou použity při výpočtech.

Počet částic	Aplikace sil		Aplikace viskozity		Aplikace pružin		Aplikace hustoty		Aplikace kolizí		Celkem
	čas[s]	[%]	čas[s]	[%]	čas[s]	[%]	čas[s]	[%]	čas[s]	[%]	
5000	0,0004	0,02	0,0287	1,31	0,2849	12,98	1,8798	85,65	0,0008	0,04	2,1946
10000	0,0040	0,11	0,0720	2,04	1,2185	34,45	2,2406	63,35	0,0020	0,06	3,5371

Tabulka 4.1: Časové rozložení výpočtu mezi jednotlivými úseky metody SPH.

4.1 Simulační část

Při implementaci simulační části bylo vhodné vytvořit základní třídy pro objekt simulátoru, kapaliny a částice. Simulátor v sobě uchovává jednotlivé kapaliny a obstarává výpočet samotné simulace. V objektech kapalin jsou pak uloženy jejich příslušející částice a pružiny mezi nimi. Jelikož se při výpočtech musí z kapaliny jednotlivé částice získávat, jsou v simulátoru ze všech kapalin uloženy ještě v samostatném poli, ze kterého jsou využívány při úpravách nezávislých na kapalině. Tato

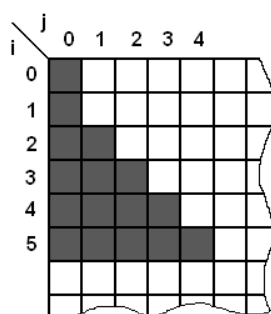
optimalizace však nepřináší téměř žádné zrychlení a vyžaduje naopak zavedení speciální správy přidávaných a odebíraných částic a kapalin.

Každý objekt částice v sobě obsahuje svůj identifikátor, polohu, rychlost a identifikátor do pole pružin. Kromě těchto vlastností je u nich potřeba mít možnost uložit ještě další dvě hodnoty polohy. První slouží k zazálohování aktuální polohy na začátku každého simulačního kroku a při zpracování kolizí je z ní určena rychlost použitá u modelu mokrého tření s nulovou restitucí. Do druhé se dočasně ukládá nová poloha částice vypočtená během aplikace metody zachování dvojí hustoty, aby se posuny vlivem tlaku počítaly v závislosti na hustotě kolem částice správně. Identifikátor částice se využívá k porovnávání částic mezi sebou a může být nahrazen identifikátory samotných objektů, pokud je lze ve zvoleném implementačním jazyce srovnávat podle velikosti.

Pružiny

Pružiny jsou v kapalině zastoupeny pouze hodnotami svých klidových délek. Uchovávají se v dvourozměrném poli, ve kterém jsou dostupné pro každou uspořádanou dvojici částic přes jejich indexy pružin. Protože mezi částicemi mohou vznikat dynamicky, je velikost pole při stálém počtu částic neměnná, aby nedocházelo k velkému zpomalení při alokaci a uvolňování paměti. Pružina má proto hodnotu své klidové délky, pokud mezi částicemi existuje, a nulovou hodnotu, pokud jsou na sobě nezávislé.

Jelikož mezi dvěma částicemi má smysl uchovávat vždy pouze jednu pružinu, je z pole využita jenom polovina jeho hodnot. Prostor pole již při 20000 částicích zabírá v paměti 1,5GB a s jejich zvětšujícím se počtem velikost kvadraticky roste. Je tedy vhodné vytvářet pole jen o poloviční velikosti a při dynamické alokaci jeho druhý rozměr vytvářet jen do velikost využitých pružin. Nezmění se tím práce s polem, ale musí se při jeho použití striktně dodržovat pořadí indexů, aby nedocházelo k přístupu do nealokované oblasti paměti. Symbolické znázornění takto vytvořeného pole je na obrázku 4.1. Tmavá políčka v něm představují jeho alokované buňky a světlá naopak volnou paměť.



Obrázek 4.1: Alokované pole pružin poloviční velikosti pro 6 částic. Tmavá políčka značí alokované buňky pole. Při přístupu do $pole[i][j]$ přes identifikátory pružin i a j musí být $i > j$. Buňka $[0][0]$ bude nevyužitá, ale musí být alokována pro správné umístění pole v paměti.

Sousední částice

Kritickým místem ve výpočtu simulace je vyhledávání sousedních částic. V kapitole 7.1 práce Claveta a spol.[5] se doporučuje částice ukládat do pravidelné mřížky a sousedy získávat z okolních buněk. Velikost buňky je určena interakční vzdáleností částic. V aplikaci je pro tento účel vytvořen objekt manažeru částic, který je vlastní pro každou kapalinu. Částice jsou v něm uchovány v trojrozměrném poli pevné velikosti a za účelem jednoduchosti jsou při každé změně polohy všechny částice opětovně roztrženy do jednotlivých buněk. To však znamená zbytečný nárůst času výpočtu a je proto doporučeno ošetřovat pouze změny v poli při přechodu částic mezi buňkami. Pevně daná velikost pole také značně omezuje prostor dostupný pro pohyb částic. Proto Clavet a spol. ukládají buňky do hašovacích tabulek podle metody Teschnera a spol.[32] pro optimalizované hašování prostoru.

Hranice, kolize

K vyřešení problému hranic jsou v aplikaci zavedeny neviditelné hraniční částice. Jejich vytvoření je ponecháno jednotlivým objektům a každý má proto při vložení do scény možnost přidat i své hraniční částice. O jejich uchovávání se pak stará simulátor, a aby bylo možné mezi nimi efektivně vyhledávat sousedící částice podobně jako u kapaliny, je pro ně vytvořen vlastní objekt manažeru částic. K uspořádání hraničních částic je u všech objektů použita pravidelná mřížka s rozstupem určeným experimentálně podle metody popsané v kapitole 2.3.3 v sekci Neviditelné hraniční částice.

Reakci na kolize obstarávají v aplikaci rovněž jednotlivé objekty. Při průniku částice kapaliny na ni aplikují model mokrého tření s nulovou restitucí s hodnotou tření uloženou v daném objektu. Jelikož by tření mělo záviset i na kapalině, bylo by vhodnější uchovávat vlastnost tření i v ní a jako výslednou hodnotu použít například jejich součin. Takovéto skládání ovšem není fyzikálně založené a slouží pouze k modelování vztahu tření mezi objektem a kapalinou.

K udržení kapaliny v prostoru vyhrazeném rozměry pole manažeru částic byly kolem něj vytvořeny hraniční zdi. Jelikož je ale kontrolován přístup mimo pole, nemusí být zdi posunuty způsobem popisovaným v sekci Hraniční podmínky kapitoly 2.3.2.. Aby však měly jejich hraniční částice vliv na hustotu, musí být umístěny uvnitř pole kvůli výpočtu okolních buněk při vyhledávání sousedních částic.

Časový krok simulace

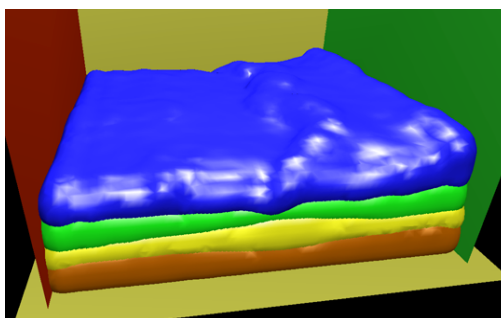
Velikost časového kroku může být buď nastavena staticky, nebo určována dynamicky podle uběhnutého času. Variabilní velikost časového kroku je vhodná pro aplikace běžící v reálném čase k zachování plynulosti pohybu. Přestože je možné ji ve vytvořeném simulátoru zapnout, není kvůli jeho nízké rychlosti použitelná.

U staticky určeného časového kroku se ukázaly jako vhodné hodnoty do velikosti 0,1 a pro kvalitnější simulaci problému protržení přehrady dokonce do velikosti 0,05. Při vyšších hodnotách dochází u hranic již ke zdatelnému stlačení částic vlivem použitého kolizního systému. Proto by měl být větší časový krok rozdělen na menší, se kterými dojde následně k několikanásobnému spuštění simulace ve smyčce. Velikost časového kroku zde záměrně udávám jako bezrozměrnou veličinu, jelikož pro převod na reálný čas závisí na použitém měřítku simulace.

Interakce více kapalin

V aplikaci byl do výpočtu dvojí hustoty implementován algoritmus podle Mariana, J. [20] s úpravami uvedenými v kapitole 2.3.4 a lze v ní proto simulovat interakci libovolného množství různých mísitelných kapalin. Ukázka ze simulace čtyř na sobě plovoucích kapalin s různými hmotnostmi je na obrázku 4.2.

U neviditelných hraničních částic se aplikuje metoda kopírování vlastností, pro kterou byly vytvořeny u každé kapaliny, pro rychlejší přístup manažeri jejich neviditelných částic. Ty jsou do nich ukládány po určení jejich příslušnosti ke konkrétní kapalině vždy před aplikováním hustoty. Jelikož neviditelné hraniční částice obecně přináší do výpočtu značné zpomalení, je tato oblast výhodná pro zavedení optimalizací. Kromě způsobu rozdělování do samostatných manažerů lze místo nich použít jeden neměnný pro všechny hraniční částice, jako při použití jedné kapaliny. Pak je



Obrázek 4.2: Vrstvy čtyř kapalin s různou hmotností.

ovšem nutné, při jejich získávání pro výpočet tlaků, odlišit výpočet na částice stejných a různých kapalin. Volba závisí hlavně na implementaci manažeru a jeho rychlosti přidávání a odebrání částic.

4.2 Vizualizační část

Vizualizace kapaliny je v aplikaci rozdělena na dvě fáze. V první je metodou Marching Cubes vytvořen trojúhelníkový povrch kapaliny, který lze jednoduše zobrazovat pomocí OpenGL. Ve druhé fázi se pak renderuje scéna s využitím metody distributivního sledování paprsku.

4.2.1 Marching Cubes

Při implementaci metody Marching Cubes byly použity postupy, které jsou zmíněny v kapitole 3.3, včetně interpolace normál pro hladší průběh povrchu.

Uložení v paměti

Mřížka, do které se ukládají hodnoty implicitní funkce a mezivýpočty pro interpolaci normál, se skládá z buněk a jejich vrcholů podle doporučení Mariana, J.[20]. V aplikaci jsou buňky i vrcholy vygenerovány pro celý simulační prostor do trojrozměrného pole neměnné velikosti. Právě proto je velikost těchto buněk v poměru k interakční vzdálenosti částic velice důležitá. Musí být totiž dostatečně malá, aby bylo možné zobrazit i samostatné částice, a zároveň nesmí jejich uložení vyčerpat dostupný paměťový prostor. Již při počtu 100 buněk na jeden rozměr je potřeba pro pole alokovat 0,5GB paměti. Bylo by proto vhodné toto pole udržovat dynamicky. Jelikož se povrch počítá jen v oblasti částic, nabízí se možnost integrace Marching Cubes s hašovanou verzí pole pro ukládání částic. Buňky by se tím vytvářely jen v místech, kde se skutečně může nacházet povrch kapaliny, a zároveň by se odstranilo pevné ohraničení výpočtu, stejně jako tomu bylo u statické reprezentace manažeru částic.

Do velikosti paměti pro uložení mřížky nebyla při předchozích úvahách započtena paměť potřebná pro uložení vygenerovaných trojúhelníků povrchu. Jelikož jich v každé buňce může být vytvořeno až 5, mohou teoreticky obsadit stejný paměťový prostor jako pole samotné. Takováto situace však nemůže nastat již z principu velikosti buňky stanovené několikanásobně menší než velikost interakční vzdálenosti částic. Reálně se velikost paměti potřebná pro uložení trojúhelníků pohybovala v řádu jednotek, maximálně desítek procent velikosti alokovaného pole.

Počet trojúhelníků se v závislosti na povrchu kapaliny může během simulace drasticky měnit a jejich dynamické vytváření a rušení by znamenalo výrazné zpomalení. V aplikaci se proto při snížení jejich počtu objekty trojúhelníků nevolňují a jsou při vytváření povrchu využívány opakovaně. Pouze pokud jich je potřeba více, než je současný počet, jsou pro ně vytvářeny nové objekty.

Problém hranic

Povrch kapaliny není v aplikaci u hranic objektů ošetřen a prostupuje proto do jejich oblastí. Pokud se ve scéně nenachází jiné průhledné předměty nebo příliš tenké překážky, nemá takovéto chování na výsledný výstup žádný vliv. Pro správné zobrazení kapaliny s průhlednými objekty však musí být pronikající povrch ořezán. K tomuto účelu existuje rozšíření algoritmu Marching Cubes o průnik s pevnými objekty a popisují ho samotní autoři algoritmu Lorensen a Cline v kapitole 5.2 své práce[18].

Jelikož v aplikaci použitý kolizní systém při simulaci dovoluje částicím přístup až těsně k hranicím překážek, může být více jak polovina objemu kapaliny schována uvnitř objektů. Proto, aby u hranic nedocházelo ke zmenšování kapaliny, měl by se skrytý objem kompenzovat. Toho by šlo docílit změnou vzdálenosti isoplochy povrchu dynamickým určováním hodnoty isolevelu. Je však nutné počítat s celkovým objemem propojených částí kapaliny. Úprava bude muset být zároveň provedena několikrát, jelikož při zvětšení vzdálenosti povrchu může dojít k novému propojení kapalin nebo k průniku do dalších objektů. Ve výsledku se proto ukazuje jako vhodnější řešení při kolizích udržovat odstup částic od okolních hranic.

Problém více kapalin

Při zavedení více kapalin se musí mřížka Marching Cubes uchovávat pro každou kapalinu zvlášť. Kromě nárůstu alokované paměti je však větší problém úprava algoritmu pro výpočet povrchu více mísitelných kapalin.

Řešení, které pro tento účel nabízí Marian v kapitole 3.2.13 své práce[20], není například obecně použitelné. Využívá rozšíření výpočtu funkce isoplochy o částice druhé kapaliny s vhodně nastaveným parametrem jejich vlivu, aby mezi kapalinami vytvořil co nejtěsnější průběh povrchu. Lze ho tedy využít jen v případě vzájemně neprostupných látek.

Při implementaci více kapalin jsem se proto pokoušel různými způsoby upravit výpočet isoplochy odečítáním vlivu částic pronikající kapaliny. Upravený povrch vždy však buď vůbec nekopíroval průběh druhé kapaliny, nebo se v něm dokonce vytvářely díry. Alespoň zdánlivě uspokojivých výsledku jsem tedy u tohoto problému v aplikaci nakonec nedosáhl a řešení se mi bohužel ani nepodařilo nikde najít. Přesto však určitě existuje a je tedy pravděpodobně obecně známé, jelikož se v pracích simulujících více kapalin algoritmus Marching Cubes běžně používá[15, 17, 23]. Nejsou však nikdy zmiňovány užitečné implementační detaily, které by mi napomohly k úspěšnému vyřešení problému rozhraní mezi kapalinami.

I když vytvořená aplikace nabízí možnost simulace více kapalin, jejich povrchy jsou v ní na sobě nezávislé a není proto možné kvalitně renderovat více průhledných kapalin zároveň.

4.2.2 Ray Tracing

Protože reflexivní a transparentní plochy vytváří ve scéně nepřímé zdroje osvětlení, ovlivňují polohu a tvar vržených stínů. S nepřímými zdroji však základní metoda sledování paprsku nepočítá a proto je při implementaci ray traceru vhodné během výpočtu osvětlení stínové paprsky skrz průhledné objekty propouštět. Výsledná scéna sice není fyzikálně přesná, ale vizuálně stále vytváří poměrně kvalitní dojem. Zároveň lze takto vytvořený ray tracer poměrně snadno rozšířit o výpočet osvětlení některou z metod globálního osvětlení a vrátit scéně fyzikální podstatu. Například velice dobře popisuje Dwight Folllick na stránkách svého projektu princip implementace metody sledování fotonů do tradičního ray traceru[8].

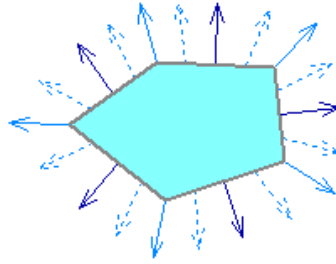
Barva kapaliny

Pro výpočet osvětlení je v aplikaci implementován Phongův osvětlovací model popsáný v kapitole 3.2.2, zjednodušený o výpočet útlumu světla, kdy je v rovnici (3.4) konstantně nastaven parametr $a = 1$ a parametr $b = c = d = 0$. V materiálech objektů se udržují pro všechny tři složky Phongova materiálu barvy ve tvaru RGBA, přičemž hodnota alfa slouží pouze pro vykreslování pomocí OpenGL.

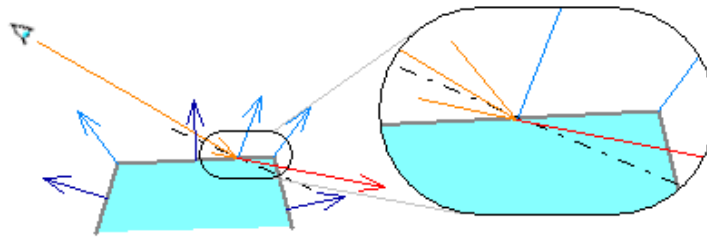
Určování jednotlivých barev samostatně pro ambientní, difuzní a spekulární složku Phongova materiálu je výhodné pro neprůhledné povrchy a pro světla, jelikož umožňují příhodně modelovat jejich vlastnosti. U transparentních objektů je však nutné hodnoty ambientní a difuzní složky volit velice opatrně, protože mají vliv na jejich výslednou průhlednost. To je zapříčiněno skládáním barev pomocí rovnice (3.11) a pro absolutně čiré objekty proto musí být tyto složky nulové. Z toho důvodu byla pro transparentní materiály vytvořena v aplikaci ještě jedna hodnota barvy podle článku Jacco Bikker[1], kterou se násobí výsledná barva při vstupu paprsku do průhledného objektu. Ambientní a difuzní složky potom mohou zůstat nulové a obarvení světla procházejícího přes objekt lze jednoduše určovat touto novou hodnotou barvy. Přestože, jak Jacco Bikker píše, toto chování není úplně korektní, vytváří velice kvalitní výsledky výstupu.

Intersekcce paprsku s kapalinou

Jelikož jsou v algoritmu Marching Cubes trojúhelníky vytvářeny v pravidelné mřížce, lze s výhodou tohoto dělení prostoru využít při hledání intersekcce paprsku s povrchem kapaliny. Proto byl v aplikaci implementován algoritmus 3DDDA, což je rozšíření algoritmu DDA, neboli Digital Differential Analyser, do trojrozměrného prostoru. Algoritmus DDA slouží běžně k rasterizaci úsečky v dvojrozměrném prostoru. Jeho 3D verze lze však použít k procházení buněk zasažených paprskem v



Obrázek 4.3: Interpolované normály povrchu při metodě Marching Cubes. Tmavě modře jsou zakresleny skutečné normály trojúhelníků, světle modře interpolované normály. Čárkovaně jsou naznačeny normály, které se interpolují pro jednotlivé body z normál vrcholů.



Obrázek 4.4: Chybný odraz do kapaliny vlivem interpolace normál

oblasti rozdělené pravidelnou mřížkou. Při implementaci jsem vycházel z dokumentu pro předmět Computer Graphics Brigham Youngovi Univerzity[3], ve kterém je popsána 2D verze 3DDDA přímo upravená pro průchod paprskem. Přesně v této podobě by ovšem paprsek začínal vždy na okraji mřížky a protože mnoho jich vznikne, díky odrazu a lomu, právě v ní, je vhodné posouvat počátek průchodu do buňky, ze které paprsek vychází.

Problém interpolace normál u metody Marching Cubes

Použití interpolovaných normál trojúhelníků u metody Marching Cubes vede k vytvoření průběhu povrchu způsobem naznačeným na obrázku 4.3. To má při renderování ray tracerem za následek vznik určitých anomálií, se kterými je nutné při implementaci počítat.

Jelikož se směr interpolované normály liší na daném místě od skutečné normály trojúhelníku, může se odražený paprsek dostat do roviny lomu a naopak lomený paprsek do roviny dopadu. Při hledání intersekcí v kapalině je proto nutné kontrolovat stranu příchodu paprsku k povrchu a vyloučit případy, kdy došlo k jeho protnutí zevnitř, ovšem paprsek se přitom v dané kapalině nenacházel, nebo kdy došlo ke střetu zvenku, ale paprsek již uvnitř kapaliny byl. Znázornění chybně lomeného paprsku vlivem interpolace normály je na obrázku 4.4.

Kvůli odlišnosti normál může také nastat, že úhel paprsku od normály bodu střetu je větší než 90 stupňů. V takovém případě je nutné vrátit s bodem střetu místo interpolované normály skutečnou normálu trojúhelníku, jelikož použití interpolované normály by vedlo k chybnému výpočtu jednak u osvětlení a jednak u směru odraženého a transparentního paprsku.

Největším problémem při použití interpolovaných normál je však vznik prostorových nespojitostí vlivem skokových změn na horizontu povrchu. Tento problém však není možné úplně odstranit, ale lze ho zmírnit jemnější mřížkou algoritmu Marching Cubes.

Nekonečné totální odrazy

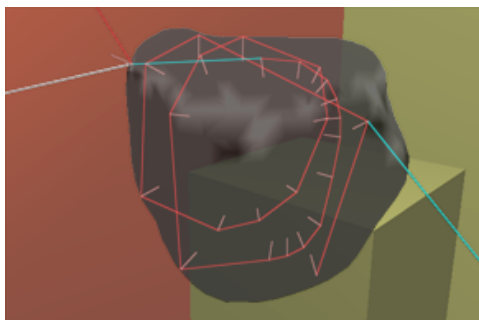
Když se paprsek zlomí do uzavřeného povrchu kapaliny, například do kapky, může nastat situace, kdy je v ní odražen totálními odrazy až do vyčerpání maximální hloubky rekurze. Přitom nemusí během své cesty z okolí posbírat žádnou barevnou informaci a výsledný pixel proto zůstane černý. Zvýšením hloubky rekurze se šance na vznik tohoto jevu snižuje, není však možné takto dosahovat řešení, jelikož vždycky může nastat situace, kdy se paprsek v kapalině vlivem odrazů zacyklí.

Způsob, jak tento problém úplně odstranit, bohužel není. Lze však učinit jisté kroky, které šanci na jeho výskyt radikálně sníží, nebo je možné jeho vzniku určitými opatřeními předejít.

Tato opatření, které je nutné provést k zamezení jeho výskytu, jsou dvě. Jednak v aplikaci musí být zavedena absorpce světla v transparentním materiálu a zároveň se nesmí ve scéně vyskytovat kapalina bez minimální absorpce světla. Hodnota této minimální absorpce je pak nepřímou úměrou závislá na maximální hloubce rekurze. Bylo by možné ji určit matematicky z minimálního rozměru pravidelného n -stěnného hranolu, který je možné vytvořit způsobem, aby stěny v něm svíraly úhel totálního odrazu při použitém indexu lomu. Analytické stanovení takovéto rovnice by ale bylo velice náročné, ne-li nemožné, a mnohem praktičtější je zvolit hodnotu podle vizuálních výsledků.

Pokud bychom přesto požadovali zobrazení ideálně čiré nebo nedostatečně absorbující kapaliny, musí se s výskytem nekonečných odrazů počítat a z výsledného obrazu je třeba ho odstranit. Lze tak učinit zavedením metody supersamplingu, při které je pixel rozdělen na více subpixelů. Pokud následně označíme paprsky, které dosáhnou maximální hloubky rekurze, můžeme je z výpočtu výsledného pixelu odebrat a na výstup se proto nedostanou. V případě, že by se i po použití supersamplingu v obraze vyskytovaly vadné pixely, je možné je ještě nahradit interpolací okolních pixelů.

Ukázka nekonečných totálních odrazů vzniklých při renderování kapaliny je na obrázku 4.5. Bílý paprsek, který zleva vstupuje do kapaliny, je primární, paprsky tyrkysové barvy označují lom a červené naopak odraz. Při výpočtu tohoto paprsku byla maximální hloubka rekurze nastavena na 25.



Obrázek 4.5: Nekonečné totální odrazy.

Absorpce světla

Úbytek energie paprsku při průchodu materiálem absorbujícím světlo je v aplikaci implementován Lambertovým-Beerovým zákonem, popsaným v kapitole 3.1.2. Při každém vržení paprsku se, po nalezení bodu intersekce s objektem, spočítá ze vzdálenosti úbytek energie podle vzorce (3.2), kterým je následně zmenšena energie pro rekurzivní paprsky. Aby měl ovšem úbytek úseku vliv i na výstup, musí jím být vynásobena i výsledná barva vrácená paprskem z výpočtu (3.11).

Pro snadnější nastavování absorpce materiálu doporučuje Jacco Bikker ve svém článku[1] spojit molární absorpční koeficient α a molární koncentraci c do jedné konstanty pro hustotu materiálu. Osobně jsem však v aplikaci nechal koeficienty oddělené, abych mohl podobným materiálům nastavovat stejný koeficient α a mezi sebou je rozlišovat různou hodnotou molární koncentrace.

Kromě spojení do jedné společné hodnoty koeficientu, Jacco Bikker zároveň také používá hodnotu absorpce pro každou barevnou složku zvlášť. Tento přístup sice odpovídá koncepci pro využití Lambertova-Beerova zákona u monochromatického záření, pro ray tracer s jedním paprskem reprezentujícím všechny barevné složky ale nemá velký význam a k započítávání do energie paprsku je výhodnější uchovávat koeficient společný pro všechny barevné komponenty. V případech, kdy však potřebujeme ovlivňovat úbytek světelné energie v závislosti na barvě, je příhodné právě takovéto rozdělení na složky u absorpce.

4.2.3 Distributivní Ray Tracing

Základní ray tracer byl v aplikaci rozšířen o metody distributivního ray tracingu při výpočtu osvětlení, odrazu a lomu a umožňuje díky tomu zobrazovat polostín vržený plošným zdrojem světla a objekty s neostrým reflexním povrchem nebo neostrým lomem.

Protože se u zvolených implementací metod distributivního ray tracingu často využívají náhodně generovaná čísla z prostoru $\langle 0;1 \rangle^2$, jsou pro optimalizaci předem vygenerovány v požadovaném počtu do statického pole při spuštění aplikace. Vzhledem k tomu, že jejich použití souvisí s vlastnostmi povrchů objektů, uchovávají se tato pole u jednotlivých materiálů.

Plošné zdroje světla

K vytvoření polostínů se ve scéně aplikace využívají homogenní plošné zdroje světla. Při jejich výpočtu jsem vycházel z materiálu Josefa Pelikána, Distribuované sledování paprsku[27]. Použitý princip je však naprosto intuitivní a proto k němu nebudu vypisovat příslušné vzorce. Vliv osvětlení homogenního plošného zdroje na určitý bod je určen obsahem jeho viditelné plochy z daného bodu vůči celkové ploše zdroje a po integraci se počítá jako poměr počtu jeho viditelných vzorků vůči jejich celkovému počtu.

V aplikaci byla k vzorkování plochy použita pravidelná mřížka a k odstranění prostorového aliasingu jsou vzorky v buňkách rozřeseny pomocí uložených náhodných čísel v materiálu příslušného světla. Počet vzorků by měl záviset na plošné velikosti zdroje, a je proto vhodné stanovit si obsah plochy na jeden vzorek a podle ní pak zdroje rozdělit. Je třeba však brát ohled na to, že s každým vzorkem se musí ve scéně hledat intersekcce s objekty a také že se zvýšením počtu vzorků dochází k lineárnímu nárůstu časové složitosti.

Neostrý lom a odraz

Neostrý paprsek aplikovaný u kapaliny rozmazává její lomené a odražené prostředí a spolu s absorpcí světla vytváří vizuální dojem zakalené látky. U reflexivního či transparentního povrchu se neostrého obrazu dosahuje distribucí několika paprsků kolem ideálního směru odrazu či lomu s následným zprůměrováním získaných hodnot. Jeho zavedení tedy vytváří novou vlastnost povrchu, velikost rozostření, která určuje, jak moc se paprsky mohou vzdálit od směru ostrého odrazu nebo lomu.

K určení směru neostrého paprsku se v aplikaci používá metoda Monte Carlo, popsaná v učebních materiálech Fabia Pellacini pro předmět počítačové grafiky[28]. Výpočet směru je, pro základní představu principu metody, zapsaný v rovnici 4.1. Z ní je patrné, jak metoda upravuje směr paprsku posunem po na něj kolmé ploše.

$$R_i = \frac{R + (0,5 - r_{i,1})l\vec{u} + (0,5 - r_{i,2})l\vec{v}}{|R_i|} \quad (4.1)$$

R vektor směru ideálního lomu nebo odrazu paprsku

R_i normalizovaný vektor směru pro neostrý paprsek i

l velikost rozostření

\vec{u} , \vec{v} jednotkové vektory ortogonální s vektorem R

$r_{i,1}$, $r_{i,2}$ náhodné vzorky s rovnoměrným rozložením z prostoru $\langle 0;1 \rangle^2$
pro paprsek i

Distributivní metodou pro odraz a lom lze velice výhodně rozšířit zabezpečení proti nekonečným totálním odrazům. Neplatné paprsky je v ní možné zahazovat již při prvním výskytu a nemusí se proto ani dostat do metody supersamplingu. Pro urychlení výpočtu lze také využít předpokladu, že pokud není žádný paprsek z první poloviny vzorků validní, je pravděpodobné, že nebudou v pořádku ani ty zbylé a nemá smysl je tedy dále počítat. K zamezení chyby způsobené nepříznivým uspořádáním náhodných vzorků je ale vhodnější testovat z nich například tři čtvrtiny a jednu čtvrtinu zahazovat.

Zavedením distributivního ray tracingu do výpočtu odrazu a lomu vede, se zvyšováním počtu neostrých paprsků, k polynomiálnímu růstu časové složitosti podle použité hodnoty maximální hloubky rekurze. Například při použití 8 paprsků pro neostrý odraz a 6-ti násobné hloubce rekurze by mohlo dojít k vytvoření až 262144 paprsků na jeden primární paprsek. Při nerozumné kombinaci počtu neostrých paprsků a hloubky rekurze proto může velice snadno dojít k neúnosně dlouhému času výpočtu snímku.

4.3 Zpracování na více počítačích

Účelem vytvořené aplikace je, aby umožňovala v průběhu simulace kapaliny vytvářet snímky ve vysoké kvalitě, ze kterých by bylo možné následně sestavit video. Kvůli nízkým optimalizacím však renderování snímku může trvat řádově v desítkách minut. K vytvoření dostatečného počtu snímků pro video v rozumném čase byla proto aplikace přizpůsobena ke spuštění na clusteru počítačů.

Spouštění aplikací na využitém počítačovém clusteru probíhá vytvářením jejich samostatných instancí na jednotlivých počítačích. Jelikož simulační krok aplikace trvá mnohonásobně kratší dobu než renderování a, na rozdíl od něho, je závislý na předchozím stavu, byla simulace provedena v celku na jednom počítači s ukládáním pozic jednotlivých částic do souborů při každém simulačním kroku. Samotné renderování snímků již pak mohlo být provedeno nezávisle na jednotlivých počítačích clusteru.

Pro spuštění na clusteru aplikace umožňuje aktivovat režim příkazového řádku, při kterém nevytváří grafická okna a po své inicializaci pouze vyrenderuje scénu do určeného souboru dle zvoleného nastavení. Volba nastavení se v tomto režimu provádí přes soubory, které lze aplikaci předat přes argumenty příkazového řádku. Stejným způsobem se jí předávají i soubory s předem nasimulovanými částicemi. Spouštění jednotlivých instancí aplikace na clusteru pak probíhá aktivací módu příkazového řádku, specifikací společných souborů s nastavením a předáním souboru s částicemi příslušného kroku simulace.

Jelikož má dostupný cluster počítačů omezení v maximální době výpočtu pro jednotlivé spuštěné instance, bylo v aplikaci zavedeno zálohování současného stavu renderování do dočasného souboru, které se provádí po každém úspěšně vyrenderovaném řádku. Protože je ale limitní doba maximální délky spuštění stanovena na 3 hodiny, nemělo by teoreticky nikdy dojít k jeho využití. V případě náhlého ukončení však aplikace takto umožňuje pokračovat v renderování od uložené pozice.

5 Závěr

Prvním úkolem této práce bylo prozkoumat problematiku realistické simulace kapaliny se zaměřením na interakci kapalin. Jako nejvhodnější se ukázaly fyzikálně založené metody, které vycházejí z Navier-Stokesových rovnic. Existující metody se pak liší hlavně v přístupu k diskretizaci prostoru pro numerické řešení těchto rovnic. Jednotlivé přístupy proto byly popsány v kapitole 2.2.

Pro podrobnější popis byla vybrána realizace metody SPH, která umožňuje simulovat visko-elastické vlastnosti kapaliny a zároveň je dostupné její rozšíření pro interakci dvou kapalin. Samotná metoda byla popsána v kapitole 2.3 a kapitola 2.3.4 se věnuje tomuto rozšíření a navrhuje jeho úpravy pro simulaci více než dvou kapalin.

Dalším cílem práce bylo zaměřit se na metody vizualizace kapaliny podporujících široké spektrum světelných efektů. Pro tyto účely existují metody globálního osvětlení a metoda sledování paprsku. Jelikož metody globálního osvětlení pracují jako nadstavba jiných zobrazovacích metod, byla v kapitole 3.2.2 podrobněji popsána metoda sledování paprsku. V kapitole 4.2.2 je pak probíráno její rozšíření o absorpci světla a efekt obarvení průhledných materiálů a v kapitole 4.2.3 je část věnovaná implementaci neostrého lomu a odrazu. Největším problémem této metody s ohledem na kapaliny je však absence kaustik, pro jejichž zobrazování jsou vhodné právě metody globálního osvětlení.

Součástí této práce byla i implementace popsaných algoritmů a proto se kapitola 4 věnuje problémům, které při ní mohou nastat. Vytvořená aplikace umožňuje realistickou simulaci interakce více kapalin visko-elastických a plastických vlastností. Protože se však při implementaci nepodařilo upravit metodu Marching Cubes pro extrakci rozhraní mezi kapalinami, jsou na sobě jejich povrchy nezávislé a ve výsledku tedy není pro vizualizaci více kapalin použitelná. U jedné kapaliny však umožňuje zobrazení měkkých stínů, ostrého i neostrého odrazu a lomu a absorpci a obarvení světla průchodem přes průhledný materiál.

Jelikož hlavními nedostatky této práce je chybějící kapitola o metodách globálního osvětlení a neúspěšná implementace metody Marching Cubes pro více kapalin, mělo by se její případné pokračování zaměřit zejména na odstranění těchto problémů a na optimalizace použitých algoritmů.

Literatura

- [1] Bikker, Jacco: „Raytracing: Theory & Implementation Part 3, Refractions and Beer's Law“ [online]. 06. 10. 2005. [citováno 08. 05. 2011]. Dostupný z WWW: <http://www.devmaster.net/articles/raytracing_series/part3.php>
- [2] Bourke, Paul.: Polygonising A Scalar Field [online]. 1994 [citováno 08.01.2011]. Dostupný z WWW: <<http://local.wasp.uwa.edu.au/~pbourke/geometry/polygonise/>>
- [3] Brigham Young University, Computer Science, Computer Graphics CS455-1 Winter 2005: 3DDDA (in 2D) [online]. 30. 03. 2011 [citováno 04. 05. 2011]. Dostupný z WWW: <<http://tom.cs.byu.edu/~455/3DDDA.pdf>>
- [4] Calvert, James B.: Caustics. [online]. 12 říjen 2004 [citováno 09.01.2011]. Dostupný z WWW: <<http://mysite.du.edu/~ctuttle/math/caustic.htm>>
- [5] Clavet, Simon; Beaudoin, Philippe; Poulin, Pierre: Particle-based viscoelastic fluid simulation. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05). ACM, New York, NY, USA, 219-228, 2005, DOI=10.1145/1073368.1073400
- [6] Cutler, Barb; Durand, Fredo: MIT 6.837 - Ray Tracing [online] [citováno 07.01.2011]. Dostupný z WWW: <http://groups.csail.mit.edu/graphics/classes/6.837/F04/lectures/05_RayTracing.pdf>
- [7] Dong, Tianwen; Jiang, Shunliang: Comparisons of Mirror and Static Boundary Conditions in Incompressible Smoothed Particle Hydrodynamics. Computational and Information Sciences (ICCIS), 2010 International Conference on , vol., no., pp.1212-1215, 17-19 Dec. 2010, 2010, DOI: 10.1109/ICCIS.2010.299
- [8] Follick, Dwight: Ray Tracing with Caustics. [online]. c2011 [citováno 04. 05. 2011]. Dostupný z WWW: <<http://www.follick.ca/rt/index.php>>
- [9] Foster, Nick; Fedkiw, Ronald: Practical animation of liquids. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). ACM, New York, NY, USA, 23-30, 2001, DOI=10.1145/383259.383261
- [10] Foster, Nick; Metaxas, Dimitri: Realistic animation of liquids. In Proceedings of the conference on Graphics interface '96 (GI '96), Wayne A. Davis (Ed.). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 204-212, 1996
- [11] Foster, Nick; Metaxas, Dimitris: Modeling the motion of a hot, turbulent gas. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 181-188, 1997, DOI=10.1145/258734.258838
- [12] Guendelman, Eran; Bridson, Robert; Fedkiw, Ronald: Nonconvex rigid bodies with stacking. In ACM SIGGRAPH 2003 Papers (SIGGRAPH '03). ACM, New York, NY, USA, 871-878, 2003, DOI=10.1145/1201775.882358
- [13] Halliday, David; Resnick, Robert; Walker, Jearl: Fyzika. Brno, Praha : Vysoké učení technické v Brně - Nakladatelství VUTIUM a PROMETHEUS, 2001. ISBN 80-214-1868-0. Kapitola 34.7
- [14] Institut fyziky, VŠB Technická univerzita Ostrava: Základy fyziky – 7. Mechanika tekutin – p. 16/22 [online]. c2006 [citováno 5. 01. 2011]. Dostupný z WWW: <http://if.vsb.cz/Studium/FEI/Zakl_fyziky/ZakladyFyziky7.pdf>
- [15] Lee, Dongwoon : Physics-Based Simulations for Fluid Mixtures, 2007, Dostupný z WWW: <<http://www.cs.toronto.edu/pub/reports/na/DWLee.Research.Paper.2007.pdf>>
- [16] Lepil, Oldřich; Bednařík, Milan; Hýblová, Radmila: Fyzika pro střední školy, I. díl, Praha, Prometheus, 1999, ISBN 80-85849-87-9
- [17] Liu, Shiguang; Liu, Qiguang; Peng, Qunsheng: Realistic simulation of mixing fluids. Vis. Comput. 27, 3 (March 2011), 241-248, 2011, DOI=10.1007/s00371-010-0531-1
- [18] Lorensen, W. E.; Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. 21, 4 (August 1987), 163-169, 1987, DOI=10.1145/37402.37422
- [19] Ludvigsen, Holger; Elster, Anne: Real-Time Ray Tracing Using NVIDIA OptiX. Presented and published at EuroGraphics in Norrköping, May 3-7, 2010
- [20] Marian, Jakub: Simulace kapalin částicovým přístupem a jejich vizualizace algoritmem Marching Cubes. Praha, 2008 Dostupný z WWW: <<http://www.kubaz.cz/data/simulace.pdf>>
- [21] Mařík, Robert: Inženýrská matematika, Mendelova Univerzita v Brně, p.19, gradient [online]. c2010 [citováno 06.01.2011]. Dostupný z WWW: <<http://user.mendelu.cz/marik/inzmat/in-mat-screen.pdf>>
- [22] Monaghan, J. J.: Simulating free surface flows with SPH. J. Comput. Phys. 110, 2 (February 1994), 399-406, 1994, DOI=10.1006/jcph.1994.1034
- [23] Müller, M.; Solenthaler, B.; Keiser, R.; Gross, M.: Particle-based fluid-fluid interaction. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05). ACM, New York, NY, USA, 237-244, 2005, DOI=10.1145/1073368.1073402
- [24] Nealen, Andrew; Mueller, Matthias; Keiser, Richard; Boxerman, Eddy; Carlson, Mark. Physically Based Deformable Models in Computer Graphics. Computer Graphics Forum, Vol. 25, No. 4. (December 2006), pp. 809-836, 2006, DOI:10.1111/j.1467-8659.2006.01000.x
- [25] nVidia, Interactive ray tracing on nvidia quadro professional graphics solutions [online]. c2011 [citováno 11.01.2011]. Dostupný z WWW: <<http://www.nvidia.com/object/optix.html>>
- [26] Obr, Jakub, Fadrhonc, Pavel: 49. - Disperze světla, nové poznatky, duben 2009
- [27] Pelikán, Josef: Distribuované sledování paprsku. c1996-2001. KSVI MFF UK Praha [online]. 17. 02. 2009 [citováno 12. 05. 2011]. Dostupný z WWW: <<http://cgg.mff.cuni.cz/~pepca/lectures/old/distrt/>>

- [28] Pellacini, Fabio: distribution ray tracing. c2006. Department of Computer Science at Dartmouth College, Computer Graphics CS52 – Winter 2008 [online] 2008 [citováno 12. 05. 2011]. Dostupný z WWW: <http://www.cs.dartmouth.edu/~fabio/teaching/graphics08/lectures/16_DistributionRayTracing_Web.pdf>
- [29] Song, Bo; Dong, Lanfang: A New Boundary Treatment Method for SPH and Application in Fluid Simulation. Information and Computing (ICIC), 2010 Third International Conference on , vol.4, no., pp.82-85, 4-6 June 2010, DOI=10.1109/ICIC.2010.291
- [30] Stam, Jos: Stable fluids. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 121-128, 1999, DOI=10.1145/311535.311548
- [31] Šmejkal, Petr: Spektroskopické vlastnosti látek (Základy spektroskopie) [online]. 30.11.2006. [citováno 14.04.2011]. Dostupný z WWW: <<http://www.pierresm.cholerik.cz/Spektroskopie/Spektroskopie.ppt>>
- [32] Teschner, M.; Heidelberger, B.; Mueller, M.; Pomeranets, D.; Gross, M.: Optimized spatial hashing for collision detection of deformable objects. In Vision, Modeling, and Visualization (2003), pp. 47–54, 2003
- [33] Uppsala University, Department of Scientific Computing, Computer Graphics MN1/DV1 Fall 2000: L08 - Shading [online] [citováno 08.01.2011]. Dostupný z WWW: <<http://user.it.uu.se/~grafik/ht00/lectures/shading2.pdf>>
- [34] Vach, Marek: Transport znečištění v ovzduší – Navier-Stokesovy rovnice a jejich zdůvodnění. [online]. [citováno 05.01.2011]. Dostupný z WWW: <http://knc.czu.cz/~vachm/fluid/vychozi_asp_text.doc>
- [35] Widen, Elizabeth; Carmon, Tal: Reflection and Transmittance for Two Dielectrics [online]. 7. prosinec 2008, [citováno 09.01.2011]. Dostupný z WWW: <<http://wm.eecs.umich.edu/webMathematica/eecs434/f08/ideliz/final.jsp>>
- [36] Wolfe, Charles Thomas: Acoustic Modeling of Reverberation using Smoothed Particle Hydrodynamics. University of Colorado at Colorado Springs, 2007
- [37] Yildiz, Mehmet; Rook, Rusty; Suleman, Afzal: SPH with the multiple boundary tangent method. International Journal for Numerical Methods in Engineering , 2008, ISSN 0029-5981

Ostatní zdroje

- [38] Geiss, Ryan: NVIDIA Corporation. GPU Gems 3, Chapter 1. Generating Complex Procedural Terrains Using the GPU, Figure 1-4 The 14 Fundamental Cases for Marching Cubes [online] [citováno 09.01.2011]. Dostupný z WWW: <http://http.developer.nvidia.com/GPUGems3/gpugems3_ch01.html>
- [39] Wikipedie: Otevřená encyklopedie: Soubor:Disperze svetla.png [online]. c2007 [citováno 11.01.2011]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Soubor:Disperze_svetla.png>
- [40] Wikipedie: Otevřená encyklopedie: File:Kaustik.jpg [online]. c2006 [citováno 11.01.2011]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/File:Kaustik.jpg>>
- [41] Wikipedie: Otevřená encyklopedie: Soubor:Snelluv zakon.svg [online]. c2010 [citováno 11.01.2011]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Soubor:Snelluv_zakon.svg>

Seznam příloh

Příloha 1. Manuál k aplikaci LiquidSim

Příloha 2. Snímek uživatelského rozhraní aplikace LiquidSim

Příloha 3. Ukázkové výstupní snímky z aplikace

Příloha 4. DVD s aplikací LiquidSim

Příloha 1. - Manuál k aplikaci LiquidSim

Popis aplikace

Aplikace LiquidSim byla vytvořena za účelem demonstrace zvolených metod pro simulaci a vizualizaci kapaliny a zároveň vhodně posloužila jako platforma pro jejich testování. Implementována byla metoda Smooth Particle Hydrodynamics pro simulaci a Distributivní Ray Tracing pro vizualizaci a jejich podrobný popis lze nalézt v 2. a 3. kapitole této práce.

Tato aplikace umožňuje v přednastavené scéně vytvářet kapaliny různých vlastností a simulovat jejich chování v čase. Průběh simulace lze kdykoliv pozastavit a změnit její parametry nebo upravit vlastnosti kapaliny. Také je možné ve scéně zvolit vhodný směr pohledu pro výstup snímku ve vysoké kvalitě. Vytváření jednotlivých snímků se pak řídí buď ručně při pozastavené simulaci, nebo automaticky s každým jejím krokem. Sekvence automaticky vytvořených snímků je následně vhodná k sestavení videa, ale k tomu je již potřeba využít externí aplikaci.

Překlad

Aplikace je na přiloženém DVD pro operační systémy Windows XP a Windows 7 zkompileovaná do binární podoby pro přímé spuštění ve svém základním módu. Pro unixové operační systémy nebo pro volbu některého z módů ladění aplikace je však nutný překlad ze zdrojového kódu. Ten lze provést pomocí příkazu `make`, ke kterému je nutné zvolit vhodný soubor Makefile podle operačního systému a zvoleného módu. Aplikace byla naprogramována v jazyce C++ a ke svému překladu přes soubory Makefile vyžaduje GNU kompilátor `g++`. Přestože vývoj neprobíhal v prostředí aplikace Microsoft Visual Studio, bylo příležitostně využíváno ke krokování kódu a proto jsou na DVD dostupné i projektové soubory aplikace pro toto IDE.

Pod tímto odstavcem jsou uvedeny dostupné soubory Makefile pro překlad. K získání skutečného názvu souboru v uvedeném jméně nahraďte hodnotu `{OS}` podle použitého operačního systému za zkratku „win“, pro systémy Windows, nebo „unix“, pro unixové operační systémy. Zkratky se do názvu vkládají bez uvozovek.

<code>Makefile-<code>{OS}</code></code>	překlad bez dalších rozšíření
<code>Makefile-<code>{OS}</code>-debug</code>	překlad v režimu ladění paprsků
<code>Makefile-<code>{OS}</code>-no_ghosts</code>	překlad bez použití neviditelných hraničních částic v SPH
<code>Makefile-<code>{OS}</code>-sph_time</code>	překlad pro měření času simulačního kroku SPH (časy jsou vypisovány na standardní výstup ve formátu vhodném pro soubory typu csv)

Spuštění

Pro základní spuštění aplikace slouží zkompileovaný binární soubor `LiquidSim.exe` (pod unixem `LiquidSim`) bez udávání dalších parametrů. Aplikace využívá k zobrazování OpenGL a pro její spuštění proto musí být toto rozhraní dostupné. Pod unixovými operačními systémy musí být navíc nainstalovaná knihovna GLUT.

Přes parametry příkazové řádky lze spustit aplikaci v režimu pro výpočet na clusteru počítačů a nebo jí předat soubory s pokročilým nastavením. Také lze nechat na standardní výstup vypsat základní nápovědu aplikace. V prostředí Windows je však nutné pro její zobrazení převést standardní výstup do souboru, jelikož okenní systém vytvářený přes GLUT znemožňuje přímé vypisování do okna příkazové řádky.

Pro jednoduchou práci s aplikací byly také vytvořeny dávkové soubory/skripty pro spuštění přednastavených scén.

Parametry příkazové řádky aplikace

-h, --help

Vypíše nápovědu pro příkazový řádek.

-c, --cmdline

Spustí aplikaci v command-line módu. V tomto módu nejsou vytvářena žádná okna a aplikace pouze vyrenderuje scénu a uloží výstup do souboru. Nastavení scény a renderování v tomto módu lze ovlivnit pouze přes soubory s nastavením předané přes argumenty příkazové řádky.

-f {soubor}, --filesettings {soubor}

Nahraje do aplikace nastavení ze souboru {soubor}. Soubor nesmí ve svém názvu obsahovat mezery.

-o {nazev}, --output {nazev}

Při ručním uložení renderovaného výstupu nebo při renderování v command-line módu se pro jméno výstupního souboru použije {nazev}. Tento název bude sloužit také jako základ jména dočasného souboru pro renderování výstupu. Název se zadává bez přípony a nesmí obsahovat mezery.

Soubory s nastavením

Nastavení aplikace přes soubory pro výpočet na clusteru počítačů je velice intuitivní a jelikož pro běžnou práci s aplikací nejsou potřeba, je na DVD pro případné zájemce k dispozici okomentovaný vzorový soubor s nastavením.

Uživatelské rozhraní

Po spuštění aplikace dojde k vytvoření tří oken, přes které se aplikace ovládá. Ukázka tohoto rozhraní je na obrázku v příloze 2. a následující tři podkapitoly popisují vzhled jednotlivých oken.

Okno LiquidSim

V okně s titulkem „LiquidSim“ je k dispozici rychlý náhled na simulovanou scénu s kapalinou. Lze v něm nahlížet myší na scénu z různých úhlu a zobrazuje, mimo scény s kapalinou, žlutý jehlan zastupující bod virtuálního oka a zobrazovací plochu pro ray tracing, červenou mřížku použitého algoritmu Marching Cubes k vytvoření polygonového povrchu kapaliny a žlutou mřížku manažeru částic pro metodu SPH.

Okno LiquidSim RayTracer

Výstup ray traceru je vykreslován do okna „LiquidSim RayTracer“. To při kompilaci bez ladění paprsků slouží pouze k zobrazování. Při kompilaci s laděním paprsků je pak možné kliknutím na jednotlivé vyrenderované pixely znázornit ve scéně průběh příslušného paprsku.

Okno LiquidSimStatus

Většinu nastavení aplikace je možné měnit přes toto okno a kromě upravitelných položek obsahuje ještě řádek s aktuálním stavem aplikace. Zobrazené hodnoty s nastavením jsou zároveň rozděleny na dvě samostatné stránky a proto je na začátku každé stránky žlutým písmem vypsán její název. Aktuálně zvolená možnost s nastavením je zobrazena červeným písmem a změna její hodnoty je možná přes klávesy „+“ a „-“. Mezi jednotlivými řádky s nastavením se pak přepíná klávesami Page down a Page up a mezi stránkami klávesami Tab a Shift+Tab.

Při normální kompilaci jsou k nastavení k dispozici tyto vlastnosti:

Stránka General settings:

Ray Tracer Resolution.....rozlišení ray traceru (počet sloupců a řádků na zobrazovací ploše)

Ray Tracer viewplane size.....výška a šířka zobrazovací plochy

Supersampling subsamples.....počet vzorků superamplingu

<i>Save rendered images</i>	vypnutí/zapnutí ukládání vyrenderovaného výstupu do souboru
<i>Save particles to file</i>	vypnutí/zapnutí ukládání pozic simulovaných částic do souboru
<i>Use binary file for saving</i>	vypnutí/zapnutí binárního módu pro soubor s ukládanými částicemi
<i>Fluid GL render</i>	volba módu zobrazování kapaliny ve scéně
<i>Fluid particles render</i>	volba módu zobrazování částic ve scéně
<i>Render ghost particles</i>	vypnutí/zapnutí zobrazování neviditelných hraničních částic
<i>Fluid springs</i>	vypnutí/zapnutí výpočtu pružin u kapaliny
<i>Fluid density</i>	vypnutí/zapnutí výpočtu hustoty u kapaliny
<i>Simulation variable step size</i>	vypnutí/zapnutí variabilní velikosti kroku simulace
<i>Simulation step size</i>	velikost časového kroku simulace při vypnuté variabilní velikosti kroku
<i>Fluid spray particles</i>	počet částic vytvářených generátorem částic na scéně
<i>Fluid spray speed magnitude</i>	počáteční rychlost částic vytvářených generátorem
<i>Particles insert mode</i>	uspořádání částic při jejich ručním vkládání do scény
<i>Sphere radius</i>	průměr pomocné koule s materiálem první kapaliny

Stránka Fluid settings:

<i>Fluid</i>	volba kapaliny, pro kterou se na této stránce s natavení budou upravovat hodnoty, zvolená kapalina bude zároveň použita generátorem i při ruční vkládání
<i>Mass</i>	hmota částic kapaliny
<i>Diffusibility</i>	difuzivita kapaliny
<i>Repulsion</i>	odpudivost kapaliny
<i>Rest density</i>	klidová hustota částic
<i>Stiffness parameter</i>	parametr tuhosti (v textu o SPH označen k)
<i>Near density stiffness</i>	parametr tuhosti pro blízký tlak (v textu o SPH označen k^{near})
<i>Spring stiffness</i>	tuhost pružin (v textu o SPH označen k^{spring})
<i>Plasticity constant</i>	konstanta plasticity (v textu o SPH označena α)
<i>Linear viscosity</i>	lineární člen viskozity (v textu o SPH označen σ)
<i>Quadratic viscosity</i>	kvadratický člen viskozity (v textu o SPH označen β)
<i>Deformation yield ratio</i>	činitel poddajnosti (v textu o SPH označen γ)
<i>Refractive index</i>	index lomu materiálu kapaliny
<i>Molar absorptivity</i>	molární absorpční koeficient materiálu kapaliny
<i>Concentration</i>	molární koncentrace materiálu kapaliny

Při kompilaci s laděním se na první stránce zpřístupní ještě tato nastavení:

<i>RTDebug primary rays</i>	vypnutí/zapnutí zobrazení primárních paprsků
<i>RTDebug reflected rays</i>	vypnutí/zapnutí zobrazení odražených paprsků
<i>RTDebug transparent rays</i>	vypnutí/zapnutí zobrazení lomených paprsků
<i>RTDebug ray intersection normals</i>	vypnutí/zapnutí zobrazení normál v místě střetů paprsků
<i>RTDebug display single ray</i>	zobrazení jednoho vybraného paprsku, pokud nastaveno na YES, nebo všech paprsků, pokud nastaveno na NO
<i>RTDebug current ray id</i>	identifikační číslo vybraného primárního paprsku
<i>RTDebug current ray color</i>	výsledná barva paprsku vybraného bodu
<i>RTDebug current ray vertical</i>	číslo řádku vybraného paprsku na zobrazovací ploše
<i>RTDebug current ray horizontal</i>	číslo sloupce vybraného paprsku na zobrazovací ploše

Pohyb ve scéně

Nahlížení na scénu z různých úhlů se v okně „LiquidSim“ ovládá myší při současném stisknutí levého, pravého nebo prostředního tlačítka. Natáčení kolem scény probíhá při stisknutém levém tlačítku myši, přibližování a oddalování při pravém a posun po ploše kolmé ke směru pohledu, neboli span, při stisknutém prostředním tlačítku.

Klávesové zkratky

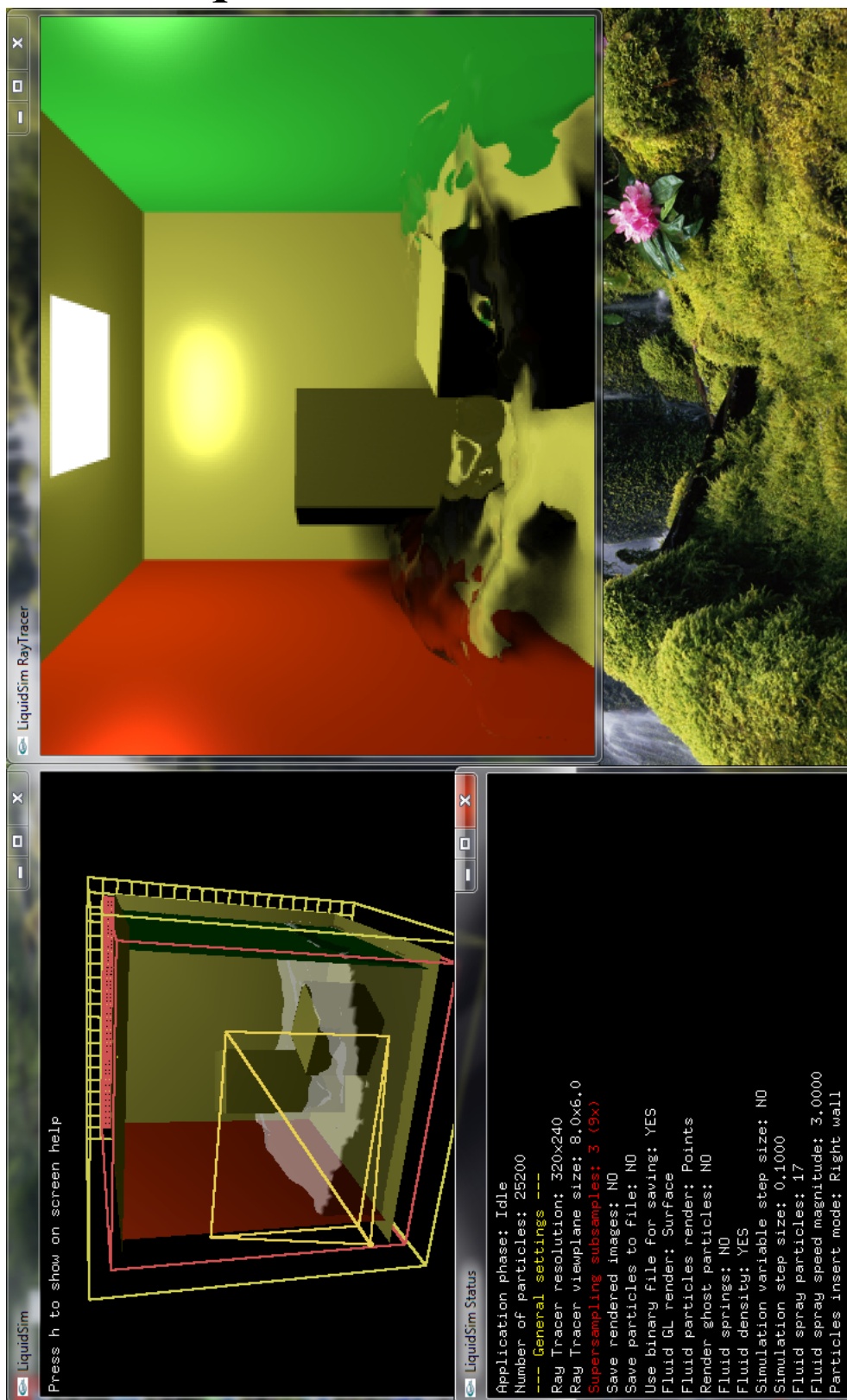
Ovládat aplikaci pomocí klávesových zkratk je možné z jakéhokoliv jejího okna stejným způsobem.

Klávesa	Funkce
<i>q</i> / <i>Esc</i>	ukončení aplikace
<i>h</i>	zobrazení OSD nápovědy aplikace
<i>Page down</i>	výběr předchozí položky v nastavení
<i>Page up</i>	výběr následující položky v nastavení
+/-	změna hodnoty vybrané položky
<i>Tab</i>	přechod na další stránku s nastavením
<i>Shift+Tab</i>	přechod na předchozí stránku s nastavením
*	vložení částic kapaliny do scény; uspořádání částic a druh kapaliny závisí na nastavení ve stavovém okně (viz podkapitola Okno LiquidSimStatus)
<i>c</i>	nastaví kameru ray traceru podle pozice pozorovatele v náhledovém okně „LiquidSim“
<i>s</i>	uloží renderovaný výstup do souboru
<i>p</i>	uloží částice kapalin do souboru <i>stored_particles_out.txt</i> (pokud soubor již existuje, dojde k jeho přepsání)
<i>shift+p</i>	uloží částice kapalin v binární podobě do souborů <i>stored_particles_bin_out.txt</i> a <i>stored_particles_bin_out_X.dat</i> , kde <i>X</i> značí pořadové číslo kapalin
<i>g</i>	spuštění/zastavení generátoru částic v simulaci
kurzorové klávesy	posun generátoru částic ve směru os <i>x</i> a <i>z</i>
<i>j</i>	zapne automatický posun generátoru částic zleva doprava a zpět během simulace
<i>m</i>	vynutí sestavení povrchu kapalin algoritmem Marching Cubes
<i>F1</i>	spuštění/zastavení simulace
<i>F2</i>	spuštění jednoho kroku simulace
<i>F3</i>	spuštění/zastavení renderování ray tracerem po každém simulačním kroku
<i>F4</i>	vyrenderování jednoho snímku ray tracerem
<i>Delete</i>	ukončí spuštěné renderování
<i>F12</i>	ukončí spuštěnou simulaci a renderování
<i>End</i>	pozastaví renderování a simulaci
<i>Home</i>	pokračuje v pozastaveném renderování a simulaci
<i>0</i> (nula)	zobrazí před scénou kouli s materiálem stejným jako má první kapalina; v nastavení ke změně průměru této koule slouží položka <i>Sphere radius</i>

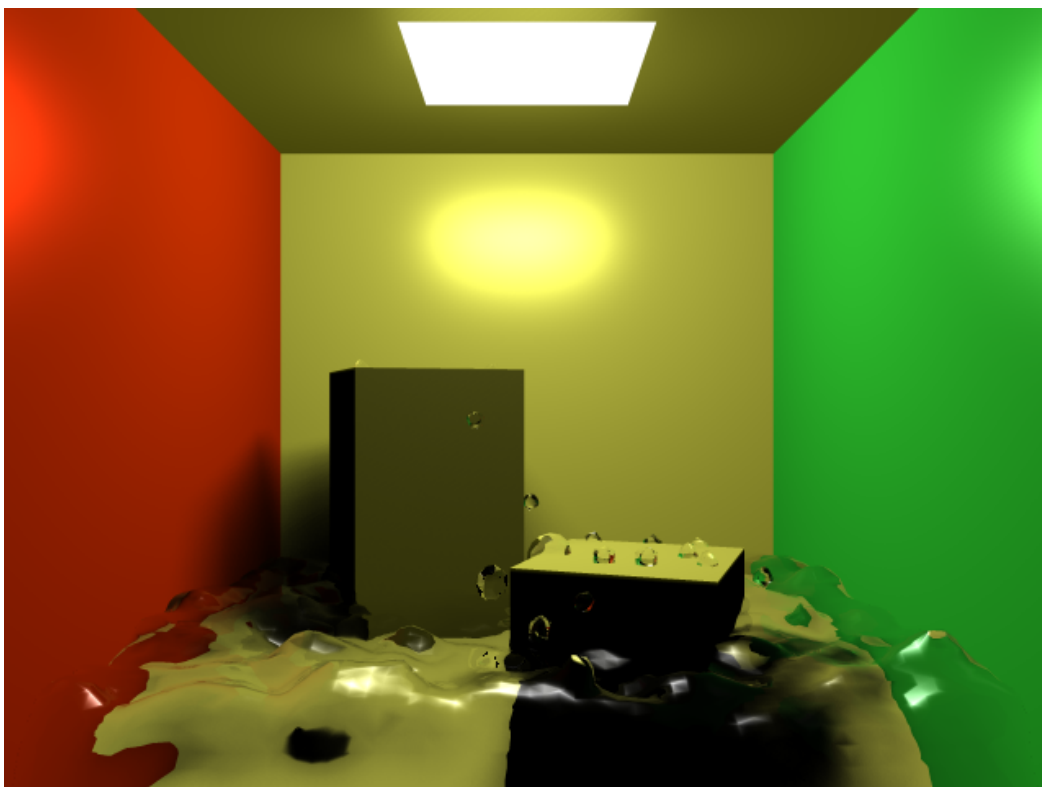
V ladícím módu jsou pak ještě dostupné následující klávesové zkratky:

Klávesa	Funkce
<i>F5</i>	zapnutí/vypnutí zobrazování paprsků
<i>F6</i>	zapnutí/vypnutí zobrazování scény (zůstanou pouze paprsky)
<i>F9</i>	zapnutí/vypnutí zobrazování obrysů trojúhelníků kapaliny
<i>F10</i>	zapnutí/vypnutí zobrazování interpolovaných normál vrcholů trojúhelníků kapalin
<i>F11</i>	zapnutí/vypnutí zobrazování skutečných normál trojúhelníků kapalin

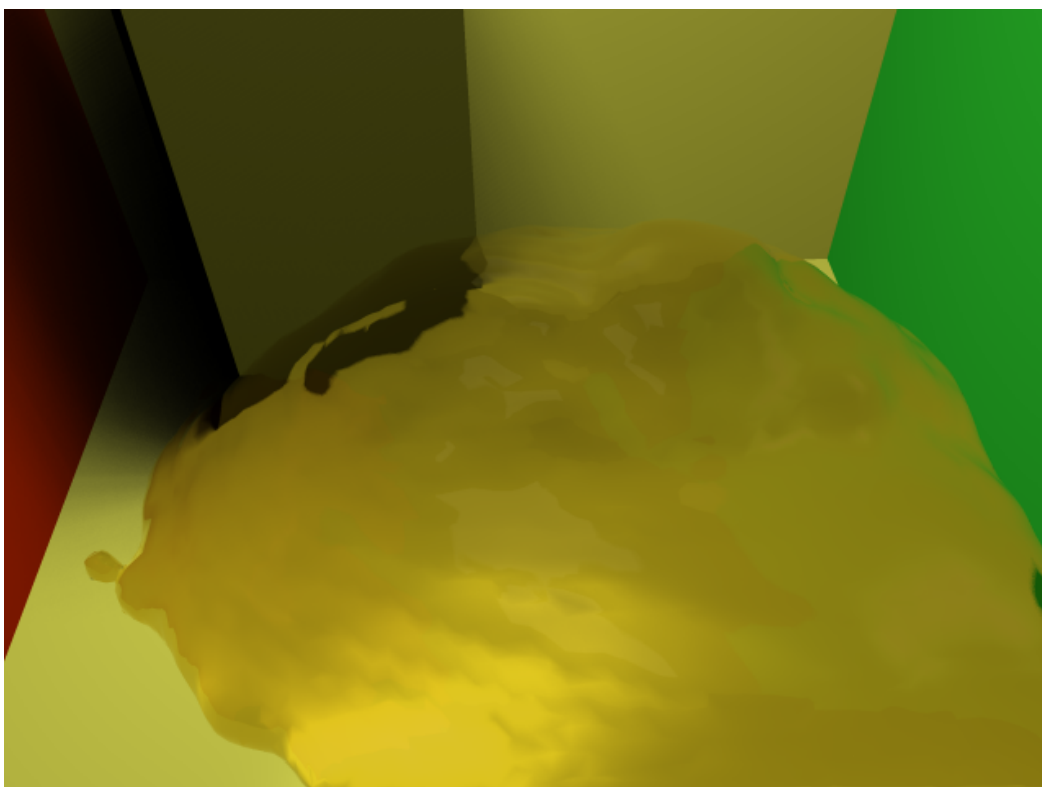
Příloha 2. - Snímek uživatelského rozhraní aplikace LiquidSim



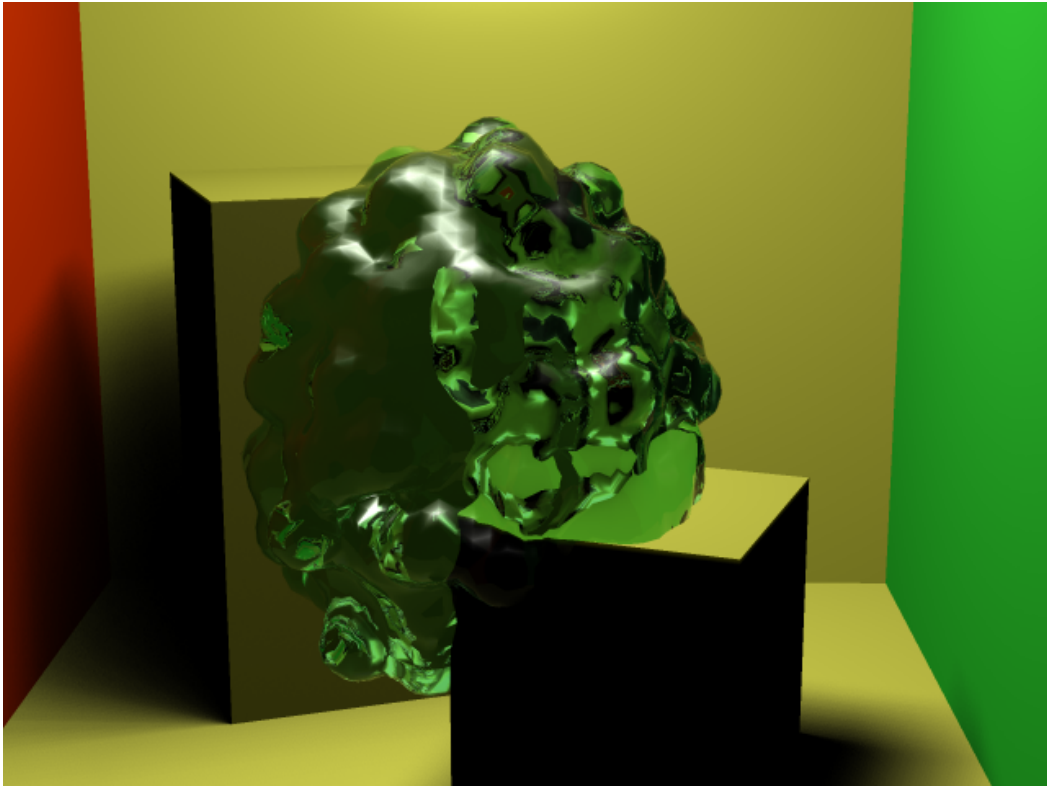
Příloha 3. - Ukázkové výstupní snímky



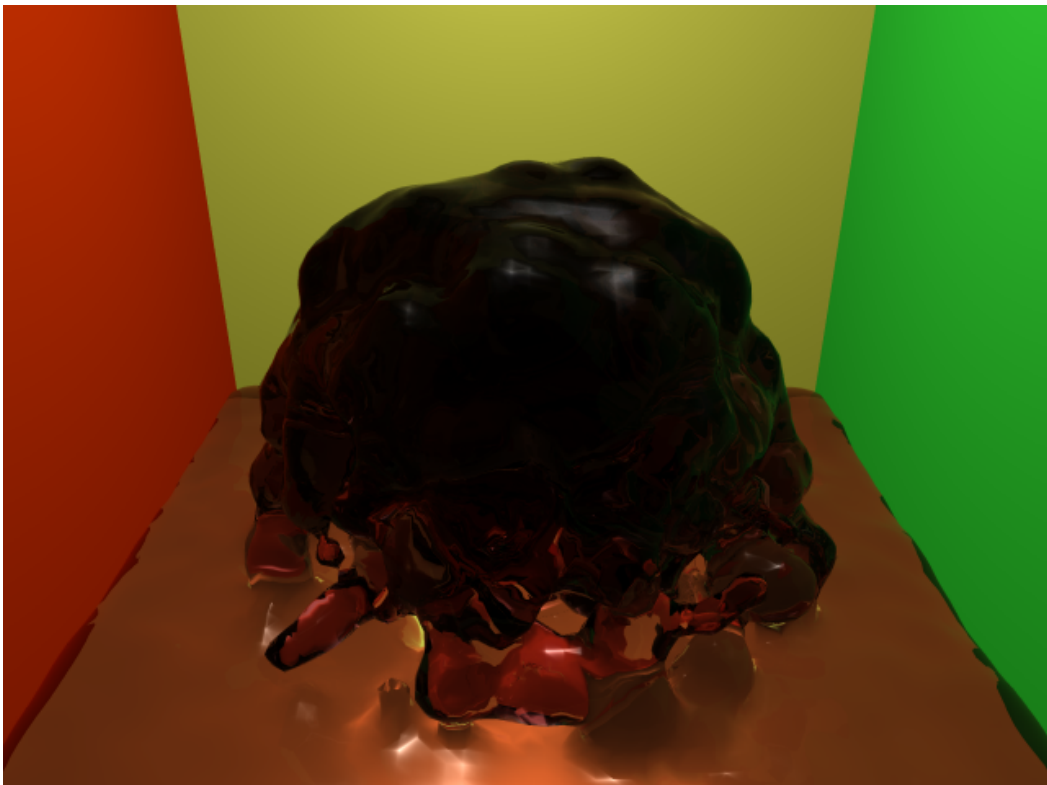
Obrázek 1: Hladina vody.



Obrázek 2: Velmi viskózní kapalina podobná medu.



Obrázek 3: Zelený blob.



Obrázek 2: Fontána z rudé kapaliny s vysokou absorpcí světla.