

Zobrazování dýmu simulovaného pomocí částicového systému

Petr Mohelník `xmohel01@stud.fit.vutbr.cz`,
Tomáš Růžička `rxruzic42@stud.fit.vutbr.cz`

29. prosince 2015

1 Úvod

Tato práce se zaměřuje na realistickou simulaci a zobrazení dýmu pomocí grafické karty. Využití by mělo být v aplikacích pracujících v reálném čase, např. hrách. V počítačových hrách může být dým využit u explozí, mlhy, ohně apod.

2 Teorie

V této kapitole je popsána teorie použitá při výpočtu a zobrazení dýmu.

2.1 Simulace

Přístupy pro simulaci kapalin a plynů se dají rozdělit do dvou kategorií. Jeden je *Eulerův přístup*, kde je prostor rozdělen na fixní 2D nebo 3D mřížku. Každá buňka v mřížce obsahuje informace o kapalině nebo plynu na dané neměnné pozici. Tyto informace mohou být tlak, hustota, teplota, viskozita aj. Oproti tomu *Lagrangeův přístup* využívá částice s proměnlivou pozicí jako nositele informace. Není vázán na fixní mřížku a může se libovolně rozpínat v prostoru. Na druhou stranu může být výpočetně náročnější kvůli nutnosti vyhledávání okolních částic. Tyto přístupy se někdy kombinují.

V tomto projektu jsme zvolili *Lagrangeovu metodu* založenou na *Smoothed Particle Hydrodynamics (SPH)*, která pro simulaci využívá částice. Řeší nestlačitelnost, symetrii sil, konzervaci hybnosti. *SPH* je interpolační metoda. Každá částice má prostorovou vzdálenost h určující, které okolní

částice na ní mají vliv. V *SPH* se fyzikální hodnota na pozici \mathbf{r} určí jako vážená suma fyzikálních hodnot ϕ_j sousedních částic j :

$$\phi(\mathbf{r}) = \sum_j m_j \frac{\phi_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \quad (1)$$

kde m_i je hmotnost částice. Hmotnost je konstantní po celou dobu simulace a shodná pro všechny částice a $W(\mathbf{r}, h)$ je symetrická vyhlazovací funkce. Hustota ρ_i se vypočítá:

$$\rho_i = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2)$$

kde \mathbf{r}_i je pozice částice i . Akcelerace \mathbf{a}_i částice i se určí následovně:

$$\mathbf{a}_i = \frac{\mathbf{f}_i}{\rho_i} \quad (3)$$

kde \mathbf{f}_i se spočítá jako $\mathbf{f}_i = \mathbf{f}_i^{viscosity} + \mathbf{f}_i^{pressure} + \mathbf{f}_i^{external}$.

Tlaková síla $\mathbf{f}_i^{pressure}$ je ze vztahu 1 určena:

$$\mathbf{f}_i^{pressure} = \sum_j m_j \frac{p_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4)$$

Bohužel tato síla není symetrická, jak může být vidět při interakci pouze dvou částic. Gradient je nula uprostřed, proto částice i využívá pouze tlak částice j pro výpočet síly a naopak. Tlak v lokacích dvou různých částic není shodný. Tlaková síla je symetrizována následovně (mohou být i jiné tvary rovnice pro symetrizaci):

$$\mathbf{f}_i^{pressure} = \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (5)$$

Tlak p_i se určí pomocí modifikované rovnice ideálního plynu:

$$p_i = k(\rho - \rho_0) \quad (6)$$

kde ρ_0 je klidová hustota.

Síla viskozity $\mathbf{f}_i^{viscosity}$ je z rovnice 1 určena:

$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{\mathbf{v}_j}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (7)$$

kde μ je viskozita kapaliny nebo plynu a \mathbf{v}_i je rychlost částice i . Tato síla je také nesymetrická. Je symetrizována následovně:

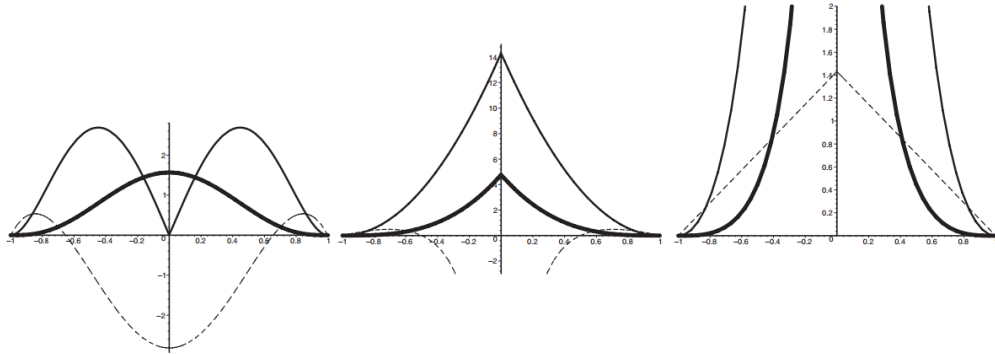
$$\mathbf{f}_i^{viscosity} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (8)$$

Pro kapalinu by bylo vhodné počítat povrchové napětí. To my pro dým nepotřebujeme. Určujeme vztlakovou sílu, která je způsobena šířením teplot. My modelujeme izotermální plyn proto ji vypočítáme jako:

$$\mathbf{f}_i^{buoyancy} = b(\rho_i - \rho_0)\mathbf{g} \quad (9)$$

kde $b > 0$ je koeficient vztlaku. Pokud bude hustota menší než klidová, částice budou tlačeny vzhůru.

Vyhlažovací funkce $W(\mathbf{r}, h)$ velmi ovlivňují rychlost, stabilitu a přesnost *SPH* metod. Používáme tři různé kernely, pro dosažení co nejlepších výsledků simulace, (obr. 1). Jeden pro výpočet hustoty, druhý pro tlakovou sílu a třetí pro sílu viskozity.



Obrázek 1: Použité kernely. Zleva doprava použity pro výpočet hustoty, tlakovou sílu a sílu viskozity. Tlusté čáry jsou kernely, tenké gradienty a šrafované Laplaciány.

Pro integraci částic v čase používáme Eulerovo schéma. Zde se prvně aktualizuje pozice rychlost \mathbf{v} :

$$\mathbf{v}_{t+\delta t} = \mathbf{v}_t + \delta t \mathbf{a}_t \quad (10)$$

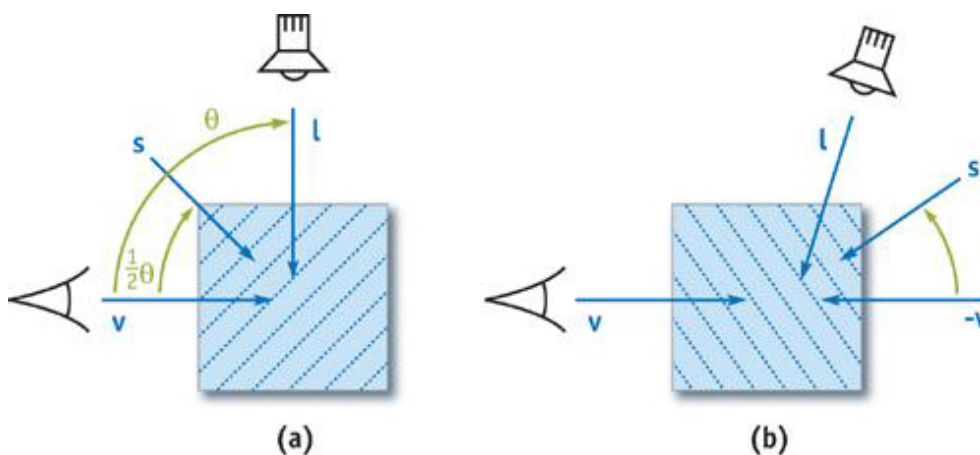
Poté se aktualizuje pozice \mathbf{r} :

$$\mathbf{r}_{t+\delta t} = \mathbf{r}_t + \delta t \mathbf{v}_{t+\delta t} \quad (11)$$

Následně se určují kolize. Při kolizi s prostředím je částice odražena směrem od překážky.

2.2 Vykreslení

Při kroku vykreslování jsme vycházeli z [Gre12], kde byla představena technika seřazení pomocí tzv. *half vektoru* (HV). Tento vektor je otočen mezi směrem pohledu kamery a pozice světla (obr. 2). Hlavní výhodou je, že k vykreslení stačí pouze jedna hloubková mapa. Dým je pak vykreslen v postupných řezech, které jsou kolmé ke směru HV . A díky tomu lze akumulovat zastínění pomocí *blendingu* do tzv. *light bufferu* (LB) a také vypočítat *hloubkovou mapu*.



Obrázek 2: Ukázka dvou případů výsledku *half vektoru*, který je umístěn mezi vektor pohledu kamery a směru světla. Převzato z [Gre12].

Nicméně při kreslení je důležité, aby byl dým kreslen správným směrem. Pokud je směr světla a pozorovatele podobný (úhel mezi nimi je menší než 90°) potom je dým kreslen zepředu dozadu. Pokud světlo leží naproti kameře, potom jsou částice kresleny zezadu dopředu.

Samotné řazení řezů pracuje na principu projekce pozice částice na HV . Tuto projekci lze jednoduše spočítat pomocí skalárního součinu.

Při stínování je pak využit výsledek *light bufferu* spolu se *stínovou mapou*.

3 Popis řešení

Pro implementaci jsme použili *OpenGL* ve verzi 4.3. K simulaci dýmu byly využity *compute shadery* spolu s *shader storage buffery*. V části stínovány zajišťovaly *framebuffery* záznam do textur.

3.1 Simulace

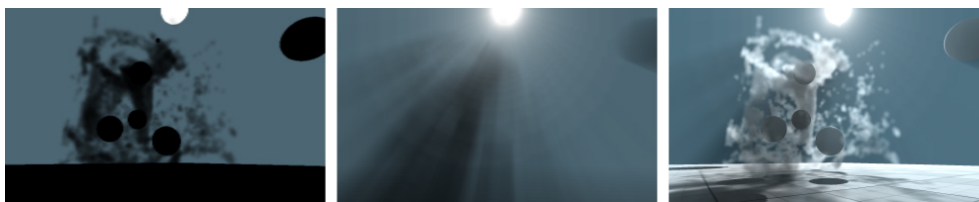
Text...

3.2 Vykreslení

Při vykreslování byla použita metoda seřazení řezů dýmu pomocí *half vektoru*, nicméně nepodařilo se nám správně nastavit režim míchání barev. Z toho důvodu byl dým po simulaci řazen dvakrát, a to pro pohled z pozice světla a z pozice kamery.

Při vykreslování z pozice kamery byly nejprve vykreslovány neprůhledné objekty scény do *hloubkové mapy*. Poté byl vykreslován dým do *light bufferu* a také do *hloubkové mapy*. *Light buffer* zde představuje klasickou texturu, do které je ukládán světelný přírůstek, který představují částice dýmu.

Dalším krokem při vykreslování je vytvoření efektu pro paprsky slunce (*sun shafts*). Nejprve je potřeba vykreslit do *framebufferu* scénu z pozice kamery, přičemž scéna je specificky obarvená. Objekt slunce má bílou barvu, pozadí tmavě modrou a ostatní objekty včetně dýmu je vykreslen s černou barvou. Pro další postup je pak vypočtena projekce pozice světla na projekční plátno. Pomocí této pozice je v dalším kroku vykreslen pouze čtverec s texturou výsledku z předchozího kroku. Při tomto kreslení je výstup rozmazán ve směru od pozice světla – tedy ve shaderu pro každý pixel ve směru ke světlu. Tento výsledek je nakonec skombinován s klasickým kreslením scény pomocí *additivního míchání* (obr. 3).



Obrázek 3: Postup vytvoření efektu paprsků slunce. Vlevo: vykreslení scény ve speciálních barvách. Uprostřed: rozmazání textury. Vpravo: Additivně smíchané s výstupem.

Při standardním vykreslování scény z pozice kamery jsou nejprve kresleny neprůhledné objekty. Pro ně jsou využity *stínové mapy* a *akumulační buffer* (*light*) z předchozích kroků. *Stínové mapy* (*hloubkové*) jsou klasicky použity s *shadow samplerem* a s projektivním dotazováním se na zastínění. Dále je využit *light buffer* stejným způsobem pro určení průsvitnosti dýmu. Tento postup je dále kombinován se *stínovou mapou* dýmu, pro oříznutí zastínění pro bližší předměty ke světlu.

Při stínování částic dýmu je také využity *stínové mapy* předmětů a dýmu stejným způsobem. Pro určení průsvitnosti v dýmu pro částice, které neleží přímo na světle, ale jsou zastíněny jinými částicemi se využívá kombinace *stínové mapy* dýmu a *light bufferu*. Tato stínová mapa je však použita jako *hloubková mapa*, pomocí které, spolu s hloubkou fragmentu je určen rozdíl hloubky. Ten je využit k výpočtu míry zastínění.

Tady stručně popište, jakým způsobem jste prakticky projekt řešili. Uveďte zejména použité technologie a algoritmy. Zaměřte se hlavně na zajímavé a důležité části implementace a také na problémy, které jste řešili. Není nutné popisovat každou třídu.

Např. uveďte, jak jste matematický popis z předchozí kapitoly implementovali prakticky.

4 Vyhodnocení

Tady by mělo být napsané jak to funguje. Protože se jedná o počítačovou grafiku nebo vidění, tak by tady měl být screenshot, ze kterého bude poznat jak to funguje. K tomu by měla být ideálně tabulka s vyhodnocením jak přesně/rychle to funguje.

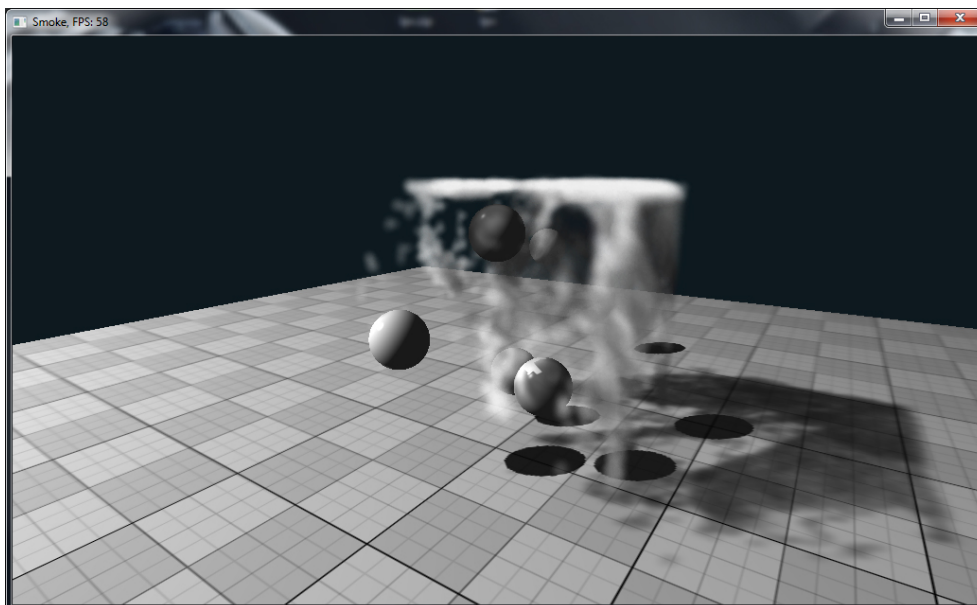
Obr. 4 5 6 7.

5 Závěr

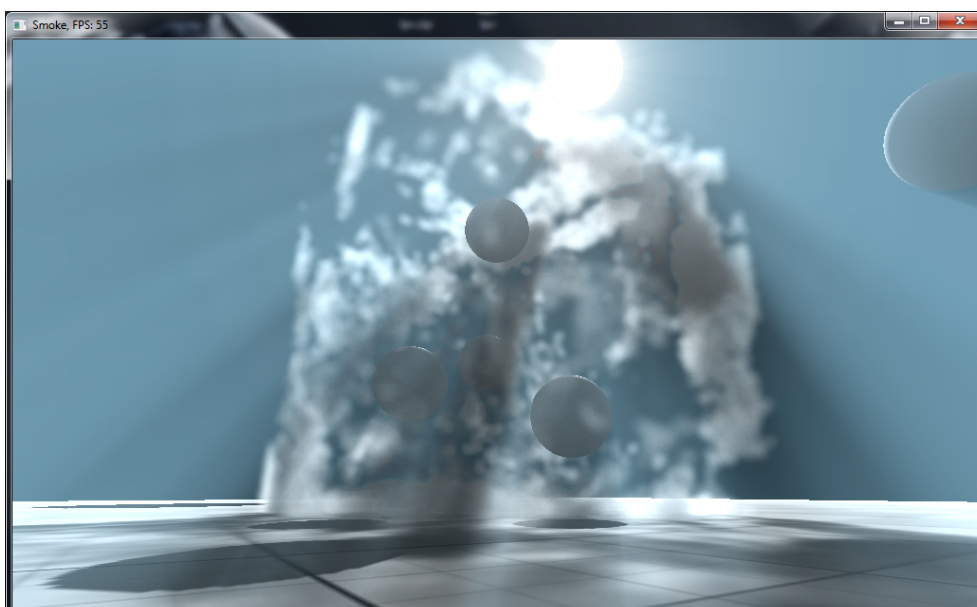
Tady by mělo být stručně napsané jak to funguje.

Reference

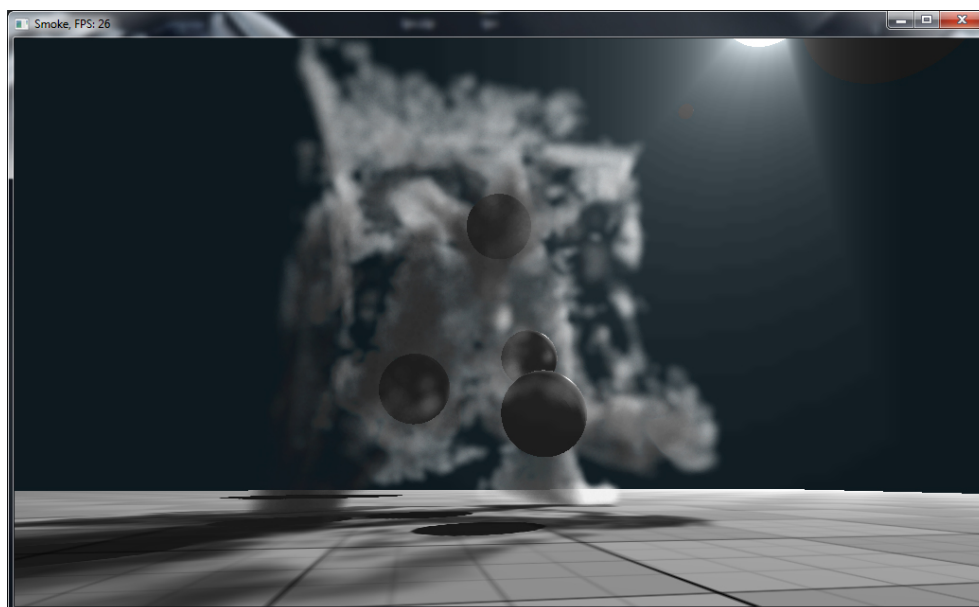
- [Gre12] Simon Green. Volumetric particle shadows. Technical report, NVIDIA, 2012.



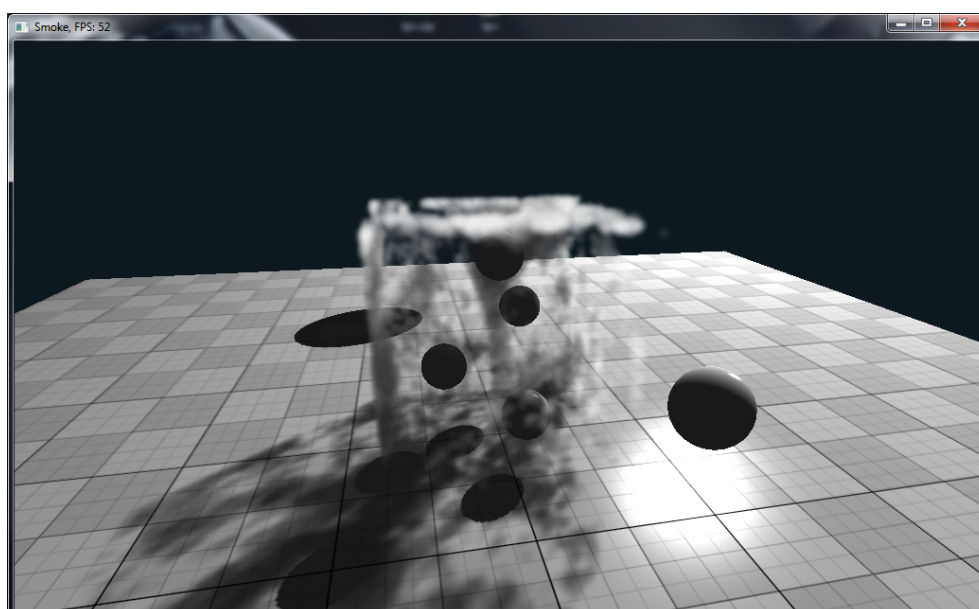
Obrázek 4: Výsledná aplikace.



Obrázek 5: Výsledná aplikace.



Obrázek 6: Výsledná aplikace.



Obrázek 7: Výsledná aplikace.