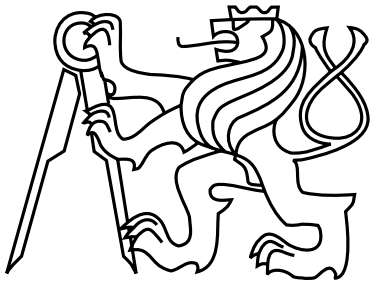
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Diplomová práce

Pohledově závislá aplikace textur v reálném čase

Bc. Daniel Princ

Vedoucí práce: Ing. David Sedláček, Ph.D.

Studijní program: Otevřená informatika, Magisterský

Obor: Softwarové inženýrství

22. dubna 2014

Poděkování

Chtěl bych poděkovat vedoucímu této práce, panu Ing. Davidovi Sedláčkovi, Ph.D., za poskytnutí cenných rad a nápadů.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.
Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, 12. 5. 2014 .....

Abstract

TBD

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

V dnešní době umožňuje počítačová grafika renderovat realistické obrazy 3D světa. Nicméně aby toto bylo možné, musí existovat 3D modely objektů, které chceme zobrazovat. Tradiční způsob, jak získat tyto modely, je jejich ruční vytváření v modelovacích programech. To je velmi pracný a zdlouhavý proces a jeho výsledek není dostatečně realistický. Proto se v současnosti vyvíjí postupy, jak přímo digitalizovat reálné objekty. Existují dva základní přístupy, které se o toto pokoušejí. Nejpřesnější metoda je pravděpodobně laserové skenování objektů. Druhou, mnohem levnější a dostupnější variantou, je rekonstrukce objektů z fotografií. Těmito problémy se zabývá počítačové vidění a částečně také počítačová grafika, do které spadá zobrazování rekonstruovaných modelů. Cílem 3D počítačové rekonstrukce je tedy co nejvěrněji převést reálné objekty do digitální podoby. Aplikaci najdeme ve virtuální realitě, online prohlídkách, počítačových animacích nebo v herním či filmovém průmyslu.

Tato práce se zabývá jednou částí 3D rekonstrukce a to texturování objektů z fotografií. Na vstupu očekáváme již zrekonstruovaný 3D model (síť trojúhelníků) a větší množství (desítky až stovky) kalibrovaných fotografií z různých úhlů. Textura společně s materiálem slouží k popisu povrchu a je důležitá pro vnímání barvy a detailní struktury povrchu [ŽBSF04]. Aplikace textury vede z výraznému zvýšení vizuální kvality objektu za relativně malou cenu. Často je vizuálně i výkonově jednodušší použít detailní texturu a jednoduchý model s menším počtem vrcholů.

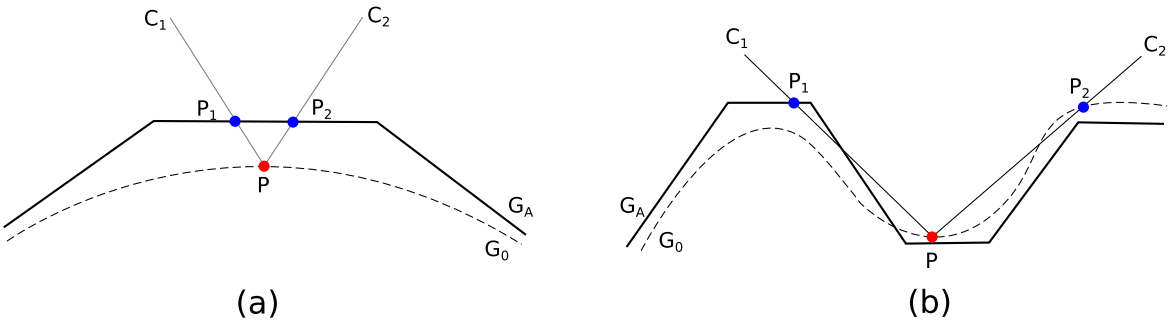
Syntéza textur z fotografií je poměrně složitá, protože na výsledný model se díváme i z jiných pohledů, než ze kterých byly pořízeny fotografie. Problémy, které při tomto procesu vznikají jsou popsány zejména v následující kapitole. Cílem této práce je vytvořit aplikaci, která v reálném čase mapuje fotografie na model. Protože aplikace funguje v reálném čase, může zohlednit aktuální pohled virtuální kamery a na základě pozice kamery vybrat nejvhodnější fotografie pro vytvoření textury. K tomu využívá velkého výkonu současných GPU.

Kapitola 2

Analýza problému a návrh řešení

2.1 Popis problému

Předpokládejme, že se díváme na statickou scénu s konstantním osvětlením, kde se pohybuje pouze pozorovatel. Za těchto podmínek můžeme popsat libovolnou fotografii pomocí pozice a orientace kamery. Pokud bychom vyfotili sférickou fotografii v každé možné pozici kamery, mohli bychom vyrenderovat kompletní scénu z libovolného pohledu. Kombinací všech 3D pozic kamery (x, y, z) a směrů (θ, ϕ) získáme plenoptickou funkci $\mathbf{L}(x, y, z, \theta, \phi)$ [AB91]. Snahou je aproximovat funkci $\mathbf{L}$ pomocí konečného množství diskrétních vzorků (x, y, z, θ, ϕ) a z této reprezentace efektivně renderovat nové pohledy (s pomocí 3D geometrie). Povrch geometrie můžeme popsat jako funkci $\mathbf{G} : (x, y, z, \theta, \phi) \rightarrow (x_0, y_0, z_0)$, tedy jako mapování pohledových paprsků na 3D souřadnice povrchu. Jako $\mathbf{G}_0$ si označíme funkci skutečného povrchu, $\mathbf{G}_A$ reprezentuje aproximovaný povrch (rekonstruovaný 3D model), viz obr. 2.1.



Obrázek 2.1: (a) chyby v geometrii způsobují chybnou projekci, bod P na původní geometrii je promítnut na body P_1 a P_2 na aproximované geometrii. (b) chyby ve viditelnosti, bod P je chybně viditelný z kamery C_2 a naopak není viditelný z kamery C_1.

Důležitým prvkem je projekční matice kamery, která nám určuje projekční mapování 3D bodu (x, y, z) do 2D souřadnic (s, t) v i-té fotografii $\mathbf{P}_i : (x, y, z) \rightarrow (s, t)$, viz sekce 2.1.1. Z promítnuté pozice v obrázku pak můžeme přiřadit 3D bodu barvu $\mathbf{I}_i : (s, t) \rightarrow (r, g, b)$.

Poté libovolný nový pohled $\mathbf{I}^V$ z virtuální kamery V můžeme vyjádřit pomocí rovnice 2.1 [EDDM+08].

$$\mathbf{I}^V(x,y,z,\theta,\phi) = \sum_i \mathbf{I}_i^V(x,y,z,\theta,\phi)\ \omega_i \tag{2.1}$$

kde

$$\mathbf{I}_i^V(x,y,z,\theta,\phi) = \mathbf{I}_i(\mathbf{P}_i(\mathbf{G}_A(x,y,z,\theta,\phi))) \tag{2.2}$$

$$\omega_i = \delta_i(\mathbf{G}_A(x,y,z,\theta,\phi))w_i(x,y,z,\theta,\phi) \tag{2.3}$$

a $\sum_i \omega_i = 1$. Notace $\mathbf{I}_i^V$ značí vyrenderovaný obrázek z pohledu V promítnutím vstupní fotografie $\mathbf{I}_i$ jako textury na povrch $\mathbf{G}_A$. δ_i určuje viditelnost a je 1, pokud je bod na povrchu $\mathbf{G}_A$ viditelný v kameře i a 0 pokud není. w_i je funkce, která určuje váhu kamery i pro každý paprsek.

Zásadní problém spočívá v tom, že $\mathbf{G}_A \neq \mathbf{G}_0$, tedy již vstupní data vztahu 2.2 jsou zatížena chybou (obr. 2.1). Problémy přináší také nepřesná kalibrace kamery $\mathbf{P}_i$, což způsobuje další nepřesnosti při mapování.

2.1.1 Matice kamery

Nejběžněji se v počítačové grafice a vidění používá perspektivní projekce, která promítá 3D body p na 2D body x vydělením jejich z souřadnicí [Sze10]. V homogeních souřadnicích má kanonická perspektivní matice $\mathbf{P}_0$ jednoduchou formu:

$$x = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}_0} p \tag{2.4}$$

Po promítnutí 3D bodu projekční kamerou je nutné transformovat souřadnice na základě vlastností senzoru a orientace kamery vzhledem k počátku souřadnicového systému. *Kalibrační matice* $\mathbf{K}$ transformuje kanonickou perspektivní kameru na standardní projekční kameru $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{K}\mathbf{P}_0 \tag{2.5}$$

kde f je ohnisková vzdálenost v pixelech a bod (c_x, c_y) je optický střed vyjádřený v pixelech. Toto je zjednodušená matice kamery $\mathbf{K}$, která uvažuje senzor kolmý vůči optické ose a stejnou ohniskovou vzdálenost v osách x a y (což je v praxi nejběžnější varianta). Orientaci kamery vůči počátku souřadnicového systému definujeme pomocí 3×3 rotační kamery $\mathbf{R}$ a vektoru $\mathbf{t}$, čímž získáme výslednou 3×4 *matici kamery*:

$$\mathbf{P} = \mathbf{K}\ [\ \mathbf{R}\ |\ \mathbf{t}\ ] \tag{2.6}$$

která provádí mapování bodu p_w v 3D světových souřadnicích do 2D souřadnic x ve fotografii:

$$x = \mathbf{P} \cdot p_w \tag{2.7}$$

2.2 Související práce

Existuje několik základních metod, jak mapovat fotografie z více pohledů na model. Jednou z možností je projekce všech fotografií a jejich následné sloučení pomocí nějakého váženého průměru, jako např. v [BMR01]. Nevýhody takové metody je zejména vznik artefaktů ve výsledné textuře, často se projevuje tzv. ghosting, kdy se v textuře objeví několik kopií jednoho obrazce. Další variantou je vytvoření atlasu textur [APK08], kdy každá část modelu dostane svojí texturu z unikátního pohledu. Tato varianta je využita např. v [AMK10]. Tento přístup má nevýhodu ve vzniku švů na okrajích jednotlivých částí textur, které je pak nutné odstraňovat [LI07]. Kromě artefaktů je u těchto přístupů častým problémem rozostření některých částí textur. Většina zmiňovaných problémů vzniká kvůli nepřesné kalibraci fotografií (špatnému odhadu radiální distorze či ohniskové vzdálenosti) nebo nepřesně rekonstruovaným modelům. Tyto nepřesnosti jsou obvyklé, i když jsou dnes již algoritmy pro 3D rekonstrukci velmi kvalitní. Získat velmi přesně kalibrované fotografie je časově náročný proces a někdy i téměř nerealizovatelný, např. ve venkovních scénách.

V článku [AMK10] jsou uvažovány nepřesně vytvořené modely, které je nutné otexturovat z původních fotografií, i když na model přesně nepasují. Navrhují podle modelu upravit původní fotografie a z nich následně vytvořit atlas textur. Úpravu provádí tak, že ve fotografiích identifikují významné prvky, které naleznou na vytvořeném modelu a zpětně je promítají do původních pohledů. Poté deformují všechny fotografie, aby co nejlépe odpovídaly nepřesným modelům. Touto metodou se nezbaví všech artefaktů, ale omezí jejich výskyt.

Podobný přístup je použitý v [TM09], kde je popsán způsob, jak dynamicky deformovat několik textur najednou podle 3D modelu a poté je pomocí vážených faktorů sloučit dohromady na základě pozice virtuální kamery.

Odlišný přístup je použitý v článku [CCCS08]. Zde řeší texturování hustých modelů s jednotkami až desítkami milionů trojúhelníků. Pro takto husté modely nepoužívají textury, ale barvu přiřazují pouze vrcholům, což vzhledem k počtu vrcholů v modelu poskytuje dostatečné detaily. Základní princip algoritmu je takový, že model je vykreslen z pohledu jednotlivých kamer a poté jsou viditelné vrcholy promítnuty zpět do původních fotografií. Pokud je jeden vrchol vidět na více fotografiích, je použita funkce, která vybere nejvhodnější barvu pro daný vrchol. Tato funkce používá vážené masky, které jsou vygenerovány pro každou fotografii a udávají kvalitu jednotlivých pixelů. Pro určení kvality pixelu je použito několik různých metrik, pro každou fotografii je tedy vytvořeno více masek (každá maska má rozlišení stejné jako fotografie). Použité metriky jsou následující:

- Úhlová metrika - nejjednodušší metrika, která porovnává směr ke kameře s normálou plochy. Největší váha je, pokud jsou oba směry stejné.
- Hloubková metrika - váha pixelu je větší, pokud je povrch blíže ke kameře.
- Hraniční metrika - tato maska udává, jak daleko je pixel od okrajů fotografie a silulety v hloubkové mapě. Čím dále je pixel od okrajů, tím je jeho kvalita lepší.

Výsledná váha pixelu je získána vynásobením hodnot v jednotlivých maskách. Tím je zaručeno zachování lokálních minim v každé masce, což pomáhá odstraňovat pixely, které jsou v li-

bovolné masce považovány za velmi špatné. Výsledná barva pro každý vrchol se získá porovnáním masek u všech fotografií, ze kterých je daný vrchol viditelný, a následným vybráním nejlepšího pixelu.

Výhoda tohoto způsobu je, že se nejedná o výpočetně složité operace, většina výpočtů je prováděna nad fotografiemi a není závislá na složitosti modelu. Další výhodou je možnost určit kvalitu jednotlivých fotografií podle maximální či průměrné kvality masky. Tím je možné některé nevhodné fotografie automaticky eliminovat a zrychlit celý proces. Nevýhodou této metody je nutnost hustých modelů, u modelů s nižším počtem vrcholů by výsledky této metody nebyly příliš kvalitní. Další problém nastává, pokud je rozlišení fotografií vyšší než rozlišení modelu (jeden vrchol se mapuje na více pixelů ve fotografiích), to vyžaduje další zpracování dat a zvyšuje složitost problému.

Všechny tyto algoritmy mají společné to, že z původních pohledů předem vytvoří texturu či atlas textur spojením všech fotografií, přičemž se snaží minimalizovat vznik artefaktů nebo případně vzniklé artefakty odstraňovat. Metoda navržená v této práci se zásadně liší tím, že žádnou takovou texturu nevytváří, ale pro každý aktuální pohled virtuální kamery vybírá množinu fotografií z nejvhodnějších pohledů a z této množiny vybírá nejvhodnější texely. Zásadní nevýhodou tohoto přístupu je velký výpočetní výkon, který je potřeba při zobrazování modelu. Tuto nevýhodu se snažíme minimalizovat efektivním využitím GPU.

Tato myšlenka není úplně nová, na podobném principu je založena např. metoda plovoucích textur [EDDM+08]. Tento algoritmus používá adaptivní nelineární metodu, která opravuje lokální nezarovnání textur vůči 3D modelu. K tomu určuje optický tok [1] mezi promítanými fotografiemi a příslušné textury kombinuje.

Navrhovaný postup využívá kombinaci lineární interpolace a odhadu optického toku. Nejprve je provedena projekce fotografií I_i na model z původních pohledů a scéna je vyrenderována z aktuálního pohledu virtuální kamery V. Tím vzniknou dočasné textury I_i^V. Poté je na jednotlivé páry textur I_i^V aplikován odhad optického toku, čímž vzniknou pole $W_{I_i^V \to I_j^V}$. Pro více než dvě vstupní fotografie je nutné provést lineární kombinaci vytvořených polí a sloučit je do výsledné textury I^v_{Float}.

Dále je nutné vypořádat se s vlastním zastíněním částí modelu, což je velmi běžná situace. K tomu je využita jemná mapa viditelnosti, která pro každý pixel určuje, zda je z dané kamery viditelný nebo ne. Oproti tradičním postupům, které obvykle využívají pouze hodnoty 1 a 0 (je nebo není vidět), je zde použita metoda, která nastavuje hodnotu z intervalu $(0, 1)$ pixelům, které jsou blízko hranic zastínění. Tím jsou odstraněny ostré hrany podél zastíněných částí, které jsou velmi často nepřesné a snižují výslednou vizuální kvalitu.

Plovoucí textury porušují epipolární geometrii, což umožňuje texturám kompenzovat nekvalitní kalibraci kamery a nepřesně zrekonstruované 3D modely.

[1] v orig. optical flow

Literatura

[AB91] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[AMK10] Ehsan Aganj, Pascal Monasse, and Renaud Keriven. Multi-view texturing of imprecise mesh. In *Computer Vision – ACCV 2009*, volume 5995 of *Lecture Notes in Computer Science*, pages 468–476. Springer Berlin Heidelberg, 2010.

[APK08] C. Allene, J.-P. Pons, and R. Keriven. Seamless image-based texture atlases using multi-band blending. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008.

[BMR01] Fausto Bernardini, Ioana M. Martin, and Holly Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7:318–332, 2001.

[CCCS08] M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3d models. *Comput. Graph.*, 32(4):464–473, August 2008.

[EDDM+08] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Computer Graphics Forum (Proc. of Eurographics)*, 27(2):409–418, April 2008.

[LI07] V. Lempitsky and D. Ivanov. Seamless mosaicing of image-based texture maps. In *Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[Sze10] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[TM09] Takeshi TAKAI and Takashi MATSUYAMA. Harmonized texture mapping. *The Journal of The Institute of Image Information and Television Engineers*, 63(4):488–499, apr 2009.

[ŽBSF04] J. Žára, B. Beneš, J. Sochor, and P. Felkel. *Moderní počítačová grafika*. Computer Press, 2004.

Příloha A

Seznam použitých zkratek

2D Two-Dimensional

ABN Abstract Boolean Networks

ASIC Application-Specific Integrated Circuit

⋮

Příloha B

Instalační a uživatelská příručka

Příloha C

Obsah přiloženého CD