

Posudek oponenta bakalářské práce

Autor/autorka práce: **Miroslav Levora**

Název práce: **Efektivní implementace registrace 3D povrchů**

Obsah práce

Práce se zabývá efektivní implementací specifického algoritmu pro registraci trojúhelníkových sítí vyvinutého na KIV. Cílem práce bylo reimplementovat stávající algoritmus co nejefektivnějším možným způsobem, tedy minimalizovat dobu jeho vykonání. Zadání zjevně nepředpokládá jakékoliv inovace algoritmu jako takového.

Trochu mě zarazila délka abstraktu, který má pouze jednu větu a lehce přesahuje jeden řádek. Nakonec to v zásadě není nic proti ničemu, protože i tak se čtenář z abstraktu dozví, co je jádrem práce, nicméně téměř stejnou informaci poskytnete i název práce.

Kapitola 1 se věnuje základním pojmům z počítačové grafiky, které jsou dále v práci používány a popisuje problém registrace – zde by možná bylo vhodné zmínit, že u registrovaných povrchů se předpokládá nějaký překryv. U popisu odhadu křivosti by autor mohl trochu více popsat, jak se dojde k rovnici na začátku strany 12. Kapitola 2 popisuje samotný algoritmus registrace, jež má být efektivně reimplementován. V sekci 2.1 podle mého názoru chybí některé detaily, např. jakým způsobem jsou vybírány body z objektu Q , ke kterým jsou následně hledány body s nejpodobnější křivostí v objektu P , nebo kolik kandidátních transformací se vlastně vytváří. Obecně vzato má metoda několik parametrů, přičemž autor nikde nezmiňuje, jak jsou nastaveny jejich hodnoty. V sekci 2.2 autor uvádí, že pro nalezení výsledné transformace je třeba najít oblast s největší hustotou v prostoru kandidátů, nijak ale nezmiňuje, proč by zrovna kandidát v místě největší hustoty měl být tou nejlepší transformací pro výslednou registraci. Dále je zde uvedeno, že funkce K v hustotní funkci mapuje vzdálenosti na rozumné hodnoty, kde však není nijak specifikováno, co v této větě znamená slovo „rozumné“. Autor vůbec nezmiňuje, že účelem funkce je v podstatě mapovat vzdálenost na podobnost (tedy čím větší vzdálenost, tím menší hodnota K) a dále jen pokračuje výpisem požadavků na funkci K kladených, ze kterých tento fakt tak nějak vyplývá. Na začátku sekce 2.2.1 je chybná rovnice zvolené funkce pro měření vzdálenosti transformací, což je patrně důsledek chyby, která je vidět v sekci 2.2.3, kde autor předpokládá, že odstraněním odmocnin jednotlivých členů sumy ve funkci d dostaneme funkci d^2 , což není pravda. Dále pak v této sekci chybí definice operátoru $:$, což pokud se nepletu, by měl být tzv. Frobenius inner product (český název neznám). Na konci kapitoly 2 by možná bylo vhodné zmínit, že i verze vzdálenostní funkce počítající s trojúhelníky místo body, je spočítatelná v $O(1)$ čase. Pouze na základě popisu v této kapitole bych měl vcelku silné pochybnosti o tom, zda autor algoritmu skutečně rozumí. Vzhledem k tomu, že se mu ho ale podařilo úspěšně implementovat a urychlit (viz dále), tato pochybnost do značné míry padá.

Kapitoly 3 a 4 poskytují uživatelskou dokumentaci a nějaké základní informace o implementaci algoritmu – dozvíme se zde například, že autor pro reimplementaci použil jazyk C, jehož volba je dobře odůvodněna a osobně ji považuji za rozumnou, a také že autorova implementace je závislá na OS Linux a není přenositelná na OS Windows. Nejzajímavější část práce začíná kapitolou 5 a pokračuje kapitolou 6, kde autor popisuje vlastní optimalizace použité v nové implementaci. Autor se rozhodl optimalizovat čistě na úrovni kódu a přehledně ukazuje efekty různých provedených

optimalizací na vygenerovaném strojovém kódu. Dále také autor využívá vektorizaci (SIMD). Tento způsob optimalizování programu považuji za relevantní a autor zde ukázal chvályhodnou znalost nízkourovňového programování, která je v dnešní době nejspíše poměrně vzácná. Na začátku strany 33 je podle mě drobná chyba (možná překlep), jelikož je zde uvedeno, že se $3x$ násobí a $3x$ sčítá, přičemž podle předchozí rovnice se sčítá pouze $2x$.

Kapitola 7 popisuje několik datových struktur, které jsou v algoritmu registrace k různým účelům využity. Text mírně vyznívá tak, že zapojení těchto struktur je autorovou inovací (i když explicitně zmíněno to nikde není). Pokud je mi však známo, všechny struktury popsané v této kapitole byly již součástí původní implementace a tudíž samotného algoritmu. Jejich popis by mi tedy dával větší smysl někde poblíž kapitoly 2 nebo přímo v ní. Kapitola 8 ukazuje výsledky nové implementace. Autor zde vůbec neukazuje výsledky samotné registrace (jen v kapitole 9 je pak zmíněno, že jsou stejné, jako u původní implementace), pouze uvádí porovnání časů původní a nové implementace. Nikde však není uvedeno, na jakém stroji, s jakou konfigurací bylo měření prováděno. Výsledky nicméně ukazují, že nová implementace je v průměru zhruba $3x$ rychlejší než ta původní, což je úctyhodné zlepšení. Trochu mě zde zmátla věta „Diverzita výsledků je způsobena jinou strategií získávání kandidátních transformací...“, přičemž si vůbec nejsem jist, o jaké diverzitě zde autor hovoří a taktéž si nejsem vědom, že by někde v textu bylo zmíněno, že nová implementace používá jinou strategii získávání kandidátních transformací. Trochu mě také zarazí, že celá kapitola s výsledky má pouze necelé dvě strany. U práce, jejímž cílem je implementovat efektivnější variantu již existující metody, bych čekal alespoň trochu podrobnější porovnání nové a původní implementace. Kapitola 9 pak stručně shrnuje celou práci.

Kvalita řešení a dosažených výsledků

Jak již bylo zmíněno, autor přepsal celý algoritmus do jazyka C a s pomocí nízkourovňových optimalizací se mu podařilo jeho vykonání zhruba trojnásobně urychlit, což považuji za více než slušný výsledek. Zdrojové kódy se mi podařilo bez problému přeložit na Linuxu a aplikaci spustit. Trochu mě ale mrzí, že aplikace pouze vypíše výslednou transformaci v podobě matice, a to je její jediný výstup. Pro ověření výsledků by bylo vhodné, kdyby aplikace také uložila do souboru výsledky spojení síti po registraci nebo alespoň transformovaný objekt Q. Taktéž by aplikace mohla vypsat čas běhu.

Aplikaci se mi po drobných úpravách podařilo přeložit i na OS Windows jako C++ projekt ve Visual Studiu. Bohužel výsledný program padal z důvodů, které Visual Studio indikovalo jako souběh vláken. Toto však v žádném případě není výtka, jelikož autor jasně deklaroval, že aplikace vyžaduje Linux, na kterém očividně funguje bez problému, a chyba klidně může být ve Windowsovské emulaci knihovny pthread, kterou jsem použil. Naopak považuji za plus, že zprovoznění aplikace na Windows by zjevně s trochou snahy nebylo až tak problematické.

Formální úroveň

Formální úroveň bohužel patří spíše mezi slabší stránky práce. Text obsahuje několik obrázků, na které není nikde z textu odkaz, např. Obr. 1.1, 1.2, 1.3, 7.7, 7.8, 7.9, 7.10 či 7.11, a na straně 16 se vyskytuje slovní spojení „viz. obrázek“, ale není uvedeno který (nemluvě o přebývajícím teče za viz).

Autor v těchto případech nejspíše předpokládá, že si čtenář spojí obrázek s textem, ke kterému se vztahuje, což je sice předpoklad vcelku oprávněný, ale po formální stránce to není správný přístup. Místy text obsahuje překlepy či drobné pravopisné chyby, ale nejedná se o zásadní problém. V kapitole 8 se objevují podsekce s číslem sekce 0 (8.0.1, 8.0.2), což je způsobeno chybějící sekci v kapitole 8, přičemž tyto dvě podsekce zjevně neměly být podsekcemi ale sekcemi (8.1, 8.2). Autor také místy nezvolil úplně vhodnou matematickou notaci – např. na straně 15 je nejdříve malé t translačním vektorem a o kousek níže na stejné straně je transformací, přičemž transformace byla ještě o kus výše popsána velkým T . Obecně také bývá zvykem rozlišovat notačně skaláry a vektory, což autor nedělá a snižuje tak čitelnost některých rovnic.

Práce s literaturou

Text obsahuje celkem 8 zdrojů, které vzhledem k implementačnímu charakteru práce považuji za dostatečné.

Splnění zadání

Zadání práce považuji za splněné.

Celkové hodnocení

Z implementačního hlediska se zjevně jedná o zdařilou práci a autor prokázal dobré znalosti nízkourovňového programování, které vhodně využil a podařilo se mu předmětnou registrační metodu značně urychlit. Z těchto důvodů práci jednoznačně **doporučuji** k obhajobě. Vzhledem k výše uvedeným připomínkám k obsahové a formální stránce textu práce a prezentaci výsledků navrhuji hodnocení známkou **dobře**.

K práci mám dále následující dotazy:

1. Na jakém stroji (s jakými výkonnostními parametry) jste prováděl porovnání časů běhu obou implementací?
2. Můžete vysvětlit větu „Diverzita výsledků je způsobena jinou strategií získávání kandidátních transformací...“ uvedenou v kapitole 8? O jaké diverzitě výsledků se zde hovoří a jakou jinou strategii získávání kandidátů máte na mysli?
3. Je do budoucna v plánu přenést Vaší implementaci i na OS Windows a domníváte se, že by tento přenos bylo možné provést s rozumně malým množstvím práce?