

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



František Psoška

### **Syntéza animované malby**

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán

Studijní program: informatika

2006

Predovšetkým ďakujem RNDr. Josefovi Pelikánovi za vedenie bakalárskej práce i ročníkového projektu a ústretovosť, Jánovi Klímovy za pomoc s 3D Studiom MAX a Katke Jakubovíe, ktorá vie lepšie po Slovensky ako ja. Ďalej chcem poďakovať vývojarom Open Source projektov za inšpiráciu a za to, že som mohol použiť ich kód. Konečne ďakujem všetkým zamestnancom fakulty, rodičom, spolužiakom a priateľom, bez ktorých by to nikdy nebolo možné.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a s jej zverejňovaním.

V Prahe dňa 11.8.2006

František Psotka

# Obsah

<b>1</b>	<b>Úvod</b>	<b>6</b>
1.1	Stručný úvod do NPR . . . . .	6
1.2	Ciele a kritériá úspechu NPR . . . . .	7
1.3	Aplikácie NPR . . . . .	8
<b>2</b>	<b>Animovaná malba</b>	<b>9</b>
2.1	Špecifiká . . . . .	9
2.2	Riešenie shower-door effectu . . . . .	10
2.2.1	Barbara Meier . . . . .	10
2.2.2	Optical Flow . . . . .	11
2.2.3	Metódy bez ťahov štetca . . . . .	11
2.3	Komplexnejší pohľad na problematiku . . . . .	11
2.3.1	Pokoj Svätý - motivácia a cieľ . . . . .	11
2.4	Animovaná malba v praxi . . . . .	12
<b>3</b>	<b>Pokoj Svätý - Implementácia</b>	<b>13</b>
3.1	Finálna aplikácia . . . . .	13
3.2	Platforma . . . . .	14
3.3	Dizajn . . . . .	14
3.3.1	Celkový prehľad . . . . .	14
3.3.2	Triedy pluginov . . . . .	15
3.4	Modul FAX . . . . .	16
3.5	Modul GUI . . . . .	18
3.6	Pluginy . . . . .	18
3.6.1	Polluter3ds . . . . .	18
3.6.2	Filler3ds . . . . .	18
3.6.3	EasyBrush . . . . .	18
3.6.4	Blur . . . . .	19

<b>4</b>	<b>Užívateľská príručka</b>	<b>20</b>
4.1	Prehľad . . . . .	20
4.2	Inštalácia a konfigurácia . . . . .	20
4.3	Príprava dát a spúšťanie . . . . .	21
4.4	Grafické užívateľské rozhranie . . . . .	22
4.5	Prvá animácia - krok za krokom . . . . .	25
<b>5</b>	<b>Záver</b>	<b>28</b>
5.1	Pokoj Svätý ako experimentálny framework . . . . .	29
5.1.1	Zhrnutie dosiahnutých výsledkov . . . . .	29
5.1.2	Smer ďalšej práce . . . . .	30
5.2	Pokoj Svätý ako užívateľská aplikácia . . . . .	31
5.2.1	Zhrnutie dosiahnutých výsledkov . . . . .	31
5.2.2	Smer ďalšej práce . . . . .	32
	<b>Literatura</b>	<b>33</b>
	<b>A Obsah disku CD-ROM</b>	<b>34</b>
	<b>B Farebná príloha</b>	<b>35</b>

Název práce: Syntéza animovanej maľby  
Autor: František Psotka  
Katedra (ústav): Kabinet software a výuky informatiky  
Vedoucí bakalářské práce: RNDr. Josef Pelikán  
e-mail vedoucího: Josef.Pelikan@mff.cuni.cz

Abstrakt: V predloženej práci študujem úlohu syntézy animovanej maľby pomocou simulácie ťahov štetca. Animovaná maľba, ako nová výtvarná forma, dáva animátorovi veľký rozmer slobody. Cieľom projektu je navrhnúť framework, ktorý by dokázal obsiahnuť túto komplexnú úlohu, spojiť flexibilitu a užívateľský komfort. Tento framework by mal slúžiť tiež na ďalší prieskum výtvarných i technických možností animovanej maľby. V rámci neho implementujem metódu Barbary Meier [4], pričom výsledkom je užívateľská aplikácia, s pomocou ktorej aj neskúsený animátor dokáže vytvoriť esteticky príťažlivé práce v maliarskom štýle. Implementáciu je nutné previesť efektívne a vysporiadať sa so vznikom artefaktov.

Klíčová slova: nefotorealistické zobrazovanie, animácia, OpenGL

Title: Painterly animation synthesis  
Author: František Psotka  
Department: Department of software and computer science education  
Supervisor: RNDr. Josef Pelikán  
Supervisor's e-mail address: Josef.Pelikan@mff.cuni.cz

Abstract: In this work I study painterly animation synthesis using simulated brush strokes. Painterly animation, as a new artistic form, gives the animator new space and freedom. The aim of the project is to design a framework which could solve this complex task - to put together both the flexibility and simplicity with user comfort. This framework should be helpful in further research of artistic and technical potential of painterly animation. Implementation the method of Barbara Meier [4] within this framework results in the application which provides a possibility of creating attractive pieces in brushwork even for less experienced animators.

Keywords: non-photorealistic rendering, animation, OpenGL

# Kapitola 1

## Úvod

### 1.1 Stručný úvod do NPR

Aký je účel obrazu<sup>1</sup>? Obraz spravidla vytvorí jeden človek s cieľom, aby ho iný človek videl. Týmto obrazom mu chce niečo oznámiť, chce prostredníctvom obrazu komunikovať. Komunikovať znamená odovzdať nejakú informáciu. (napr. „Pozri ako pekne bolo na dovolenke.“ alebo „Súčiastka má takýto tvar.“) Pojem informácie v tomto kontexte nemusí zahrňovať len myšlienku, ale aj náladu a celkom mimoverbálne pocity. Hovorí sa, že „lepšie je raz vidieť ako tisíckrát počuť“. Prvotným cieľom obrazu je teda komunikácia a obraz dokáže v mnohých prípadoch zachytiť nepomerne väčšie množstvo informácie ako text.

Nie všetka informácia musí byť pre ten-konkrétny cieľ relevantná. Príklad si požičiam z [1]: „Predstavte si fotografiu plachetnice na vode, na sklonku dňa. Z takejto fotky dokáže divák vyčítať množstvo informácií: dennú dobu, počasie, smer a rýchlosť vetra, vzťahy medzi posádkou lode atď. Avšak takýto obraz by pramálo poslúžil niekomu, kto chce postaviť loď. Ten by určite preferoval výkresy alebo náčrty, kým niekomu, kto chce preniesť len ideu plachetnice, by postačil obrys lode a trojuholník reprezentujúci plachtu.“

Fotorealistické zobrazovanie (PR<sup>2</sup>), ako prejav odvekej túžby človeka podobať sa Bohu, dosiahlo v uplynulých dekádach mimoriadnych výsledkov: efekty použité vo filmoch sú na nerozoznanie od reality už aj pre

---

<sup>1</sup>Obrazom v tomto texte myslím akúkoľvek dvojrozmernú vizuálnu formu vytvorenú človekom.

<sup>2</sup>z angického „photorealistic rendering“

mnohých CG odborníkov.

Pri fotorealistickom zobrazení je zvyčajne žiadúca čo najvyššia úroveň detailu. Mnohokrát však (nekonečný) detail vystkytujúci sa v prírode, môže pôsobiť mätúco až kontraproduktívne. Umelci sa snažia zachytiť tie črty reality, ktoré nie sú (dobře) čitateľné z fotografie. Presne takýto cieľ má aj nefotorealisticke zobrazovanie (NPR<sup>3</sup>): Skôr než simulovať realitu, *abstrahovať* od nej, *preinterpretovať* ju a čo najefektívnejšie prostredníctvom obrazu *komunikovať*.

Nefotorealisticke zobrazovanie sa teda nesnaží simulovať prírodu a obracia sa k umeniu, kognitívnym vedám a psychológii: vzniká tak interdisciplinárny obor, obrovské pole pre výskum a umeleckú kreativitu. Na mieste sú otázky ako ([1]): „Je obtiažne imitovať rozhodnutia skutočného umelca? Môžu mať počítačové programy umeleckú vynaliezavosť? Môžu byť počítačové programy expresívne?“ Prchavé pojmy zachycujúce subjektívne kategórie prenechajme filozofom.

## 1.2 Ciele a kritériá úspechu NPR

V PR, kde je miera úspechu jednoznačne daná mierou aproximácie reality, úspešnosť NPR algoritmu závisí od cieľa, ktorý sledujeme. Pokúsím sa tieto ciele a kritériá zhrnúť a kategorizovať, nadviažem na [2].

**Simulácia inteligencie.** Cieľom je simulovať ľudskú zručnosť, kreativitu, invenciu či dokonca intuíciu. Testom úspešnosti môže byť varianta Turingovho testu pre NPR.

**Zmyslupnosť.** Zmyslupnosť ako miera efektivity komunikácie. Snaha čo najviac zvýšiť čitateľnosť a expresívnosť obrazu. Potlačiť redundantnú informáciu, resp. šum.<sup>4</sup>

**Nové formy umenia.** Výpočtová technika navždy zmenila tvár hudby, filmu či fotografie. Tým, že radikálne posunula hranice kreativity. Umenie reflektuje dobu a spolu technologickým pokrokom vznikajú aj nové

---

<sup>3</sup>z angického „non-photorealistic rendering“

<sup>4</sup>Ako príklad môže poslúžiť učebnica angličtiny. Aj v učebnici pre dospelých nájdeme infantilné kreslené obrázky, za čím stoja poznatky z psychológie vzdelávania.

médiá<sup>5</sup> a formy umenia. Miera úspešnosti je daná expresívno-estetickým pôsobením, čo je samozrejme vždy subjektívne.

**Výskum vzťahu jazyk-obraz-človek.** Myslenie v obrazoch je u mnohých ľudí rozvinuté viac, než myslenie pojmové. Prchavý charakter obrazu a chabé výtvarné cítenie väčšiny populácie však intenzívne zťažuje výskum v tejto oblasti psychológie a antropológie. NPR podáva pomocnú ruku.

### 1.3 Aplikácie NPR

Oblasti výskumu a aplikácie NPR možno rozdeliť do troch prelínajúcich sa kategórií:

**Simulácia médií.** Cieľom je simulovať fyziku umeleckých médií, použitých pri rôznych umeleckých technikách ako sú napr. olejomalba, akvarel, perokresba alebo litografia.

**Asistencia užívateľovi pri tvorbe obrazu.** Snaha o zabudovanie ľudských zručností a techník do expertných systémov, ktoré umožňujú aj laikom generovať obrazy zachovávajúce vzhľad a pocit ručne vyrobených umeleckých diel. Snaha automatizovať rutinné kroky výtvarnej činnosti.

**Automatická tvorba obrazu.** Automatické generovanie obrazu vzhľadom na daný komunikačný cieľ, napr. automatické generovanie technických náčrtov alebo medicínskych kresieb.

---

<sup>5</sup>Pojem „nové média“ označuje zvyčajne počítačovú animáciu, web-design, počítačové hry, vj-ing a najrôznejšie experimentálne (interaktívne aj neinteraktívne) formy, ktoré využívajú pokročilú technológiu, bez ktorej by ich nebolo možné realizovať. V Českej Republike na tomto poli pôsobí napr. agentúra CIANT (<http://www.ciant.cz>).



# Kapitola 2

## Animovaná maľba

### 2.1 Špecifiká

Hlavným špecifikom animovanej maľby je, že nemá v klasickej animácii ekvivalent. Nemožno ju s ničím porovnať. Klasické animačné techniky, využívajúce maliarsky štýl (ako ich poznáme z rozprávok), postupujú tak, že rozpohybujú len statické namaľované objekty.

Naším cieľom je *obohatiť maľbu o ďalší rozmer - čas* a vdýchnuť tak maľbe život. Budeme de-facto maľovať animáciu políčko po políčku. To je použitím klasických techník značne pracné. Navyše tak vzniká artefakt, ktorý nie je v ľudských silách odstrániť. Tento artefakt sa nazýva *shower-door effect*.

**Shower-door effect** je následkom nedeterministickej povahy maľby. Nie je jednoznačne určená poloha ani vzhlád jednotlivých ťahov štetca<sup>1</sup>. Ak by sme použili techniky simulácie maľby určené pre statické obrazy na každé políčko animácie, výsledkom by bolo nepríjemné „hmýrenie“ ťahov v rovine premietania. Akoby sme hľadeli na scénu cez náhodne sa meniace dekoratívne sklo - odtiaľ názov „shower-door effect“. Naším cieľom je zabezpečiť súdržnosť nadväzujúcich políčok<sup>2</sup>, determinizmus v zmysle: „Čím menej sa mení scéna, tým menej sa mení jej maľba, špeciálne požadujeme aby maľba statickej scény zostala nemenná.“

---

<sup>1</sup>anglicky „brush strokes“

<sup>2</sup>anglicky „frame-to-frame coherence“

## 2.2 Riešenie shower-door effectu

### 2.2.1 Barbara Meier

Vstupom algoritmu Barbary Meier [4] je 3D scéna. Scéna je modelovaná množinou trojrozmerných častíc, ktoré sú náhodne, ale uniformne rozptýlené po jej povrchu. Tieto častice po transformácii do roviny premietania slúžia ako pozície ťahov štetca. Výsledkom je, že ťahy v priestore scény „priliehajú“ na povrch a tým je zaručená súdržnosť nadväzujúcich políčok. Atribúty jednotlivých ťahov (farba, orientácia...) pochádzajú z tzv. *referenčných snímok*, v ktorých je zobrazená scéna klasickými postupmi za pomoci shaderov. Ťahy štetcom sa kreslia odzadu dopredu. Algoritmus prebieha nasledovne:

```
vytvor častice reprezentujúce geometriu

foreach( políčko animácie )
begin
    vytvor referenčné snímky použitím geometrie,
    atribútov povrchu, osvetlenia...

    transformuj častice do priestoru zorného poľa
    zotried' častice podľa vzdialenosti od pozorovateľa

    foreach( časticu, začínajúc od najvzdialenejšej )
    begin
        transformuj časticu do priestoru priemetne
        vyhodnoť atribúty ťahu štetca z referenčných snímok

        pridaj užívateľom kontrolovaný náhodný faktor

        vykresli ťah štetca

    end // foreach časticu

end // foreach políčko animácie
```

Hlavnou výhodou metódy je, že napriek jednoduchosti algoritmu a rýchlosti výpočtu, umožňuje modelovať značné množstvo (aj bizarných) maliarskych štýlov. Idea referenčných snímok sa uplatnila v mnohých

iných algoritmoch NPR. V originálnej implementácii sú ťahy štetcom modelované pomocou billboardov.

Hlavnou nevýhodou je, že algoritmus z princípu nedokáže generovať maľbu z videa. Komplexné scény si vyžadujú značné množstvo častíc, teda operačnej pamäte. To možno riešiť generovaním častíc real-time, čo je však na úkor rýchlosti.

### 2.2.2 Optical Flow

O generovanie maľby z videa sa pokúsili napríklad James Hays a Irfan Essa v [6], alebo Aaron Hertzmann a Ken Perlin v [5]. Na vyhodnotenie pohybu častíc v scéne použili tzv. optický tok, známy z digitálneho spracovania obrazu.

### 2.2.3 Metódy bez ťahov štetca

Výsledky pri generovaní statickej maľby použitím štatistických metód alebo metód umelej inteligencie mnohokrát kvalitou prevyšujú metódy založené na simulácii ťahov štetca. Je otázkou, aké alternatívne postupy sa uplatnia pri generovaní animovanej maľby.

## 2.3 Komplexnejší pohľad na problematiku

Ako som vyššie spomínal, *nikde nie je definované, ako má animovaná maľba vyzerať*. Je to len otázkou našej fantázie. Čo ak je v niektorých prípadoch shower-door effect žiadúci? Uvediem inšpiratívnejší príklad; predstavme si jednoduchú scénu, napríklad pohyblivý záber na interiér - steny miestnosti, záber z postele. Pre rozmiestenie ťahov štetca použijeme geometriu úplne inej scény, napr. (pohyblivý) záber na ženské telo. Výsledkom bude animácia nebývalej sémantickej bohatosti. Budeme vidieť a uvedomovať si priestor, v ktorom sa kamera nachádza, ale aj priestor druhej scény. Tento prirodzený nápad sa dá ďalej rozvíjať, napr. prelínačkami medzi referenčnými snímkami oboch scén možno vytvoriť zaujímavý prechod.

### 2.3.1 Pokoj Svätý - motivácia a cieľ

*Animovaná maľba otvára obrovský priestor pre kreativitu, kde jediným majákom je náš vlastný vkus a intuícia.*

Toto je motivačné východisko mojej práce. Obrovské množstvo kombinácií, ktoré dávajú umelcovi veľký rozmer slobody, musí umelec nejako kontrolovať. V ideálnom prípade by mal byť animátor schopný kontrolovať nielen štýl maľby (typ a štýl ťahov), ale život (pohyb, čas vzniku a zániku, zmena atribútov atď.) každého jedného ťahu tak, ako to dokáže maliar stojaci pri plátne. To je samozrejme nesmierne komplexná úloha.

Mojim zámerom v tejto bakalárskej práci je navrhnúť a implementovať náčrt systému, ktorý by bol *jednoduchý a ovládateľný* natoľko, aby ho mohol používať umelec, ale ktorý *zahrnie čo najviac* vyššie zmienených *stupňov slobody*. Tento systém by mal tiež predstavovať *framework pre ďalší prieskum* animovanej maľby.

V rámci tohoto systému implementujem modifikáciu algoritmu Barbary Meier.

## 2.4 Animovaná maľba v praxi

Animovaná maľba sa zatiaľ objavila snáď len vo videoklipech a na festivaloch animovaných filmov, zvyčajne spolu s článkom prezentujúcim tú-ktorú techniku. Známe komerčné filmy, v ktorých boli použité efekty maliarskych štýlov sú napr. *What Dreams May Come* alebo *Naqoyquatsi*.

Jediný dostupný (komerčný) software je plug-in do 3D Studia Max s názvom *NPR1eyes* od firmy *Infografica*. Na stránkach firmy možno stiahnuť ukážkové animácie<sup>3</sup>. Software zjavne používa prístup Barbary Meier.

O real-time implementáciu metódy sa pokúsil Daniel Sperl v [3].

---

<sup>3</sup>s ktorými je možné porovnať výsledky mojej práce

# Kapitola 3

## Pokoj Svätý - Implementácia

### 3.1 Finálna aplikácia

Program POKOJ SVÄTÝ slúži na off-line rendering animácií imitujúcich maliarsky štýl. Vstupné dáta tvorí *3D scéna* s príslušnými textúrami vo formáte *.3ds*<sup>1</sup>, čo je exportný formát *3D Studio MAX*. Väčšina 3D modelovacích nástrojov podporuje export tohto formátu na disk. Výstupom je *animácia*, resp. jej jednotlivé políčka vo formáte *.png*. Predvolený výstupný formát animácie je *avi* obsahujúci *mpeg4 layer*.

PS je vizuálny nástroj v zmysle WYSIWYG. Užívateľ má možnosť nastavovať množstvo parametrov, ktorými ovláda vzhľad maliarskeho štýlu. Má možnosť použiť aj vlastné textúry štetca. Všetky zmeny parametrov sa prejavia v reálnom čase, v okne predstavujúcom náhľad v plnej kvalite.

Z pohľadu užívateľa je PS *NPR off-line renderer*, alternatívny k štandardným PR rendererom.

Snažil som sa držať vyššie uvedeného cieľa práce. Z pohľadu vývojára je PS prvým krokom smerom k otvorenému a značne modulárnemu frameworku na výskum animovanej maľby, ale aj jej aplikácie v praxi. Viz. podkapitola *Dizajn*, ďalej.

---

<sup>1</sup>Formát *.3ds* obsahuje geometriu (mesh), pozície kamery a svetiel a kľúčovaný pohyb všetkých objektov v scéne.

## 3.2 Platforma

Cieľovou platformou je ľubovoľná pracovná stanica podporujúca *OpenGL 1.4* a *nVidia Cg*. Kód, vrátane GUI, je vystavaný nad OpenGL a nasledujúcimi free<sup>2</sup> / open-source knižnicami:

**FreeGLUT** - Free verzia *OpenGL Utility Toolkit* - nadstavby OpenGL.

**GLEW** - Nahrávanie rozšírení OpenGL.

**GLUI** - Tvorba grafického užívateľského rozhrania vystavaná nad GLUT.

**lib3ds** - Manažment scény vo formáte *.3ds*.

**FreeImage** - Vstup/výstup obrázkov. Zahrňuje *libJPEG*, *libPNG* a *zlib*.

**nv\_math** - Práca s maticami a vektormi, súčasť nVidia SDK.

Spustiteľné súbory sú preložené pre platformu *win32* kompilátorom *gcc*, použitím známeho free-software balíka *minGW*.

Kódovanie videa je riešené volaním predkompilovanej binárky balíka free kodekov *ffmpeg*.

## 3.3 Dizajn

### 3.3.1 Celkový prehľad

Systém je založený na pluginoch. Každý plugin predstavuje jednu zameniteľnú komponentu a definuje jednu z črt výstupného maliarskeho štýlu. Modul GUI využíva služby týchto pluginov, stará sa o nastavovanie parametrov, komunikáciu z užívateľom a prípravu dát. Modul FAX vytvára vyššiu vrstvu funkcií nad OpenGL a použitými knižnicami.

Dizajn programu je pripravený na ľubovoľné prepájanie komponentov. Ich zapojenie v preloženej aplikácii je však fixné<sup>3</sup> - predstavuje ľahkú modifikáciu spomínaného algoritmu Barbary Meier. *Na tejto metóde je*

---

<sup>2</sup>Detailnú licenciu ku konkrétnej knižnici možno nájsť na domovských jednotlivých projektov.

<sup>3</sup>Je tomu tak z dvoch dôvodov; prepájanie pluginov by si vyžadovalo mnoho práce na užívateľskom rozhraní a na predvedenie zaujímavých výsledkov by bolo potrebných viac netriviálnych pluginov.

*demonštrované rozdelenie maliarskeho algoritmu do vrstiev.* Každá vrstva predstavuje jednu z tried pluginov.

### 3.3.2 Triedy pluginov

Pluginy sú rozdelené do štyroch tried. Každý plugin musí implementovať metódy `QueryParams` a `SetParams`, prostredníctvom ktorých komunikuje s užívateľským rozhraním. Rozhrania, ktoré musí plugin danej triedy implementovať, nesú výstižný názov:

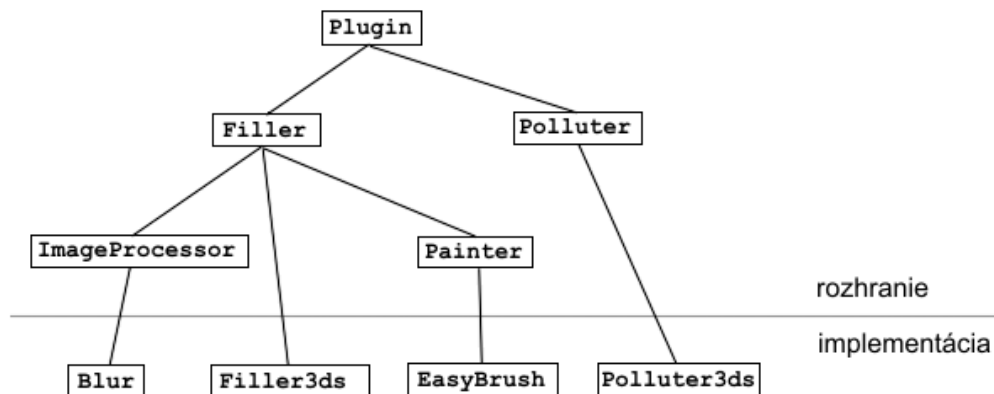
- `Filler`
- `ImageProcessor`
- `Polluter`
- `Painter`

**Filler** je najjednoduchší z typov pluginov, implementuje len metódu `Render(void)`, ktorou môže kresliť v aktuálnom OpenGL kontexte. Pluginy tohoto typu slúžia na vytváranie referenčných snímok.

**ImageProcessor** rozširuje *Filler* o metódu `BindInput(Filler*)`. Pluginy tohto typu slúžia na ďalšie spracovanie referenčných, event. výstupných snímok.

**Polluter** má na starosti rozmiestnenie ťahov štetca na plátne. Toto rozmiestnenie by malo *pôsobiť dojem náhodnosti*, ale malo by pokrývať plátno *uniformne*. To znamená, že hustota prokrytia celého plátna by mala byť rovnaká. Najdôležitejšou podmienkou je *zachovanie súdržnosti snímok v čase* vo vyššie uvedenom zmysle. Ide najmä o potlačenie shower-door efektu. Rozhranie implementuje metódy: `SetDensity(float)`, `FirstFrame(vector<StrokeOrigin*>&)` a `NextFrame(vector<StrokeOrigin*>&)`.

**Painter** má na starosti simuláciu média - vykreslenie políčka animácie. Rozširuje *Filler* o metódy `BindPolluter(Polluter*)` a `BindFillers(vector<Filler*>&)`. Vyhodnocuje atribúty ťahov štetca na základe vstupných referenčných snímok a pozícií a prevedie samotný rendering v aktuálnom OpenGL kontexte.



Obr. 3.1: Strom dedičnosti

Strom dedičnosti môžete vidieť na obrázku 3.1. Vývojový diagram metódy Barbary Meier v kontexte implementácie prehľadne zachycuje obrázok 3.2. Schému fixného zapojenia pluginov môžete vidieť na obrázku 3.3. Šípky znázorňujú volania metód cez príslušné rozhrania. Ďalšie podrobnosti nájdete v zdrojovom kóde alebo dokumentácii vygenerovanej systémom *Doxygen*.

## 3.4 Modul FAX

Modul FAX vytvára ďalšiu vrstvu abstrakcie nad OpenGL, Cg, FreeImage a lib3ds. Obsahuje nasledujúce triedy a iné často volané funkcie:

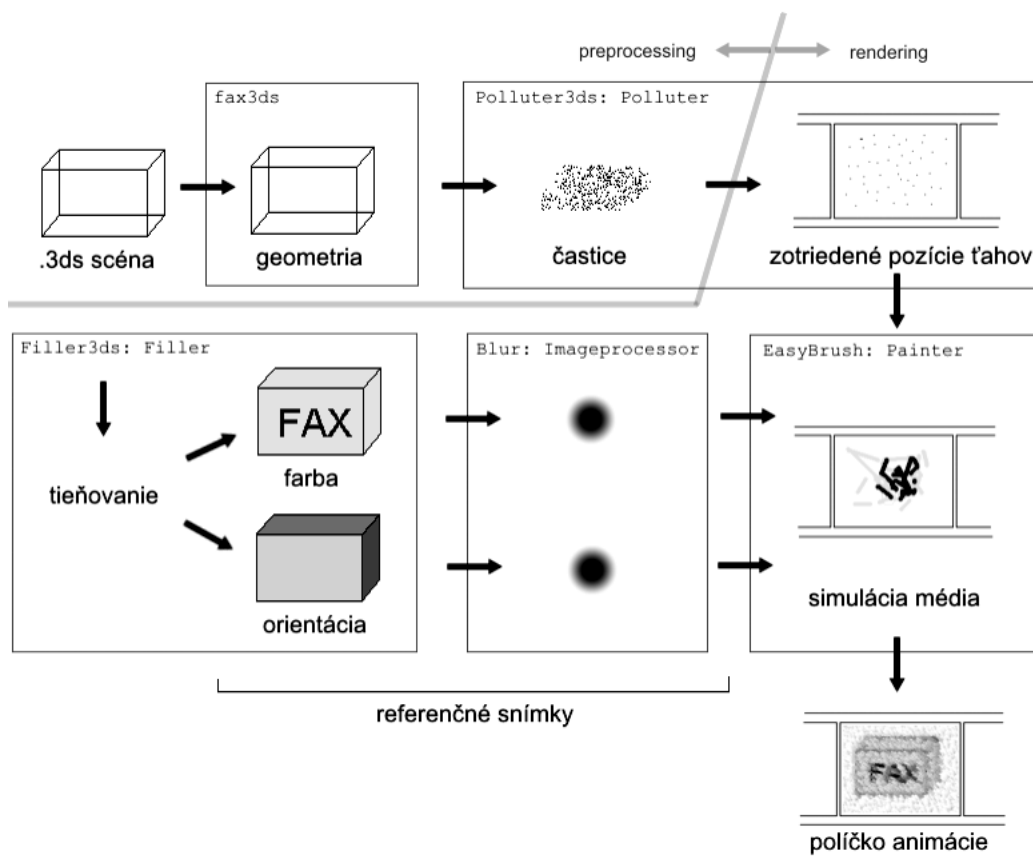
**faxTexture** - Handler pre rôzne typy OpenGL textúr, vstup a výstup bitmáp na disk.

**faxFramebuffer** - Implementácia render-to-texture rozhrania pomocou rozšírenia `EXT_framebuffer_object`. Zachovanie integrity zobrazovacieho kontextu pri viacnásobne vnorenom zobrazovaní.

**fax3ds** - Handler .3ds scény; pohyb kamery a objektov v scéne, OpenGL rendering.

**faxShader** - Handler Cg shaderov, zachovanie integrity zobrazovacieho kontextu.





Obr. 3.2: Vývojový diagram

## 3.5 Modul GUI

Implementuje front-end aplikácie pomocou knižnice GLUI. Úlohou modulu je prepojenie a ovládanie pluginov, inicializácia potrebných knižníc, príprava vstupných a výstupných dát. Modul takisto sprostredkuje komunikáciu medzi užívateľom a pluginmi, ktoré sami o sebe žiadne ovládacie rozhranie nemajú.

## 3.6 Pluginy

### 3.6.1 Polluter3ds

Úlohou pluginu je zachovať tri podmienky rozmiesnenia ťahov: *náhodnosť*, *uniformitu* a *súdržnosť*. Súdržnosť plynie z použitej metódy Barbary Meier automaticky. Pri inicializácii sa každý polygón<sup>4</sup> scény pokryje časticami. Pri vykresľovaní sa použije len toľko častíc, koľko je (pri danej hustote) potrebných na pokrytie polygónu v priemetni.

Náhodnosť a uniformita plynú zo spôsobu, akým sa na ňom rozmiestňujú častice. (Pri triviálnom náhodnom pokrytí majú tendenciu vznikať diery.) Na povrch polygónu sa umiestni vždy len jedna častica, potom sa polygón rozdelí na dva, podľa najkratšej ťažnice, a postup sa rekurzívne zopakuje. Týmto spôsobom sa vytvoria akési „poschodia“, takže neskôr môžeme hustotu pokrytia plynule regulovať.

Elegantne je potlačený artefakt pribúdania a miznutia (preblikávania) ťahov štetca — deje sa tak vždy v spodných vrstvách maľby, takže užívateľ spozoruje artefakty len na ostrých hraniciach objektov.

### 3.6.2 Filler3ds

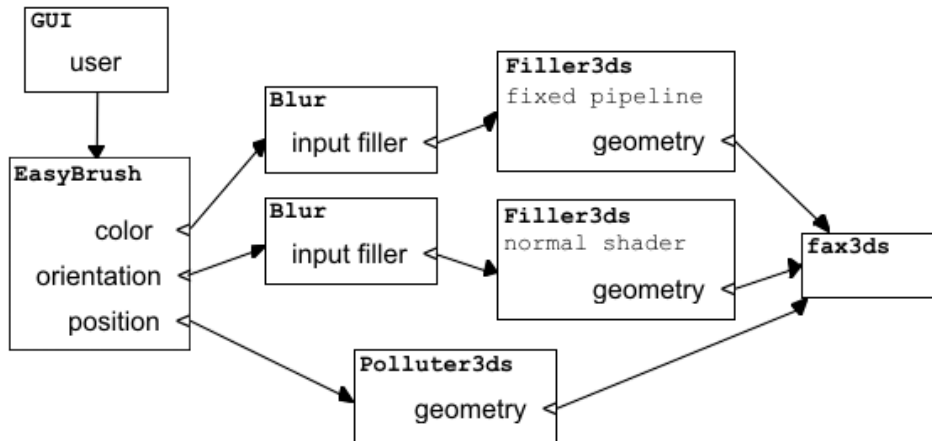
Jednoducho vykreslí geometriu v danom čase pomocou Cg shadera alebo fixed pipeline.

### 3.6.3 EasyBrush

Modeluje ťahy štetca polopriesvitnými billboardami. Užívateľ má možnosť kontrolovať maliarsky štýl množstvom parametrov. Vid. kapitola *Užívateľská príručka*. Metóda referenčných snímkov vytvára nepríjemný

---

<sup>4</sup>spravidla trojuholník



Obr. 3.3: Konfigurácia: Barbara Meier

artefakt, ktorého vznik sa nedá principiálne potlačiť. Je ním šum atribútov jednotlivých ťahov, spôsobený konečnou presnosťou (rozmermi) referenčných snímok. Tento problém riešim variantou motion-bluru. Vytváram tak síce iný artefakt, ten však nie je taký rušivý, laické oko ho prehliadne.

Ďalší artefakt je pribúdanie a miznutie (preblikávanie) ťahov štetca. Plugin tento problém nerieši, možno ho riešiť napr. plynulým prechodom cez alfa kanál.

### 3.6.4 Blur

Je jednoduchou implementáciou priemerujúcej 2D konvolúcie s ľubovoľnou veľkosťou jadra na GPU.

# Kapitola 4

## Užívateľská príručka

### 4.1 Prehľad

Program POKOJ SVÄTÝ slúži na off-line rendering animácií imitujúcich maliarsky štýl. Vstupné dáta tvorí *3D scéna* s príslušnými textúrami vo formáte *.3ds*, čo je exportný formát *3D Studio MAX*. Väčšina 3D modelovacích nástrojov podporuje export tohoto formátu na disk. Výstupom je *animácia* napr. vo formáte *.avi*, resp. jej jednotlivé políčka vo formáte *.png*.

PS je vizuálny nástroj v zmysle WYSIWYG. Užívateľ má možnosť nastavovať množstvo parametrov, ktorými ovláda vzhľad maliarskeho štýlu. Má možnosť použiť vlastné textúry štetca. Všetky zmeny parametrov sa prejaví v reálnom čase, v okne predstavujúcom náhľad v plnej kvalite.

PS má veľmi jednoduché ovládanie. Nasledujúci text popíše nastavenie, ovládanie a používanie do hĺbky. Tieto informácie sú na záver prehľadne zhrnuté v tutoriáli *Prvá animácia - krok za krokom*.

### 4.2 Inštalácia a konfigurácia

Inštaláciu PS prevedieme rozbalením archívu na *oblúbené* miesto na disku. Na korektné spustenie aplikácie musí systém spĺňať nasledujúce požiadavky:

- Nainštalovaný driver OpenGL 2.0.1 alebo vyšší, podporujúci rozšírenia `EXT_framebuffer_object` a `ARB_texture_rectangle`.
- Nainštalovaný Cg runtime.

- Cca 300 Mb RAM. Veľkosť použitej pamäte závisí od veľkosti geometrie scény.
- Dostatočný priestor na disku pre export animácie.

PS je konzolová aplikácia. Všetky oznamy o svojom behu, event. chybové hlásenia vypisuje na konzolu.

### 4.3 Príprava dát a spúšťanie

Vstupné *.3ds* súbory, spolu s použitými textúrami, musia byť umiestnené v adresári `import/`. V adresári `easybrush/brush/` sa nachádzajú textúry štetcov. Exportované animácie a screenshots sa nachádzajú v adresári `export/`. Adresár `tmp/` užívateľ nepoužíva, slúži ako dočasné úložisko dát pri exporte animácie.

Vstupný *.3ds* súbor **musí** obsahovať aspoň jednu kameru. V prípade, že scéna obsahuje viac kamier, program použije prvú. V prípade, že súbor kameru neobsahuje, program ohlásí chybu na konzolu. Vstupná scéna môže obsahovať:

- ľubovoľný počet svetiel
- Blinn / Phong materiály s textúrou farby
- animáciu (pohyb) jednotlivých objektov v scéne

Vďaka obmedzenosti popisu *.3ds* formátu a možnostiam OpenGL renderingu, sú mnohé parametre scény *ignorované*. Typicky sú to rôzne modifikátory a zložité materiály. Formát *.3ds* totiž zachycuje polygonálnu sieť (mesh) predstavujúcu model, takže všetky modifikátory geometrie sú z princípu ignorované.

Program vyžaduje na príkazovom riadku jediný parameter, ktorým je meno súboru vstupnej *.3ds* scény bez cesty! Napr: „`a.exe scene.3ds`“, pričom súbor `scene.3ds` sa spolu so všetkými textúrami nachádza v adresári `import/`.

V distribúcii je priložených niekoľko jednoduchých scén.

Textúry štetcov sa nachádzajú v adresári `easybrush/brush/`. Textúrou môže byť súbor s ľubovoľnými rozmermi, proporcie sú však ignorované. *Textúra štetca je vždy štvorcová*. Odporúča sa napr. rozlíšenie 64x64 RGBA, formát PNG. Obrázok by mal obsahovať alfa kanál. V prípade, že chce užívateľ vytvoriť štetec bez farebnej textúry, celý obrázok obrázok obsahuje biele pixely. Viz *brushtest.png*.

Export animácie prebieha v dvoch krokoch:

1. PS vyexportuje všetky snímky vo formáte .jpg s vysokou kvalitou
2. PS zavolá súbor `_encode.bat`, ktorý obrázky „zošije“ do animácie pomocou programu *ffmpeg* a vymaže ich.

Ak si užívateľ praje zmeniť parametre resp. priebeh kódovania videa, musí editovať tento súbor. Štandardne je nastavený výstup do formátu *.avi*, ktorý obsahuje *10Mbps mpeg4 layer*. V prípade potreby možno nastaviť iný codec, napríklad *xvid* alebo *mpeg2*.

## 4.4 Grafické užívateľské rozhranie

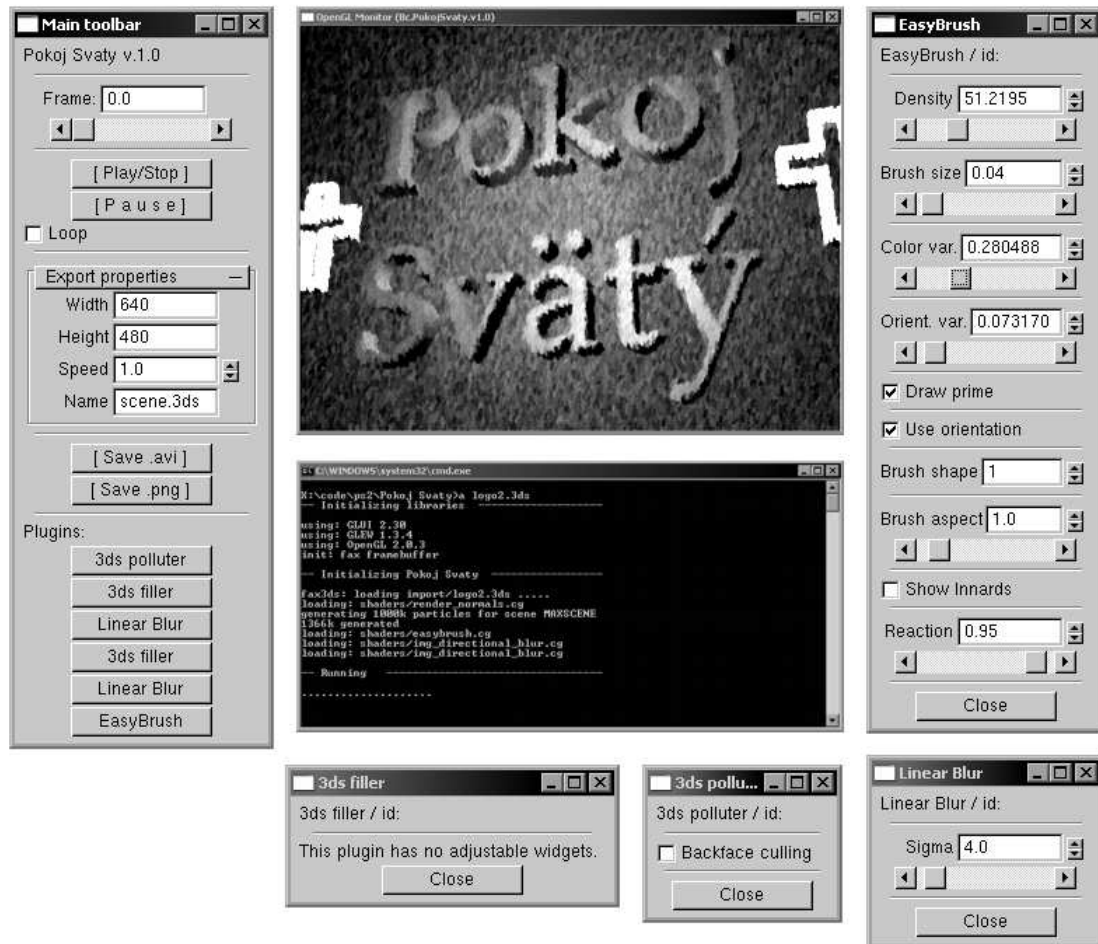
Program má jednoduché a intuitívne vizuálne ovládanie. Všetky zmeny parametrov sa prejavia v reálnom čase v náhľadovom okne v *plnej kvalite*. Okná programu zachycuje obrázok 4.1. Detailnejší popis jednotlivých ovládacích prvkov k tomuto obrázku odkazuje. Program užívateľ ukončuje zatvorením ľubovoľného okna aplikácie.

**OpenGL Monitor** poskytuje náhľad v reálnom čase. Veľkosť okna možno meniť podľa potreby.

**Konzola** informuje užívateľa o priebehu výpočtu a vnútornom stave aplikácie. V prípade akéhokoľvek problému alebo neštandardného ukončenia programu doporučujem pozorne čítať výpisy na konzolu. Môže ísť o triviálnu chybu.

**Main Toolbar** predstavuje hlavný panel programu, prostredníctvom ktorého užívateľ:

- Spúšťa animáciu, prípadne výpočet (save .avi) a orientuje sa v čase.
- Nastavuje veľkosť plátna a meno výstupného súboru.



Obr. 4.1: Okná programu

- Otvára okná jednotlivých pluginov.

**EasyBrush** je plugin zodpovedný za simuláciu ťahov štetca. Má nasledujúce parametre:

- **Density** - hustota pokrytia plátna. Užívateľ sa snaží nastaviť čo najmenšiu hustotu tak, aby v povrchu scény nevznikali diery. Typicky tento parameter nastavuje ako posledný.
- **Brush size** - veľkosť billboardov predstavujúcich ťahy štetca. Užívateľ typicky nastavuje tento parameter ako prvý, keďže podstatne určuje výsledný vzhľad.
- **Color var.** - miera náhodnosti variácie farby. Štandardne nulová.
- **Orient. var.** - miera náhodnosti variácie rotácie. Billboardy predstavujúce ťahy štetca sa snažia pootočiť tak, aby vytvorili dojem, že prilínajú k 3D povrchu.
- **Draw prime** - zapína/vypína podmaľbu. Podmaľbou rozumieme referenčný snímok farby.
- **Use orientation** - rotáciu ťahov štetca možno vypnúť.
- **Brush shape** - tvar štetca, vyberá textúru z `brush/easybrush/`.
- **Show innards** - ukáže práve nastavenú textúru štetca.
- **Reaction** - rýchlosť reakcie ťahov štetca na (event. prudkú) zmenu atribútov. Varianta motion-bluru. Je to technické riešenie problému šumu atribútov jednotlivých ťahov, plynúceho z konečnej presnosti (rozmerov) referenčných snímkov.

**Blur** má jediný parameter - veľkosť priemerovacieho okna v pixeloch.

**Polluter3ds** poskytuje možnosť znížiť množstvo zbytočne generovaných ťahov ignorovaním odvrátených polygónov.

**Filler3ds** nemá ovládacie prvky.



## 4.5 Prvá animácia - krok za krokom

1. Overím si, či mám grafickú kartu podporujúcu OpenGL 2.0.1 a rozšírenie `EXT_framebuffer_object`, napr. *nVidia GeForce 6200* alebo *ATI Radeon X800*. Ak nie, vzdám sa nádeje na spustenie programu.
2. Rozbalím `POKOJ SVÄTÝ` na disk.
3. V *oblúbenom*<sup>1</sup> 3D modelovacom programe, napr. 3D Studio MAX vytvorím požadovanú scénu. To, okrem iného, znamená, že:
  - Myslím na to, že výsledkom bude NPR rendering, takže mi stačí hrubý detail.
  - Používam materiály s Blinn / Phong osvetlovacím modelom.
  - Ak potrebujem, použijem farebné textúry.
  - Ak chcem, pridám do scény svetlá, môžem ich rozanimovať.
  - Nezabudnem pridať kameru.<sup>2</sup>
4. Uložím scénu spolu s použitými textúrami do podadresára `import/`.
5. Ak chcem, do adresára `easybrush/brush/` si pripravím vlastné textúry štetca.
6. Spustím program z príkazového riadku s jedným parametrom - menom súboru vstupnej scény. Napr. „`a.exe scene.3ds`“.
7. *Pre potreby tohoto tutoriálu spustím súbor `scene.3ds.bat` s predpripravenou scénou.* Rovnaký postup samozrejme môžem vyskúšať aj so svojou vlastnou scénou.
8. Ak zlyhá inicializácia aplikácie, pozorne prečítam výpisy na konzolu.
9. Kliknem na tlačítko [ `Play/Stop` ], aby som videl, či sa scéna približuje mojim predstavám. Vyskúšam si aj slider, ktorým sa môžem ľubovoľne posúvať v čase.
10. Skúsím nastaviť nejaký štýl. To znamená, že sa nebojím posúvať slidermi divoko sem a tam.

---

<sup>1</sup>Ak môj obľúbený nástroj `.3ds` nepodporuje, na internete nájdem exportér.

<sup>2</sup>Scénu bez kamery nenazývame scénou ale *modelom*.

11. Zastavím animáciu a kliknem na tlačítko **EasyBrush**, otvorím okno s nastavením parametrov štetca.
12. Najprv si nastavím: **Density: 20**, čím regulujem hustotu pokrytia plátna ťahmi štetca. Nízka hodnota znamená vysokú rýchlosť renderingu, ale viac artefaktov.
13. Nastavím si základné črty štýlu: Vyberiem si **Brush shape: 1** a **Size: 0.062**.
14. Nastavím si mieru náhodnosti maľby: **Color var.: 0.35** a **Orient.var.: 0.21**.
15. Môžem skúsiť kliknúť na **Use orientation**. Ale orientáciu určite chcem použiť.
16. Potlačím vznik artefaktov. Otvorím okná oboch pluginov **Blur**. (Prípadom si obrázok 3.3.) Farba konkrétneho ťahu štetca by mala približne zodpovedať farebnému priemeru pixelov zakrytej plochy. Nemusí to tak byť, ak nechcem. Nastavím preto **Sigma: 9.14**. Desatinná časť je zanedbaná, priemerovacie okienko je veľké 9x9 pixelov.
17. Mám možnosť regulovať aj dobu reakcie ťahov štetca v čase. Nastavím **Reaction: 0.9**. Pri totálnej reakcii (1.0) majú tendenciu ťahy štetcom blikať. Pri nízkej (0.6) majú tendenciu zanechávať šmuhy. To môžem použiť ako špeciálny efekt. Prednastavenú hodnotu 0.95 v zvyčajne nemusím meniť - nezbadám ani jeden z artefaktov.
18. Zrušením podmaľby (**Draw prime**) si môžem vyskúšať, že hustota nie je dostatočná - na povrchu vzniknú diery. Preto nastavím hustotu **Density: 140**. Niekedy nemusí byť vysoká hustota žiadúca.
19. Nastavil som maliarsky štýl. Teraz sa vrátim do *Main toolbar*-u.
20. Kamera sa v scéne pohybuje príliš rýchlo: nastavím **Speed: 0.5**.
21. Ak chcem, môžem zmeniť veľkosť plátna. Mohol som postupovať aj tak, že som parametre štýlu ladil na malom plátne, napr. 320x180, kvôli rýchlosti odozvy<sup>3</sup>. Teraz chcem vytvoriť HD rendering a preto nastavím veľkosť plátna na 1280x720.

---

<sup>3</sup>Všetky framebufferu použité pri renderingu majú rovnaký rozmer ako výstupné plátno. Rýchlosť odozvy tiež intenzívne závisí od hustoty - počtu častíc na plátne.

22. Ak chcem zmeniť formát videa, editujem súbor `_encode.bat`. Ak sa exportovaná animácia nedá pustiť, nepanikárim, ale nainštalujem recentné kodeky, *VLC Player* alebo skúsim editovať zmieneny súbor a zameniť `-vcodec mpeg4` za `-vcodec xvid` popřípade `-vcodec mpeg2`.
23. Kliknem na [ `save .avi` ]. Sledujem výpisy na konzole.
24. Ak všetko dobre dopadlo, animáciu mám pripravenú v adresári `export/`.
25. Výslednú animáciu môžem porovnať s prerenderovanou animáciou na CD.

# Kapitola 5

## Záver

V práci som sa opieral o nasledujúce teoretické východiská:

1. Úloha syntézy animovanej maľby je veľmi slabo definovaná. Medzi klasickými výtvarnými postupmi nemá ekvivalent. Preto môže vyzerať prakticky *akokoľvek*. Je to nová výtvarná forma, ktorá otvorila možnosti pre výtvarný experiment. Čo by jeden považoval za nepríjemný artefakt, môže druhý považovať za lákavú črtu.
2. Jediné kritérium hodnotenia výsledkov, je kritérium *estetické*. Animovaná maľba musí uspieť v súťaži s ostatnými výtvarnými technikami.
3. Úloha animovanej maľby je nesmierne komplexná. Ako sa má maľba správať v čase? Reálny systém musí umelcovi poskytnúť čo najvyššiu dimenziu slobody. Nemenej dôležité je však jednoduché a intuitívne užívateľské rozhranie.

V práci som sledoval dva ciele:

1. Primárnym cieľom bolo navrhnuť a implementovať *náčrt* systému, ktorý by umožňoval umelcovi užívateľsky ovládnuť komplexnú úlohu syntézy animovanej maľby. Tento systém by mal tak isto slúžiť ako framework pre ďalší výskum možností animovanej maľby.
2. Druhým cieľom bolo *v rámci tohto frameworku* implementovať modifikáciu algoritmu Barbary Meier [4] a vytvoriť tak užívateľskú aplikáciu, s ktorou by aj málo skúsený animátor vedel vytvoriť esteticky príťažlivé práce.

Výsledky svojej práce teda budem hodnotiť vzhľadom k týmto dvom cieľom. Ukážkové animácie možno nájsť na priloženom CD. Statické obrázky možno nájsť na CD a vo farebnej prílohe. Preložená aplikácia je určená pre platformu *win32*.

## 5.1 Pokoj Svätý ako experimentálny framework

### 5.1.1 Zhrnutie dosiahnutých výsledkov

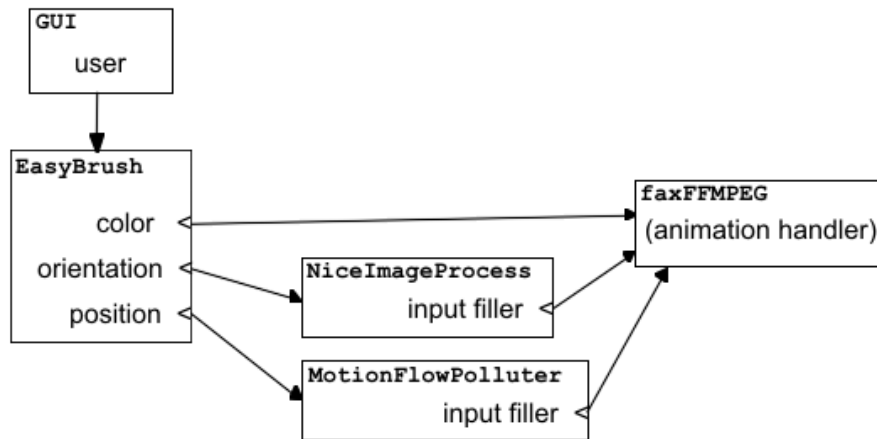
Úlohu syntézy animovanej maľby simulovaním ťahov štetca som rozdelil do troch *navzájom nezávislých vrstiev*. Na základe tohoto rozdelenia som navrhol otvorený a modulárny systém na báze tzv. *pluginov*. Vznikajú tak 3 triedy pluginov, každá zodpovedná za špecifickú úlohu v procese syntézy. *Pluginy možno ľubovoľne prepájať, čím sa radikálne zvyšuje škálovateľnosť prístupu - diverzita maliarskych štýlov a efektov.*

Triedy pluginov:

1. **Rozmiestnenie ťahov štetca** v čase na plátne. Toto rozmiestnenie by malo zachovávať nasledujúce vlastnosti<sup>1</sup>:
  - *Zdanie náhodnosti.* Ťahy na plátne rozmiestnil umelec, nie počítačový program.
  - *Uniformita.* Zabránenie vzniku rušivých zhlukov ťahov štetca.
  - *Súdržnosť v čase.* Potlačenie shower-door efektu a iných artefaktov.
2. **Príprava referenčných snímok**, predstavujúcich alternatívne popisy scény. Táto vrstva zahŕňa aj ich ďalšie spracovanie, resp. miešanie. Práve tu vzniká najviac možností pre výtvarný experiment.
3. **Simulácia ťahov štetca.** Atribúty jednotlivých ťahov sú vyhodnotené na základe interpretácie vstupných referenčných snímok.

---

<sup>1</sup>Ak sú v konkrétnom prípade žiaduce.



Obr. 5.1: Konfigurácia: Motion Flow

Navrhnutý framework implementuje v C++ tieto oddelené vrstvy abstraktnými triedami (rozhraniami). Plugin vzniká implementáciou konkrétneho rozhrania. GUI umožňujúce ľubovoľné prepájanie pluginov zatiaľ nie je implementované. Fixné zapojenie pluginov predstavuje implementáciu modifikácie algoritmu Barbary Meier v kontexte frameworku. Schematicky je zachytené na obr. 3.3.

Framework je postavený výlučne nad OpenGL a open-source/free-software knižnicami a je preložený v *gcc* pomocou balíka *minGW*.

### 5.1.2 Smer ďalšej práce

Prvým krokom mala byť implementácia front-end / GUI frameworku umožňujúca prepájanie pluginov bez rekompilácie. Ďalej možno postupovať napríklad nasledovne:

- Implementácia handleru pre animácie a pluginu rozmiestňujúceho ťahy štetca na základe optického toku (napr. [5], [6]). Tým by sa framework oslobodil od potreby poznať geometriu scény. Príklad možného prepojenia pluginov zachycuje obr. 5.1.
- Implementácia pluginov základných operácií digitálneho spracovania obrazu.
- Implementácia pokročilejšej techniky simulácie média.

- Prieskum možností syntézy v reálnom čase. Rýchlosť frameworku závisí len od rýchlostí jeho komponentov.

## 5.2 Pokoj Svätý ako užívateľská aplikácia

### 5.2.1 Zhrnutie dosiahnutých výsledkov

Jediným hodnotiacim kritériom, z pohľadu užívateľa - umelca, je estetická príťažlivosť, jednoduchosť ovládania a prístupnosť alternatívneho riešenia.

POKOJ SVÄTÝ predstavuje alternatívu jedinému riešeniu; komerčnému softwaru *NPR1eyes* od firmy *Infografica*, integrovaného do prostredia 3DS Max.<sup>2</sup> Hlavným nedostatkom PS je strohé užívateľské rozhranie - komerčnému softwaru sa ani zďaleka nepribližuje, poskytuje však *porovnateľné* výsledky. Na druhú stranu je ovládanie PS natoľko jednoduché, že ho ľahko ovládne aj laik, ktorý nemá s animačnými nástrojmi veľké skúsenosti.

Lákavou črtou PS je, že poskytuje náhľad v reálnom čase.<sup>3</sup> To (aj) vďaka rýchlemu hardwaru, ktorý podporuje shader model verziu 3. Časť výpočtu je presunutá na GPU. Dôraz som kládol na estetickú príťažlivosť a zachovanie dizajnu, rýchlosť bola až na druhom mieste.

Pri implementácii (metódy B. Meier) som sa potýkal s mnohými technickými ťažkosťami. Niektoré z nich sa mi podarilo vyriešiť, iné odložiť. Celý postup som sa snažil efektívne rozdeliť medzi GPU a CPU. Hlavné úzke hrdlo nepredstavuje rasterizácia (desať)tisícov billboardov, na modernom GPU dostatočne rýchla, ale ich zotriedenie. Ťahy sa musia kresliť v správnom poradí a toto triedenie nemožno principiálne obísť. Preto je kľúčové pokryť 3D scénu čo najmenším počtom ťahov. Moje riešenie rozdeľuje rekurzívne polygón a tým zabezpečuje uniformitu rozptýlenia častíc — pri minimálnom počte častíc zabraňuje vzniku dier. Možným vylepšením by bolo využiť paralelnej architektúry a triediť na GPU.

Tvorbu neprijemného artefaktu - šumu atribútov jednotlivých ťahov, ktorý z princípu vyplýva z nepresnosti referenčných snímok, som riešil

---

<sup>2</sup>Riešenie tohoto softwaru je *zjavne* inšpirované Barbarou Meier.

<sup>3</sup>Rýchlosť intenzívne závisí od nastavení maliarskeho štýlu a rozmerov plátna.

variantou motion-bluru. Užívateľ má možnosť regulovať rýchlosť reakcie ťahov na zmenu atribútov a dosiahnuť tak väčšej spojitosti v čase.

Vzniku iného nepríjemného artefaktu - pribúdaniu a miznutiu jednotlivých ťahov štetca som zamedzil tým, že sa tak deje vždy vo spodných vrstvách maľby. Ťahy na povrchu sú najstabilnejšie. Artefakt teraz možno zbadať len na ostrých hranách. Konečným vylepšením by bol fade-in / fade-out ťahov cez alfa kanál.

### 5.2.2 Smer ďalšej práce

Okrem vyššie uvedených krokov, týkajúcich sa GUI a implementácie ďalších pluginov, je žiadúce aby PS:

- Predovšetkým umožňoval load/save nastavení.
- Umožňoval klúčovanie parametrov.
- Bol rýchlejší - lepší manažment framebufferov, implementácia triedenia na GPU, ešte lepšie generovanie ťahov.
- Tvoril menej artefaktov. K tomu je potrebný intenzívny beta-testing.
- Vedel pracovať s akoukoľvek zložitou geometriou. Vhodný by bol aj exportér z 3D studia zachycujúci zmenu parametrov modifikátorov, ktoré formát .3ds nepopisuje.
- Implementoval aj iné maliarske štýly, napr. akvarel.
- Dokázal spracovávať vstupné animácie (ako referenčné snímky). Oslobodil by sa tak od obmedzení OpenGL renderingu a posunul do kategórie nástrojov animačného postprocessingu.

Pokoj Svätý je predovšetkým výskumný framework, ktorý urobil smerom k bežnému používaniu len základné kroky. Pokúsil sa ale zdefinovať smer, ktorým by tieto kroky mohli viesť. Dosiahnuté praktické výsledky - animácie posúdi najlepšie každý sám.



# Literatúra

- [1] Gooch A., Gooch B.: *Non-Photorealistic Rendering*, A K Peters, Massachusetts.
- [2] Schlechtweg S., Strohotte T.: *Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation*, Morgan Kaufmann Publishers, San Francisco, 2002.
- [3] Sperl D.: *Realtime Painterly Rendering for Animation*, cite-seer.ist.psu.edu/673440.html, 2004
- [4] Meier B.: *Painterly Rendering for Animation*, SIGGRAPH 96 Conference Proceedings, New York, 1996
- [5] Hertzmann A., Perlin K.: *Painterly rendering for video and interaction.*, Proceedings of NPAR 2000, Annency, 2000
- [6] Hays J. Essa I.: *Image and video based painterly animation*, Proceedings of NPAR '04, Annency, 2004

# Dodatok A

## Obsah disku CD-ROM

V nasledujúcich adresároch na CD možno nájsť:

- /app - POKOJ SVÄTÝ - aplikačnú distribúciu, ktorá obsahuje:
  - Binárky preložené pre platformu *win32*.
  - Všetky potrebné *.dll* knižnice.
  - Niekoľko predpripravených textúr štetca.
  - Niekoľko predpripravených *.3ds* scén, ktoré možno spustiť *.bat* súbormi.
- /doc/doxygen - Programátorskú dokumentáciu vytvorenú systémom *Doxygen*.
- /doc/thesis - Text bakárskej práce vo formáte *.pdf* spolu so zdrojovým textom v jazyku *csLaTeX* a obrázkami.
- /src - Zdrojové kódy. Pokyny pre preloženie obsahuje súbor *README*.
- /visual/avi - Ukážkové animácie v formáte *.avi*.
- /visual/png - Ukážkové obrázky vo formáte *.png*.

**Dodatok B**

**Farebná príloha**