

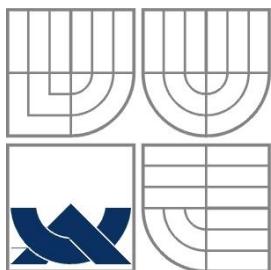
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií  
Faculty of Information Technology

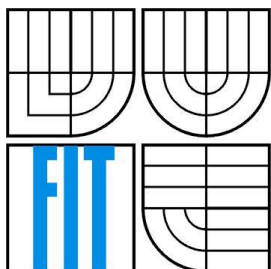
BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

Brno, 2016

Jan Benda



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

## UMĚLECKÉ ZOBRAZENÍ SCÉNY

ARTISTIC SCENE VISUALIZATION

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN BENDA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. JOZEF KOBRTEK

BRNO 2016

## **Abstrakt**

Cílem této práce je prostudovat vybrané metody nerealistického uměleckého zobrazení (cell shading, pencil sketch shading, watercolor shading) a napodobit dané techniky při vykreslování objektů ve 3D prostoru s použitím knihovny *OpenGL*. Oblastí využití těchto technik je široké – od grafických aplikací, přes počítačové hry až po animované filmy. Tato práce popisuje návrh a implementaci těchto technik: cell shading, skica tvořená tužkou a akvarelová malba.

## **Abstract**

The goal of this work is studying of selected artistic scene visualization (cell shading, pencil sketch shading) and simulate these techniques in the rendering of objects in 3D space using *OpenGL*. Applications of these methods is wide – from graphical application, through computer games to animated films. This work describes design and implementation of these methods: cell shading, pencil sketch shading and watercolor shading.

## **Klíčová slova**

OpenGL, C++, umění, umělecké zobrazení scény, počítačová grafika, cell shading, skica tvořená tužkou, akvarelová malba

## **Keywords**

OpenGL, C++, art, artistic scene visualization, computer graphic, cell shading, pencil sketch shading, watercolor shading

## **Citace**

Benda Jan: Umělecké zobrazení scény, bakalářská práce, Brno, FIT VUT v Brně, 2016

# Umělecké zobrazení scény

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jozefa Kobrtka.

.....  
Jan Benda  
16. května 2016

## Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu Ing. Jozefu Kobrtkovi, za jeho čas, cenné připomínky a odbornou pomoc na konzultacích k této bakalářské práci. Dále bych rád poděkoval svým rodičům, přítelkyni a přátelům za podporu při studiu.

© Jan Benda, 2016

*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

<b>Obsah</b>	<b>5</b>
<b>1. Úvod</b>	<b>2</b>
<b>2. Umělecké a grafické techniky</b>	<b>3</b>
2.1 Cell Shading .....	3
2.2 Skica tvořená tužkou .....	5
2.3 Akvarelová malba .....	7
2.4 Použité techniky .....	9
2.4.1 Osvětlovací modely .....	9
2.4.2 Detekce a zvýraznění hran .....	10
2.4.3 Vyhlazení textur .....	10
2.4.4 Stupně šedi .....	11
<b>3. Navrhované techniky implementace</b>	<b>12</b>
3.1 Návrh aplikace.....	12
3.2 Knihovny .....	12
<b>4. Implementace</b>	<b>13</b>
4.1 Tvorba aplikace .....	13
4.1.1 Vykreslování s pomocí OpenGL.....	13
4.2 Návrh implementace technik.....	14
4.2.1 Cell shading.....	14
4.2.2 Skica tvořená tužkou .....	14
4.2.3 Akvarelová malba .....	14
4.3 Implementace aplikace .....	15
4.4 Implementace technik.....	15
4.4.1 Implementace cell shadingu .....	15
4.4.2 Implementace pencil sketch shadingu.....	19
4.4.3 Implementace watercolor shadingu.....	21
4.5. Výsledné implementace .....	22
<b>5. Závěr</b>	<b>23</b>
<b>Použité zdroje</b>	<b>24</b>
<b>Seznam příloh</b>	<b>25</b>
Příloha A: Popis spuštění a ovládání aplikace.....	25
Obsah DVD .....	25
<b>Příloha A</b>	<b>26</b>

# 1. Úvod

Umění je součástí lidské kultury již od počátků civilizace. Jedním z jeho prvních odvětví byla kresba a později i malířství. Vzniklo nepřeberné množství technik a způsobů kresby. Oproti tomu počítačová grafika existuje kratší dobu. Za dobu své existence však prodělala drastický rozvoj. V současné době lze počítačovou grafiku rozdělit do několika kategorií např. 2D grafika, 3D grafika, GUI atd.

Grafika ve 3D se od svých počátků v 60. letech do současnosti značně změnila. Z počátku se jednalo o zobrazování jednoduchých drátěných modelů, postupem času se zvyšujícím se výkonem počítačů zvětšovala složitost vykreslovaných objektů. Od počátku však bylo hlavním cílem napodobit reálný svět. Z této myšlenky vychází celá řada algoritmů a technik dnes již zcela běžných. Oproti tomu umělecké techniky a výrazná stylizace se držely na okraji tohoto trendu. Tyto techniky byly limitovány malým výpočetním výkonem své doby a menším využitím. Proto se z počátku tyto techniky nejčastěji objevovaly ve 2D grafice. Avšak s přibývajícím výkonem počítačů a zájmem o jiné styly vykreslování se dané techniky postupně dostaly do popředí zájmu.

Dnes existuje celá řada algoritmů a technik pro vytvoření efektu uměleckých technik. Ne vždy je možné je zcela napodobit. Daná práce se pokusí napodobit dané techniky.

Zvolenými technikami jsou cell shading, skica vytvořená tužkou a akvarelová malba. Těmto technikám bude věnována kapitola 2. Budou zde popsány jejich podstatné znaky a nastudované implementace. Dále budou v práci popsány matematické metody použité pro jejich napodobení. Kapitola 3 bude věnována návrhu aplikace, ve které budou popsány podstatné body, jež bylo potřeba vyřešit. Následující kapitola 4 je věnována návrhu a implementaci uměleckého vykreslování. Všechny nalezené problémy zde budou popsány a řešeny. Výsledné shrnutí bude popsáno v kapitole 5. Zde budou zhodnoceny dosažené výsledky a případné možnosti dalšího rozšíření. Poslední kapitolou je závěr, shrnující dosažené výsledky.

Tato práce se zaměřuje na 3D grafiku napodobující umělecké styly. Tato grafika je velmi často využívána v počítačových hrách a animovaných filmech. Daná práce se snaží napodobit již existující metody, případně vytvářet vlastní na základě již existujících technik. Tyto metody pak mohou sloužit v dalších grafických aplikacích.

Cílem této bakalářské práce je napodobit a demonstrovat vybrané techniky nerealistického zobrazování s použitím aplikačního rozhraní *OpenGL*. na téma Umělecké zobrazení scény

. Praktická část se zaměřuje na vytvoření aplikace v jazyce *C++*, jež demonstruje techniky uměleckého/nerealistického zobrazování scény s využitím tohoto rozhraní. Z praktické práce poté bude vytvořeno video, propagující danou práci.

## 2. Umělecké a grafické techniky

Obsah této kapitoly je věnován seznámení s vybranými uměleckými/grafickými technikami, které jsou v dané práci dále zpracovány nebo napodobeny. Zvolenými technikami jsou cell shading, pencil scatch shading a watercolor shading. Postupně je popsáno jejich použití v běžném životě, jejich podstatné znaky a nejčastější implementace těchto technik v počítačové grafice. Dále jsou zmíněny očekávané problémy s napodobením těchto technik.

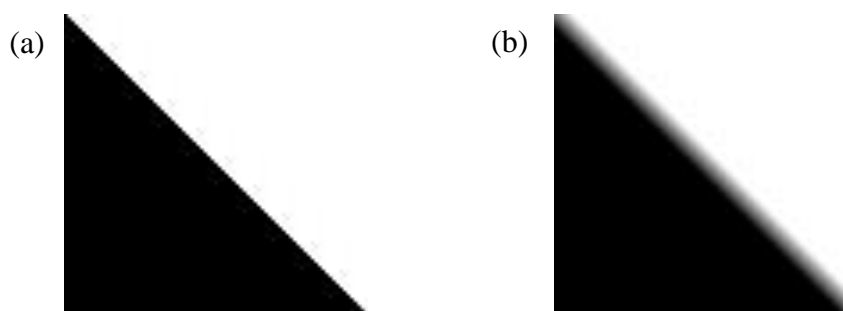
### 2.1 Cell Shading

Daná technika je známá také pod názvem toon shading. Jde o typ nerealistického zobrazení pro počítačovou 3D grafiku. Tato technika je navržena tak, aby napodobovala umělecký styl komiksů a kreslených filmů. Nejčastěji se s ní lze setkat v počítačových hrách (obr. 1), ale objevuje se také v animovaných filmech. Technika je často doplněna speciálně upravenými modely a texturami, jejichž proporce neodpovídají reálným, ale zdůrazňují komiksovou stylizaci. Existuje celá řada způsobů implementace této techniky. Od technik prováděných přímo během vykreslování scény až po post-processingové úpravy.



Obr. 1 Ukázka cell shadingu z videohry „Naruto Shippuden: Ultimate Ninja Storm 4“

V komiksech se tento styl vyznačuje pastelovými barvami, ostrými přechody mezi jednotlivými barvami a stíny. V kresbě tak vznikají stejnobarevné plochy pro osvětlenou část a šedé až černé plochy pro zastíněnou část. Přechody mezi barevnými oblastmi mohou být skokové (obr. 2a), ale také částečně gradientní (obr. 2b). To znamená, že tenká část mezi jednotlivými barevnými oblastmi plynule přechází z jedné barevné části do druhé.

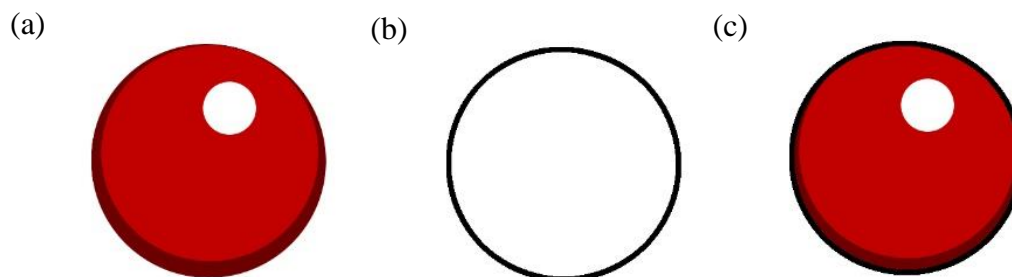


Obr. 2 Znárodnění: (a) skokového přechodu barev a (b) gradientního přechodu barev

V počítačových hrách jsou tyto efekty často doplněny za pomoci upravených textur. Odlesky a odrazy bývají tlumené. Oblasti v texturách jsou často rozděleny tak, aby každá oblast byla ve stejném odstínu barvy a působila jako vybarvená jednou barvou. Cílem těchto úprav je zajistit efekt plochosti zobrazovaného modelu.

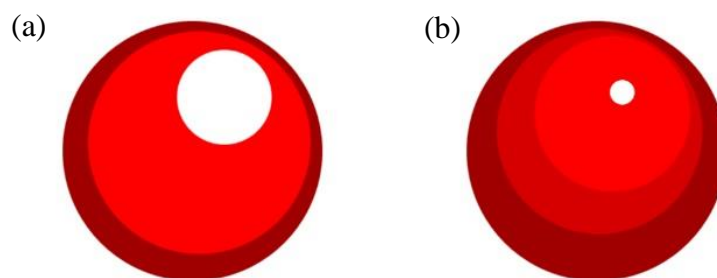
Dalším komiksovým znakem této techniky bývají zvýrazněné obrysy kreslených předmětů. Ty jsou buď jednolitě černé, nebo obarveny podle dané barevné oblasti tmavším odstínem dané barvy. To znamená, že pokud je například daná barevná oblast červená, tak okraje této oblasti budou tmavě červené.

Princip implementace této techniky lze shrnout do tří částí, viz obr. 3. První část se zaměřuje na vykreslení objektů na obrazovku. Stínování vykreslených objektů nebývá provedeno realisticky, tak aby vznikly plynulé přechody odstínů barev mezi osvětlenými a neosvětlenými částmi. Namísto toho je to provedeno tak, aby vzniklo několik oddělených oblastí barev/odstínů. Vykreslená scéna díky tomu již působí ploše. V další části vykreslování je vytvořeno zvýraznění okrajů. Obvykle se jedná o černou čáru okolo všech vykreslených objektů buď tmavším odstínem ohraničené oblasti a/nebo čistě o černou. Existuje více vhodných technik vytvoření obrysu, které budou vysvětleny dále v této kapitole. Tímto postupem získáme obrysy všech objektů ve scéně. V poslední části algoritmu se vezme vykreslená scéna z první části a k ní je přidán zvýrazněný okraj. Výsledek působí komiksovým dojmem.



Obr. 3 Algoritmus cell shadingu: (a) vykreslení objektu, (b) vytvoření obrysu a (c) kombinace

V článku [1] jsou popsány různé metody vytvoření cell shadingu. Nejjednodušší metoda nejprve provádí vykreslování objektů s použitím Lambertova osvětlovacího modelu pro každý bod objektu. Takto získaná hodnota intenzity osvětlení v bodě je poté porovnána s uživatelem vytvořenou vzestupnou množinou desetinných čísel. Tato množina představuje rozdělení osvětlení. Pokud se vypočtená intenzita osvětlení nachází mezi dvěma body této množiny, je vrácena dolní (nebo naopak horní) hranice. Tím vznikne skokové osvětlení. Rozložení množiny by mělo být vhodně zvoleno, protože výrazně ovlivňuje výsledné ladění obrazu, viz obr. 4.



Obr. 4 Rozložení intenzit osvětlení pro množinu: (a)  $\{0; 0.2; 0.8; 1\}$  a (b)  $\{0.3; 0.5; 0.95; 1\}$



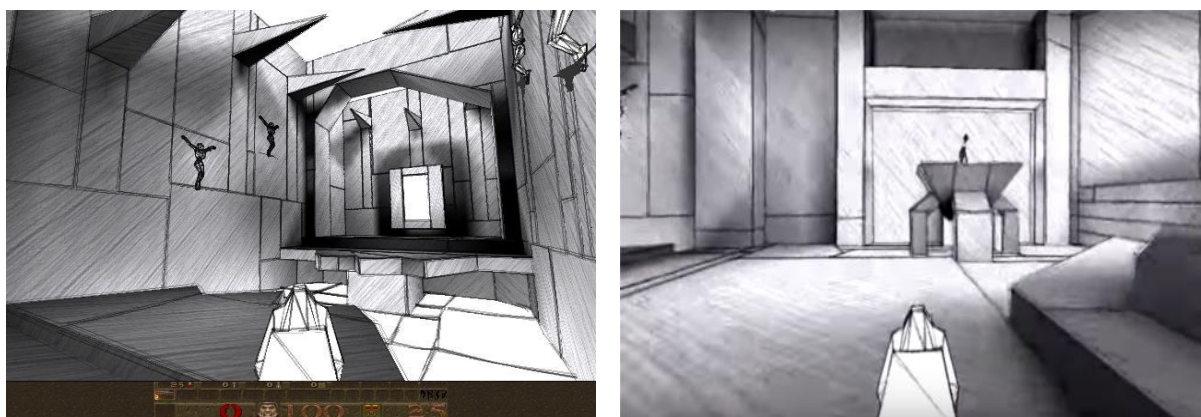
Zvýraznění okrajů je provedeno ještě před vykreslením modelů. A to za pomoci drátěného modelu právě vykreslovaného objektu. Ten je nejprve vykreslen tlustou čarou a na něj je vykreslen model samotný. Takto vykreslený objekt má zvýrazněnou siluetu a cell shadingové osvětlení.

Další technika popsaná v článku je také zaměřena na cell shading, jenž mění úroveň detailů podle vzdálenosti. S větší vzdáleností kamery od vykreslovaného modelu se snižují detaily vykreslovaného objektu. To je dosaženo pomocí mipmap textur. Dále je zkoumán stylizovaný odraz. Toho je dosaženo za pomoci použití Blinnovi metody přes Phongovo spekulární osvětlení.

## 2.2 Skica tvořená tužkou

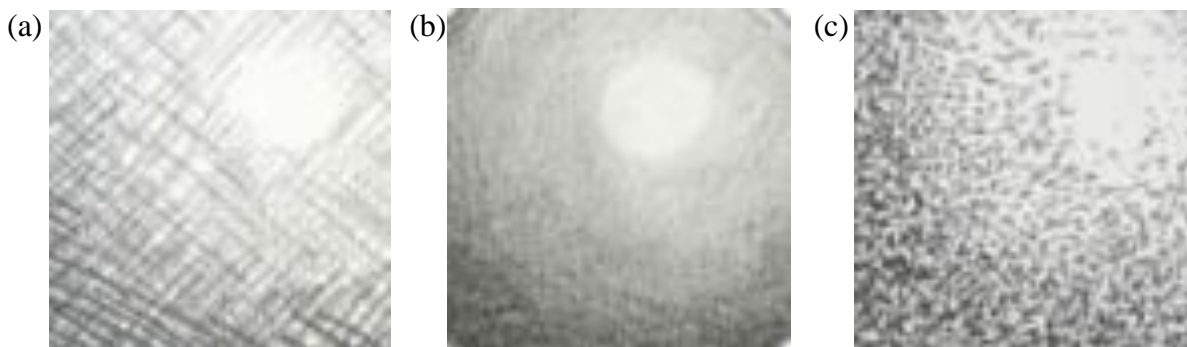
Náčrt (též skica) je jednou z nejstarších technik kreslení. Je možno jej vytvořit uhlím, pastelem, perem anebo jak je tomu v případě zvolené techniky tužkou. Nejčastěji se daná technika zhotovuje na papír, ale konkrétní typ papíru však není dán. Úroveň detailu v náčrtu je značně závislá na autorovi. Může se jednat o rychlý náčrt, jehož vytvoření trvá pár minut až po komplexní malbu, zachycující všechny detaily zobrazované scény. Skica nejčastěji slouží jako prvotní zachycení konceptů, nápadů a uplatňuje se v celé řadě odvětví průmyslu a umění.

Nejčastěji je technika spojena s designem. Skica může být barevná, ale nejčastěji je však v odstínech šedi. Daná technika je obvykle spojena s reálnou kresbou či úpravou fotografií ve formě filtrů. Často se objevuje i v kreslených filmech, v nich bývá hodně upravená a velice detailní. Pouze zřídka se objevuje přímo ve videohrách např. *Non-PhotoRealistic Quake*, viz obr. 5 [5].



*Obr. 5 Ukázka z projektu Non-PhotoRealistic Quake*

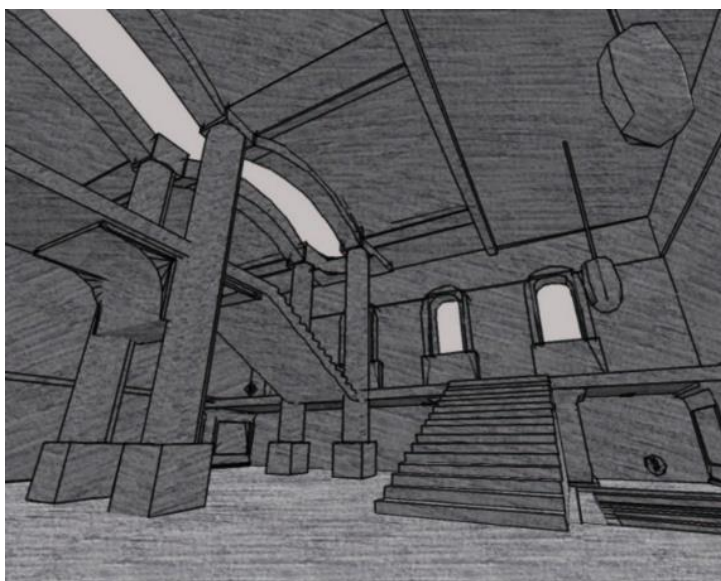
Hlavní rysy této techniky jsou zvýrazněné obrysy objektů ve skice. Stínování nemusí být vždy zahrnuto. Povrch a struktura papíru, typ použité tužky, zkušenosti a schopnosti malíře jsou nejčastější faktory ovlivňující výslednou skicu. Pokud jsou tahy tužky vidět, jedná se nejčastěji o čáry, směřující do jednoho směru a zpět, které se navíc mohou vzájemně křížit. Mezi nepoužívanější techniky stínování tužkou patří plynulý přechod čar (obr. 6a), odstínů (obr. 6b) a tečkování (obr. 6c). Technik na stínování existuje celá řada a lze je vzájemně mezi sebou kombinovat.



*Obr. 6 Způsoby stínování pomocí tužky: (a) čáry, (b) gradient a (c) tečky*

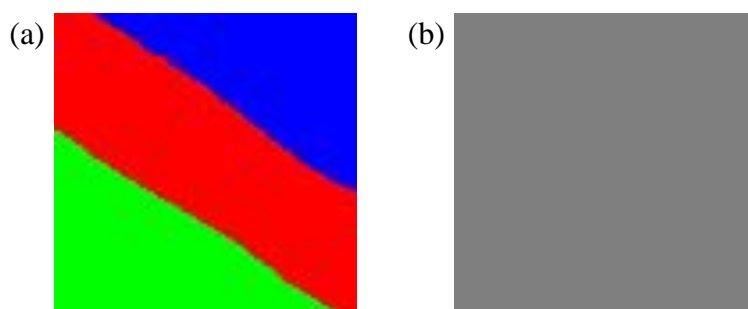
V projektu Non-PhotoRealistic Quake [6] je tato technika napodobena na známé počítačové hře Quake. V prvním kroku této techniky jsou vykresleny pouze polygony objektů a prostředí. To je provedeno kvůli tomu, aby hrany, které mají být zakryté, byly zakryty. Tento krok je udělán tak, že jsou vykresleny čáry z jednotlivých vrcholů k ostatním vrcholům. Čáry jsou vykresleny mírně kmitavě, s proměnlivou šířkou a s jistou průhledností, aby vznikl dojem rychlých tahů tužky. Všechny čáry jsou kresleny několikrát, čímž vznikl dojem čar kreslených přes sebe. Tím je zajištěno, že vykreslený obraz působí dojmem skici, kreslené v reálném čase. Původní textury vykreslovaných objektů jsou však zcela zanedbány.

Z projektu Non-PhotoRealistic Quake se odkazují také v [6]. Zde je tato technika napodobena podobně jako v projektu. Je však aplikována na již existující aplikace. V prvním kroku techniky jsou opět vykresleny pouze siluety objektů a jejich ostré hrany tlustou tmavou čarou. Tím jsou zdůrazněny hrany těchto objektů tak, aby byl jejich tvar nejlépe zachycen. Poté jsou na všechny trojúhelníky ve scéně aplikovány textury tužky a to tak, aby směr čar co nejlépe odpovídal orientaci povrchu objektu. Toho je dosaženo tak, že je vybrán směr těchto čar, který je kolmý k normále vykreslovaného trojúhelníku v projekční rovině. Výsledný obraz působí dojmem skici v reálném čase, viz obr. 7.



*Obr. 7 Ukázka techniky na hře Quake*

Mimo zmiňované implementace existuje celá řada dalších. Často je tato technika použita na post-processingové zpracování již vytvořeného obrazu (úprava fotografií). Pro zobrazení ve 3D je vhodné upravit originální textury modelů tak, aby výsledek byl v odstínech šedi a odpovídal původní textuře. Dále je vhodné zachovat zvláště obrysy, protože samotné stínování nemusí vždy dávat přesnou představu o geometrii. To je dáno tím, že všechny barvy jsou převedeny do odpovídajícího odstínu šedi. Tím je však ztracena informace o různých barevných oblastech, viz obr. 8. Ohraničení hran modelu lze vytvořit například již zmiňovanou detekcí hran.



Obr. 8 Problém převodu: (a) barevné textury do (b) odpovídajícího odstínu šedi

### 2.3 Akvarelová malba

Akvarel je jedna z nejvíce rozšířených a velice starých malířských technik. Daná technika je jedna z nejsvobodnějších uměleckých stylů a výsledný obraz se liší autor od autora. Existuje nepřeberné množství uměleckých stylů spojených s akvarelem. Od realistických, přes surrealistické až zcela abstraktní styly. Způsob a styl kresby je plně závislý na autorovi. Provádí se barvami ředěnými vodou, které ne zcela kryjí podklad. Podklad pro malbu může být různý, nejčastěji se používá plátno a papír, ale lze použít i jiné materiály např. plast, kůži či dřevo. Kresba se provádí pomocí různých druhů štětců, ale lze malovat i jinými nástroji a pomůckami např. pomocí rukou.

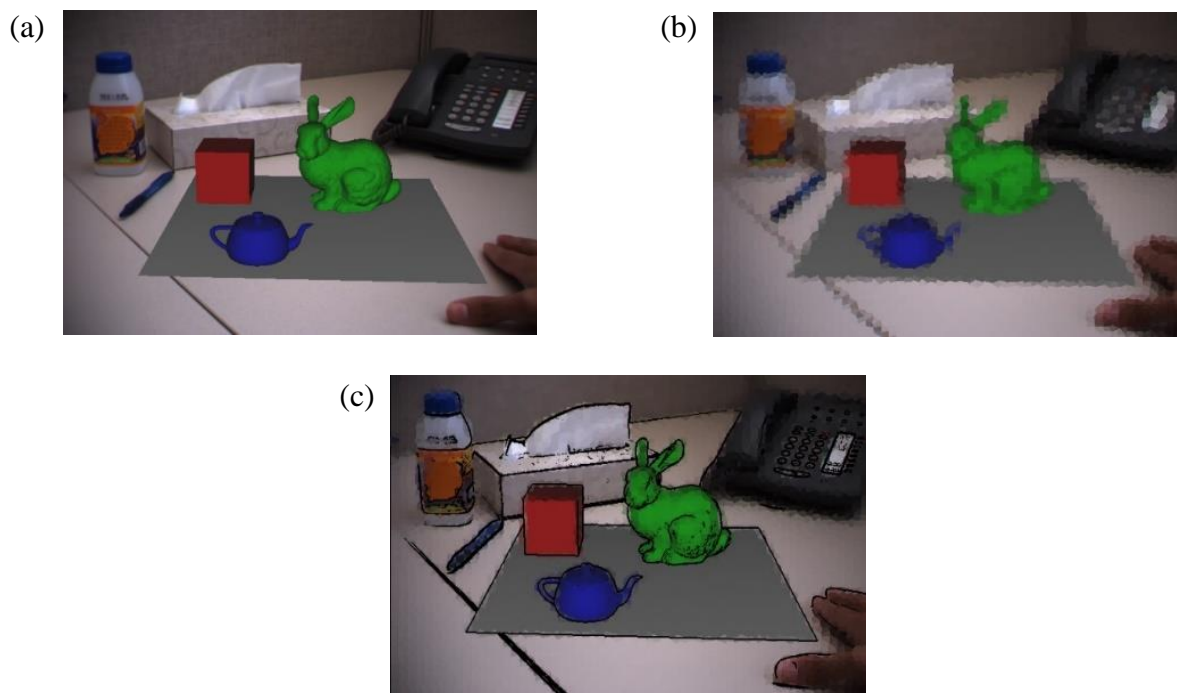
Využití této techniky je zcela umělecké. Před vynalezením fotografie byla tato technika společně s olejomalbou jediným způsobem zachycení zobrazení reality. To vedlo ke značné oblíbenosti této techniky a jejímu rozšíření napříč dějinami. Vynález fotografie ji však vytlačil jen do oblasti umění. Velice zřídka se lze s danou technikou setkat v kreslených filmech a i tak jde většinou o experimentální snímky. V informatice lze akvarel nalézt opět ve spojení s úpravou fotografií. Občas se vyskytuje v počítačových hrách, obvykle však ve 2D hrách nebo ve spojení s dalšími technikami 3D zobrazování.

Hlavními rysy je velká kolísavost barev, kdy i jednotná oblast barvy má značné výkyvy. To je způsobeno nestejným přilnutím a schnutím barvy na podkladu. Dále je přes barvu vidět struktura a povrch podkladu. Mezi jednotlivými barvami navíc vznikají rozpité oblasti, kde se jednotlivé barvy vzájemně mísí (obr. 9) [8]. Barvy se navíc občas i přelévají z plánovaných oblastí a tím pádem se mohou objekty stát méně zřetelnými. Proto často bývají kresleny i částečné okraje na místech, kde barva přetekla. Navíc příliš malé detaily nelze kvůli rozpíjivosti barvy vůbec vykreslit.



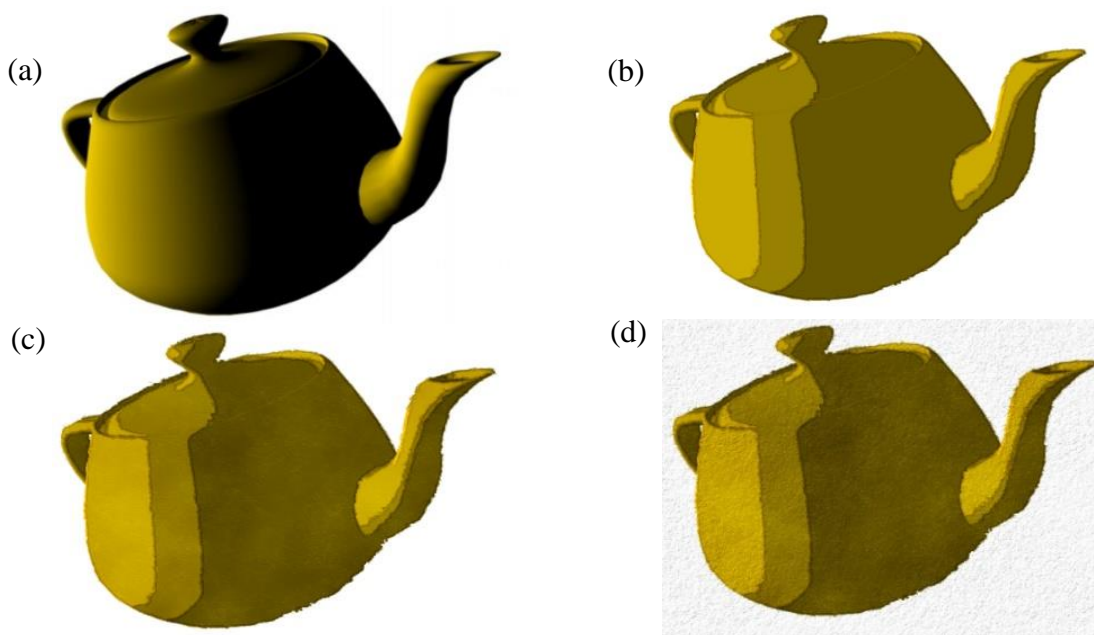
*Obr. 9 Ukázka akvarelu*

Nejčastější implementací na počítači bývají post-processingové filtry použité v programech na úpravu fotografií či videích. V článku [3] je provedeno napodobení akvarelu na rozšířené realitě. Prostor vstupního obrazu (obr. 10a) je nejprve rozdělen do dlaždic pomocí Voreného diagramů. Pouze každým desátým průchodem je vytvořeno nové dlaždicové rozdělení obrazovky a to z důvodu optimalizace, protože vytváření Voreného diagramů je časově náročný proces. Každá dlaždice je poté vybarvena průměrnou hodnotou barvy pro danou dlaždici z originálního obrázku. Tím je obraz vizuálně rozdělen do barevných částí. To ukazuje možnost implementovat danou techniku pomocí rozdělení do barevně a vizuálně oddělených oblastí (obr. 10b). V poslední části je provedena detekce hran na původním vstupním obraze. Hrany jsou poté vykresleny společně s obrazem rozděleným do barevných oblastí. Výsledný obraz působí dojmem akvarelu (obr. 10c).



*Obr. 10 Postup vytváření akvarelu: (a) vstupní obraz (b) rozdělení do barevných oblastí a (c) výsledný obraz*

V literatuře [4] je popsána další metoda implementace této techniky. Pro tuto techniku je vytvořeno několik speciálních textur. V první fázi se nejprve provede cell shading bez zvýraznění okrajů. Tím je objekt vizuálně rozdělen do několika barevných oblastí. Poté je doplněn efekt kolísavosti barvy, který je proveden posunutím souřadnic barvy pro daný pixel za pomoci předvytvořené textury papíru. Následuje ztmavení hran, které je provedeno podle předpokladu, že hrany se objevují tam, kde jsou náhlé změny barev. Je tedy provedena detekce hran a v místech hran jsou jednotlivé pixely více ztmaveny (obr. 11b). Dalším krokem úpravy je přidání turbulentního proudění barvy a zrnitosti. Tím je do obrazu přidán efekt nerovnoměrného rozložení barvy (a to pro jinak stejně barevné oblasti) a zrnitosti. Toho je dosaženo za pomoci předvytvořených textur v odstínu šedé. Výsledek daného kroku lze vidět na obr. 11c. V poslední fázi úprav je aplikována textura papíru, tak aby celý obraz působil dojmem akvarelu na plátně, viz obr. 11d.



Obr. 11 Ukázka napodobení akvarelu: (a) vstupní obraz, (b) cell shading a kolísavost barev, (c) turbulentní proudění a zrnitost a (d) výsledný obraz

## 2.4 Použité techniky

Pro napodobení jednotlivých technik je potřeba řada metod pro práci s obrazem. Dané techniky budou popsány v této podkapitole.

### 2.4.1 Osvětlovací modely

V práci jsou použity dva osvětlovací modely a to Phongův a v něm zahrnutý Lambertův model. Phongův osvětlovací model se skládá ze tří složek osvětlení: (i) ambientní, (ii) difuzní a (iii) spekulární. Ambientní složka osvětlení napodobuje odražené a rozptýlené světlo vzniklé odrazy od okolních předmětů. Další složkou je difuzní osvětlení. Tato složka je popsána Lambertovým osvětlovacím modelem. Udává intenzitu světla odraženého od matného povrchu objektu do všech směrů. Spekulární světlo (též lesklé světlo) udává intenzitu světla odraženého od povrchu tělesa v jednom směru podle zákona odrazu.

## 2.4.2 Detekce a zvýraznění hran

Na detekci hran existuje celá řada algoritmů. V dané práci jsou popsány obvyklé techniky na detekci hran. První dvě popisované techniky nepoužívají detekci hran z obrazu.

V první technice je nejprve provedeno vykreslení černého drátěného modelu a přes něj je běžným způsobem vykreslen model [1]. Tím vznikne černý obrys kolem modelu. Danou techniku je možné provést na jeden průchod shaderem, je rychlá a poměrně jednoduchá. Vzniklý obrys je rovnoměrný, zdůrazňuje okraje objektu, ale vnitřní geometrie je však zanedbána.

Další možností zvýraznění je ztmavení vykreslovaných trojúhelníků, které jsou příliš odkloněny od směru pohledu [1]. Této technice opět stačí jeden průchod shaderem, její rychlost je rovna normálnímu vykreslení modelu. Vzniklý obrys zdůrazňuje jednak okraje modelu, ale také vnitřní geometrii. Obrys je však nerovnoměrný, protože jsou ztmavovány celé trojúhelníky. Ne vždy jsou trojúhelníky na hraně objektu natočeny ve správném úhlu. Tím na některých místech chybí zvýraznění obrysů a naopak na některých je zvýraznění příliš velké.

Třetí možností zvýraznění obrysů je za použití detekce hran z obrazu. Daná technika funguje na principu porovnání sousedních pixelů okolo zvoleného pixelu. Pokud jsou rozdíly mezi barvami pixelů příliš velké, jedná se o hranu a v tomto místě je provedeno zvýraznění. Výpočet rozdílu lze provést několika způsoby.

Jeden z možných způsobů zvýraznění porovnává protilehlé pixely, ze kterých vypočítá průměrnou odchylku barev. V rovnici (1) je uveden příklad pro výpočet  $\pm 1$  okolí okolo zvoleného pixelu (celkové okolí pixelu je tedy opět matice velikosti  $3 \times 3$ ) [2].

$$\begin{matrix} c_0 & c_1 & c_2 \\ c_3 & c_4 & c_5 \\ c_6 & c_7 & c_8 \end{matrix} \Rightarrow \frac{|c_0 - c_8| + |c_1 - c_7| + |c_2 - c_6| + |c_3 - c_5|}{4} \quad (1)$$

Přičemž  $c$  představuje barvu zvoleného pixelu a  $i$  je index pixelu. Vypočítaná hodnota je poté porovnána s vhodně zvoleným limitem. Pokud hodnota spadá do limitu, jde o hranu. Tato metoda produkuje dobrý obrys, který zdůrazňuje jak vnější obrys, tak vnitřní geometrii. Na vnitřní geometrii se však obrys na některých místech částečně ztrácí a je nerovnoměrný.

Poslední zde zmiňovaný způsob detekce hran nejprve prohledává okolí pixelu a nalezne pixely s nejsvětější a nejtmavší hodnotou barvy. Odečtením minima od maxima je získán jejich rozdíl. Ten je poté porovnán s vhodně zvoleným limitem. Pokud rozdíl překračuje limit, jedná se o hranu. Obrys získaný touto technikou je spojitý a zdůrazňuje obrys i geometrii objektu.

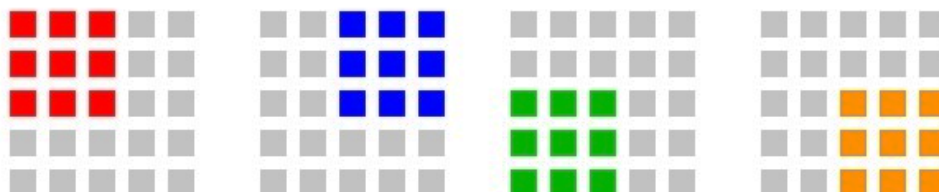
## 2.4.3 Vyhlazení textur

Pro účely vyhlazení (roztření) textur je v práci použit Gassův a Kuwaharův filtr. Principem Gausovy metody je výpočet průměrné hodnoty barvy pro daný pixel a jeho okolí a tato hodnota je poté dosazena na místo původní barvy pixelu. Čím větší okolí je zahrnuto do výpočtu, tím je dosaženo většího rozmazání. Například pro okolí  $\pm 1$  okolo zvoleného pixelu vznikne čtvercová matice o rozměrech  $3 \times 3$ , viz rovnice (2).

$$\left( \sum_{i=0}^{n-1} c_i \right) / n \Rightarrow \left( \sum_{i=0}^{(3 \times 3)-1} c_i \right) / (3 \times 3) \Rightarrow \begin{matrix} c_0 & c_1 & c_2 \\ c_3 & c_4 & c_5 \\ c_6 & c_7 & c_8 \end{matrix} / (3 \times 3) = (c_0 + c_1 + \dots + c_8) / (3 \times 3) \quad (2)$$

Kde parametr  $\mathbf{c}$  představuje barvu zvoleného pixelu,  $i$  je index pixelu a  $\mathbf{n}$  je strana matice umocněna na druhou. Daným výpočtem se získá výsledná průměrná barva zvoleného pixelu (vždy ve středu matice), která se dosadí na místo původní barvy pixelu. Tím se získá rozmazaná textura, která je vhodná pro aplikaci rozložení intenzity osvětlení.

Kuwaharův filtr je adaptivní, hrany zachovávající, vyhlazovací filtr [7]. Okolo pixelu je vhodně zvoleno okolí, čímž vznikne čtvercová matice pixelů. Ta je rozdělena do 4 stejně velkých pod-částí (obr. 12) a poté je vybrána část s nejmenší odchylkou (s nejnižším rozpětím hodnot). Z vybrané oblasti je poté vrácena průměrná hodnota barvy. Tato technika nezvládá příliš mnoho náhodného šumu v obraze. Proto je vhodné aplikovat nejprve jednoduché Gausovo rozostření na vykreslený objekt a poté použít danou metodu.



Obr. 12 Rozdělení okolí pixelu do 4 částí podle Kuwaharova filtru

#### 2.4.4 Stupně šedi

Pro převod barevné textury do odstínu šedi je pro každý pixel vypočítána průměrná hodnota barvy pixelu, viz rovnice (3)[9].

$$0.2126c_r + 0.7152c_g + 0.0722c_b \quad (3)$$

Kde  $\mathbf{c}_r$  je červená,  $\mathbf{c}_g$  je zelená a  $\mathbf{c}_b$  je modrá složka barvy převáděného pixelu. Výsledek výpočtu je poté aplikován zpět do všech složek barvy. Tím je pixel převeden do odpovídajícího odstínu šedi. Vzniká tak problém se ztrátou různých barevných oblastí, jak je patrné na obr. 7.

## 3. Navrhované techniky implementace

V této kapitole je probrán návrh aplikace, demonstrující výše zmíněné techniky. Jsou zde popsány použité jazyky, knihovny, frameworky, zdroje atd. Dále je zde podrobně probrán návrh implementace samotných technik. Níže popisované techniky jsou prováděny na grafické kartě pomocí aplikačního rozhraní *OpenGL*.

### 3.1 Návrh aplikace

Navrhovaná aplikace je implementovaná v jazyce *C++*, pro vykreslování ve 3D je použito rozhraní *OpenGL*. Aplikace dále využívá řadu knihoven, usnadňující načítání textur, modelů a dalších souborů. Samotná aplikace by měla zvládnout demonstrovat všechny zmíněné techniky a zároveň by měla mít také možnost mezi technikami přepínat, takže bude možné vizuálně jednotlivé techniky mezi sebou porovnat.

Pro vývoj aplikace je zvoleno vývojové prostředí *Visual Studio 2013*. Cílová architektura aplikace je 64 bitová. V tomto prostředí je vytvořen projekt s aplikací.

### 3.2 Knihovny

V aplikaci bude použita řada knihoven, které jsou potřebné pro funkčnost aplikace. Pro přístup k aplikačnímu rozhraní *OpenGL* je použita knihovna *GLEW*. Je to platformě nezávislá open source knihovna obalující *OpenGL* a jeho rozšíření. Poskytuje run-time mechanismus pro detekci podporovaných rozšíření *OpenGL* na dané platformě. Díky němu jsou všechny verze *OpenGL* a jeho rozšíření, fungující na dané platformě přístupná z jednoho hlavičkového souboru. *GLEW* však nezajišťuje žádnou funkcionalitu spojenou s vytvářením uživatelského okna či interakci s periferiemi počítače.

Z toho důvodu je použita knihovna *SDL 2*. Jedná se o multiplatformní knihovnu navrženou tak, aby poskytovala nízkoúrovňový přístup k audio, vstupním periferiím (klávesnice, myš, apod.) a grafickému hardwaru prostřednictvím *OpenGL*. *SDL* je napsáno v jazyce *C* a je nativně fungující v *C++*. Právě pro tyto vlastnosti byla zvolena *SDL 2* knihovna pro vytváření oken projektu, práci s myší a klávesnicí. Dále jsou potřebné knihovny pro zpracování souborů s modely a texturami. Pro tyto účely jsou zvoleny knihovny *FreeImage* a *Assimp*. *FreeImage* je multiplatformní, open-sourcovou knihovna pro načítání různých formátů obrázků. Umožňuje jednoduché načítání obrázkových souborů ve všech populárních formátech. Pro načítání modelů je použita knihovna *Assimp*. Ta je určena pro načítání 3D modelů z různých populárních formátů. Dále poskytuje prostředky pro další úpravu načtených dat a celkově usnadňuje práci s modely. Poslední použitou knihovnou je *GLM*. Je to platformě nezávislá matematická knihovna.



## 4. Implementace

V této kapitole je probrán návrh technik napodobujících umělecké styly a použité techniky, implementační část a problémy spojené s danými technikami.

### 4.1 Tvorba aplikace

V první části tvorby aplikace bylo nutné zvolit cílovou platformu programu. Je zvoleno 64 bitové prostředí a to z důvodů, že není známa paměťová náročnost aplikace. Před samotnou implementací zvolených metod je zapotřebí vyřešit obecnou část implementace.

#### 4.1.1 Vykreslování s pomocí OpenGL

O zobrazení 3D objektů se v aplikaci stará aplikační rozhraní *OpenGL*, pomocí kterého jsou prováděna volání na grafické kartě. K urychlení grafických operací jsou náročnější operace prováděny na grafické kartě v programech. Ty se skládají z několika částí, jenž na sebe navazují. Těmto částem (podprogramům) se říká shadery. Veškeré shadery v aplikaci jsou napsány v jazyce *GLSL*. Jedná se o jazyk vytvořený pro potřeby *OpenGL*- na tvorbu programů běžících na grafické kartě. Jeho syntaxe vychází z jazyka *C* a přidává řadu matematických funkcí a struktur, usnadňující práci s grafikou.

Pro tuto práci je nejdůležitější fragment shader, který odpovídá za zpracování jednotlivých pixelů viditelných na obrazovce. Právě daný shader je pro danou práci nejpodstatnější. Veškeré efekty spojené se změnou barvy pixelů, osvětlení a další efekty jsou prováděny za pomoci tohoto shaderu.

Další nezbytnou součástí je framebuffer objekt. Základní framebuffer objekt tvoří defaultně *OpenGL* a umožňuje vykreslování na obrazovku. V té to práci však budou využity i uživatelské framebuffer objekty. K těm budou připojeny textury a další potřebné buffery, do nich bude prováděno vykreslení. Jednotlivé framebuffer objekty lze zřetězit. To je užitečné z důvodu, že ne vždy je možné provést všechny potřebné úpravy vykreslovaného obrazu v jednom průchodu. Například v prvním průchodu je vykreslována celá scéna do textury připojené k framebuffer objektu. V dalším průchodu je tato textura dále zpracována a výsledek bude vykreslen do základního framebuffer objektu na obrazovku. Tento proces je též známý jako post-processing.

## 4.2 Návrh implementace technik

Pro všechny vybrané techniky je nejprve proveden návrh implementace podle nastudované literatury. Z návrhu poté vychází implementace a případné změny.

### 4.2.1 Cell shading

Je to již známá technika vytvořená pro 3D aplikace a programy, tak aby napodobovala komiksový styl. Ze samotného rozdělení barev u této techniky je poměrně viditelné, že jde o speciální případ osvětlovacího modelu. Zvýraznění okrajů lze vytvořit přímo během vykreslení objektu anebo jako dodatečný efekt detekce obrazu. Pro účely této práce je rozhodnuto, že zvýraznění hran bude provedeno pomocí detekce hran po vykreslení objektu.

Vykreslení je provedeno tak, že v prvním průchodu shaderem jsou postupně vykresleny jednotlivé objekty na scénu. Pro osvětlení objektů je použit Phongův osvětlovací model. Na difuzní složku osvětlení je poté aplikována funkce mapující spojitou intenzitu osvětlení do několika oblastí. Tím bude osvětlení vizuálně rozděleno do několika oblastí. V poslední fázi bude provedeno zvýraznění hran.

### 4.2.2 Skica tvořená tužkou

Tato technika se snaží napodobit tužkou kreslenou skicu při zobrazování objektů ve 3D. Pro její implementaci existuje více metod. Z důvodu, že jsou všechny skici kresleny tužkou na podklad, je patrné, že výsledný obraz bude v odstínech šedi. Dále by bylo vhodné, aby ve výsledném obrazu byly viditelné stopy tužky. A to jak na místech obrysů v objektu, tak v modelu pro vytvoření efektu stínování [6]. Na méně osvětlených místech bude větší hustota čar, které budou tmavší a opačně. Dále je patrné, že v obrazu nemůžou být plně černá místa. Je tomu tak, protože nejtmaší odstín tužky je šedý, nikoliv černý.

Tato technika opět používá Phongovo osvětlení. Pro vykreslení objektu jsou použity namísto původních textur modelu nové textury s čarami vytvořenými tužkou. Tato textura je ručně vytvořena před spuštěním aplikace. Ve výsledném obraze však vznikají čistě černá místa. Na ty je aplikována hustá textura tužky. Ta je opět vytvořena již před spuštěním aplikace. Vykreslená scéna již působí jako nakreslená tužkou. Chybí však poslední krok, pro skicu typické obtažení okrajů tužkou. To je provedeno jednou z již zmíněných technik pro detekci hran. S její pomocí je provedeno zvýraznění obrysu. Namísto černé barvy bude opět použita textura silně vybarvená tužkou. Technika by tím měla být zcela napodobena.

### 4.2.3 Akvarelová malba

Tato technika se snaží napodobit techniku kresby akvarelem. Pro implementaci na počítači opět neexistuje jedna ustálená implementace. Pro danou techniku je typická kolísavost barevných oblastí a oddělené předěly barev. Dále je pro ni podstatná viditelnost pigmentu podkladu, jenž částečně prosvítá skrze barvu.

Pro osvětlení objektů ve vykreslené scéně je opět použit Phongův osvětlovací model. Osvětlení je rozděleno do několika oblastí. Oproti cell shadingu však v obraze nebudou čistě tmavé oblasti. Namísto nich je tmavá barva. Dále je rozhodnuto, že bude provedeno napodobení rozdělení barev v textuře do barevných oblastí, jak je tomu v literatuře [3]. Poté je do výsledného obrazu nutné dodat efekt barvy na podkladu (v tomto případě papíru), na který je objekt vykreslen podle literatury [4].

Textura papíru je dopředu ručně vytvořena a vykreslena přes celou scénu včetně vykreslovaného objektu.

### 4.3 Implementace aplikace

Výsledná aplikace je implementována jako nekonečná smyčka. V té je řešeno ovládání a vykreslování.

Aplikace je řešena převážně objektově. Nejdůležitější částí aplikace je interface *Renderer*, jenž má jednu virtuální metodu *render(RenderingContext&)*. V této metodě je realizováno veškeré vykreslování na obrazovku a transformace s ním spojené. Její parametr *RenderingContext* je kontext aplikace, v němž jsou předány nejdůležitější a často se měnící data potřebná pro vykreslování. Z interface *Renderer* dědí ostatní vykreslující třídy, představující jednotlivé techniky či jejich varianty. Hlavní třídou je *SceneRenderer*, jenž dědí z třídy *Renderer* a zapouzdřuje v sobě ostatní vykreslující třídy. V této třídě je volána metoda *renderScene()*, v níž je vytvořen kontext aplikace, jsou v ní provedeny základní transformace a poté je z ní zavoláno vykreslení aktivní třídy *Renderer*, představující danou techniku.

### 4.4 Implementace technik

Jednotlivé techniky jsou implementovány podle zde uvedeného pořadí. První implementovaná technika je cell shading a z technik v ní použitých se dále vycházelo.

#### 4.4.1 Implementace cell shadingu

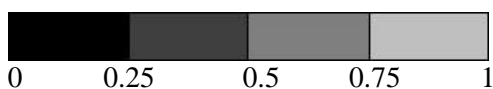
Daná technika je poměrně snadno implementovatelná a slouží jako základ pro další techniky. Proto v práci vznikly tři její varianty. Tyto varianty postupně vznikly jako vylepšení předchozí verze, z důvodu nevyhovujícího vzhledu. Techniky se mezi sebou liší jednak způsobem osvětlení, úpravou textur, ale také zvýrazněním okrajů. Všechny však používají Phongovo osvětlování.

První varianta cell shadingu ještě nepočítala s rozmazáním textur, zvýraznění hran bylo provedeno zčernáním vykreslovaných trojúhelníků na okraji. Pro osvětlení objektů ve scéně je použita difuzní a spekulární složka osvětlení. Ambientní složka bude vynechána, protože výsledný obraz výrazně neovlivňovala. Je na něj použito rozdělení osvětlení do oblastí, jak bylo popsáno v kapitole 3.1 a to podle funkce z algoritmu 1.

```
if (intensity < 0.25)    return 0;  
else if (intensity < 0.5)  return 0.25;  
else if (intensity < 0.75) return 0.5;  
else                    return 0.75;
```

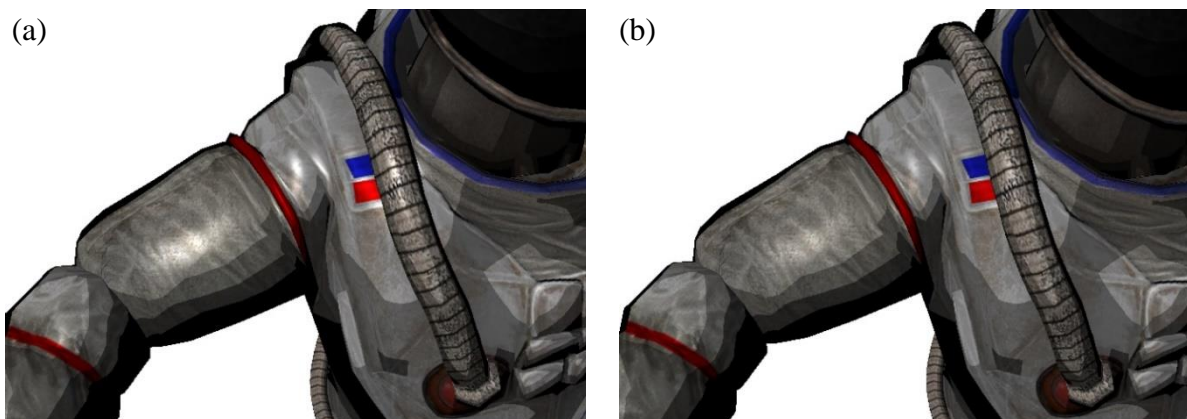
*Algoritmus 1 Rozdělení oblastí osvětlení pro první variantu cell shadingu*

Rozdělení je v intervalu od nuly do jedné, kde nula představuje nejtmaší (černou barvu) a jedna představuje nejsvětější osvětlená místa (bílá barva). Toto rozdělení osvětlení si lze představit jako jednorozměrnou texturu, viz obr. 13.



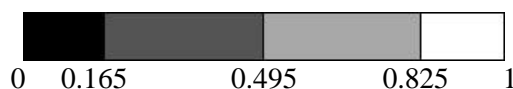
Obr 13. Rozložení osvětlení v první variantě cell shadingu

Vstupem funkce je intenzita osvětlení daného bodu a výsledkem je tedy rozdělení osvětlení do oblastí namísto plynulého přechodu. Jelikož jsou vrácené hodnoty vždy tmavší, než v původních oblastech, dojde i k celkovému ztmavení obrazu. Vykreslení obrazu je provedeno na jeden průchod. Výsledný obraz už působí jako cell shading, ale ztmavení nepůsobilo podle představ. Navíc spekulární složka osvětlení ve spojení s cell shadingem působila poněkud nepatřičně (obr. 14a) a bylo jasné, že bude muset být přitlumena. Nejjednodušším řešením bylo konečnou hodnotu ještě vydělit dvěma, čímž spekulární složka byla utlumena a vizuálně již zapadala do vykreslovaného obrazu, viz obr. 14b.



Obr. 14 První varianta cell shadingu: (a) s normálním spekulárním osvětlením a (b) s polovičním spekulárním osvětlením

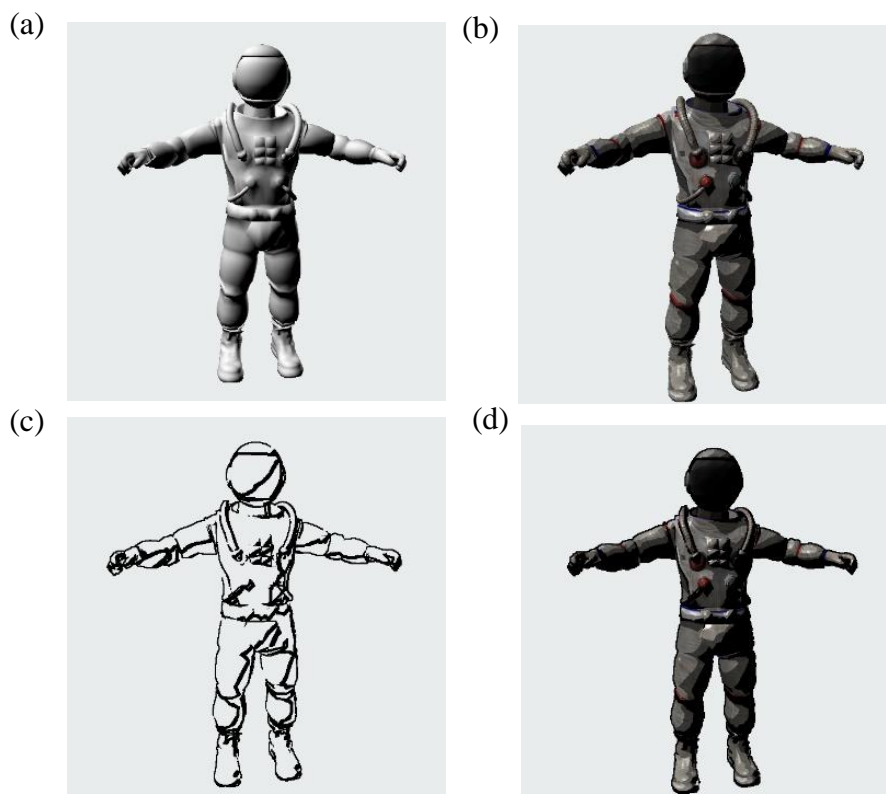
Výsledný obraz však nevypadal dostatečně komiksově. Textura byla příliš detailní, osvětlení v obraze bylo příliš tmavé a okraj byly nerovnoměrné. Proto byla vytvořena další varianta. V té bylo vykreslení rozděleno na dva průchody. V prvním průchodu je vykreslování prováděno do vytvořeného framebufferu, do dvou připojených textur. Ty mají stejnou velikost jako obrazovka okna. Nejprve je v tomto průchodu provedeno Gausovo rozostření pro každý vykreslený pixel originální textury modelu. Tím jsou z textury odstraněny detaily. Osvětlení je opět Phongovo, jsou použity difuzní a spekulární části osvětlení. Osvětlení je rozděleno na oblasti podle množiny intenzity světla (0.0; 0.33; 0.66; 1), viz obr. 15. Z intervalu, do kterého spadá vypočtené osvětlení, je vrácena hodnota, ke které má tato hodnota blíže. Pro příklad, pokud je pro daný bod vypočtena hodnota 0.2, tak tato hodnota spadne do intervalu 0.0 až 0.33. Vracena je hodnota 0.33, protože 0.2 je blíže 0.33 než nule. Tato hodnota je zapsána jako barva do první textury. Poté jsou s touto hodnotou opět pronásobeny difuzní a spekulární části osvětlení. Výsledný obraz je zapsán do druhé textury. Následuje další průchod, v němž je nejprve vykreslena přes celou obrazovku druhá textura s cell shadingem předchozího průchodu shaderu. Poté je provedena detekce hran nad první texturou (s difuzním osvětlením) a pokud je na daném místě hrana, je zvýrazněna černě.



Obr. 15 Rozložení osvětlení v druhé variantě cell shadingu

Celý tento postup je popsán na obr. 16. V prvním průchodu je vytvořena textura s difuzním osvětlením (obr. 16a) a cell shadingovým osvětlením modelu (obr. 16b). V dalším průchodu je provedena detekce hran nad difuzní texturou z předchozího průchodu, viz obr. 16c. Přes celou scénu

je vykreslena textura s cell shadingovým osvětlením z předchozího průchodu. Na místech, kde jsou detekovány hrany, jsou zvýrazněny okraje, jak je patrné z obr. 16d.



*Obr. 16 Vykreslení cell shadingu: (a) difuzní textury, (b) textury s modelem s použitím cell shadingu, (c) zvýraznění hran a (d) výsledek*

Výsledný obraz (obr. 17) je již znatelně vizuálně lepší oproti původní technice. Ovšem okraje nejsou rovnoměrné a rozdělení osvětlení stále nepůsobí dostatečně komiksově.



*Obr. 17 Druhá varianta cell shadingu*

Z toho důvodu je vytvořena poslední varianta této techniky. Je provedena opět na dva průchody. Rozdělení oblastí osvětlení je zobrazeno v obr. 18 a je dáno algoritmem 2.



Obr. 18 Rozložení intezity osvětlení ve finální variantě cell shadingu

```
if (intensity > 0.85)      return 1;  
else if (intensity > 0.5)  return 0.7;  
else if (intensity > 0.15) return 0.35;  
else                       return 0;
```

Algoritmus 2 Rozdělení intezity osvětlení do oblastí

Detekce hran se provádí tak, že v prvním průchodu byly do jedné textury zapsány souřadnice bodu namísto barvy. Tím vznikla poziční textura, na které byla v dalším průchodu provedena detekce hran, přičemž je brána pouze z-tová souřadnice pozice. Jde tedy o vzdálenost vykreslených bodů od kamery. Díky detekci vycházející z rozdílů vzdáleností je výsledná detekce hran mnohonásobně přesnější než původní detekce z obrazu, viz obr. 19. Je však nutné opět vhodně zvolit limit, na němž je rozhodnuto, zda jde o hranu, či nikoli. Tento limit se přímo určuje, zda bude vnitřní geometrie zvýrazněna, či nikoliv.

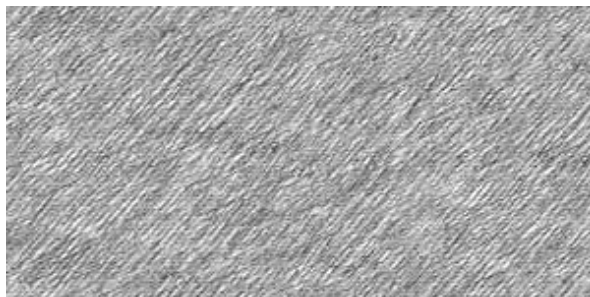


Obr. 19 Finální verze cell shadingu

#### 4.4.2 Implementace pencil sketch shadingu

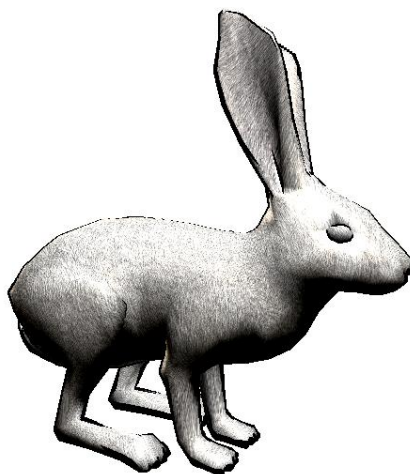
Implementace této techniky vychází ze znalostí získaných z cell shadingu. Vznikla pouze jedna varianta používající Phongovo osvětlení.

Nejprve bylo nutné vytvořit texturu čar tužky. Čáry v textuře musely směřovat jedním směrem, ale zároveň musely působit lehce chaotickým dojmem (obr. 20). Touto texturou jsou nahrazeny všechny difuzní textury vykreslovaných objektů.



*Obr. 20 Textura čar tužkou*

Vykreslení na obrazovku je provedeno ve dvou krocích. V prvním kroku jsou vykresleny jednotlivé modely na obrazovku, na stínování je použito Phongovo stínování. Pouze pro difuzní část stínování je použita nahrazující textura. Zbývající dvě části stínování používají původní texturu modelu. V následujícím kroku je provedena detekce hran a ty jsou zvýrazněny, viz obr. 21.



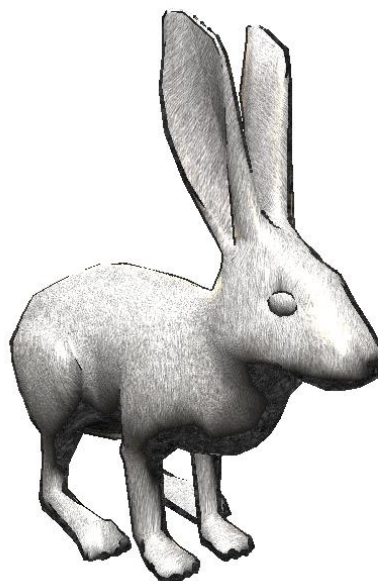
*Obr. 21 Počáteční verze pencil sketch shadingu*

Jak je patrné z obr. 15, černé oblasti působí rušivým dojmem a budou muset být nahrazeny také texturou tužky pro tmavší oblasti. Tato textura měla zajistit dojem tmavě vybarvených oblastí, kdy na tužku bylo silně tlačeno. Tmavost textury musela být několikrát měněna tak, aby ladila s celkovým obrazem. Výsledná textura je znázorněna na obr. 22.



*Obr. 22 Ukázka tmavé textury*

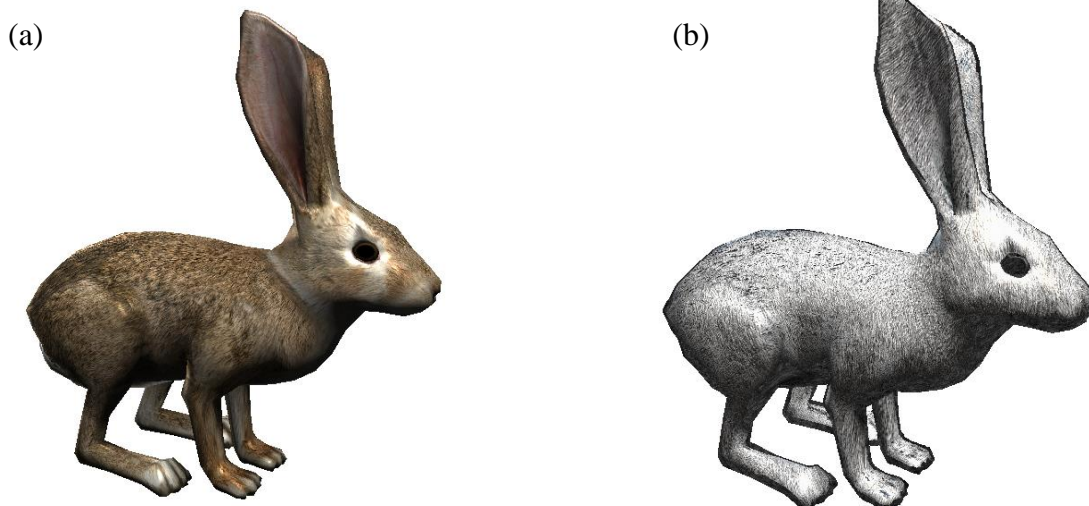
Danou texturou jsou v prvním i druhém kroku nahrazeny všechny černé oblasti a to včetně zvýrazněných okrajů. Výsledný obraz již působil dojmem skici, viz obr 23.



*Obr. 23 Výsledná verze pencil sketch shadingu*

Jak je však patrné z obr. 23, původní textura vykreslených modelů byla naprosto ztracena. Proto bylo rozhodnuto, že se provede pokus o přidání původní textury. Původní textura byla nejprve převedena do odpovídající reprezentace v odstínech šedi. Poté byla zesvětlena a bylo na ni aplikováno difuzní osvětlení. To bylo provedeno v prvním průchodu a výsledná textura byla přidána k výsledné textuře s tužkou. Výsledek již obsahoval jak čáry tužkou, tak i části původní textury (obr. 24).





*Obr. 24 Porovnání technik: (a) Phongovo stínování a (b) pencil sketch shadingu*

Během všech popisovaných úprav bylo několikrát nutné upravit kontrast textur čar, tak aby textury vzájemně ladily svými odstíny. Výslednou techniku lze aplikovat na celou řadu modelů, jejími limity jsou však příliš tmavé textury modelů. U nich je na větší část zobrazeného objektu aplikována tmavá textura tužky. Tím bude většina modelů příliš tmavá a oblasti a geometrie modelů budou špatně rozlišitelné.

#### **4.4.3 Implementace watercolor shadingu**

Daná technika vychází z nastudované literatury [5, 6]. Na osvětlení ve scéně je použit cell shading. V práci byla implementována pouze jedna varianta této techniky.

V prvním kroku implementace techniky bylo nutné vytvořit texturu papíru s viditelným pigmentem (drsňým povrchem), viz obr. 25. Ta bude v druhém kroku aplikována přes celou vykreslenou scénu. Textura je ručně vytvořena před spuštěním aplikace.



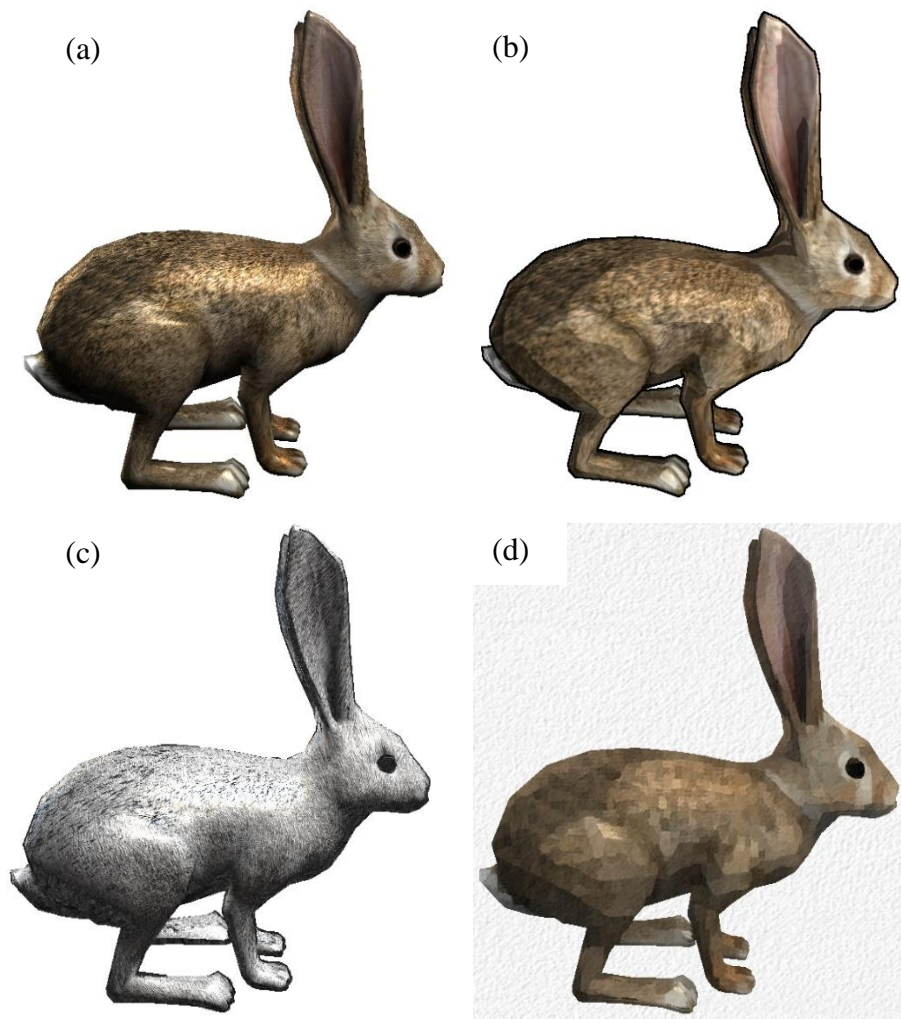
*Obr. 25 Textura papíru*

Implementace techniky je provedena dvěma průchody. V prvním průchodu je nejprve provedeno Gausovo rozostření nad texturami vykreslovaných objektů. Poté je objekt vykreslen s použitím cell shadingu bez zvýraznění hran. Pro nejméně zastíněné oblasti daná funkce vrací tmavý odstín, nikoli černou. V dalším kroku bylo nad vstupním obrazem provedeno Kuwaharovo rozostření a poté je přes obraz aplikován pigment papíru. Výsledný obraz působí lehce rozpitým dojmem akvarelu, jak je patrné na obr. 26.



Obr. 26 Ukázka akvarelu

#### 4.5. Výsledné implementace



Obr. 27 Porovnání technik: (a) Phongův osvětlovací model, (b) cell shading, (c) pencil sketch shading a (d) watercolor shading

## 5. Závěr

Cílem této práce bylo prostudovat nerealistické umělecké zobrazování, konkrétně techniky cell shading, pencil sketch shading a watercolor shading. Dále pak navrhnout implementaci těchto metod a implementovat aplikaci s použitím aplikačního rozhraní *OpenGL*. Poté z aplikace vytvořit demonstrační video.

Tohoto cíle bylo dosaženo. Všechny zmíněné techniky se povedlo implementovat a napodobit. Implementované techniky vycházejí z technik používaných k realistickému zobrazování a dále je rozšiřují. Většina těchto technik je však značně závislá na post-procesingu a úpravách prováděných až po vykreslení scény.

Konečná implementace cell shadingu se drží poměrně dobře definice této metody. K možným zlepšením postupu vykreslování techniky by mohlo být přidání částečně gradientní přechodů mezi osvětlenými oblastmi. Dále by mohlo být zlepšeno zvýraznění okrajů. Ideálně provedené zvýraznění by bylo na jeden průchod. Možným řešením tohoto problému by bylo předělání originální textury, aby zvýraznění oddělených oblastí bylo přímo v ní. Stylizace textur modelu by výrazně pomohla celkovému vzhledu a tím by ubyla i nutnost aplikování Gausova rozostření. Avšak tato stylizace jde za hranici úprav popisovaných v této práci. Přesto je výsledná implementace této metody poměrně zdařilá a bylo by ji možné použít i v dalších aplikacích.

Výsledek implementace pencil sketch shadingu působí poměrně dobrým dojmem. Přestože nepůsobí zcela dojmem skici, má všechny její znaky a náležitosti. Možným zlepšením by bylo vytvoření ještě lepších nahrazujících textur tužky, tak aby výsledek působil ještě více dojmem kreslení tužkou. Případně použít namísto předvytvořené nahrazující textury tužky texturu vytvářenou za běhu aplikace podle osvětlení. Nejlepší variantou by pak bylo vytvořit tyto textury na základě osvětlení a původní textury. Dále by mohly být zlepšeny tužkou zvýrazněné okraje, a to tak že budou zcela kolem vykreslených modelů. Šíře těchto okrajů se může měnit, ale nikdy by se neměly zcela vytrácet. Ideální metoda by měla vytvořit takové okraje, aby působily jako několik čar nakreslených přes sebe.

Watercolor shading je také dobře napodoben. Oproti ostatním metodám se výsledný vzhled metody nejvíce odchýlil od nastudovaných metod. Výsledný vzhled je však dobrý. Možným zlepšením by bylo větší rozptýlení pixelů okolo přechodů barev. K tomu by se dala například využít zmiňovaná detekce hran na pomezí přechodů na cell shadingovém osvětlení objektů ve scéně. Tato umělecká technika je ovšem velice svobodná a to by potenciálně vedlo k nepřehlednému množství implementací této techniky. Proto je třeba brát výslednou implementaci jako jednu ukázkou z potenciálně nekonečné množiny technik.

Výsledkem této práce je demonstrační aplikace s jednotlivými technikami uměleckého nerealistického zobrazování a z ní vytvořené video. Získané informace a implementace lze aplikovat v řadě grafických aplikací.

# Použité zdroje

- [1] HUTCHINS, Adam a Sean KIM. *Advanced Real Time Cel Shading Techniques in OpenGL* [online], s. 1-15 [cit. 2016-04-24]. Dostupné z:  
[http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S12/final\\_projects/hutchins\\_kim.pdf](http://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S12/final_projects/hutchins_kim.pdf)
- [2] Edge detection Pixel Shader. *Coding experiments* [online]. 2010 [cit. 2016-04-24]. Dostupné z: <http://coding-experiments.blogspot.cz/2010/06/edge-detection.html>
- [3] CHEN, Jiajian, Greg TURK a Blair MACINTYRE. Watercolor inspired non-photorealistic rendering for augmented reality. In: *Proceedings of the 2008 ACM symposium on Virtual reality software and technology - VRST '08*. New York, New York, USA: ACM Press, 2008, s. 231. DOI: 10.1145/1450579.1450629.
- [4] HOARE, Charlotte. *Interactive non-photorealistic rendering using GLSL*[online]. 2012, s.1-13 [cit. 2016-04-25]. Dostupné z:  
[https://nccastaff.bournemouth.ac.uk/jmacey/MastersProjects/MSc12/Hoare/CharlotteHoare\\_Thesis.pdf](https://nccastaff.bournemouth.ac.uk/jmacey/MastersProjects/MSc12/Hoare/CharlotteHoare_Thesis.pdf)
- [5] MOHR, Alex a Michael GLEICHER. HijackGL. In: *Proceedings of the second international symposium on Non-photorealistic animation and rendering - NPAR '02* [online]. New York, New York, USA: ACM Press, 2002, s. 13- [cit. 2016-05-13]. DOI: 10.1145/508530.508533. ISBN 1581134940. Dostupné z: <http://portal.acm.org/citation.cfm?doid=508530.508533>
- [6] MOHR, Alex. *NPRQuake* [online]. Wisconsin, 2000 [cit. 2016-04-30]. Dostupné z: <http://research.cs.wisc.edu/graphics/Gallery/NPRQuake/>
- [7] HALL, Matt. Smooth operator: Smoothing seismic interpretations and attributes. *The Leading Edge*. 2007, 26(1), 16-20. DOI: 10.1190/1.2431821.
- [8] File:Una akvarel.jpg - Wikimedia Commons. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-12]. Dostupné z: [https://commons.wikimedia.org/wiki/File:Una\\_akvarel.jpg](https://commons.wikimedia.org/wiki/File:Una_akvarel.jpg)
- [9] STOKES, Michael, Matthew ANDERSON, Srinivasan CHANDRASEKAR a Ricardo MOTTA. *A Standard Default Color Space for the Internet — sRGB* [online]. [cit. 2016-05-12]. Dostupné z: [www.w3.org/Graphics/Color/sRGB](http://www.w3.org/Graphics/Color/sRGB)

# Seznam příloh

## Příloha A: Popis spuštění a ovládání aplikace

### Obsah DVD

- Zdrojové kódy implementované aplikace v adresáři /src/
- Shadery použité v aplikaci v adresáři /src/models/
- Modely použité v aplikaci v adresáři /src/shaders/
- Demonstrační video z aplikace v adresáři /video/
- Tato práce ve formátu PDF v adresáři /thesis/
- Zdrojové kódy této práce ve formátu systému Microsoft Word v adresáři /thesis/word/

# Příloha A

## Popis spuštění a ovládání aplikace

Zdrojové soubory aplikace se nachází v adresáři **/src/**.

### A.1 Překlad aplikace

Aplikace je implementována ve vývojovém prostředí Visual Studio 2013 v jazyce C++. Adresář **/src/** obsahuje soubor *ArtisticSceneVisualization.sln*, jenž otevře projekt s aplikací ve Visual Studiu. Poté stačí přeložit celý projekt nebo jen pod-projekt Source. Přeložený spustitelný soubor se nachází v adresáři **/src/bin/**. Spustitelný soubor se jmenuje Source.exe.

### A.2 Spuštění aplikace

Pro spuštění aplikace je doporučeno mít aktualizované ovladače grafické karty. Aplikaci lze spustit s několika vstupními parametry:

- 1) 's' rozlišení okna aplikace bude 1280x720
- 2) 'b' rozlišení okna aplikace bude 1600x900
- 3) 'm:cesta\_k\_modelu' spustí aplikaci s modelem načteným se zadané cesty

Mnou testované modely byly ve formátu .obj s texturami ve formátu .png. Načítání ostatních formátů modelů nebylo testováno. Příklad spuštění aplikace:

*Source.exe s "m:../models/Astronaut/Astronaut.obj"*

V případě, že aplikace je spuštěna bez parametru, je načten defaultní model. Ten se nachází v adresáři **/src/models/** a jde o model králíka (složka **/src/models/ Rabbit/**). Tento soubor s modelem se nedoporučuje upravovat, ani mazat. Samotná aplikace je bez parametrů spuštěna v rozlišení 800x600.

### A.3 Ovládání aplikace

Aplikace používá FPS kameru, tedy pro pohyb je použita klávesnice a pro rozhlížení myš. Popis ovládání je zde:

- 1) 'w' pohyb vpřed
- 2) 's' pohyb vzat
- 3) 'a' pohyb doleva
- 4) 'd' pohyb doprava
- 5) 'mezerník' zapne/vypne uzamčení kurzoru myši k oknu aplikace
- 6) 'esc' konec aplikace
- 7) 'F1' vykreslení pomocí Phongova osvětlení
- 8) 'F2 až F4' vykreslení pomocí cell shadingu
- 9) 'F5' vykreslení pomocí pencil sketch shadingu
- 10) 'F6' vykreslení pomocí watercolor shadingu
- 11) 'pohyb kolečka myši' změna zorného pole