

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

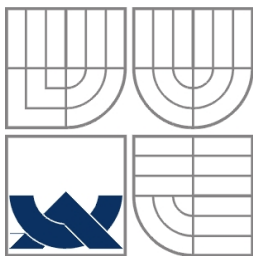
ANALÝZA SCÉNY ZALOŽENÁ NA 2D OBRAZECH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

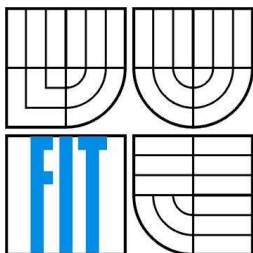
AUTOR PRÁCE
AUTHOR

BC. MARTIN HEJTMÁNEK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ANALÝZA SCÉNY ZALOŽENÁ NA 2D OBRAZECH

SCENE ANALYSIS BASED ON THE 2D IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. MARTIN HEJTMÁNEK

VEDOUCÍ PRÁCE
SUPERVISOR

PH.D. ING. FILIP ORSÁG

BRNO 2007

Zadání diplomové práce

Řešitel: **Hejtmánek Martin, Bc.**

Obor: Inteligentní systémy

Téma: **Analýza scény založená na 2D obrazech**

Kategorie: Umělá inteligence

Pokyny:

1. Prostudujte problematiku zpracování a analýzy 2D obrazů.
2. Navrhněte vlastní algoritmus pro analýzu tvaru 3D povrchů z 2D obrazů scény založený na analýze osvětlení povrchu. Můžete se inspirovat v existujících algoritmech.
3. Implementujte navržený algoritmus a otestujte ho na jednoduchých 3D objektech s homogenním povrchem. Srovnajte kvalitu výstupu a rychlost s existujícími algoritmy.
4. Provéřte algoritmus na složitější scéně, texturovaných objektech, jednoduchých reálných scénách.
5. Zhodnoťte dosažené výsledky a navrhněte možná vylepšení algoritmu.

Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1-2

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Orság Filip, Ing., Ph.D., UITS FIT VUT**

Datum zadání: 28. února 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Ústav inteligentních systémů

612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Bc. Martin Hejtmánek**
Id studenta: 49329
Bytem: Jedlová 406, 569 91 Jedlová u Poličky
Narozen: 13. 12. 1982, Polička
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Analýza scény založená na 2D obrazech
Vedoucí/školicel VŠKP: Orság Filip, Ing., Ph.D.
Ústav: Ústav inteligentních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.


Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel


.....

Autor

Abstrakt

Tato práce se zabývá analýzou povrchu objektů v jednoduché scéně reprezentované dvourozměrným obrazem. Shrnuje běžně používané metody v tomto oboru informačních technologií a popisuje jejich výhody a nevýhody. Na základě získaných znalostí a zkušeností představuje vlastní návrh algoritmu pro analýzu povrchu objektů založený na světelné informaci. Obsahuje podrobný popis algoritmu implementovaného v rámci této práce a diskutuje výsledky provedených experimentů. Na základě zkušeností s implementovaným algoritmem navrhuje možné předpoklady pro jeho další vylepšení.

Klíčová slova

Počítačové vidění, analýza, metody, rastr, vektor, osvětlení, textura, neuronová síť, mřížka, algoritmus, klasifikace, obraz, segmentace, předzpracování, návrh, implementace, scéna, sekvence, ROAM, zakřivení, deformace

Abstract

This thesis deals with an object surface analysis in a simple scene represented by two-dimensional raster image. It summarizes the most common methods used within this branch of information technology and explains both their advantages and drawbacks. It introduces the design of an surface profile analysis algorithm based on the lighting analysis using knowledge and experiences from previous work. It contains a detailed description of the implemented algorithm and discusses the experimental results. It also brings up options for the possible enhancement of the projected algorithm.

Keywords

Computer vision, analysis, methods, raster, vector, lighting, texture, neural network, grid, algorithm, classification, image, segmentation, preprocessing, design, implementation, scene, sequence, ROAM, deformation

Citace

Martin Hejtmánek: Analýza scény založená na 2D obrazech, **diplomová práce, Brno, FIT VUT v Brně, 2007**

Analýza scény založená na 2D obrazech

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením

Ing. Orsága Filipa, Ph.D., UITŠ FIT VUT

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Martin Hejtmánek
10. 5. 2007

Poděkování

Děkuji svému vedoucímu práce Ing. Filipu Orságovi, Ph.D. za podporu a konzultace během vytváření této práce.

© Martin Hejtmánek, 2007.

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

Obsah	1
1 Úvod.....	3
2 Stávající postupy analýzy obrazu.....	5
2.1 Klasifikace metod.....	5
2.2 Řetězec zpracování obrazu.....	8
2.3 Rastrové metody.....	10
2.3.1 Optimalizace rastrových metod	11
2.3.2 Segmentace obrazu	13
2.3.3 Analýza hran	14
2.3.4 Význačné body v obraze.....	14
2.3.5 Houghova transformace	15
2.3.6 Optický tok.....	16
2.3.7 Analýza textury.....	17
2.4 Vektorové metody	18
2.4.1 Analýza mřížky.....	18
2.4.2 Analýza osvětlení.....	19
2.4.3 Hranové modely.....	19
2.4.4 Optimalizace vektorových metod	20
2.5 Kombinace metod	20
3 Návrh algoritmu	22
3.1 Segmentace obrazu.....	22
3.2 Analýza zakřivení povrchu.....	23
3.3 Analýza textury	24
3.3.1 Mřížka segmentace pro rastrové operace.....	25
3.4 Geometrie scény a objektů	26
3.4.1 Jednorozměrné geometrické útvary	27
3.4.2 Dvourozměrné geometrické útvary.....	28
3.4.3 Vícerozměrné geometrické útvary	29
3.5 Klasifikace objektů.....	29
3.6 Souhrn systému analýzy.....	30
4 Implementace systému.....	32
4.1 Proces analýzy.....	33
4.2 Segmentace obrazu.....	35
4.2.1 Algoritmus ROAM	35

4.2.2	Adaptivní dělení mřížky	37
4.3	Analýza zakřivení povrchu.....	39
4.4	Separace objektů a detekce hran	42
4.5	Analýza geometrických vzorů.....	44
4.5.1	Detekce přímek	44
4.5.2	Detekce kružnic	46
4.6	Analýza textury	49
5	Experimenty	52
5.1	Syntetická scéna	53
5.2	Texturovaná scéna.....	54
5.3	Složitě texturovaná scéna	55
5.4	Reálná scéna.....	56
5.5	Umělá reálná scéna.....	57
5.6	Shrnutí experimentů a návrhy do budoucna	58
6	Závěr	60
	Literatura	61
	Seznam příloh	64

1 Úvod

Obor počítačového vidění je velmi rozsáhlý a proto není možné jeho komplexní problematiku zpracovat v rámci jediné publikace nebo dokonce v rámci jediné aplikace v praxi. Důvodů proč se tento obor v poslední době velmi rozvíjí a věnuje se mu nemalá pozornost, je mnoho. Na jedné straně stojí zábavní průmysl, kde analýza obrazu dává autorům do ruky mocnou zbraň pro použití na speciálních efektech filmů, optimalizaci záběrů a dalších činnostech, na straně druhé pak seriózní vědecké a průmyslové obory, kde je analýza obrazu klíčovým prvkem k získání důležitých informací. Mezi tyto obory patří v poslední době především lékařství, průmyslová výroba (validace kvality výrobků), dopravní systémy a v neposlední řadě i robotika, kde je počítačové vidění nezbytným základem pro vytvoření autonomních robotů orientujících se v reálném světě. Robotika pak více či méně obecně zasahuje do všech vědních disciplín a průmyslových oborů včetně výzkumu vesmíru. Nejen proto právě tento obor vyžaduje nejkomplexnější řešení analýzy obrazu. Těžko si lze představit, že stroj nasazený v nebezpečných nebo neobyvatelných prostorách bude autonomní, pokud nebude mít k dispozici alespoň hrubou představu o prostoru kolem sebe a objektech, které se v něm nacházejí.

Problémem většiny stávajících algoritmů počítačového vidění je to, že jsou navrženy a úspěšně pracují většinou pouze v laboratorních podmínkách s umělými scénami, v některých případech pak sice se scénami reálnými, ale předem známými. Právě proto nejsou tyto „běžné“ algoritmy dostatečně robustní pro použití v neznámém nebo pro danou metodu nevhodném prostředí. Nevhodnost existujících algoritmů spočívá především v tom, že každý z nich pracuje korektně pouze ve specifických případech a nejen proto, že mnoho z nich je umělými cestami a optimalizacemi dotazeno k maximu jejich možností, nenabízí mnoho cest pro vylepšení nebo aplikaci právě na reálnou scénu. Jistým řešením je v tomto případě kombinace několika různě orientovaných algoritmů, jejichž blok jako celek dokáže scénu zpracovat lépe než algoritmy samostatně, nicméně ani toho řešení nezvedá kvalitu zpracování obrazu na takovou úroveň, aby byly použitelné v obecném reálném prostředí. Hlavním nedostatkem je v tomto případě nutnost kvalitní segmentace a klasifikace scény a z výsledků vyplývající heuristika pro vyhodnocení vhodnosti aplikace jednotlivých částí systému na konkrétní oblasti scény. Právě vhodná a efektivní segmentace pro další zpracování je v předem neznámé scéně největší překážkou.

Cílem této práce je prozkoumat a zhodnotit možnosti jednotlivých algoritmů počítačového vidění a na základě získaných informací vytvořit pokud možno optimální návrh algoritmu, respektive komplexního systému, pro analýzu obrazu. Důraz bude kladen především na robustnost a možnost dalších úprav algoritmu pro zlepšení analýzy v konkrétním případě. Systém by měl být v ideálním případě schopen pracovat nezávisle na zkoumané scéně, jejím jasu a osvětlení, šumu a dalších negativních vlivech. Základní proces by měl používat jediný vstupní obraz, v případě potřeby a

nedostatečnosti informace nebo vysoké náročnosti scény by měl být rozšiřitelný pro použití více obrazů v sekvenci nebo stereovidění. Analýza bude založená na kvalitní segmentaci obrazu doplněnou o analýzu zakřivení povrchu na základě světelné informace. Práce jako celek je výzkumná, jejím cílem není vytvoření systému pro konkrétní aplikaci, ale především analýza možností a vytvoření vhodného teoretického i praktického základu pro možný budoucí systém počítačového vidění.

2 Stávající postupy analýzy obrazu

Zpracování obrazu je velmi širokým oborem, a přesto že se jednotlivé algoritmy zdají relativně jednoduché, při jejich aplikaci v praxi často vzniká komplexní celek složený z mnoha postupů a metod. Dalo by se říci, že v problému počítačového vidění pro konkrétní aplikaci lze jednoznačně definovat vhodnost použitého algoritmu a obecně i metodiky, ale často se v praxi setkáváme s tím, že zvolená aplikace se při sebemenší odchylce zpracovávaného prostředí od laboratorních podmínek, v jakých byl daný algoritmus projektován, jeví jako naprosto nevhodná a nedostačující. Právě z těchto důvodů je vhodné mít o tématice počítačového vidění přehled jako o celku, aby bylo možné použitý algoritmus adaptivně přehodnotit nebo upravit na základě průběhu projektu nebo předchozích získaných výsledků, popřípadě chyb.

Protože jednotlivé metody se v mnoha případech velmi překrývají, doplňují, někdy dokonce i vylučují, je potřeba stanovit určitou základní klasifikaci metod pro zlepšení orientace a vhodného výběru při kombinaci metod. Klasifikace metod není často jednoznačná a jednotlivé zdroje se v zařazení metod do kategorií liší podle konkrétní modifikace metody. V rámci této publikace provedeme rozdělení pouze orientační podle základních kritérií a až v podrobnějším popisu metod se zaměříme na jejich podrobnější zařazení a vlastnosti. Rozdělení metod je založeno především na zkušenostech z předchozí práce [1], na kterou tato navazuje a z níž vychází.

2.1 Klasifikace metod

Klasifikací, která je společná pro všechny metody, a zároveň i jednoznačná, je dělení (resp. charakteristika) metod podle vlastností vstupního obrazu. Ačkoli mnoho metod nepracuje interně s rastrem, na počátku celého zpracování obrazu stojí v naprosté většině případů rastrový obraz nebo diskretní signál. Dělení podle typu vstupního obrazu je mnoho, nicméně zde probereme jen ty nejdůležitější pro počítačové vidění.

Nejpodstatnější z hlediska předzpracování obrazu, ale mnohdy i přímé analýzy algoritmem počítačového vidění je **barevný model** [2] vstupního obrazu, především pak rozdělení na modely:

- **Barevný obraz** – je charakterizován několika, ve většině případů třemi složkami, jejichž poměr a hodnota určuje jednak barvu, ale také i jas konkrétního pixelu. Vícesložkový obraz nese více informace, a proto je vhodné jej použít v případě potřeby podrobnější analýzy tam, kde obraz monochromatický nedostačuje k řešení problému kvůli ztrátě informace. Pro obraz v počítačovém vidění se používá mnoho barevných modelů v závislosti na konkrétních požadavcích na invariantnost daného modelu k používaným transformacím a jejich informační hodnotě. Nejčastěji pak jde o modely **RGB**, **HLS** a **HSV**. Nevýhodou barevného obrazu je složitost jeho

zpracování, kde pro základní operace složitost lineárně vzrůstá s počtem složek obrazu a pro některé operace složitější může vzrůstat dokonce až exponenciálně v závislosti na prováděné operaci (příkladem takové operace může být shluková analýza barevného schématu obrazu v n-rozměrném prostoru barev).

- **Monochromatický obraz** – je charakterizován pouze jedinou složkou, nejčastěji hodnotou jasu konkrétních bodů. Tento typ obrazu je jednoduchý na zpracování většinou metod, nicméně nenese tolik informace jako obraz barevný a proto se hodí pro řešení velmi omezené množiny vstupních problémů a okolních podmínek. Metody pracující s monochromatickým barevným modelem (ve většině případů ve stupních šedi odpovídající intenzitě světla z daného bodu) vyhodnocují nejčastěji model osvětlení scény, a s ním související analýzu zakřivení a hran obrazu. Výhodou takového obrazu je také to, že jej lze zpracovat jako vstup ze senzorů pracujících mimo viditelnou oblast spektra (infračervené, ultrafialové světlo, rádiové vlny) nebo dokonce pro vyhodnocení vstupního obrazu získaného průběhem jiné, neelektromagnetické veličiny naměřené jako rastru sítí senzorů.

Často se v metodách zpracování obrazu provádí převod barevného modelu scény na monochromatický pro přibližné vyhodnocení scény, a po segmentaci obrazu se na jednotlivé části aplikují metody pracující s obrazem barevným pro upřesnění detailů, popř. vyhodnocení chyb analýzy. Podrobnější informace o řetězci zpracování budou uvedeny v následující kapitole.

Další důležitou charakteristikou vstupního obrazu je jeho **kvantitativnost**. V zásadě se jedná o stanovení počtu obrazů do metody vstupujících nebo počtu obrazů, jaký metoda pro úspěšné vyhodnocení scény ve skutečnosti potřebuje. Obecně lze metody rozdělit na tři základní skupiny:

- **Metody pracující s jediným obrazem** – Z hlediska aplikace algoritmů je jediný obraz nejvhodnějším vstupem, protože je jednoznačný a není třeba řešit konflikty vzniklé nepřesností analýzy a transformacemi objektů a okolí. Ve skutečnosti ale tato výhoda přináší mnoho problémů způsobených právě nedostatkem informace z takto jednoduchého systému získané. Často lze takový obraz použít jen pro základní analýzu statických a předem známých prvků v obraze a v jiných než laboratorních podmínkách nebo syntetických scénách je složité příslušné algoritmy adaptovat na daný vstupní obraz. V naprosté většině případů se pak při analýze využívá principu částečné znalosti scény nebo sledovaných objektů a jejich korelace k modelům uloženým v databázi rozpoznávacího algoritmu.
- **Metody pracující s více obrazy téže scény** – Podobně jako pro lidské vidění, i pro počítačové vidění je vhodné mít k dispozici více vstupů. Metody pracující s více obrazy téže scény patří do oboru **Stereoskopie** [12], a nejčastěji se používají k získání přibližné 3D informace o scéně. Díky

posunutí kamer scény je možné jednotlivé body projekce korelovat a získat tak hloubkový profil scény. Stejně jako metody jiné, i tento postup není samospasitelný a tak pro je nutné jej pro korektní výsledky kombinovat s dalšími metodami pro upřesnění vjemu.

- **Metody pracující se sekvencí obrazů** – Sekvencí obrazů rozumíme časovou souslednost vstupních snímků scény, díky které je možné na základě dodatečných informací o scéně nebo pohybu kamery kombinovat první dva typy metod a podle potřeb získávat informace o statických nebo dokonce i dynamických objektech ve scéně. Metoda spočívá v upřesňování informace v rámci jednotlivých snímků s využitím výstupů analýzy snímku předchozího. Ve vhodných podmínkách je možné tuto metodu zkombinovat s metodami stereovidění a získat tak ještě podrobnější informace o scéně nebo pohybu kamery. Problémem dynamické analýzy snímku je potřeba zpracování v reálném čase, která je vzhledem k současným výkonostním možnostem počítačů stále nedostačující pro složitější metody a proto se použitelnost dynamických metod omezuje většinou jen na dodatečné zpracování zaznamenaných scén.

Podstatnou charakteristikou metod je rozdělení na základě **vnitřní reprezentace obrazu**. Reprezentace obrazu v rámci konkrétní metody i systému je důležitá, protože na ní z velké části závisí výpočetní náročnost algoritmu, která úzce souvisí právě s počtem zpracovávaných prvků scény. Metody podle reprezentace obrazu se dělí na tyto:

- **Rastrové** – Obraz v těchto metodách je zpracováván na úrovni jednotlivých pixelů se všemi výhodami i nevýhodami z toho vyplývajícími. Rastrové metody patří k metodám s největším počtem zpracovávaných prvků obrazu, nicméně nelze říct, že by kvůli tomu patřily k metodám nejpomalejším. Tyto metody jsou často používány pouze k získání elementárních informací o obraze, které jsou dále využívány metodami jinými, nicméně existuje i mnoho aplikací, kdy rastrová metoda dokáže být dostatečně komplexní pro získání informace složitější. Nevýhodou komplexnějších aplikací takových metod je pak ovšem vysoká výkonová náročnost. Díky pravidelnosti rastru a jeho reprezentaci v kartézských souřadnicích je možné tyto metody optimalizovat použitím tzv. **pyramidálního** [13] zpracování kdy metoda pracuje s rastrem na různé úrovni detailu (podrobněji bude rozebráno níže). Při rastrovém zpracování je nutné brát zvýšený ohled na šum v obraze, který má na výsledky těchto metod nezanedbatelný vliv.
- **Vektorové** – Narozdíl od rastrových metod jsou metody vektorové silně závislé na předzpracování obrazu. Samotné vektorové metody nelze v praxi téměř použít, protože právě segmentace rastru vytváří základní vektorovou reprezentaci scény pro další zpracování. Vektorově předzpracovaný obraz se oproti rastrovému vyznačuje relativně nízkým počtem prvků a díky tomu možností aplikovat na něj složitější algoritmy a matematické postupy při zachování

uspokojitelné složitosti zpracování. Tyto metody často neprodukují samy o sobě výstup s dostatečnými detaily, a proto se jejich výsledky dále lokálně kombinují s metodami rastrovými pro upřesnění charakteristiky zpracovávaných oblastí.

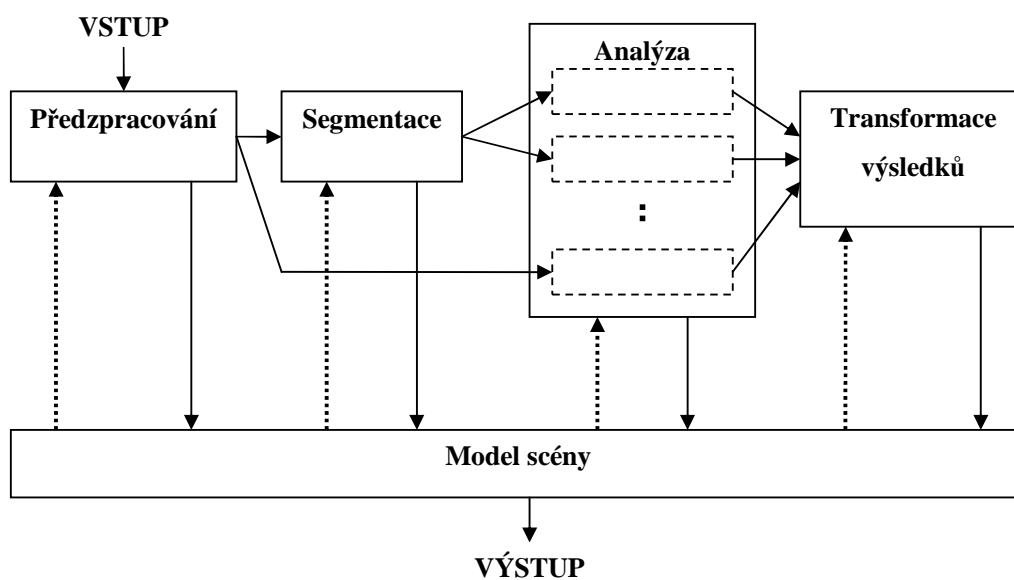
- **Kombinované** – Existují metody, které nelze jednoznačně zařadit do jedné ani druhé skupiny. Tyto metody lze nazvat kombinované a vyznačují se většinou tím, že pracují s omezenou informací z rastrového obrazu, tu zpracovávají interně, a produkují vektorové výstupy. Mezi takové metody lze zařadit např. Houghovu transformaci (bude probrána později) nebo obecně metody pro segmentaci obrazu. Opačným postupem, tedy vektorovými vstupy a rastrovým výstupem, je možné naopak provést nějakou rastrovou operaci se vstupním obrazem a přizpůsobit ho tak na základě výsledků analýzy metodě, která jej zpracovává.

Jak již bylo řečeno, rozdělení metod z hlediska dalších charakteristik je velmi nepřesné a proto není vzhledem k charakteru této práce vhodné jej pevně stanovovat. Zaměříme se proto nyní více na zpracování obrazu jako celek a poté na konkrétní metody a jejich vlastnosti, výhody a nevýhody.

2.2 Řetězec zpracování obrazu

Je jen velmi málo případů, kdy zpracování obrazu v počítačovém vidění probíhá cestou jediné metody aplikované přímo nad vstupním obrazem a získáním jejího výstupu. Zpracování obrazu jako celek je složitý proces a jen těžko si lze představit, že by bylo možné nějakým jednoduchým způsobem naučit počítač “vidět“. Vezmeme-li v úvahu vidění lidské, jedná se bezpochyby o složitý řetězec operací. Jestliže chceme naučit počítač vidět ve smyslu, v jakém tento pojem sami chápeme, musíme nutně řetězec zpracování co nejvíce přizpůsobit nebo přiblížit právě procesu vidění člověka. Z tohoto hlediska je nutné v ideálním případě provést nejprve předzpracování obrazu, ošetření vlivů jako šum, změny světlosti apod., dále provést základní zpracování hrubé informace, segmentaci, podrobnější zpracování pro získání detailů, vyhodnocení obrazu jako celku a v neposlední řadě aplikovat zpětné vazby na jednotlivé prvky řetězce.

Typický řetězec zpracování obrazu podobně jako rozdělení metod nemá přesná pravidla, a proto si zde popíšeme řetězec komplexní, jak by měl vypadat a pracovat v ideálním případě a popíšeme jednotlivé jeho funkční bloky a závislosti, viz obrázek 2.1. K tomuto modelu se snaží konvergovat většina postupů počítačového vidění, nicméně většina aplikací naráží na omezení jednotlivých metod, které neposkytují dostatečnou variabilitu a rozšiřitelnost na to, aby dokázaly řetězec jako celek realizovat, proto ve většině případů je v existujících aplikacích systém realizován pouze jako omezená část zpracujícího řetězce nebo řetězec kompletní, ale s velmi omezenými vazbami mezi jednotlivými segmenty.



Obrázek 2.1: Model analýzy obrazu.

Základním blokem každého systému analýzy je **předzpracování vstupu**. Tento blok typicky transformuje **vstup** na reprezentaci validní pro vstup dalších bloků zpracování, a provádí některé jednoduché operace s obrazem pro zlepšení interpretovatelnosti vstupu, například filtraci šumu, redukci zkreslení a podobně. Dalším blokem v řetězci zpracování, nutným především ve složitějších systémech, je **segmentace obrazu**. Proces segmentace dělí obraz na jednotlivé logické bloky pro zvýšení efektivity zpracování, ale také rozdělení obrazu na části vyžadující různý způsob zpracování. Správná segmentace je často klíčovým prvkem zpracování obrazu. Následující strukturu řetězce není možné popsat universálně, protože právě zde se konkrétní metody analýzy výrazně liší, jde o blok **analýzy obrazu**. Analýza sestává v jednoduchých systémech z jediné metody, ve složitějších pak obecně ze sítě několika různých metod zpracovávajících stávající vstupní informace, a transformujících tyto do požadovaných výstupních informací. Konkrétní implementace se liší v závislosti na použitých metodách, a proto k tomuto bloku zpracování nelze obecně říci více. Důležitou částí systému, především v případě použití více metod je **zpracování výsledků**. Tento blok řetězce typicky kombinuje výsledky jednotlivých metod a provádí jejich normalizaci a převod do informace využitelné v rámci modelu scény systému.

Model scény je typicky stavový prostor, který registruje výsledky analýzy a uchovává je pro výstup nebo další použití v rámci zpětných vazeb metod, nebo dalších bloků systému, zde pojem "systém" již nepředstavuje pouze systém počítačového vidění, ale například komplexní systém vnímání robota. Je důležité, aby informace zde uložená byla kompatibilní v rámci ostatních modulů, aby nedocházelo k misinterpretaci modelu na výstupu. **Výstup** je pak typicky již jen výběr konkrétní

informace právě z modelu scény v závislosti na požadované informaci. Zaměříme-li se na propojení jednotlivých modulů řetězce zpracování, je snadno vidět, že model není plochou strukturou. To je dáno tím, že pro kvalitní zpracování je třeba, aby měla každá část systému k dispozici co nejvíce vstupní informace, nejlépe ze všech předcházejících bloků. Model by samozřejmě bylo možné převést na plochý, nicméně v tom případě by docházelo k nežádoucí redundanci velkého množství informace, a proto se toto v praxi neprovádí.

Jestliže aplikace vyžaduje systém s udržováním stavu, nebo je nutné provádět analýzu na základě výsledků analýzy předchozí, je vhodné v systému zavést **zpětnou vazbu** mezi moduly. Zpětná vazba pomáhá v systému eliminovat některé chyby analýzy, vlastní analýzu zrychlit, nebo získat další užitečné informace, které by nebylo možné získat bez jejího využití. Může se vyskytovat obecně mezi libovolnými bloky systému, nicméně je vhodné kvůli rychlosti zpracování omezit tyto vazby pouze na ty nejpodstatnější. Zároveň je třeba důkladně zvážit její vliv na zpracování, a eliminovat případy, kdy může být naopak nežádoucí, nekvalitně zvolená zpětná vazba může výrazně snížit stabilitu metod a tím také i kvalitu výsledků analýzy. Problematika zpětné vazby v systému je složitá a pravděpodobně by sama o sobě zabrala mnohem více prostoru než je rozsah této práce, proto se jí dále nebudeme zabývat. Místo toho raději probereme jednotlivé metody analýzy.

2.3 Rastrové metody

Rastrové metody zpracování obrazu jsou založené na přímé interpretaci pixelů obrazu a práci s nimi. Jejich vstupem je rastrový obraz pocházející přímo ze snímacího prvku, popřípadě výstupní obraz metody jiné, ve většině případů opět rastrové. Často je možné použít řetězec triviálních rastrových metod nad vstupním obrazem pro získání robustnějšího algoritmu analýzy obrazu. Vzhledem k předchozím zkušenostem (v rámci [1]) s metodami založenými na rastru se ukázalo, že tyto metody nejsou příliš vhodné pro analýzu komplexnějších scén, protože jsou často citlivé na šum nebo nejednoznačnosti obrazu, a také jsou výpočetně náročné vzhledem k množství získané informace. Naopak velmi často je vhodné tyto metody použít jako metody pro předzpracování, segmentaci, úpravy nebo filtrování obrazu před vlastní analýzou metodami vektorovými. Rastrové metody jsou citlivé na aliasing v obraze a šum, proto je nutné již při jejich návrhu s těmito skutečnostmi počítat. V opačném případě totiž často dochází k tomu, že metoda testovaná na syntetické scéně je v reálném prostředí zcela nepoužitelná právě kvůli omezením vyplývajícím z jejich charakteru. Typicky je třeba v metodě počítat s nedodržením lokality, měřítka, světelných podmínek, deformací a mnoha dalších vlivů.

Existuje pouze málo rastrových metod, které lze opravdu úspěšně použít v reálné scéně, nicméně z mnoha z nich je možné získat dostatek relevantních informací o charakteru obrazu nebo jeho částí. Budeme-li se zabývat analogií rastrových metod a struktury ústrojí lidského zraku tak jak jeho principu rozumíme, lze usoudit na to, že čistě “rastrové“ zpracování je doménou pouze

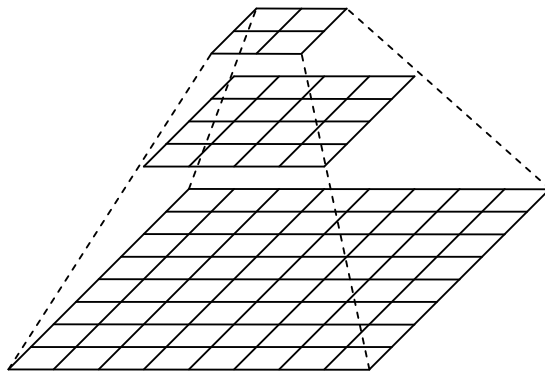
omezeného množství vrstev neuronů nejbližze sítnici, a směrem více k mozku je zpracování záležitostí souhrnných informací o jednotlivých oblastech obrazu zachyceného sítnicí a jejich vazeb jak je vidět na struktuře zrakového systému [4]. Lidský zrak a zpracování vnímané informace je nicméně oblastí dosud nedostatečně prozkoumanou a tak lze tyto předpoklady brát pouze jako inspiraci pro tvorbu vidění počítačového, nikoli jako fakta. Pozdější výsledky této práce a navrhovaného algoritmu mohou dokázat nebo vyvrátit, jestli byl tento předpoklad vhodný.

2.3.1 Optimalizace rastrových metod

Před vlastním rozborem jednotlivých metod je nutné nastínit alespoň základní principy používané pro optimalizaci rastrových metod. Důvodem optimalizace je velké množství prvků obrazu (pixelů) vstupujících do metody. Ne každá metoda, popřípadě její část potřebuje s obrazem pracovat v měřítku vstupního rozlišení, často dokonce není vůbec nutné pracovat s některými poměrně velkými oblastmi tohoto obrazu. Cílem optimalizace v rastrových metodách je tedy redukovat měřítko obrazu nebo omezit zpracovávanou oblast podle požadavků jednotlivých kroků algoritmu. V závislosti na použité metodě a její výpočetní náročnosti lze takto nejen lineárně snížit potřebný čas zpracování, ale dokonce změnit její časovou náročnost, často až v rámci řádu změny mocniny počtu prvků. Při výběru optimalizačních metod je nutné důkladně analyzovat přínos a omezení, které optimalizace přináší, protože zatímco na jedné straně dokáže metodu významně zrychlit, na straně druhé v případě nevhodně zvoleného způsobu optimalizace tato přináší dokonce zpomalení metody, v horším případě dokonce i degradaci jejích výsledků. Nesporným faktem je také to, že rychlostní optimalizace snadno může způsobit právě snížení přesnosti výstupu, v případě vhodné volby její zachování a tedy i kvalitní výsledky implementace. Optimalizace naopak nikdy nepřinese zlepšení výsledků a s tím je vždy nutné počítat. Optimalizačních metod je mnoho a proto zde rozebereme pouze ty nejpoužívanější.

Nejčastější metodou použitou při optimalizaci rastrového zpracování obrazu je tzv. **pyramidální zpracování** [13]. Optimalizace spočívá v tom, že kromě původního obrazu v originálním rozlišení má metoda k dispozici i podvzorkované verze téhož obrazu. Koeficient podvzorkování a udává poměr stran (rozměrových pixelů) původního obrazu a obrazu podvzorkovaného. Ve výsledku to znamená, že na pixel podvzorkovaného obrazu je mapováno $n = a^2$ pixelů, tedy celkové snížení počtů elementů obrazu je rovno právě hodnotě n . Nejčastější volbou je podvzorkování na polovinu původního obrazu, tedy s koeficientem $a = 2$, $n = 4$. Rekursivní aplikací podvzorkování vzniká soustava obrazů tvořících strukturu připomínající pyramidu, odtud název **pyramidální zpracování**. V praxi je tato struktura nejčastěji reprezentována MIPmapou (Multiple In Parvo), jejíž charakter i zpracování přesně odpovídá požadavkům pro uložení i zpracování dat touto metodou. Použití jednotlivých úrovní detailu obrazu pak může být v režii řídicího cyklu algoritmu nebo může být plně v kompetenci jednotlivých kroků zpracovávajícího algoritmu. V prvním případě

je náročnost algoritmu předem odhadnutelná na úkor rizika snížení přesnosti zpracování a algoritmus nepotřebuje žádné informace o tom, že pracuje s proměnným detailem obrazu, metoda je aplikovatelná na existující algoritmus bez nutnosti úprav. V případě, že úroveň detailu určuje zpracovávající algoritmus, musí k tomu být náležitě upraven. Tento proces je náročnější na návrh a implementaci, ale obecně přináší lepší výsledky, protože nabízí všechny možnosti předchozího zpracování volitelně rozšiřitelné o adaptivitu na základě průběhu zpracování. V případě vhodné scény a postupu dokonce ani nedochází k degradaci časové náročnosti přesto, že algoritmus pracuje s více úrovněmi detailu obrazu.



Obrázek 2.2: Pyramidální zpracování.

Druhou častou metodou optimalizace rastrových metod je použití **prahování** nebo jednoduchých **filtrů**, které do vlastní metody propustí vždy jen omezený počet pixelů [15]. V principu se jedná o samostatnou rastrovou metodu, která předchází vlastnímu zpracování, nicméně vzhledem k jednoduchosti je tento proces často integrován přímo do konkrétních metod právě pro zrychlení zpracování. Nejčastěji je použito jednoduché prahování, které do metody propustí pouze pixely se stanovenou intenzitou, náročnějším předzpracováním je použití jednoduchých hranových operátorů, ve výjimečných případech se používá pro rozlišení zpracovávaných a nezpracovávaných pixelů složitější funkce. Použití složitějších metod omezení množství zpracovávaných pixelů v sobě skrývá nebezpečí zvýšení náročnosti algoritmu jako celku v případě, že nedostatečně zvážíme poměr náročnosti metody filtrace a vlastního zpracování.

Třetí nejpoužívanější metodou pro optimalizaci je **specifikace oblasti zájmu** [15]. **Oblast zájmu** může být obecně libovolnou oblastí obrazu, o které předem víme, že obsahuje informace, které chceme analyzovat. Důležitým aspektem této metody je alespoň přibližná znalost analyzované scény, proto se často používá pouze v umělých prostředích, například v průmyslu při použití statické kamery nebo až po předzpracování metodou jinou, na základě které se teprve oblast zájmu stanoví. Oblast zájmu je nejčastěji specifikována jedinou funkcí nebo množinou funkcí ořezávajících definovanou oblast, nebo **mapou významnosti** pixelů. Mapa významnosti je většinou binární, určující přímo zda daný pixel má být zpracován nebo ne, v některých případech diskrétní nebo spojitá určující váhu

pixelu pro výstupní informaci nebo další informace, například v souvislosti s optimalizací pyramidální označení úrovně detailu která se má pro danou oblast použít. Podobně funguje v rámci lidského zraku také koncentrace na objekty, které jsou aktuálně zaostřené a zbytek scény je pro vnímání potlačen snížením úrovně detailu [5].

Další metody optimalizace se v rastrových metodách analýzy obrazu používají jen výjimečně a často jsou specifické pro konkrétní typ obrazu. Obecně ovšem platí, že optimalizace má své místo všude, a proto je vždy vhodné důkladně zvážit její použití.

2.3.2 Segmentace obrazu

Segmentace obrazu má mnoho podob a variant, nicméně faktem je, že právě rastrové metody jsou klíčovými pro tento typ předzpracování obrazu. Pojem „předzpracování“ je zvolen proto, že v naprosté většině případů pouhá segmentace nepřináší dostatek informace pro požadované zpracování. Mnohem častěji je výsledek segmentace obrazu využíván jako vstup metod dalších, často vektorových, které s výslednými segmenty dokáží pracovat mnohem efektivněji než rastrové principy. Nejčastějším důvodem segmentace je právě snížení počtu zpracovávaných oblastí a tedy celkové výpočetní náročnosti analýzy obrazu. Podrobný rozbor algoritmů segmentace by dokázal zabrat sám o sobě celou práci, proto rozebereme opět pouze základní používané principy.

Častým principem používaným pro segmentaci je takzvané **vyplňování oblastí** [16]. Proces provádí činnost podobnou vyplňování známou z počítačové grafiky s tím rozdílem, že funkce určující další postup algoritmu vyplňování je dána charakterem pixelů, nevyplňuje se pouze homogenní oblast. Jednotlivé oblasti jsou při segmentaci barveny konkrétními hodnotami barev a tyto jsou pak registrovány dalšími moduly analýzy. Vlastní vyplňování oblastí pak má mnoho podob a algoritmů stejně jako v počítačové grafice při rasterizaci. Nevýhodou tohoto procesu je kvůli pixelové orientaci náchylnost na šum a nežádoucí lokální změny v obraze, také produkuje rastrový výstup, a proto se nehodí pro použití v systému s vektorovou analýzou bez dalších modifikací výstupu.

Procesem, který používají především dynamické systémy nebo systémy se vstupem více obrazů je **segmentace rozdílů** [17]. Cílem je oddělit pozadí scény od popředí, jímž je typicky pohybující se objekt, v případě statické scény pak rozdíl trvalého stavu a aktuálního snímku. Na základě rozdílů pixelů v obraze je pak možné separovat pozadí od popředí scény a získat tak informace o velikosti, nebo dokonce obrysu sledovaného objektu. Zde je třeba počítat s možnou změnou osvětlení scény v čase a aplikovat jeho redukci na vstup metody. Tato metoda se často používá při analýze dopravních scén nebo gestikulace člověka.

Méně používanou, nicméně efektivní metodou segmentace je **shlukování** [18]. Shlukování umožňuje separovat jednotlivé objekty jako shluky základních elementů ve scéně a tyto dále analyzovat. Metoda je relativně pomalá, protože algoritmy shlukování vyžadují náročné nebo

rekurzivní průchody, náchylná na typ scény, a proto se používá spíše v průmyslových aplikacích nebo tam kde není nutné zachovat zpracování v reálném čase.

Segmentace podle jasu [15] je metoda rychlá s kvalitním výsledkem. Jedná se o víceúrovňové prahování obrazu, které produkuje segmenty v podobě oblastí se světlostí definované homogenity. V případě adaptivního prahování nemusí jít nutně o oblasti se stejnou úrovní jasu, ale i například o oblasti, kde změna jasu nepřesahuje stanovený limit v rámci dané vzdálenosti v obraze. Metoda je náchylná na výstřelkový šum, ostré stíny, ale hodí se pro zpracování jednoduchých a především dvourozměrných scén.

Segmentace na mřížku [19] je nejvhodnějším typem pro segmentaci vektorovou, protože produkuje vektorovou strukturu, kterou lze snadno dále analyzovat. Metoda provádí rekurzivní dělení obrazu na specifikovaný typ mřížky, která podle zadaných pravidel může být rovnoměrná nebo adaptivní v závislosti na segmentovaném obraze. Mřížka jako taková zde neurčuje konkrétní celky obrazu, proto je třeba ji dále analyzovat, jedná se zde pouze o předzpracování pro vhodný vstup vektorových metod.

2.3.3 Analýza hran

Mnoho reálných aplikací oboru počítačového vidění pracuje na principu analýzy hran v obraze. Za hranový pixel lze obecně považovat libovolný pixel obrazu, jehož sousední pixel nebo pixely přesahují stanovený limit rozdílnosti, tedy tvoří oddělení dvou sousedních oblastí obrazu, tedy hranu.

Hrana nemusí být nutně definována pouze změnou jasu, jak je na tuto problematiku často pohlíženo. Zdrojem hran může být obecně libovolná operace, která dokáže dostatečným způsobem vyhodnotit rozdílnost dvou oblastí obrazu (typicky změna jasu, barvy, textury, ale například i kontrastu). Metody analyzující hrany se obecně používají prvek předzpracování, který vyhodnotit v obraze hranové pixely jako binární obraz (pouze rozlišení hrana/plocha) nebo obraz hodnot určujících míru důvěryhodnosti hrany, ale i dalších informací o hraně (směr, charakteristiku, apod.).

Druhou částí metody je pak vlastní algoritmus, který obraz hran dále analyzuje a zpracovává buď rastrově, nebo vektorově, podle potřeby dané aplikace. Nejčastěji se pak množina hran prvků obrazu porovnává s databází hranové reprezentace objektů a na jeho základě se objekty v obraze detekují. Informace o hranách v obraze je z hlediska počítačového vidění velmi cenná a proto je některá z metod využívající právě hrany použita v rámci většiny větších celků pro zpracování obrazu, ať již pro účely segmentace, vlastní analýzy, nebo doplnění informací potřebných pro jiné metody.

2.3.4 Význačné body v obraze

Hledání význačných bodů (feature points) úzce souvisí s metodou analýzy hran, protože se o ni často opírá. Význačný bod v obraze je takový bod, který z hlediska obrazu jako celku přináší více

informace než body okolní a lze je v obraze libovolným způsobem detekovat. Pojem bod zde nutně nemusí odpovídat jedinému bodu, často je tato metoda aplikována na celé oblasti nebo prvky obrazu, nicméně tyto jsou často v rámci analýzy pro zjednodušení nahrazovány právě body, a proto se často v praxi setkáváme právě s tímto označením. Význačným bodem pak může být vrchol objektu, hrana, oblast se specifickou texturou, konkrétní objekt, ale i další prvky obrazu. Obecně závisí konkrétní volba na určení dané aplikace a analyzované scéně.

Význačné body pak lze využít v rámci jediného obrazu pro detekci objektů nebo jejich deformace, v takovém případě je nutné mít k dispozici databázi objektů obsahující význačné body takto uložených objektů. V případě více snímků pak lze na základě význačných bodů detekovat, popřípadě analyzovat pohyb, optický tok, ale i 3D informaci o objektech, bez nutnosti jejich předchozí znalosti.

Metody používající hledání význačných bodů jsou užitečné pro předzpracování obrazu a vyhodnocení základních podmínek pro další metody. Samostatné použití této metody se vyskytuje spíše výjimečně nebo v případech, kdy je scéna předem definovaná, například v průmyslových aplikacích. Často jsou tyto metody upravovány na stochastický systém pro zvýšení odolnosti proti šumu a změnám v obraze.

2.3.5 Houghova transformace

Houghova transformace [1] je jednou z mála robustních rastrových metod, která dokáže i z poškozeného obrazu získat kvalitní vektorový výsledek, proto ji probereme podrobněji. Metod transformace je několik a jsou obecně založeny na převodu pixelů obrazu do jiného parametrického prostoru obrazu než je prostor původní, ze kterého lze potřebnou informaci lépe získat. Metody hledají v obraze základní elementy a vytvářejí histogramy četnosti bodů na hledaných prvcích se podílejících. Extrakce informace probíhá hledáním lokálních maxim v tomto výsledném histogramu, přičemž souřadnice v prostoru transformace určují parametry daného prvku v obraze.

Základní (**lineární**) metoda Houghovy transformace slouží pro detekci přímých čar, tj. přímk a úseček, v obraze. Metoda spočívá v postupné transformaci bodů hran v obraze do parametrického prostoru (r, θ) . Každý bod tohoto prostoru představuje reprezentaci přímky v obraze, parametrizované souřadnicemi $[r, \theta]$ bodu. Parametr r představuje kolmou vzdálenost přímky od počátku souřadného systému obrazu, parametr θ pak úhel, který svírá kolmice k přímce s osou x souřadného systému obrazu. Pro histogram se transformují body, které úspěšně prošly definovaným hranovým operátorem, ve většině případů Sobelovým filtrem. Mapováním do prostoru transformace vzniká histogram takto parametrizovaných přímek, ze kterého lze analýzou lokálních maxim množinu nejpravděpodobnějších přímek získat. Dalším zpracováním je možné tyto přímky mapovat na obraz a vyfiltrovat jejich části, které se v obraze ve skutečnosti nevyskytují.

Druhou nejčastější variantou je **cirkulární** transformace. Tato metoda slouží k detekci kruhových útvarů v obraze. Metoda spočívá v nalezení význačných bodů (středů kružnic) jako lokálních maxim v histogramu vzniklém transformací. Na rozdíl od transformace lineární, histogram této metody odpovídá svým parametrickým prostorem prostoru obrazu, tj. body obrazu korespondují přímo s odpovídajícími body v histogramu. Vlastní transformace se provede tak, že pro každý bod obrazu hran překračující prahovou hodnotu zahrneme do histogramu všechny reprezentace potenciálních středů kružnic, jejichž bodem by zkoumaný bod mohl být. Množina takových středů leží na přímce dané směrem gradientu obrazu ve zkoumaném bodě, a procházející tímto bodem. Každý takový potenciální střed zahrneme do histogramu zvýšením jeho hodnoty na příslušných souřadnicích podobně jako u transformace lineární. Pro získání informace o hranách v obraze se opět většinou používá **Sobelova operátoru** [1], z jeho částí je možné získat jak intenzitu hrany, tak i gradient.

Stejně jako u transformace lineární, i zde nalezneme po vytvoření histogramu lokální maxima, která v tomto případě udávají středy potenciálních kružnic v obraze. Opět je zde velkým otázkem, jak správně nastavit prahovou hodnotu pro danou scénu, protože při nesprávné volbě nemusí být úsilí nalezení kružnice dostatečné, přesto že kružnice je výrazná, nebo naopak můžeme najít potenciální kružnici i v místech, kde se nic ani zdaleka připomínajícího kružnici nevyskytuje, například může být jako kružnice analyzována i soustava dvou různoběžných úseček.

Třetí významnou variantou je transformace **obecná**. Obecná Houghova transformace pracuje podobně jako transformace kruhová s tím rozdílem, že hledanou křivkou není kružnice, tedy množina bodů od hledaného středu stejně vzdálených, ale obecná křivka. Vstupem transformace je pak rovnice hledané křivky nebo tabulka bodů tuto křivku aproximující, a hranové body jsou uvažovány jako body ležící na všech potenciálních křivkách bodem procházejících. Podobně jako u cirkulární transformace i zde vzniká histogram potenciálních středů křivek, jehož maxima odpovídají nejpravděpodobnějšímu výskytu odpovídající křivky (křivek) v obraze.

2.3.6 Optický tok

Metoda vyhodnocování optického toku v obraze je metodou používanou v případě vstupu dynamického obrazu. Na základě korelace sekvence obrazů stejné scény je možné vyhodnotit tzv. **optický tok** obrazu a na jeho základě usuzovat na tvar pohybujících se předmětů. Optický tok definovaný nad sekvencí obrazů je dvourozměrné pole vektorů směrem a velikostí odpovídající pohybu jednotlivých elementů (pixelů) obrazu. Jestliže se objekt nebo jeho části pohybují v 3D prostoru, jsou dráhy jednotlivých bodů jeho povrchu v projekční rovině závislé na pozici těchto bodů vůči souřadnému systému objektu a pohybové rovnici objektu. Na základě deformace pole drah bodů povrchu objektu je pak možné proložit polem optického toku plochu a aproximovat tak průběh tvaru objektu. Přestože je metoda v případě kvalitního vyhodnocení optického toku pixelů velmi robustní a

produkuje kvalitní výsledky, právě tato počáteční operace přináší metodě více problémů než užítku v reálném prostředí a proto se v praxi příliš nepoužívá. Vyhodnocení optického toku je velmi náročný proces, protože je třeba kvalitně korelovat jednotlivé body v následných obrazech a kvůli tomu je metoda pomalá a citlivá na šum v obraze a změny osvětlení. Mnohem častěji než k přímé analýze tvaru předmětů se tato metoda v praxi používá ke stabilizaci obrazu nebo jeho částí, popřípadě k vyhodnocení pohybu kamery vůči scéně pro zpřesňování analýzy metod jiných.

2.3.7 Analýza textury

Analýza textury je důležitým procesem při zpracování reálných scén. Právě textura zanáší do reálného obrazu šum znemožňující potřebnou analýzu a proto je potřeba se s touto obrazovou informací nějak vyrovnat. Textura je pravidelně či nepravidelně se opakující vzor povrchu objektu. Podobně jako barva, i textura může přinést mnoho informace pro segmentaci obrazu. Analýzu textury je možné v principu rozdělit na dva procesy, **klasifikace** a **analýzu průběhu** v obraze.

Klasifikace textury je proces, kdy se snažíme na základě analýzy konkrétní oblasti obrazu určit třídu textury, o jakou se jedná (dřevo, kámen, mřížka, šachovnice...). Tuto informaci je možné získat pomocí získání normalizovaných charakteristik oblasti a jejich porovnání s naučenými nebo předem známými modely. Důležitými prvky pro klasifikaci jsou metriky jako průměrná hodnota pixelu, po normalizaci odstín barvy, rozptyl pixelů v dané oblasti, histogramy pixelů, význačných bodů, prvků, ale i další. Množství informace použité pro klasifikaci textury by mělo odpovídat počtu klasifikovaných tříd tak, aby bylo možné je jednoznačně identifikovat. Vlastní obraz složený z pixelů není jako metrika dostatečně vhodný, protože není invariantní vůči většině transformací, a proto vede na uložení mnoha variant dané třídy textury. Mnohem vhodnější je použít pouze momenty charakterizující obraz jako celek a invariantní vůči operacím s obrazem právě kvůli tomu, že textura může být deformovaná, posunutá, natočená, nedosvětlená, přebarvená, a podobně.

Po provedení klasifikace textury nebo v případě že předem texturu zkoumaného objektu nebo scény známe, je možné výstupní informaci použít jednak pro zvýšení efektivity segmentace scény, kdy se díky znalosti textury můžeme vyhnout dělení texturovaných oblastí na menší kontraproduktivní celky, ale také pro analýzu 3D vlastností objektu, kdy na základě deformace pravidelné textury v projekční rovině dokážeme vyhodnotit zakřivení povrchu. Přestože analýza textury je často náročná, nejen kvůli tomu, že tříd textur existuje velmi mnoho a často se i překrývají, měl by sofistikovaný systém pracující s reálnou scénou s texturovanými předměty počítat a tedy analýzu provádět. Zpracování textury samo o sobě musí být robustním systémem odolným proti šumu a dalším nepříznivým vlivům reálné scény, a proto by bylo pravděpodobně vhodné uvažovat v rámci kvalitního systému počítačového vidění použití právě této metody jako část řetězce zpracování, která by měla pomoci právě s eliminací těchto vlivů. Systém analyzující texturu by měl pracovat, a také

často pracuje s neurčitostí, proto je vhodné použít právě zde metody jako neuronové sítě, fuzzy logiku, hrubé množiny ale i další variabilní systémy.

2.4 Vektorové metody

Narozdíl od metod rastrových, vektorové metody pracují nad scénou s vektorově definovanými částmi obrazu a provádějí jejich analýzu. Uvážíme-li extrémní pohled na obraz, každý jeho pixel lze chápat jako vektorový objekt a tedy metody rastrové by bylo možné tímto způsobem snadno kategorizovat i mezi vektorové, nicméně z hlediska dělení metod budeme za vektorové považovat metody pracující obecně s definovanou oblastí nebo prvkem obrazu, které v závislosti na aktuálně zpracovávané úrovni detailu mohou obsahovat proměnlivý počet pixelů nebo v případě nižšího rozměru objektu (body, přímky) nemusí obsahovat pixely žádné. Důležitým aspektem a výhodou právě takového zpracování je snížení počtu zpracovávaných prvků obrazu, protože homogenní oblasti čítající mnoho pixelů lze nahradit aproximovanými polygony, sadu pixelů tvořící hranu jednoduchou úsečkou, a podobně. Díky tomu lze složitý, rastrem tvořený model redukovat na jednodušší, vytvořit metody s variabilní úrovní detailu zpracování, vyvarovat se nepříznivým vlivům reálných scén jako je šum, lokální změny v obraze, a zároveň produkovat z metod přímo model scény bez nutnosti jeho dalších úprav. Jak se v experimentech ukazuje, právě vektorové metody jsou pro reálné a předem neznámé scény jako základ efektivnější a dokáží si s takovou scénou lépe poradit na úrovni 3D modelu, než metody čistě rastrové. Proto se tato práce a navrhovaný algoritmus bude opírat právě o tyto postupy.

Základem vektorových metod je rastrové předzpracování. To je nutné vzhledem k charakteru vstupního obrazu. Předzpracování obrazu pro vektorovou metodu typicky provádí segmentaci obrazu na vektorové celky, přičemž často je tato operace doplněna i jinými rastrově orientovanými bloky zpracování nad oblastmi segmentace pro získání většího množství informace v rámci jediného průchodu. Souhrnná informace o oblasti pak často postačuje v jednoduché scéně k dostatečné analýze, metody využívající takové informace jsou pak často založeny na analýze průběhu osvětlení a stínů na povrchu objektu.

2.4.1 Analýza mřížky

Analýza mřížky je postup provádějící komplexní a často i rychlou analýzu základní geometrie scény. Metoda spočívá ve vektorizaci obrazu do mřížky polygonů, které aproximují konkrétní části obrazu. Výhoda mřížky spočívá především ve snížení počtu analyzovaných objektů. Díky spojitosti struktury mřížky je pak tato metoda vhodná především pro analýzu tvaru nebo vlastností spojitého povrchu objektu. Typicky se při analýze využívá informace o světelné intenzitě jednotlivých částí mřížky nebo dalších sumarizujících informací a obecně se pracuje s uzly mřížky pro vytvoření modelu scény.

Výhodou tohoto postupu je především možnost přímé reprezentace modelu zpracování (mřížky) jako výstupního modelu. Tato metoda je poměrně odolná vůči šumu a dalším nepříznivým vlivům, nicméně předpokládá dostatečné rozlišení vstupního obrazu, při kterém jednotlivé elementy mřížky mohou vhodně sumarizovat několik pixelů obrazu. Použitá mřížka pak v závislosti na metodě může být pravidelná nebo adaptivní, složená z různých typů polygonů, s různým způsobem sumarizace jednotlivých oblastí. Díky variabilitě mřížky se pak lze úspěšně vyvarovat nežádoucí chybné analýze scény v problematických místech pouhou degradací detailu mřížky v dané oblasti. Zároveň je podobným způsobem možné zaměřit detailní analýzu na konkrétní objekt detekovaný nějakým předchozím postupem, většinou metodami segmentace nebo detekce objektů.

2.4.2 Analýza osvětlení

Analýza osvětlení je metodou spíše kombinovanou než čistě vektorovou. Nicméně proto, že se kvůli rychlosti a větší robustnosti zpracování jednotlivé oblasti často aproximují do větších celků, a často se oblast analýzy nezaměřuje na oblasti definované pravoúhlou strukturou pixelů, lze ji považovat za vektorovou. Způsobů analýzy osvětlení je mnoho, všechny však pracují na stejném principu. Metoda spočívá v aplikaci rovnice osvětlení na plochu nebo křivku proloženou projekční rovinou. Na základě rovnice osvětlení je možné transformovat body křivky nebo roviny do odpovídajících hodnot souřadnic v ose Z. Algoritmus potřebuje ke správné funkci jako vstup hodnotu intenzity jednotlivých pixelů nebo oblastí obrazu a proto je ve složitějších scénách nutné provést předzpracování, které dokáže právě tuto intenzitu vyhodnotit a eliminovat nežádoucí vlivy nad pixely scény. V případě složitější scény se pak systém používající analýzu osvětlení neobejde bez kvalitního analyzátoru textury, protože právě textura má velmi negativní vliv na lokální intenzitu pixelů v obraze. Na objekty s homogenním povrchem je naopak tato metoda velmi vhodná. Nevýhodou této metody je nutnost znalosti modelu osvětlení scény a proto se používá spíše v laboratorních nebo předem známých podmínkách, tj. v průmyslu nebo například ve 3D skenerech v lékařství. Často je osvětlení generováno uměle pro zvýšení efektivity metody. Profil získaný analýzou osvětlení je pak dále analyzován.

2.4.3 Hranové modely

Hranové modely jsou vektorovým rozšířením rastrové analýzy hran. Právě rastrová analýza hran zde tvoří základní prvek představující transformaci scény na množinu vektorových hran, ve kterých je poté vyhledáván odpovídající hranový model objektu z databáze. Klíčovou operací je zde právě nutnost výběru správných hran a jejich porovnání s modelem, na to se v nejjednodušších metodách používá často suma čtverců, při složitějším zpracování sofistikovanější postupy využívající pravděpodobnostního počtu, někdy i neuronové sítě. Hlavní nevýhodou metody je omezení analýzy, resp. detekce objektů na známé modely a velká náchylnost k nesprávné detekci v případě že scéna

obsahuje velké množství hran. Vhodné je tuto metodu doplnit o metodu jinou, například analyzátor textury, právě kvůli eliminaci těchto vlivů. Metoda se často používá pro monitorování dopravy nebo v průmyslu, většinou v kombinaci s Houghovou transformací pro analýzu hran v obraze.

2.4.4 Optimalizace vektorových metod

Podobně jako metody rastrové lze optimalizovat i vektorové metody. Stejně jako v předchozím případě spočívá optimalizace především ve snížení počtu prvků analyzovaných objektů. Typicky se jedná v rámci plošných objektů (polygonů) o slučování do větších celků, při pohledu návrhu shora dolů omezení dělení obrazu na konkrétní úroveň. V případě jednorozměrných objektů (obecně křivek) se optimalizace provádí vynecháním některých nevýznamných bodů nebo opět nezahrnutím takových bodů do křivky již při její detekci. Z hlediska bezrozměrných objektů (bodů) nelze proces označit přímo za optimalizaci, protože vynechaný bod představuje ztrátu informace o konkrétním místě obrazu, nicméně je možné zvážit takový vliv na výsledek a na základě toho omezit množinu analyzovaných bodů pro snížení náročnosti algoritmu jako celku.

Stejně jako u optimalizací rastrových algoritmů i zde obecně platí, že optimalizace snižuje přesnost analýzy, ale naopak dokáže při vhodně zvoleném postupu eliminovat vliv šumu nebo nepřesností v obraze. Volba správné metody optimalizace je často klíčová pro úspěšnost algoritmu i celého systému.

2.5 Kombinace metod

Jak už bylo výše zmíněno, jednotlivé metody, popřípadě typy metod, lze kombinovat pro dosažení lepšího výsledku systému jako celku. V některých případech je kombinace obou přístupů dokonce nutná a některé metody jsou na ní přímo založeny. Především metody vektorové samy o sobě nelze téměř použít (vyžadovaly by vektorový vstup) a proto vektorově orientovaný systém lze v naprosté většině případů spíše považovat za kombinovaný.

Typická struktura kombinovaného systému sestává z rastrového předzpracování, řetězce rastrových a vektorových metod a bloku transformace výstupní informace z výstupu metod do výstupního modelu scény, typicky již vektorově reprezentovaného.

Důležitou součástí kombinovaného systému je definice propojení jeho jednotlivých částí, především konverze mezi formáty vstupu a výstupu jednotlivých modulů. Nevhodná konverze výsledků konkrétního modulu pro vstup modulu dalšího může v systému jako celku způsobit vážné problémy s propustností i kvalitou výsledků. Je vhodné tuto skutečnost uvažovat již při návrhu algoritmů, především tam, kde je to možné zajistit, aby jednotlivé algoritmy pracovaly s totožnými nebo velmi podobnými (snadno a kvalitně převoditelnými) datovými strukturami. To lze ve většině případů zajistit relativně snadno, protože autor systému jako celku bývá shodný s autorem konkrétních modulů a tak lze zpětně modifikovat návrh pro dosažení maximální kompatibility.

U systémů složitějších nebo systémů s rozhraním pro dodatečné přidávání externích modulů je třeba zajistit, aby propojení na takové moduly obsahovalo buď univerzálně definované typy objektů odpovídající konkrétnímu standardu a obsahující odpovídající informace potřebné pro externí modul, nebo dostatečně dokumentované struktury informací nabízející maximum informace ze stávající analýzy. V případě nedostatečného propojení může snadno dojít k nutnosti redundance analýzy v případě, že externí modul vyžaduje neexportovanou (byť jednoduchou) informaci která je známá systému pouze interně. Jako nejvhodnější řešení se jeví ukládání veškerých informací analýzy centralizovaně v jádru systému a povolení přístupu jak pro čtení tak pro modifikaci takového úložiště všem modulům.

3 Návrh algoritmu

Primárním úkolem této práce je vytvořit vlastní algoritmus analýzy 2D obrazu. Vzhledem k prozatímním zkušenostem s jednotlivými algoritmy pro analýzu [1] a nemožnosti jejich dalšího rozvoje se ukázalo, že cesta standardních metod není pro reálnou scénu příliš vhodná. Cílem tedy bude vytvořit vhodný základ pro analýzu jednoduché scény, který bude dostatečně robustní a modulární pro možnost rozšiřitelnosti pro případ složitějších scén.

Jestliže se chceme přiblížit vlastnostem lidského zraku, bude vhodné se inspirovat právě v procesu lidského vidění, jak jej známe, a aplikovat metody pracující s neurčitostí, abychom se vyhnuli nepříznivým vlivům vstupního obrazu, jako je šum a podobně. Výsledný algoritmus by tedy neměl striktně počítat s pevnou matematikou, ale spíše dokázat se přizpůsobit analyzované scéně na úkor „strojové“ přesnosti silně ovlivněné chybami. Koneckonců lidské vidění také postrádá přesnost analýzy viděného objektu a dostatečně se spokojí s přibližným modelem prostředí. Upřesnění vjemu je vždy možné provést posléze s využitím přibližné informace získané ze základní metody analýzy.

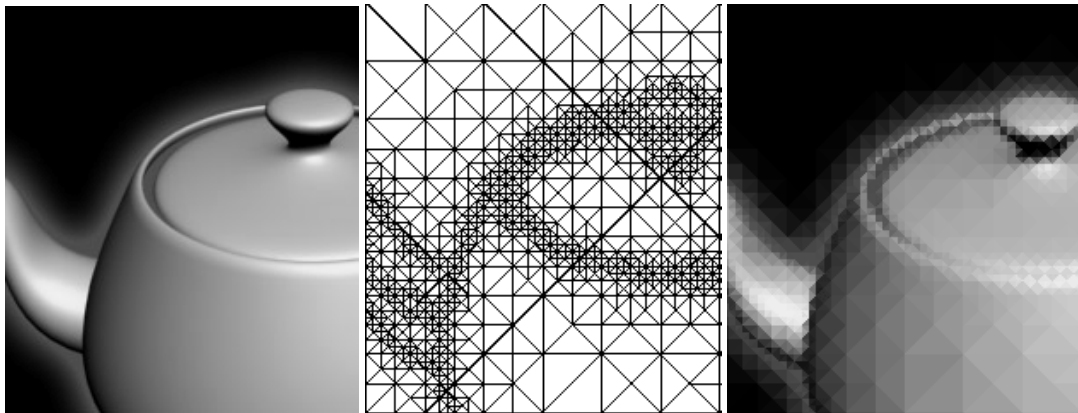
Kvůli robustnosti bude vhodné jádro algoritmu založit na vektorových metodách, s odpovídajícím rastrovým předzpracováním. Bude vhodné pokusit se vytěžit ze vstupního obrazu maximum informace, a pokud to nebude nutné jinak, spokojit se pouze s jedním vstupním obrazem. Jestliže bude algoritmus úspěšný být i na jediném obrazu, použití více vstupních obrazů přinese výrazné zlepšení jeho výsledků, při aplikaci stereovidění pak i lepší aproximaci počátečních podmínek analýzy.

Celek by měl v základu pracovat jako dopředný vrstvý model s možností pozdějšího rozšíření o zpětnovazební smyčky. Zpětná vazba by měla být realizovatelná jednak mezi konkrétními vrstvami analýzy pro upřesnění výsledků, ale také v rámci systému jako celku pro odhad počátečního nastavení dalšího kroku v případě zpracování více obrazů. Zpětná vazba nicméně nebude předmětem této práce, pouze budou nastíněny předpoklady a návrhy její realizace na základě zkušeností s vytvořeným systémem. Stejně tak rozšiřující moduly budou spíše tématem návrhu než vlastní implementace.

3.1 Segmentace obrazu

Správná segmentace obrazu je základním klíčem úspěchu analýzy obrazu. Budeme se snažit o takovou segmentaci, jejíž pracovní model by mohl být vhodným vstupním modelem 3D analýzy samotných objektů. Bude vhodné se zabývat analýzou scény jako celku, protože scéna jako taková může stejně tak být pohled na jediný objekt nebo i pouze jeho část, proto bude v počátku vhodné proložit projekční rovinou jedinou plochu, a až po základní analýze ji případně segmentovat na samostatné oblasti.

Nejvhodnější pro analýzu spojitého povrchu se jeví segmentace na mřížku, kterou lze pak relativně snadno analyzovat na deformaci a získat tak 3D profil povrchu. Pro snížení množství analyzovaných elementů a možnost zaměření se na konkrétní oblasti obrazu v detailu bude vhodné použít mřížku nepravidelnou, která se dokáže snadno adaptovat na scénu. Nejvhodnější bude použít mřížku trojúhelníkovou, protože u trojúhelníku jako nejjednoduššího plošného objektu není třeba provádět korekce vrcholů, aby byl v prostoru plošně konzistentní. Uvážíme-li požadavek na adaptivitu (tj. proměnnou úroveň detailu) a nutnost prostorové konzistence, bude vhodné segmentovat obraz na strukturu trojúhelníků, jakou tvoří algoritmus ROAM pro reprezentaci výškové mapy [6] (princip algoritmu bude vysvětlen níže). Výšková mapa pak je ve skutečnosti druh výstupu, který budeme v algoritmu zpracovávat, protože se ve své podstatě jedná o analýzu výškové souřadnice z bodů objektu v rámci scény. V rámci modelu scény bude takto deformovaná mřížka odpovídat 3D profilu analyzované plochy objektu.



Obrázek 3.1: Algoritmus ROAM nad scénou a aproximace reprezentace scény.

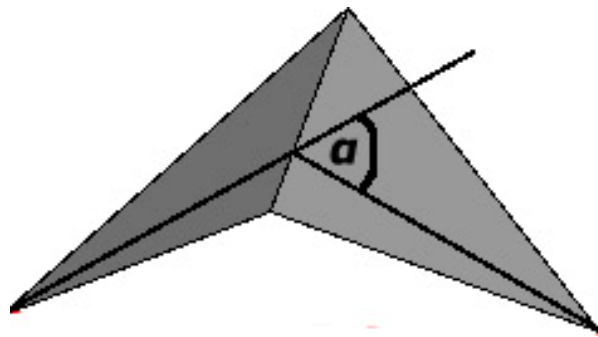
3.2 Analýza zakřivení povrchu

Trojúhelníkovou mřížku vzniklou segmentací, respektive její segmenty je třeba analyzovat, abychom byli schopni získat 3D reprezentaci scény. Zaměříme se tedy na úroveň osvětlení (intenzity) jednotlivých trojúhelníků mřížky, především pak jejich rozdílů v rámci sousedních trojúhelníků. Jestliže definujeme v jednotlivých hranách mřížky veličinu odpovídající požadované ostrosti dané hrany, tedy úrovní zakřivení povrchu mezi dvěma polygony, a budeme tuto veličinu považovat za pnutí povrchu v daném místě mřížky, bude mít síť trojúhelníků při pokusu o simulaci deformace této mřížky pnutím tendenci změnit svůj stav z rovinného povrchu, kdy pnutí v mřížce je nejvyšší, do stavu s nižší energií, tedy takového, kdy zakřivení povrchu jednotlivými hranami bude nejméně působit proti pnutí mřížky. V takovém stavu by mřížka měla mít právě minimální energii.

Uvážíme-li analogii systému mřížky k možnostem prvků lidského zraku, snadno zjistíme, že podobnou operaci by velmi efektivně dokázalo provést jednoduché pole neuronů, kdy každému

trojúhelníku by odpovídal jeden neuron, každé hraně vazba mezi sousedními neurony, se vstupem intenzity v obraze a výstupem hodnoty 3D souřadnice. Vzhledem k tomu, že mnoho jednodušších organismů než je člověk zrakem disponuje, a často jejich zrak nemá k dispozici barevnou informaci, ale pouze intenzitu světla, stálo by pravděpodobně za úvahu, jestli právě takovýto proces není základním stavebním kamenem monokulárního podnětu. Samozřejmě nelze tento složitý proces degradovat na jedinou operaci, nicméně uvážíme-li možnost přidání dalších bloků zpracování napomáhajících analýze, nabízí se velmi zajímavá možnost rozšíření systému analýzy obrazu pro adaptaci na složitější nebo reálnou scénu a získat tak spolehlivý systém počítačového vidění.

Po vyhodnocení základního 3D profilu objektů by bylo vhodné jednotlivé objekty na základě informace o hranách nebo obecně změně textury nějakým způsobem separovat, a jednotlivé objekty scény klasifikovat. Jak bylo uvedeno výše, systém jako celek by se měl vyvarovat složité matematické analýzy, aby si zachoval robustnost a rychlost. Právě v této operaci by se měl tento předpoklad projevit, a proto by měla být funkce eliminace pnutí v mřížce být co nejjednodušší rovnicí založenou pouze na změně intenzity osvětlení mezi jednotlivými segmenty. Složitější operace by zde mohla zanést nechtěná zkreslení způsobená šumem a výrazné zvýšení časové náročnosti algoritmu.



Obrázek 3.2: Zakřivení povrchu – změna osvětlení.

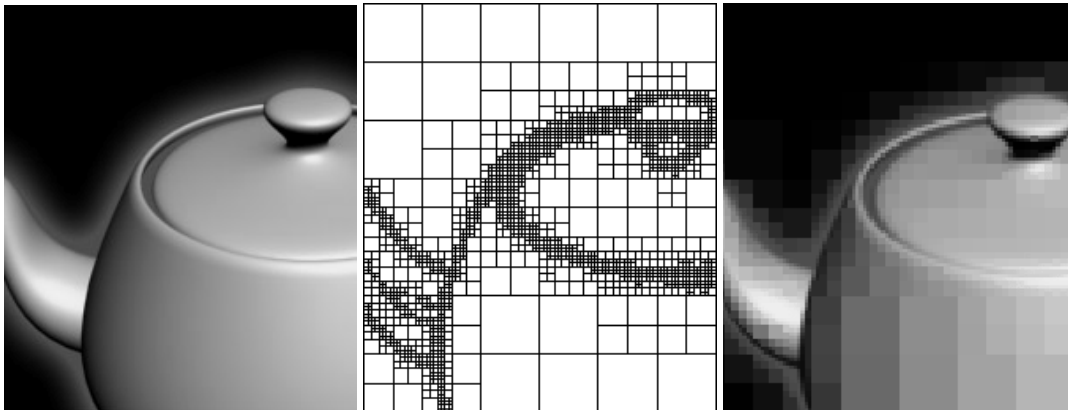
3.3 Analýza textury

Důležitou částí procesu počítačového vidění, má-li být robustní a aplikovatelný na složitou scénu, je bezpochyby analyzátor textury. Textura může výrazně ovlivnit schopnosti konvenčních metod analýzy obrazu, především těch, které jsou náchylné na výskyt šumu a zkreslení obrazu, protože se v ní často vyskytují změny tvořící falešné hrany, neodpovídající separace jednotlivých objektů v obraze a další nepříznivé vlivy. Právě kvůli těmto problémům analýzy je třeba použít při předzpracování obrazu přinejmenším základní analýzu textury. Budeme-li při základním zpracování uvažovat texturu jako rušivý vliv, cílem použití analyzátoru bude v daném segmentu obrazu typ textury detekovat a na základě této detekce upravit (snížit) úroveň detailu mřížky v daném místě tak, aby její elementy pokrývaly větší texturovanou plochu a aproximací jejich pixelů se vliv textury eliminoval.

Uvážíme-li variabilitu rastrové reprezentace textur, bude vhodné pro tento blok zpracování použít neuronovou síť, která právě s takovou nepřesností dokáže efektivně pracovat. Aby byl systém robustní a dostatečně adaptivní, bude vhodné zvolit neuronovou síť s proměnným počtem neuronů, aby se dokázala průběžně učit, a jako vstupní vektor jí nepředávat rastr obrazu, ale pouze transformačně invariantní charakteristiky oblastí. Přestože analýza textury by měla být v rámci této práce pouze navrhovaným rozšířením, pokusíme se tento algoritmus alespoň částečně realizovat pro získání představy o jeho účinnosti a vhodnosti pro systém.

3.3.1 Mřížka segmentace pro rastrové operace

Lze předpokládat, že trojúhelníková mřížka nebude příliš vhodná pro definici zpětného rastrového zpracování obrazu, proto bude vhodné v rámci projektu implementovat a experimentovat ještě na pravoúhlém modelu mřížky pro efektivní zpracování rastrových operací jakou je právě analýza textury. Pro tento model mřížky bude vhodné zvolit reprezentaci založenou na modelu **Quadtree** [8], tedy adaptivní čtvercové (obecně obdélníkové) mřížce, jejíž každý element může a nemusí být pravidelně rozdělen na 4 subelementy další úrovně. Tato mřížka umožní snadnější analýzu kompaktních rastrových segmentů při analýze textury.



Obrázek 3.3: Algoritmus QUADtree + aproximace obrazu

Nabízí se otázka, jestli by nebylo vhodné tuto rastrově přívětivější mřížku použít i na analýzu zakřivení povrchu a vyhnout se tak rozdělení na dva separované modely. Tato mřížka nevyhovuje jednomu z požadavků na model analýzy plochy, a sice tomu, že v rámci změny lokace jejích jednotlivých bodů ztrácí konzistenci v 3D prostoru (může obsahovat trhliny v místech, kde se setkávají různé úrovně detailu). Na druhou stranu jestliže by analýza byla prováděna v rámci všech úrovní detailu, mohl by být tento problém minimalizovaný na relativně zanedbatelné hodnoty a i taková analýza by mohla zprostředkovat dostatečné výsledky. Zároveň jak je vidět na příkladu tento typ mřížky generuje lokálně skokové přechody, které se příliš nehodí pro analýzu spojitého povrchu.

Trojúhelníková mřížka je tedy vhodnější, protože její elementy nabízí celkem 8 směrů hran pro aproximaci průběhu polygonů scény oproti 4 směrům této mřížky.

3.4 Geometrie scény a objektů

Přestože by analyzovaná scéna měla být obecná, tedy její geometrie a geometrie jednotlivých objektů není předem známá, lze učinit několik předpokladů, které by mohly být významné z hlediska analýzy. Podíváme-li se kolem sebe, snadno zjistíme, že člověkem vytvořené objekty obsahují několik základních geometrických pravidelností. Zaměříme-li se konkrétně více na lidský zrak, můžeme pozorovat fakt, že i lidská mysl je na tyto vzory poměrně striktně naučená a snaží se je interpretovat mnohdy i v místech, kde ve skutečnosti nejsou. Na tomto jevu je založený takzvaný **optický klam** [21], tedy ovlivnění vidění pravidelnostmi v obraze, které zrak vyhodnotí jako dobře známý vzor pouze na základě shodné 2D reprezentace (zrakového vjemu). Tyto předpoklady našemu vidění často přinášejí mnoho informací o pozorované scéně, především při chybějící 3D informaci jako například sledování obrazovky nebo prohlížení fotografií. Negativní vliv vlastního optického klamu je minoritní v rámci množství pozorovaných objektů, a proto nepřináší problémy vidění jako celku.

Jestliže zahrneme geometrické předpoklady do procesu analýzy obrazu, můžeme získat důležité zlepšení vlastností systému a přiblížit jej lidskému vidění se všemi výhodami i nevýhodami z toho vyplývajícími. Ačkoli by se mohla náchylnost systému na optický klam jevit jako negativní hledisko, nemusí tomu být nutně tak. Je zde nutné především ujasnit si, co si budeme představovat pod pojmem **počítačové vidění**.

První variantou je, že takto budeme označovat proces získání reprezentace scény obecně jiný, než jak jej vidí člověk, samozřejmě pokud možno správný, ovšem reprezentovaný viděním stroje i s jeho interpretační skutečností. Druhou možností je uvažování implementace a zpracování lidského vidění jako procesu v rámci strojových operací, tedy proces kdy počítač si vyloží scénu stejným způsobem jako by si ji vyložil člověk.

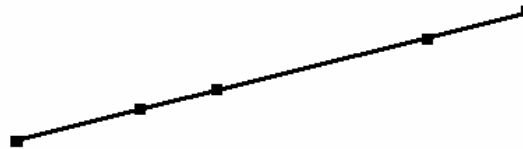
V této práci budeme uvažovat variantu druhou, protože jestliže v budoucnu budeme chtít, aby lidé s stroje (především roboti) navzájem interagovali, měli by vnímat prostředí stejným způsobem. První způsob je vhodnější spíše pro průmyslové nasazení kde je optický klam samozřejmě nechtěný. V rámci této práce by tedy aplikace vzorů geometrie měla být přínosem v procesu analýzy obrazu.

Vzory, které můžeme často ve scéně pozorovat, se týkají všech úrovní rozměrového prostoru. Zde ihned přeskočíme bezrozměrný objekt, tedy bod, protože ten z hlediska geometrie sám o sobě nepřináší vzor, který by byl dále analyzovatelný. Posuneme se tedy rovnou na jednorozměrné objekty (křivky). Přestože objekty jako takové mají svůj počet rozměrů, mohou se všechny vyskytovat obecně v prostoru, například křivku budeme pro jednoduchost zařazení umísťovat do jednorozměrných objektů, přestože ve skutečnosti může být vícerozměrně definovaná.

3.4.1 Jednorozměrné geometrické útvary

Jednorozměrné útvary jsou často hranami objektů nebo konkrétními objekty, jejichž délka je mnohem větší než ostatní rozměry a které se v projekci projevují typicky šířkou jediného pixelu. Nejčastějšími útvary, jaké lze nalézt ve většině scén, jsou tyto:

- **Úsečka** (line) – Množina bodů ležících na jedné přímce, tvoří přímou hranu. V projekci se projevuje také jako úsečka. Vyskytuje se častěji u člověkem vyrobených objektů, kde je přítomna velmi často.



Obrázek 3.4: Úsečka.

- **Kruhový oblouk** (arc) – Množina bodů ležících na kružnici v jedné rovině, typicky tvoří hranu objektů kulatého tvaru. V projekci se projevuje jako elipsa nebo část elipsy. Obecně není předem z projekce poznat, je-li ve scéně elipsa nebo kružnice, nicméně v reálném prostředí je kružnice mnohem častější a proto i lidské vidění má tendenci upřednostňovat kružnici pokud není v rozporu s konzistencí scény. Vyskytuje se většinou u člověkem vyrobených objektů, v přírodě téměř výhradně jako kružnice u pravidelných objektů.



Obrázek 3.5: Kruhový oblouk.

- **Hladká křivka** (spline) – Výsledkem minimalizace energie zakřivení hrany nebo plochy je hladká křivka, ve scéně pak množina bodů na takové křivce ležících. Typicky reprezentuje průřez nebo hranu zaoblených objektů, vyskytuje se v přírodních útvarech i člověkem vyrobených objektech. V projekci je reprezentována opět hladkou křivkou. Je možné ji využít pro eliminaci chyb způsobených šumem a optimalizací analýzy.

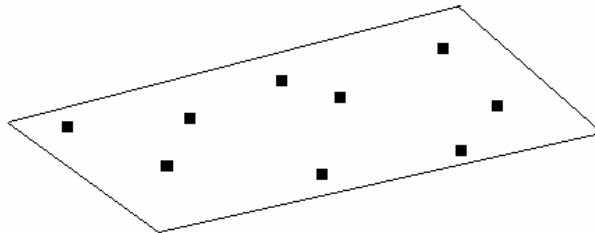


Obrázek 3.6: Křivka.

3.4.2 Dvourozměrné geometrické útvary

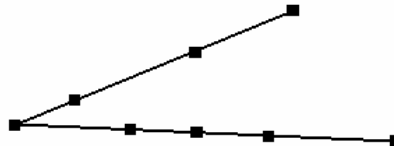
Dvourozměrné objekty v prostoru mohou být roviny (včetně zakřivených) nebo soustavy jednorozměrných objektů tvořících jednu rovinu. Nejpodstatnějšími geometrickými pravidelnostmi dvourozměrných objektů jsou:

- **Body v rovině** (plane) – Často lze najít a detekovat případy, kdy množina bodů leží v jedné rovině, opět především u lidských výtvorů. Takové body mohou tvořit obecnou hranu, přičemž většina hran objektů leží právě v jediné rovině a na tento fakt je také lidský zrak naučený.



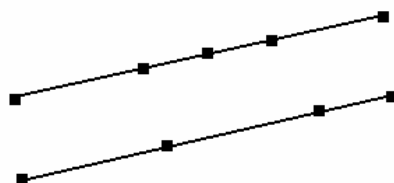
Obrázek 3.7: Body v rovině.

- **Roh** (corner) – Roh je obecně bod setkání dvou úseček nebo křivek (horší analýza). Je dominantou především člověkem vyrobených objektů. Jeho určujícím faktorem pro geometrii je úhel, které úsečky svírají a lidské vidění je naučené, aby se nejčastěji snažilo připodobnit geometrii rohu k pravému úhlu (90°). Pravý úhel je také nejčastěji používaný člověkem, což implikuje předchozí fakt. V projekci je roh reprezentován rohem s obecným úhlem.



Obrázek 3.8: Roh.

- **Rovnoběžky** (parallel) – Rovnoběžky jsou obecně dvě nebo více různých úseček se stejnou směrnici v prostoru. Stejně jako roh i rovnoběžky jsou častým jevem a vjemem v rámci lidské činnosti a vidění, nepřímou souvisí s použitím pravých úhlů na objektech, jejichž jsou výsledkem. V projekci jsou rovnoběžné úsečky také rovnoběžné.



Obrázek 3.9: Rovnoběžky.

- **Paralelní křivky** (parallel curves) – Podobně jako rovnoběžky lze analyzovat obecné paralelní křivky. Jedná se v zjednodušeném podání o soustavu navazujících dvojic rovnoběžek. S výjimkou soustředných kružnic jsou tyto dominantou člověka a jím vyrobených objektů. V projekci se projevují jako paralelní, leží-li ve stejné rovině, pokud leží v zakřivené rovině dané křivkami, mají v projekci stejný nebo podobný tvar.



Obrázek 3.10: Paralelní křivky.

3.4.3 Vícerozměrné geometrické útvary

V rámci analýzy scény by bylo analyzovat útvary s vyšší prostorovou dimenzí, nicméně vstupní 2D scéna ve skutečnosti nedává dostatečnou informaci o více než dvourozměrných objektech, a proto se vícerozměrnými objekty nebudeme zabývat. Tuto problematiku bude řešit další část systému a kapitola týkající se klasifikace objektů, která s geometrií úzce souvisí. Důležitým předpokladem zde bude pouze nutnost analyzovat základní geometrická primitiva, podobně jako geometrické vzory redukuje pouze na ty základní a detekovatelné.

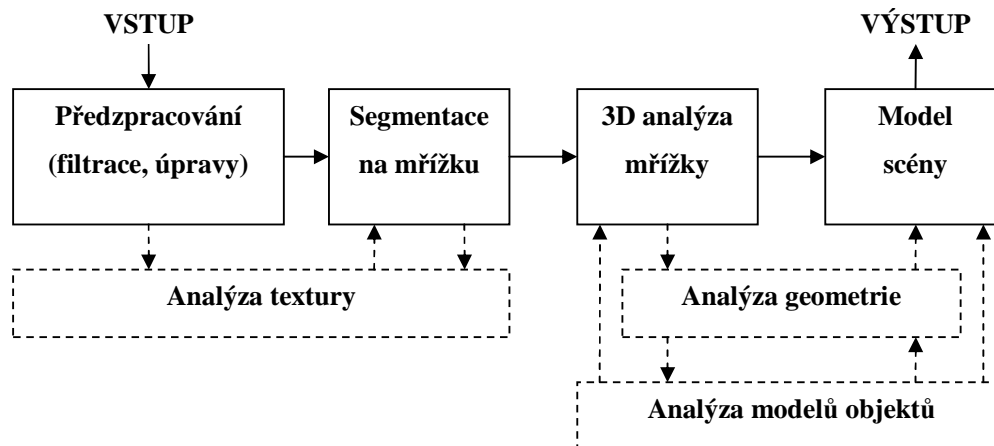
3.5 Klasifikace objektů

Jestliže máme k dispozici alespoň aproximaci základního 3D profilu objektu, je vhodné se jej pokusit klasifikovat. Klasifikace objektů ve scéně je důležitá jednak z hlediska detekce objektů, ale také umožňuje upřesnit tvar objektů na základě jejich modelů uložených v databázi známých objektů.

Klasifikace objektů by měla být volitelným dodatečným modulem celého systému. Bylo by vhodné tento proces aplikovat v rámci operací pracujících s neurčitostí kvůli robustní analýze, nejlépe s využitím neuronové sítě. Zásadním problémem při korelaci modelu zde bude nalezení počátku modelu vzorového a analyzovaného povrchu tak, aby byl dostatečně invariantní vůči prostorové transformaci a jednoznačně detekovatelný v rámci analýzy scény. Zde bude pravděpodobně vhodné využít lokálních maxim a minim zakřivení povrchu jako charakterizujících bodů pro korelaci pozice modelů. Stejně jako analýza textury i toto je součástí pouze pro prozkoumání možností systému, a proto bude předmětem návrhu předpokladů pro vylepšení systému a, v závislosti na úspěchu analýzy implementovaných součástí, také experimentů.

3.6 Souhrn systému analýzy

Nyní bude vhodné shrnout systém analýzy jako celek, včetně základního blokového schématu.



Obrázek 3.11: Schéma systému analýzy.

Na vstup systému tedy nejprve postavíme základní rastrové operace, analýzu textur, a segmentaci na mřížku, v ideálním případě se zpětnou vazbou mezi jednotlivými operacemi. Následovat bude analýza mřížky s vyhodnocením 3D profilu objektů. Analýza mřížky by měla úzce spolupracovat s moduly analýzy geometrických vzorů a oddělení jednotlivých objektů s využitím informace o hranách. Posledním prvkem analýzy by pak měla být klasifikace jednotlivých objektů v obraze, opět s využitím informace z předchozích kroků a aplikací neuronové sítě aplikovaná do modelu scény.

Systém jako celek by měl v ideálním případě mít možnost zpracovat následující typy scén s požadovanou kvalitou zpracování a využitím konkrétních prvků:

- **Jednoduchá syntetická scéna** – V základním procesu úspěšná analýza 3D profilu objektů, v případě zahrnutí modulu pro detekci objektů jejich klasifikace.
- **Jednoduchá texturovaná scéna** – S využitím analyzátoru textury eliminace vlivů textury na analýzu geometrie a pokus o další zpracování jako jednoduché scény bez textur. Bez klasifikace textury přibližná analýza povrchu na menší úrovni detailu.
- **Plně texturovaná scéna** – Schopnost oddělení objektů od texturovaného pozadí a další analýza jako jednoduchá texturovaná scéna. Tento typ scény již bude vyžadovat navrhovaná rozšíření.

- **Reálná scéna** – Schopnost eliminace šumu aproximací a zpracování textur reálné scény, vliv reálného osvětlení a nepřesnosti povrchu. Systém by měl využívat všech navrhovaných součástí.

Lze usoudit, že všechny výsledky nebudou podle předpokládání, protože systém zpracování obrazu je složitý, nicméně pokud dojde alespoň k úspěšné analýze jednoduché scény, lze projekt považovat za úspěšný. Důraz bude kladen na striktní oddělení jednotlivých kroků kvůli možnosti oddělení konkrétních modulů, vložení jiných modulů nebo dokonce kompletní změnu procesu analýzy.

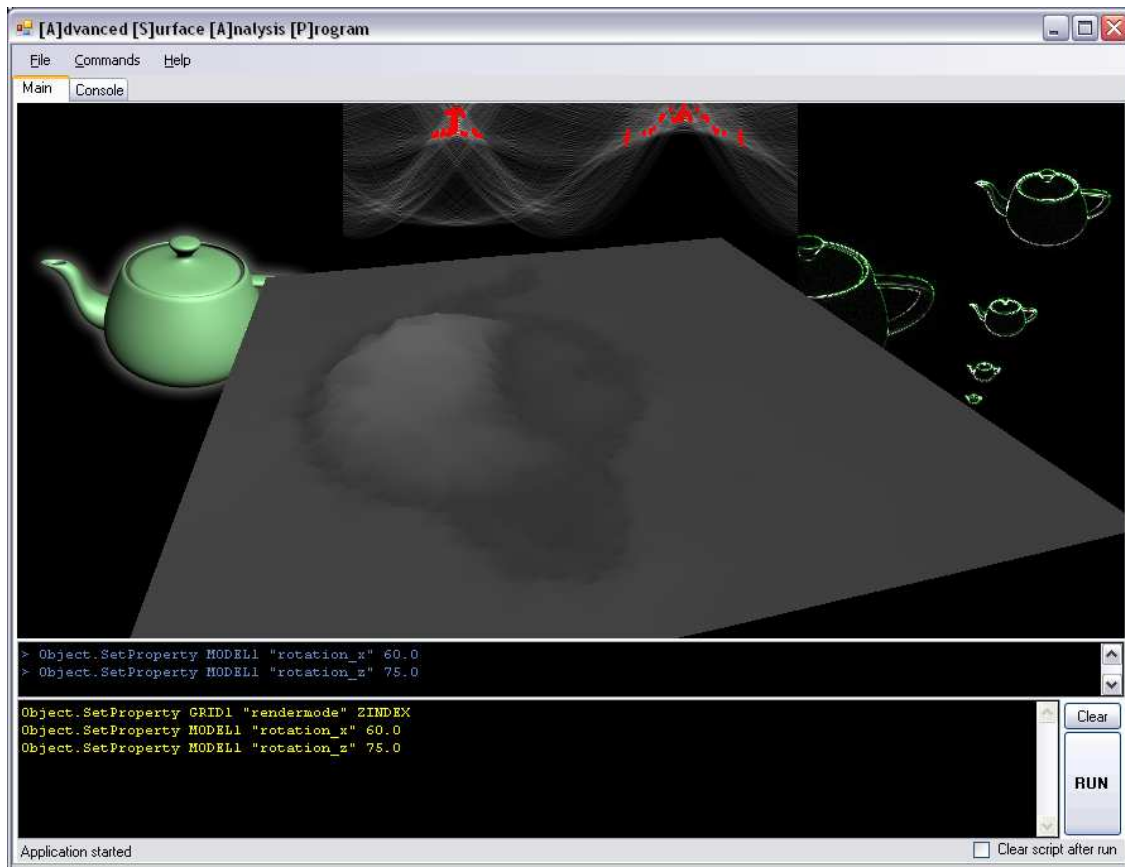
Vzhledem k předchozím zkušenostem s rozhraním programu vytvořeného v rámci bakalářské práce [1] a faktu, že požadovaná variabilita systému nemůže být zajištěna pevnou strukturou ovládacích prvků na formuláři aplikace, bude použito jednoduché konzolové rozhraní a systém jako celek bude ovládán pomocí příkazů konzole. Grafický výstup pak bude zajištěný s pomocí 3D grafického rozhraní OpenGL aby bylo možné snadno vykreslovat obrazy i 3D modely analyzované scény v požadovaném prostorovém pohledu pro ověření práce algoritmu.

4 Implementace systému

System jako celek představuje jednoduché rozhraní sestávající z textové konzole a okna grafického výstupu. Program je vytvořen v prostředí Microsoft Visual Studio .NET 2005, a to ve dvou programovacích jazycích. Základní uživatelské rozhraní je vytvořeno v jazyce C# kvůli snadnějšímu vývoji a kompatibilitě ovládacích prvků se systémem Windows. Knihovny na pozadí jsou pak vytvořeny v jazyce C++ z důvodu interoperability s grafickým rozhraním OpenGL a také lepší přenositelnosti. Zároveň je tak program přívětivější pro nízkourovňové operace a dovoluje v budoucnu snadno zahrnout optimalizace na úrovni instrukcí procesoru v jazyce assembler, do budoucna se předpokládá především využití rozšíření SSE pro blokové operace s obrazem popřípadě vertex shaderů pro hromadné vektorové operace. System sestává z následujících součástí:

- **ASAPMain** – Hlavní rozhraní programu, Windows aplikace.
- **GLBase** – Základní definice rozhraní a objektů pro komunikaci s OpenGL rozhraním.
- **GLC** – Knihovna s komponentou OpenGL okna umožňující grafický výstup programu.
- **ASAPLib** – Knihovna systému analýzy, obsahující veškeré algoritmy, struktury, prostředí a rozhraní potřebné pro analýzu. Komunikace s knihovnou je zajištěna přes konzolové rozhraní a grafický výstup navázaný na OpenGL okno z předchozí knihovny.

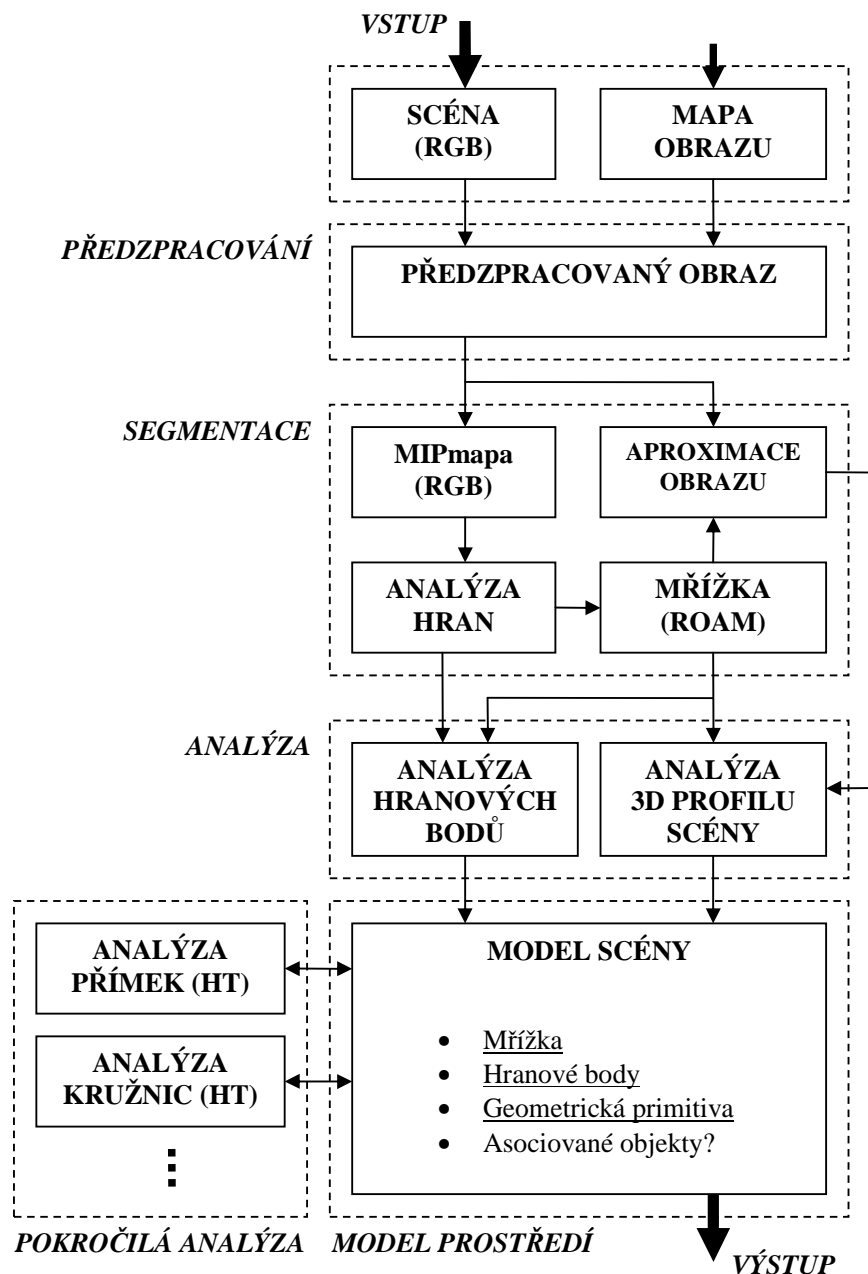
Proces analýzy programu ASAP (**A**dvanced **S**cene **A**nalysis **P**rogram) je ovládán konzolovým skriptem. Podrobnosti k ovládání programu a příkazům skriptu viz **Příloha 1 – Manuál programu ASAP**. Podrobnosti ke kódu programu viz **Příloha 2 – Programová dokumentace** na přiloženém CD. Je kladen důraz na jednoduchost rozhraní a veškeré operace analýzy, včetně nastavení vlastního výstupu zobrazení jsou řízeny zpracovávaným skriptem. Program vyžaduje pro běh grafickou kartu s podporou OpenGL, nainstalovaný .NET Framework 2.0 a z důvodů některých nízkourovňových operací analýzy také práva superuživatele.



Obrázek 4.1: Hlavní okno programu.

4.1 Proces analýzy

Proces analýzy obrazu je do značné míry ovlivněn provádějícím skriptem a je možné jej dodatečně modifikovat. Probereme zde tedy proces založený na návrhu systému a provedených experimentech, který se vzhledem k současným vlastnostem systému jeví jako nejvhodnější a přináší nejlepší výsledky. Podle původních předpokladů se ukázalo, že analýza založená striktně na jednom přístupu nenabízí dostatek prostředků k eliminaci některých nepříznivých vlivů vlastností scény (respektive vstupního obrazu). Proto celkový proces analýzy kombinuje několik přístupů pro dosažení ideálních výsledků. Proces jako celek se drží původního návrhu, přičemž se snaží v maximální míře využít již zpracované informace tam kde je to možné. Znázornění procesu jako celku je na obrázku 4.2. Jednotlivé celky systému analýzy, jak byly popsány v úvodu práce, jsou znázorněny čárkovně.



Obrázek 4.2: Model procesu analýzy.

Základní analýza scény probíhá následovně: Nejprve je načtena scéna ze souboru ve standardním barevném modelu RGB, barva scény je zachována kvůli možnosti získání většího množství informace. Volitelně se načítá mapa oblastí zpracování pro potlačení nebo zvýraznění některých částí obrazu, touto mapou v rozsahu $\langle 0;1 \rangle$ se vstupní obraz jednoduše násobí. Poté je obraz scény převeden na MIPmapu pro vytvoření základu pro algoritmy pracující s pyramidální optimalizací a také analýzu obrazu v různých úrovních detailu. Na MIPmapu je aplikován hranový operátor pro získání hranových bodů v různých úrovních detailu. Na základě hranových bodů

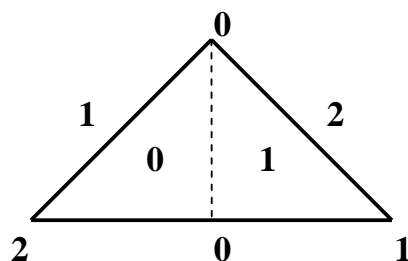
konkrétních úrovní se poté dělí základní trojúhelníková mřížka algoritmem ROAM [6] pro dosažení požadované úrovně detailu. Mřížka aproximuje části obrazu, proto je hned po jejím vytvoření provedena rastrová statistika nad obrazem pro jednotlivé její elementy. Výsledkem je souhrnná informace o barvě a charakteru oblasti pro každý takový element mřížky, včetně odpovídajících rodičovských elementů vyšších úrovní. Následuje analýza zakřivení povrchu na základě rozdílů právě těchto získaných aproximací prvků mřížky. Výsledkem analýzy zakřivení je 3D profil povrchu objektů scény založený čistě na osvětlení. Zde končí základ v této práci vytvářeného algoritmu a následuje analýza podrobnější, zkoumající možnosti a vhodnost rozšíření, které by bylo možné v budoucnu na algoritmus dále aplikovat. Základem této vektorové analýzy je detekce hranových bodů mřížky na základě již existující MIPmapy hran (pouze jedné nižší úrovně, protože tato plně dostačuje). V hranových bodech jsou pak experimentálně vyhledávány vzory geometrických primitiv, které by mohly být užitečné při podrobnější analýze využívající geometrii. Analyzovány jsou vzory úseček a kruhových oblouků. V poslední části je pak odděleně testován analyzátor textury jako samostatný modul, který by mohl být do systému přidán později pro zvýšení kvality analýzy ve složitějších scénách.

4.2 Segmentace obrazu

Segmentace obrazu je v projektu provedena nepravidelnou trojúhelníkovou mřížkou. Výstupní mřížka je v rámci geometrických transformací (především posunu souřadnic bodů v ose Z) konzistentní v 3D prostoru, a proto je dále přímo použita pro analýzu 3D profilu scény. Mřížka je založena na algoritmu ROAM [6] pro segmentaci výškové mapy a její úroveň detailu je určena na základě analýzy víceúrovňového hranového modelu scény.

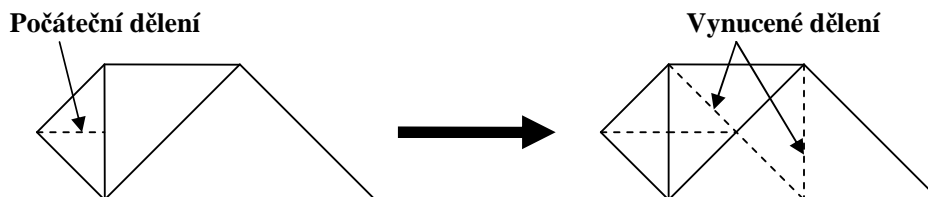
4.2.1 Algoritmus ROAM

Algoritmus ROAM v původní verzi slouží k segmentaci výškové mapy s proměnlivou úrovní detailu. Název algoritmu pochází ze zkratky **Realtime Optimally-Adapting Meshes** [6] (adaptivní mřížka pro zpracování v reálném čase) a jejím účelem je minimalizace počtu objektů pro vykreslení scény zakřivené plochy. V této aplikaci je stejná mřížka využita pro jiný účel, nicméně je využita právě pro její základní vlastnosti: **adaptivitu** detailu (snížení počtu elementů) a **konzistenci** v 3D prostoru. Základním elementem mřížky je trojúhelník, který může být obecně libovolný, ovšem ve většině aplikací z důvodů optimalizace je základním prvkem rovnostranný pravoúhlý trojúhelník. Prvním krokem algoritmu je rozdělení oblasti (typicky pravoúhlé) na základní síť trojúhelníků, úroveň detailu 0. Poté jsou jednotlivé trojúhelníky rekurzivně děleny na menší části, dokud není dosažena v konkrétním místě potřebná úroveň detailu. Každý trojúhelník je rozdělen na dva menší trojúhelníky rozdělením přepony prostředním bodem viz obrázek 4.3. Zároveň jsou znázorněny indexy vrcholů, hran (nebo také přilehlých trojúhelníků) a výsledných podřízených trojúhelníků mřížky.



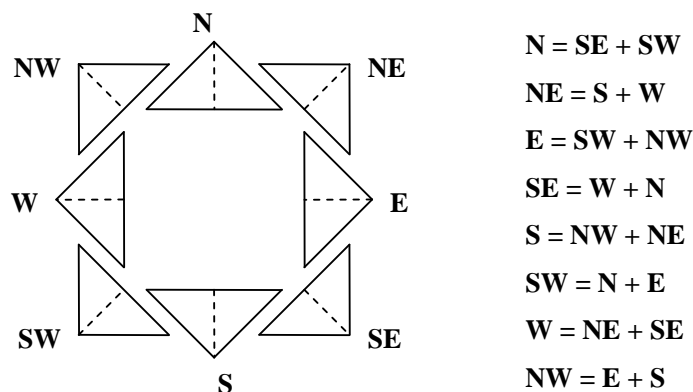
Obrázek 4.3: Dělení trojúhelníku mřížky.

Pro zajištění konzistence plochy v případě změny souřadnice Z dělicího bodu je nutné eliminovat uzly sítě tvaru T, aby v takovém případě nevznikaly nechtěné “trhliny”. Toho je docíleno rekurzivním dělením trojúhelníka sousedícího s dělicí hranou. Rekurzivní dělení sousedních trojúhelníků probíhá, dokud nejsou eliminovány všechny vzniknuvší uzly tvaru T s výjimkou krajních uzlů sítě (takových, které nemají sousední trojúhelníky). Taková operace se nazývá **vynucené dělení** (forced split) a musí pokračovat do té doby, dokud není zachována konzistence mřížky.



Obrázek 4.4: Rekurzivní dělení mřížky.

V případě, že potřebujeme informaci o poloze jednotlivých trojúhelníků kvůli dalšímu zpracování (například zpětné vazbě na rastr), je vhodné již při procesu dělení sítě u jednotlivých trojúhelníků registrovat jejich směr. Za směr trojúhelníka budeme označovat polohu vrcholu 0 (protilehlý vrchol k přeponě) vzhledem ke středu trojúhelníka. Tento pak intuitivně tvoří šipku ve směru, který zaujímá. Pro zjednodušení použijeme zeměpisné značení světových stran se stejným významem jako na plošné mapě (sever nahoře). Základních směrů trojúhelníků v pravoúhlé mřížce je 8 viz obrázek 4.5, pro lepší ilustraci jsou znázorněna i jednotlivá dělení. Jak je vidět na obrázku, směr trojúhelníka při dělení na větší úroveň detailu přímo určuje směr podřízených trojúhelníků, při bližším pozorování je patrné že směry po dělení jsou dva nejbližší protilehlé směry v rámci různé směry k původnímu směru trojúhelníka. Je tedy možné tuto informaci v rámci algoritmu snadno distribuovat. Směry jsou pro lepší ilustraci označeny odpovídajícími světovými stranami při zobrazení plošné mapy.



Obrázek 4.5: Směry trojúhelníků a závislost směru na dělení.

Při podrobnějším prozkoumání mřížky jako celku snadno zjistíme, že veškeré její elementy jsou tvořeny právě trojúhelníky v těchto základních směrech. Fakt znalosti polohy trojúhelníka je důležitý v případě, že hledáme element mřížky v konkrétním bodě (zadaném souřadnicemi, nikoliv polohou v rámci struktury mřížky) nebo potřebujeme v tomto bodě zajistit požadovanou úroveň detailu. Vyhledávání místa pak se znalostí směru a aktuální úrovní snadněji určuje, ve kterém z podřízených trojúhelníků požadovaný bod leží a je tak zachována logaritmická složitost vyhledávání v rámci hierarchie mřížky. Tuto výhodu často využívá algoritmus použitý v tomto projektu, protože právě vyhledávání konkrétního místa mřížky je častá operace. Stejnou vlastnost využívají i další postupy, například aproximace rastru elementem mřížky. Zřejmě není nutné zachovat pravoúhlost a rovnostrannost trojúhelníků, stejnou pravidelnost by mělo vykazovat i dělení obecně definovaného trojúhelníka.

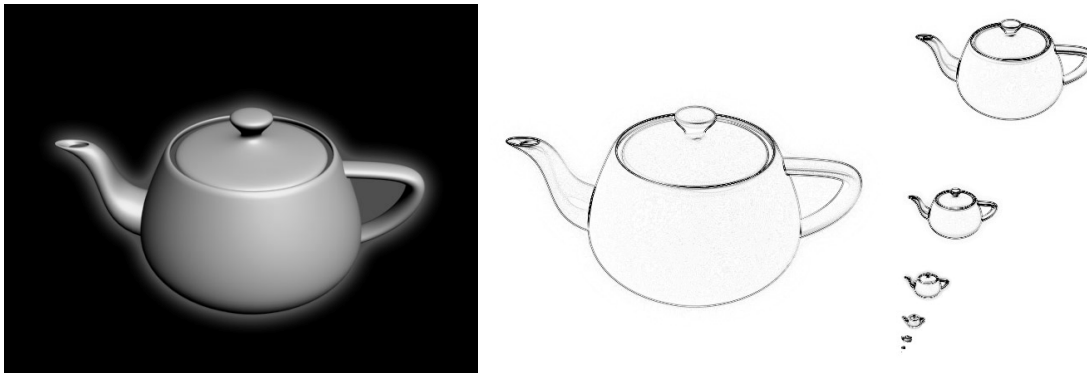
4.2.2 Adaptivní dělení mřížky

Adaptivní dělení mřížky v projektu je založeno na detekci hran v obraze. Detekovaná hrana je obecně místo v obraze, kde rozdíl mezi sousedními pixely (nebo obecně oblastmi) přesahuje zadaný limit. Vzhledem k tomu, že hrana nemusí být nutně ostrá, tj. vyskytovat se přímo mezi sousedními pixely, je nutné pojem hrany a její detekci generalizovat. V ideálním případě by měla být jako hrana detekován i pozvolný přechod mezi dvěma oblastmi s různou světlostí povrchu nebo barvou. Proto není v projektu pro detekci hran využít pouze obraz na úrovni vstupního rozlišení, ale je použito výše zmíněné pyramidální zpracování a detekce hran v MIPmapě a jejích jednotlivých úrovních. Výhoda detekce hran ve snížené úrovni detailu je právě detekce pozvolných přechodů, která v nižší úrovni rozlišení přechází na ostrou hranu detekovatelnou hranovým operátorem s relativně malým jádrem.

Proces dělení mřížky tedy sestává z vytvoření víceúrovňového obrazu hran a jeho aplikaci na stávající model mřížky ve všech místech, kde hrana přesahuje zvolený práh. Při tvorbě systému bylo vyzkoušeno několik hranových operátorů, nicméně jako nejvhodnější se jeví jeden z nejjednodušších

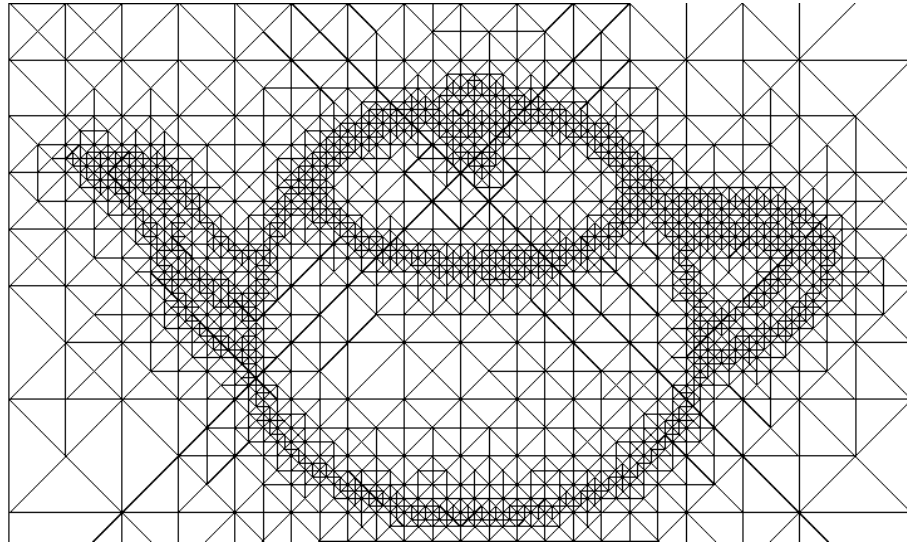
a sice všesměrový operátor rozměru 3x3 založený na průběhu Gaussovy křivky viz rovnice 4.1, kde \mathbf{N} je matice hranového operátoru, \mathbf{P} výsledná hodnota pixelu po konvoluce, \mathbf{N}_i hodnota konkrétního místa operátoru, a \mathbf{I}_i hodnota odpovídajícího obrazového pixelu. Výhodou tohoto operátoru je jednak všesměrovost a jednak fakt, že produkuje úzké (pouze 1 pixel široké) hrany a tak nevyžaduje dodatečné operace pro snížení náročnosti segmentačního algoritmu tvorby mřížky jako celku. Jak se ukázalo, běžná konvoluce s hranovým operátorem nepřináší dostatečné výsledky detekce hran na všech úrovních intenzity osvětlení. Proto je použita adaptivní konvoluce zesilující hrany v horní části rozsahu intenzity, kde jsou menší rozdíly osvětlení patrnější jako hrana než v oblastech tmavých. V rovnici (4) je vzorec pro úpravu hodnoty hranového pixelu pro rozsah hodnoty pixelu $\langle 0;1 \rangle$. Koeficient změny hodnoty pixelu na základě jasu obrazu byl určen experimentálně pozorováním výraznosti hrany při konkrétní změně intenzity při dané úrovni jasu.

$$N = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix} \quad (2) \quad P = \sum_{i=0}^9 N_i I_i \quad (3) \quad P_A = P \cdot \frac{\sum_{i=0}^9 \frac{1}{9} I_i + 1}{5} \quad (4)$$



Obrázek 4.6: Hranový operátor a detekce hran v obraze.

Obraz hran je pak v rámci jednotlivých úrovní detailu zpracován a pro každý bod hrany je v mřížce zajištěno, aby úroveň detailu mřížky v daném místě odpovídala požadované úrovni detailu na základě rozlišení zpracovávaného obrazu hran. Výsledkem po projití všech úrovní MIPmapy hran je adaptivně segmentovaná mřížka založená na úrovni detailu v rámci obrazu viz obrázek 4.7.



Obrázek 4.7: Adaptivní mřížka nad obrazem.

4.3 Analýza zakřivení povrchu

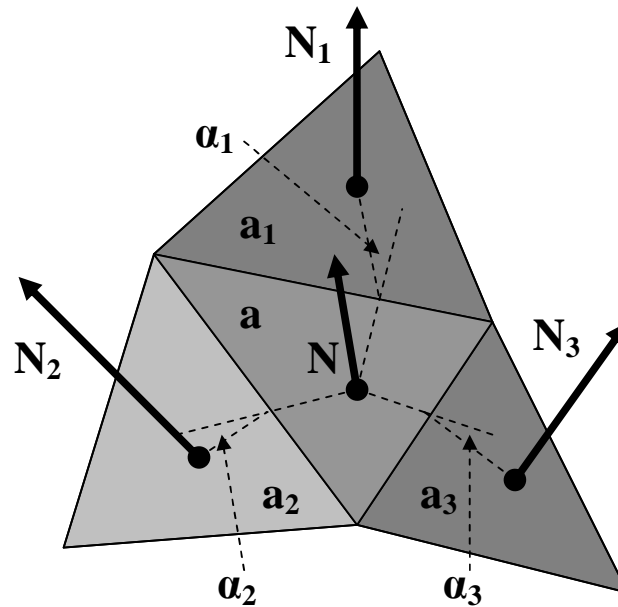
Ihned po segmentaci obrazu na mřížku se provádí analýza zakřivení povrchu mřížky. Protože mřížka sama o sobě nemá kromě úrovně detailu informace o vlastním obsahu obrazu, je nutné před vlastním procesem analýzy provést nad obrazem proces rastrové analýzy, kdy pro každý segment mřížky (trojúhelník) provádíme vytvoření základních statistik oblasti kterou v rámci zdrojového obrazu pokrývá. Tyto statistiky typicky obsahují průměrnou barvu, sumu hodnot pixelů, sumu druhých mocnin pixelů, volitelně histogram, počet pixelů a další.

V rámci základní analýzy je využita pouze informace o průměrné hodnotě pixelů (barvě, kterou oblast aproximuje), ostatní hodnoty jsou získávány kvůli případným budoucím požadavkům jiných modulů. Přestože model analýzy není založen přímo na intenzitě pixelů (úrovni světlosti v odstínech šedi), vysvětlíme jeho proces pro zjednodušení právě na této veličině. Ve skutečnosti je pak rozdíl intenzit použit jako maximum rozdílů intenzit v jednotlivých barevných složkách segmentů.

Aby vlastní analýza mohla fungovat, musíme učinit předpoklad, že intenzita osvětlení zakřiveného povrchu se mění s velikostí odklonu jeho normály od směru světelného zdroje, popřípadě směru pozorovatele. Právě zde je místo, kde se pokusíme eliminovat veškerou složitou matematickou analýzu, která by mohla celý proces zkomplikovat a pokusíme se celý proces degradovat na co nejjednodušší operace vzhledem k množství informací, které o scéně máme k dispozici.

Především je třeba si uvědomit, že prakticky jediná informace kterou o zatím neznámé scéně máme, je právě barva a intenzita jednotlivých pixelů (oblastí) a jejich úroveň detailu. Proto ani při nejlepší vůli nemůžeme zohlednit při analýze neznámou pozici nebo typ osvětlení, tvar objektů, ale ani mnoho dalších faktů, které jsou v podstatě informacemi, které by vlastní proces měl zjistit. V případě sekvenční analýzy by samozřejmě tyto informace mohly figurovat jako jakési inicializační

hodnoty dalšího kroku, nicméně základní proces nepředpokládá jakoukoliv předchozí znalost scény. Zde je opět na místě analogie s lidským zrakem, který stejně jako algoritmus, který se zde snažíme vytvořit, dokáže při pozorování jediného 2D obrazu rekonstruovat jeho obsah „pouze“ s využitím základních operací zprostředkovaných elementy jeho sítě zpracování – neurony. Proto se pokusíme založit základní analýzu zakřivení mřížky pouze na jednoduché funkci pracující jen s hodnotami barvy sousedících elementů plochy. Na obrázku 4.8 je znázorněn základní koncept zakřivení plochy vzhledem k sousedním trojúhelníkům, kdy úhel α_i závisí na vzájemném odklonu normál povrchu \mathbf{N} a \mathbf{N}_i , jehož velikost je definována rozdílem intenzit elementů \mathbf{a} a \mathbf{a}_i .

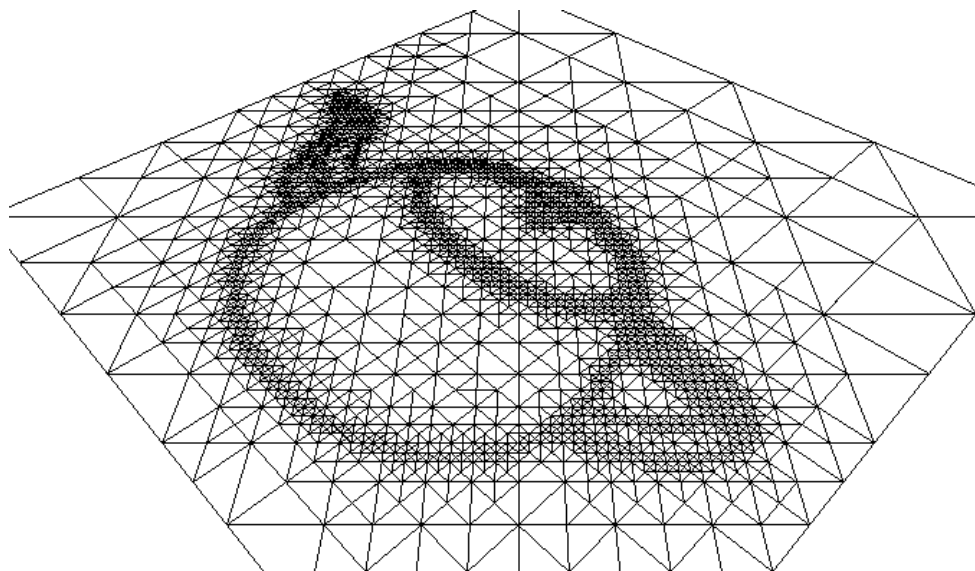


Obrázek 4.8: Definice zakřivení plochy

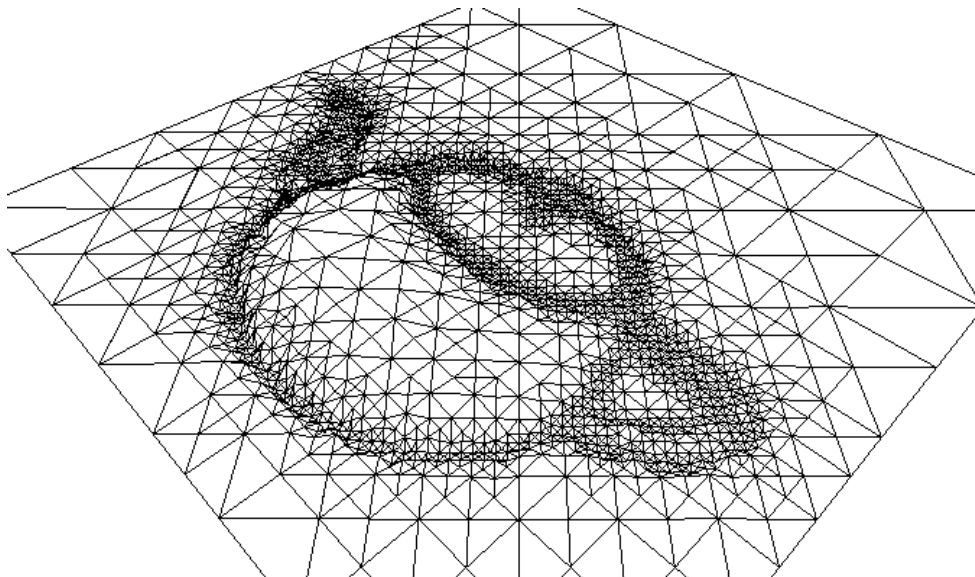
Proložíme-li rovinou $\mathbf{Z} = \mathbf{0}$ mřížku a definujeme-li potřebu mřížky zakřivit se o určitou hodnotu úhlu v konkrétním místě závislou úměrně na změně intenzity osvětlení povrchu, vznikne v takové mřížce virtuální **pnutí** (nepřirozený nestabilní stav), snažící se minimalizovat energii mřížky zakřivením jejího povrchu. Pokud uvolníme souřadnici \mathbf{Z} bodů mřížky, v případě reálného modelu mřížka zaujme pozici s minimální energií odpovídající předpokládanému průběhu 3D plochy scény. Tento proces nazveme **kolaps** mřížky (analogie zhroucení materiálu objektu v případě porušení podmínek pro stabilizaci pnutí).

V prostředí programu je tento proces simulován cyklickým výběrem elementu mřížky a provedením operace kolapsu nad tímto elementem. Způsobů výběru elementů je několik, experimentálně bylo zjištěno, že nejlepší výsledky vzhledem k charakteru algoritmu přináší náhodný výběr s rovnoměrným rozložením v souřadném systému mřížky. Kolaps trojúhelníku je pak proces, při kterém vrcholy trojúhelníku zaujmají novou polohu souřadnice \mathbf{Z} tak, aby jeho hrany

minimalizovaly pnutí mřížky vzhledem k sousedním elementům mřížky. Cyklické opakování procesu přes všechny elementy mřížky dospěje v určitém stádiu procesu k bodu, kdy mřížka jako celek má minimální energii a operace kolapsu osciluje kolem této hodnoty nebo se v ideálním případě mřížka ustálí na minimální hodnotě energie v rámci všech svých segmentů. Tento stav nelze přímo detekovat, nicméně experimenty ukázaly, že je vhodnější proces ovlivnit časovým intervalem nebo konkrétním počtem iterací a získat tak dostatečnou přesnost analýzy pro další postupy. V případě zvýšeného počtu iterací by bylo třeba definovat v mřížce stabilní bod nebo body, aby tato jako celek neměla tendenci vystupovat do prostoru právě při oscilaci kolem minima. Příklad mřížky na obrázku 4.9 po provedení základního kolapsu řízeného pouze intenzitou je na obrázku 4.10.

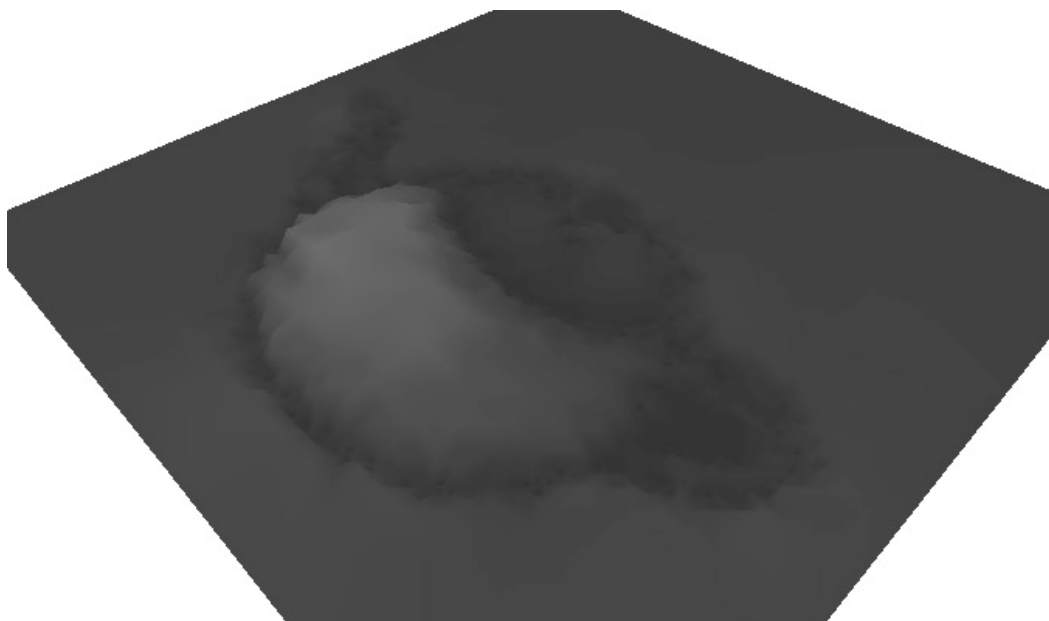


Obrázek 4.9: Mřížka před kolapsem.



Obrázek 4.10: Mřížka po kolapsu.

Jak se v rámci experimentů ukázalo, kolaps mřížky funguje korektně i v případě, že objekt není v obraze celý, v tom případě se na hranici obrazu vytvoří řez objektem. Tohoto faktu by jistě bylo možné využít i při klasifikaci objektů a klasifikovat tak i objekty pouze částečně viditelné. Zároveň se ukázalo, že v některých případech je proces schopný si poradit i s konkávním zakřivením, přestože tento fakt není v algoritmu nijak zahrnut a explicitně se považuje zakřivení povrchu za konvexní. To je dáno pravděpodobně minoritním výskytem konkávních ploch na analyzovaných objektech a podřízením zakřivení plochy minimalizaci energie mřížky. Také je zde nutné zohlednit oddělení jednotlivých objektů od sebe navzájem a od pozadí, protože jejich vzájemná interakce má negativní vliv na průběh analyzované mřížky na hranách objektů, jak je také vidět na výstupním obraze. Na obrázku 4.11 je pro lepší znázornění 3D profilu tentýž výsledek obarvený pomocí hodnoty souřadnice Z elementů.



Obrázek 4.11: 3D profil při zobrazení Z-index.

4.4 Separace objektů a detekce hran

Aby bylo možné jednotlivé objekty (respektive jejich hranice) určit, musíme nejprve nalézt hrany objektů. Hranou budeme při aplikaci na mřížku, se kterou pracujeme v rámci analýzy, rozumět množinu bodů na hranici dvou dostatečně rozdílných oblastí, které tvoří navazující celek. Nabízí se hned několik možností využití zdrojových dat pro detekci hranových bodů v mřížce. První skupina možností se zabývá přímou analýzou mřížky na základě vyhodnocení odlišností sousedících elementů. Rozdílnost je zde možné určovat na základě mnoha metrik. V rámci experimentů byla

vyzkoušena analýza změny intenzity osvětlení, barvy, zohlednění velikosti elementů, a to s použitím různých pravidel pro práh hrany.

Jak se ukázalo, přímá analýza mřížky nepřináší výsledky dostatečné pro analýzu, protože fakt že mřížka vlastní obraz výrazně aproximuje, zde představuje zhoršení vstupu pro vlastní analýzu. Výsledek aproximace mřížky se jeví „rozmazáním“ a změnou polohy hran a proto i vlastní hranice objektu nemohou být dostatečně zřetelné. Z toho důvodu byl v práci použit postup využívající jako vstupu rastrového obrazu hran (stejného jako byl použit při dělení mřížky). Každý hranový bod rastrového obrazu je pak přímo mapován na nejbližší bod mřížky scény a tento je označen jako hranový bod. Důvod mapování je zřejmý, a sice ten, že hranové body a hrany jako takové je třeba analyzovat v rámci stejného geometrického modelu, s jakým pracuje algoritmus 3D analýzy mřížky.

Ve výsledku pak stejné body používáme pro mřížku i její hrany a tak se změna automaticky projevuje mezi oběma způsoby zpracování bez nutnosti explicitních úprav modelů. Experimenty s tímto postupem ukázaly, že hranové body jsou detekovány kvalitně a v závislosti na detailu mřížky je dokonce možné pro analýzu použít podvzorkovaný obraz z nižší úrovně MIPmapy hran a znatelně tak optimalizovat vlastní proces s minimální ztrátou kvality detekce. Vlastní hranové body je pak možné dodatečně filtrovat pro snížení jejich počtu, typicky aplikací pravidla že trojúhelník mřížky smí obsahovat pouze dva hranové body. Tímto postupem je možné redukovat hrany na šířku jednoho bodu (místo alternujících několika bodů kolem společné hrany) a zpřesňovat a optimalizovat tak další analýzu. Na obrázku 4.12 jsou hranové body scény detekované na základě rastrového obrazu hran a tatáž množina po filtraci. Filtrace je aplikována na úrovni trojúhelníků mřížky zavedením pravidla nad trojúhelníkem, při kterém se v případě, že všechny jeho body jsou detekovány jako hranové, vynechá bod, který ze vstupního obrazu hran byl detekován s nejnižší hodnotou pro hranu (bod nejslabší hrany).



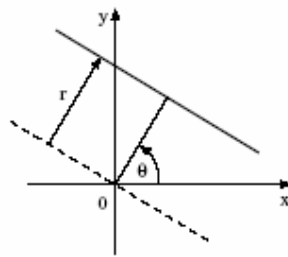
Obrázek 4.12: Detekované hranové body před a po filtraci.

4.5 Analýza geometrických vzorů

Aby bylo možné do analýzy scény zahrnout geometrická pravidla, je nejprve nutné ve scéně konkrétní geometrické vzory najít. Vzhledem k tomu že vzory, které budeme hledat, jsou vesměs křivky a projevují se v obraze jako hrany, je vhodné použít techniku, která dokáže z množiny bodů vypreparovat takovéto pravidelnosti (tj. množiny navazujících bodů tvořících křivky). Jako nejvhodnější se jeví použití metody Houghovy transformace [1] upravené pro potřeby této analýzy. Pro snížení výpočetní náročnosti a také vzhledem k faktu, že křivky potřebujeme detekovat v rámci mřížky (stejně jako v předchozím případě z důvodů souběžné analýzy modelů), budou vstupem transformace nikoliv body rastrového obrazu, ale body mřížky, které byly v předchozím kroku detekovány jako hranové. V této množině bodů budeme detekovat přímky a kružnice, což by mělo postačovat k detekci základních geometrických pravidelností hran.

4.5.1 Detekce přímek

Pro detekci přímek v množině hranových bodů je v projektu použita standardní verze Lineární Houghovy transformace. Metoda spočívá v postupné transformaci bodů hran do parametrického prostoru (r, θ) . Každý bod tohoto prostoru představuje reprezentaci přímky v obraze, parametrizované souřadnicemi $[r, \theta]$ bodu. Parametr r představuje kolmou vzdálenost přímky od počátku souřadného systému obrazu, parametr θ pak úhel, který svírá kolmice k přímce s osou x souřadného systému obrazu (viz obrázek 4.13).



Obrázek 4.13: Reprezentace přímky pomocí parametrů.

$$r(\theta, x, y) = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (1)$$

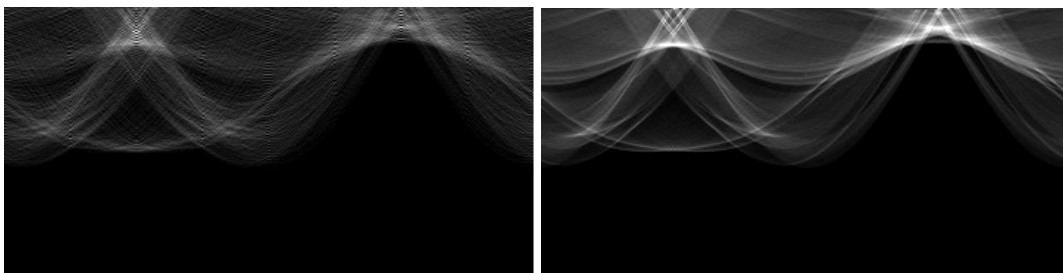
Parametrický prostor transformace (r, θ) zde představuje histogram pro reprezentaci přímek daného obrazu, jehož lokální maxima po ukončení transformace představují parametry rozpoznávaných přímek v obraze. Vzhledem k tomu, že v programovém prostředí se musíme spokojit s reprezentací diskrétní, volba rozlišení parametrického prostoru určuje přesnost analýzy. Každý bod obrazu, reprezentující

hranu, se transformuje do parametrického prostoru v podobě sinusové křivky složené z bodů $[r, \theta]$, která reprezentuje množinu všech přímk, procházejících bodem obrazu (x, y) .

Vzdálenost přímky od počátku soustavy souřadnic obrazu je dána rovnicí (1), která se postupně vyhodnotí pro úhel θ diskrétně vzorkovaný v rozsahu $\langle 0; \pi \rangle$ nebo $\langle 0; 2\pi \rangle$ v závislosti na konkrétní reprezentaci rozsahu vzdálenosti r . Pro první případ musíme použít rozsah vzdáleností v rámci kladných i záporných hodnot, v případě druhém pak postačují hodnoty kladné na úkor většího rozsahu zkoumaného úhlu.

Druhá možnost se pak používá hlavně v případě, že za počátek soustavy souřadnic při transformaci považujeme střed obrazu a nikoliv jeho roh. V obou případech pak není nutné vyhodnocovat zdánlivě chybějící polovinu rozsahu parametrického prostoru (ať již vzdálenost r nebo úhel θ), protože je symetrická s polovinou již vyhodnocenou. V rámci projektu je použit plný rozsah úhlu přímky a pouze kladná vzdálenost přímky od počátku, který je zároveň středem analyzovaného obrazu (mřížky).

Vzhledem k tomu, že parametrický prostor je histogramem, takováto křivka se do obrazu zaneše navýšením bodů na souřadnicích všech bodů křivky o 1. V praxi to znamená, že konečná hodnota v histogramu na souřadnicích $[r, \theta]$ představuje počet hranových bodů na přímce v obraze dané parametry (r, θ) . Z toho důvodu se jako výsledně rozpoznané přímky chápou lokální maxima histogramu, protože hodnoty nižší v jejich okolí jsou jakési „stíny“ přímek způsobené vlastní diskretizací obrazu i metody. Parametrický prostor základní testované scény je vidět na obrázku 4.14, pro porovnání je zde histogram vytvořený přímo z obrazu hran i ten vytvořený pouze z hranových bodů mřížky, který se ve skutečnosti analyzuje. Jak je vidět, histogram zůstává zachován a metoda s použitím pouze omezené množiny bodů neztrácí na kvalitě výsledků. Rozdílnost je způsobena pouze počtem účastnících se bodů a na hledání maxima v histogramu nemá zásadní vliv.

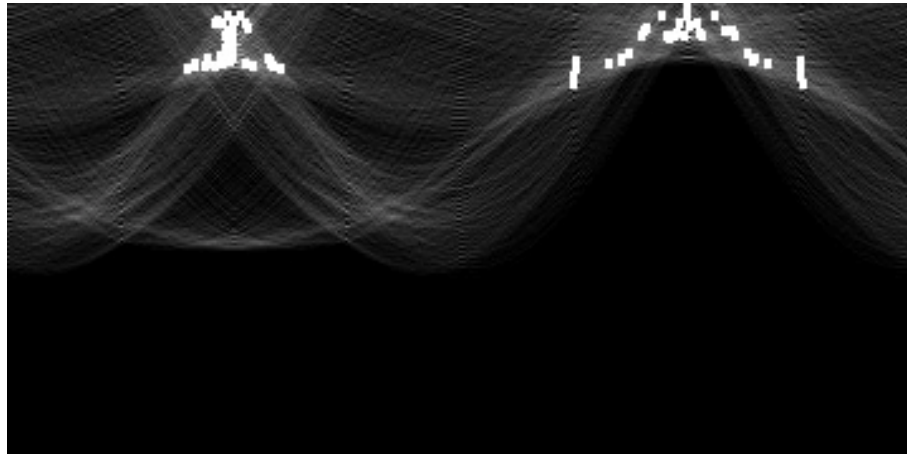


Obrázek 4.14: Lineární Houghova transformace mřížky (vlevo) a obrazu hran (vpravo).

Po nalezení lokálních maxim v histogramu (parametrů obrazových přímek) je třeba pro tato maxima najít odpovídající hranové body mřížky. Kvůli paměťové náročnosti nebylo možné již při transformaci uchovávat vazby na body, které se na daném maximu podílely, proto je jejich detekce nutná dodatečně. Body ležící na přímce jsou v tomto případě všechny body, které leží v rámci zvolené vzdálenosti od přímky určené maximem. Operace určení vzdálenosti bodu od přímky by v

prostoru kartézských souřadnic vyžadovala zpětné vyhodnocení rovnice přímky na základě polohy maxima a poté standardním způsobem výpočet vzdálenosti bodu od přímky.

Díky charakteru vlastní transformace není tato relativně složitá operace nutná, protože lze využít přímo vlastností parametrizace přímky k určení vzdálenosti bodu. Proložíme-li bodem přímku rovnoběžnou s přímku určenou maximem, bude tato mít stejnou směrnici v prostoru Houghovy transformace (parametr θ), a bude se lišit pouze vzdáleností od počátku souřadnic. Je zřejmé, že rozdíl vzdáleností obou přímek od počátku systému souřadnic (rozdíl parametrů r) odpovídá vzdálenosti přímek a tedy i vzdálenosti testovaného bodu od přímky. Pro výpočet, zda bod na přímce leží, tedy stačí na tento bod běžným způsobem aplikovat rovnici transformace (1) pro úhel přímky analyzovaného maxima a rozdíl parametrů r porovnat s požadovanou maximální vzdáleností. Na obrázku 4.15 jsou nalezená maxima v histogramu mřížky.



Obrázek 4.15: Maxima histogramu transformace.

Po asociaci bodů k jednotlivým přímekám je třeba body seřadit kvůli další analýze. Vlastní směr řazení není podstatný, důležitý je pouze fakt, aby body byly seřazeny v rámci sekvence mezi prvním a posledním bodem přímky. Po řazení je možné na množinu bodů aplikovat pravidlo minimálního počtu bodů na přímce a maximální vzdálenosti bodů a rozdělit tak přímku na úsečky, které v obraze ve skutečnosti tvoří hrany. Výsledkem je množina hranových úseček, které je možné dále analyzovat a případně použít pro vyhodnocení geometrie scény. Protože analýza přímek není primárním cílem projektu, nebudeme se jí dále zabývat.

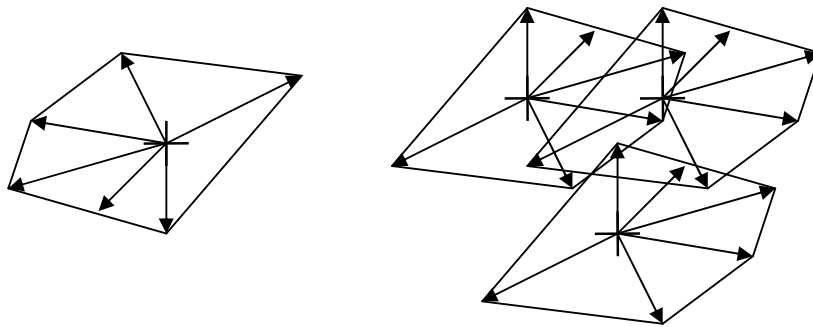
4.5.2 Detekce kružnic

Pro detekci kružnic je v projektu použita opět Houghova transformace, ovšem nikoliv kruhová založená na směru gradientu obrazu, ale modifikovaná obecná Houghova transformace [9] využitá pro detekci kružnic nebo jejich částí. Důvody použití obecné transformace jsou dva: použití gradientu pro směr středu kružnice není obecně spolehlivé a silně závisí na kvalitě scény, a jako zdroj bodů je

použita množina hranových bodů mřížky, kde taková informace ani není k dispozici a i v případě že by byla, by byla silně zkreslená aproximací mřížky.

Obecná Houghova transformace, stejně jako předchozí její varianty vytváří histogram pravděpodobnosti výskytu prvku v obraze. Narozdíl od lineární transformace tato pracuje přímo v geometrickém prostoru obrazu, přičemž subjektem hledání je konkrétní křivka. Podobně jako transformace lineární, i tato uvažuje v případě výskytu hranového bodu všechny výskyty hledané křivky, které by se mohly na tomto bodu podílet. Křivka je zde reprezentována středovým bodem (typicky těžištěm) a množinou vektorů vlastní křivku charakterizujících. Pro bod hrany pak do histogramu zapisujeme všechny potenciální výskyty středů křivky pro daný bod.

Je zřejmé, že v tomto případě množina potenciálních středů křivky kolem daného bodu odpovídá svisle i vodorovně převrácenému obrazu hledané křivky a proto je možné křivku do bodů v histogramu rasterizovat běžným způsobem, pouze příslušně převrácenou, viz obrázek 4.16. Množina bodů ležících na křivce pak v histogramu tvoří lokální maximum v místě zvoleného středu křivky.

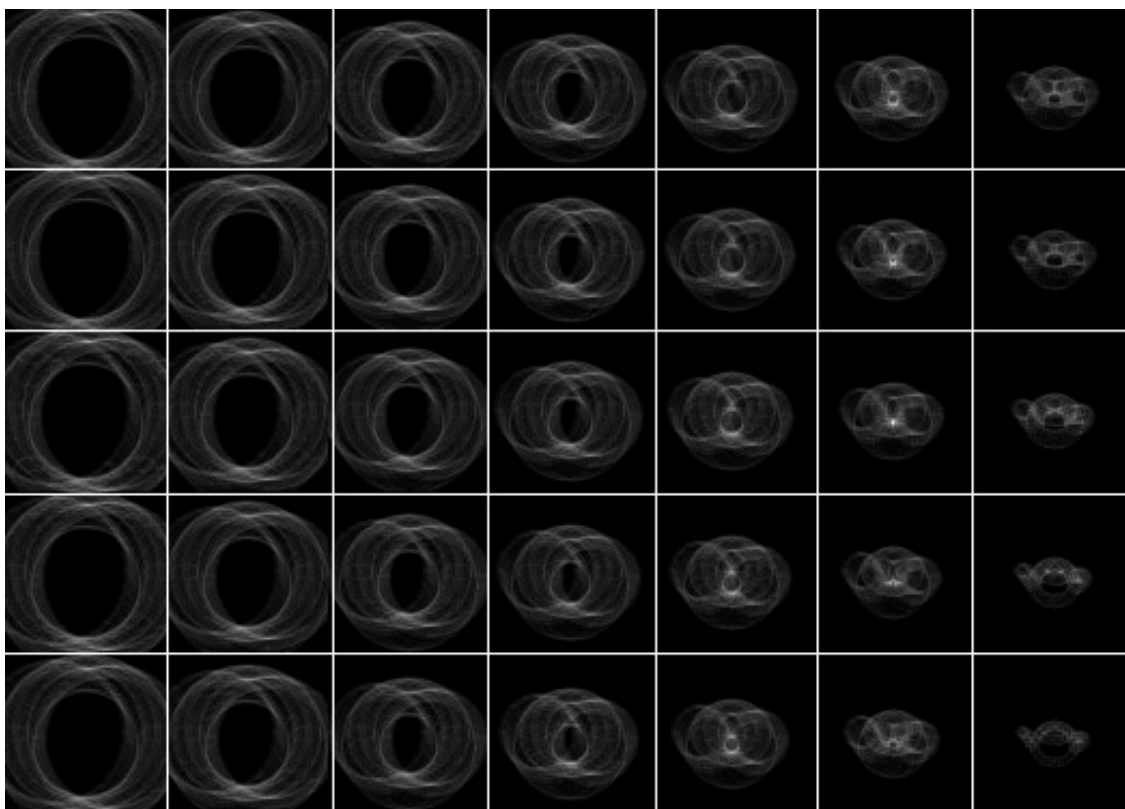


Obrázek 4.16: Reprezentace křivky v obecné Houghově transformaci.

V rámci tohoto projektu budeme v obraze hledat tímto způsobem pouze kružnice, proto můžeme zanedbat princip převrácení křivky (převrácená kružnice je stále stejná kružnice). Z popisu algoritmu a faktu že pracuje v rastrovém obraze s použitím konkrétního obrazu křivky je vidět, že základní varianta pohybující se v prostoru 2D histogramu obrazu není invariantní vůči změně velikosti křivky. Proto musíme pro transformaci použít trojrozměrný prostor, kde třetí souřadnice bude představovat poloměr hledané kružnice.

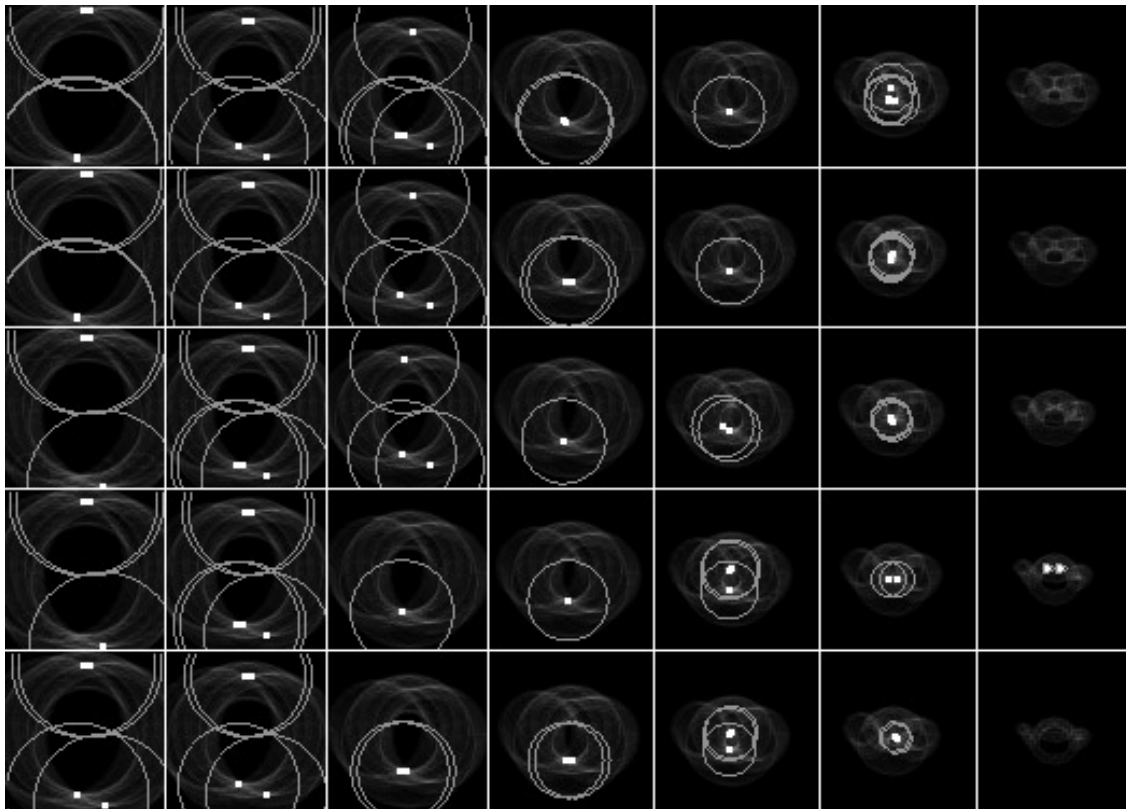
Jako celek si lze tento proces představit jako několik na sobě nezávislých transformací hledajících kružnice s různými poloměry tvořící jednotlivé “vrstvy” celku. Integrací do spojitého 3D prostoru ovšem získáváme možnost analýzy maxim se zohledněním poloměru a detekce kružnice s optimálním poloměrem pro daný středový bod. Jestliže má poloměr kružnice lineární závislost na třetí souřadnici prostoru histogramu, projeví se původní dvourozměrná kružnice v 3D histogramu jako množina kružnic se zmenšujícím se poloměrem tvořící kužel (komolý kužel v případě omezení minimálního poloměru). Optimální bod (maximum histogramu) vzniká “zaostřením” průsečíků ploch

kuželů v jediném bodu. Na obrázku 4.17 je vidět transformace scény, třetí souřadnice (poloměr) je vykreslena v podobě vrstev obrazu. Jak je vidět, pro získání použitelného histogramu stačí i velmi malý prostor parametrů. Vlastní scéna zabírá v tomto prostoru pouze přibližně třetinu rozměru, protože je nutné počítat s faktem, že střed křivky (kružnice) je mimo oblast scény a tak je nutné prostor transformace o tuto potenciální oblast rozšířit.



Obrázek 4.17: Obecná Houghova transformace pro kružnici s rozšířením do 3D prostoru.

Maximum histogramu přímo určuje geometrické parametry kružnice, proto lze snadno najít množinu hranových bodů, které se na maximum podílely. Jedná se o body ležící v dané vzdálenosti (třetí souřadnice maxima) od středu maxima (první dvě souřadnice maxima) opět se zohledněním nějaké tolerance kvůli diskretizaci analýzy a charakteru vstupních bodů. Stejně jako v předchozím případě i zde je nutné nalezené body seřadit a provést rozdělení kružnice na jednotlivé úseky odpovídající shlukům bodů v obraze se skutečně vyskytujících. Výsledek analýzy maxima je na obrázku 4.18. Protože prostor této transformace odpovídá prostoru obrazu, jsou zde zároveň znázorněny pro lepší přehlednost i detekované kružnice, respektive jejich části. Protože prostor transformace je trojrozměrný, je možné maxima hledat s trojrozměrným jádrem a detekovat tak pouze ta maxima, která mají nejlepší ostrost v rámci prostoru transformace.



Obrázek 4.18: Maxima analýzy kružnic v 3D prostoru transformace.

Po nalezení křivek je možné tyto dále analyzovat, případně spojovat a aplikovat na ně geometrická pravidla v rámci analýzy 3D modelu scény. Stejně jako u přímků i zde je nutné provést asociaci hranových bodů mřížky k jednotlivým kružnicím, seřadit body a detekovat v jejich posloupnosti kruhové oblouky skutečně se ve scéně vyskytující. Opět jde více o návrh pro rozšíření systému než primární cíl analýzy, a proto se jím nebudeme dále zabývat, protože pro základní analýzu a algoritmus v této práci vytvořený nejsou výsledky této rozšířené analýzy nezbytné.

4.6 Analýza textury

V rámci této práce byl jako rozšíření vytvořen jako modul i analyzátor textury. Analyzátor pracuje samostatně, nicméně pracuje pouze s částečně rozšířenými statistikami obrazu, jaké se používají při aproximaci obrazu mřížkou, proto by z hlediska systému jako celku nemělo činit potíže tento analyzátor později do systému integrovat přímo do procesu analýzy. Pro tuto možnost byl modul také navrhován. Pro klasifikaci je použit algoritmus **ART** (Adaptive Resonance Theory [10]) z důvodů předem neznámého počtu tříd ke klasifikaci, nutnosti průběžného učení a také vhodné kompatibility se vstupními daty.

Z analyzované (učené) textury je vytvořen normalizovaný vektor příznaků obsahující n hodnot v intervalu $\langle 0;1 \rangle$, a tento je přikládán na vstupní vrstvu neuronů. Výstupní vrstvu tvoří m neuronů odpovídajících známým (naučeným) třídám. Výstupní vrstva pracuje s radiální bázovou funkcí [11] (**RBF**) a porovnává vstupní vektor na základě vah mezi výstupním neuronem a vstupní hodnotou. Sumarizovaná hodnota je opět normalizována do intervalu $\langle 0;1 \rangle$, kdy hodnota odpovídá 0-100% požadované shody. Je-li shoda větší než požadovaný parametr ostražitosti v (vigilance), pak v případě učení, jestliže název třídy odpovídá odpovědi výstupního neuronu nebo třída byla neznámá (vzniklá z předchozí neurčité klasifikace), upravují se jeho váhy. Pokud žádný neuron při klasifikaci neuspěl, vytvoří se neuron nový. V případě klasifikace se upravují váhy, pokud je ostražitost splněna, pokud ne, vytvoří se nový neuron neznámé třídy pro další zpracování.

Vstupní vektor i velikosti n obsahuje normalizovanou RGB hodnotu průměrné barvy (odstín barvy textury), normalizovanou RGB hodnotu rozptylu hodnot pixelů (složitost textury), histogram normalizovaný na 90% prvků kvůli ošetření odlehlých hodnot, roztažený na celou šířku kvůli eliminaci světelných podmínek a komprimovaný na 64 tříd na barevnou složku kvůli částečnému vyhlazení, protože původních 256 hodnot je příliš podrobné dělení pro analýzu. Dále vektor obsahuje normalizovaný histogram vzoru pixelů. Pro každou kombinaci okolních pixelů jedna třída v histogramu, reprezentace okolí se chápe jako 1 byte složený z bitů, jejichž hodnota je 1 pokud sousední pixel je větší nebo roven aktuálnímu, 0 pokud je menší než aktuální. Celkově tak při ztotožnění rotovaných vzorů vzniká 36 kombinací okolní pixelu. Příklad vyhodnocení vzoru pixelů je na obrázku.

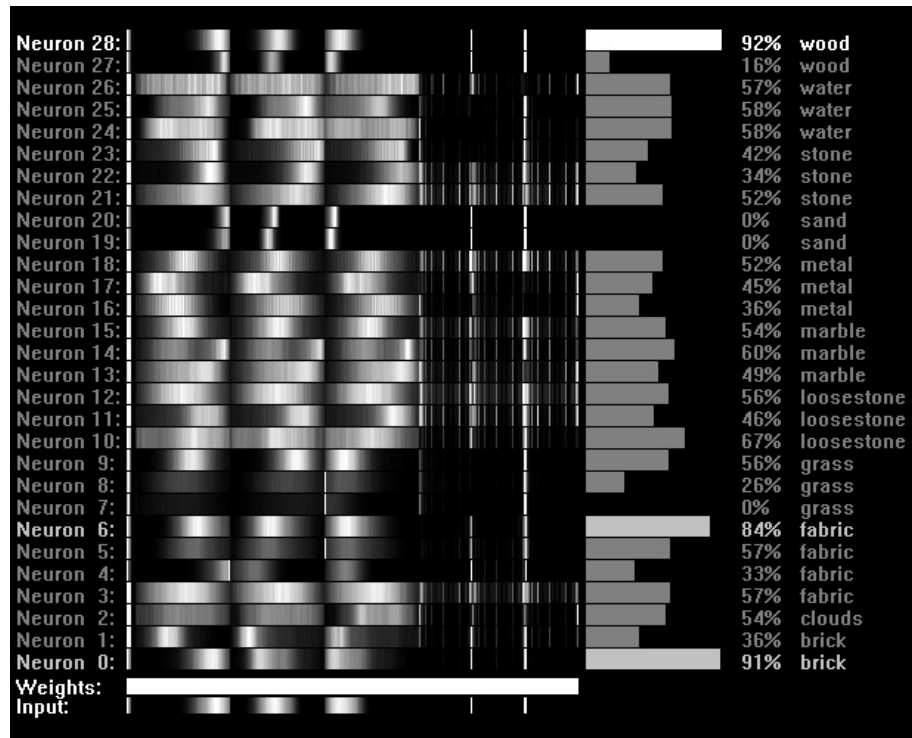
0	1	2	
7	X	3	= 00110110 = 54
6	5	4	

Obrázek 4.19: Příklad vzoru pixelů

Vektor vstupních vah v o velikosti m umožňuje váhově upravit důležitost jednotlivých složek vektoru při klasifikaci, v projektu jsou všechny jeho hodnoty rovny 1, nicméně v budoucnu se předpokládají experimenty s jiným rozložením. Dále systém obsahuje matici vah mezi neurony w o velikosti $m \times n$ obsahující hodnoty asociované s vektorem příslušné třídy textury. Funkce výstupního neuronu a vzorec pro úpravu vah mezi neurony jsou charakterizovány rovnicemi (5) a (6). Koeficient učení I v rozsahu $\langle 0;1 \rangle$ určuje rychlost adaptace neuronu na nový podnět při klasifikaci textury. Hodnota 0 odpovídá žádnému učení, hodnota 1 okamžitě adaptaci na nový vektor.

$$o_j = \max \left(0 ; 1 - \frac{\sum_{i=1}^n v_i (w_{i,j} - i_i)^2}{\sum_{i=1}^n v_i \cdot w_{i,j}} \right) \quad (5) \quad w_{i,j}^{n+1} = w_{i,j}^n - (w_{i,j}^n - i_i^n) \cdot l \cdot o_j^n \quad (6)$$

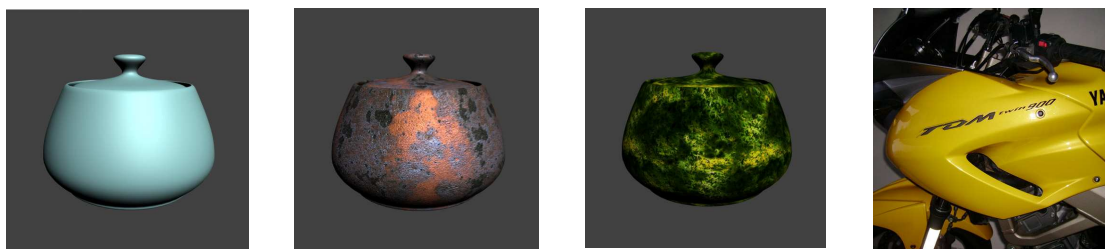
Výstup modulu analyzátoru textury graficky znázorňuje stav sítě a odezvy na vstupní charakteristický vektor textury. Podobně jako zbytek systému i jeho chování, především parametry, lze ovlivnit skriptem zpracování. Příklad výstupu je na obrázku 4.20. Nejspodnější pás odpovídá vektoru příznaků aktuálního vstupního vektoru. Jeho obsahem je: Průměrná hodnota, rozptyl, RGB histogram jasu, RGB histogram vzorů. Hned nad ním je pás zobrazující váhy vstupního vektoru, zde v projektu váhy všude rovny 1, proto jen bílý pás. Následují pásy odpovídající stavu výstupních neuronů, vždy nejprve vektor vah k danému výstupnímu neuronu ze vstupu, napravo pak velikost odezvy. Zvýrazněn je vítězný neuron a neurony, které vyhověly parametru ostrážitosti. Při experimentech se ukázalo, že klasifikátor dokáže poměrně kvalitně reagovat i na úplně neznámou texturu, která má pouze podobné charakteristiky, proto bude vhodné s ním v dalším rozvoji systému počítat, především při zpracování reálných scén.



Obrázek 4.20: Výstup analyzátoru textury

5 Experimenty

Experimenty se systémem byly prováděny s použitím několika základních jednoduchých scén. Protože algoritmy jsou vytvářeny pro analýzu scény jako celku, byly pro experimenty zvoleny scény s jediným objektem nebo nepřekrývajícími-se objekty. Pro analýzu scény s objekty složitějšími bylo nutné aplikovat navrhované rozšíření segmentace, omezení oblasti analýzy, analyzátor textury a další. Cílem této práce je nicméně vyhodnocení vhodnosti navrženého algoritmu a jeho testování se také testované scény musí podřídit. Přesto se ukázalo, že systém si při experimentech počíná dobře i v podmínkách, které do algoritmu nejsou explicitně zahrnuty, a dokonce se úspěšně vyrovnává s některými nepříznivými vlivy vstupní scény. Cílem experimentů pak nebude najít nejvhodnější scénu z dané kategorie, ale naopak vyzkoušet scénu běžnou a zhodnotit klady i zápory výsledků jejího zpracování. Pro analýzu použijeme 4 základní typy scény, jak byly nastíněny již na začátku práce při návrhu algoritmu. Přehled testovaných scén pro vzájemné porovnání je na obrázku 5.1.



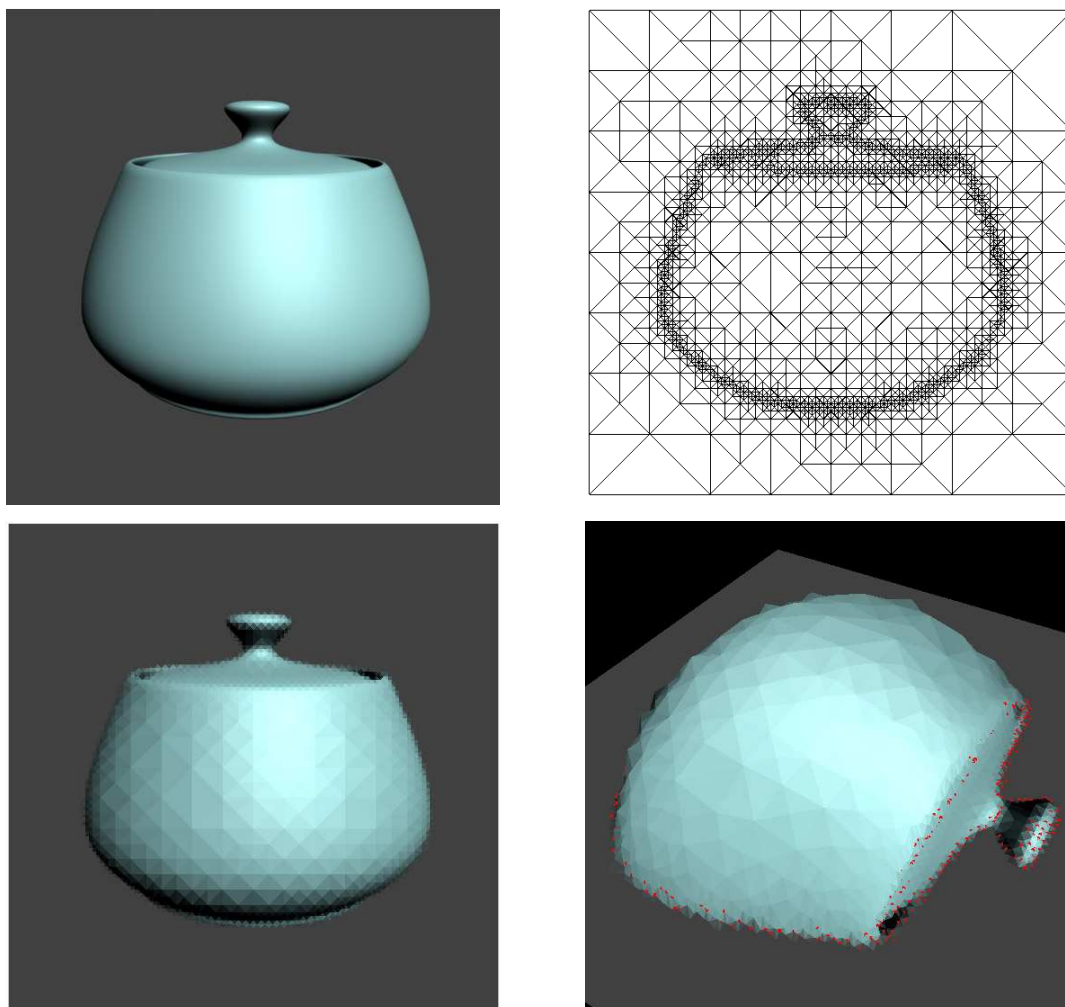
Obrázek 5.1: Testované scény.

Scény syntetické používají stejný objekt, pouze s různou texturou, aby bylo možné objektivně zhodnotit vliv textury na kvalitu analýzy. Poslední (reálná) scéna pochází z běžné fotografie a bude se zaměřovat na analýzu souvislé plochy v rámci složitějšího okolí. Originální obraz nebyl nijak upravován a ani osvětlení není nijak uměle namodelované.

Experimenty byly prováděny na vývojové verzi systému, tedy bez optimalizací v rámci kódu. Analýza scén byla prováděna na počítači: *Notebook ASUS M6BNe, Intel Pentium M 1.6 GHz, 1.5 GB RAM DDR333, pod Windows XP.*

5.1 Syntetická scéna

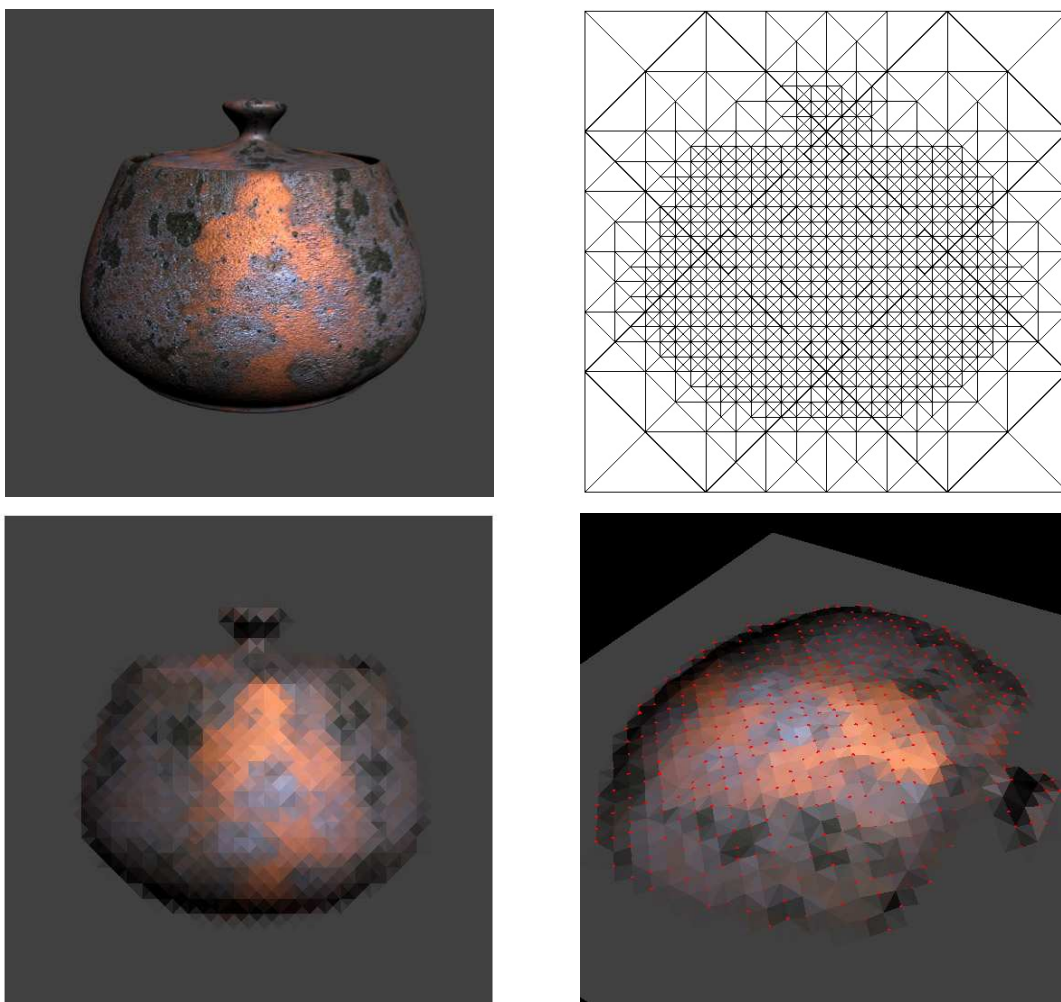
Syntetická scéna s jednoduchými objekty je vstup, s jakým by se měl algoritmus vypořádat bez větších potíží. Charakter algoritmu již z principu počítá s konvexními objekty, nicméně navzdory tomu byla zvolena scéna složitější, aby se ukázalo, jak se v takovém případě algoritmus zachová vzhledem k neočekávaným zalomením prostoru. Výsledek analýzy je na obrázku 5.1. Jak je vidět, algoritmus pracuje správně, a dokonce je si schopen částečně poradit s nekonvexním zakřivením povrchu, při dalším rozšíření by jej zřejmě bylo možné poměrně úspěšně aplikovat i na složitější tvary. Základní cíl tohoto projektu byl tedy úspěšně splněn, protože proces je schopný poměrně kvalitní analýzy jednoduché syntetické scény. Výstupní model scény je dostatečně kvalitní na to, aby mohl být použit zároveň s detektorem a klasifikátorem objektů a díky tomu částečně modelovat i skryté části scény jako je odvrácená strana objektu.



Obrázek 5.2: Analýza syntetické scény.

5.2 Texturovaná scéna

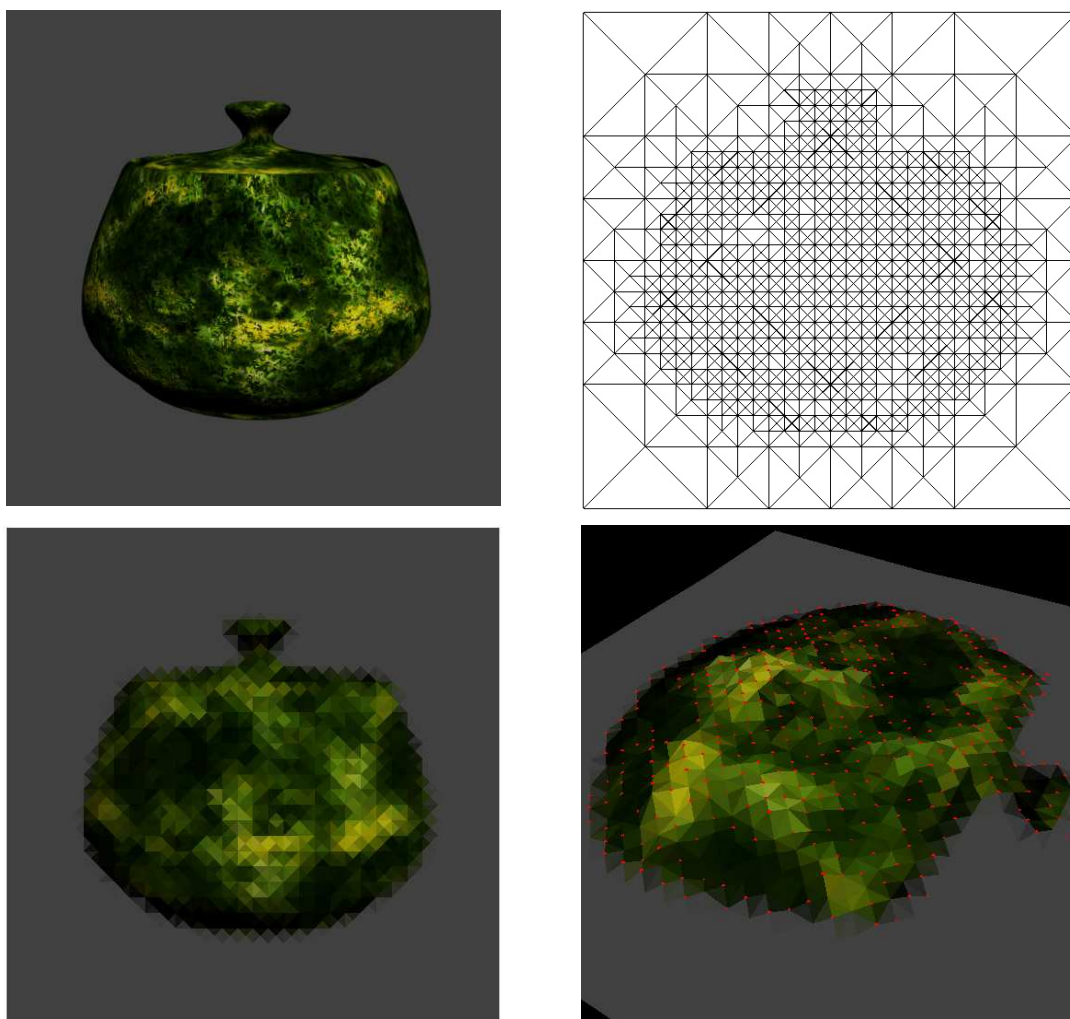
Analýza texturované scény bude pro systém složitější, protože v současném stavu se systém při analýze texturou nezabývá. Díky aproximaci by nicméně měl být algoritmus schopný podobně kvalitní analýzy při snížení úrovně detailu mřížky. Na obrázku 5.3 jsou vidět výsledky analýzy pro stejnou scénu kde objekt je texturovaný. Abychom předešli přílišné podobnosti se scénou syntetickou a ověřili tak robustnost algoritmu, je použita poměrně nepravidelná textura. Jediný požadavek na texturu byl, aby zachovávala průběh intenzity v závislosti na osvětlení objektu. Výsledky jsou překvapivě dobré, nicméně je třeba počítat s tím, že ne každá textura je s procesem analýzy dostatečně kompatibilní a mnohdy může dojít ke zkreslením daným charakterem textury. Důležitým faktem je, že scéna musela být zpracována se sníženým detailem aproximace, aby nedošlo ke zkreslením způsobeným strukturou textury. Budoucí použití analyzátoru textury jistě přinese výrazné zlepšení v tomto směru.



Obrázek 5.3: Analýza texturované scény.

5.3 Složitě texturovaná scéna

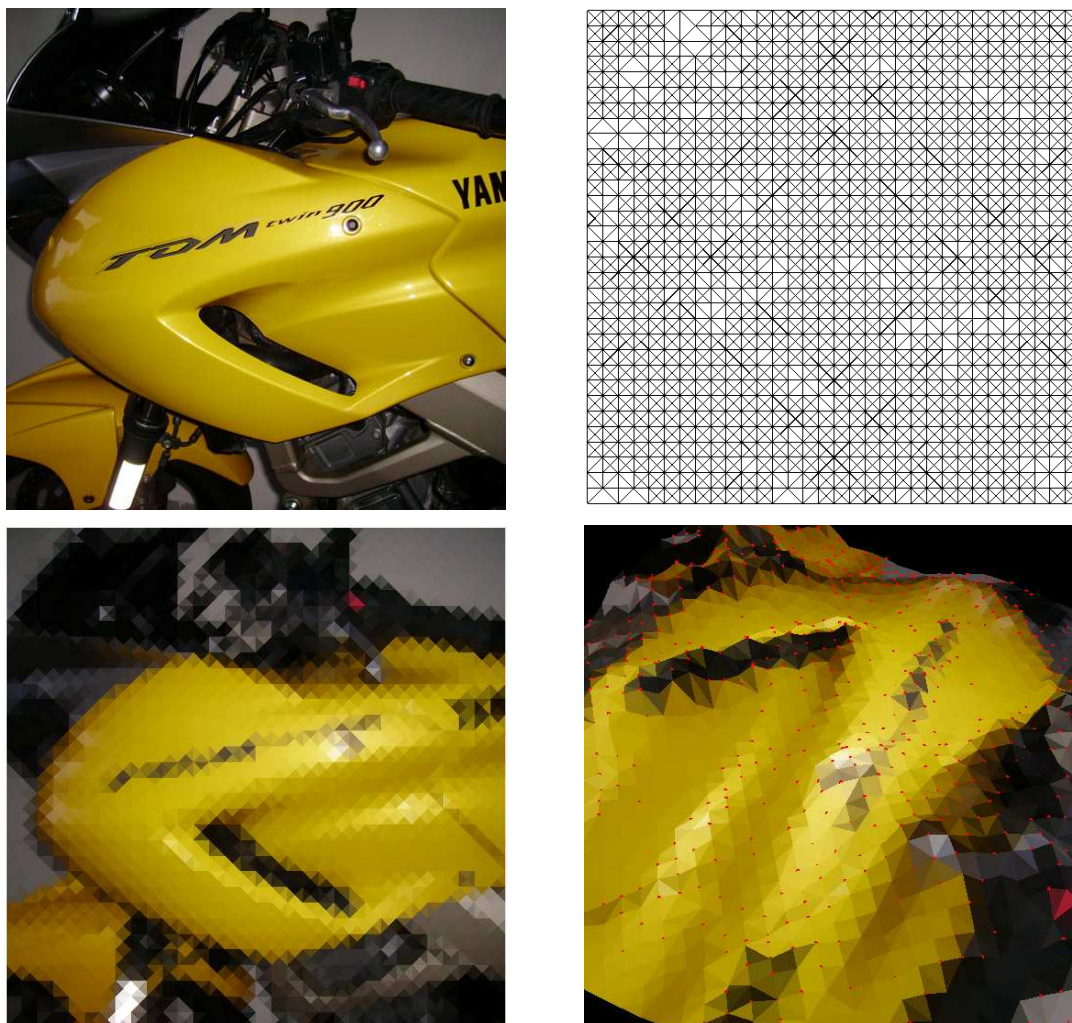
Další úroveň testování systému analýzy je scéna se složitou texturou. Zajímá nás, jak si algoritmus poradí s texturou, která sama o sobě přináší do scény množství hran a nepravidelností v osvětlení objektu. Zvolíme texturu, která je z hlediska analýzy složitá i pro lidské oko, a budeme pozorovat vlastnosti algoritmu při jejím zpracování. Na základě výsledků se pak pokusíme učinit závěr možnosti využití procesu analýzy v takové scéně. Na obrázku 5.4 je výsledek analýzy složitě texturované scény. Jak je vidět, samotná struktura textury narušuje algoritmus, který neanalyzuje objekt jako celek, ale konverguje k analýze průběhu textury jako takové. Zde by pravděpodobně stálo za úvahu aplikovat tento proces na detailnější analýzu částí objektu ve složitějším systému.



Obrázek 5.4: Analýza složitě texturované scény.

5.4 Reálná scéna

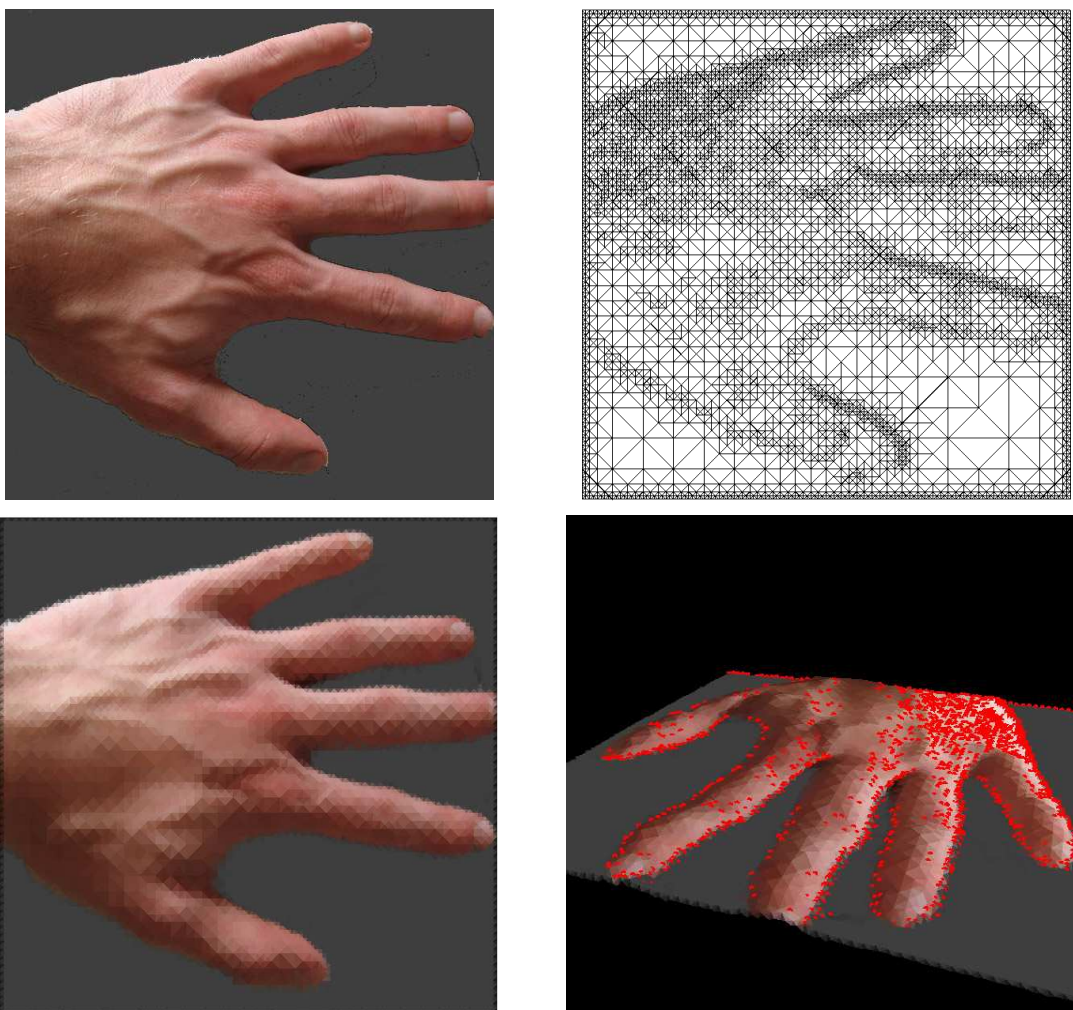
Lze předpokládat, že reálné texturované scény nebudou mít výsledky analýzy lepší než ty syntetické, proto se při testování reálné scény zaměříme na takovou, kde povrch objektů je homogenní a osvětlení scény je vhodné. Pro analýzu složitější scény by bylo nejprve potřeba integrovat analyzátor textury a další pokročilou analýzu a zlepšit výsledky analýzy syntetických texturovaných scén. Na obrázku 5.5 je výsledek analýzy reálné scény. Jak je vidět na mřížce, tento typ scény je postižen stejně jako složitě texturovaná scéna množstvím hran, které vytváří příliš podrobný detail a tak by se dalo předpokládat, že analýza bude nekvalitní. Jak se ovšem ukázalo, výsledek předčil očekávání, protože díky homogenní barvě spojitého povrchu kolaps mřížky proběhl korektně. Analýza má daleko k dokonalosti, nicméně profil analyzovaného povrchu má tendenci zaujímat odpovídající tvar, a dá se předpokládat, že při rozšíření systému by i taková scéna mohla být vhodným vstupem pro algoritmus.



Obrázek 5.5: Analýza reálné scény.

5.5 Umělá reálná scéna

Vzhledem k tomu, že výsledky z obecné reálné scény byly poměrně uspokojivé, rozhodl jsem se podrobit experimentu ještě jeden druh scény, umělou reálnou scénou. S tímto typem scény se můžeme běžně setkat při zpracování obrazu v průmyslových aplikacích. Scéna je charakteristická tím, že objekty jsou jednoznačně oddělené od homogenního pozadí podobně jako v syntetické scéně a směr osvětlení je přizpůsoben systému analýzy. Při této analýze se tedy zpracovává běžný snímek objektu v částečně umělém prostředí. Na obrázku 5.6 jsou vidět výsledky analýzy profilu ruky, scény, která se používá například v biometrických systémech [22]. Navzdory tomu, že systém pro tento typ scény nebyl navrhován, se výsledky jeví jako kvalitní a při dalších úpravách by systém mohl být použit pro podobné účely v různých aplikacích.



Obrázek 5.6: Analýza umělé reálné scény

5.6 Shrnutí experimentů a návrhy do budoucna

Navzdory počátečním problémům s implementací algoritmů a vhodné aproximaci scény se ukázalo, že navržený algoritmus má při splnění jistých podmínek dostatečně kvalitní výstup na to, aby mohl být použit v rámci nějakého složitějšího systému. Obrovskou výhodou procesu analýzy vytvořeného v rámci této práce je možnost rozšiřování procesu analýzy, procesu předzpracování i dodatečného zpracování. Díky tomu nabízí kvalitní základ pro komplexní systém analýzy na rozdíl od většiny stávajících algoritmů, které jsou často dotaženy k maximu svých možností a nenabízejí velkou kompatibilitu s jinými postupy.

Provedeme-li přímé porovnání systému se stávajícími algoritmy, snadno zjistíme, že nabízí mnohá vylepšení, ale bohužel i nedostatky. Existující algoritmy využívající analýzu osvětlení často nabízejí přesný a kvalitní výstup, který je ovšem podmíněný prakticky laboratorními podmínkami a přesným nasvícením scény, jehož rovnic pak využívají. Systém vytvořený v rámci této práce se naopak zdá robustní vzhledem k neznámým světelným podmínkám za cenu menší přesnosti analýzy, nicméně nabízí široké možnosti rozšíření analýzy, které by tento nedostatek mohly dostatečně kompenzovat. Srovnání z hlediska rychlosti analýzy s jinými algoritmy není příliš vhodné, protože získané výsledky by musely mít stejný charakter, nicméně důležitým faktem je to, že mnoho z takových systémů je velmi náročných především kvůli použití složitých matematických konstrukcí. Pokud se omezíme pouze na základní analýzu jednoduché scény, zabere tato při zpracování vstupního obrazu s rozlišením 512x512 pixelů přibližně jednu vteřinu, podrobnější zpracování nebo složitější scéna pak zabere více času, ovšem stále se pohybujeme v řádu jednotek. Čas zpracování nebyl měřen přesně, protože část algoritmu pracuje s náhodností a také doba zpracování závisí na složitosti scény. Díky charakteru zpracování je možné jednotlivé kroky oddělit a vytvořit tak řetězec zpracování, který by v reálném prostředí mohl mít dostatečnou propustnost pro zpracování jednoduchých scén v reálném čase. Zároveň se dá předpokládat paralelizace některých operací analýzy.

Rozšíření do budoucna, která by bylo vhodné implementovat, byla zmíněna již v průběhu popisu jednotlivých modulů, zde je tedy pouze shrneme:

- **Analýza geometrických vzorů** – Stávající detekci přímých a zaoblených hran rozšířit pro pokročilou analýzu modelu scény se zahrnutím geometrických vzorů do procesu kolapsu mřížky.
- **Analýza textury** – Je jistě možné a vhodné integrovat stávající analyzátor textury do procesu zpracování a tím zajistit robustnější analýzu texturovaných scén.

- **Detekce a klasifikace objektů** – Analyzovaný profil povrchu objektů scény je možné korelovat s databází objektů (základních geometrických primitiv) a upřesnit tak analýzu mřížky, popřípadě rozšířit model scény o skryté oblasti.
- **Rastrové předzpracování** – V závislosti na dodatečné informaci o charakteru scény by bylo vhodné rastrově předzpracovat vstupní obraz a eliminovat tak některé nedostatky reálných scén jako je šum a podobně.
- **Zpětná vazba** – V případě zpracování sekvence obrazů by jistě bylo vhodné implementovat zpětnou vazbu na prvky analýzy vycházející z výsledků zpracování předchozího snímku. Tím by se mohla analýza zpřesnit a zrychlit.
- **Aplikace dalších postupů** – Pro počáteční podmínky analýzy, ale i upřesnění výstupů by bylo vhodné použít jiné metody analýzy scény. Především by bylo vhodné využít výstupu stereoskopické analýzy pro počáteční nastavení kolapsu mřížky, pokud by byl k dispozici vstup ze dvou paralelních kamer.
- **Optimalizace výkonu systému** – Paralelizací a řetězením operací dosáhnout lepší propustnosti systému jako celku při použití více procesorů. Zároveň by bylo vhodné zvážit možnost implementace nejnáročnějších částí (především kolapsu mřížky) v rámci výpočetních možností vertex shaderů grafické karty nebo pomocné karty pro fyzikální výpočty.

Možností dalšího postupu při rozšiřování systému je mnoho, zde byly nastíněny jen ty nejdůležitější, s jakými systém jako celek do budoucna počítá a které jsou z hlediska jeho struktury dostatečně kompatibilní z hlediska vstupů i výstupů. Podstatným předpokladem zde je to, aby takové rozšíření zachovávalo nebo zvyšovalo kvalitu stávajícího zpracování a výrazně nezvyšovalo výpočetní náročnost systému jako celku.

6 Závěr

V práci byly probrány nejpoužívanější postupy při analýze obrazu, vyzvednuty jejich přednosti a zmíněny nedostatky. Důvodem tohoto průzkumu a shrnutí bylo především zlepšení orientace v problematice počítačového vidění, a získání přehledu o jednotlivých metodách. Tento přehled je důležitý při samotném návrhu a analýze postupů zpracování scény. Obecně se nedá říct, že by některá metoda byla vyloženě nevhodná pro analýzu, protože pojem analýza ve své komplexnosti zahrnuje mnoho procesů a postupů, a dává tak široké možnosti použití libovolného typu zpracování. Zároveň je počítačové i lidské vidění oborem tak neprozkoumaným, že by byla chyba předem ztracovat nějakou metodu jen proto, že ve stávajících modifikacích nedává dostatečné výsledky. Jak už jsme se mohli mnohokrát přesvědčit, jiný pohled na problematiku, a aplikace modifikací dokáže z takto „neúspěšných“ metod vytvořit robustní bloky, a právě proto je třeba obor počítačového vidění neustále rozvíjet ve všech směrech. Podobně jako v předchozí bakalářské práci byly zkušenosti se studiem postupů počítačového vidění a experimenty s vlastní implementací velmi užitečné. Opět se potvrdilo, že jednostranný přístup k problému není vhodný a mnohem lepší výsledky přináší kombinace několika přístupů ze skupiny rastrových i vektorových metod.

Navržený a implementovaný algoritmus se jeví jako úspěšná alternativa ke stávajícím postupům analýzy a nabízí poměrně široké možnosti pro další vylepšování systému. Jeho výsledky jsou kvalitní, místy dokonce až překvapivě. Kromě vlastního systému pro analýzu scény bylo vytvořeno i několik rozšíření kvůli prozkoumání dalších možností systému. V rámci práce byly použity algoritmy převzaté, upravené, i zcela vlastní. Modifikace se týkají především algoritmu ROAM (Realtime Optimally Adapting Meshes), který je přizpůsoben procesu analýzy, dále procesu obecné Houghovy transformace použité v rozšíření pro detekci kružnic, a v neposlední řadě i algoritmu ART (Adaptive Resonance Theory) v modulu analýzy textury. Použití jednoduché matematiky při analýze 3D profilu objektů se podle předpokladů ukázalo jako efektivní a dostatečné.

Práce jako taková splnila požadavky zadání bez výhrad, nicméně při její tvorbě jsem narazil na mnoho neprozkoumaných oblastí analýzy obrazu, které si zaslouží další pozornost. Vzhledem k tomu, že vytvořený systém se jeví jako efektivní základ komplexního systému počítačového vidění, bude dále rozvíjen i nad rámec této práce, a dost možná se v budoucnu stane základem pro systém vidění autonomního robota. Pro takové užití nicméně bude nutné doimplementovat mnoho z navrhovaných rozšíření a zaměřit se více na charakteristické vlastnosti reálné scény při zpracování.

Literatura

- [1] Hejtmánek Martin, FIT VUT Brno: *Analýza pohybu objektů v sekvenci obrazů*, 2005
- [2] *Wikipedia*, internetová encyklopedie
<http://www.wikipedia.org>
- [3] *CVonline*, Compendium of Computer Vision, The University of Edinburgh
<http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [4] Bernd Viros, Stanford University, *Human visual perception*, 2001
<http://www.stanford.edu/class/ee368b/Handouts/09-HumanPerception.pdf>
- [5] Ciftcioglu, Bittermann, Sariyildiz, Delft University of Technology, *Modelling Human Vision*
http://www.bk.tudelft.nl/live/binaries/9c0182ab-14e6-4594-b876-20467e6c1f36/doc/Mod_Hum_Vis.pdf
- [6] Mark Duchaineau a kol., Los Alamos National Laboratory,
ROAMing terrain: Realtime Optimally Adapting Meshes, 1997
<http://www.llnl.gov/graphics/ROAM/roam.pdf>
- [7] Takeo Miyasaka, Kazuo Araki, Chukyo University, Japan,
Development of real time 3D measurement system using intensity ratio method, 2002
<http://www.isprs.org/commission3/proceedings02/papers/paper015.pdf>
- [8] John A. Porche, North Carolina A&T State University,
An architecture for quad-tree based image compression, 2001
<http://www.ncat.edu/~porche/research/final.PDF>
- [9] Giorgio Metta, Lorenzo Natale, Hough Transform, Lira-Lab, *Hough Transform*, 2006
<http://www.liralab.it/teaching/SINA/slides/sina-image-processing-scene-analysis.pdf>
- [10] G. Carpenter, S. Grossberg, D. Rosen, Boston University,
An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition, 1991
<http://cns.bu.edu/Profiles/Grossberg/CarGroRos1991INNART2A.pdf>

- [11] Mark. A. Kon, Leszek Plaskota, Boston University, University of Warsaw, *Neural Networks, Radial Basis Functions, and Complexity*, 2000
<http://math.bu.edu/people/mkon/nnpap3.pdf>
- [12] Reinhard Koch, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung *Model-Based 3-D Scene Analysis from Stereoscopic Image Sequences*, 1998
<http://graphics.cs.msu.su/library/publications/mb3d.pdf>
- [13] D. Heeger, Stanford University, J. Bergen, SRI David Sarnoff Research Center, *Pyramid-based texture analysis/synthesis*, 1995
<http://delivery.acm.org/10.1145/220000/218446/p229-Heeger.pdf?key1=218446&key2=1173319711&coll=portal&dl=ACM&CFID=18586673&CFTOKEN=66200180>
- [14] E. Adelson a kol., *Pyramid Methods in Image Processing*, 1984
http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf
- [15] Jean Hsang Lin a kol., IEEE, *Nonlinear Filtering by Threshold Decomposition*
<http://web.njit.edu/~ansari/papers/IP99.pdf>
- [16] Zbigniew Wasik and Allesandro Saffiotti, Center for Applied Autonomous System, *Robust Color Segmentation for the RoboCup Domain*, 2002
<ftp://aass.oru.se/pub/saffiotti/robot/icpr02.pdf>
- [17] J. Odobez, P. Bouthemy, Campus Universitaire de Beaulieu, *Segmentation of moving regions from background in an image sequence acquired with a mobile camera*
http://www.irisa.fr/vista/Papers/1997_chap_odobez.pdf
- [18] Ying Wu, Northwestern University, *Segmentation: Clustering, Graph Cut and EM*
<http://www.ece.northwestern.edu/~yingwu/teaching/EECS432/Notes/segmentation.pdf>
- [19] F. Galland a kol., Ecole Nationale Supérieure de Physique de Marseille, *Active Grid Segmentation by Stochastic Complexity Minimization Adapted to Different Noise Models*
http://www.fresnel.fr/perso/galland/Galland03_PSIP_preprint.pdf
- [20] B. Horn, B. Schunck, Massachusetts Institute of Technology, *Determining Optical Flow*, 1980
http://people.csail.mit.edu/bkph/papers/Optical_Flow_OPT.pdf

- [21] S. Bolouki a kol., Stanford University, *Optical Illusion*, 2006
<http://www.stanford.edu/class/cs229/proj2006/Bolouki-OpticalIllusion.pdf>
- [22] Chris Roberts, Centrum for Critical Infrastructure Protection,
Biometric Technologies – Palm and Hand, 2006
<http://www.ccip.govt.nz/ccip-publications/ccip-reports/Biometrics%20Technologies%20-%20Palm%20and%20Hand.pdf>

Seznam příloh

Příloha 1. – Manuál k programu ASAP

Příloha 2. – CD s kompletním obsahem práce včetně zdrojových kódů programu