



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

**CLIFFORDOVY ALGEBRY V KOLORIMETRII A ANALÝZE
OBRAZU**

CLIFFORD ALGEBRAS IN COLOUR THEORY AND IMAGE ANALYSIS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Radek Tichý

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Jaroslav Hrdina, Ph.D.

BRNO 2018

Zadání diplomové práce

Ústav:	Ústav matematiky
Student:	Bc. Radek Tichý
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	doc. Mgr. Jaroslav Hrdina, Ph.D.
Akademický rok:	2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Cliffordovy algebry v kolorimetrii a analýze obrazu

Stručná charakteristika problematiky úkolu:

Barevné prostory je možné reprezentovat pomocí různých Cliffordových algeber. Algebraické vlastnosti je pak možné využívat pro práci s barevným prostorem a pro snadnější aplikaci standardních filtrů.

Cíle diplomové práce:

Nastudování algebraické teorie Cliffordových algeber a jejich modelů. Nastudování základů kolorimetrie. Vlastní návrh řešení pro práci se zvoleným barevným prostorem.

Seznam doporučené literatury:

PERWASS, C. Geometric algebra with applications in engineering. Berlin: Springer, c2009. ISBN 35-408-9067-X.

DORST, L., D. H. F. FONTIJNE a S. MANN. Geometric algebra for computer science: an object-oriented approach to geometry. Amsterdam: Elsevier, c2007. Morgan Kaufmann series in computer graphics. ISBN 978-0-12-369465-2.

ELL, T. A. Quaternion fourier transforms for signal and image processing. 1. London: Wiley-ISTE, 2014. ISBN 978-184-8214-781.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá využitím konformní geometrické algebry CGA pro zpracování barevného obrazu. Zaměřuje se především na segmentaci barev. K tomuto účelu není vhodné pracovat v barevném prostoru RGB. Z tohoto důvodu je představen prostor CIELAB, který je opatřen euklidovskou metrikou odpovídající lidskému vnímání barev. Součástí práce je návrh algoritmu využívajícího CGA pro detekci objektu v obraze na základě barevných vlastností. Závěrečná část práce je věnována prokládání bodů v prostoru sférou využívající metodu nejmenších čtverců. Pro měření vzdáleností je v CGA použit skalární součin. Aproximace bodů sférou je poté použita na barevný obraz za účelem úpravy vzdáleností mezi barvami.

Abstract

This thesis deals with conformal geometric algebra CGA for colour image processing, particularly with colour segmentation. For this reason it is not sufficient to work in RGB colour space. It is more convenient to use a colour space called CIELAB. CIELAB is endowed by Euclidean metric corresponding with human perception of colours. Afterwards an algorithm for an object detection via CGA based on colour differences is included. The final part of the thesis deals with least squares fitting of sphere to points using CGA. The sphere fitting is then used to adjust colour differences in an image to improve the algorithm for an object detection.

klíčová slova

geometrická algebra, CGA, segmentace obrazu, barevný prostor CIELAB

key words

geometric algebra, CGA, image segmentation, colour space CIELAB

Prohlašuji, že jsem diplomovou práci *Cliffordovy algebry v kolorimetrii a analýze obrazu* vypracoval samostatně pod vedením doc. Mgr. Jaroslava Hrdiny, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Bc. Radek Tichý

Chtěl bych poděkovat především vedoucímu práce doc. Mgr. Jaroslavu Hrdinovi, Ph.D. za pozitivní a vstřícný přístup a četné rady při vedení mé diplomové práce a také všem přátelům a rodině za neutuchající psychickou podporu v průběhu celého studia.

Bc. Radek Tichý

Obsah

Úvod	13
1 Geometrické algebry	14
1.1 Úvod do geometrických algeber	14
1.2 Operace na geometrických algebrách	15
1.2.1 Vnější součin (wedge)	16
1.2.2 Vnitřní součin a skalární součin	18
1.2.3 Chevalleyova konstrukce geometrické algebry	19
1.2.4 Vlastnosti geometrického součinu	19
2 Konformní geometrická algebra - CGA	21
2.1 Základní geometrické objekty	22
2.1.1 Sféra	24
2.1.2 Rovina	24
2.1.3 Kružnice a přímka	25
2.1.4 Dvojbod	25
2.1.5 Podalgebry CGA	25
3 Aplikace CGA v detekci hran a segmentaci obrazu	27
3.1 Detekce hran	27
3.1.1 Rotace	28
3.1.2 Prewittové filtr	30
3.2 Segmentace barev	34
3.2.1 Barevný prostor CIELAB	34
3.2.2 Implementace CGA v prostoru CIELAB	36
3.2.3 Algoritmus pro detekci objektu	37
4 Aplikace CGA pro popis okolí v prostoru barev	42
4.1 Hledání aproximace	42
4.2 Implementace v MATLABU	47
4.3 Využití prokládání bodů sférou ve zpracování obrazu	48
Závěr	52
Literatura	54
A Algoritmus pro detekci objektu	56
B Pomocné funkce	59

Úvod

Pojem geometrické algebry v tomto textu chápeme jako alternativní název pro Cliffordovy algebry při aplikacích využívajících euklidovskou geometrii. Na teorii geometrických algeber můžeme nahlížet jako na matematický jazyk kombinující algebru a geometrii. Umožňují nám jednoduše reprezentovat a řešit velké množství geometrických problémů. Z tohoto důvodu jsou dnes geometrické algebry využívány v robotice či počítačové grafice a jejich vliv stále roste s novými počítačovými softwary, které poskytují jejich stále se zlepšující implementaci.

Cílem této práce je zkoumat možnosti využití geometrických algeber ve zpracování obrazu, především v segmentaci barevného obrazu. K tomuto účelu lze velice efektivně využít konformní geometrickou algebru (CGA), což je speciální typ geometrické algebry umožňující jednoduchou interpretaci geometrických objektů, jako jsou body, roviny, sféry, atd. a následnou práci s nimi. CGA nám umožní pracovat s celým barevným obrazem, na rozdíl od klasického přístupu, kde se obraz typicky převádí do odstínů šedi nebo se pracuje s jednotlivými barevnými složkami zvlášť. Při segmentaci barevného obrazu pomocí CGA není vhodné pracovat v barevném prostoru RGB z důvodu k tomuto účelu nevyhovujících vlastností barev. Vhodnějším prostorem je CIELAB, který díky euklidovské metrice mezi jednotlivými barvami navíc vhodně koresponduje s vlastnostmi CGA. Kombinace CGA a prostoru CIELAB přináší v segmentaci obrazu celou řadu výhod.

Práce je členěna do několika hlavních kapitol. Nejprve bude vybudována teorie geometrických algeber. Geometrické algebry lze definovat více způsoby, dva z nich zde budou uvedeny s důrazem na souvislosti mezi Grassmannovou a geometrickou algebrou. Dále představíme CGA, tato algebra nám umožní reprezentovat a transformovat geometrické objekty. Třetí a čtvrtá kapitola budou věnovány aplikaci CGA pro zpracování obrazu. Konkrétně ve třetí kapitole se budeme zabývat především segmentací obrazu, kde se nám kombinací barevného prostoru CIELAB a algebry CGA podařilo ve vývojovém prostředí MATLAB, za pomoci speciálního toolboxu navrženého pro práci s geometrickými algebry, vytvořit algoritmus, který je schopný detekce zvoleného objektu na barevném obraze. Toolbox pro práci s geometrickými algebry byl v průběhu psaní této práce značně modifikován a vylepšen, k čemuž přispěly naše experimenty a následná komunikace s autory toolboxu. Zdrojový kód a některé obrázky získané při testování algoritmu jsou součástí přílohy této práce. V poslední kapitole bude představeno prokládání bodů v prostoru sférou pomocí CGA založené na metodě nejmenších čtverců. Pro výpočet optimální sféry použijeme pouze základní toolbox pro lineární algebru. Prokládání bodů sférou se nám povedlo využít pro úpravu vzdáleností barev obrazu. Tuto operaci jsme zakomponovali do algoritmu pro detekci objektu a následné experimenty ukázali, že došlo ke zlepšení dosažených výsledků.

Kapitola 1

Geometrické algebry

Geometrická algebra¹ je asociativní algebra nad polem skalárů (nejčastěji nad polem reálných čísel \mathbb{R}), která nám umožňuje jednoduchou interpretaci základních geometrických útvarů (přímka, rovina, sféra, ...) a jejich transformace pomocí operací příslušné geometrické algebry (viz [9, 12, 13]). Jak uvidíme dále v této kapitole, základní operací definovanou v geometrických algebrách je takzvaný geometrický součin (v anglické literatuře se používá pojem „geometric product“). V této kapitole popíšeme strukturu geometrických algeber a zvláštní pozornost bude poté v kapitole 2 věnována tzv. konformní geometrické algebře, která bude klíčová pro aplikaci v kapitole 3 této práce.

1.1 Úvod do geometrických algeber

Způsobů jak lze definovat geometrickou algebru je více (viz [12]). Pro náš účel použijeme následující.

Definice 1.1. Necht' V je vektorový prostor (nad \mathbb{R}) a Q je kvadratická forma na V . Asociativní algebru nazveme *geometrická algebra* $\mathbb{G}(Q)$ nedegenerované kvadratické formy Q na vektorovém prostoru V , jestliže obsahuje V a $\mathbb{R} = \mathbb{R} \cdot 1$ jako různé podprostory, takové že

1. $\mathbf{x}^2 = Q(\mathbf{x})$ pro $\forall \mathbf{x} \in V$
2. V generuje algebru $\mathbb{G}(Q)$ nad \mathbb{R}
3. Algebra $\mathbb{G}(Q)$ není generovaná žádným vlastním podprostorem V .

Vezmeme-li například vektorový prostor $V = \mathbb{R}^{p,q}$, tedy \mathbb{R}^{p+q} s kvadratickou formou typu² (p, q) , podle definice 1.1 máme jednoznačně určenou algebru generovanou tímto prostorem, kterou budeme značit $\mathbb{G}_{p,q}$. Základními prvky této algebry jsou báze vektory prostoru $\mathbb{R}^{p,q}$, které tvoří ortogonální bázi $\{e_1, e_2, \dots, e_n\}$, kde $n = p + q$. Podle typu (*signature*) algebry (kvadratické formy) platí pro tyto vektory dle podmínky 1 v definici 1.1 následující vztahy:

$$e_i^2 = 1 \quad \text{pro } 1 \leq i \leq p, \quad e_i^2 = -1 \quad \text{pro } p < i \leq n.$$

¹V některých zdrojích se používá pojem Cliffordovy algebry, pojem geometrické algebry se upřednostňuje zejména v případě aplikací využívajících euklidovskou geometrii.

²V anglické literatuře se používá pojem „signature“.

Tyto prvky budeme v geometrické algebře nazývat *bázové elementy* stupně 1. Násobení na geometrické algebře nazýváme *geometrický součin* a z definice lze odvodit základní pravidlo pro geometrický součin mezi těmito elementy. Mějme libovolné bázové elementy stupně 1, tedy e_i, e_j , $i \neq j$, z důvodu distributivity (algebra je distributivní z definice) platí

$$(e_i + e_j)^2 = e_i^2 + e_i e_j + e_j e_i + e_j^2,$$

zároveň, protože elementy e_i a e_j jsou ortogonální a skalární součin je bilineární, platí

$$(e_i + e_j)^2 = Q(e_i + e_j) = e_i^2 + e_j^2,$$

složením těchto dvou vztahů dostáváme

$$e_i^2 + e_i e_j + e_j e_i + e_j^2 = e_i^2 + e_j^2 \Rightarrow e_i e_j + e_j e_i = 0.$$

Tuto podmínku lze vyjádřit v ekvivalentním tvaru

$$e_i e_j = -e_j e_i, \text{ pro } i \neq j. \quad (1.1)$$

Zároveň jsme tímto postupem získali další bázové elementy $e_i e_j$, $i \neq j$. Tyto elementy jsou formální kompozicí dvou elementů stupně 1, říkáme tedy, že mají stupeň 2. Elementy stupně 2 tvoří vektorový prostor dimenze $\binom{n}{2}$ s bází $e_i e_j$, $i < j$, kterou budeme značit $e_i \wedge e_j$. Obdobným postupem lze získat bázové elementy všech stupňů až do stupně n , element stupně n (existuje pouze jeden) se nazývá *pseudoskalár* a budeme ho značit $I = e_1 \wedge e_2 \wedge \dots \wedge e_n$. Skalár 1 je potom bázový element stupně 0. Geometrická algebra dimenze n má tedy $\sum_{i=0}^n \binom{n}{i} = 2^n$ základních elementů (v tabulce 1.1 najdeme celou bázi algebry $\mathbb{G}_{3,0}$). Lineární kombinací elementů stupně k získáme k -vektory (vektory, bivektory, trivektory, atd.). Lineární kombinací elementů různých stupňů získáme obecný prvek geometrické algebry zvaný *multivektor*. V dalším popíšeme důležité operace definované v geometrických algebrách a detailněji strukturu některých důležitých algeber.

Poznámka 1.1. Dále v textu si ukážeme, že symbol \wedge můžeme chápat jako operaci v celé geometrické algebře $\mathbb{G}_{p,q}$ zvanou *wedge* nebo *vnější součin*. Význam a vlastnosti této operace budou vysvětleny v dalším úseku této práce.

1.2 Operace na geometrických algebrách

Základní operací geometrické algebry je *geometrický součin*. Dalšími dvěma důležitými operacemi, které lze odvodit z geometrického součinu, jsou vnější součin neboli *wedge* „ \wedge “ a vnitřní součin „ \ast “. Podrobnější popis operací geometrických algeber a kompletní výčet jejich vlastností najdeme v monografii [13].

Nyní uvedeme definice vnějšího a poté i vnitřního součinu. Poznamenejme nejprve, že množinu bázových elementů geometrické algebry $\mathbb{G}_{p,q}$ budeme značit $\overline{\mathbb{G}}_{p,q}$. Než přistoupíme k samotné definici vnějšího součinu, budeme potřebovat ještě jednu operaci, kterou nazýváme *projekce na stupeň*.

Definice 1.2. Označme $E \in \overline{\mathbb{G}}_{p,q}$ libovolný bázový element. Potom projekci elementu E na stupeň k označíme $\langle E \rangle_k$ a definujeme

$$\langle E \rangle_k = \begin{cases} E, & gr(E) = k, \\ 0, & gr(E) \neq k, \end{cases}$$

kde $gr(E)$ je stupeň bázového elementu.

Definici 1.2 lze snadno zobecnit na libovolný multivektor $\mathbf{A} \in \mathbb{G}_{p,q}$. Každý multivektor je totiž lineární kombinací bázevých elementů, tedy $\mathbf{A} = \sum_i a_i E_i$, $a_i \in \mathbb{R}$, $E_i \in \overline{\mathbb{G}}_{p,q}$. Operátor projekce na stupeň definujeme jako zobrazení $\langle \cdot \rangle_k : \mathbb{G}_{p,q} \rightarrow \mathbb{G}_{p,q}^k$, kde $\mathbb{G}_{p,q}^k$ je množina k -vektorů, které je distributivní, asociativní a platí $\langle a_i E_i \rangle_k = a_i \langle E_i \rangle_k$. Projekci multivektoru \mathbf{A} tedy můžeme napsat jako:

$$\langle \mathbf{A} \rangle_k = \sum_i a_i \langle E_i \rangle_k. \quad (1.2)$$

Nyní můžeme konečně definovat vztah pro výpočet vnějšího součinu dvou bázevých elementů.

Definice 1.3. Mějme $E_k, E_l \in \overline{\mathbb{G}}_{p,q}$, takové že $gr(E_k) = k$, $gr(E_l) = l$, potom vnější součin $E_k \wedge E_l$ definujeme vztahem

$$E_k \wedge E_l = \langle E_k E_l \rangle_{k+l}. \quad (1.3)$$

Vztah (1.3) lze s využitím vztahu (1.2) jednoduše rozšířit na libovolný multivektor. Pro názornou ukázkou můžeme nyní ověřit, že $e_1 e_2 e_3 = e_1 \wedge e_2 \wedge e_3$, tedy že vyjádření $e_1 e_2 e_3$ a $e_1 \wedge e_2 \wedge e_3$ (ve smyslu operace vnější součin) představují tentýž bázevý element.

$$e_1 \wedge e_2 \wedge e_3 = \langle e_1 e_2 \rangle_{1+1} \wedge e_3 = \langle (e_1 e_2) e_3 \rangle_{2+1} = \langle e_1 e_2 e_3 \rangle_3 = e_1 e_2 e_3,$$

stejný výsledek získáme i jiným „uzávorkováním“

$$e_1 \wedge e_2 \wedge e_3 = e_1 \wedge \langle e_2 e_3 \rangle_{1+1} = \langle e_1 (e_2 e_3) \rangle_{1+2} = \langle e_1 e_2 e_3 \rangle_3 = e_1 e_2 e_3.$$

Celou bázi geometrické algebry $\mathbb{G}_{3,0}$ vyjádřenou pomocí vnějšího součinu najdeme v tabulce 1.1 a tutéž bázi vyjádřenou pomocí geometrického součinu demonstruje tabulka 1.2.

Podobným způsobem, jakým jsme nadefinovali vnější součin, nadefinujeme i *vnitřní součin* a také operaci *skalární součin* na bázevých elementech geometrické algebry.

Definice 1.4. Mějme $E_k, E_l \in \overline{\mathbb{G}}_{p,q}$, takové že $gr(E_k) = k$, $gr(E_l) = l$, $k \neq 0, l \neq 0$. Vnitřní součin značíme $E_k \star E_l$, skalární součin značíme $E_k \cdot E_l$ a obě tyto operace definujeme vztahy:

$$\begin{aligned} E_k \star E_l &= \langle E_k E_l \rangle_{|k-l|}, \\ E_k \cdot E_l &= \langle E_k E_l \rangle_0, \end{aligned}$$

$E_k \star E_l = E_k \cdot E_l = 0$ pokud $k = 0$ nebo $l = 0$.

Tuto definici lze opět s využitím vztahu (1.5) jednoduše aplikovat na libovolné multivektory. Dále se podíváme na vlastnosti těchto operací.

1.2.1 Vnější součin (wedge)

Operace wedge může být v jistém smyslu použita pro určování rovnoběžnosti dvou geometrických objektů (přímek, rovin, ...), proto se jí budeme zabývat podrobněji. Mějme $\mathbf{A}, \mathbf{B} \in \mathbb{G}_{p,q}$ libovolné multivektory z geometrické algebry se signaturou (p, q) , potom pro libovolné $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{G}_{p,q}$ platí následující vlastnosti:

$$1. \text{ Distributivita}^3: \mathbf{A} \wedge (\mathbf{B} + \mathbf{C}) = \mathbf{A} \wedge \mathbf{B} + \mathbf{A} \wedge \mathbf{C}, \quad (\mathbf{B} + \mathbf{C}) \wedge \mathbf{A} = \mathbf{B} \wedge \mathbf{A} + \mathbf{C} \wedge \mathbf{A}$$

$$2. \text{ Asociativita: } \mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C}) = (\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{C}$$

Pro vektory (lineární kombinace bázových elementů stupně 1) $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{p,q} \subset \mathbb{G}_{p,q}$ navíc platí:

$$3. \text{ Antikomutativita: } \mathbf{u} \wedge \mathbf{v} = -(\mathbf{v} \wedge \mathbf{u})$$

Z této vlastnosti lze odvodit důsledek. Pro libovolný vektor \mathbf{u}

$$\mathbf{u} \wedge \mathbf{u} = -(\mathbf{u} \wedge \mathbf{u}) = 0. \tag{1.4}$$

Navíc pro $a, b \in \mathbb{R}$ platí:

$$4. \text{ Násobení skalárem: } (a\mathbf{A}) \wedge (b\mathbf{B}) = ab(\mathbf{A} \wedge \mathbf{B})$$

Tedy konečně pro vektor \mathbf{u} dostáváme vztah

$$(a\mathbf{u}) \wedge (b\mathbf{u}) = 0,$$

kde $a, b \in \mathbb{R}$. To znamená, že wedge dvou rovnoběžných vektorů je 0. Později v této práci si ukážeme, že v geometrických algebrách lze další geometrické objekty vyjádřit pomocí wedge vektorů, funkce wedge tedy může být využita například pro „měření“ rovnoběžnosti rovin. Vlastnost 3 operace wedge a tedy i vztah (1.4) však neplatí obecně pro multivektory, což budeme demonstrovat na následujícím jednoduchém příkladu.

Příklad 1.1. Necht' $\mathbf{A} = e_1 \wedge e_2 + e_3$, potom

$$\begin{aligned} \mathbf{A} \wedge \mathbf{A} &= (e_1 \wedge e_2 + e_3) \wedge (e_1 \wedge e_2 + e_3) \\ &= (e_1 \wedge e_2) \wedge (e_1 \wedge e_2) + (e_3 \wedge e_3) + (e_1 \wedge e_2 \wedge e_3) + (e_3 \wedge e_1 \wedge e_2) \\ &= -(e_2 \wedge e_1) \wedge (e_1 \wedge e_2) + (e_1 \wedge e_2 \wedge e_3) + (-e_1 \wedge e_3 \wedge e_2) \\ &= -e_2 \wedge (e_1 \wedge e_1) \wedge e_2 + (e_1 \wedge e_2 \wedge e_3) + (e_1 \wedge e_2 \wedge e_3) \\ &= 2(e_1 \wedge e_2 \wedge e_3). \end{aligned}$$

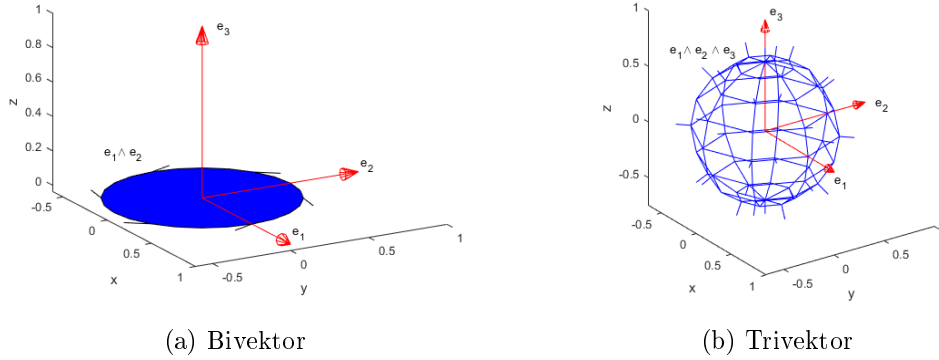
V prvním kroku jsme využili distributivitu, v druhém kroku antikomutativitu vektorů a poté asociativitu.

Bázový element	Stupeň
1	0
e_1, e_2, e_3	1
$e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3$	2
$I = e_1 \wedge e_2 \wedge e_3$	3

Tabulka 1.1: Geometrická algebra $\mathbb{G}_{3,0}$

Ve 3D geometrické algebre $\mathbb{G}_{3,0}$ (viz Tabulka 1.1) bázové vektory představují základní vektory euklidovského 3D prostoru. Tedy libovolný euklidovský vektor vyjádříme

$$(x, y, z) = xe_1 + ye_2 + ze_3.$$



Obrázek 1.1: Geometrická interpretace operace wedge

Wedge dvou vektorů, tedy bivektor potom v euklidovském prostoru představuje rovinu a trivektor neboli pseudoskalár představuje objem. Pro ukázkou geometrické interpretace bivektoru a trivektoru jsme použili toolbox pro Matlab s názvem GABLE [5] (obr. 1.1). Bivektory jsou tu znázorněny pomocí kruhu, kde rovina, ve které kruh leží, je rovina určená bivektorem a plocha kruhu je velikost jeho normy. Trivektory jsou znázorněny pomocí sféry, jejíž objem je norma příslušného trivektoru.

1.2.2 Vnitřní součin a skalární součin

Vnitřní a skalární součin jsou obecně v geometrické algebře dvě rozdílné operace, ačkoliv mají některé společné vlastnosti. Mějme libovolné $\mathbf{A}, \mathbf{B} \in \mathbb{G}_{p,q}$, $a, b \in \mathbb{R}$, potom pro vnitřní součin platí následující vlastnosti:

1. Distributivita: $\mathbf{A} \star (\mathbf{B} + \mathbf{C}) = \mathbf{A} \star \mathbf{B} + \mathbf{A} \star \mathbf{C}$, $(\mathbf{B} + \mathbf{C}) \star \mathbf{A} = \mathbf{B} \star \mathbf{A} + \mathbf{C} \star \mathbf{A}$
2. Asociativita: $\mathbf{A} \star (\mathbf{B} \star \mathbf{C}) = (\mathbf{A} \star \mathbf{B}) \star \mathbf{C}$
3. Násobení skalárem: $(a\mathbf{A}) \star (b\mathbf{B}) = ab(\mathbf{A} \star \mathbf{B})$

Pro vektory $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{p,q}$ navíc platí:

4. Komutativita: $\mathbf{u} \star \mathbf{v} = \mathbf{v} \star \mathbf{u}$

Tyto vlastnosti platí i pro skalární součin. Stejně jako u vnějšího součinu pro $\mathbf{A} := \sum_i a_i E_i$, $\mathbf{B} := \sum_j b_j E_j$ platí pro obě tyto operace

$$\mathbf{A} \star \mathbf{B} = \sum_{i,j} a_i b_j (E_i \star E_j), \quad \mathbf{A} \cdot \mathbf{B} = \sum_{i,j} a_i b_j (E_i \cdot E_j). \quad (1.5)$$

Oproti operaci wedge, která zvyšuje stupeň prvků multivektoru, vnitřní součin stupeň snižuje. Všimněme si, že výsledkem skalárního součinu je vždy skalár a pro vektory, tedy lineární kombinace bázových elementů stupně 1, operace skalární a vnitřní součin splývají. Při aplikaci geometrické algebry CGA v této práci bude hrát skalární součin významnou roli, protože lze pomocí této operace měřit vzdálenost mezi vektory a stejně jako v klasickém euklidovském prostoru platí, že skalární součin dvou navzájem kolmých euklidovských vektorů je 0.

³Operace sčítání funguje v geometrických algebrách obvyklým způsobem, tedy sčítáme skaláry příslušející ke stejným bázovým elementům.

1.2.3 Chevalleyova konstrukce geometrické algebry

Při konstrukci geometrické algebry pomocí definice 1.1 jsme nejprve získali operaci geometrický součin a z něho jsme poté definovali operace vnitřní součin a vnější součin. V této kapitole si ukážeme jinou konstrukci geometrické algebry, kde budeme postupovat opačně. Nechť V je vektorový prostor nad \mathbb{R} s nedegenerovanou kvadratickou formou Q , která je zúžením skalárního součinu $\langle \mathbf{u}, \mathbf{v} \rangle$, $u, v \in V$. Mějme na vektorovém prostoru generovaným $\overline{\mathbb{G}}_{p,q}$ definovaný vnější součin splňující vlastnosti z kapitoly 1.2.1, tím získáváme asociativní algebru, která se nazývá *Grassmannova* nebo také vnější algebra a značíme ji $\wedge V$ (pro prvky této algebry zachováme názvosloví z kapitoly 1.1). Naším záměrem je na Grassmannově algebře nadefinováním geometrického součinu zkonstruovat geometrickou algebru. Nejprve definujeme operaci *levá kontrakce*, která je rozšířením skalárního součinu na multivektory:

Definice 1.5. Mějme vektory $\mathbf{u}, \mathbf{v} \in V$ a multivektory $A, B, C \in \wedge V$, potom levou kontrakci $A \lrcorner B \in \wedge V$ definujeme vztahy:

1. $\mathbf{u} \lrcorner \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle$,
2. $\mathbf{u} \lrcorner (A \wedge B) = (\mathbf{u} \lrcorner A) \wedge B + \hat{A} \wedge (\mathbf{u} \lrcorner B)$,
3. $(A \wedge B) \lrcorner C = A \lrcorner (B \lrcorner C)$,

kde \hat{A} značí involuci stupně⁴.

Pomocí operace levá kontrakce nyní definujeme geometrický součin mezi vektorem $\mathbf{u} \in V$ a libovolným multivektorem $A \in \wedge V$ vztahem

$$\mathbf{u}A = \mathbf{u} \lrcorner A + \mathbf{u} \wedge A. \quad (1.6)$$

Vztah (1.6) lze díky linearitě a asociativitě rozšířit na libovolné multivektory z $\wedge V$, čímž získáváme geometrickou algebru $\mathbb{G}_{p,q}$.

1.2.4 Vlastnosti geometrického součinu

Pro úplnost uvedeme, podobně jako v případě vnějšího a vnitřního součinu, základní vlastnosti geometrického součinu, které lze odvodit přímo z definice 1.1 nebo vztahu (1.6).

Pro libovolné multivektory $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{G}_{p,q}$, a libovolné $a \in \mathbb{R}$ platí:

1. Distributivita: $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{A}\mathbf{B} + \mathbf{A}\mathbf{C}$, $(\mathbf{B} + \mathbf{C})\mathbf{A} = \mathbf{B}\mathbf{A} + \mathbf{C}\mathbf{A}$
2. Asociativita: $\mathbf{A}(\mathbf{B}\mathbf{C}) = (\mathbf{A}\mathbf{B})\mathbf{C}$
3. Násobení skalárem: $a\mathbf{A} = \mathbf{A}a$

V případě vektorů \mathbf{u}, \mathbf{v} platí, že geometrický součin je součtem operace wedge a skalárního součinu

$$\mathbf{u}\mathbf{v} = \mathbf{u} \wedge \mathbf{v} + \mathbf{u} \cdot \mathbf{v}. \quad (1.7)$$

⁴ $\hat{\mathbf{u}}$ je operace *involuce stupně k -vektoru* $\mathbf{u} \in \wedge^k V$ definována vztahem $\hat{\mathbf{u}} = (-1)^k \mathbf{u}$.

Pokud do vztahu (1.7) dosadíme bázové vektory prostoru $\mathbb{R}^{p,q} \subset \mathbb{G}_{p,q}$, potom vzhledem k ortogonalitě báze $\mathbb{R}^{p,q}$ je člen skalárního součinu nulový a dostáváme tedy:

$$e_i e_j = e_i \wedge e_j, \quad i \neq j.$$

Pro geometrický součin mezi různými bázovými vektory platí antikomutativita (viz vztah 1.1), ovšem pro obecné vektory tato vlastnost neplatí.

Příklad 1.2. Necht' $\mathbf{a} = e_1 + e_2$, $\mathbf{b} = e_2$, potom:

$$\mathbf{ab} = (e_1 + e_2)e_2 = e_1 e_2 + 1,$$

$$\mathbf{ba} = e_2(e_1 + e_2) = e_2 e_1 + 1 = -e_1 e_2 + 1 \neq -(e_1 e_2 + 1).$$

Můžeme také naopak vyjádřit vnější součin a skalární součin vektorů \mathbf{u}, \mathbf{v} pomocí geometrického:

$$\mathbf{u} \wedge \mathbf{v} = \frac{1}{2}(\mathbf{uv} - \mathbf{vu}),$$

$$\mathbf{u} \cdot \mathbf{v} = \frac{1}{2}(\mathbf{uv} + \mathbf{vu}).$$

Bázový element	Stupeň
1	0
e_1, e_2, e_3	1
$e_1 e_2, e_1 e_3, e_2 e_3$	2
$I = e_1 e_2 e_3$	3

Tabulka 1.2: Geometrická algebra $\mathbb{G}_{3,0}$

Kapitola 2

Konformní geometrická algebra - CGA

Konformní geometrická algebra (dále CGA) je speciální typ $5D$ algebry $\mathbb{G}_{4,1}$. Přestože se jedná o algebru nad prostorem $\mathbb{R}^{4,1}$, budeme ji využívat pro aplikace ve $3D$ prostoru. Důvodem je, že CGA umožňuje velmi jednoduše a intuitivně pracovat s geometrickými objekty ve $3D$ prostoru, realizovat konformní transformace (tedy takové transformace, které zachovávají velikosti úhlů) a v neposlední řadě nám umožňuje velice výhodně reprezentovat body a sféry ve $3D$ prostoru, což využijeme právě pro naši aplikaci v kapitole 3.

Mějme klasickou ortogonální bázi vektorů $\{e_1, e_2, e_3, e_+, e_-\}$, důvod označení elementů e_+, e_- je vzhledem k signatuře algebry $\mathbb{G}_{4,1}$ zřejmý. Na tomto prostoru zvolíme neortogonální bázi tak, že zavedeme nové báze vektory e_0, e_∞ definované vztahy:

$$e_0 = \frac{1}{2}(e_- - e_+), \quad e_\infty = e_- + e_+. \quad (2.1)$$

S takto nadefinovanými elementy dostáváme tedy vektorovou bázi CGA $\{e_1, e_2, e_3, e_0, e_\infty\}$,

Bázový element	Stupeň
1	0
$e_1, e_2, e_3, e_0, e_\infty$	1
$e_1 \wedge e_2, e_1 \wedge e_3, e_1 \wedge e_0, e_1 \wedge e_\infty,$ $e_2 \wedge e_3, e_2 \wedge e_0, e_2 \wedge e_\infty, e_3 \wedge e_0,$ $e_3 \wedge e_\infty, e_0 \wedge e_\infty$	2
$e_1 \wedge e_2 \wedge e_3, e_1 \wedge e_2 \wedge e_0, e_1 \wedge e_2 \wedge e_\infty,$ $e_1 \wedge e_3 \wedge e_0, e_1 \wedge e_3 \wedge e_\infty, e_1 \wedge e_0 \wedge e_\infty,$ $e_2 \wedge e_3 \wedge e_0, e_2 \wedge e_3 \wedge e_\infty, e_2 \wedge e_0 \wedge e_\infty,$ $e_3 \wedge e_0 \wedge e_\infty$	3
$e_1 \wedge e_2 \wedge e_3 \wedge e_0,$ $e_1 \wedge e_2 \wedge e_3 \wedge e_\infty,$ $e_1 \wedge e_2 \wedge e_0 \wedge e_\infty,$ $e_1 \wedge e_3 \wedge e_0 \wedge e_\infty,$ $e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	4
$e_1 \wedge e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$	5

Tabulka 2.1: CGA báze elementy

kde e_1, e_2, e_3 představují Euklidovské bázové vektory, e_0 reprezentuje počátek 3D souřadnicové soustavy a e_∞ reprezentuje nekonečno. Jelikož jsme při definování geometrických algeber až dosud uvažovali pouze ortogonální vektorové báze, což pro naši novou volbu neplatí, některé vlastnosti, jako například antikomutativita geometrického součinu mezi bázovými vektory, zde budou porušeny. Ačkoliv tímto přechodem k neortogonální bázi ztratíme některé vlastnosti, důvodem tohoto formálního kroku je značné zjednodušení a větší přehlednost geometrických reprezentací. Celou bázi CGA najdeme v tabulce 2.1. Oba bázové vektory e_∞, e_0 jsou v druhé mocnině rovny nule, což lze snadno ověřit přímým výpočtem ze vztahů (2.1):

$$\begin{aligned} e_\infty^2 &= (e_- + e_+)(e_- + e_+) = e_-^2 + e_+^2 + e_-e_+ + e_+e_- = e_-^2 + e_+^2 + e_-e_+ - e_-e_+ = -1 + 1 = 0, \\ e_0^2 &= \frac{1}{2}(e_- - e_+)\frac{1}{2}(e_- - e_+) = \frac{1}{4}(e_-^2 + e_+^2 - e_-e_+ - e_+e_-) = \frac{1}{4}(e_-^2 + e_+^2 - e_-e_+ + e_-e_+) = 0. \end{aligned}$$

Stejným způsobem ukážeme, čemu se rovná skalární a geometrický součin mezi těmito elementy. Díky komutativitě skalárního součinu dvou vektorů víme, že $e_\infty \cdot e_0 = e_0 \cdot e_\infty$, stačí tedy vypočítat pouze jeden vztah:

$$e_\infty \cdot e_0 = (e_- + e_+)\frac{1}{2}(e_- - e_+) = \frac{1}{2}(e_- \cdot e_- - e_+ \cdot e_+ + e_+ \cdot e_- - e_- \cdot e_+) = \frac{1}{2}(-1 - 1) = -1.$$

V případě geometrického součinu mezi dvěma vektory platí vztah (1.7) a jelikož už víme, čemu se rovná skalární součin $e_\infty \cdot e_0$, dostáváme

$$\begin{aligned} e_\infty e_0 &= e_\infty \wedge e_0 + e_\infty \cdot e_0 = e_\infty \wedge e_0 - 1, \\ e_0 e_\infty &= e_0 \wedge e_\infty + e_0 \cdot e_\infty = -e_\infty \wedge e_0 - 1. \end{aligned}$$

Všechny odvozené výsledky jsou přehledně znázorněny vztahy:

$$\begin{aligned} e_\infty^2 &= e_0^2 = 0 \\ e_\infty \cdot e_0 &= e_0 \cdot e_\infty = -1 \\ e_\infty e_0 &= e_\infty \wedge e_0 - 1 \\ e_0 e_\infty &= -e_\infty \wedge e_0 - 1 \\ e_\infty e_0 &= -e_0 e_\infty - 2 \end{aligned}$$

Výsledek skalárního součinu nám potvrzuje, že vektory e_∞, e_0 nejsou navzájem ortogonální, jsou však ortogonální vůči všem ostatním bázovým vektorům.

2.1 Základní geometrické objekty

Hlavní předností CGA, která již byla zmíněna, je možnost snadné reprezentace základních geometrických objektů a práce s nimi. V této kapitole se budeme zabývat těmito reprezentacemi.

Bod $(x, y, z) \in \mathbb{R}^3$ reprezentujeme v CGA vztahem

$$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2 e_\infty + e_0, \quad (2.2)$$

kde $\mathbf{x} = xe_1 + ye_2 + ze_3$. Poznamenejme, že \mathbf{x}^2 ve vztahu (2.2) představuje skalární součin vektoru báze $\{e_1, e_2, e_3\}$ se sebou samým, jehož výsledkem je:

$$\mathbf{x}^2 = x^2 + y^2 + z^2 = \|\mathbf{x}\|^2.$$

Jedná se tedy o druhou mocninu obvyklé euklidovské normy vektoru. Dosazením $(0, 0, 0)$ do vztahu (2.2) se snadno přesvědčíme, že bázový element e_0 skutečně reprezentuje počátek souřadnicového systému

$$P(0, 0, 0) = e_0.$$

Nyní ještě ukážeme, čemu se rovná skalární součin mezi dvěma body. Mějme body $X = \mathbf{x} + \frac{1}{2}\mathbf{x}^2e_\infty + e_0$ a $Y = \mathbf{y} + \frac{1}{2}\mathbf{y}^2e_\infty + e_0$, potom

$$\begin{aligned} X \cdot Y &= (\mathbf{x} + \frac{1}{2}\mathbf{x}^2e_\infty + e_0) \cdot (\mathbf{y} + \frac{1}{2}\mathbf{y}^2e_\infty + e_0) \\ &= \mathbf{x} \cdot \mathbf{y} + \frac{1}{2}\mathbf{x}^2e_\infty \cdot e_0 + \frac{1}{2}\mathbf{y}^2e_0 \cdot e_\infty \\ &= \mathbf{x} \cdot \mathbf{y} - \frac{1}{2}\mathbf{x}^2 - \frac{1}{2}\mathbf{y}^2 \\ &= -\frac{1}{2}(-2\mathbf{x} \cdot \mathbf{y} + \mathbf{x}^2 + \mathbf{y}^2) \\ &= -\frac{1}{2}(\mathbf{x} - \mathbf{y})^2. \end{aligned}$$

Dostáváme tedy druhou mocninu euklidovské normy násobenou koeficientem $-\frac{1}{2}$ rozdílu dvou euklidovských vektorů odpovídajících příslušným CGA bodům

$$X \cdot Y = -\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad (2.3)$$

tedy skalární součin dvou bodů měří vzdálenost mezi nimi.

Kromě bodu lze v CGA reprezentovat další geometrické objekty, jejichž výčet nalezneme v tabulce 2.2. Pro tyto objekty lze použít dvě různé reprezentace, které jsou navzájem

Objekt	IPNS	OPNS
Bod	$P = \mathbf{x} + \frac{1}{2}\mathbf{x}^2e_\infty + e_0$	
Sféra	$S = P - \frac{1}{2}r^2e_\infty$	$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4$
Rovina	$\pi = \mathbf{n} + de_\infty$	$\pi^* = P_1 \wedge P_2 \wedge P_3 \wedge e_\infty$
Kružnice	$C = S_1 \wedge S_2$	$C^* = P_1 \wedge P_2 \wedge P_3$
Přímka	$L = \pi_1 \wedge \pi_2$	$L^* = P_1 \wedge P_2 \wedge e_\infty$
Dvojbod	$Pp = S_1 \wedge S_2 \wedge S_3$	$Pp^* = P_1 \wedge P_2$

Tabulka 2.2: IPNS a OPNS reprezentace geometrických objektů v CGA algebře. Hvězdička značí dualitu mezi zmíněnými reprezentacemi (viz [13]).

duální. Jsou to reprezentace vnitřním součinem, dále IPNS (z anglického „inner product null space“), a reprezentace vnějším součinem, dále OPNS (z anglického „outer product null space“). Lze ukázat, že pro geometrické objekty zmíněné v tabulce 2.2 platí následující. Jestliže $P(\mathbf{x}) \in \mathbb{G}_{p,q}$ je CGA reprezentace bodu $\mathbf{x} \in \mathbb{R}^3$ (viz vztah (2.2)), dále $\mathcal{A} \subseteq \mathbb{R}^3$ je jeden z euklidovských geometrických objektů z tabulky 2.2 a $A \in \mathbb{G}_{4,1}$ jeho IPNS reprezentace. Potom

$$\mathbf{x} \in \mathcal{A} \iff A \star P(\mathbf{x}) = 0. \quad (2.4)$$

Pro $A^* \in \mathbb{G}_{4,1}$, tedy OPNS reprezentaci objektu A platí

$$\mathbf{x} \in \mathcal{A} \iff A^* \wedge P(\mathbf{x}) = 0.$$

Poznámka 2.1. Všimněme si, že sféra a rovina v IPNS reprezentaci jsou CGA vektory. Jelikož bod je v CGA vždy vektor, můžeme pro sféru a rovinu ve vztahu (2.4) vnitřní součin nahradit skalárním součinem.

Nyní se blíže podíváme na význam vztahů pro jednotlivé geometrické objekty.

2.1.1 Sféra

V IPNS reprezentaci definujeme sféru S pomocí bodu P představujícího střed sféry a poloměru r vztahem

$$S = P - \frac{1}{2}r^2e_\infty. \quad (2.5)$$

Dosazením vztahu (2.2) a úpravou získáme

$$S = \mathbf{x} + \frac{1}{2}(\mathbf{x}^2 - r^2)e_\infty + e_0.$$

Tedy bod je vlastně sféra s poloměrem $r = 0$. O tom, že vztah (2.5) představuje skutečně sféru, se můžeme přesvědčit použitím vztahu (2.4) pro IPNS reprezentaci. Mějme libovolný bod $\mathbf{y} \in \mathbb{R}^3$, potom

$$\begin{aligned} S \cdot P(\mathbf{y}) &= (P - \frac{1}{2}r^2e_\infty) \cdot (\mathbf{y} + \frac{1}{2}\mathbf{y}^2e_\infty + e_0) \\ &= P \cdot P(\mathbf{y}) - \frac{1}{2}r^2e_\infty \cdot P(\mathbf{y}), \end{aligned}$$

využitím (2.3) získáváme

$$S \cdot P(\mathbf{y}) = -\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \frac{1}{2}r^2.$$

Odtud je ihned vidět, že platí

$$S \cdot P(\mathbf{y}) = 0 \iff \|\mathbf{x} - \mathbf{y}\|^2 = r^2,$$

neboli \mathbf{y} je bod ve vzdálenosti r od bodu \mathbf{x} představujícího střed sféry, množina všech takových bodů skutečně vyjadřuje sféru S . Sféru můžeme definovat také pomocí 4 bodů, kterými daná sféra prochází v OPNS reprezentaci

$$S^* = P_1 \wedge P_2 \wedge P_3 \wedge P_4.$$

2.1.2 Rovina

Rovina je definována vztahem

$$\pi = \mathbf{n} + de_\infty,$$

kde $\mathbf{n} = n_1e_1 + n_2e_2 + n_3e_3$ je 3D normálový vektor roviny a d je vzdálenost roviny od počátku. Druhým způsobem jak definovat rovinu, tedy OPNS reprezentace, je pomocí 3 bodů ležících v rovině a bodu v nekonečnu

$$\pi^* = P_1 \wedge P_2 \wedge P_3 \wedge e_\infty.$$

Ze vztahu pro OPNS reprezentaci si můžeme všimnout, že rovina je vlastně sféra s bodem v nekonečnu.

2.1.3 Kružnice a přímka

Kružnice je dána průsečíkem dvou sfér

$$C = S_1 \wedge S_2,$$

nebo třemi body, kterými prochází

$$C^* = P_1 \wedge P_2 \wedge P_3.$$

Podobně přímka je dána průsečíkem dvou rovin

$$L = \pi_1 \wedge \pi_2,$$

nebo dvěma body, které na dané přímce leží a bodem v nekonečnu

$$L^* = P_1 \wedge P_2 \wedge e_\infty.$$

Přímka je tedy opět kružnice s jedním bodem v nekonečnu.

2.1.4 Dvojbod

Posledním objektem, který tu zmíníme, je tzv. dvojbod. Jedná se, jak už název napovídá, o dvojici bodů danou buď průsečíkem tří sfér

$$Pp = S_1 \wedge S_2 \wedge S_3,$$

nebo přímo dvěma body

$$Pp^* = P_1 \wedge P_2.$$

Jedná se o $1D$ sféru.

2.1.5 Podalgebry CGA

CGA algebra v sobě obsahuje mnoho známých a velice užitečných algeber. Jednou z nich je například algebra komplexních čísel izomorfní s podalgebrou generovanou $\{1, I_e\}$, kde $I_e = e_1 \wedge e_2 \wedge e_3$. Bázový element I_e zřejmě splňuje vztah pro násobení imaginární jednotky

$$i^2 = I_e^2 = (e_1 \wedge e_2 \wedge e_3)^2 = e_1 e_2 e_3 e_1 e_2 e_3 = e_2 e_3 e_2 e_3 = -1. \quad (2.6)$$

Významnou algebrou pro aplikace ve zpracování barevného digitálního obrazu jsou kvaterniony (viz [15, 7]). Kvaterniony mohou být určeny bází $\{1, e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3\}$. Ukážeme, že identifikace imaginárních jednotek

$$i \rightarrow -e_2 \wedge e_3, \quad j \rightarrow e_1 \wedge e_3, \quad k \rightarrow -e_1 \wedge e_2$$

splňuje známé Hamiltonovy vztahy pro násobení

$$\begin{aligned} i^2 = j^2 = k^2 &= -1, \\ ij = k, \quad jk &= i, \quad ki = j. \end{aligned}$$

Druhé mocniny imaginárních jednotek lze vypočítat stejným způsobem jako ve (2.6), ukážeme si výpočet pro jednu z nich:

$$i^2 = (-e_2 \wedge e_3)^2 = e_2 e_3 e_2 e_3 = -e_2 e_2 = -1.$$

Vynásobíme-li jednotlivé imaginární jednotky mezi sebou, získáme

$$\begin{aligned} ij &= (-e_2 \wedge e_3)(e_1 \wedge e_3) = e_2 e_3 e_3 e_1 = e_2 e_1 = e_2 \wedge e_1 = k, \\ jk &= (e_1 \wedge e_3)(-e_1 \wedge e_2) = e_3 e_1 e_1 e_2 = e_3 e_2 = e_3 \wedge e_2 = i, \\ ki &= (-e_1 \wedge e_2)(-e_2 \wedge e_3) = e_1 e_2 e_2 e_3 = e_1 e_3 = e_1 \wedge e_3 = j. \end{aligned}$$

V CGA můžeme tedy realizovat rotace pomocí jednotkových kvaternionů, neboli jednotkových bivektorů. Transformací rotace se budeme dále zabývat v kapitole 3.1.

Další algebrou je například algebra duálních čísel. Duální číslo má tvar

$$x + y\epsilon,$$

kde x, y jsou skaláry a ϵ je duální jednotka s vlastností $\epsilon^2 = 0$. Tuto duální jednotku můžeme identifikovat například bázovým elementem $I_{e_0} = e_1 \wedge e_2 \wedge e_3 \wedge e_0$. Kombinací duálních čísel a kvaternionů získáme duální kvaterniony, které umožňují rotace kolem obecné osy (viz [10]). Těmito algebry se dále v této práci nebudeme zabývat, pro více detailů viz [13, 9].

Kapitola 3

Aplikace CGA v detekci hran a segmentaci obrazu

V této kapitole uvedeme několik aplikací geometrických algeber, zejména CGA, v odvětví nazvaném zpracování digitálního obrazu. Hlavní motivací k použití tohoto matematického aparátu je, že můžeme pracovat přímo s barevným digitálním obrazem jako celkem, tedy pomocí příslušné algebry lze barevný digitální obraz vyjádřit jednou maticí. Naproti tomu v klasickému přístupu například pro RGB reprezentaci máme číselnou matici pro každou jednotlivou barevnou složku (tedy 3 matice) nebo se obraz převádí na obraz intenzit barev (tedy odstínů šedi). V celé kapitole bude uvedeno pouze nezbytně nutné množství informací z teorie počítačové grafiky pro pochopení příslušných aplikací, důraz bude kladen především na samotný způsob aplikací geometrických algeber. Pro více informací o počítačové grafice a barevných prostorech viz [8, 17, 16]. Softwarem použitým pro zpracování obrazu bude MATLAB. Práci s geometrickými algebry v MATLABU umožňuje toolbox s názvem `Clifford Multivector Toolbox`, v našem případě verze 1.3 [14].

3.1 Detekce hran

Detekcí hran ve zpracování digitálního obrazu rozumíme jejich zvýraznění, kde výsledkem bývá zpravidla binární obraz hran. Hranou rozumíme oblast v obrazu, kde dochází k výrazné změně sousedních pixelů. Existuje již několik publikací o detekci hran barevného obrazu pomocí kvaternionů, viz [7, 15]. Jelikož kvaterniony jsou obsaženy v CGA jako podalgebra (viz kapitola 2.1.5), lze zde aplikovat naprosto totožný postup. My ovšem ukážeme, že pro vhodné filtry na detekci hran lze využít i další vlastnosti geometrické algebry CGA.

Pracujeme-li v barevném prostoru RGB, potom digitální obraz je vyjádřen třemi obrazovými maticemi R, G, B , které odpovídají intenzitám jednotlivých barevných složek pixelů. Pomocí geometrické algebry budeme takový obraz reprezentovat jednou maticí. Nejprve vyjádříme každý pixel jako euklidovský vektor pomocí vektorové báze $\{e_1, e_2, e_3\}$, dostaneme tedy matici nad $\mathbb{G}_{3,0}$

$$X = Re_1 + Ge_2 + Be_3. \quad (3.1)$$

Prvek výsledné matice na pozici i, j má tedy tvar $x_{ij} = r_{ij}e_1 + g_{ij}e_2 + b_{ij}e_3$, kde r_{ij}, g_{ij}, b_{ij} jsou prvky matic R, G, B (v tomto pořadí) na stejné pozici. Matice X je tedy matice, jejíž prvky jsou vektory geometrické algebry $\mathbb{G}_{3,0} \subset \mathbb{G}_{4,1}$. Dále každý pixel vyjádříme jako

CGA bod, výsledná reprezentace RGB obrazu je tedy dána maticí nad CGA

$$I = X + \frac{1}{2}X^2e_\infty + Oe_0, \quad (3.2)$$

kde X je matice dána vztahem (3.1), X^2 zde chápeme jako geometrický součin po složkách (nejde o násobení matic) a O představuje matici jedniček stejného rozměru, jako je matice X , viz vztah pro reprezentaci CGA bodu (2.2). Kvaternionové filtry pro detekci hran jsou založeny na rotaci vektorů kolem konkrétní osy. Než tedy přistoupíme k samotnému filtru, ukážeme jak lze v CGA realizovat rotace.

3.1.1 Rotace

Transformace v CGA libovolného objektu o (uvažujeme geometrické objekty, viz tabulka 2.2) jsou realizovány vztahem

$$o_t = Vo\tilde{V}, \quad (3.3)$$

kde o_t je objekt po transformaci, $V \in \mathbb{G}_{4,1}$ je tzv. versor, což je speciální typ multivektoru vyjadřující zvolenou transformaci, viz kapitola 3.3 v [13] a \tilde{V} značí jeho *reverzi*. Reverze multivektoru M je multivektor \tilde{M} , jehož každý bázový element je zapsán v opačném pořadí ve smyslu vnějšího součinu. Tedy například pro $M = 1 + e_1 \wedge e_2 + e_1 \wedge e_2 \wedge e_3$ máme

$$\tilde{M} = 1 + e_2 \wedge e_1 + e_3 \wedge e_2 \wedge e_1.$$

Protože operace reverze pouze mění znaménko příslušného bázového elementu, můžeme tuto operaci pro libovolný bázový element $E_{\langle k \rangle} \in \overline{\mathbb{G}}_{4,1}$ stupně k definovat vztahem

$$\tilde{E}_{\langle k \rangle} = (-1)^{\frac{k(k-1)}{2}} E_{\langle k \rangle}.$$

Tento vztah lze snadno rozšířit na libovolný multivektor $M = \sum_i a_i E_i$, $a_i \in \mathbb{R}$, $E_i \in \overline{\mathbb{G}}_{4,1}$

$$\tilde{M} = \sum_i a_i \tilde{E}_i.$$

Poznamenejme, že pro libovolné dva bázové elementy $E_1, E_2 \in \overline{\mathbb{G}}_{4,1}$ platí

$$(\widetilde{E_1 \wedge E_2}) = \tilde{E}_2 \wedge \tilde{E}_1, \quad (\widetilde{E_1 E_2}) = \tilde{E}_2 \tilde{E}_1.$$

Rotace realizujeme pomocí rotorů, rotor popisující rotaci podle počátku a normalizované osy $\mathbf{L} = L_1 e_1 + L_2 e_2 + L_3 e_3$ o úhel ϕ vyjádříme vztahem

$$R = e^{-\frac{\phi}{2} l}, \quad (3.4)$$

kde $l = L_1(e_2 \wedge e_3) + L_2(e_3 \wedge e_1) + L_3(e_1 \wedge e_2)$ je bivektor určující osu nebo také rovinu rotace. Rotor (3.4) lze zapsat i ve tvaru

$$R = \cos\left(\frac{\phi}{2}\right) - l \sin\left(\frac{\phi}{2}\right).$$

Dosazením rotoru R do vztahu (3.3) tedy dostáváme vztah pro rotaci libovolného geometrického objektu o

$$o_r = Ro\tilde{R}, \quad (3.5)$$

kde o_r je orotovaný objekt. Ukážeme si tento postup na příkladu.

Příklad 3.1. Uvažujme sféru se středem v bodě $(2, 0, 0)$ a poloměrem $r = 1$, tedy

$$S = (2e_1 + \frac{1}{2}4e_\infty + e_0) - \frac{1}{2}e_\infty.$$

Tuto sféru chceme rotovat o úhel $\pi/2$ podle osy z , osa rotace je tedy $\mathbf{L} = e_3$. Příslušný rotor má tedy tvar

$$R = \cos\left(\frac{\pi}{4}\right) - (e_1 \wedge e_2) \sin\left(\frac{\pi}{4}\right).$$

Reverze rotoru je

$$\tilde{R} = \cos\left(\frac{\pi}{4}\right) + (e_1 \wedge e_2) \sin\left(\frac{\pi}{4}\right).$$

Nyní vypočítáme výsledný objekt po rotaci

$$\begin{aligned} S_r &= RS\tilde{R} \\ &= \left(\frac{\sqrt{2}}{2} - (e_1 \wedge e_2)\frac{\sqrt{2}}{2}\right) S \left(\frac{\sqrt{2}}{2} + (e_1 \wedge e_2)\frac{\sqrt{2}}{2}\right) \\ &= \frac{\sqrt{2}}{2}S\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}S(e_1 \wedge e_2)\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}(e_1 \wedge e_2)S\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}(e_1 \wedge e_2)S(e_1 \wedge e_2)\frac{\sqrt{2}}{2} \\ &= \frac{2}{4}(S + S(e_1 \wedge e_2) - (e_1 \wedge e_2)S - (e_1 \wedge e_2)S(e_1 \wedge e_2)) \\ &= \frac{2}{4}(S + (2e_2 + 2e_\infty e_1 e_2 + e_0 e_1 e_2 - 0.5e_\infty e_1 e_2) - (-2e_2 + 2e_1 e_2 e_\infty + e_1 e_2 e_0 - 0.5e_1 e_2 e_\infty) \\ &\quad - (e_1 \wedge e_2)S(e_1 \wedge e_2)) \\ &= \frac{2}{4}((2e_1 + 1.5e_\infty + e_0) + 4e_2 - (2e_1 + 2e_1 e_2 e_\infty e_1 e_2 + e_1 e_2 e_0 e_1 e_2 - 0.5e_1 e_2 e_\infty e_1 e_2)) \\ &= \frac{2}{4}(1.5e_\infty + e_0 + 4e_2 + 2e_\infty + e_0 - 0.5e_\infty) \\ &= 2e_2 + 3e_\infty + 2e_0 \\ &= 2e_2 + \frac{1}{2}(4 - 1)e_\infty + e_0. \end{aligned}$$

Výsledkem je tedy opět kružnice s poloměrem 1 a se středem v bodě $(0, 2, 0)$. Došlo tedy opravdu k rotaci, kterou jsme požadovali.

Poznámka 3.1. Pomocí operace reverze lze zavést normu multivektoru $\mathbf{A} \in \mathbb{G}_{p,0}$, tedy takového, který není tvořen báзовými vektory, jejichž druhá mocnina je -1 . Tuto normu definujeme vztahem

$$\|\mathbf{A}\| = \sqrt{A \star \tilde{A}} = \sqrt{A \cdot \tilde{A}}.$$

Pro rotaci určenou vztahem (3.5) využíváme jednotkové rotory splňující

$$R\tilde{R} = 1,$$

z tohoto vztahu lze vyvodit, že pro tyto rotory platí

$$R^{-1} = \tilde{R}.$$

Zde můžeme vidět zřejmou podobnost s jednotkovými ryzími kvaterniony, kde místo reverze se používá konjugace kvaternionu, viz [15]. Tuto vlastnost splňují všechny jednotkové multivektory z $\mathbb{G}_{p,0}$, které lze zapsat ve tvaru vnějšího součinu lineárně nezávislých vektorů a zároveň neobsahují vektory se zápornou druhou mocninou. Mějme takový multivektor A , potom platí

$$\sqrt{A \star \tilde{A}} = \sqrt{A\tilde{A}}. \quad (3.6)$$

Pro libovolný multivektor $A \in \mathbb{G}_{p,0}$ s vlastností $A^2 \neq 0$, který lze zapsat ve tvaru vnějšího součinu lineárně nezávislých vektorů, tedy existuje inverzní prvek vzhledem ke geometrickému součinu definovaný vztahem

$$A^{-1} = \frac{\tilde{A}}{A\tilde{A}}. \quad (3.7)$$

Pro obecný multivektor však vztahy 3.6 a tedy ani 3.7 neplatí a inverzní prvek nemusí existovat. Pro libovolný multivektor $A \in \mathbb{G}_{p,0}$ můžeme však stále najít alespoň inverzi vůči vnitřnímu součinu A^+ pomocí vztahu

$$A^+ = \frac{\tilde{A}}{A \star \tilde{A}},$$

pro kterou platí $A \star A^+ = 1$. Pro určení normy multivektorů, které obsahují báze vektory se zápornou druhou mocninou, je potřeba nahradit reverzi operací konjugace multivektoru definovanou následujícím způsobem. Mějme báze element $E \in \overline{\mathbb{G}}_{p,q}$, potom konjugaci E^\dagger definujeme vztahem

$$E^\dagger = (-1)^r \tilde{E}, \quad r := gr_-(E),$$

kde $gr_-(E)$ je počet báze vektorů, jejichž druhá mocnina je -1 , které jsou obsaženy v E . Již několikrát zmíněným postupem lze opět operaci rozšířit na libovolný multivektor. Báze rotorů však tvoří pouze vektory s kladnou druhou mocninou, takže při rotaci si vystačíme s operací reverze.

3.1.2 Prewittové filtr

Pro detekci hran použijeme tzv. Prewittové filtr upravený pro použití v CGA. Základní Prewittové filtr využívá $2D$ konvoluci vstupního obrazu s konvolučním jádrem daným maticí

$$h = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

pro detekci horizontálních hran a transponované jádro \bar{h} pro detekci vertikálních hran. Pro náš účel použijeme bi-konvoluci danou vztahem

$$I_{out}(n, m) = \sum_{i=1}^N \sum_{j=1}^M h_L(i, j) I(n-i+1, m-j+1) h_R(i, j), \quad (3.8)$$

kde I_{out} je matice výstupního obrazu, I (viz (3.2)) je vstupní obraz a h_L respektive h_R je levá respektive pravá bi-konvoluční maska. Celý obraz získáme tím, že vztah (3.8) aplikujeme postupně na každý pixel obrazu $n = 1, \dots, N$, $m = 1, \dots, M$. Bi-konvoluční masky jsou matice o rozměrech 3×3

$$h_L = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ R & R & R \end{pmatrix}, \quad h_R = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ \tilde{R} & \tilde{R} & \tilde{R} \end{pmatrix}$$

pro detekci horizontálních hran a transponováním těchto masek získáme vertikální hrany. Člen R v bi-konvolučních maskách představuje rotor

$$R = e^{-\frac{\pi}{2}l} = e^{-\frac{\pi}{2\sqrt{3}}(e_2 \wedge e_3 - e_1 \wedge e_3 + e_1 \wedge e_2)}. \quad (3.9)$$

Při aplikaci vztahu (3.8) se díky spodním řádkům bi-konvolučních masek objevuje člen $RI\tilde{R}$ a tím dochází k rotaci příslušných pixelů o úhel π kolem osy l ze vztahu (3.9) představující „šedou osu“ RGB krychle. Při bi-konvoluci navíc po násobení příslušných prvků dochází ještě ke sčítání. Po sečtení CGA bodů musíme výsledek ještě normovat (vydělit koeficientem u vektoru e_0), abychom získali opět CGA bod. Tento krok je náhradou za násobení bi-konvolučních masek redukčním koeficientem v případě kvaternionové detekce hran, viz [7, 15]. Vlivem rotací a následným sčítání pixelů je výstupem obraz, ve kterém



(a) Originální obraz



(b) Barevné hrany



(c) Detail barevných hran

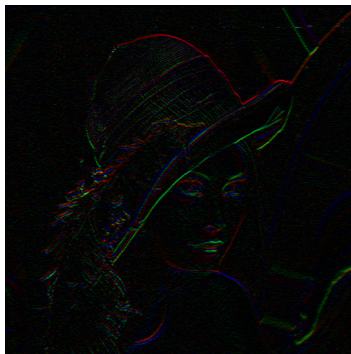
Obrázek 3.1: CGA detekce hran

jsou barevné hrany a v místech, kde se v původním obrazu nevyskytovala hrana, jsou odstíny šedi. Výsledek detekce hran i se vstupním obrazem a detailem barevných hran znázorňuje obrázek 3.1. Nyní můžeme pro názornost jednoduchým způsobem separovat

barevné hrany od zbytku výstupního obrazu. K tomuto účelu budeme pixely výsledného obrazu 3.1b reprezentovat jako vektory $\mathbb{G}_{3,0} \subset \mathbb{G}_{4,1}$ (viz vztah (3.1)), tedy euklidovské vektory, a na těchto vektorech zavedeme ortogonální projekci (operace „rejection“). Mějme vektor $\mathbf{x} \in \mathbb{G}_{3,0}$, jeho ortogonální projekce \mathbf{x}_\perp vzhledem k vektoru $\mathbf{n} \in \mathbb{G}_{3,0}$ je určena vztahem

$$\mathbf{x}_\perp = (\mathbf{x} \wedge \mathbf{n})\mathbf{n}^{-1}, \quad (3.10)$$

kde $\mathbf{n}^{-1} = \frac{\tilde{n}}{\|\mathbf{n}\|^2}$. Výsledkem je část vektoru \mathbf{x} , která je kolmá k vektoru \mathbf{n} . Myšlenka detekce



Obrázek 3.2: Ortogonální projekce barevných hran

hran barevného obrazu pomocí rotorů je založena na tom, že v ideálním případě mají ve výstupním obraze pouze vektory představující hranu nenulovou ortogonální projekci vzhledem k vektoru představujícímu šedou osu RGB prostoru. Aplikací vztahu (3.10) na výstupní obraz, kde \mathbf{x} budou vektory obrazu a $\mathbf{n} = e_1 + e_2 + e_3$ je vektor představující šedou osu, získáváme obraz pouze barevných hran, viz 3.2. Z obrázku je patrné, že po detekci hran nemají nenulovou ortogonální projekci vzhledem k šedé ose pouze hrany. To je způsobeno tím, že v místech, kde se v původním obraze hrana nevyskytuje, nemají všechny pixely v oblasti 3×3 stejné hodnoty barvy, ale mírně se liší. Tím dojde při bi-konvoluci k tomu, že příslušný pixel ve výsledném obraze se nachází pouze v blízkosti šedé osy. Je však patrné, že skutečné hrany mají výrazně vyšší intenzitu než barevné pixely, kde se hrany nevyskytují.

Poznámka 3.2. Doplnkem k operaci „rejection“, viz vztah (3.10), je operace „projection“ vektoru \mathbf{x} , tedy projekce na vektor \mathbf{n} , kterou značíme \mathbf{x}_\parallel a lze ji určit vztahem

$$\mathbf{x}_\parallel = (\mathbf{x} \cdot \mathbf{n})\mathbf{n}^{-1}. \quad (3.11)$$

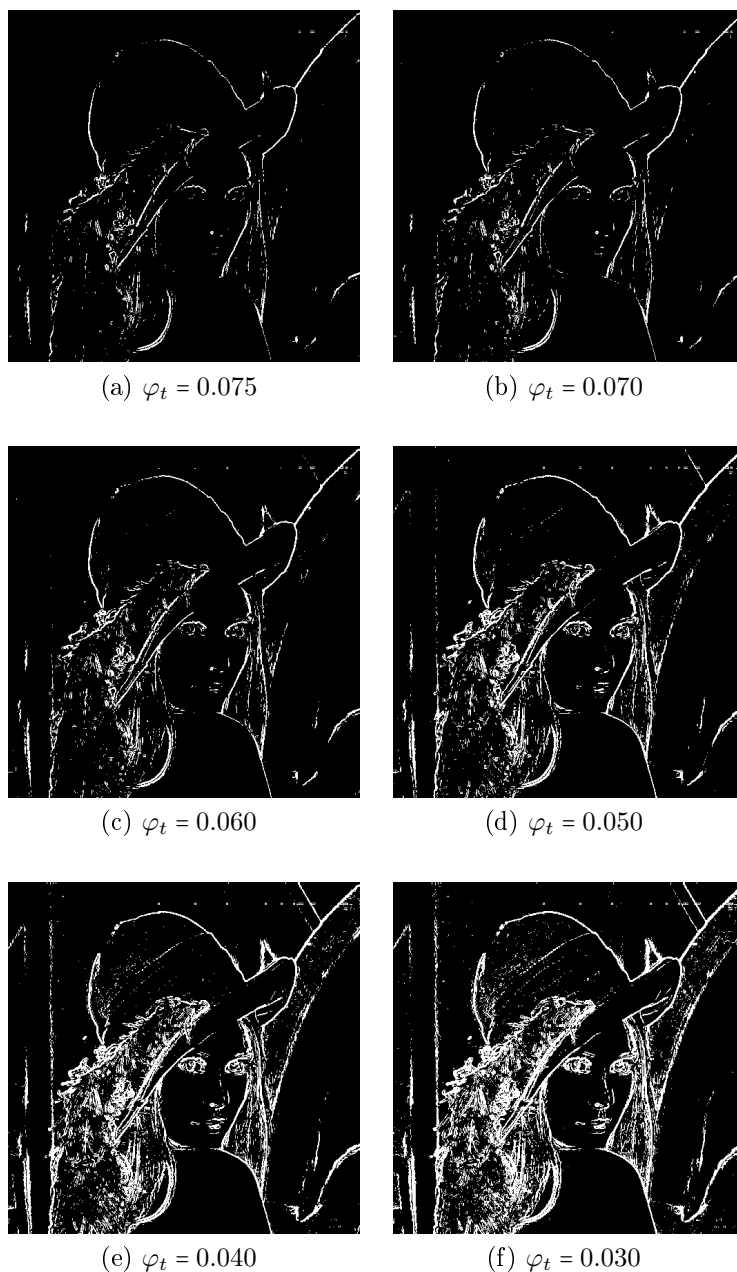
S využitím (1.7) lze snadno ukázat, že platí

$$\mathbf{x} = \mathbf{x}_\parallel + \mathbf{x}_\perp. \quad (3.12)$$

Vztahy (3.10) a (3.11) lze použít na libovolné vektory algebry $\mathbb{G}_{p,q}$ a operace „projection“ a „rejection“ lze obdobně zavést i na libovolných bázevých elementech, viz [13]. Vztah (3.12) však pro vektory $\mathbb{G}_{p,q}$ obecně neplatí.

Výsledným produktem filtrů detekujících hrany obrazu bývá obvykle binární obraz hran, tedy pixely odpovídající hranám mají hodnotu 1 a ostatní pixely mají hodnotu 0. Ukážeme si tedy, jakým způsobem lze získat takový obraz z našeho výstupního obrazu 3.1b. Běžnou

metodou v případě obrazu v odstínech šedi, kde obraz je reprezentován jednou číselnou maticí, je prahování. Zvolí se prahová hodnota a pixelům, které tuto hodnotu překročí, je přiřazena hodnota 1, ostatním hodnota 0. V našem případě lze obdobný postup použít buď na obraz separovaných barevných hran 3.2, kde lze tímto způsobem porovnávat normu jednotlivých vektorů (čím vyšší hodnota, tím vyšší pravděpodobnost výskytu hrany v původním obraze). Druhým způsobem, který umožňuje získání binárního obrazu hran přímo z obrazu po bi-konvoluci 3.1b, a který zde použijeme, je prahování na základě velikosti úhlů vektorů pixelů s vektorem představujícím šedou osu RGB krychle. Mějme



Obrázek 3.3: Binární obraz hran pro různé prahové hodnoty φ_t v radiánech

tedy obraz 3.1b reprezentovaný maticí

$$X_{out} = Re_1 + Ge_2 + Be_3,$$

kde R, G, B jsou číselné matice intenzit příslušných barevných složek. Úhel φ mezi vektorem na pozici $[n, m]$ a vektorem šedé osy $\mathbf{n} = e_1 + e_2 + e_3$ získáme výpočtem vztahu

$$\varphi(n, m) = \arccos \frac{X_{out}(n, m) \cdot \mathbf{n}}{\|X_{out}(n, m)\| \|\mathbf{n}\|}.$$

Konečný binární obraz I_{bin} tedy získáme zvolením prahové hodnoty úhlu φ_t a následným přiřazením jedničky nebo nuly příslušnému pixelu obrazu:

$$I_{bin}(n, m) = \begin{cases} 1, & \varphi(n, m) > \varphi_t, \\ 0, & \text{jinak.} \end{cases}$$

Pro výsledný binární obraz pro různé prahové hodnoty φ_t viz 3.3. Na závěr nutno poznamenat, že filtr Prewittově modifikovaný pro detekci hran barevného obrazu je velmi jednoduchým typem v porovnání s jinými „rotačními“ filtry (Sobel, Kirsch) fungujícími na stejném principu. To znamená, že detekce hran při použití tohoto filtru je méně kvalitní, použití v této práci je zejména z důvodu intuitivnější představy principu bi-konvoluce CGA bodů.

3.2 Segmentace barev

Segmentace barevného obrazu je obecně proces, při kterém je obraz rozdělen do jednotlivých disjunktních regionů, kde pixely daného regionu mají podobné barevné vlastnosti a tvoří tedy jeden spojitý objekt obrazu. Formální definici segmentace obrazu najdeme například v monografii [8]. Tento princip se využívá například pro detekci konkrétního objektu na obrazu, kdy segmentace končí po úspěšném izolování daného objektu a není potřeba pokračovat v segmentaci dále (například automatizované hledání závad na elektronických součástkách apod.). Jednou z metod segmentace obrazu je již zmíněná detekce hran, viz kapitola 3.1. Segmentací barev v této kapitole rozumíme ovšem jiný přístup založený na metodě „zvětšování okolí“. Tato metoda spadá na rozdíl od detekce hran do prostorové analýzy obrazu a podstatou je připojování okolních pixelů již sestaveného regionu (začínáme jedním zvoleným pixelem) na základě určitých kritérií. Na rozdíl od běžného rozdělování obrazu na jednotlivé složky (v případě RGB obrazu jsou to složky R,G,B) a následné segmentaci na jednotlivých složkách zvláště umožňuje CGA pracovat s celým barevným obrazem zároveň.

V této kapitole ukážeme, jakým způsobem je možné izolovat v obrazu konkrétní barvu pomocí CGA a následně tuto myšlenku použijeme pro návrh jednoduchého algoritmu na detekci objektu v obraze. Základní literaturou pro tuto kapitolu, kde lze najít více informací o zpracování digitálního obrazu, je kniha [8]. Článek [2] pojednává detailněji o různých metodách segmentace barevného obrazu a článek [3] se zaměřuje na segmentaci obrazu z hlediska různých barevných prostorů. Segmentací obrazu pomocí CGA se zabývá článek [11].

3.2.1 Barevný prostor CIELAB

Při segmentaci barev budeme pracovat v barevném prostoru CIE $L^*a^*b^*$ (nebo zkráceně CIELAB). Tento barevný prostor je odvozen od základního prostoru XYZ (je založen

na trichromatické teorii, tedy každou barvu získáme jako lineární kombinaci hodnot tří základních barev), který získáme lineární transformací prostoru RGB danou maticí

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Tento prostor se někdy nazývá imaginární, protože základní barvy XYZ nelze fyzikálně realizovat, ale jakoukoliv barvu lze vyjádřit jako součet kladných hodnot těchto základních barev. Odtud lze přejít do prostoru CIELAB nelineární transformací

$$\begin{aligned} L^* &= 116 \sqrt[3]{\frac{Y}{Y_0}} - 16, \\ a^* &= 500 \left(\sqrt[3]{\frac{X}{X_0}} - \sqrt[3]{\frac{Y}{Y_0}} \right), \\ b^* &= 200 \left(\sqrt[3]{\frac{Y}{Y_0}} - \sqrt[3]{\frac{Z}{Z_0}} \right), \end{aligned}$$

kde $X/X_0, Y/Y_0, Z/Z_0 > 0.008856$. X_0, Y_0, Z_0 jsou standardizované hodnoty bílé barvy, viz tabulka 3.1. Dále v aplikaci CGA budeme využívat standard $D50$, který simuluje denní

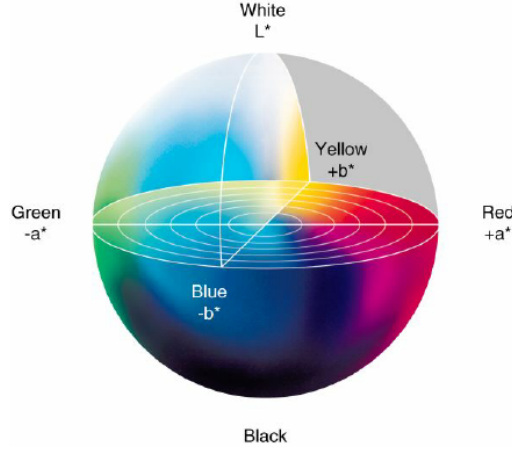
Zdroj světla	X_0	Y_0	Z_0
$D50$	96.680	100	82.519
A	111.144	100	35.20
$TL84$	103.863	100	65.607
C	97.285	100	116.145
$F2(CWF)$	103.279	100	69.027

Tabulka 3.1: Standardizované hodnoty bílé barvy

světlo. Trojice souřadnic reprezentuje světlost barvy L^* ($L^* = 0$ představuje černou barvu a $L^* = 100$ je bílá barva), a^* , s rozsahem obvykle od -128 do 127 , je pozice mezi červenou a zelenou barvou (záporné hodnoty a^* znázorňují zelenou barvu a kladné hodnoty červenou) a souřadnice b^* , se stejným rozsahem jako a^* , je pozice mezi žlutou a modrou barvou (záporné hodnoty přísluší modré a kladné hodnoty žluté barvě), viz obrázek 3.4. Hlavní motivací využití tohoto prostoru pro segmentaci barev je euklidovská metrika na prostoru odpovídající lidskému vnímání rozdílu jednotlivých barev. To znamená, že pokud máme dvě dvojice barev se stejnými vzdálenostmi, potom vnímaná vizuální odchylka mezi těmito dvojicemi bude stejná. Tuto vlastnost prostor RGB nemá, může se tedy stát, že barvy, které vnímáme jako velmi „rozdílné“, mají relativně malou vzdálenost v RGB prostoru a naopak barvy, které se nám jeví jako „podobné“, mohou být v prostoru vzdálenější. Barevnou diferencí dvou barev v prostoru CIE $L^*a^*b^*$ lze tedy vyjádřit jako

$$\Delta E = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}.$$

Tyto vlastnosti jsou klíčově právě pro segmentaci barev, kde chceme rozlišit různě vnímané barvy. Navíc nám tento prostor, jak se přesvědčíme dále, umožní pro tyto účely využít potenciál CGA algebry.



Obrázek 3.4: Model barevného prostoru CIE $L^*a^*b^*$. Zdroj [1]

3.2.2 Implementace CGA v prostoru CIELAB

Jak již bylo řečeno, barevný prostor CIE $L^*a^*b^*$ je opatřen euklidovskou metrikou vzhledem k vnímanému rozdílu barev. Tedy máme-li určenou nějakou referenční barvu, potom barvy nepřesahující danou odchylku od referenční barvy tvoří kouli. Chceme-li barevně klasifikovat určitý pixel, podíváme se na polohu pixelu vzhledem ke sféře s poloměrem představujícím danou barevnou odchylku a středem znázorňujícím referenční barvu. Pokud zkoumaný pixel leží vně sféry, potom tento pixel hodnotíme jako nevyhovující a pokud leží uvnitř sféry, potom pixel vyhovuje. S využitím CGA můžeme snadno rozhodnout o poloze bodu vzhledem ke sféře pomocí skalárního součinu. Mějme IPNS reprezentaci sféry $S = P - \frac{1}{2}r^2e_\infty$ ($P = \mathbf{p} + \frac{1}{2}\mathbf{p}^2e_\infty + e_0$ je bod určující střed sféry) a bod $X = \mathbf{x} + \frac{1}{2}\mathbf{x}^2e_\infty + e_0$. Zopakujme výpočet skalárního součinu

$$S \cdot X = P \cdot X - \frac{1}{2}r^2e_\infty \cdot X.$$

S využitím vztahu (2.3) pro skalární součin dvou bodů získáváme

$$\begin{aligned} S \cdot X &= -\frac{1}{2}\|\mathbf{p} - \mathbf{x}\|^2 - \left(\frac{1}{2}r^2e_\infty\right) \cdot \left(\mathbf{x} + \frac{1}{2}\mathbf{x}^2e_\infty + e_0\right) \\ &= -\frac{1}{2}\|\mathbf{p} - \mathbf{x}\|^2 - \left(\frac{1}{2}r^2e_\infty \cdot \mathbf{x} + \frac{1}{4}r^2\mathbf{x}^2e_\infty \cdot e_0 + \frac{1}{2}r^2e_\infty \cdot e_0\right) \\ &= -\frac{1}{2}\|\mathbf{p} - \mathbf{x}\|^2 + \frac{1}{2}r^2. \end{aligned}$$

Jelikož $\|\mathbf{p} - \mathbf{x}\|$ představuje vzdálenost mezi středem sféry S a bodem X , platí pro polohu bodu X vzhledem ke sféře S zřejmě následující tvrzení:

$$S \cdot X \begin{cases} > 0, & \text{bod leží uvnitř sféry,} \\ = 0, & \text{bod leží na sféře,} \\ < 0, & \text{bod leží vně sféry.} \end{cases} \quad (3.13)$$

Výše uvedený postup aplikujeme na $L^*a^*b^*$ obraz (L^*, a^*, b^* jsou číselné matice obrazu). Každý pixel obrazu reprezentujeme jako CGA bod, vstupní obraz je tedy vyjádřen maticí

$$I = L^*e_1 + a^*e_2 + b^*e_3 + \frac{1}{2}(L^*e_1 + a^*e_2 + b^*e_3)^2e_\infty + e_0. \quad (3.14)$$



Obrázek 3.5: Vstupní obraz pro segmentaci barev

Referenční barvu $[L_0^*, a_0^*, b_0^*]$ (v $L^*a^*b^*$ souřadnicích) pro segmentaci vyjádříme pomocí sféry

$$S_0 = P_0 - \frac{1}{2}r_0^2e_\infty, \quad (3.15)$$

kde r_0 je hodnota barevné odchylky od referenční barvy, která je již vyhodnocená jako nepřipustná a P_0 je střed sféry

$$P_0 = L_0^*e_1 + a_0^*e_2 + b_0^*e_3 + \frac{1}{2}(L_0^*e_1 + a_0^*e_2 + b_0^*e_3)^2e_\infty + e_0.$$

Výsledný obrázek získáme aplikací (3.13), kde počítáme hodnotu skalárního součinu každého prvku matice (3.14) a sféry (3.15). Pokud pixel na pozici $[m, n]$ leží uvnitř sféry, tedy $S_0 \cdot I(m, n) > 0$, je vyhodnocen jako vyhovující a je ponechán ve výstupním obrazu. Pokud pixel nevyhovuje kritériu, tedy leží vně nebo na sféře, je z obrazu odstraněn (je mu přiřazena většinou černá nebo bílá barva). Obrázek 3.6 demonstruje tento postup aplikovaný na vstupní obraz 3.5 pro různé referenční barvy a hodnoty přípustných barevných odchylek. V dalším využijeme myšlenku segmentace barev uvedenou v této kapitole pro návrh jednoduchého algoritmu, jehož úkolem je detekovat zvolený objekt v obrazu na základě barevné odlišnosti. Pokusíme se také porovnat výsledky při práci s RGB a $L^*a^*b^*$ prostorem.

3.2.3 Algoritmus pro detekci objektu

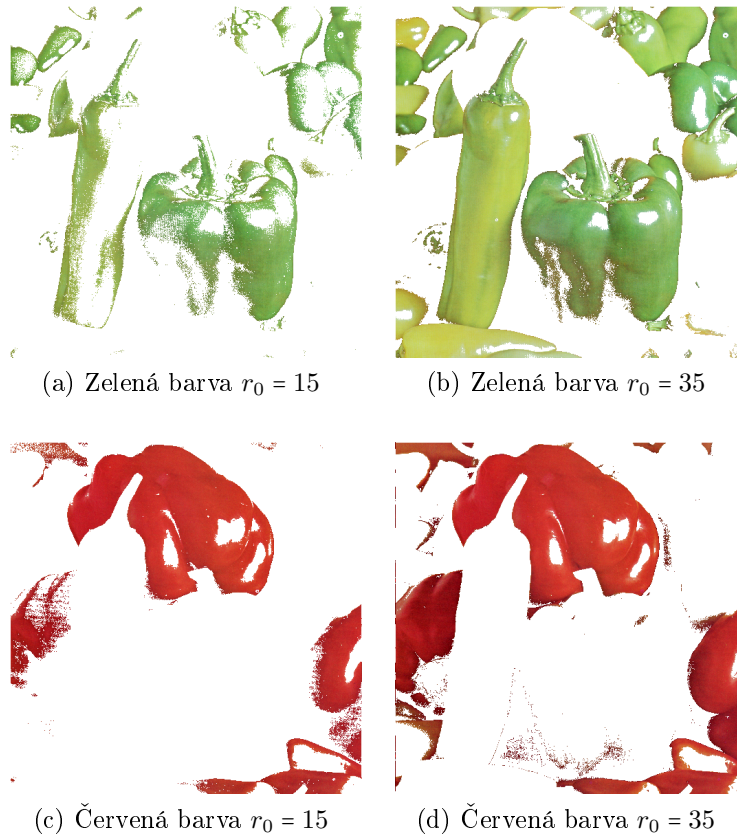
V této části práce popíšeme algoritmus, jehož cílem je detekce objektu v obraze. V průběhu textu budou pro doplnění přidávány úseky zdrojového kódu ze softwaru MATLAB. Celý zdrojový kód i se všemi použitými pomocnými funkcemi nalezneme v příloze této práce. Pro práci s CGA v toolboxu Clifford Multivector Toolbox načteme vhodný typ geometrické algebry, tedy $\mathbb{G}_{4,1}$, příkazem

```
clifford_signature(4,1,0);
```

Vstupní RGB obraz převedeme do prostoru $CIEL^*a^*b^*$ pomocí příkazu

```
Ib = rgb2lab(I, 'WhitePoint', 'd50');
```

kde I je vstupní originální obraz 3.7, pro transformaci mezi barevnými prostory volíme standardizovanou hodnotu bílé barvy $D50$, viz tabulka 3.1. $CIEL^*a^*b^*$ obraz Ib reprezentujeme jako CGA body:



Obrázek 3.6: Segmentovaný obraz. RGB souřadnice pro jednotlivé referenční barvy: zelená $[0.47, 0.62, 0.27]$, červená $[0.77, 0.21, 0.18]$.

```
v = Ib(:,:,1)*e1 + Ib(:,:,2)*e2 + Ib(:,:,3)*e3;
[r,c,d] = size(Ib);
Icga = v + 0.5*v.^2*einf + ones(r,c)*eo;
```

Jelikož toolbox umí definovat algebru jen signaturou, musíme tedy pracovat s ortogonální bází $\{1, e_1, e_2, e_3, e_4, e_5\}$ a pro zjednodušení použijeme pomocné funkce `einf` a `eo`, viz příloha B.

Samotný algoritmus je založen na metodě postupně se rozrůstajícího okolí (anglicky region growing) kolem jednoho iničiálního pixelu na základě určitých kritérií. Nejprve zvolíme v obrazu iničiální pixel, jehož okolní pixely budeme zkoumat. Tento pixel reprezentovaný opět jako CGA bod označíme P . Dále zkoumáme vlastnosti, v našem případě barvu, okolních pixelů. Ty z okolních pixelů, které nepřekračují zvolenou barevnou odchylku, jsou připojeny k pixelu P , tím dojde ke zvětšení oblasti vhodných pixelů a pokračujeme ve vyhodnocování nového okolí této oblasti. Dále probíhá algoritmus iteračně, kde v každém kroku jsou připojeny okolní pixely splňující tři podmínky, které znázorníme pomocí terminologie CGA. Nechť Q je analyzovaný pixel v k -tém iteračním kroku a S_1, S_2, S_3 jsou sféry charakterizující jednotlivé podmínky. S_1 je sféra se středem v bodě P a poloměrem r_1 . Vnitřek této sféry tedy tvoří barvy, které jsou vzdáleny maximálně o r_1 od barvy pixelu P . Sféra S_2 s poloměrem r_2 má střed v bodě P_2 , který představuje průměr barev množiny pixelů vyhodnocených jako vyhovující v předchozí $(k-1)$ iteraci. Jelikož nelze sčítat CGA



Obrázek 3.7: Vstupní obraz pro detekci objektu

body a získat opět CGA bod, průměr n barev A_i vypočítáme podle vztahu

$$\bar{A} = \left(\frac{1}{n} \sum_{i=1}^n \mathbf{a}_i \right) + \left(\frac{1}{n} \sum_{i=1}^n \mathbf{a}_i \right)^2 e_\infty + e_0,$$

kde $\mathbf{a}_i = L_i^* e_1 + a_i^* e_2 + b_i^* e_3$. Konečně sféra S_3 má střed P_3 představující průměr barev všech pixelů z dosud vyhovujícího regionu a poloměr r_3 . Pixel Q tedy vyhodnotíme jako vyhovující a připojíme k regionu pokud platí současně:

1. $S_1 \cdot Q > 0$,
2. $S_2 \cdot Q > 0$,
3. $S_3 \cdot Q > 0$.

Podmínka 1 je ověřena pro všechny pixely obrazu v jednom kroku, ve zdrojovém kódu je realizována příkazem

```
A1 = scalar_product(Icga,S1) > 0;
```

Podmínky 2 a 3

```
A2 = scalar_product(double(C).*Icga , S2) > 0;
```

```
A3 = scalar_product(double(C).*Icga , S3) > 0;
```

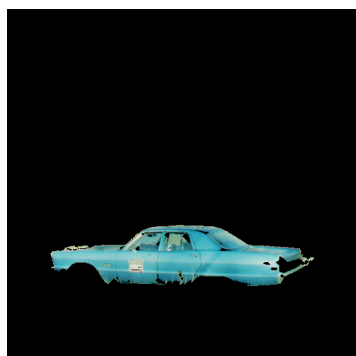
jsou naproti tomu vyhodnocovány v každé iteraci, jelikož v každé iteraci pracujeme s novými sférami S_2 a S_3 (C je matice pixelů označených k analýze). Celý proces končí ve chvíli, kdy v nové iteraci není žádný pixel vyhodnocen jako vyhovující, tedy region vhodných pixelů už nadále neroste.

Celý postup aplikujeme na obrázek 3.7 a pokusíme se detekovat pouze modré auto. Volbu iničiálního pixelu pro algoritmus znázorňuje červená šipka na obrázku 3.8. Prahové hodnoty pro jednotlivé podmínky vyhodnocování pixelů (poloměry sfér) jsou voleny experimentálně na začátku programu. Poznamenejme ještě, že v první iteraci jsou všechny tři sféry totožné. Algoritmus byl testován pro různé prahové hodnoty r_1, r_2, r_3 , vizuálně nejlepší výsledky jsou zobrazeny na obrázku 3.9. Pro porovnání ještě uvádíme výstupní obraz při použití algoritmu na RGB obraz, viz 3.10. Při práci s RGB obrazem se nám nepodařilo získat žádné uspokojivé výsledky, což potvrzuje předchozí domněnku, že RGB prostor není vhodný pro segmentaci barevného obrazu.

Pro shrnutí znázorníme celý algoritmus zjednodušeně v několika klíčových krocích:



Obrázek 3.8: Iniciální pixel pro algoritmus vyznačen červenou šipkou

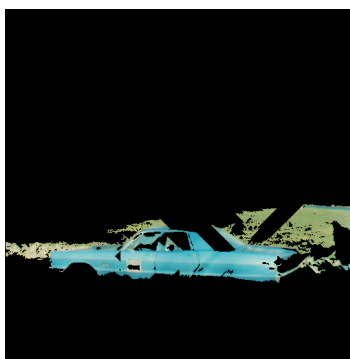


(a) (40, 30, 40)



(b) (40, 50, 40)

Obrázek 3.9: Výstupní obraz auta pro prahové hodnoty (r_1, r_2, r_3) , pro které jsme dosáhli nejlepších výsledků.



Obrázek 3.10: Výstupní obraz po použití algoritmu na RGB obraz pro prahové hodnoty $r_1 = r_2 = r_3 = 0.45$.

1. Vyjádření vstupního obrazu jako matici CGA bodů I_{CGA} .
2. Volba iniciálního pixelu.
3. Sestrojení sfér S_1, S_2, S_3 znázorňujících podmínky 1, 2 a 3.
4. Ověření podmínky 1 pro všechny pixely obrazu $S_1 \cdot I_{CGA} > 0$.
5. Ověření podmínek 2 a 3 pro určené pixely.

6. Připojení pixelů vyhovujících všem třem podmínkám současně.
7. Pokud byly v předchozím kroku přidány nějaké nové pixely sestrojení nových sfér S_2 a S_3 a návrat ke kroku 5. Jinak konec algoritmu.

Je nutné zmínit, že celý algoritmus byl navržen především pro experimentální účely se snahou zkoumat a ukázat možný směr vývoje uplatnění CGA v segmentaci obrazu. Pro skutečnou aplikaci je nutné zabývat se danou problematikou dále a navrhnout sofistikovanější postupy, pro účely prezentace výhod CGA je však tento přístup postačující. Předností je především jednoduchost aplikace CGA, všechny operace s obrazem jsou realizovány intuitivně pomocí maticových operací. Nevýhodou naší implementace je rychlost algoritmu, jeden obraz se zpracovává řádově v sekundách, což je způsobeno stávající implementací geometrické algebry v MATLABU skrze `Clifford Multivector Toolbox`. Tento toolbox je v současnosti vyvíjen, jeho rychlost se zvyšuje a experimenty uvedené v této práci byly s autory toolboxu diskutovány a přispěly k jeho vývoji. Zdrojový kód celého algoritmu najdeme v příloze A a další výstupní obrázky v příloze C.

Kapitola 4

Aplikace CGA pro popis okolí v prostoru barev

V této kapitole si ukážeme, že CGA nám umožňuje efektivně aproximovat množinu bodů v prostoru pomocí geometrických objektů ve smyslu metody nejmenších čtverců. Naším hlavním zájmem bude aproximace bodů sférou. Budeme čerpat především z článku [6], který se tímto problémem zabývá. Naším hlavním záměrem bude implementace této aproximace v MATLABU a dále se pokusíme aplikovat tuto myšlenku na reálná data v podobě barevného obrazu.

Metoda nejmenších čtverců je dobře známá matematicko-statistická metoda umožňující prokládání dat v podobě bodů nějakou křivkou (obyčejně přímkou, parabolou, atd.). Metoda spočívá v minimalizaci součtu druhých mocnin vzdáleností jednotlivých bodů od proložené křivky. Ukážeme si, že tuto myšlenku můžeme využít pro prokládání bodů v prostoru sférou tak, že minimalizujeme součet vzdáleností jednotlivých bodů od této sféry (podobně lze body prokládat rovinou představující sféru s nekonečným poloměrem, kružnicí nebo přímkou představující kružnici s nekonečným poloměrem). Vstupem našeho problému tedy bude množina bodů v prostoru a výstupem sféra, která tyto body nejlépe aproximuje. Metrika pro měření vzdáleností mezi CGA vektory je definována skalárním součinem.

4.1 Hledání aproximace

Na rozdíl od kapitoly 3, kde jsme pro práci s CGA využívali `Clifford Multivector Toolbox`, se nyní omezíme z důvodu absence potřebných nástrojů v tomto toolboxu na práci bez jeho použití a ukážeme si, jakým způsobem lze pracovat s CGA jen pomocí základního balíčku pro lineární algebru. Jelikož budeme při hledání vhodné sféry pro

\cdot	e_1	e_2	e_3	e_0	e_∞
e_1	1	0	0	0	0
e_2	0	1	0	0	0
e_3	0	0	1	0	0
e_0	0	0	0	0	-1
e_∞	0	0	0	-1	0

Tabulka 4.1: Skalární součin mezi bázovými elementy CGA definující metriku.

aproximaci bodů pracovat s body a sférami (případně je možné uvažovat rovinu jako degenerovanou sféru), tedy v IPNS reprezentaci s CGA vektory (viz 2.2), můžeme vzdálenosti určovat pomocí skalárního součinu. Skalární součin nám tedy, jak již bylo řečeno, definuje metriku na CGA vektorech danou tabulkou 4.1. Připomeňme, že pro skalární součin dvou bodů P, Q , jejichž pozice je v euklidovském prostoru dána vektory \mathbf{p}, \mathbf{q} , platí

$$P \cdot Q = -\frac{1}{2} \|\mathbf{p} - \mathbf{q}\|^2.$$

Skalární součin CGA vektorů lze jednoduše implementovat pouze s využitím balíčku pro lineární algebru následovně. Mějme matici skalárního součinu M daného tabulkou 4.1, tedy

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix},$$

kteřá reprezentuje skalární součin mezi bázovými vektory. CGA bod $P = \mathbf{p} + \mathbf{e}_0 + \frac{1}{2} \|\mathbf{p}\|^2 \mathbf{e}_\infty$ vyjádříme jako vektor v \mathbb{R}^5

$$P = \begin{pmatrix} \mathbf{p} \\ 1 \\ \frac{1}{2} \|\mathbf{p}\|^2 \end{pmatrix},$$

kde $\mathbf{p} = (p_1, p_2, p_3)^T$. Skalární součin bodů P, Q potom vypočítáme následovně:

$$\begin{aligned} P \cdot Q &= (P)^T (M) (Q) \\ &= (\mathbf{p}^T \quad 1 \quad \frac{1}{2} \|\mathbf{p}\|^2) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ 1 \\ \frac{1}{2} \|\mathbf{q}\|^2 \end{pmatrix} \\ &= (\mathbf{p}^T \quad -\frac{1}{2} \|\mathbf{p}\|^2 \quad -1) \begin{pmatrix} \mathbf{q} \\ 1 \\ \frac{1}{2} \|\mathbf{q}\|^2 \end{pmatrix} \\ &= \mathbf{p}^T \mathbf{q} - \frac{1}{2} \|\mathbf{p}\|^2 - \frac{1}{2} \|\mathbf{q}\|^2 \\ &= -\frac{1}{2} \|\mathbf{p} - \mathbf{q}\|^2. \end{aligned}$$

Stejným způsobem lze pracovat i se sférami či rovinami. Než přejdeme k samotné aproximaci bodů sférou, uvedeme si některé základní poznatky z diferenciálního počtu multivektorů, které budeme k aproximaci potřebovat (podrobný popis derivace multivektorů nalezneme v [4]). Nejprve připomeňme, že inverze vektoru \mathbf{x} s vlastností $\mathbf{x}^2 \neq 0$ je

$$\mathbf{x}^{-1} = \frac{\mathbf{x}}{\mathbf{x}^2} = \frac{\mathbf{x}}{\mathbf{x} \cdot \mathbf{x}}.$$

Operátor derivace vektoru v \mathbb{R}^3 je běžně značen symbolem ∇ a definován jako

$$\nabla := e_1 \frac{\partial}{\partial x} + e_2 \frac{\partial}{\partial y} + e_3 \frac{\partial}{\partial z}.$$

Tento operátor můžeme v jistém smyslu rozšířit na prostor libovolné dimenze. K tomu však potřebujeme ještě pojem *reciproká* báze.

Definice 4.1. Necht' $\{\mathbf{a}_i\} \subset \mathbb{R}^{p,q}$ je nějaká (ne nutně ortogonální) báze vektorů v $\mathbb{G}_{p,q}$. Množinu $\{\mathbf{a}_i^*\} \subset \mathbb{R}^{p,q}$ nazveme *reciproká báze*, jestliže tvoří bázi vektorů v $\mathbb{G}_{p,q}$ a zároveň platí $\mathbf{a}_i^* \cdot \mathbf{a}_j = \delta_{i,j}$ pro $j = 1, \dots, p+q$, kde $\delta_{i,j}$ je Kroneckerovo delta.

Poznamenejme, že pro ortonormální bázi $\{e_i\} \subset \mathbb{R}^{p,q}$ máme reciprokou bázi $\{e_i^\dagger\}$, kde \dagger značí konjugaci, viz poznámka 3.1.

Nyní můžeme zavést operátor parciální derivace vzhledem k vektoru $\mathbf{x} \in \mathbb{R}^{p,q}$, kde $\mathbf{x} = \sum_i x_i e_i$, který budeme značit $\partial_{\mathbf{x}}$ a definujeme ho vztahem

$$\partial_{\mathbf{x}} := \sum_i e_i^* \partial_{x_i},$$

kde e_i^* je reciproký vektor e_i a ∂_{x_i} je parciální derivace vzhledem k x_i . Nyní konečně můžeme zavést derivaci funkce, která vektoru přiřazuje skalární hodnotu. Pro zdůraznění, že pracujeme v geometrické algebře, označíme množinu všech vektorů geometrické algebry $\mathbb{G}_{p,q}$ jako $\mathbb{G}_{p,q}^1$.

Definice 4.2. Mějme funkci $\mathcal{F} : \mathbb{G}_{p,q}^1 \rightarrow \mathbb{R}$, potom derivaci funkce vzhledem k vektoru \mathbf{x} značíme $\partial_{\mathbf{x}} \mathcal{F}$ a definujeme:

$$\partial_{\mathbf{x}} \mathcal{F}(\mathbf{x}) = \sum_i e_i^* \partial_{x_i} \mathcal{F}(\mathbf{x}) = \sum_i e_i^* \mathcal{F}_i(\mathbf{x}), \quad (4.1)$$

kde $\mathcal{F}_i(\mathbf{x}) := \partial_{x_i} \mathcal{F}(\mathbf{x})$ je parciální derivace funkce \mathcal{F} podle x_i v \mathbf{x} .

Poznámka 4.1. Definici 4.2 lze snadno rozšířit na derivaci funkce $\mathcal{F} : \mathbb{G}_{p,q} \rightarrow \mathbb{R}$ vzhledem k multivektoru $X := \sum_i x_i E_i$, kde $\{E_i\} = \mathbb{G}_{p,q}$ je ortonormální báze. Multivektorový operátor derivace je tvaru

$$\partial_X := \sum_i E_i^* \partial_{x_i},$$

kde pro reciproké bázové elementy platí $E_i^* \star E_j = \delta_{i,j}$ pro $j = 1, \dots, 2^{p+q}$. Derivaci funkce \mathcal{F} potom získáme podobně jako ve vztahu 4.1:

$$\partial_{\mathbf{x}} \mathcal{F}(X) = \sum_i E_i^* \partial_{x_i} \mathcal{F}(X) = \sum_i E_i^* \mathcal{F}_i(X).$$

Derivace obecné funkce, jejíž definičním oborem i oborem hodnot je $\mathbb{G}_{p,q}$, můžeme najít v [13, 4]. Jelikož budeme dále potřebovat pouze derivaci danou vztahem 4.1, nebudeme se dále teorií diferenciálního počtu multivektorů zabývat.

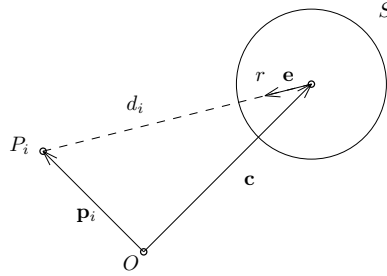
Bez důkazů uvedeme výčet některých vlastností multivektorových derivací, které využijeme dále při hledání optimální sféry. Pro přehlednost vynecháme při zápisu operátoru derivace vektor, podle kterého derivujeme. Mějme funkce $\mathcal{F}(\mathbf{x}), \mathcal{G}(\mathbf{x})$, potom platí známé vlastnosti z klasického diferenciálního počtu:

- $\partial(\mathcal{F} + \mathcal{G}) = \partial\mathcal{F} + \partial\mathcal{G}$,
- $\partial(\mathcal{F}\mathcal{G}) = (\partial\mathcal{F})\mathcal{G} + \mathcal{F}(\partial\mathcal{G})$.

Nyní jsme připraveni přejít k samotné formulaci našeho problému, tedy nalezení optimální sféry, která aproximuje zadané body v prostoru. Vektor \mathbf{p}_i představující pozici bodu P_i můžeme charakterizovat pomocí relativní polohy vzhledem ke sféře S se středem \mathbf{c} a poloměrem r vztahem

$$\mathbf{p}_i = \mathbf{c} + (r + d_i)\mathbf{e}, \quad (4.2)$$

kde \mathbf{e} je vhodně zvolený jednotkový vektor a d_i je vzdálenost mezi bodem P_i a sférou S (viz obrázek 4.1).



Obrázek 4.1: Zobrazení relativní polohy bodu P_i vzhledem ke sféře S . Pro jednoduchost je situace zobrazena v rovině a sféra je zobrazena jako kružnice.

Úpravou vztahu 4.2 vyjádříme vzdálenost d_i :

$$\begin{aligned} \|\mathbf{p}_i - \mathbf{c}\| &= \|(r + d_i)\mathbf{e}\| \\ \sqrt{(\mathbf{p}_i - \mathbf{c}) \cdot (\mathbf{p}_i - \mathbf{c})} &= r + d_i, \end{aligned}$$

platí tedy

$$d_i = \sqrt{(\mathbf{p}_i - \mathbf{c}) \cdot (\mathbf{p}_i - \mathbf{c})} - r. \quad (4.3)$$

Optimální sféru aproximující množinu bodů $\{P_i\}$ v prostoru hledáme takovou, aby součet druhých mocnin vzdáleností bodů od sféry $\sum_i d_i^2$ byl minimální. Snaha nalézt vhodné řešení minimalizací pro vzdálenosti ve tvaru 4.3 vede typicky na iterativní optimalizační metody. Díky využití CGA můžeme vzdálenosti d_i však aproximovat pomocí jiného algebraického výrazu, díky čemuž bude možné vhodné řešení nalézt efektivněji. Bod P_i a sféru S vyjádříme pomocí CGA, tedy $P_i = \mathbf{p}_i + \frac{1}{2}\mathbf{p}_i^2 e_\infty + e_0$ a $S = C - \frac{1}{2}r^2 e_\infty$, kde C je CGA bod představující střed sféry. Přímým výpočtem lze snadno ukázat, že pro sféru S platí

$S^2 = r^2$. Vypočítáme, čemu se rovná vztah $(P_i \cdot S)^2/S^2$:

$$\begin{aligned}
\frac{(P_i \cdot S)^2}{S^2} &= \frac{1}{4} (r^2 - \|\mathbf{p}_i - \mathbf{c}\|^2)^2 / r^2 \\
&= \frac{1}{4} (r^2 - (d_i + r)^2)^2 / r^2 \\
&= \frac{1}{4} (r^2 - d_i^2 - 2d_i r - r^2)^2 / r^2 \\
&= \frac{1}{4} (-d_i(d_i + 2r))^2 / r^2 \\
&= d_i^2 \left(\frac{d_i}{2r} + 1 \right)^2 \\
&= d_i^2 \left(1 + \frac{d_i}{r} + \left(\frac{d_i}{2r} \right)^2 \right).
\end{aligned}$$

Získáváme tedy aproximaci druhé mocniny vzdálenosti mezi bodem P_i a sférou S

$$\frac{(P_i \cdot S)^2}{S^2} \approx d_i^2, \quad (4.4)$$

s relativní chybou $(d_i/r) + \frac{1}{4}(d_i/r)^2$. Hlavní předností tohoto vyjádření je, že je možné jej oproti vztahu 4.3 snadno derivovat. Problém nalezení optimální sféry x , která aproximuje zadané body P_i lze nyní formulovat jako optimalizační úlohu:

$$\text{Najdi } x \text{ minimalizující: } \mathcal{L}(x) = \frac{1}{N} \sum_i (P_i \cdot x)^2 / x^2, \quad (4.5)$$

kde x je hledaná sféra a N je počet zadaných bodů. Pro nalezení optima je nutné, aby derivace $\partial_x \mathcal{L}(x)$ byla rovna nule. Dostáváme tedy:

$$\begin{aligned}
0 &= \partial_x \mathcal{L}(x) \\
&= \partial_x \left(\frac{1}{N} \sum_i (P_i \cdot x)^2 x^{-2} \right) \\
&= 2 \frac{1}{N} \sum_i ((P_i \cdot x) P_i x^{-2} - (P_i \cdot x)^2 x^{-3}) \\
&= 2 \frac{1}{N} \sum_i (P_i \cdot x) (P_i x^{-2} - (P_i \cdot x) x^{-3}) \\
&= 2 \frac{1}{N} \sum_i (P_i \cdot x) (P_i x - P_i \cdot x) x^{-3} \\
&= 2 \frac{1}{N} \sum_i (P_i \cdot x) (P_i \wedge x) x^{-3}.
\end{aligned}$$

Jelikož vektor x představuje sféru a tedy $2x^{-3}$ má inverzi $\frac{1}{2}x^3$, vynásobíme rovnici tímto členem a získáme

$$\frac{1}{N} \sum_i (P_i \cdot x) (P_i \wedge x) = 0. \quad (4.6)$$

Protože pro multivektory $A, B \in \mathbb{G}_{p,q}$ a skalár $\alpha \in \mathbb{R}$ platí

$$\alpha(A \wedge B) = \alpha A \wedge B,$$

můžeme ještě rovnici 4.6 upravit na konečný tvar

$$\left(\frac{1}{N} \sum_i P_i(P_i \cdot x) \right) \wedge x = 0. \quad (4.7)$$

Tuto podmínku lze interpretovat i tak, že vektor v v závorce musí být roven k násobku vektoru x , jelikož pro vektory $\mathbf{u}, \mathbf{v} \in \mathbb{G}_{p,q}$ platí:

$$\mathbf{u} \wedge \mathbf{v} = 0 \Leftrightarrow \mathbf{u} = \alpha \mathbf{v}, \quad \alpha \in \mathbb{R}. \quad (4.8)$$

Podmínku 4.7 tedy lze ekvivalentně vyjádřit ve tvaru

$$\frac{1}{N} \sum_i P_i(P_i \cdot x) = \lambda x, \quad (4.9)$$

kde $\lambda \in \mathbb{R}$. Vyjádříme-li tuto rovnici pomocí reálných matic podle postupu uvedeného na začátku této kapitoly, obdržíme podmínku extrému funkce \mathcal{L} z naší minimalizační úlohy 4.5 ve tvaru

$$\frac{1}{N} \sum_i (P_i)(P_i)^T(M)(x) = \lambda(x),$$

odkud lze ihned vidět, že vektor λ představuje vlastní číslo a (x) odpovídající vlastní vektor matice

$$X = \frac{1}{N} \sum_i (P_i)(P_i)^T(M).$$

Tedy problém nalezení x minimalizující funkci \mathcal{L} se stává problémem hledání vlastních čísel a vlastních vektorů matice X . Vlastní vektor (x_*) , který minimalizuje druhou mocninu vzdálenosti zadaných bodů od hledané sféry, odpovídá nejmenšímu nezápornému vlastnímu číslu λ_* matice X (důkaz nalezneme v [6]). Z příslušného vlastního vektoru už lze jednoduše získat parametry hledané sféry. Dále si ukážeme implementaci tohoto postupu pro hledání sféry aproximující zadané body v Matlabu včetně obdržených výsledků.

4.2 Implementace v MATLABU

Jak již bylo řečeno, pro implementaci hledání optimální sféry aproximující body v prostoru pomocí CGA budeme využívat pouze základní balíček pro lineární algebru. Výhodou toho, že pracujeme pouze s reálnými maticemi, je rychlost výpočtů. Celý postup popíšeme v několika následujících krocích:

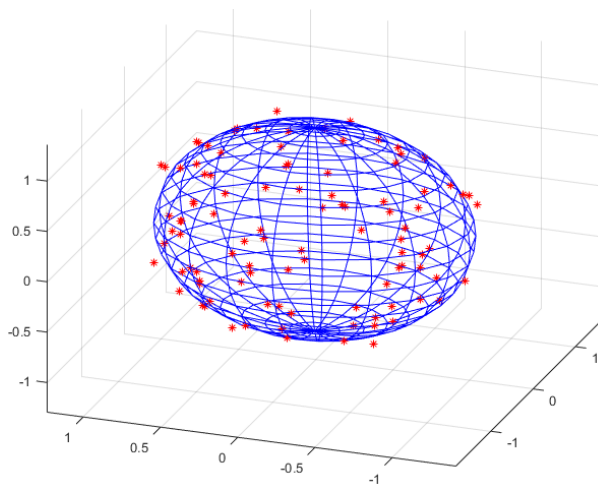
1. CGA bod $P = \mathbf{p} + e_0 + \frac{1}{2}\mathbf{p}^2 e_\infty$ vyjádříme jako sloupcový vektor $(P) = (\mathbf{p}^T, 1, \frac{1}{2}\|\mathbf{p}\|^2)^T$, kde $\mathbf{p} = (p_1, p_2, p_3)^T$. Matice X má tvar:

$$\begin{aligned} X &= \frac{1}{N} \sum_i (P_i)(P_i)^T(M) \\ &= \frac{1}{N} \sum_i \begin{pmatrix} \mathbf{p}_i \\ 1 \\ \frac{1}{2}\|\mathbf{p}_i\|^2 \end{pmatrix} (\mathbf{p}_i^T \quad 1 \quad \frac{1}{2}\|\mathbf{p}_i\|^2) \begin{pmatrix} (1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & 0 & -1 \\ \mathbf{0}^T & -1 & 0 \end{pmatrix} \\ &= \frac{1}{N} \sum_i \begin{pmatrix} \mathbf{p}_i \mathbf{p}_i^T & -\frac{1}{2}\|\mathbf{p}_i\|^2 \mathbf{p}_i & -\mathbf{p}_i \\ \mathbf{p}_i^T & -\frac{1}{2}\|\mathbf{p}_i\|^2 & -1 \\ \frac{1}{2}\|\mathbf{p}_i\|^2 \mathbf{p}_i^T & -\frac{1}{4}\|\mathbf{p}_i\|^4 & -\frac{1}{2}\|\mathbf{p}_i\|^2 \end{pmatrix}, \end{aligned}$$

kde (1) je 3×3 jednotková matice. Výslednou matici lze efektivně vypočítat pomocí matic všech dat (zadaných bodů) (D), kde sloupec i odpovídá vektoru (P_i), $(X) = (D)(D)^T(M)/N$.

2. Najdeme vlastní čísla a vlastní vektory matice (X). Zvolíme nejmenší nezáporné vlastní číslo λ_* a k němu odpovídající vlastní vektor (x_*).
3. K získání hledané sféry nejprve normalizujeme vektor (x_*) ve smyslu získání hodnoty 1 na druhé pozici vektoru, která představuje koeficient u bázevého vektoru e_0 příslušného CGA vektoru x_* . Normalizovaný vlastní vektor má potom tvar $(x_*) = (\mathbf{c}^T, 1, \frac{1}{2}(\|\mathbf{c}\|^2 - r^2))^T$, kde $\mathbf{c} \in \mathbb{R}^3$ je vektor představující střed sféry a r je její poloměr.

Na obrázku 4.2 je znázorněna sféra vygenerovaná tímto postupem. Množina 100 bodů byla generována podle rovnoměrného rozdělení z jednotkové sféry se středem v počátku souřadné soustavy. Každý bod je od sféry ještě vychýlen o náhodnou hodnotu dle rovnoměrného rozdělení nepřesahující hodnotu 0.1. Výsledná sféra získaná při simulaci má střed v bodě $s = (0.0075, 0.0059, -0.0009)$ a poloměr $r = 1.0085$. V případě generování vstupních bodů bez jakékoliv výchyly od původní sféry, tedy takových, které leží na původní sféře, jsme při simulaci jako výsledek získali jednotkovou sféru se středem v počátku, tedy totožnou s původní sférou. Tento případ zde tedy neuvádíme.



Obrázek 4.2: Sféra (modře) aproximující zadané body (červeně).

4.3 Využití prokládání bodů sférou ve zpracování obrazu

V této kapitole si ukážeme možné využití prokládání bodů sférou při práci s barevným obrazem. Nejprve si uvedeme, jakým způsobem lze v CGA realizovat translace geometrických objektů. Připomeňme, že transformovaný CGA objekt o_t získáme z původního objektu o aplikací vztahu

$$o_t = V o \tilde{V},$$

kde versor V určuje příslušnou transformaci. V kapitole 3.1.1 byla touto transformací rotace realizovaná rotorem. Pro translaci využijeme tzv. *translátor*. Translátor je v CGA bivektor ve tvaru

$$T = e^{-\frac{1}{2}\mathbf{t}e_\infty}, \quad (4.10)$$

kde $\mathbf{t} = t_1e_1 + t_2e_2 + t_3e_3$ je vektor vyjadřující příslušnou translaci. Rozvineme-li translátor 4.10 do Taylorovy řady, získáme:

$$T = 1 + \frac{-\frac{1}{2}\mathbf{t}e_\infty}{1!} + \frac{(-\frac{1}{2}\mathbf{t}e_\infty)^2}{2!} + \frac{(-\frac{1}{2}\mathbf{t}e_\infty)^3}{3!} + \dots$$

Protože platí $e_\infty^2 = 0$ a $e_ie_\infty = -e_\infty e_i$ pro $i = 1, 2, 3$, získáváme pro translátor vyjádření

$$T = 1 - \frac{1}{2}\mathbf{t}e_\infty.$$

Podobně jako při rotaci, translaci objektu o realizujeme pomocí vztahu

$$To\tilde{T}.$$

Nyní přejdeme k samotné aplikaci prokládání bodů sférou, při které využijeme i translaci CGA bodů v podobě barevných pixelů obrazu. Hlavní myšlenkou aplikace je, že pro pixely obrazu sestrojíme sféru, která je aproximuje a poté pomocí translací bodů ve směru ke středu sféry nebo od středu sféry můžeme zmenšit nebo zvětšit vzdálenosti mezi jednotlivými barvami obrazu. Ukážeme si, jakým způsobem tuto myšlenku lze zrealizovat. Mějme digitální obraz v barevném prostoru $L^*a^*b^*$ reprezentovaný maticí nad CGA

$$I = L^*e_1 + a^*e_2 + b^*e_3 + \frac{1}{2}(L^*e_1 + a^*e_2 + b^*e_3)^2e_\infty + e_0.$$

Pro pixely obrazu vypočítáme optimální aproximační sféru S . Nechť střed sféry má souřadnice (c_1, c_2, c_3) , potom mějme vektor $\mathbf{c} \in \mathbb{G}_{4,1}$, který představuje tento střed, ve tvaru

$$\mathbf{c} = c_1e_1 + c_2e_2 + c_3e_3.$$

Pro každý pixel určíme translační vektor vztahem

$$\mathbf{t}_{ij} = (L_{ij}^*e_1 + a_{ij}^*e_2 + b_{ij}^*e_3) - (c_1e_1 + c_2e_2 + c_3e_3), \quad (4.11)$$

kde $i = 1, \dots, M$, $j = 1, \dots, N$ jsou indexy obrazové matice. Translátory realizující posun pixelů tedy mají tvar

$$T_{ij} = 1 - \frac{1}{2}\mathbf{t}_{ij}e_\infty.$$

A konkrétní translace pixelu na pozici ij je dána vztahem

$$T_{ij}I_{ij}\tilde{T}_{ij}. \quad (4.12)$$

Poznamenejme, že translace daná vztahem (4.12) představuje posun směrem ke středu aproximační sféry S . Pro posun směrem od středu sféry je potřeba translační vektor (4.11) vynásobit -1 . Myšlenka těchto translací je ta, že v případě translace směrem od středu sféry 4.3b se barevná diference jednotlivých pixelů zvyšuje úměrně jejich původní



(a)



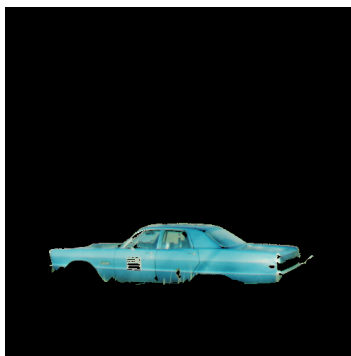
(b)



(c)

Obrázek 4.3: Translace pixelů vzhledem ke středu aproximační sféry. (a) představuje originální obrázek, (b) je translace pixelů směrem ke sféře o hodnotu 30 a (c) je translace od sféry o hodnotu 30.

vzdálenosti. Tedy u pixelů, které se nachází „blízko“ u sebe, se jejich barevná diference zvětší při translaci méně, než u pixelů, které se nachází dále od sebe. Stejným způsobem funguje i opačná translace, tedy ke středu sféry 4.3c. Tuto úpravu obrazu lze využít při segmentaci barev (viz algoritmus v kapitole 3.2). Pro obraz, kde se nám nedaří najít vhodné parametry pro dosažení požadovaného výsledku, nejprve posuneme všechny pixely směrem od středu aproximační sféry a tento upravený obraz použijeme jako vstup algoritmu pro segmentaci barev. Tento postup byl aplikován na obraz 3.7, obrázek 4.4 představuje výsledný obraz pro parametry $(r_1, r_2, r_3) = (80, 60, 60)$. Takto provedená seg-



Obrázek 4.4: Detekce objektu po úpravě barev obrazu s parametry algoritmu pro detekci hran $(80, 60, 60)$.

mentace vypadá skutečně o něco kvalitněji, než výsledky získané bez použití úpravy barev před samotným algoritmem pro detekci objektu. Zejména množství černých oblastí, kde drobné oblasti pixelů na autě nebyly detekovány, je menší a také je patrné zlepšení detekce přední kapoty auta, kde na původním algoritmu byly detekovány některé nežádoucí pixely. Pro vyhodnocení, zda tento postup skutečně vede ke zlepšení detekce objektu, je potřeba v budoucnu provést další experimenty, to už však není obsahem této práce.

Závěr

V této práci jsme se zabývali aplikací konformní geometrické algebry (CGA) ve zpracování barevného obrazu. Zvláštní pozornost byla věnována především oblasti segmentace barev. V první kapitole jsme definovali geometrickou algebru s obecnou signaturou $\mathbb{G}_{p,q}$. Operace násobení na této algebře se nazývá geometrický součin. Dále jsme zavedli operace na geometrických algebrách, zejména vnější součin a vnitřní součin. Důraz byl kladen na vlastnosti těchto operací a uvedena byla i geometrická interpretace. Pomocí Chevalleyovy konstrukce geometrické algebry byl potom znázorněn vztah mezi geometrickou a Grassmannovou algebrou.

Druhá kapitola byla věnována konformní geometrické algebře, neboli CGA. Jedná se o geometrickou algebru $\mathbb{G}_{4,1}$ s neortogonální bází $\{e_1, e_2, e_3, e_0, e_\infty\}$. CGA nám umožňuje jednoduše reprezentovat geometrické objekty, jako jsou bod, sféra, rovina, atd. Body a sféry jsou v CGA reprezentovány pomocí vektorů a bod lze chápat jako sféru s nulovým poloměrem. Ukázali jsme také, že skalárním součinem lze měřit vzdálenost mezi dvěma body. V závěru kapitoly byly představeny některé podalgebry CGA, například komplexní čísla, kvaterniony, či duální čísla.

Kapitoly 3 a 4 jsou věnovány aplikaci CGA. Kapitola 3 obsahuje dvě oblasti využití CGA ve zpracování obrazu. První z nich je detekce hran barevného obrazu pomocí rotačních filtrů vycházející z kvaternionové detekce hran. Obraz je zde reprezentován maticí nad CGA, každý pixel obrazu představuje CGA bod. V CGA je možné realizovat transformace pomocí versorů, kde konkrétně pro rotace máme versory, kterým říkáme rotory. Sestavili jsme bi-konvoluční masky obsahující tyto rotory a následně pomocí bi-konvoluce byly detekovány hrany v obraze. Navíc byly uvedeny 2 způsoby, jak lze separovat z výsledného obrazu detekované hrany. V prvním způsobu byla využita ortogonální projekce a výstupem byl obraz pouze barevných hran. Ve druhém způsobu jsme využili prahování úhlů mezi euklidovskými vektory obrazu s detekovanými hranami a vektoru šedé osy a výsledkem byl binární obraz hran. Hlavním tématem této práce byla segmentace barev. Představili jsme barevný prostor CIELAB, který je opatřen euklidovskou metrikou odpovídající lidskému vnímání barev. Pro segmentaci byla využita skutečnost, že pomocí skalárního součinu bodu a sféry lze rozhodnout o jejich vzájemné poloze. Zvolili jsme referenční barvu pro segmentaci a maximální přípustnou odchylku od této barvy. Množina barev nepřesahující tuto odchylku tvoří v prostoru CIELAB sféru, výsledný obraz jsme tedy poté získali skalárním násobením této sféry s body představujícími pixely vstupního obrazu. Na základě této myšlenky jsme navrhli vlastní algoritmus pro detekci objektu v obraze využívající segmentaci barev. Pro implementaci v Matlabu byl využit speciální toolbox pro práci s geometrickými algebrami. Zdrojový kód celého algoritmu a vygenerované obrázky jsou součástí přílohy této práce. Experimentálně bylo také ověřeno, že prostor CIELAB je pro segmentaci barev využívající CGA vhodnější, než barevný prostor RGB.

V závěrečné kapitole práce jsme se zabývali aproximací bodů sférou ve smyslu metody

nejmenších čtverců. Hledání optimální aproximační sféry pomocí CGA vedlo na problém hledání vlastních čísel a vlastních vektorů matice nad \mathbb{R} . Při implementaci metody v Matlabu byl zvolen odlišný přístup práce s CGA, při kterém nebylo potřeba žádných speciálních balíčků obsahujících geometrické algebry. Možným využitím aproximace bodů sférou ve zpracování obrazu je úprava vzdáleností mezi jednotlivými barvami pomocí translace barev po přímkách procházejících středem sféry aproximující všechny pixely obrazu a jednotlivými pixely. Translací barev směrem od středu sféry bylo dosaženo zvýšení kontrastu mezi jednotlivými barvami, při posunutí ke středu sféry došlo naopak ke snížení tohoto kontrastu. Tento postup by v budoucnu mohl být zakomponován do segmentace barev, pro náš konkrétní experiment došlo k mírnému zlepšení dosaženého výsledku. Pro zhodnocení přínosu této metody je však potřeba provést další, podrobnější experimenty. Celá tato práce je zaměřena především na aplikaci CGA ve zpracování obrazu a obsahuje jak již publikované postupy doplněné o nové vlastní poznatky, tak i některé nové směry využití CGA.

Literatura

- [1] Agudo, J.; Pardo, P.; Sánchez, H.; aj.: A Low-Cost Real Color Picker Based on Arduino. *Sensors*, ročník 14, č. 7, jul 2014: s. 11943–11956, doi:10.3390/s140711943.
- [2] Busin, L.; Vandenbroucke, N.; Macaire, L.: Color Spaces and Image Segmentation. In *Advances in Imaging and Electron Physics*, Elsevier, 2009, s. 65–168, doi:10.1016/s1076-5670(07)00402-8.
- [3] Cheng, H.; Jiang, X.; Sun, Y.; aj.: Color image segmentation: advances and prospects. *Pattern Recognition*, ročník 34, č. 12, dec 2001: s. 2259–2281, doi:10.1016/s0031-3203(00)00149-7.
- [4] D. Hestenes, G. S.: *Clifford Algebra to Geometric Calculus*. Springer Netherlands, 1987, ISBN 9027725616.
URL <https://www.ebook.de/de/product/4351585/d_hestenes_garret_sobczyk_clifford_algebra_to_geometric_calculus.html>
- [5] Dorst, L.: GABLE. [online].
URL <<http://www.cgl.uwaterloo.ca/smnn/GABLE/>>
- [6] Dorst, L.: Total Least Squares Fitting of k -Spheres in n -D Euclidean Space Using an $(n+2)$ -D Isometric Representation. *Journal of Mathematical Imaging and Vision*, ročník 50, č. 3, jan 2014: s. 214–234, doi:10.1007/s10851-014-0495-2.
- [7] Ell, T. A.; Bihan, N. L.; Sangwine, S. J.: *Quaternion Fourier Transforms for Signal and Image Processing*. Hoboken, New Jersey: Wiley, 2014, ISBN 978-1-118-93091-5.
URL <http://www.ebook.de/de/product/22516714/todd_a_ell_nicolas_le_bihan_stephen_j_sangwine_quaternion_fourier_transforms_for_signal_and_image_processing.html>
- [8] Gonzalez, R. C.; Woods, R. E.: *Digital Image Processing*. PRENTICE HALL, třetí vydání, 2007, ISBN 013168728X.
URL <http://www.ebook.de/de/product/6707061/rafael_c_gonzalez_richard_e_woods_digital_image_processing.html>
- [9] Hildenbrand, D.: *Foundations of Geometric Algebra Computing*. Springer Berlin Heidelberg, 2013, ISBN 978-3-642-31793-4, doi:10.1007/978-3-642-31794-1.
- [10] HRUBÝ, J.: *Algebra duálních kvaternionů v analýze obrazu*. Bakalářská práce, Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016, 50 s. Vedoucí bakalářské práce doc. Mgr. Jaroslav Hrdina, Ph.D.

- [11] Jaroslav, H.; Petr, V.; Radomil, M.; aj.: Geometric algebras for uniform colour spaces. *Mathematical Methods in the Applied Sciences*, ročník 0, č. 0, doi:10.1002/mma.4489, <<https://onlinelibrary.wiley.com/doi/pdf/10.1002/mma.4489>>. URL <<https://onlinelibrary.wiley.com/doi/abs/10.1002/mma.4489>>
- [12] Lounesto, P.: *Clifford Algebras and Spinors*. CAMBRIDGE UNIV PR, 2006, ISBN 0521005515.
- [13] Perwass, C.: *Geometric Algebra with Applications in Engineering (Geometry and Computing)*. Springer, 2009.
- [14] Sangwine; Hitzer: Clifford Multivector Toolbox. [Online], 2015, software library available at: <<http://clifford-multivector-toolbox.sourceforge.net/>>.
- [15] TICHÝ, R.: *Reprezentace a transformace barevných prostorů pomocí algebry kvaternionů*. Bakalářská práce, Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016, 44 s. Vedoucí bakalářské práce doc. Mgr. Jaroslav Hrdina, Ph.D.
- [16] Vik, M.: *Základy měření barevnosti I. díl*. Technická univerzita v Liberci, Duben 1995, ISBN 80-7083-162-6.
- [17] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Brno: Computer Press, druhé vydání, 2004, ISBN 80-251-0454-0.

Příloha A

Algoritmus pro detekci objektu

```
%Notations and variables:
%
% I...original image in RGB color space, double format
% Ib...basic image (matrix of real numbers) in Lab color space
%     (White point d50 - norm daylight)
% v...matrix of euclidean clifford vectors
% Icga...image as a matrix of CGA points
% R...matrix of neighbours - in each iteration it is necessary to keep
%     sequence logical(uint8(logical(a_n)-logical(a_n-1))) (we need to
%     sum previous matrices) - It contains even info about amount of
%     summed pixels
% C...matrix of region (neighbour pixels in process) - normed
% A...matrix of ones on possitions of choosen pixels which are considered
%     as correct
% Iout...final output image
% _1,_2,_3 ... indices according to original algorithm

clifford_signature(4,1,0);

%% Load image and alocate matrices

[fName,fPath] = uigetfile('*.tiff;*.tif;*.png;*.jpg',...
'Select reference image','MultiSelect','off');
I = im2double(imread([fPath,fName]));

%I = im2double(imread('../obrazky/color images/kosticky.png'));

Ib = rgb2lab(I,'WhitePoint','d50');
v = Ib(:,:,1)*e1 + Ib(:,:,2)*e2 + Ib(:,:,3)*e3;
[r,c,d] = size(Ib);
Icga = v + 0.5*v.^2*einf + ones(r,c)*eo; %einf a eo jsou pomocne funkce
Ieo = v + ones(r,c)*eo;
clear v;

R0 = false(r,c);
```



```

f1 = figure('Name','Original'); imshow(I);
fprintf('Pravym klikem oznac pixel uprostred objektu\n');
[cP,rP,colP] = impixel;
close

%% Processing

tr1 = 40;
tr2 = 30; %radius of sphere -> initial threshold for neighbor average
tr3 = 40; %threshold for whole region average

x = Ib(rP,cP,1)*e1+Ib(rP,cP,2)*e2+Ib(rP,cP,3)*e3;
P = x + 0.5*x^2*eof + eo;
S2 = P - 0.5*tr2^2*eof; %.....INITIAL SPHERE
S3 = P - 0.5*tr3^2*eof;
S1 = P - 0.5*tr1^2*eof;
clear P2 P3 x;
%%%%%

%Iteration
R0(rP,cP) = true;

h = [1 1 1
     1 0 1
     1 1 1];

A = R0;

Aout = zeros(r,c);
Aout(rP,cP) = true;

%% condition 1 - only one pixel, we are comparing result
A1 = scalar_product(Icga,S1) > 0;

stop = 0;
while stop == 0
R = conv2(double(R0),h,'same') + R0;
R = logical(R);

C = (double(logical(conv2(double(A),h,'same')))) - R0) > 0;

%% condition 2 - average color from confirmed neighbor of
% pixels (only previous step confirmed pixels)

A2 = scalar_product(double(C).*Icga , S2) > 0;

%temporary variable for averaging colors

```

```

temp2 = sum(sum(double(C).*Ieo));
temp2 = temp2/CGA_part(temp2,'eo');

v2 = part(temp2,2)*e1+part(temp2,3)*e2+part(temp2,4)*e3;
P2 = v2 + 0.5*v2^2*einf + eo;
S2 = P2 - 0.5*tr2^2*einf; %new sphere given as average of choosen
    %pixels (A*Icga)
clear temp2 v2 P2;

%% condition 3 - average color from whole confirmed region
% of pixels (all confirmed pixels from all previous steps)

A3 = scalar_product(double(C).*Icga , S3) > 0;

temp3 = sum(sum(double(Aout).*Ieo));
temp3 = temp3/CGA_part(temp3,'eo');

v3 = part(temp3,2)*e1+part(temp3,3)*e2+part(temp3,4)*e3;
P3 = v3 + 0.5*v3^2*einf + eo;
S3 = P3 - 0.5*tr3^2*einf;
clear temp3 v3 P3;

A = A2 .* A3 .* A1;

R0 = R;
Aout = Aout + double(A);
if A == zeros(r,c)
stop = 1;
end

end

Aoutcolor = cat(3,Aout,Aout,Aout) .* I;
Aoutmiss = cat(3,double(Aout==0),double(Aout==0),double(Aout==0)) .* I;

figure('Name','Original'); imshow(I);
figure('Name','Segmented object true'); imshow(Aout);
figure('Name','Segmented original'); imshow(Aoutcolor);

```

Příloha B

Pomocné funkce

function CGA_part

```
function p = CGA_part(m, s)
%Extract eo or einf component from CGA multivector
%-----
%second parametr is string 'eo'/'einf'

narginchk(1, 2), nargoutchk(0, 1)

if nargin < 2
error('Chose component "eo" or "einf"');
end

global clifford_descriptor;

if ~isa(m,'clifford')
error('Input argument has to be of class clifford');
end

if any(bsxfun(@ne,clifford_descriptor.signature,[4 1 0]))
error('Input has to be CGA multivector');
end

X = (part(m,5)+part(m,6))/2;
Y = (part(m,6)-X)*2;

if strcmp(s,'einf')
p = X;
elseif strcmp(s,'eo')
p = Y;
else
error('Second input has to be "eo" or "einf".');
end

end
```

function einf

```
function result = einf
%Basis element einf

narginchk(0,0), nargoutchk(0,1)

global clifford_descriptor;
if any(bsxfun(@ne,clifford_descriptor.signature,[4 1 0]))
error('Basis element e0 is available only for CGA_signature [4 1 0]');
end

result = e5 + e4;
```

function eo

```
function result = eo
%Basis element eo

narginchk(0,0), nargoutchk(0,1)

global clifford_descriptor;
if any(bsxfun(@ne,clifford_descriptor.signature,[4 1 0]))
error('Basis element e0 is available only for CGA_signature [4 1 0]');
end

result = 0.5*(e5-e4);
```

Příloha C

Obrázky

Detekce objektu



Obrázek C.1: Vlevo originál, vpravo detekce žlutého květu $(r_1, r_2, r_3) = (60, 60, 60)$.



Obrázek C.2: Vlevo originál, vpravo detekce těla ptáka $(r_1, r_2, r_3) = (70, 70, 70)$.



Obrázek C.3: Vlevo originál, vpravo detekce bílé věže $(r_1, r_2, r_3) = (40, 30, 40)$.