VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY
INSTITUTE OF MATHEMATICS

REPREZENTACE A TRANSFORMACE BAREVNÝCH PROSTORŮ POMOCÍ ALGEBRY KVATERNIONŮ
REPRESENTATIONS AND TRANSFORMATIONS OF COLOR SPACES VIA QUATERNIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Radek Tichý
AUTHOR

VEDOUCÍ PRÁCE — doc. Mgr. Jaroslav Hrdina, Ph.D.
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Student:	**Radek Tichý**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**doc. Mgr. Jaroslav Hrdina, Ph.D.**
Akademický rok:	2015/16

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 bakalářské práce:

Reprezentace a transformace barevných prostorů pomocí algebry kvaternionů

Stručná charakteristika problematiky úkolu:

Algebra kvaternionů se v mechanice využívá pro eukleidovské transformace třírozměrného prostoru. Prostor barev je také možné chápat jako třírozměrný prostor. Pro použití kvaternionů je ale potřeba pracovat ve vhodně zvolenou reprezentací barev.

Cíle bakalářské práce:

Nastudování algebry kvaternionů a jejich aplikací v mechanice, nastudování základů teorie barevných prostoru. Diskuze transformací určených kvaterniony pro zvolené barevné reprezentace s ohledem na praktické problémy. .

Seznam literatury:

Todd A. Ell, Nicolas Le Bihan, Stephen J. Sangwine, Quaternion Fourier Transforms for Signal and Image Processing, Wiley-ISTE, 2014, 160 s.

Dorst, L., Fontijne D., Mann S., Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry, Morgan Kaufmann, 2010, 664 s.

Perwass CH., Geometric Algebra with Applications in Engineering (Geometry and Computing), Springer, 2009, 386 s.

Motl, L. - Zahradník, M. Pěstujeme lineární algebru, Praha : Univerzita Karlova v Praze, nakladatelství Karolinum, 2002. 348 s.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/16

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 kvaternionů pro detekci hran obrazu. Zaměřuje se především na práci s obrazem reprezentovaným v různých barevných prostorech. Nejprve jsou uvedeny základní pojmy a vlastnosti algebry kvaternionů. Dále jsou popsány některé využívané barevné prostory a poté jsou uvedeny základní filtry sloužící pro detekci hran obrazu v odstínech šedi. Následně jsou představeny barevné filtry využívající kvaterniony pro detekci hran obrazu v barevném prostoru RGB. Na závěr se práce zabývá použitím těchto filtrů pro detekci hran obrazu v prostoru HSV.

Abstract

This thesis deals with quaternions for edge detection, particularly on image processing represented in various color spaces. First, the key concepts and the quaternion algebra properties are mentioned, then some of the commonly used color spaces are described and afterwards the thesis dedicates to the basic filters for edge detection in grayscale image. After that there are presented the color filters that use quaternions for color edge detection in RGB color space. Towards the end, these filters are used for color edge detection in HSV color space.

klíčová slova

algebra kvaternionů, barevný prostor, zpracování obrazu, detekce hran, barevný filtr

key words

quaternion algebra, color space, image processing, edge detection, color filter

TICHÝ, R. *Reprezentace a transformace barevných prostorů pomocí algebry kvaternionů.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 45 s. Vedoucí bakalářské práce doc. Mgr. Jaroslav Hrdina, Ph.D..

Prohlašuji, že jsem bakalářskou práci *Reprezentace a transformace barevných prostorů pomocí algebry kvaternionů* vypracoval samostatně pod vedením doc. Mgr. Jaroslava Hrdiny, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Radek Tichý

Chtěl bych poděkovat vedoucímu práce doc. Mgr. Jaroslavu Hrdinovi, Ph.D. za pozitivní a vstřícný přístup a četné rady při vedení mé bakalářské práce.

Radek Tichý

Obsah

Úvod

Kvaterniony jsou číselný obor, který je zobecněním komplexních čísel se čtyřmi základními jednotkami $1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$. Objevitelem kvaternionů je irský matematik William Rowan Hamilton, který v roce 1843 popsal vztahy pro násobení mezi základními jednotkami $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. Motivací pro tento objev byla geometrická interpretace komplexních čísel jako bodů roviny a snaha vytvořit obecnější obor vyjadřující body v prostoru. Hlavní předností kvaternionů je snadná realizace rotací v prostoru, díky tomu se často využívají například v robotice nebo počítačové grafice (viz [16]).

Tato práce se zabývá využitím kvaternionů pro detekci hran obrazu s ohledem na různé barevné reprezentace obrazu. Práce je rozčleněna do tří hlavních kapitol.

V první kapitole této práce se budeme zabývat právě kvaterniony. Zavedeme důležité operace na množině kvaternionů a budeme zkoumat algebraickou strukturu této množiny. Dále si ukážeme různé reprezentace kvaternionů a především skutečnost, že pomocí kvaternionů lze realizovat rotaci v prostoru. Pro práci s kvaterniony lze využít program Matlab v kombinaci s volně stažitelným balíčkem *Quaternion Toolbox for Matlab* [20].

Druhá kapitola bude věnována práci s obrazem. Matematickou reprezentací obrazu je obrazová matice, která svými prvky jednoznačně určuje barvu jednotlivých pixelů a tím i celý obraz. Tyto matice se však liší podle barevného prostoru, ve kterém jsou jednotlivé barvy obrazu určeny. Uvedeme několik používaných barevných prostorů a závěr kapitoly věnujeme detekci hran obrazu v odstínech šedi. Detekcí hran rozumíme postup vedoucí k nalezení oblasti pixelů, kde se výrazně mění jas. Pro detekci hran lze použít jednoduché detekční filtry, které jsou jednoznačně definovány svými operátory. Tyto operátory jsou reprezentovány maticemi s různými hodnotami pro každý filtr (viz [23]).

V poslední kapitole práce se zaměříme na využití kvaternionů pro detekci hran barevného obrazu. Pro využití detekčních filtrů je potřeba daný obraz reprezentovat pomocí kvaternionů. Tato reprezentace je pro různé barevné prostory rozdílná. Našim cílem bude ukázat právě takové reprezentace obrazu pomocí kvaternionů pro různé barevné prostory. Pro doplnění informací lze doporučit například [11, 13, 4].

1 Algebra kvaternionů

1.1 Definice a pojmy

Nechť $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ jsou komplexní jednotky, pro které platí následující vztahy:

$$\begin{aligned} \boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \\ \boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \\ \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}, \\ \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}. \end{aligned}$$

Potom kvaternion q má tvar

$$q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}, \tag{1.1}$$

kde $a, b, c, d \in \mathbb{R}$. Množinu všech kvaternionů značíme písmenem $\mathbb{H}$. Pro zápis kvaternionu $q \in \mathbb{H}$ můžeme použít tvar

$$q = \mathrm{S}(q) + \mathbf{V}(q), \tag{1.2}$$

kde $\mathrm{S}(q) = a$ je skalární část kvaternionu a $\mathbf{V}(q) = q - \mathrm{S}(q) = b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$ je vektorová část kvaternionu. Tento zjednodušený zápis využijeme později v této kapitole.

Poznámka 1.1. Zřejmě $\mathrm{S}(q) \in \mathbb{R}$, místo skalární část používáme označení reálná část kvaternionu, můžeme značit $\Re(q) = a$. Čísla b, c, d se podobně jako u komplexních čísel běžně označují jako imaginární čísla. Pro rozlišení různých imaginárních čísel se používá značení $\Im_i(q) = b, \Im_j(q) = c, \Im_k(q) = d$ [5].

Definice 1.1. Řekneme, že kvaternion $q \in \mathbb{H}$ je ryzí, jestliže jeho reálná část $\mathrm{S}(q) = 0$. Množinu všech ryzích kvaternionů označíme $\mathbf{V}(\mathbb{H})$.

Jestliže má kvaternion $q \in \mathbb{H}$ nulovou vektorovou část $\mathbf{V}(q) = 0$, potom můžeme tento kvaternion reprezentovat jako prvek z množiny reálných čísel. Množinu všech kvaternionů s nulovou vektorovou částí značíme $\mathrm{S}(\mathbb{H})$ a platí $\mathrm{S}(\mathbb{H}) \equiv \mathbb{R}$.

V úvodu této kapitoly jsme uvedli základní, tzv. kartézský tvar kvaternionu (viz vztah (1.1)). Takto zadefinovaný kvaternion můžeme chápat jako prvek vektorového prostoru dimenze 4 s bází $\{1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$, kvaterniony lze však reprezentovat i jinými způsoby. Některé z nich si v této kapitole uvedeme. Nejdříve se však budeme věnovat základním operacím mezi kvaterniony a uvedeme některé důležité vlastnosti algebry kvaternionů.

Nechť $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$ a $p = e + f\boldsymbol{i} + g\boldsymbol{j} + h\boldsymbol{k}$ jsou libovolné kvaterniony, $q, p \in \mathbb{H}$. Potom jejich součet zavedeme sečtením odpovídajících si složek, tedy:

$$q + p = a + e + (b + f)\boldsymbol{i} + (c + g)\boldsymbol{j} + (d + h)\boldsymbol{k}. \tag{1.3}$$

Užitím vztahu (1.2) lze součet kvaternionů zapsat jako: $q+p = S(q)+S(p)+\mathbf{V}(q)+\mathbf{V}(p)$. Pro součin dvou kvaternionů $q, p \in \mathbb{H}$ máme

$$\begin{aligned} qp &= (a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k})(e + f\boldsymbol{i} + g\boldsymbol{j} + h\boldsymbol{k}) \\ &= ae + af\boldsymbol{i} + ag\boldsymbol{j} + ah\boldsymbol{k} + be\boldsymbol{i} + bf\boldsymbol{i}^2 + bg\boldsymbol{ij} + bh\boldsymbol{ik} \\ &\quad + ce\boldsymbol{j} + cf\boldsymbol{ji} + cg\boldsymbol{j}^2 + ch\boldsymbol{jk} + de\boldsymbol{k} + df\boldsymbol{ki} + dg\boldsymbol{kj} + dh\boldsymbol{k}^2 \\ &= (ae - bf - cg - dh) + (af + be + ch - dg)\boldsymbol{i} \\ &\quad + (ag - bh + ce + df)\boldsymbol{j} + (ah + bg - cf + de)\boldsymbol{k}. \end{aligned} \tag{1.4}$$

Věta 1.1. *Mějme libovolné kvaterniony $q, p \in \mathbb{H}$, $q = \mathrm{S}(q) + \mathbf{V}(q)$, $p = \mathrm{S}(p) + \mathbf{V}(p)$. Potom pro jejich součin platí vztah*

$$qp = \mathrm{S}(q)\mathrm{S}(p) - \langle \mathbf{V}(q), \mathbf{V}(p) \rangle + \mathrm{S}(q)\mathbf{V}(p) + \mathrm{S}(p)\mathbf{V}(q) + \mathbf{V}(q) \times \mathbf{V}(p), \tag{1.5}$$

kde $\langle \mathbf{V}(q), \mathbf{V}(p) \rangle$ je skalární součin vektorů, tedy $\langle \mathbf{V}(q), \mathbf{V}(p) \rangle = bf + cg + dh$,
$\mathbf{V}(q) \times \mathbf{V}(p)$ je klasický vektorový součin dvou vektorů dimenze 3, tedy
$\mathbf{V}(q) \times \mathbf{V}(p) = (ch - dg)\boldsymbol{i} + (df - bh)\boldsymbol{j} + (bg - cf)\boldsymbol{k}$.

Důkaz: Provedeme přímým výpočtem

$$\begin{aligned} qp &= \mathrm{S}(q)\mathrm{S}(p) - \langle \mathbf{V}(q), \mathbf{V}(p) \rangle + \mathrm{S}(q)\mathbf{V}(p) + \mathrm{S}(p)\mathbf{V}(q) + \mathbf{V}(q) \times \mathbf{V}(p) \\ &= ae - (bf + cg + dh) + af\boldsymbol{i} + ag\boldsymbol{j} + ah\boldsymbol{k} + eb\boldsymbol{i} + ec\boldsymbol{j} + ed\boldsymbol{k} \\ &\quad + (ch - dg)\boldsymbol{i} + (df - bh)\boldsymbol{j} + (bg - cf)\boldsymbol{k} \\ &= (ae - bf - cg - dh) + (af + be + ch - dg)\boldsymbol{i} \\ &\quad + (ag - bh + ce + df)\boldsymbol{j} + (ah + bg - cf + de)\boldsymbol{k}. \end{aligned}$$

□

Poznámka 1.2. Ve výpočtech záměrně nepíšeme tečku mezi kvaterniony při jejich násobení, aby nedošlo k záměně se skalárním součinem.

Tvrzení 1.2. *Násobení kvaternionů není komutativní*

$$qp \neq pq,$$

pro $q, p \in \mathbb{H}$.

Důkaz: Důkaz plyne z věty 1.1, protože vektorový součin vektorů není komutativní. Například $\boldsymbol{ij} = \boldsymbol{k}$, $\boldsymbol{ji} = -\boldsymbol{k}$, takže $\boldsymbol{ij} \neq \boldsymbol{ji}$. □

Poznámka 1.3.

1. Násobení kvaternionů je asociativní, pro každé tři kvaterniony $q, p, r \in \mathbb{H}$ platí

   $$q(pr) = (qp)r.$$

2. Pro každé kvaterniony $q, p, r \in \mathbb{H}$ platí levá i pravá distributivita vzhledem ke sčítání, tj.

   $$q(p + r) = qp + qr,$$
   $$(q + p)r = qr + pr.$$

3. Dle tvrzení 1.2 není násobení komutativní, není však ani antikomutativní, tj.

   $$qp \neq -pq,$$

   protože například $1\boldsymbol{i} = \boldsymbol{i} \neq -\boldsymbol{i} = -\boldsymbol{i}1$

Definice 1.2. Nechť $q \in \mathbb{H}$ je libovolný kvaternion. Norma kvaternionu je zobrazení $\mathbb{H} \to \mathbb{R}^+$ definované vztahem

$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{1.6}$$

Jestliže platí $\|q\| = 1$, potom říkáme že q je *jednotkový kvaternion*. Množinu všech jednotkových kvaternionů značíme $\mathbb{H}_1$.

Poznámka 1.4. Normu kvaternionu chápeme jako jeho velikost v Euklidovském 4D prostoru (viz [8]). Pro $\forall p, q \in \mathbb{H}$, $\forall \alpha \in \mathbb{R}$ lze snadno ověřit platnost následujících vlastností.

1. $\|\alpha q\| = |\alpha| \, \|q\|$,
2. $\|qp\| = \|q\| \, \|p\|$,
3. $\|qp\| = \|pq\|$,
4. $\|q\| = 0 \Leftrightarrow q = 0$.

Definice 1.3. Buď $q = \mathrm{S}(q) + \mathbf{V}(q)$ libovolný kvaternion $q \in \mathbb{H}$. Potom kvaternion tvaru

$$\bar{q} = \mathrm{S}(q) - \mathbf{V}(q) = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k} \tag{1.7}$$

nazveme *konjugovaným kvaternionem* ke q a značíme[1] $\bar{q}$.

Poznámka 1.5. Vedle pojmu *konjugovaný kvaternion* existuje pojem *opačný kvaternion* $-q = -a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}$. Tyto dva pojmy splývají v případě ryzích kvaternionů. Pro konjugaci kvaternionu platí následující vlastnosti:

1. $\bar{\bar{q}} = q$,
2. $\overline{qp} = \bar{p}\bar{q}$,

pro $\forall q, p \in \mathbb{H}$.

Vztah pro výpočet normy kvaternionu lze zapsat pomocí konjugovaného kvaternionu. Tento vztah udává následující věta.

Věta 1.3. *Nechť $q \in \mathbb{H}$ je libovolný kvaternion a $\bar{q} \in \mathbb{H}$ je k němu konjugovaný kvaternion. Pak normu $\|q\|$ lze vyjádřit vztahem*

$$\|q\| = \sqrt{q\bar{q}} = \sqrt{\bar{q}q}. \tag{1.8}$$

Důkaz: Podle věty 1.1 dostaneme

$$\begin{aligned}\|q\| = \sqrt{q\bar{q}} = \Big(&a^2 + b^2 + c^2 + d^2 - (ab\boldsymbol{i} + ac\boldsymbol{j} + ad\boldsymbol{k}) + (ab\boldsymbol{i} + ac\boldsymbol{j} + ad\boldsymbol{k}) \\ &+ (-cd + dc)\boldsymbol{i} + (-db + bd)\boldsymbol{j} + (-bc + cb)\boldsymbol{k}\Big)^{\frac{1}{2}} \\ &= \sqrt{a^2 + b^2 + c^2 + d^2}.\end{aligned}$$

Analogicky se ukáže rovnost $q = \sqrt{\bar{q}q}$. Tím je věta dokázána. □

V další větě uvedeme vztah pro inverzi kvaternionu i s důkazem. Tuto větu budeme potřebovat dále v této kapitole pro určení struktury, kterou tvoří množina kvaternionů s operacemi sčítání a násobení.

[1] V některé literatuře se používá jiné značení, např. $\tilde{q}$, q^t, q^*.

Věta 1.4. *Pro každý nenulový kvaternion $q \in \mathbb{H}$ existuje inverzní kvaternion, který je dán vztahem*

$$q^{-1} = \frac{\bar{q}}{\|q\|^2}. \tag{1.9}$$

Tento kvaternion je zároveň jediným inverzním prvkem ke q.

Důkaz: Nejprve dokážeme, že q^{-1} je inverzním kvaternionem ke q, tj. $q^{-1}q = qq^{-1} = 1$. S využitím věty 1.3 dostaneme

$$qq^{-1} = \frac{q\bar{q}}{\|q\|^2} = \frac{\|q\|^2}{\|q\|^2} = 1,$$
$$q^{-1}q = \frac{\bar{q}q}{\|q\|^2} = \frac{\|q\|^2}{\|q\|^2} = 1.$$

Tím jsme dokázali, že q^{-1} je inverzním kvaternionem ke q.
Zbývá dokázat, že je také jediný, to provedeme sporem s využitím asociativity násobení. Nechť p_1, p_2 jsou dva různé inverzní kvaterniony ke kvaternionu q, potom máme

$$p_1 = p_1 1 = p_1(qp_2) = (p_1 q)p_2 = 1p_2 = p_2.$$

Dojdeme tedy ke sporu a tím je věta dokázána. □

Poznámka 1.6. Pro jednotkový kvaternion $q \in \mathbb{H}_1$ zřejmě platí:

$$q^{-1} = \bar{q}. \tag{1.10}$$

Pro dva libovolné kvaterniony $q, p \in \mathbb{H}$, $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$
a $p = e + f\boldsymbol{i} + g\boldsymbol{j} + h\boldsymbol{k}$, definujeme jejich skalární součin[2] vztahem

$$q \cdot p = ae + bf + cg + dh, \tag{1.11}$$

tedy jako skalární součin v $\mathbb{R}^4$.

Definice 1.4. Jestliže pro dva libovolné kvaterniony $q, p \in \mathbb{H}$ platí

$$q \cdot p = 0,$$

řekneme že kvaterniony q a p jsou ortogonální.

Nyní již můžeme vzhledem k předchozím vlastnostem v následující větě popsat algebraickou strukturu, kterou tvoří množina kvaternionů s operacemi sčítání a násobení. Nejprve uvedeme dvě lemmata, která ve větě poté použijeme.

Lemma 1. *Množina $(\mathbb{H}, +)$ tvoří komutativní grupu, tj. pro každé $q, p, r \in \mathbb{H}$ platí následující vlastnosti.*

1. $(q + p) + r = q + (p + r)$,

[2]Značení operace *skalární součin* tečkou vychází z anglického názvu „dot-product“.

2. $q + p = p + q$,

3. *Existuje neutrální prvek* $O \in \mathbb{H}$, *takový že* $q + O = O + q = q$,

4. *Ke každému* $q \in \mathbb{H}$ *existuje opačný prvek* $-q \in \mathbb{H}$.

Důkaz: 1. a 2. lze ihned vidět ze vztahu (1.3), protože víme, že sčítání reálných čísel je komutativní i asociativní. Neutrálním prvkem je nulový kvaternion $0 + 0i + 0j + 0k = 0$. Ze vztahu (1.3) lze snadno vidět, že $q + 0 = 0 + q = q$ pro každé $q \in \mathbb{H}$. Opačným prvkem ke q je opačný kvaternion $-q = -S(q) - \mathbf{V}(q)$. □

Lemma 2. *Množina kvaternionů s operací násobení* $(\mathbb{H}, \cdot)$ *tvoří nekomutativní pologrupu, to znamená, že násobení kvaternionů není komutativní a platí asociativní zákon.*

Důkaz: Násobení kvaternionů je asociativní, viz poznámka 1.3 č. 1. □

Věta 1.5. *Množina všech kvaternionů s operacemi sčítání a násobení* $(\mathbb{H}, +, \cdot)$ *tvoří nekomutativní těleso. To znamená, že* $(\mathbb{H}, +, \cdot)$ *splňuje následující vlastnosti.*

1. $(\mathbb{H}, +)$ *je komutativní grupa.*

2. $(\mathbb{H}, \cdot)$ *je pologrupa (asociativní grupoid).*

3. *Pro každé* $q, p, r \in \mathbb{H}$ *platí levý a pravý distributivní zákon vzhledem ke sčítání,*

$$q(p + r) = qp + qr,$$
$$(q + p)r = qr + pr.$$

4. *Existuje právě jedna jednička (neutrální prvek vzhledem k násobení)* $1 \in \mathbb{H}$.

5. *Pro každý nenulový kvaternion* $q \in \mathbb{H}$ *existuje právě jeden inverzní kvaternion* $q^{-1} \in \mathbb{H}$.

Důkaz:

1. Viz lemma 1.

2. Viz lemma 2.

3. Víme, že levá a pravá distributivita platí, viz poznámka 1.3 č. 2.

4. Jednička je kvaternion $1 + 0i + 0j + 0k = 1$. Ze vztahu (1.5) lze vidět, že pro každé $q \in \mathbb{H}$ platí $q1 = 1q = q$, takže 1 je neutrálním prvkem vzhledem k násobení. Jednoznačnost dokážeme sporem. Nechť p_1, p_2 jsou dva různé neutrální prvky. Protože p_1 je neutrální prvek, platí $p_1p_2 = p_2p_1 = p_2$. Dále, protože p_2 je také neutrální prvek, platí $p_1p_2 = p_2p_1 = p_1$. To ale znamená, že $p_1 = p_2$ a tím jsme dokázali jednoznačnost.

5. Viz věta 1.4.

□

Poznámka 1.7. Kromě základní báze $\{1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ použité v zápisu (1.1), lze použít pro $\mathbb{H}$ i jiné báze. Mějme dva ryzí jednotkové kvaterniony $\boldsymbol{\mu}$ a $\boldsymbol{\nu}$, které jsou navzájem ortogonální. Pro každý ryzí jednotkový kvaternion $q = b\boldsymbol{i}+c\boldsymbol{j}+d\boldsymbol{k}$, $\|q\| = 1$ dostaneme využitím vztahu (1.5):

$$\begin{aligned} q^2 = qq &= -(b^2 + c^2 + d^2) + (cd - dc)\boldsymbol{i} + (db - bd)\boldsymbol{j} + (bc - cb)\boldsymbol{k} \\ &= -1(b^2 + c^2 + d^2) \\ &= -1 \|q\|^2 \\ &= -1. \end{aligned}$$

Pro $\boldsymbol{\mu}, \boldsymbol{\nu}$ tedy platí: $\boldsymbol{\mu}^2 = \boldsymbol{\nu}^2 = -1$. Dále pro součin dvou ryzích, ortogonálních kvaternionů q, p dostaneme:

$$qp = (b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k})(f\boldsymbol{i} + g\boldsymbol{j} + h\boldsymbol{k}) = -\langle \mathbf{V}(q), \mathbf{V}(p) \rangle + \mathbf{V}(q) \times \mathbf{V}(p).$$

Protože skalární součin dvou ortogonálních kvaternionů je nulový, můžeme psát:

$$qp = \mathbf{V}(q) \times \mathbf{V}(p).$$

Pro součin kvaternionů $\boldsymbol{\mu}$ a $\boldsymbol{\nu}$ tedy platí: $\boldsymbol{\mu\nu} = \boldsymbol{\mu} \times \boldsymbol{\nu}$, a proto $\boldsymbol{\mu\nu}$ je ryzí kvaternion ortogonální k $\boldsymbol{\mu}$ a $\boldsymbol{\nu}$. Z vlastnosti normy v poznámce 1.4 č. 2 plyne, že je také jednotkový. Zřejmě jsou všechny tyto kvaterniony taky ortogonální k 1, dostáváme tedy bázi $\{1, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\mu\nu}\}$ a každý kvaternion lze v této bázi zapsat ve tvaru $q = a1 + b\boldsymbol{\mu} + c\boldsymbol{\nu} + d\boldsymbol{\mu\nu}$, kde $a, b, c, d \in \mathbb{R}$. Klasickou bázi $\{1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ dostaneme volbou $\boldsymbol{\mu} = \boldsymbol{i}, \boldsymbol{\nu} = \boldsymbol{j}$.

V předchozím textu jsme uvedli některé důležité vlastnosti algebry kvaternionů a zavedli jsme základní operace na množině kvaternionů. Tento výčet není samozřejmě úplný, více nalezneme např. v [9, 5].

1.2 Další způsoby reprezentace

V této kapitole uvedeme některé další způsoby, jak můžeme reprezentovat kvaterniony.

1.2.1 Polární tvar

Jednou z možností jak reprezentovat kvaterniony je, podobně jako u komplexních čísel, v jejich polárním tvaru. Nejprve ukážeme, že exponenciální funkce, kde proměnnou je ryzí kvaternion, lze zapsat pomocí funkcí cos a sin stejně jako v případě komplexních čísel.

Definice 1.5. Nechť $\xi \in \mathbf{V}(\mathbb{H})$ je nenulový ryzí kvaternion. Exponenciální funkci $e : \mathbf{V}(\mathbb{H}) \to \mathbb{H}$ definujeme pomocí rozvoje do mocninné řady:

$$e^{\xi} = \sum_{n=0}^{+\infty} \frac{\xi^n}{n!} = \sum_{n=0}^{+\infty} \frac{\|\xi\|^n \boldsymbol{\xi}^n}{n!}, \tag{1.12}$$

kde $\boldsymbol{\xi}$ je ryzí jednotkový kvaternion ($\boldsymbol{\xi}^2 = -1$) a $\|\xi\| \in \mathbb{R}^+$.

Poznámka 1.8. Ve vztahu (1.12) je použita vlastnost, že každý kvaternion $q \in \mathbb{H}$ lze zapsat ve tvaru $q = \|q\| \boldsymbol{q}$, kde $\boldsymbol{q}$ je jednotkový kvaternion.

Věta 1.6. *Nechť ξ je ryzí kvaternion, pak pro exponenciální funkci proměnné ξ platí:*

$$e^{\xi} = \cos\|\xi\| + \boldsymbol{\xi}\sin\|\xi\|, \tag{1.13}$$

kde $\boldsymbol{\xi}$ je jednotkový ryzí kvaternion.

Důkaz: Využijeme vztahu (1.12). Víme, že každý kvaternion komutuje s reálným číslem, takže $\|\xi\|$ a $\boldsymbol{\xi}$ komutují. Dále pro jednotkový ryzí kvaternion platí:

$$\boldsymbol{\xi}^n = \begin{cases} (-1)^k & \text{pro} \quad n = 2k \\ (-1)^k\boldsymbol{\xi} & \text{pro} \quad n = 2k+1 \end{cases} \qquad k, n \in \mathbb{Z}^+.$$

Vztah (1.12) tedy můžeme rozepsat do tvaru

$$e^{\xi} = \sum_{k=0}^{+\infty}(-1)^k\frac{\|\xi\|^{2k}}{(2k)!} + \boldsymbol{\xi}\sum_{k=0}^{+\infty}(-1)^k\frac{\|\xi\|^{2k+1}}{(2k+1)!},$$

kde řady na pravé straně představují mocninný rozvoj funkcí cos a sin, dostáváme tedy vztah (1.13). □

Věta 1.7 (Eulerova formule)**.** *Mějme libovolný kvaternion $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$, $q \in \mathbb{H}$. Tento kvaternion můžeme zapsat ve tvaru*

$$q = \|q\|\, e^{\boldsymbol{\mu}_q\phi_q} = \|q\|\,(\cos\phi_q + \boldsymbol{\mu}_q\sin\phi_q). \tag{1.14}$$

$\|q\|$ je norma kvaternionu q, $\boldsymbol{\mu}_q \in \mathbf{V}(\mathbb{H})$ je ryzí jednotkový kvaternion označovaný pojmem osa a ϕ_q je fáze (úhel) kvaternionu q, tedy

$$\|q\| = \sqrt{a^2+b^2+c^2+d^2},$$
$$\boldsymbol{\mu}_q = \frac{b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}}{\sqrt{b^2+c^2+d^2}},$$
$$\phi_q = \arctan\left(\frac{\sqrt{b^2+c^2+d^2}}{a}\right).$$

Důkaz: Druhá rovnost ve vztahu (1.14) platí, viz věta 1.6. Dokážeme rovnost $q = \|q\|\,(\cos\phi_q + \boldsymbol{\mu}_q\sin\phi_q)$ a tím i celý vztah. Víme, že skalární kvaternion a ryzí kvaternion jsou navzájem ortogonální, jejich součtem je původní kvaternion. Roznásobením závorky a vyjádřením goniometrických funkcí $\cos\phi_q, \sin\phi_q$, dostaneme pro jednotlivé členy:

$$\|q\|\cos\phi_q = \left(\sqrt{a^2+b^2+c^2+d^2}\right)\frac{a}{\sqrt{a^2+b^2+c^2+d^2}} = a,$$
$$\|q\|\,\boldsymbol{\mu}_q\sin\phi_q = \left(\sqrt{a^2+b^2+c^2+d^2}\right)\frac{b\boldsymbol{i}+c\boldsymbol{j}+d\boldsymbol{k}}{\sqrt{b^2+c^2+d^2}}\frac{\sqrt{b^2+c^2+d^2}}{\sqrt{a^2+b^2+c^2+d^2}} = b\boldsymbol{i}+c\boldsymbol{j}+d\boldsymbol{k}.$$

Sečtením těchto členů tedy dostáváme rovnost $q = \|q\|\,(\cos\phi_q + \boldsymbol{\mu}_q\sin\phi_q)$. Ve vztahu (1.12) tedy musí platit i první rovnost a tím je věta dokázána. □

Poznámka 1.9. Fáze kvaternionu představuje úhel mezi původním kvaternionem a jeho skalární částí (skalárním kvaternionem). Podle věty 1.7, pokud máme ryzí kvaternion, pak jeho fáze je rovna $\pi/2$. Osa kvaternionu představuje směr jeho vektorové části (vektoru) ve 3D prostoru.

Kromě zmíněné Eulerovy formule existují další způsoby zápisu kvaternionu. Například Cayleova-Dicksonova forma, která umožňuje zapsat kvaternion pomocí dvou komplexních čísel ve tvaru $q = z_1 + z_2\boldsymbol{j}$, kde $z_1 = a + b\boldsymbol{i} \in \mathbb{C},\ z_2 = c + d\boldsymbol{i} \in \mathbb{C}$, nebo tvar s Eulerovými úhly atd. Více k tomuto tématu najdeme v [5, 18].

1.2.2 Maticová reprezentace

Kvaterniony můžeme vyjádřit také pomocí matic. Existují dva způsoby takové reprezentace, ve tvaru komplexních matic $\mathbb{C}^{2\times2}$ nebo ve tvaru reálných matic $\mathbb{R}^{4\times4}$.

Nejprve uvedeme reprezentaci pomocí komplexní matice.

Věta 1.8. *Libovolný kvaternion $q \in \mathbb{H}$ můžeme zapsat pomocí komplexní matice $\mathcal{M}_{\mathbb{C}}(q) \in \mathbb{C}^{2\times2}$ ve tvaru:*

$$\begin{aligned}\mathcal{M}_{\mathbb{C}}(q) &= \begin{pmatrix} \alpha & -\beta \\ \beta^* & \alpha^* \end{pmatrix} \\ &= a\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + b\begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} + c\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + d\begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix},\end{aligned} \tag{1.15}$$

kde $\alpha = a + bi$, $\beta = c + di$ jsou komplexní čísla. Zápisem α^* chápeme komplexně sdružené číslo.

Důkaz: Porovnáním této reprezentace s kartézským tvarem získáme identifikaci základních jednotek:

$$1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, i = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}, j = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, k = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}.$$

Maticovým násobením ověříme platnost vztahů pro násobení mezi komplexními jednotkami.

$$\begin{aligned}\boldsymbol{i}^2 &= \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}\begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \\ \boldsymbol{j}^2 &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \\ \boldsymbol{k}^2 &= \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \\ \boldsymbol{ijk} &= \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}\begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.\end{aligned}$$

□

Nyní reprezentace pomocí reálné matice.

Věta 1.9. *Libovolný kvaternion $q \in \mathbb{H}$ lze zapsat pomocí reálné matice $\mathcal{M}_{\mathbb{R}}(q) \in \mathbb{R}^{4\times 4}$ ve tvaru:*

$$\begin{aligned}\mathcal{M}_{\mathbb{R}}(q) &= \begin{pmatrix} a & b & c & d \\ -b & a & -d & c \\ -c & d & a & -b \\ -d & -c & b & a \end{pmatrix} \\ &= a\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + b\begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ &+ c\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} + d\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}.\end{aligned} \tag{1.16}$$

Důkaz: Ověřením vztahů mezi základními jednotkami stejným způsobem jako pro reprezentaci komplexní maticí. □

Prostory takto zadefinovaných kvaternionů (pomocí reálných nebo komplexních matic) jsou izomorfní s prostorem kartézské reprezentace kvaternionů. Například násobení kvaternionů je zde reprezentováno násobením matic. Ukážeme si, čemu je roven determinant matice. Pro reprezentaci komplexními maticemi máme:

$$\begin{aligned}\det\begin{pmatrix} \alpha & -\beta \\ \beta^* & \alpha^* \end{pmatrix} &= \det\begin{pmatrix} a+bi & -c-di \\ c-di & a-bi \end{pmatrix} \\ &= (a+bi)(a-bi) - (-c-di)(c-di) \\ &= a^2 - (bi)^2 + c^2 - (di)^2 \\ &= a^2 + b^2 + c^2 + d^2,\end{aligned}$$

což je druhá mocnina normy kvaternionu. Stejný výsledek dostaneme i pro reálné matice. Více o kvaternionových maticích najdeme v [22].

1.3 Rotace ve 3D

V této kapitole si ukážeme jak lze pomocí kvaternionů realizovat rotace v prostoru a uvedeme si základní vlastnosti těchto rotací. Čerpat budeme především z [5, 9, 7, 10].

Poznamenejme nejprve, že vektory a body v prostoru můžeme reprezentovat pomocí ryzích kvaternionů. Například vektor $\mathbf{v} = (x, y, z),\ \mathbf{v} \in \mathbb{R}^3$ napíšeme pomocí kvaternionu $q \in \mathbb{H}$ jako $q = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$. Dále z věty 1.7 vyplývá, že každý jednotkový kvaternion (tzn. $q \in \mathbb{H},\ \|q\| = 1$) můžeme napsat ve tvaru

$$q = \cos\phi + \boldsymbol{\mu}\sin\phi. \tag{1.17}$$

Takovéto kvaterniony označujeme pojmem *rotory*. Připomeňme, že pro jednotkové kvaterniony $q, p \in \mathbb{H}_1$ platí:

$$\|qp\| = 1, \qquad \bar{q} = q^{-1}.$$

Tvrzení 1.10. *Množina jednotkových kvaternionů (rotorů) s operací násobení tvoří grupu.*

Důkaz: Stačí ukázat, že rotory jsou uzavřené na násobení včetně neutrálního prvku a inverze. Uzavřenost na násobení ihned plyne z vlastnosti $\|qp\| = 1$ pro $\forall q, p \in \mathbb{H}_1$. Dále platí $\|1 + 0\boldsymbol{i} + 0\boldsymbol{j} + 0\boldsymbol{k}\| = \|1\| = 1$, tím jsme ukázali, že neutrální prvek $1 \in \mathbb{H}_1$. Pro každé $q \in \mathbb{H}_1$ platí: $q^{-1} = \bar{q}$ (viz poznámka 1.6). Ukážeme, že $\bar{q}$ je také jednotkový kvaternion. Nechť $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$, $\|q\| = 1$, $a, b, c, d \in \mathbb{R}$. Norma inverzního prvku je:

$$\begin{aligned}\left\|q^{-1}\right\| &= \|a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}\| \\ &= \sqrt{a^2 + (-b)^2 + (-c)^2 + (-d)^2} \\ &= \sqrt{a^2 + b^2 + c^2 + d^2} \\ &= \|q\| = 1,\end{aligned}$$

platí tedy, že $q^{-1} \in \mathbb{H}_1$ a tím je tvrzení dokázáno. □

Nyní můžeme v následující větě popsat rotaci vektoru v prostoru pomocí jednotkových kvaternionů.

Věta 1.11. *Mějme třírozměrný vektor $(x, y, z) \in \mathbb{R}^3$ reprezentovaný ryzím kvaternionem $p = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ a libovolný rotor $q = \cos\phi + \boldsymbol{\mu}\sin\phi$, $\boldsymbol{\mu} \in \mathbb{H}_1$. Potom*

$$\mathcal{R}_q(p) = qp\bar{q} \tag{1.18}$$

reprezentuje rotaci vektoru p kolem osy $\boldsymbol{\mu}$ o úhel 2ϕ. $\mathcal{R}_q(\cdot)$ nazýváme operátor rotace příslušný kvaternionu q.

Důkaz: Důkaz nalezneme v [9]. □

Demonstrujeme platnost předchozí věty na příkladu s konkrétně zvolenými parametry.

Příklad 1.1. Chceme například realizovat rotaci kolem osy z o úhel α. Osu rotace tedy bude představovat kvaternion

$$\boldsymbol{\mu} = 0\boldsymbol{i} + 0\boldsymbol{j} + 1\boldsymbol{k}$$

a příslušný rotor je tedy

$$q = \cos\frac{\alpha}{2} + \boldsymbol{\mu}\sin\frac{\alpha}{2} = \cos\frac{\alpha}{2} + \boldsymbol{k}\sin\frac{\alpha}{2}.$$

Použijeme-li operátor rotace na obecný vektor v $\mathbb{R}^3$ $p = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$, dostáváme:

$$\begin{aligned}\mathcal{R}_q(p) &= (\cos\frac{\alpha}{2} + \boldsymbol{k}\sin\frac{\alpha}{2})(x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k})(\cos\frac{\alpha}{2} - \boldsymbol{k}\sin\frac{\alpha}{2}) \\ &= x\boldsymbol{i}\cos^2\frac{\alpha}{2} + y\boldsymbol{j}\cos^2\frac{\alpha}{2} + z\boldsymbol{k}\cos^2\frac{\alpha}{2} \\ &+ x\boldsymbol{j}\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} - y\boldsymbol{i}\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} + z\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} \\ &+ x\boldsymbol{j}\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} - y\boldsymbol{i}\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} - z\cos\frac{\alpha}{2}\sin\frac{\alpha}{2} \\ &- x\boldsymbol{i}\sin^2\frac{\alpha}{2} - y\boldsymbol{j}\sin^2\frac{\alpha}{2} + z\boldsymbol{k}\sin^2\frac{\alpha}{2} \\ &= \boldsymbol{i}\left[x(\cos^2\frac{\alpha}{2} - \sin^2\frac{\alpha}{2}) - 2y\cos\frac{\alpha}{2}\sin\frac{\alpha}{2}\right] \\ &+ \boldsymbol{j}\left[y(\cos^2\frac{\alpha}{2} - \sin^2\frac{\alpha}{2}) + 2x\cos\frac{\alpha}{2}\sin\frac{\alpha}{2}\right] \\ &+ \boldsymbol{k}\left[z(\sin^2\frac{\alpha}{2} + \cos^2\frac{\alpha}{2})\right].\end{aligned}$$

Pro úpravu poslední rovnice využijeme goniometrických identit:

$$\sin(2\alpha) = 2\cos\alpha\sin\alpha,$$
$$\cos(2\alpha) = \cos^2\alpha - \sin^2\alpha.$$

Dostáváme tedy výsledný tvar

$$\mathcal{R}_q(p) = \boldsymbol{i}(x\cos\alpha - y\sin\alpha) + \boldsymbol{j}(y\cos\alpha + x\sin\alpha) + \boldsymbol{k}z,$$

který při identifikaci s $\mathbb{R}^3$ představuje transformaci, kterou lze zapsat maticově jako

$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Dostali jsme tedy známou matici rotace, která rotuje vektor p podle osy z o úhel α (viz [17]). Ukázali jsme, že operátor $\mathcal{R}_q(p)$ rotuje libovolný vektor $p \in \mathbb{R}^3$ o úhel $\alpha \in \langle 0, 2\pi\rangle$ kolem předem zvolené osy z.

Poznámka 1.10. Protože q je jednotkový kvaternion, můžeme pro operátor rotace použít zápis:

$$\mathcal{R}_q(p) = qpq^{-1},$$

který je ekvivalentní se vztahem (1.18).

Inverzní operátor rotace $\mathcal{R}_q^{-1}(\cdot)$ je dán vztahem:

$$\mathcal{R}_q^{-1}(p) = \bar{q}pq$$

a představuje rotaci kolem stejné osy jako operátor $\mathcal{R}_q(p)$ ve vztahu (1.18), ovšem v opačném směru, tedy o úhel -2ϕ.

Věta 1.12. *Skládání rotací lze reprezentovat násobením příslušných kvaternionů. To znamená, že výsledek rotace získané použitím operátoru $\mathcal{R}_{q_2}(\cdot)$ následovaného operátorem $\mathcal{R}_{q_1}(\cdot)$ je stejný jako v případě použití operátoru $\mathcal{R}_{q_1q_2}(\cdot)$.*

Důkaz: Víme, že množina jednotkových kvaternionů tvoří grupu, násobením jednotkových kvaternionů q_1, q_2 tedy získáme opět jednotkový kvaternion. Složením příslušných rotací získáváme:

$$\begin{aligned}\mathcal{R}_{q_1}(\mathcal{R}_{q_2}(p)) &= q_1(q_2p\overline{q_2})\overline{q_1} \\ &= (q_1q_2)p(\overline{q_2}\,\overline{q_1}) \\ &= (q_1q_2)p\overline{(q_1q_2)} = \mathcal{R}_{q_1q_2}(p).\end{aligned}$$

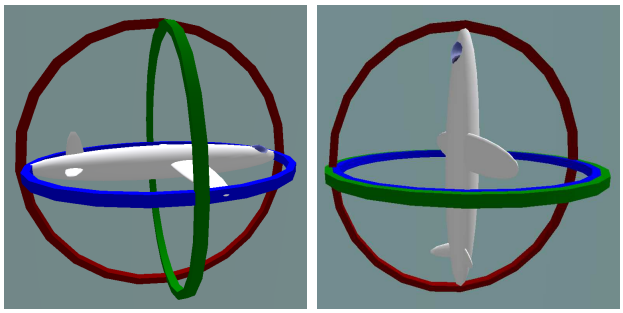
□

Rotace využívající kvaterniony mají několik výhod oproti maticím rotací. Rotace pomocí matic využívají tzv. Eulerovy úhly, kde pro rotaci kolem obecné osy je potřeba skládat postupně rotace kolem tří základních os (viz [21]). V tomto případě je tedy důležité pořadí těchto jednotlivých rotací, dále matice rotace obsahuje oproti kvaternionu

nadbytečné množství informací (9 hodnot oproti 4 hodnotám v případě kvaternionu). Matici rotace kolem obecné osy získáme násobením tří matic představujících rotace kolem jednotlivých základních os:

$$R(\alpha,\beta,\gamma) = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{pmatrix}.$$

Další nevýhodou matic je, že rotace kolem jedné osy ovlivňuje i ostatní osy. Tím může dojít k překrytí dvou os a ztrátě jednoho stupně volnosti. Tomuto efektu se říká „gimbal lock“ (viz obrázek 1), více o tomto problému nalezneme v [9, 6].



Obrázek 1: „Gimbal lock“ efekt[3]

V případě kvaternionů tento nežádoucí efekt nemůže nastat. Podrobnější srovnání rotací pomocí matic a jednotkových kvaternionů nalezneme v [9, 12].

V praxi lze využít kvaterniony téměř ve všech simulacích, kde se využívají rotace, a pro které nejsou vhodné maticové reprezentace, například letové simulace nebo popis oběhů družic. Uplatnění kvaternionů najdeme i v lékařství, kde se využívají pro grafickou vizualizaci nebo ovládání lékařských sond [15]. Dalším odvětvím, kde kvaterniony nachází uplatnění je robotika. Pro ovládání polohy robotického ramena je výhodnější využít kvaterniony než matice. V další části textu se budeme zabývat využitím kvaternionů v počítačové grafice, konkrétně v disciplíně nazvané zpracování obrazu.

1.4 Kvaternionový balíček pro Matlab

V další kapitole budeme pro zpracování obrazu využívat software *Matlab R2015a*. Samotný Matlab v sobě však nemá implementovány kvaterniony, tudíž s nimi neumí žádným způsobem pracovat. K tomuto účelu byl však vyvinut balíček s názvem *Quaternion and octonion toolbox for Matlab*, zkráceně QTFM, který tuto funkci v Matlabu umožňuje. Balíček QTFM je volně ke stažení a jeho autory jsou Stephen J. Sangwine a Nicolas Le Bihan. První verze vyšla v srpnu 2005, od té doby bylo vydáno několik upravených verzí s přidanými novými funkcemi. Verze, kterou budeme používat, je QTFM 2.3 vydaná v lednu 2016, která umožňuje dokonce i práci s oktoniony, což je další zobecnění komplexních čísel dimenze 8 (viz [2]). My se však omezíme pouze na kvaterniony. V dalším se zaměříme na

[3]MathsPoetry : Commons.wikimedia.org: File:Gimbal lock.png [online]. 14 February 2009 [cit. 2016-04-15]. Dostupný pod licencí Creative Commons na www: <https://commons.wikimedia.org/wiki/File:Gimbal_lock.png>

popis některých funkcí, které QTFM umožňuje a budeme k nim přikládat ukázky zápisu v Matlabu (viz [20]).

Jak již bylo řečeno, balíček QTFM umožňuje v Matlabu pracovat s kvaterniony a maticemi kvaternionů. Je navržen tak, že je možné využívat pro práci s kvaterniony i některé funkce implementované v základní verzi Matlabu.

Zápis kvaternionu v kartézském tvaru můžeme provést dvěma způsoby. První je pomocí nadefinovaných operátorů $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$, které se v QTFM zapisují jako `qi, qj, qk`. Chceme-li tedy zapsat konkrétní kvaternion $2 + \boldsymbol{i} - 3\boldsymbol{j} + \boldsymbol{k}$, zadáme do příkazového řádku Matlabu následující.

```
>> 2+qi-3*qj+qk
ans =
2 + 1 * I - 3 * J + 1 * K
```

Tento způsob je vhodný pro zápis jednoho kvaternionu. Jinou možností je zápis pomocí funkce *quaternion*. Tato funkce má čtyři volitelné parametry `quaternion(w,x,y,z)`. Pokud nezadáme žádný parametr, Matlab uloží pole (matici) 0×0 kvaternionů, tedy prázdnou množinu, jejíž „prvky“ jsou pro Matlab typu kvaternion. Jeden kvaternion získáme pokud zadáme tři číselné parametry (ryzí kvaternion $x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$, kde x, y, z jsou zadaná čísla) nebo čtyři číselné parametry (kvaternion $w + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$, kde w, x, y, z jsou zadaná čísla). Chceme-li zapsat pomocí této funkce stejný kvaternion jako u předchozího příkladu, zadáme toto.

```
>> quaternion(2,1,-3,1)
ans =
2 + 1 * I - 3 * J + 1 * K
```

Funkce *quaternion* je užitečná především chceme-li vytvořit matici kvaternionů, potom jako parametry w, x, y, z zadáváme matice stejných rozměrů, a jejíchž prvky jsou stejných datových typů. Jednotlivé koeficienty u základních jednotek kvaternionu můžeme získat pomocí funkce `part(q,n)`, kde parametr q je kvaternion (respektive matice kvaternionů) a n je index vyjadřující pozici koeficientu (respektive koeficientů). Při použití notace z poznámky 1.1 platí: $n = 1$ pro $S(q)$, $n = 2$ pro $\Im_i(q)$, $n = 3$ pro $\Im_j(q)$ a $n = 4$ pro $\Im_k(q)$. Výstupem je číslo (respektive matice čísel). Stejné výsledky získáme i použitím elegantnějších příkazů `s(q)` pro $S(q)$, `x(q)` pro $\Im_i(q)$, `y(q)` pro $\Im_j(q)$ a `z(q)` pro $\Im_k(q)$. Balíček QTFM obsahuje také funkce pro práci s kvaternionem v polárním tvaru. Připomeňme, že každý kvaternion lze zapsat v polárnim tvaru, tedy $q = \|q\| \exp(\boldsymbol{\mu}\phi)$, kde $\boldsymbol{\mu}$ je osa a ϕ je úhel (viz vztah (1.14)). Balíček QTFM podporuje zápis kvaternionu v polárním tvaru pomocí Matlabovské funkce `exp()`, kde zadáváme součin jednotkového kvaternionu (osy) a skaláru (úhel v radiánech). Pro libovolný kvaternion můžeme také určit úhel ϕ a osu $\boldsymbol{\mu}$ jeho polárního tvaru. Pro úhel kvaternionu obsahuje balíček QTFM funkci `angle(q)`, za parametr q zadaváme kvaternion a výstupem je právě úhel ϕ kvaternionu q v radiánech. Výstupem funkce `axis(q)` je osa $\boldsymbol{\mu}$ kvaternionu q. Obě tyto funkce lze opět použít i pro celou matici kvaternionů a výstupem bude matice úhlů nebo os.

V tomto odstavci si uvedeme některé operace na algebře kvaternionů, které jsou součástí balíčku QTFM. Funkce `abs(q)` vrací normu kvaternionu (viz vztah (1.6)), respektive každého prvku matice, pokud zadáme jako vstup matici kvaternionů. Důležitou funkcí především při rotaci pomocí kvaternionů je konjugování. Pro konjugování kvaternionů obsahuje balíček QTFM funkci `conj(q)`. Ukážeme si jak bude vypadat zápis v Matlabu a výstup s konkrétním kvaternionem $q = 2 + \boldsymbol{i} - 3\boldsymbol{j} + \boldsymbol{k}$.

```
>> conj(2+qi-3*qj+qk)
ans =
2 - 1 * I + 3 * J - 1 * K
```

Zobrazený výsledek je v souladu se vztahem (1.7) pro konjugaci kvaternionu. Tato funkce může mít ještě druhý parametr `conj(q,s)`, který určuje „způsob konjugace“. Tento parametr může mít hodnotu `conj(q,'complex')`, potom jsou všechny čtyři koeficienty kvaternionu, tj. a, b, c, d pro kvaternion $q = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$, nahrazeny komplexně sdruženými čísly. Tento parametr má tedy vliv pouze pokud zadáme za koeficienty kvaternionu v Matlabu datový typ *complex*, tedy komplexní čísla. Takové kvaterniony jsou v dokumentaci balíčku označovány pojmem *bikvaterniony*. Pokud jsou koeficienty reálná čísla, výstupem bude kvaternion q. Dále může mít druhý parametr funkce *conj* hodnotu `conj(q,'total')`, potom se provede nahrazení koeficientů komplexně sdruženými čísly a zároveň dojde ke klasické kvaternionové konjugaci. Při rotaci využíváme jednotkové kvaterniony. Velký význam tedy pro nás bude mít i funkce `unit(q)`, která vrací jednotkový kvaternion.

```
>> unit(2+qi-3*qj+qk)
ans =
0.5164 + 0.2582 * I - 0.7746 * J + 0.2582 * K
```

Jak je vidět, pokud na vstup zadáme kvaternion q, výstupem bude kvaternion jehož koeficienty jsou koeficienty původního kvaternionu vydělené jeho normou. Za zmínku stojí také funkce `scalar_product(q,p)`, která vrací skalární součin zadaných kvaternionů.

Z dalších funkcí zmíníme ještě `convert(q,t)`, díky které můžeme měnit datový typ koeficientů příslušného kvaternionu. Prvním parametrem je kvaternion a druhý parametr je datový typ, ve kterém chceme, aby Matlab s daným kvaternionem pracoval, například *double* pro reálné koeficienty nebo pro zpracování obrazu často využívaný typ *uint8*, což jsou celá čísla nabývající hodnot $0-255$. Konkrétní příkaz pro převedení kvaternionu q do tohoto datového typu je potom `convert(q,'unit8')` (q může být i matice kvaternionů). Poslední funkcí, kterou uvedeme je `conv2(A,B)`. Tato funkce počítá kvaternionovou konvoluci, parametry A a B jsou matice kvaternionů. Definici konvoluce si uvedeme později. V dalších kapitolách z určitých důvodů, které si také ukážeme, budeme využívat tzv. bikonvoluci `conv2({L,R},B)`. Parametry L, R, B jsou opět matice kvaternionů (viz kapitola 3).

2 Zpracování obrazu

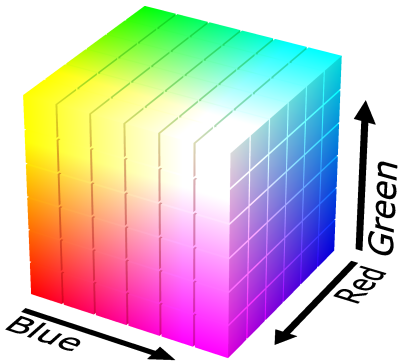
2.1 Barevné prostory

Při práci s barevným obrazem je potřeba reprezentovat nějakým způsobem každý pixel daného obrázku. K tomuto účelu se používají tzv. obrazové matice, ve kterých je uložena informace o pozici a barvě daného pixelu. K reprezentaci určité barvy je však potřeba zvolit konkrétní *barevný prostor*, který určí způsob, jakým bude daná barva reprezentována.

Barevný prostor je přesně definovaná množina barev, kterou je schopno dané zařízení zobrazit. Barevných prostorů dnes existuje již celá řada, v tomto textu uvedeme několik základních prostorů.

2.1.1 Prostory RGB a RGBα

Konkrétní barvy získáme kombinací tří základních složek (základní barvy) - *červená* (R, „red“), *zelená* (G, „green“), *modrá* (B, „blue“). Základní složky bývají uváděny buď v celočíselném rozsahu $0 - 255$, nebo nabývají reálných hodnot $\langle 0, 1\rangle$. Tento prostor můžeme potom zobrazit pomocí jednotkové krychle (viz obrázek 2) a každá barva je určena barevným vektorem $\mathbf{v} = (R, G, B) \in \langle 0, 1\rangle \times \langle 0, 1\rangle \times \langle 0, 1\rangle$.



Obrázek 2: RGB krychle[4]

Hodnota 0 znamená, že se daná složka ve složené barvě vůbec nevyskytuje, naopak hodnota 1 říká, že daná složka nabývá své maximální intenzity.

Příklad 2.1.

1. Vektor, který nabývá maximálních hodnot ve všech třech složkách $\mathbf{v}_1 = (1, 1, 1)$, tedy každá složka má maximální intenzitu, představuje bílou barvu.

2. Vektor $\mathbf{v}_2 = (0, 0, 0)$, tedy počátek souřadného systému, představuje černou barvu.

[4]SharkD : Commons.wikimedia.org: File:RGB color solid cube.png [online]. 12 January 2008 [cit. 2016-04-26]. Dostupný pod licencí Creative Commons na www: <https://commons.wikimedia.org/wiki/File:RGB_color_solid_cube.png>

3. Šedá barva se nachází ve středu krychle. Různé odstíny šedi získáme složením složek o stejných intenzitách, tedy $\mathbf{v}_3 = (k, k, k),\ k \in \langle 0, 1 \rangle$.

Při práci s obrazem bývá často potřeba převést barevný obraz na obraz v odstínech šedi, tedy nahradit barvy různými odstíny šedi. Z důvodu toho, že lidské oko vnímá různě intenzity jednotlivých složek, není možné použít tzv. *objektivní jas* pro určení odstínu šedi, který je dán vztahem $J_o = \frac{R+G+B}{3}$. Místo toho se tedy používá *subjektivní jas*, který je dán empirickým vztahem:

$$J = 0.299R + 0.587G + 0.114B.$$

Poznámka 2.1. Matematickou reprezentací barevného obrazu jsou obrazové matice. V případě prostoru RGB máme trojici obrazových matic $(\mathbf{R}, \mathbf{G}, \mathbf{B})$, kde každý prvek matice představuje hodnotu intenzity příslušné složky na konkrétní pozici obrazu. Barva pixelu na pozici (i, j) je tedy dána vektorem $\mathbf{v}_{ij} = (R_{ij}, G_{ij}, B_{ij})$. V případě transformace barevného obrazu na obraz v odstínech šedi dojde k nahrazení trojice $(\mathbf{R}, \mathbf{G}, \mathbf{B})$ jedinou maticí $\mathbf{J}$, jejíž prvky představují hodnoty jasu.

Barevný obraz zapsaný v prostoru RGB můžeme doplnit o další informaci. Touto informací může být například *průhlednost*, takový barevný prostor poté značíme RGBα. V tomto prostoru pak každý bod obsahuje navíc skalární údaj nabývající hodnot $\langle 0, 1 \rangle$. Hodnota 0 poté představuje zcela neprůhledný barevný bod a hodnota 1 průhledný. Tato doplňková informace se využívá při kombinování více obrazů do jednoho celku.

2.1.2 Prostory CMY a CMYK

Prostor RGB využívá *aditivní* skládání barev, to znamená, že čím více barev složíme, tím světlejší je výsledek. Prostor CMY funguje opačně, tedy využívá *subtraktivní* skládání barev, tj. čím více barev složíme, tím tmavší výsledek. Základní barvy tohoto prostoru jsou: *azurová* (C, „cyan“), *purpurová* (M, „magenta“) a *žlutá* (Y, „yellow“). Tento prostor můžeme opět stejně jako prostor RGB znázornit jednotkovou krychlí. Bílou barvu zde představuje vektor $(0, 0, 0)$, tedy počátek souřadného systému a černou $(1, 1, 1)$. Snadno lze převádět barvy mezi prostory RGB a CMY. Mějme barevný vektor (r, g, b) v prostoru RGB, potom stejnou barvu v prostoru CMY vyjádříme vektorem (c, m, y), který je dán vztahem

$$\begin{pmatrix} c \\ m \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} r \\ g \\ b \end{pmatrix}.$$

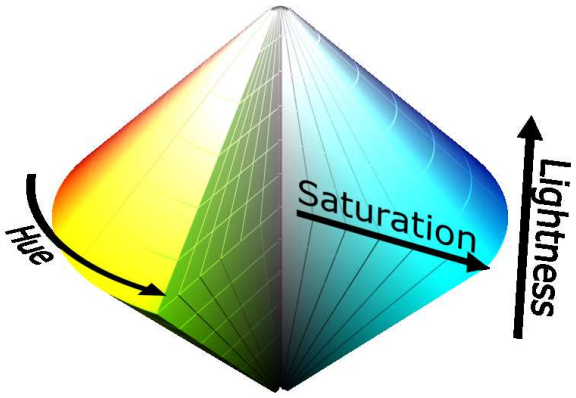
Prostor CMY se využívá například při tisku. V tomto případě je však neekonomické získávat černou barvu skládáním tří barev, proto se ke třem základním barvám přidává ještě *černá* jako samostatná barva. Tím dostáváme prostor CMYK (K, „blacK“), který má oproti RGB a CMY (dimenze 3) dimenzi 4.

2.1.3 Prostory HSV a HLS

Prostory HSV a HLS opět definují barvy třemi složkami, oproti předchozím prostorům však už tyto složky nepředstavují základní barvy.

V případě prostoru **HSV** jde o *barevný tón* (H, „hue“), *sytost* (S, „saturation“) a *jas* (V, „value“). Pro zobrazení tohoto prostoru se používá šestiboký jehlan. Vrchol jehlanu leží v počátku souřadnic HSV. Sytost S nabývá hodnot z intervalu $\langle 0, 1\rangle$ a představuje vodorovnou osu (kolmou na osu jehlanu), barevný tón H je dán úhlem $\langle 0°, 360°)$. Jas tvoří svislou osu totožnou s osou jehlanu a nabývá opět hodnot $\langle 0, 1\rangle$. Vrchol jehlanu, představuje černou barvu, bílá barva je ve středu podstavy, tedy v bodě s maximální hodnotou jasu a nulovou sytostí. Barevný tón a sytost spolu tvoří obdobu polárního souřadného systému s tím rozdílem, že při změně H se nepohybujeme po kružnici, ale po obvodu pravidelného šestiúhelníka. Dále je prostor HSV nesymetrický vzhledem k jasu. Tyto dvě skutečnosti mohou ztěžovat práci s barvami. Tyto nedostatky odstraňuje prostor HLS.

V prostoru **HLS** je oproti HSV jehlan nahrazen dvojicí kuželů (viz obrázek 3). Základní složky jsou: *barevný tón* (H, „hue“), *světlost* (L, „lightness“) a *sytost* (S, „saturation“).



Obrázek 3: Dvojice kuželů HSL[5]

Sytost a barevný tón zde mají stejný význam jako v prostoru HSV, avšak nyní už tvoří klasickou polární soustavu souřadnic. Sytost má hodnotu 1 na povrchu kuželů a klesá směrem k ose na hodnotu 0. Světlost nabývá také hodnot $\langle 0, 1\rangle$, hodnota 0 je v dolním vrcholu (černá barva) a 1 v horním vrcholu (bílá barva). Nejjasnější čisté barvy leží na obvodu podstav, mají tedy souřadnice: $H \in \langle 0°, 360°)$, $S = 1$, $L = 0.5$. Prostor HSL nejlépe odpovídá lidskému vnímání barev, nejvíce barev vnímáme při průměrné světlosti, při zesvětlování nebo ztmavování klesá schopnost rozlišit barvy. Převod barev z prostorů HSV, HSL do RGB nebo naopak lze provádět pomocí algoritmu (viz [14]), více o barevných prostorech nalezneme v [23].

2.2 Detekce hran

Detekcí hran ve zpracování digitálního obrazu rozumíme zvýraznění hran uvnitř obrazu. Hrana v diskrétním obrazu je oblast, ve které dochází k výrazné změně sousedních pixelů. V tomto textu uvedeme tři základní filtry pro detekci hran: *Prewittové*, *Sobelův* a *Kirschův*. Všechny tři zmíněné filtry jsou založeny na konvoluci, uvedeme si tedy nejprve

[5]SharkD : Commons.wikimedia.org: File:HSL color solid dblcone.png [online]. [cit. 2016-05-03]. Dostupný pod licencí Creative Commons na www: <https://commons.wikimedia.org/wiki/File:HSL_color_solid_dblcone.png>

definici této operace. Jelikož dále budeme pracovat pouze s diskrétním obrazem, omezíme se pouze na diskrétní konvoluci funkcí dvou proměnných. Definici spojité 2D konvoluce najdeme v [5, 23].

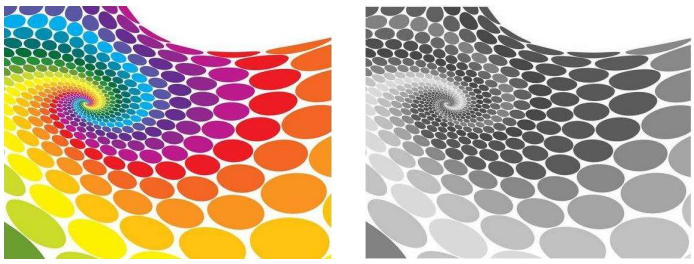
Definice 2.1. Diskrétní 2D konvoluce používaná při zpracování obrazu je definovaná vztahem:

$$(f * h)(n, m) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(n - i, m - j)h(i, j), \tag{2.1}$$

kde $f(n, m)$ je vstupní obraz dimenze $N \times M$ v odstínech šedi, tedy matice s hodnotami intenzit šedi, $(f * h)(n, m)$ je výstupní obraz opět dimenze $N \times M$. Funkci $h(i, j)$ dimenze $(2k + 1) \times (2k + 1)$ nazveme konvoluční jádro.

Konvoluční jádro bývá mnohem menší než vstupní obraz a můžeme ho popsat jako tabulku o rozměrech $\langle -k, k\rangle \times \langle -k, k\rangle$. Hodnoty konvolučního jádra určují jakým způsobem vypočítáme nový pixel v obraze.

Nyní můžeme přejít k jednotlivým výše zmíněným filtrům pro detekci hran. Tyto filtry se vzájemně liší právě konvolučním jádrem, konkrétní jádra budeme dále nazývat *operátory*, nebo *masky* a pro jejich popis použijeme matice. Efekty filtrů budeme demonstrovat na referenčním obrazu (Obrázek 4).



(a) Originál (b) Šedotónový

Obrázek 4: Referenční obraz

2.2.1 Filtry pro detekci hran

Nejjednodušším ze tří zmíněných filtrů je *Prewittové*. Tento filtr je definován operátorem o rozměrech 3×3 s reálnými hodnotami ve tvaru:

$$h_p = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}. \tag{2.2}$$

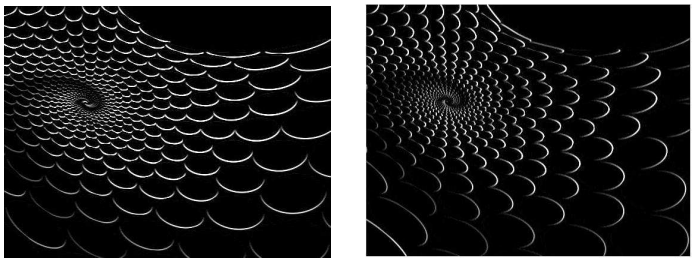
Tento operátor detekuje pouze hrany v horizontálním směru (obrázek 5a), transponováním matice získáme masku detekující hrany ve vertikálním směru (obrázek 5b):

$$\overline{h_p} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}. \tag{2.3}$$

Dalším typem, velice podobným s filtrem Prewittové, je *Sobelův* filtr. Jedná se opět o směrově orientovaný filtr. Maska detekující horizontální hrany má tvar

$$h_S \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \tag{2.4}$$

transponováním získáme opět masku detekující vertikální hrany.



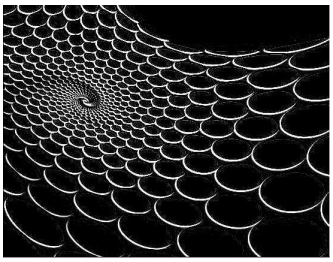
(a) Horizontální (b) Vertikální

Obrázek 5: Prewittové operátor

Posledním typem, který uvedeme, je *Kirschův* filtr. Maska tohoto filtru má tvar

$$h_k = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \tag{2.5}$$

a použitím této masky na vstupní obraz dochází ke zvýraznění hran pod úhlem 45° (obrázek 6).



Obrázek 6: Kirschův operátor

Operátory detekující hrany musí mít obecně nulový součet svých prvků, to zaručí, že v místech s konstantní intenzitou je odezva konvoluce nulová. Připomeňme také, že uvedené operátory lze použít pouze na obrázek ve stupních šedi. Obrázky 5 a 6 znázorňují efekt příslušné masky použité na referenční obrázek. Ve výstupní obrazové matici po použití masky se vyskytují záporné hodnoty, ty jsme z matice ořízli a vykreslili jsme pouze kladné hodnoty. Druhou možností jak lze tento problém vyřešit je použití absolutní hodnoty. Pro vytvoření kvalitního filtru pro detekci hran je však potřeba provést určité modifikace a další operace s obrazem, to však není obsahem této práce, více v [23, 19].

3 Využití kvaternionů pro detekci hran

V předešlé kapitole jsme si ukázali jakým způsobem lze detekovat hrany v případě obrazu v odstínech šedi. Nyní se budeme zabývat tím, jak lze předešlé filtry modifikovat pro barevný vstupní obraz. K tomuto účelu využijeme právě kvaterniony. Čerpat budeme především z [5, 19]

Nejprve je potřeba zavést modifikovanou obdobu konvoluce (2.1) pro kvaterniony. Existují tři různé definice kvaternionové konvoluce: levá, pravá a bi-konvoluce. Levá konvoluce je definována vztahem[6]

$$(h_L \circ f)(n,m) = \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} h_L(i,j) f(n-i, m-j), \tag{3.1}$$

pravá má tvar

$$(f \circ h_R)(n,m) = \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} f(n-i, m-j) h_R(i,j) \tag{3.2}$$

a bi-konvoluce

$$(h_L \prec f \succ h_R)(n,m) = \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} h_L(i,j) f(n-i, m-j) h_R(i,j), \tag{3.3}$$

kde podobně jako ve vztahu (2.1) f představuje vstupní obraz, h_L a h_R jsou levá a pravá konvoluční maska. Rozdíl oproti klasické konvoluci je v tom, že vstupní obraz je reprezentován pomocí kvaternionů a konvoluční masky budou také obsahovat místo reálných hodnot kvaterniony, jak si ukážeme dále.

Jak již bylo zmíněno, v klasických barevných prostorech dimenze tři, lze každý pixel znázornit barevným vektorem o třech složkách. Obraz tedy potom lze reprezentovat pomocí třísložkových ryzích kvaternionů. Při návrhu barevných konvolučních masek je potřeba, abychom po aplikaci konvoluce na vstupní obraz získali opět ryzí kvaterniony. Filtry, které si uvedeme, jsou založené na rotaci barevných vektorů v prostoru. Proto budeme pro filtrování obrazu využívat operaci bi-konvoluce (viz vztah (3.3)), čímž získáme kvaternionové operátory rotace, které nám zaručí, že výsledkem budou opět ryzí kvaterniony. Tuto myšlenku si vysvětlíme dále na konkrétních filtrech v příslušném konkrétním barevném prostoru.

3.1 Obraz v prostoru RGB

Připomeňme, že obraz v prostoru RGB můžeme reprezentovat maticí, jejíž prvky jsou barevné vektory představující jednotlivé pixely v obraze. Každý pixel obrazu lze tedy zapsat pomocí vektoru $\mathbf{v}_{nm} = (r_{nm}, g_{nm}, b_{nm})$, kde $r_{nm},\ g_{nm},\ b_{nm}$ jsou intenzity jednotlivých základních složek pixelu na pozici (n,m). Vektor $\mathbf{v}_{nm}$ můžeme reprezentovat pomocí

[6] Označení kvaternionové konvoluce ($\circ$) jsme použili pouze za účelem rozlišení od klasické konvoluce (viz vztah (2.1)).

ryzího kvaternionu tvaru $q = r_{nm}\boldsymbol{i} + g_{nm}\boldsymbol{j} + b_{nm}\boldsymbol{k}$. Libovolný pixel vstupního obrazu f (viz vztah (3.3)) v prostoru RGB lze tedy reprezentovat pomocí kvaternionů jako:

$$f(n,m) = r(n,m)\boldsymbol{i} + g(n,m)\boldsymbol{j} + b(n,m)\boldsymbol{k}, \tag{3.4}$$

kde $r(n,m)$ je jeho červená složka, $g(n,m)$ zelená a $b(n,m)$ modrá složka. Vstupní obraz bude v našem případě tedy matice, jejíž prvky budou ryzí kvaterniony[7] (pole ryzích kvaternionů).

RGB prostor může být reprezentován pomocí jednotkové krychle. V případě filtrů, které si uvedeme, dochází k rotaci barevných vektorů vstupního obrazu. Může se tedy stát, že nový vektor výstupního obrazu přesahuje hranici krychle, tudíž nový pixel leží mimo RGB krychli. Takový pixel je potřeba převést zpět do prostoru RGB, kde jsou definovány barvy. Pokud bychom se snažili tyto pixely posunout na povrch krychle se zachováním směru vektoru (tedy na místo, kde vektor protíná povrch krychle), mohlo by dojít k vizuálnímu zkreslení výstupního obrazu. Zmírnění tohoto zkreslení můžeme dosáhnout tím, že počátek ($O = (0,0,0)$, vrchol krychle představující černou barvu) posuneme na střed šedé osy (tj. střed úsečky spojující černý a bílý vrchol). Nyní pixel dříve reprezentovaný kvaternionem ve vztahu (3.4) bude tvaru:

$$f(n,m) = (r(n,m) - 1/2)\boldsymbol{i} + (g(n,m) - 1/2)\boldsymbol{j} + (b(n,m) - 1/2)\boldsymbol{k}. \tag{3.5}$$

Nyní tedy všechny barevné vektory směřují ze středu krychle a vzájemně doplňkové barvy (např. černá/bílá, červená/zelená, oranžová/modrá, ...) tvoří dvojice vektorů symetrických podle středu. Pokud vektor ve výstupním obraze v takto centrovaném prostoru přesahuje povrch krychle, můžeme při zachování směru zmenšit jeho velikost tak, že výsledný pixel bude představovat průsečík vektoru s povrchem krychle. V tomto případě už nedochází k barevnému zkreslení výstupního obrazu. Tento postup byl již publikován v [5, 1, 19].

3.1.1 Barevné filtry v RGB

Barevné filtry uvedené v této kapitole budeme aplikovat opět na vstupní obrázek 4a. Uvedeme si tři barevné filtry využívající kvaterniony, které jsou modifikacemi klasických filtrů Prewittové, Sobela a Krische. Budeme je pro jednoduchost označovat opět těmito názvy, ačkoli se ve skutečnosti jedná o nové filtry. Tyto filtry byly již publikovány v [5, 19].

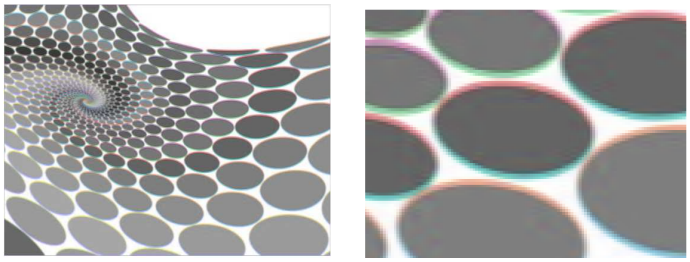
Prewittové filtr je definován pomocí dvou bi-konvolučních masek ve tvaru:

$$\alpha \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ q & q & q \end{pmatrix} (f) \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ \bar{q} & \bar{q} & \bar{q} \end{pmatrix}, \tag{3.6}$$

kde f představuje vstupní obraz, $\alpha = 1/6$ je koeficient, který kompenzuje hodnotu pixelu vzniklého sečtením šesti nenulových hodnot pixelů a $q = \exp(\boldsymbol{\mu}\pi/2)$ je jednotkový kvaternion (rotor). Vztah (3.6) je zjednodušený zápis pro bi-konvoluci (viz vztah (3.3)) s konkrétními maskami a nejedná se tedy o násobení matic. Při použití bi-konvoluce násobíme pro zisk nového pixelu vždy překrývající se prvky bi-konvolučních masek, pohybujících se po obraze, a vstupního obrazu. Ve spodním řádku bi-konvolučních masek tedy

[7] Obraz může být reprezentován i úplnými kvaterniony. Zavedení skalární složky pro reprezentaci obrazu se věnuje například článek [1].

zřejmě získáme operátor rotace $\mathcal{R}_q(\{f\}) = q\{f\}\bar{q}$, který rotuje příslušný pixel (barevný vektor) o úhel π kolem osy $\boldsymbol{\mu}$. Za osu rotace se nejčastěji volí „šedá osa“, pro kterou platí $r = g = b$ a tedy $\boldsymbol{\mu} = (\boldsymbol{i} + \boldsymbol{j} + \boldsymbol{k})/\sqrt{3}$, možné jsou však samozřejmě i jiné varianty. Masky definované vztahem (3.6) detekují horizontální hrany, stejně jako v případě klasického filtru Prewittové jejich transponováním získáme masky detekující vertikální hrany. Výsledek, který získáme aplikováním filtru podle vztahu (3.6) na barevný obraz, je znázorněn na obrázku 7.



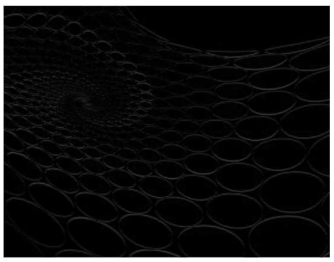
(a) Filtrovaný obraz (b) Barevné hrany

Obrázek 7: Kvaternionový Prewittové filtr

Z obrázku 7a je patrné, že kvaternionový Prewittové filtr potlačuje „barevnost“ vstupního obrazu. Pokud při bi-konvoluci působí masky ze vztahu (3.6) na oblast vstupního obrazu, kde se výrazně nemění barva, nevyskytuje se zde tedy hrana a hodnoty barevných vektorů vstupního obrazu překrývající se s horním řádkem masek a hodnoty vektorů překrývající se se spodním řádkem masek jsou podobné nebo stejné. Potom jsou vektory v horním řádku násobeny hodnotou 1, zůstanou tedy nezměněny. Vektory ve spodním řádku jsou rotovány o úhel π kolem šedé osy a vznikají tak vektory symetrické podle této osy. Výsledný pixel po sečtení všech těchto vektorů a vynásobení koeficientem α bude tedy ležet na šedé ose (nebo v její blízkosti). Naopak v místě, kde se vyskytuje ve vstupním obraze horizontální hrana, výsledný pixel vzniklý sečtením orotovaných vektorů na příslušných místech a vynásobením koeficientem α nebude ležet (až na výjimečné situace) poblíž šedé osy a má tedy nějakou barvu. Skutečně, když se podíváme na výsledný obraz (obrázek 7a), vidíme, že původní barevný obrázek je po aplikování filtru v odstínech šedi všude, kde se nevyskytují horizontální hrany. Tam, kde se v původním obrazu vyskytovaly horizontální hrany, tam se nyní objevily barevné složky. Na obrázku 7b vidíme tyto barevné hrany. Na výstupní obraz můžeme ještě pro vizuální znázornění hran aplikovat jednoduchou prahovou funkci ve tvaru:

$$C_p = \frac{r + g + b}{3} - \min(r, g, b), \tag{3.7}$$

kde r, g, b jsou intenzity jednotlivých barevných složek výstupního obrazu. Tato funkce izoluje barevné hrany od „nebarevných“ pixelů. Vykreslením této funkce (viz obrázek 8) získáváme výsledek, který je velice podobný výsledku získanému použitím klasického Prewittové filtru na obraz v odstínech šedi bez dalších úprav.



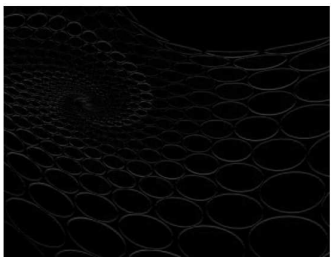
Obrázek 8: Výsledek prahové funkce

Další filtry pracují na stejném principu jako filtr Prewittové, liší se akorát tvarem bi-konvolučních masek a koeficientem α. Uvedeme si tedy pouze tvary příslušných masek a výsledky získané jejich aplikováním.

Sobelův filtr pro detekci horizontálních hran je definován dvojicí masek tvaru:

$$\frac{1}{8}\begin{pmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ q & \sqrt{2}q & q\end{pmatrix}(f)\begin{pmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ \bar{q} & \sqrt{2}\bar{q} & \bar{q}\end{pmatrix}. \tag{3.8}$$

Výsledek aplikování Sobelova filtru je téměř stejný jako v případě filtru Prewittové.



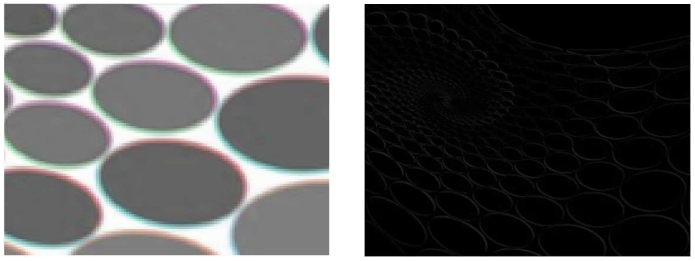
Obrázek 9: Prahová funkce Sobelova filtru

Na obrázku 9 je výsledek prahové funkce aplikované na výstupní obraz filtrovaný Sobelovým filtrem. Hodnoty této funkce nejsou stejné jako v případě Prewittové filtru, intenzita hran Sobelova filtru je sice u některých pixelů vyšší než u Prewittové, ovšem rozdíly jsou v řádech jednotek v celočíselném vyjádření $0 - 255$, můžeme tedy výsledky těchto filtrů považovat téměř za totožné.

Posledním typem je *Kirschův* filtr. Masky tohoto filtru mají tvar:

$$\frac{1}{30}\begin{pmatrix}\sqrt{3}q & \sqrt{5} & \sqrt{5}\\ \sqrt{3}q & 0 & \sqrt{5}\\ \sqrt{3}q & \sqrt{3}q & \sqrt{3}q\end{pmatrix}(f)\begin{pmatrix}\sqrt{3}\bar{q} & \sqrt{5} & \sqrt{5}\\ \sqrt{3}\bar{q} & 0 & \sqrt{5}\\ \sqrt{3}\bar{q} & \sqrt{3}\bar{q} & \sqrt{3}\bar{q}\end{pmatrix}. \tag{3.9}$$

Stejně jak tomu bylo u klasického Kirschova filtru, tyto masky detekují hrany pod úhlem 45° (viz obrázek 10).



(a) Barevné hrany (b) Prahová funkce

Obrázek 10: Kvaternionový Kirschův filtr

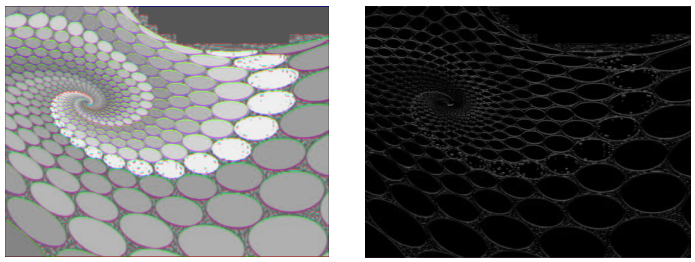
3.2 Obraz v prostoru HSV

V této kapitole se budeme zabývat detekcí hran obrazu zadaného v barevném prostoru HSV. Budeme opět aplikovat již zmíněné barevné filtry pro detekci hran, konkrétně Prewittové, Sobelův a Kirschův, definované v tomto pořadí vztahy (3.6), (3.8) a (3.9). Referenční obraz v tomto případě bude reprezentován maticí, jejíž prvky budou barevné vektory o složkách H, S, V. Pixel na pozici (n, m) tedy reprezentuje vektor $\mathbf{v_{nm}} = (H_{nm}, S_{nm}, V_{nm})$, kde H_{nm} je úhel vyjadřující barevný tón, S_{nm} je sytost nebo saturace a V_{nm} je jas (viz kapitola 2.1.3). Referenčním obrazem v této kapitole bude opět obrázek 4a ovšem převedený do reprezentace HSV, v Matlabu pomocí funkce `rgb2hsv()`. Matematický popis algoritmu pro převod obrazu z RGB reprezentace do HSV lze najít v [14]. Abychom mohli detekovat v takovém obrazu hrany pomocí výše uvedených filtrů, je potřeba opět reprezentovat každý barevný vektor pomocí kvaternionů. V následujících odstavcích si ukážeme několik různých přístupů. Inspirací pro následující úvahy byl především článek [3].

Zvolme nejprve stejný přístup jako v případě RGB referenčního obrazu. Nechť tedy referenční obraz je reprezentován pomocí kvaternionů ve tvaru:

$$f(n, m) = H(n, m)\boldsymbol{i} + S(n, m)\boldsymbol{j} + V(n, m)\boldsymbol{k}, \tag{3.10}$$

kde $f(n, m)$ je kvaternion na pozici (n, m), $n = 1, \ldots, N$ a $m = 1, \ldots, M$ ($N \times M$ je rozměr obrazové matice), $H(n, m), S(n, m), V(n, m)$ jsou hodnoty jednotlivých složek na pozici (n, m). Tuto matici kvaternionů dosadíme do vztahu pro filtr Prewittové (3.6) a užitím bi-konvoluce získáme výsledný obraz (viz obrázek 11). Kvaternion q v bi-konvoluční masce volíme stejný jako v případě detekce RGB, tedy $q = \exp(\boldsymbol{\mu}\pi/2)$ a $\boldsymbol{\mu} = (\boldsymbol{i} + \boldsymbol{j} + \boldsymbol{k})/\sqrt{3}$. Ve výstupním obraze (obrázek 11a) jsou viditelné barevné hrany. V místech, kde se hrany



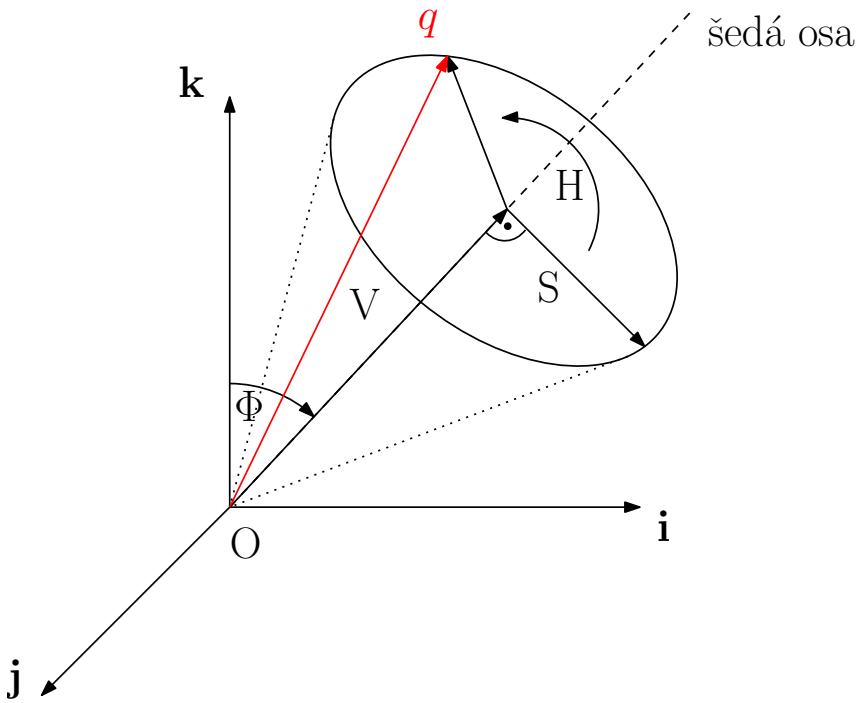
(a) Filtrovaný obraz (b) Prahová funkce

Obrázek 11: RGB přístup reprezentace obrazu

v referenčním obraze nevyskytují, nabývá výstupní obraz odstínů šedi, tedy podobně jako při filtrování RGB obrazu. K detekci hran tedy zřejmě dochází, ovšem jak lze vidět na obrázku 11b, filtr detekoval jako hrany i značné množství šumu. Tento obrázek zobrazuje barevné pixely výstupního obrazu pomocí prahové funkce (3.7). Výstupní obraz už je v prostoru RGB, to nám umožňuje použít funkci přesně v tomto tvaru. Šum je způsoben tím, že tento přístup nezohledňuje jiný tvar HSV modelu, ale k barevnému prostoru, ve kterém je definován referenční obraz, přistupujeme opět jako ke krychli. Tento přístup tedy není nejvhodnější.

Dále si uvedeme další dva přístupy, které se zdají být pro detekci hran obrazu reprezentovaného složkami H, S, V vhodnější a budeme k nim přikládat pro lepší pochopení principů úseky ze zdrojového kódu napsaného v Matlabu.

V kapitole 2.1.3 jsme uvedli, že základní geometrická reprezentace HSV prostoru je jehlan. Dále budeme za geometrickou reprezentaci HSV prostoru považovat kužel, což je v dnešních grafických programech častější případ. Budeme vycházet z obrázku 12, který zobrazuje polohu HSV kuželu vzhledem k pravoúhlým souřadnicím daných kvaterniony $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$. Je zřejmé, že černá barva se nachází ve vrcholu kuželu, což je v souladu s definicí



Obrázek 12: Reprezentace kvaternionu q

HSV prostoru. Osa kuželu (šedá osa) je totožná s šedou osou prostoru RGB, tedy je reprezentována kvaternionem $\boldsymbol{\mu} = (\boldsymbol{i} + \boldsymbol{j} + \boldsymbol{k})/\sqrt{3}$ a bude opět při filtrování osou rotace. Jelikož při použití filtrů dojde, jak je zřejmé z obrázku, k potlačení saturace S v místech, kde se nevyskytují hrany, to znamená, že na těchto pozicích bude hodnota barevného tónu H nulová, nebudeme se barevným tónem nyní vůbec zabývat. Budeme tedy pracovat pouze se složkami V a S. Libovolný vektor $\mathbf{v} = (0, S, V)$ budeme tedy reprezentovat pomocí kvaternionu

$$r = S\boldsymbol{\nu} + V\boldsymbol{\mu}, \tag{3.11}$$

kde $\boldsymbol{\mu}$ je jednotkový kvaternion představující osu kuželu a $\boldsymbol{\nu}$ jednotkový kvaternion představující osu saturace S a je tedy ortogonální k $\boldsymbol{\mu}$. Kvaternion $\boldsymbol{\mu}$ získáme v Matlabu snadno.

```
mu=unit(quaternion(1,1,1));
```

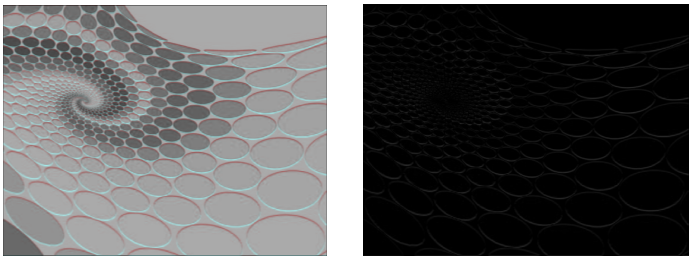
Všimněme si na obrázku 12, že $\boldsymbol{\mu}$ je rotací kvaternionu $\boldsymbol{k}$ podle nějaké konkrétní osy o úhel Φ. Kvaternion $\boldsymbol{\nu}$ tedy získáme stejnou rotací kvaternionu $\boldsymbol{i}$. Osa rotace i úhel Φ lze získat snadno pomocí goniometrických funkcí. Pro úhel Φ použijeme v Matlabu značení *fi* a kvaternion $\boldsymbol{\nu}$ je *nu*.

```
fi=acos(1/sqrt(3));
nu=(exp(unit(qj-qi)*(fi/2))*qi*exp(-unit(qj-qi)*(fi/2)));
```

Nyní už můžeme převést referenční obraz do kvaternionové reprezentace podle vztahu (3.11) a aplikovat příslušný filtr.

```
[n,m,l]=size(f);
f_q=nu*f(:,:,2)+mu*f(:,:,3);
q=mu;
q_conj=conj(q);
hxl=[1 1 1;0 0 0;q q q]; %leva bi-konvolucni maska
hxr=[1 1 1;0 0 0;q_conj q_conj q_conj]; %prava bi-konvolucni maska
bw_qx=conv2({1/6*hxl,hxr},f_q); %bi-konvoluce
bw_qx=bw_qx(2:n+1,2:m+1,:);   %odmazani krajnich pixelu (zpusobuje conv2)
bwx(:,:,1)=zeros(n,m);
bwx(:,:,2)=scalar(bw_qx(:,:)/nu);  %koeficienty u kvaternionu nu (S)
bwx(:,:,3)=scalar(bw_qx(:,:)/mu);  %koeficientz u kvaternionu mu (V)
bwx=hsv2rgb(bwx);   %prevedeni do RGB pro vykresleni
imshow(bwx); %vysledny obrazek
```

Ve zdrojovém kódu představuje f referenční obrázek v HSV reprezentaci, tedy matici, jejíž prvky jsou barevné vektory o složkách H, S, V. Všimněme si, že ve výsledném obrazu (obrázek 13) nabývají hrany dvou barev. Jedná se o barvy se stejnými hodnotami saturace S a hodnotami barevného tónu $H_1 = 0$, $H_2 = \pi$ v radiánech. U tohoto přístupu



(a) Filtrovaný obraz (b) Prahová funkce

Obrázek 13: „SV“ přístup reprezentace obrazu

už se ve výsledném obrazu nevyskytuje žádný šum. Z důvodu vynechání jedné složky referenčního obrazu však stále nemáme kvaternionovou reprezentaci referenčního obrazu, ale pracujeme s obrazem, který nabývá jiných barev. Uvedeme si tedy poslední přístup,

ve kterém už budeme používat všechny složky referenčního obrazu.

Na obrázku 12 je červenou barvou znázorněn ryzí kvaternion q (barevný vektor), který reprezentuje nějaký pixel referenčního obrazu. Tento kvaternion vyjádříme v kartézském tvaru podobně jako ve vztahu (3.11) s tím rozdílem, že do vztahu zakomponujeme i hodnotu barevného tónu H. Stejným způsobem jako v předchozím přístupu tedy sestrojíme osy jasu a saturace reprezentované kvaterniony $\boldsymbol{\mu}$ a $\boldsymbol{\nu}$ a na ně budeme nanášet hodnoty V, S podle vztahu (3.11). Abychom získali ryzí kvaternion odpovídající vektoru se složkami (H, S, V), budeme ještě kvaternion $S\boldsymbol{\nu}$ rotovat kolem šedé osy $\boldsymbol{\mu}$ o úhel H. Označme tuto rotaci pomocí kvaternionového operátoru rotace $\mathcal{R}_{\boldsymbol{\mu}_H}(\cdot)$ (viz věta 1.11). Zápis

$$p = \mathcal{R}_{\boldsymbol{\mu}_H}(S\boldsymbol{\nu})$$

tedy vyjadřuje skutečnost, že kvaternion p je výsledkem rotace kvaternionu $S\boldsymbol{\nu}$ kolem osy $\boldsymbol{\mu}$ o úhel H. Nyní tedy libovolný barevný vektor $\mathbf{v} = (H, S, V)$ budeme reprezentovat kvaternionem ve tvaru

$$r = V\boldsymbol{\mu} + \mathcal{R}_{\boldsymbol{\mu}_H}(S\boldsymbol{\nu}). \tag{3.12}$$

Při zadávání do Matlabu je však ještě potřeba brát ohled na to, jakým způsobem zadáváme úhel rotace H. Jak bylo uvedeno v úvodu kapitoly, HSV reprezentaci obrazu získáme v Matlabu následujícím způsobem.

```
f=rgb2hsv(A);
```

A je obraz v prostoru RGB a f je referenční obraz v prostoru HSV. V Matlabu jsou všechny složky tohoto obrazu H, S, V normalizovány, to znamená, že jsou to reálná čísla (datový typ *double*) ležící v intervalu $\langle 0, 1\rangle$, tedy $H \times S \times V \in \langle 0, 1\rangle \times \langle 0, 1\rangle \times \langle 0, 1\rangle$. V případě složek S, V nás to nijak neomezuje. Pro rotaci v Matlabu využijeme zápisu kvaternionu v polárním tvaru, kde musíme úhel rotace zadávat v radiánech, musíme tedy úhel H převést na radiány. Hodnota $H = 1$ odpovídá úhlu 2π rad, označme nyní zadané hodnoty úhlů $\hat{H}$ a nové hodnoty v radiánech označme H. Pro převod tedy dostáváme:

$$H = 2\pi\hat{H}.$$

```
H=2*pi*f(:,:,1);
```

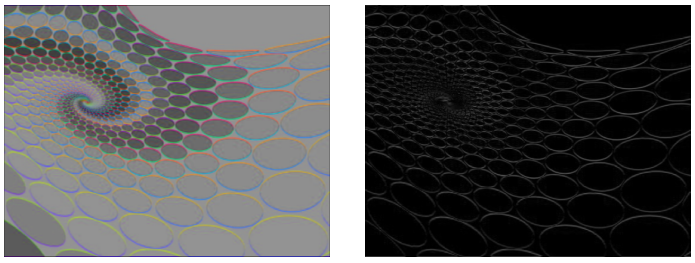
Než ukážeme zbytek kódu s vykreslením výsledného obrazu, všimněme si ještě, že v předchozí metodě, reprezentované vztahem (3.11), jsme ve zdrojovém kódu pracovali s kvaterniony $\boldsymbol{\mu}$ a $\boldsymbol{\nu}$ jako s novými základními jednotkami. To znamená, že výsledný obrázek jsme získali pomocí koeficientů příslušných těmto jednotkám a tím jsme získali HSV obraz, který jsme museli pro vykreslení ještě převést do RGB reprezentace. Nyní však budeme pracovat se základními jednotkami $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ a příslušné koeficienty budou představovat složky R, G, B (viz vztah (3.4)). Rozepíšeme-li tedy vztah (3.12) tak, aby obsahoval základní jednotky $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$, dostaneme výslednou reprezentaci libovolného vektoru $\mathbf{v} = (H, S, V)$ pomocí kvaternionu:

$$r = V\left(\frac{\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}}{\sqrt{3}}\right) + e^{\left(\frac{\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}}{\sqrt{3}}\frac{H}{2}\right)}\left[S\left(e^{\left(\frac{\boldsymbol{j}-\boldsymbol{i}}{\sqrt{2}}\frac{\Phi}{2}\right)}(\boldsymbol{i})e^{\left(-\frac{\boldsymbol{j}-\boldsymbol{i}}{\sqrt{2}}\frac{\Phi}{2}\right)}\right)\right]e^{\left(-\frac{\boldsymbol{i}+\boldsymbol{j}+\boldsymbol{k}}{\sqrt{3}}\frac{H}{2}\right)}, \tag{3.13}$$

kde $\Phi = \arccos(1/\sqrt{3})$ je úhel mezi osou $\boldsymbol{\mu}$ a $\boldsymbol{k}$ (viz obrázek 12). Zdrojový kód pro tuto reprezentaci a vykreslení obrázku vypadá tedy následovně.

```
[n,m,l]=size(f);
mu=unit(quaternion(1,1,1));
fi=acos(1/sqrt(3));
nu=(exp(unit(qj-qi)*(fi/2))*qi*exp(-unit(qj-qi)*(fi/2)));
q=mu;
q_conj=conj(q);
f_q1=f(:,:,3)*mu;   %V*mu
f_q25=f(:,:,2)*nu;        %S*nu
f_q2=exp(mu.*(H(:,:)/2)).*f_q25(:,:).*exp(-mu.*(H(:,:)/2)); %rot o uhel H
f_q=f_q1+f_q2; %vysledny kvaternion zadany souradnicemi HSV
hxl=[1 1 1;0 0 0;q q q]; %leva bi-konvolucni maska
hxr=[1 1 1;0 0 0;q_conj q_conj q_conj]; %prava bi-konvolucni maska
bw_qx=conv2({1/6*hxl,hxr},f_q); %bi-konvoluce
bw_qx=bw_qx(2:n+1,2:m+1,:);  %odmazani krajnich pixelu (zpusobuje conv2)
bwx(:,:,1)=x(bw_qx); %koeficienty u kvaternionu i (R)
bwx(:,:,2)=y(bw_qx); %koeficienty u kvaternionu j (G)
bwx(:,:,3)=z(bw_qx); %koeficienty u kvaternionu k (B)
imshow(bwx); %vysledny obrazek
```

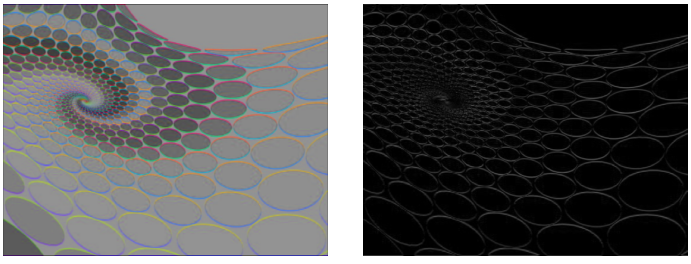
Obrázek 14 znázorňuje výsledek tohoto přístupu při použití filtru Prewittové, opět včetně prahové funkce. Je patrné, že dochází k detekci horizontálních hran, stejně jako při pou-



(a) Filtrovaný obraz (b) Prahová funkce

Obrázek 14: HSV Prewittové filtr

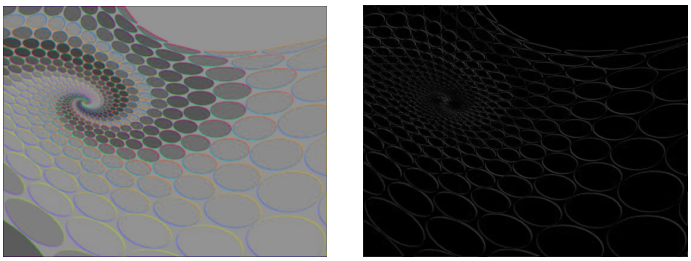
žití filtru na referenční obrázek v RGB reprezentaci. Uvedeme ještě výsledky při použití ostatních filtrů stejným způsobem, tedy Sobelův (3.8) a Kirschův (3.9), včetně prahových funkcí. Sobelův filtr (obrázek 15) vykazuje opět téměř stejné výsledky jako filtr



(a) Filtrovaný obraz (b) Prahová funkce

Obrázek 15: HSV Sobelův filtr

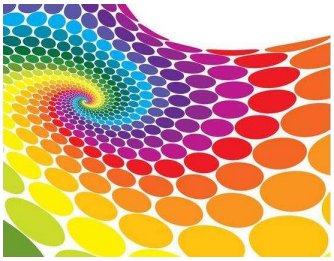
Prewittové. Oproti tomu v případě Kirschova filtru (obrázek 16) je z obrázku ihned patrná detekce šikmých hran.



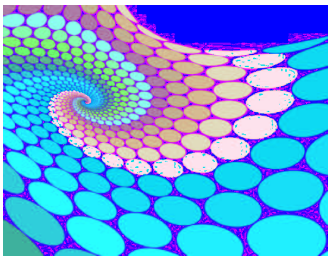
(a) Filtrovaný obraz (b) Prahová funkce

Obrázek 16: HSV Kirschův filtr

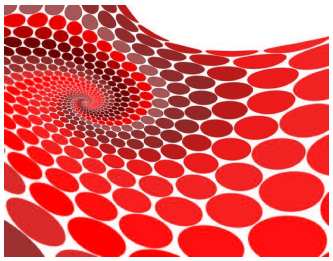
Ukázali jsme si tři různé způsoby reprezentace HSV obrazu pomocí kvaternionů a jejich detekci hran. Na závěr ještě porovnejme jednotlivé postupy. Pro porovnání vykreslíme ještě obrazy po zavedení příslušné reprezentace kvaterniony bez dalších úprav (viz obrázek 17). Z výsledných filtrovaných obrázků (berme v úvahu použití filtru Prewittové) můžeme



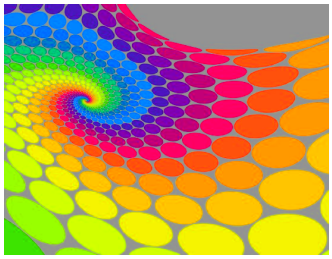
(a) Referenční obraz



(b) HSV/RGB



(c) SV



(d) HSV

Obrázek 17: Kvaternionová reprezentace HSV obrazu

usuzovat, že první přístup, nazvěme ho *HSV/RGB reprezentace* (vztah (3.10) a obrázek 11), je pro reprezentaci HSV obrazu pomocí kvaternionů nevhodný. U filtrovaného obrazu se vyskytuje nežádoucí šum, obraz reprezentace barevně neodpovídá referenčnímu obrazu a obsahuje barevné poruchy. Druhý přístup, nazveme *SV reprezentace* (vztah (3.11) a obrázek 13), má uspokojivější výsledky při detekci hran, dochází ke zvýraznění hran správným způsobem. Obraz této reprezentace, z důvodu vynechání složky barevného tónu, obsahuje pouze odstíny jedné barvy. To znamená, že ztrácíme hlavní výhodu kvaternionových filtrů, kterou je možnost pracovat s plně barevným obrazem. To umožňuje poslední uvedený přístup, nazvěme ho *HSV reprezentace* (vztah (3.13) a obrázek 14). U tohoto způsobu reprezentace dochází k nejvýraznější detekci hran (porovnáváme pomocí prahových funkcí). Obraz reprezentace se barevně nejvíce přibližuje referenčnímu obrazu. Přesto si můžeme však všimnout barevných odchylek, například šedá barva pozadí místo bílé. To je způsobeno tím, že při takto zavedené reprezentaci neleží střed podstavy HSV

kuželu ve vrcholu RGB krychle vyjadřujícího bílou barvu, ale v místě, kde nabývá osa $\boldsymbol{\mu}$ ještě šedou barvu (viz obrázek 12). Přestože se stále nejedná o přesnou reprezentaci barevných vektorů zadaných složkami H, S, V, považujeme tento přístup za nejlepší z výše uvedených, protože dosažené výsledky se nejvíce přibližují našim cílům.

Závěr

V této práci jsme se zabývali využitím kvaternionů pro detekci hran obrazu s ohledem na různé barevné prostory.

První část jsme věnovali zavedení kvaternionů a zabývali jsme se jejich vlastnostmi. Ukázali jsme, že množina kvaternionů s operacemi sčítání a násobení tvoří nekomutativní těleso. Dále jsme uvedli další dvě možné reprezentace kvaternionů, pomocí matic a pomocí polárního tvaru. Polární tvar jsme využili při realizaci rotací kvaterniony. V závěru této části práce jsme představili základní funkce balíčku QTFM pro program Matlab, který umožňuje práci s kvaterniony.

Druhá část práce je věnována především teorii barevných prostorů. Popsali jsme některé používané barevné prostory. Konkrétní barevný prostor definuje, jakým způsobem je určena každá barva obrazu. Kromě jednoznačně nejpoužívanějších prostorů RGB a CMY, které lze geometricky znázornit pomocí krychle, jsme uvedli také prostory HSV respektive HSL, které jsou zobrazovány jako jehlan respektive dvojice kuželů. Dále jsme se zabývali detekcí hran obrazu, uvedli jsme tři typy detekčních filtrů: *Prewittové*, *Sobelův* a *Kirschův*. Těmito filtry jsme s využitím konvoluce provedli detekci hran obrazu v odstínech šedi.

V poslední části této práce jsme se zabývali detekcí hran barevného obrazu s využitím kvaternionů. Bylo potřeba definovat operaci bi-konvoluce a modifikace předchozích detekčních filtrů pro použití kvaternionů. Tyto filtry pracují na základě rotace barevných vektorů, které je možné reprezentovat pomocí kvaternionů. Nejprve jsme ukázali jakým způsobem lze reprezentovat tyto vektory a tím i celý obraz v barevném prostoru RGB. V závěru práce jsme se zabývali kvaternionovou reprezentací a následnou detekcí hran obrazu v prostoru HSV. Podařilo se nám zavést tři způsoby reprezentace HSV obrazu pomocí kvaternionů. Pro každou reprezentaci jsme provedli detekci hran a na základě získaných výsledků jsme tyto reprezentace porovnali. Prováděné detekce hran jsme demonstrovali obrázky získanými za použití programu Matlab a některé postupy v závěru práce jsme doplnili okomentovanými úseky zdrojového kódu.

Pro podrobnější analýzu získaných výsledků je potřeba v budoucnu provést další zkoumání uvedených postupů. Práci lze rozšířit také o další přístupy pro zpracování obrazu, například využití kvaternionové Fourierovy transformace. Také je možné zabývat se vylepšením uvedené reprezentace HSV obrazu pomocí kvaternionů.

Literatura

[1] ANGULO, J.: Morphological colour operators in totally ordered lattices based on distances. *Computer Vision and Image Understanding*, ročník 107, č. 1, 2007 [cit. 2016-04-03]: s. 56–73, [online].
URL <http://www.sciencedirect.com/science/article/pii/S1077314206002165>

[2] CONWAY, J. H.; SMITH, D. A.: *On quaternions and octonions*. Natick, Mass.: AK Peters, 2003, ISBN 1-56881-134-9.

[3] DENIS, P.; CARRE, P.; FERNANDEZ-MALOIGNE, C.: Spatial and spectral quaternionic approaches for colour images. *Computer Vision and Image Understanding*, ročník 107, č. 1-2, 2007 [cit. 2016-05-19]: s. 74–87, doi:10.1016/j.cviu.2006.11.019, [online].
URL <http://linkinghub.elsevier.com/retrieve/pii/S1077314206002177>

[4] DORST, L.; FONTIJNE, D.; MANN, S.: *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann, 2010.

[5] ELL, T. A.; Le BIHAN, N.; SANGWINE, S. J.: *Quaternion fourier transforms for signal and image processing*. Hoboken, New Jersey: Wiley, 2014, ISBN 978-1-118-93091-5.

[6] Gimbal lock. *Wikipedia: the free encyclopedia*. [online], 2001- [cit. 2016-04-03].
URL <https://en.wikipedia.org/wiki/Gimbal_lock>

[7] HRDINA, J.: Algebry rotací a jejich aplikace. *Kvaternion*, ročník 3, č. 2, 2014 [cit. 2016-04-03]: s. 43–62, [online].
URL <http://kvaternion.fme.vutbr.cz/2014/kvat6_separaty/kv14_2_hrdina.pdf>

[8] KARÁSEK, J.; SKULA, L.: *Lineární algebra*. Brno: Akademické nakladatelství CERM, první vydání, 2005, ISBN 80-214-3100-8.

[9] KOCH, M.; DAM, E. B.; LILLHOLM, M.: Quaternions, Interpolation and Animation. Technical report, University of Copenhagen, Department of Computer Science, Denmark, 17. 7. 1998 [cit. 2016-02-21], [online].
URL <http://web.mit.edu/2.998/www/QuaternionReport1.pdf>

[10] KUIPERS, J. B.: *Quaternions and rotation sequences*. Oxford: Princeton University Press, 1999, ISBN 0-691-10298-8.

[11] MOTL, L.; ZAHRADNÍK, M.: *Pěstujeme lineární algebru*. Praha: Univerzita Karlova v Praze, nakladatelství Karolinum, 2002.

[12] OOSTEN, J.: Understanding Quaternions. *3D Game Engine Programming*, 2012 [cit. 2016-02-21], [online].
URL <http://www.3dgep.com/understanding-quaternions/>

[13] PERWASS, C.: *Geometric Algebra with Applications in Engineering (Geometry and Computing)*. Springer, 2009.

[14] PM, N.; CHEZIAN, R. M.: VARIOUS COLOUR SPACES AND COLOUR SPACE CONVERSION ALGORITHMS. *Journal of Global Research in Computer Science*, ročník 4, č. 1, 2013 [cit. 2016-04-03]: s. 44–48, [online].
URL <http://www.jgrcs.info/index.php/jgrcs/article/view/587>

[15] PROŠKOVÁ, J.: *Kvaterniony a jejich užití v geometrii*. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra matematiky, Plzeň, 2006.

[16] Quaternion. *Wikipedia: the free encyclopedia*. [online], 2001- [cit. 2016-02-21].
URL <https://en.wikipedia.org/wiki/Quaternion>

[17] Rotation matrix. *Wikipedia: the free encyclopedia*. [online], 2001- [cit. 2016-04-03].
URL <https://en.wikipedia.org/wiki/Rotation_matrix>

[18] SANGWINE, S. J.; BIHAN, N. L.: Quaternion Polar Representation with a Complex Modulus and Complex Argument Inspired by the Cayley-Dickson Form. *Advances in Applied Clifford Algebras*, ročník 20, č. 1, 2010 [cit. 2016-04-03]: s. 111–120, [online].
URL <http://link.springer.com/article/10.1007%2Fs00006-008-0128-1>

[19] SANGWINE, S. J.; ELL, T. A.: Colour image filters based on hypercomplex convolution. *IEE Proceedings - Vision, Image, and Signal Processing*, ročník 147, č. 2, 2000 [cit. 2016-04-03]: s. 89–93, [online].
URL <http://digital-library.theiet.org/content/journals/10.1049/ip-vis_20000211>

[20] SANGWINE, S. J.; Le BIHAN, N.: Quaternion Toolbox for Matlab®, Version 2 with support for Octonions. [online], 2013 [cit. 2016-05-18].
URL <http://qtfm.sourceforge.net/>

[21] SLABAUGH, G. G.: Computing Euler angles from a rotation matrix. [online], [cit. 2016-04-03].
URL <http://www.staff.city.ac.uk/~sbbh653/publications/euler.pdf>

[22] ZHANG, F.: Quaternions and matrices of quaternions. *Linear Algebra and its Applications*, ročník 251, 1997 [cit. 2016-04-03]: s. 21–57, [online].
URL <http://www.sciencedirect.com/science/article/pii/0024379595005439>

[23] ŽÁRA, J.; BENEŠ, B.; SOCHOR, J.; aj.: *Moderní počítačová grafika*. Brno: Computer Press, druhé vydání, 2004, ISBN 80-251-0454-0.