ACTA MATHEMATICA SPALATENSIA
Series didactica
Vol.7 (2024) 1–18
https://doi.org/10.32817/amssd.7.1

Zaprimljen: 23. rujna 2023.
Prihvaćen: 20. siječnja 2024.

Učinkovitost prostorne rotacije pomoću kvaterniona

Karlo Bratko, Aleksandar Hatzivelkos

Sažetak

Iako kvaternioni imaju široku upotrebu u jezgrama kompleksnih grafičkih sustava i alata, robotici, avijaciji i drugim znanostima u kojima su brzina i sigurnost rotacija od velike važnosti, njihova je pojava u nastavnim materijalima i matematičkim (pa i računarskim) predmetima rijetka, ne samo u srednjem već i u visokom školstvu. Cilj je ovoga rada stoga predstaviti algebarsku strukturu kvaterniona te opisati njihovu upotrebu u provođenju rotacija u trodimenzionalnom prostoru. Pored toga, u članku opisujemo i druge dvije standardne metode za 3D rotaciju: Eulerove rotacije i neposrednu rotaciju oko proizvoljne osi. Konačno, u zadnjem dijelu članka predstavljamo aplikaciju izrađenu kako bi se testirala učinkovitost navedene tri metode. Učinkovitost testiramo mjerenjem vremena potrebnog za provođenje rotacija te analizom broja asemblerskih naredbi i zauzeća memorije.

Ključni pojmovi: kvaternioni, rotacija u prostoru, Eulerove rotacije, učinkovitost

Abstract

Although quaternions are widely used in the cores of complex graphic systems and tools, robotics, aviation and other sciences in which the speed and safety of rotations are of great importance, their appearance in teaching materials and mathematical (and computer) subjects is rare, not only in secondary but also in higher education. The aim of this paper is therefore to present the algebraic structure of quaternions, and to describe their use in performing rotations in three-dimensional space. In addition, in the article we describe two other standard methods for 3D rotation: Euler rotations and immediate rotation around an arbitrary axis. Finally, in the last part of the article, we present an application created to test the efficiency of the mentioned three methods. We test the efficiency by measuring the time required to perform rotations, and by analyzing the number of assembler commands and memory usage.

Keywords: quaternions, rotation in space, Euler rotations, efficiency

Članak je nastao na temelju završnog rada studenta Karla Bratka na Visokom učilištu Algebra, pod vodstvom mentora Aleksandra Hatzivelkosa.

1. Uvod

Kvaternioni su elementi nekomutativne algebre s dijeljenjem koju je u prvoj polovici devetnaestog stoljeća definirao i razvio irski matematičar i fizičar William Rowan Hamilton. Danas tu algebarsku strukturu, u počast njezinom tvorcu, označavamo s $\mathbb{H}$.

Hamilton je izgradnji nove algebarske strukture pristupio s idejom poopćenja strukture kompleksnih brojeva. Osnovna motivacija bila mu je definirati algebarsku strukturu u kojoj će se množenjem elemenata te strukture realizirati rotacija u prostoru – analogno rotaciji u ravnini koja se realizira kroz množenje kompleksnim brojem $e^{i\varphi}$.

Prvi važan korak koji je Hamilton napravio jest da je kompleksne brojeve identificirao kao uređene parove realnih brojeva te opisao računske operacije s kompleksnim brojevima kao računske operacije s uređenim parovima. Na taj se način s jedne strane riješio problema objašnjavanja što je imaginarna jedinica, a s druge strane otvorio prostor za poopćenje tog koncepta – ako možemo promatrati skup uređenih parova te na tom skupu definirati operacije koje se mogu interpretirati u geometriji ravnine, tada je za očekivati da se sličan koncept može upotrijebiti i za opisivanje geometrije prostora[1].

To uvjerenje nalazimo i u pismu koje Hamilton 1841. piše De Morganu: „...ako je moje shvaćanje Algebre ispravno, *mora* postojati mogućnost za uvođenje ne samo trojki, već i n-torki, kako bismo na neki način opisali simboličku relaciju $a = (a_1, a_2, \ldots, a_n)$, gdje je a oznaka za jedan (složeni) objekt, dok $a_1, a_2, \ldots, a_n$ predstavljaju realne brojeve,

[1] Treba imati na umu da se moderna notacija vektora pojavljuje tek krajem devetnaestog stoljeća, motivirana kako kvaternionima, tako i radovima britanskog fizičara Jamesa Clerka Maxwella.

bilo pozitivne, bilo negativne." [3]

Prirodno, Hamilton je geometriju prostora pokušao opisati uređenim trojkama, analogno opisu kompleksne ravnine pomoću uređenih parova. Osnovni problem u izgradnji nove algebarske strukture predstavljala je operacija množenja takvih trojki. U konstrukciji nove algebarske strukture Hamilton je želio zadržati sljedeća svojstva operacija: asocijativnost i komutativnost množenja; distributivnost množenja u odnosu na zbrajanje; egzistenciju jedinstvenog multiplikativnog inverza za sve elemente osim za nulu (dakle, mogućnost dijeljenja sa svim brojevima osim s nulom); da se operacijom množenja čuva duljina; te konačno, da definirane operacije imaju razumnu interpretaciju u trodimenzionalnom prostoru.

Nakon trinaest godina rada na tom problemu, Hamilton je došao do rješenja, no u formi malo drugačijoj od prvotno očekivane. Umjesto uređenih trojki, tražena struktura izgrađena je nad uređenim četvorkama, (a, b, c, d). Analogno kao i s kompleksnim brojevima, za navedenu uređenu četvorku uvodimo oznaku

$$a + b \cdot i + c \cdot j + d \cdot k.$$

Takve elemente danas nazivamo *kvaternionima*.

2. Kvaternioni i njihova svojstva

Formalizirajmo algebarsku strukturu čije smo elemente opisali u prethodnom poglavlju.

Definicija 1. *Skup četvero-dimenzionalnih brojeva oblika* $q = a + b \cdot i + c \cdot j + d \cdot k$, *gdje su* $a, b, c, d \in \mathbb{R}$, *označavamo s* $\mathbb{H}$. *Na tom skupu definiramo operaciju zbrajanja:*

$$\begin{aligned} &(a_1 + b_1 \cdot i + c_1 \cdot j + d_1 \cdot k) + (a_2 + b_2 \cdot i + c_2 \cdot j + d_2 \cdot k) \\ =&(a_1 + a_2) + (b_1 + b_2) \cdot i + (c_1 + c_2) \cdot j + (d_1 + d_2) \cdot k. \end{aligned}$$

Osnovni identiteti množenja jesu

$$i^2 = j^2 = k^2 = ijk = -1,$$

iz čega slijedi:

$$ij = k, \quad ji = -k, \quad jk = i, \quad kj = -i, \quad ki = j, \quad ik = -j.$$

Pomoću zakona distributivnosti, osnovni identiteti množenja proširuju se na cijeli skup $\mathbb{H}$.

Pokazuje se kako definirana struktura ima sva tražena svojstva, osim svojstva komutativnost množenja, odnosno kako je skup $\mathbb{H}$ sa tako definiranim operacijama zbrajanja i množenja nekomutativna algebra s dijeljenjem nad skupom realnih brojeva.

Već je Hamilton uočio da se kvaternioni sastoje od dva fundamentalno različita dijela, koje je po uzoru na kompleksne brojeve nazvao realnim i imaginarnim dijelom kvaterniona. Za kvaternion $q = a + b \cdot i + c \cdot j + d \cdot k$, realni dio je $\operatorname{Re}(q) = a$, dok imaginarni dio glasi $\operatorname{Im}(q) = b \cdot i + c \cdot j + d \cdot k$.

U skladu s modernom notacijom kvaterniona, prvenstveno u okviru njihove interpretacije u geometriji prostora, a u skladu s današnjom upotrebom u računarstvu, realni i imaginarni dio kvaterniona označavat (i nazivat) ćemo malo drugačije.

Imaginarni dio kvaterniona nazivat i označavat ćemo kao vektor $\overrightarrow{v}$. Imaginarne jedinice kvaterniona i, j i k interpretiramo kao jedinične vektore smjera koordinatnih osi u trodimenzionalnom prostoru, $\overrightarrow{i}$, $\overrightarrow{j}$ i $\overrightarrow{k}$. Realni dio kvaterniona nazivamo skalarom i obilježavamo sa s. Općenito[2], kvaternion ćemo promatrati kao uređeni par $q = (s, \overrightarrow{v})$, gdje je $s \in \mathbb{R}$, $\overrightarrow{v} \in \mathbb{R}^3$.

Uz takvu notaciju dolazimo i do prilično elegantnog identiteta koji opisuje množenje kvaterniona $q_1 = (s_1, \overrightarrow{v_1})$ i $q_2 = (s_2, \overrightarrow{v_2})$

$$q_1 \cdot q_2 = (s_1 s_2 - \overrightarrow{v_1} \cdot \overrightarrow{v_2}, s_1 \overrightarrow{v_2} + s_2 \overrightarrow{v_1} + \overrightarrow{v_1} \times \overrightarrow{v_2}),$$

gdje je $\overrightarrow{v_1} \cdot \overrightarrow{v_2}$ standardni skalarni produkt vektora $\overrightarrow{v_1}$ i $\overrightarrow{v_2}$, a $\overrightarrow{v_1} \times \overrightarrow{v_2}$ standardni vektorski produkt vektora $\overrightarrow{v_1}$ i $\overrightarrow{v_2}$. Kako bismo pomoću kvaterniona definirali prostorne rotacije, potrebni su nam sljedeći pojmovi:

Definicija 2. ***Konjugat*** *kvaterniona $q = (s, \overrightarrow{v})$ jest kvaternion*

$$\overline{q} = (s, -\overrightarrow{v}).$$

Definicija 3. ***Modul*** *kvaterniona $q = (s, \overrightarrow{v})$ jest realni broj $|q|$ koji računamo po formuli:*

$$|q| = \sqrt{q\overline{q}}.$$

Iz definicije množenja kvaterniona slijedi kako se modul kvaterniona $q = (s, \lambda\overrightarrow{v_0})$, gdje je $\overrightarrow{v_0}$ jedinični vektor, može računati prema formuli:

$$|q| = \sqrt{s^2 + \lambda^2}.$$

[2]Čitatelje zainteresirane za formalno utemeljenje ove notacije upućujemo na [4].

Definicija 4. ***Normalizirani kvaternion*** *kvaterniona* $q = (s, \vec{v})$ *je kvaternion* q_0 *za koji vrijedi:*

$$q_0 = \frac{q}{|q|} = \frac{q}{\sqrt{s^2 + \lambda^2}},$$

gdje jest $\vec{v} = \lambda \vec{v_0}$, *a* $\vec{v_0}$ *jedinični vektor.*

Definicija 5. ***Inverz*** *kvaterniona* $q = (s, \vec{v})$ *jest kvaternion* q^{-1} *za koji vrijedi:*

$$q^{-1} = \frac{\overline{q}}{|q|^2}.$$

Iz definicije množenja slijedi kako je umnožak kvaterniona i njegovog inverza jednak jediničnom realnom kvaternionu, $r_1 = (1, \vec{0})$, koji je neutralni element za operaciju množenja kvaterniona.

3. Metode prostorne rotacije

3.1. Eulerove rotacije

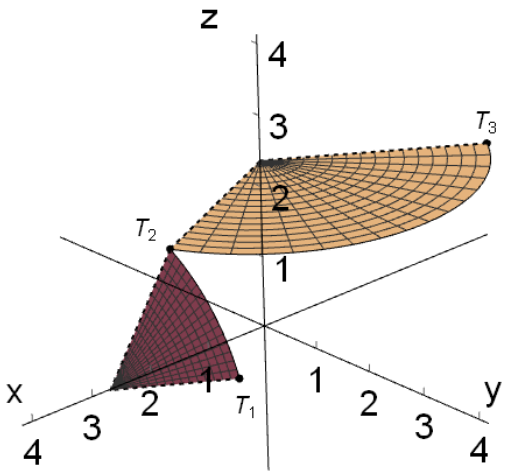
Prostornu rotaciju oko ishodišta možemo vršiti na više načina. Za početak ćemo predstaviti najjednostavniju metodu, tzv. *Eulerove rotacije.* Eulerove rotacije predstavljaju kompoziciju više rotacija oko koordinatnih osi. Kod tih rotacija, jedna koordinata rotirane točke ostaje ista (na primjer, prilikom rotiranja točke oko y-osi, y-koordinata točke se ne mijenja), dok na druge dvije koordinate primjenjujemo klasičnu ravninsku rotaciju. Sljedeće tri matrice opisuju rotacije oko koordinatnih osi za kut α u pozitivnom smjeru. [2]

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix},$$

$$R_{y,\alpha} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix},$$

$$R_{z,\alpha} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Primjer 1. *Rotirajmo točku* $T_1\,(2.7, 2.5, 1)$ *oko* x*-osi za* $40°$*, pa potom oko* z*-osi za* $120°$*.*



Slika 1. Prikaz kompozicije dviju Eulerovih rotacija iz primjera 1.

Točku $T_1\,(2.7, 2.5, 1)$ možemo zapisati u obliku stupčane matrice $\begin{bmatrix} 2.7 \\ 2.5 \\ 1 \end{bmatrix}$.
Zapišimo pripadne matrice rotacije te ih pomnožimo sa stupčanom matricom koja je zapis točke koju rotiramo:

$$r_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 40^\circ & -\sin 40^\circ \\ 0 & \sin 40^\circ & \cos 40^\circ \end{bmatrix} \cdot \begin{bmatrix} 2.7 \\ 2.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 2.7 \\ 1.272323 \\ 2.373013 \end{bmatrix}.$$

Nakon rotacije oko x-osi za 40°, točka T_1 preslikala se u točku $T_2\,(2.7, 1.272323, 2.373013)$. Sad primijenimo drugu rotaciju na dobivenu točku:

$$r_3 = \begin{bmatrix} \cos 120^\circ & -\sin 120^\circ & 0 \\ \sin 120^\circ & \cos 120^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2.7 \\ 1.272323 \\ 2.373013 \end{bmatrix} = \begin{bmatrix} -2.451864 \\ 1.702107 \\ 2.373013 \end{bmatrix}.$$

Dakle, nakon dvije zadane rotacije točka T_1 preslikala se u točku $T_3\,(-2.45186, 1.702107, 2.373013)$. Lako se provjeri kako je udaljenost točaka T_1 i T_3 od ishodišta jednaka, odnosno da se radi o rotaciji oko ishodišta. Na slici 1 vizualno su prikazane navedene rotacije.

Naravno, treba istaknuti i kako je navedenu kompoziciju rotacija moguće provesti i tako da prvo pomnožimo matrice Eulerovih rotacija, pa potom dobivenu matricu prostorne rotacije primijenimo na zadanu točku.

3.2. Rotacija oko proizvoljne osi

Drugi način za rotiranje točaka u prostoru jest rotacija oko proizvoljne osi koja prolazi kroz ishodište. Metoda rotacije oko proizvoljne osi svodi se nizom rotacija na rotaciju oko jedne od koordinatnih osi. Iako se taj postupak može zapisati i vektorski, u ovom ćemo radu (nastaviti) koristiti matrični zapis navedenih operacija.

Na slici 2 prikazana je rotacija koju analiziramo. Os rotacije dana je normaliziranim vektorom $\overrightarrow{n_0}$. Taj vektor zatvara kut θ sa svojom projekcijom na koordinatnu xz-ravninu, dok navedena projekcija zatvara kut ϕ s pozitivnim krakom x-osi. Kut rotacije (točke P u točku P') označavamo s α. Rotaciju tada provodimo prateći sljedeće korake:

1. Provedemo rotaciju za kut ϕ oko jedne od koordinatnih prostornih osi (u primjeru sa slike 2 oko z-osi), kako bi se os rotacije našla u jednoj koordinatnoj ravnini (u primjeru sa slike 2 u koordinatnoj xz-ravnini).[3]

2. Provedemo rotaciju za kut θ oko koordinatne osi kako bi se os rotacije poklopila s koordinatnom osi. U primjeru sa slike 2 rotaciju vršimo oko y-osi, kako bi se os rotacije poklopila s x-osi.

3. Provedemo rotaciju objekta za kut α oko osi rotacije koja se sada poklapa s jednom koordinatnom osi.

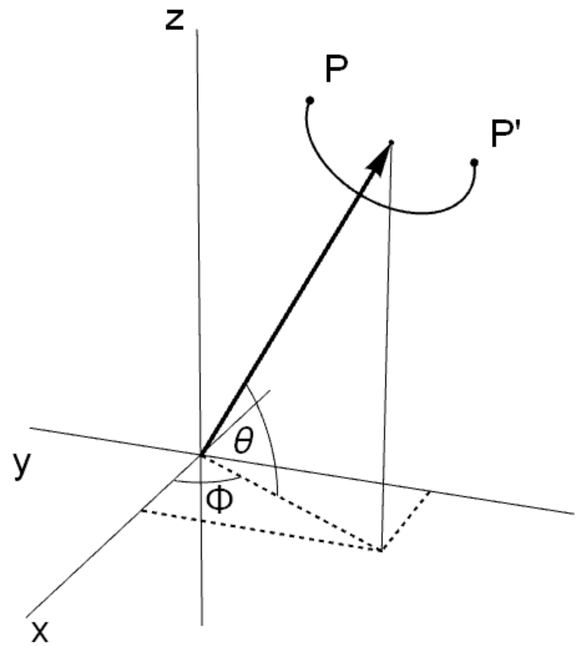
4. Provedemo rotaciju za kut $-\theta$ oko odgovarajuće koordinatne osi (u slučaju sa slike 2, oko y-osi).

5. Provedemo rotaciju za kut $-\phi$ oko odgovarajuće koordinatne osi (u slučaju sa slike 2, oko z-osi).

Opisani postupak, koji definira operator rotacije oko osi u smjeru vektora $\overrightarrow{n_0}$ za kut α, $R_{\overrightarrow{n_0},\alpha}$ u slučaju prikazanom na slici 2 matrično zapisujemo na sljedeći način:

$$R_{\overrightarrow{n_0},\alpha} = R_{z,-\phi} R_{y,-\theta} R_{x,\alpha} R_{y,\theta} R_{z,\phi}.$$

[3]Kod određivanja navedenih kutova treba paziti na njihovu orijentaciju, odnosno predznak.



Slika 2. Prikaz problema rotacije oko proizvoljne osi.

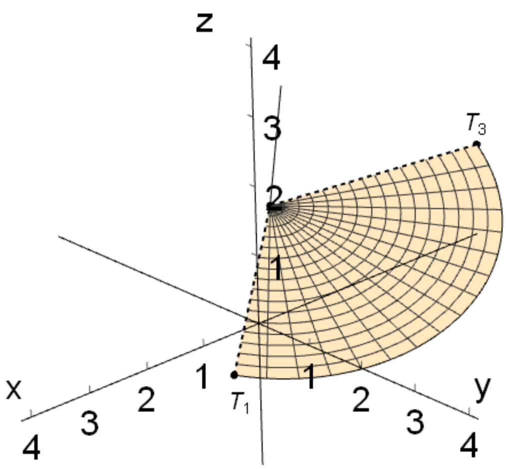
Primjer 2. *Provedimo rotaciju iz primjera 1 pomoću neposredne rotacije oko proizvoljne osi.*

Za provođenje te rotacije potrebno je upotrijebiti prijelazne formule iz Eulerovih rotacija u neposrednu rotaciju. Formule u ovom članku nećemo navoditi zbog njihove opsežnosti, već navodimo rezultat te vizualni prikaz rotacije. Za detalje, zainteresirane čitatelje upućujemo na [5].

Za provođenje rotacije opisane u primjeru 1, potrebno je izvršiti rotaciju oko osi definirane jediničnim vektorom $\overrightarrow{n_0} = \begin{bmatrix} 0.193725 \\ 0.335541 \\ 0.921891 \end{bmatrix}$ za kut $\alpha = 123.951\,358°$. Upotrebom jednostavnih trigonometrijskih identiteta u pravokutnom trokutu slijedi da je kut $\theta = -60°$ te $\phi = -67.2041°$, a onda je matrica rotacije:

$$R_{\overrightarrow{n_0},\alpha} = \begin{bmatrix} -0.5 & -0.663414 & 0.55667 \\ 0.866025 & -0.383022 & 0.321394 \\ 0 & 0.642788 & 0.766044 \end{bmatrix}.$$

Primjenom te matrice rotacije na točku $T_1\,(2.7, 2.5, 1)$ dobivamo točku $T_3\,(-2.451864, 1.702107, 2.373013)$, baš kao i u primjeru 1. Navedena rotacija vizualno je prikazana na slici 3.



Slika 3. Prikaz neposredne rotacije u prostoru iz primjera 2.

3.3. Rotacija pomoću kvaterniona

Rotaciju kvaternionima provodimo pomoću tzv. *metode dvostrukog kuta*. Za početak, ako vektor $\overrightarrow{p} \in \mathbb{R}^3$ želimo rotirati pomoću kvaterniona, moramo ga zapisati u formi kvaterniona. To činimo tako da ga zapišemo u formi tzv. čistog kvaterniona (koji ćemo označiti s

$$p = (0, \overrightarrow{p}).$$

Nadalje, u formi kvaterniona trebamo zapisati os rotacije (zadanu pomoću jediničnog vektora smjera $\overrightarrow{n_0}$) te kut rotacije α. To činimo pomoću kvaterniona q_r:

$$q_r = \left(\cos\frac{\alpha}{2}, \sin\frac{\alpha}{2}\overrightarrow{n_0}\right).$$

Iz definicije modula kvaterniona (definicija 3) lako slijedi kako je modul kvaterniona rotacije q_r jednak 1 jer je $\cos^2\frac{\alpha}{2} + \sin^2\frac{\alpha}{2} = 1$. Važno je prilikom rotiranja koristiti jedinične kvaternione, jer kvaternioni modula različitog od jedan, pored rotacije vrše i množenje skalarom vektora $\overrightarrow{p}$ (baš kao i u kompleksnoj ravnini). Posljednji element koji nam je potreban za definiranje rotacije pomoću kvaterniona jest inverz kvaterniona rotacije, q_r^{-1}. Kako je kvaternion rotacije jediničan, iz definicije 5 slijedi:

$$q_r^{-1} = \frac{\overline{q_r}}{|q_r|} = \overline{q_r} = \left(\cos\frac{\alpha}{2}, -\sin\frac{\alpha}{2}\overrightarrow{n_0}\right).$$

Pomoću navedenih elemenata možemo definirati rotaciju u prostoru pomoću kvaterniona.

Definicija 6. *Neka je $p = (0, \overrightarrow{p})$ kvaternion koji opisuje vektor koji želimo rotirati te neka je $q_r = \left(\cos\frac{\alpha}{2}, \sin\frac{\alpha}{2}\overrightarrow{n_0}\right)$ kvaternion rotacije određen pomoću kuta rotacije α i jediničnog vektora smjera osi rotacije $\overrightarrow{n_0}$. Kvaternion p' koji opisuje rezultat rotacije vektora $\overrightarrow{p}$ oko osi zadane s $\overrightarrow{n_0}$ za kut α računamo prema formuli:*

$$p' = q_r p q_r^{-1}.$$

Jednostavno je provjeriti da je duljina vektora $\overrightarrow{p'}$ jednaka duljini vektora $\overrightarrow{p}$, što slijedi iz činjenice da je q jedinični kvaternion te da je $|q_1 q_2| = |q_1| \cdot |q_2|$. No, dokaz da navedeni produkt uistinu rotira vektor $\overrightarrow{p}$ nije trivijalan, i može se naći u [3].

Tvrdnju je dokazao i sam Hamilton, no taj dokaz nije objavio. Naknadno, 1845. godine dokaz objavljuje Arthur Cayley, koji je sam priznao pravo prvenstva Hamiltonu. Ipak, treba istaknuti kako je tvrdnju o rotacijskom svojstvu navedenog produkta prvi dokazao francuski matematičar Benjamin Olinde Rodrigues, o čemu Hamilton nije imao saznanja.

Primjer 3. *Provedimo rotaciju iz primjera 2 pomoću kvaterniona.*

Radij-vektor točke T_1 glasi $\overrightarrow{r_1} = 2.7\overrightarrow{i} + 2.5\overrightarrow{j} + \overrightarrow{k}$, pa je kvaternion koji ćemo rotirati jednak $p_1 = \left(0, 2.7\overrightarrow{i} + 2.5\overrightarrow{j} + \overrightarrow{k}\right)$. Rotaciju provodimo oko pravca određenoga jediničnim vektorom $\overrightarrow{n_0} = \begin{bmatrix} 0.193725 \\ 0.335541 \\ 0.921891 \end{bmatrix}$, dok je kut rotacije $\alpha = 123.951\,358°$, pa je kvaternion rotacije jednak

$$q_r = \left(\cos 61.975\,679°, \sin 61.975\,679° \cdot \left(0.193725\overrightarrow{i} + 0.335541\overrightarrow{j} + 0.921891\overrightarrow{k}\right)\right),$$

$$q_r = \left(0.469846, 0.17101\overrightarrow{i} + 0.296198\overrightarrow{j} + 0.813797\overrightarrow{k}\right).$$

Rezultat tražene rotacije tada je jednak

$$p_2 = q_r p_1 q_r^{-1},$$

pa množenjem danih kvaterniona dobivamo

$$p_2 = \left(0, -2.45186\overrightarrow{i} + 1.70211\overrightarrow{j} + 2.37302\overrightarrow{k}\right).$$

Vidimo kako dobiveni kvaternion upravo opisuje rezultat rotacije, točku T_3 dobivenu u primjerima 1 i 2.

4. Usporedba učinkovitosti metoda za prostornu rotaciju

U ovom poglavlju predstavit ćemo rezultate usporedbe računalne učinkovitosti provođenja prethodno opisanih metoda za prostornu rotaciju. Programska rješenja korištena u ovom odjeljku dostupna su putem GitHub platforme na sljedećoj poveznici:
`https://github.com/karlobratko/rotation-visualizer`.

Cilj je aplikacije omogućiti jednostavnu demonstraciju rotacije modela u prostoru i kreiranje kompleksnih nizova rotacija, pri čemu se vremenska učinkovitost akumulacije niza rotacija može mjeriti.

Vremensku složenost mjerimo kroz definiranje broja automatski slučajno generiranih rotacija te mjerenjem vremena potrebnog za izračunavanje akumulacije danih rotacija. Rezultate mjerenja vremenske učinkovitosti prikazat ćemo u formi prosječnog, minimalnog i maksimalnog vremena u nanosekundama potrebnog za izračunavanje rotacije u danom okviru. Podatkovni set generiran aplikacijom pohranjujemo u .csv datoteku.

Iz perspektive aplikacije, veličina modela, odnosno broj opisnika točaka i indeksa opisnika točaka, nema utjecaja na brzinu izračunavanja rotacije jer se izračunavanje izvodi na procesoru, a ne tijekom iscrtavanja na grafičkoj kartici. Iz tog razloga, za sva mjerenja učitali smo model kugle s 2143 opisnika točaka i 11904 indeksa opisnika točaka koji smo podvrgnuli automatskom generiranju rotacija. Automatsko generiranje rotacija konfigurirano je da u periodu od 120 minuta (7200 sekundi) generira 100000 rotacija. Time se svake sekunde generira otprilike 14 rotacija, što će aplikaciji koja svakih 8.3 milisekundi mijenja okvir (120 okvira u sekundi) dati oko 8 okvira u kojima se uzimaju vrijednosti mjerenja i spremaju u podatkovni set. Ovakva mjerenja pokrenuli smo za sve tri metode rotacije na prijenosnom računalu sljedećih performansi:

- operacijski sustav: Windows 10

- centralna procesna jedinica: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 Mhz, 4 Core(s), 8 Logical Processor(s)
- grafička procesna jedinica: NVIDIA GeForce RTX 2060
- RAM: 32.0 GB, DDR4 sinkrona dinamička memorija s nasumičnim pristupom

Druga metoda mjerenja učinkovitosti koju smo proveli jest mjerenje procesorskih naredbi potrebnih za izračunavanje rotacije. To smo proveli pomoću web aplikacije *Compiler Explorer*. Funkcije zadužene za izvršavanje rotacije kopirane su u web aplikaciju na temelju kojih je generiran kôd u jeziku *Assembler* pomoću prevoditelja *x86-64 clang 14.0.0.* Generirani kôd smo analizirali te izvršili brojanje naredbi potrebnih za izvršavanje metode. Ovakvim mjerenjem dobili smo okvirni podatak o kompleksnosti funkcija i funkcijskih članova zaduženih za izvršavanje rotacije za sve tri vrste rotacije.

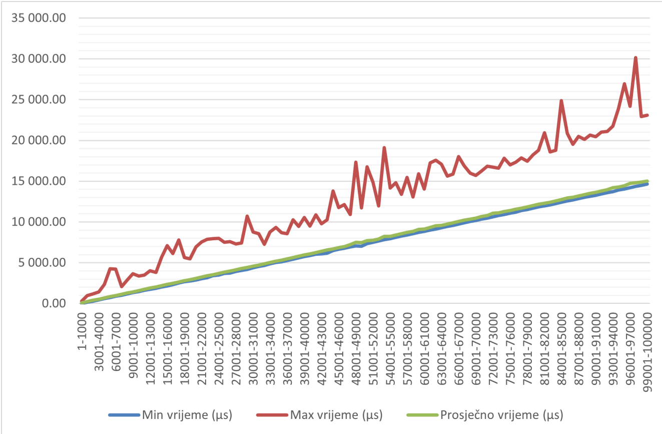
Treća informacija koju analiziramo jest okvirna potrošnja memorije. Ova analiza vrlo je jednostavna i daje uvid u količinu memorije u bajtovima koja se mora rezervirati kako bi se spremila određena algebarska struktura.

Detaljniji opis provedenih mjerenja moguće je pronaći u [1].

4.1. Analiza rotacija kvaternionima

Izračunavanje rotacije kvaternionima temelji se na množenju kvaterniona koji definiraju rotaciju. Posljednji korak u izračunavanju rotacije jest konverzija kvaterniona u matrični oblik. Mjerenje je ukazalo na linearni porast prosječnog vremena izračunavanja rotacije iskazanoga u mikrosekundama s porastom broja rotacija (slika 4). Iako postoje odskakanja u pogledu maksimalnih vrijednosti, linije prosječnih i minimalnih vrijednosti gotovo se preklapaju. Zbog navedenoga možemo zanemariti odskakanja za koja možemo pretpostaviti da su rezultat sistemskih poziva operacijskog sustava.

Analizom asemblerskog kôda utvrdili smo da su najzahtjevnije operacije one koje vrše konverziju između tipova. Tako je za konverziju kuta i osi rotacije u kvaternion koji opisuje rotaciju potrebno oko 249 naredbi. U izračun se ne uzimaju u obzir asemblerske naredbe koje ulaze u pozive standardnih funkcija `std::cos` i `std::sin`. Nadalje, množenje dva kvaterniona, uz uvjet da su oba kvaterniona unaprijed definirana, izvodi se u 165 linija asemblerskog kôda, što proizlazi iz 16 operacija množenja



Slika 4. Odnos broja rotacija i vremena za izračun konačne rotacije pomoću kvaterniona.

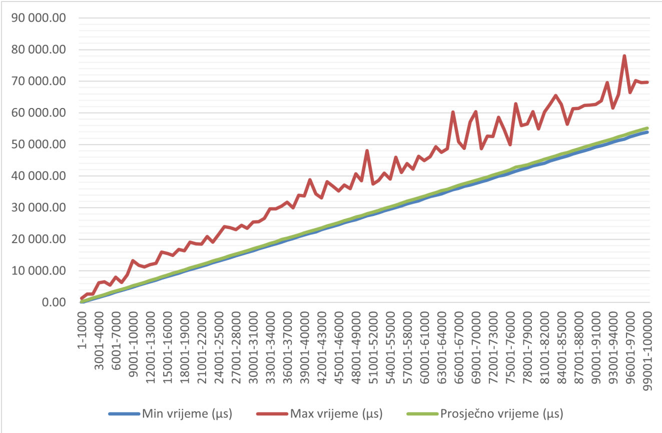
i 12 operacija zbrajanja i oduzimanja. Iz navedenoga se može zaključiti da operacija množenja kvaterniona nije zahtjevna. Posljednji korak u izračunavanju rotacije pomoću kvaterniona jest konverzija kvaterniona u matrični oblik. Ovaj korak rezultira s dodatnih 296 linija asemblerskog kôda. Valja naglasiti kako se ovaj korak ponavlja samo jednom tijekom izračunavanja rotacije, dok se prethodna dva koraka ponavljaju za svaku privremenu rotaciju.

Spremanje jednoga kvaterniona u memoriji zahtijeva mjesta za 4 numerička podatkovna tipa ograničena na decimalne brojeve, što znači da tip kvaterniona može zauzimati minimalno 128 bitova (16 bajtova), ako se spremaju vrijednosti podatkovnog tipa `float`, ili maksimalno 512 bitova (64 bajta), ako se spremaju vrijednosti podatkovnog tipa `long double` na sustavu koji podržava format standarda IEEE-754 binary128.

4.2. Analiza Eulerovih rotacija

Izračunavanje Eulerovih rotacija temelji se na množenju rotacijskih matrica. Mjerenje je ukazalo na linearni porast prosječnog vremena izračunavanja rotacije iskazanoga u mikro sekundama s porastom broja rotacija (slika 5). Kao i kod rotacije kvaternionima, postoje odskakanja u pogledu maksimalnih vrijednosti za koja možemo pretpostaviti da su rezultat sistemskih poziva operacijskog sustava.

Analizom asemblerskog kôda utvrdili smo da su najzahtjevnije operacije



Slika 5. Odnos broja rotacija i vremena za izračun konačne rotacije pomoću Eulerovih rotacija.

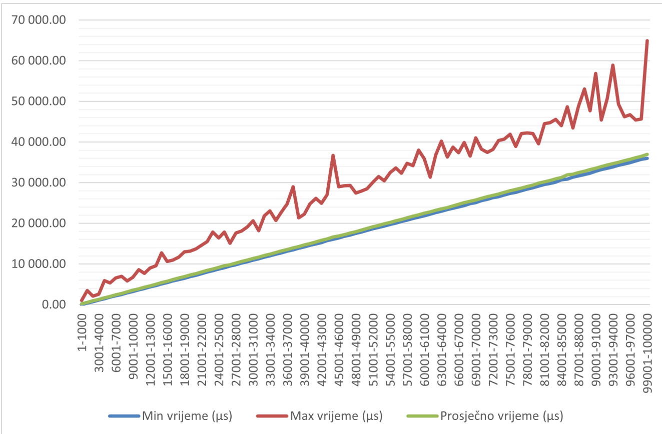
one koje vrše konverziju između tipova. Tako je za konverziju tri kuta rotacije oko glavnih osi Kartezijevog koordinatnog sustava u matricu koji opisuje Eulerovu rotaciju potrebna 501 naredba među kojima se ističe šest poziva standardnih funkcija `std::cos` i `std::sin`. U izračunu nismo uzeli u obzir asemblerske naredbe koje ulaze u pozive standardnih funkcija `std::cos` i `std::sin` kako bi se mjerenje moglo usporediti s kvaternionima.

Nadalje, množenje dviju rotacijskih matrica, uz uvjet da su obje matrice unaprijed definirane, izvodi se u 366 linija asemblerskog kôda, što proizlazi iz 64 operacije množenja i 48 operacija zbrajanja. Iz navedenoga se može zaključiti da je operacija množenja matrica otprilike dvostruko sporija od množenja kvaterniona. U slučaju matrica nije potrebno vršiti nikakvu završnu konverziju, pa nema dodatnih koraka. Valja naglasiti kako se prethodna dva koraka konverzije i množenja ponavljaju za svaku privremenu rotaciju.

Spremanje jedne matrice sa 4 retka i 4 stupca u memoriji zahtijeva mjesta za 16 aritmetičkih podatkovnih tipova, što znači da tip matrice može zauzimati minimalno 128 bitova (16 bajtova) ili maksimalno 2048 bitova (256 bajtova), ovisno o upotrebi podatkovnog decimalnog i integralnog tipa.

4.3. Analiza neposredne rotacije

Izračunavanje rotacije matricama oko proizvoljnih osi temelji se na množenju rotacijskih matrica. Mjerenje je ukazalo na linearni porast prosječnog vremena izračunavanja rotacije iskazanoga u mikro sekundama s porastom broja rotacija (slika 6). Kao i u prethodna dva slučaja, postoje odskakanja u pogledu maksimalnih vrijednosti koja zanemarujemo kao rezultat sistemskih poziva operacijskog sustava.



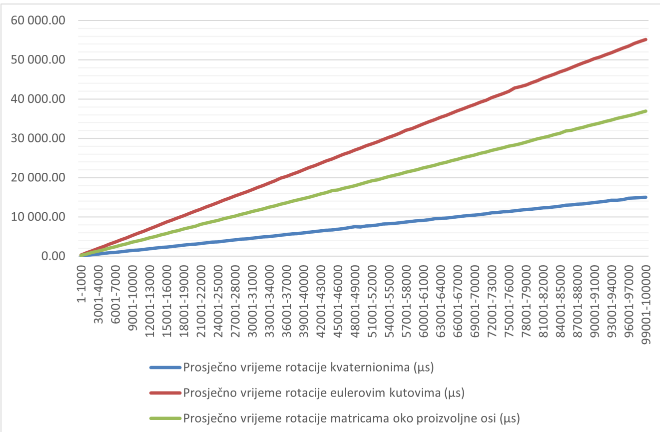
Slika 6. Odnos broja rotacija i vremena za izračun konačne rotacije pomoću neposredne rotacije.

Rotacija matrica oko proizvoljne osi implementirana je kao jedinstvena operacija koja objedinjuje funkcionalnosti transformacije kuta i osi rotacije u matricu i množenje matrica. Za izvođenje navedene operacije potrebno je 947 linija asemblerskog kôda. Iako rotacija matrica oko proizvoljne osi koristi veći broj asemblerskih naredbi od kombinacije konverzije i množenja u slučaju Eulerovih rotacija, ona sadrži samo dva poziva standardnih funkcija `std::cos` i `std::sin`. U izračunu nismo uzeli u obzir asemblerske naredbe koje ulaze u pozive standardnih funkcija `std::cos` i `std::sin` kako bi se mjerenje moglo usporediti s kvaternionima. Valja naglasiti kako se navedena operacija rotacije matrica oko proizvoljne osi ponavlja za svaku privremenu rotaciju koja se akumulira.

Zauzeće memorijskog prostora u slučaju matrica za rotaciju oko proizvoljne osi jednako je zauzeću memorijskog prostora u slučaju matrica koje opisuju Eulerove rotacije.

4.4. Usporedba učinkovitosti metoda rotacije

Vremenska analiza učinkovitosti računanja rotacije ukazala je na prednosti rotacije kvaternionima u odnosu na Eulerove rotacije i neposredne rotacije oko proizvoljnih osi. Sve tri metode ukazuju na linearni porast prosječnog vremena izračunavanja rotacije iskazanoga u mikrosekundama s porastom broja rotacija, ali različitim intenzitetom. Metoda rotacije kvaternionima pokazala je najbolje rezultate na svim razinama broja rotacija te je prosječno 245.74 % brža od drugoplasirane neposredne rotacije. Eulerove rotacije ostvarile su vremenski najslabiji rezultat. Rotacija kvaternionima u odnosu na Eulerove rotacije bilježi prosječno 367.56 % brže izvođenje. Neposredna rotacija matricama oko proizvoljne osi u odnosu na Eulerove rotacije bilježi prosječno 149.57 % brže izvođenje. Ti su rezultati izvedeni iz podataka prikazanih na slici 7.



Slika 7. Usporedba prosječnih vremena u odnosu na broj rotacija za tri analizirane metode.

Rezultate mjerenja trajanja izvršavanja rotacija za tri navedene metode potvrđuje i broj asemblerskih operacija koje svaka od metoda izvršava. Tako je za izračunavanje 10 rotacija pomoću kvaterniona potrebno 4436 asemblerskih operacija, dok Eulerove rotacije koriste 8670, a neposredne rotacije matricama 9470 asemblerskih operacija. Iako je broj asemblerskih operacija potreban za izračunavanje Eulerovih rotacija manji od broja operacija koje koristi neposredna rotacija, veći broj uzastopnih poziva standardnih funkcija `std::cos` i `std::sin` u slučaju Eulerovih rotacija rezultira duljim vremenom izvođenja.

Iz perspektive memorijske efikasnosti algebarske strukture, kvaternioni su četiri puta efikasniji od druge dvije metode rotacije. Kvaternioni omogućuju opisivanje rotacije spremanjem samo četiri numeričke vrijednosti, dok matrice zahtijevaju spremanje šesnaest numeričkih vrijednosti. Tijekom dugoročnog rada aplikacije, ovakva razlika ima utjecaj na učinkovitost programa.

5. Zaključak

Algebra kvaterniona razvila se s ciljem jednostavnijeg opisivanja rotacije u trodimenzionalnom prostoru po uzoru na kompleksne brojeve u ravnini. Iako su nastali prije matričnog i vektorskog računa, kvaternioni zbog svoje kompleksne aritmetike nisu doživjeli popularnost kao matrični i vektorski račun.

S razvojem računalnih znanosti, javila se potreba za optimizacijom metoda transformacije među kojima je i rotacija objekata u prostoru. Iako matrice predstavljaju standard za čuvanje informacija o transformaciji, kvaternioni su se pokazali kao dobra alternativa za prikazivanje i izvršavanje rotacija. Osim što omogućuju spremanje potrebnih informacija u manjem memorijskom prostoru, rotacija pomoću kvaterniona pokazala je superiornost po pitanju vremenske učinkovitosti u odnosu na druge načine rotacije.

Treba istaknuti kako rotacija kvaternionima također ne pati od *gimbal-lock* anomalije[4] kojoj su podložne Eulerove rotacije. [4]

Iako su kvaternioni alat koji uvelike ubrzava i optimizira izračunavanje rotacija, kvaternioni uvode dodatnu kompleksnost u sustave koja često nije opravdana njihovom učinkovitošću. Naime, kvaternioni i algebra kvaterniona dodatni su matematički koncept s kojim korisnici moraju biti upoznati. Iz tog razloga, alati za obradu grafike nerijetko rotaciju kvaternionima skrivaju ispod lakše razumljivih Eulerovih rotacija. Pored toga, matrice se mnogo lakše vizualiziraju te su primjenjive na sve vrste transformacija, dok kvaternioni zadržavaju svoju primjenu samo na rotacijama. Stoga kvaternione ne treba promatrati kao zamjenu za matrice, već kao alat koji u kombinaciji s matricama omogućuje učinkovitije i sigurnije transformiranje modela u prostoru.

[4] *Gimbal-lock* fenomen je koji se javlja u sustavima s tri koncentrična rotirajuća prstena ili *gimbala*. Do fenomena dolazi kada se osi rotacije rotirajućih prstenova poklope čime se gubi jedan stupanj slobode rotiranja, odnosno rotacija oko jedne od osi više nije nezavisna od ostalih rotacija.

Literatura

[1] K. Bratko, *Korištenje i analiza učinkovitosti kvaterniona za rotaciju modela u 3D prostoru*, završni rad, Visoko učilište Algebra, 2023.

[2] A. Hatzivelkos, H. Kovač, T. Milun, *Matematika za IT*, Algebra 2022.

[3] V. J. Katz, *A History of Mathematics, An Introduction*, Pearson Education, 2009.

[4] J. Vince, *Quaternions for Computer Graphics*, Springer-Verlag London Ltd., 2021.

[5] J. Vince, *Rotation Transforms for Computer Graphics*, Springer-Verlag London Ltd., 2021.

Karlo Bratko
student, Visoko učilište Algebra, Gradišćanska 24, Zagreb
E-mail: `kbratko@racunarstvo.hr`

Aleksandar Hatzivelkos
Veleučilište Velika Gorica, Zagrebačka 5, Velika Gorica
E-mail: `hatzi@vvg.hr`