

ლევან შოშიაშვილი

# მოდელირება და ვიზუალიზაცია

(ლექცია/პრაქტიკული/შენიშვნები)



# სარჩევი

<b>1</b>	<b>საკოორდინატო სისტემები</b>	<b>1</b>
1.1	საკოორდინატო სისტემები . . . . .	1
<b>2</b>	<b>გარდაქმნები</b>	<b>5</b>
2.1	2D გარდაქმნები . . . . .	5
2.2	ერთგვაროვანი კოორდინატები . . . . .	7
2.3	3D გარდაქმნები . . . . .	9
2.4	ვექტორის გარშემო მობრუნება . . . . .	10
2.5	არეკვლა სიბრტყის მიმართ . . . . .	12
2.6	ნორმალის გარდაქმნა . . . . .	12
2.7	საკოორდინატო სისტემა, წერტილი, ვექტორი . . . . .	13
2.7.1	ვექტორი სხვადასხვა ბაზისში . . . . .	13
2.7.2	წერტილები და საკოორდინატო სისტემა . . . . .	14
2.8	კოორდინატები სხვადასხვა სისტემაში . . . . .	15
2.9	გარდაქმნები შუალედური საკ. სისტემის გამოყენებით . . . . .	16
2.10	მაუსით ბრუნვა . . . . .	17
2.11	მობრუნების პარამეტრიზაცია . . . . .	18
2.12	კვატერნიონი . . . . .	19
2.13	კვატერნიონის თვისებები . . . . .	19
2.14	ვექტორის გარშემო მობრუნება. კინემატიკური გზა . . . . .	20
2.15	მობრუნება და კვატერნიონი . . . . .	21
2.16	მაუსით ბრუნვა კვატერნიონების გამოყენებით . . . . .	21

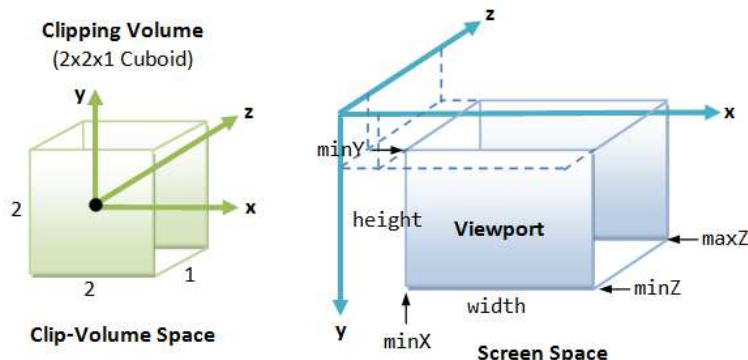


## საკოორდინატო სისტემები

### 1.1 საკოორდინატო სისტემები

გამოვიყენებთ ბიბლიოთეკას OpenGL.

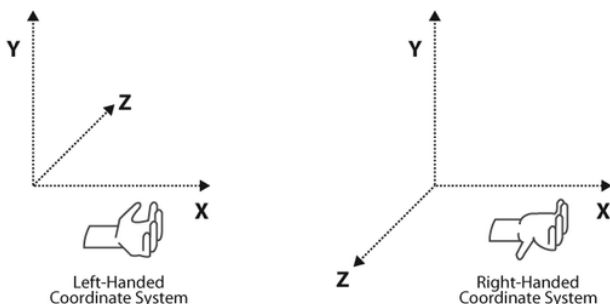
მონიტორზე კოორდინატები აითვლება მარცხენა ზედა კუთხიდან.



სურ 1.1: ეკრანის საკოორდინატო სისტემა

OpenGL კოორდინატები აითვლება მარცხენა ქვედა კუთხიდან. საკოორდინატო სისტემა არის X, Y, Z ოღონდ Z მიმართულება არის ჩვენსკენ ეკრანის სიბრტყიდან(მარჯვენა საკოორდინატო სისტემა). ანუ გეომეტრია რომელიც იხატება OpenGL-ში იხატება ამრჯვენა საკოორდინატო სისტემაში (იხ. სურ. 1.2 ). კამერა (სხვაგვარად თუ როგორ ვუყურებთ ჩვენ 3D სივრცეს სადაც არის ობიექტები) იყენებს მარცხენა საკოორდინატო სისტემას.

Figure 9-2. Left-handed versus right-handed coordinate systems



სურ 1.2: მარცხენა და მარჯვენა საკოორდინატო სისტემები

OpenGL 3D/2D გამოსახულება დამოუკიდებელია მომხმარებლის მიერ გამოყენებული ერთეულებისაგან. ანუ თქვენ შეგიძლიათ იმუშაოთ სმ, მ, მმ, ინჩ ერთეულებში. როგორ ფორმირდება OpenGL გამოსახულება მაგ. ე.წ. „კამერა ობსკურაში“ მოცემულია ენჯელის წიგნი გვ. 40<sup>2</sup>.  $x, y, z$  არის წერტილის კოორდინატები, ხოლო  $x^*, y^*$  პროექციის კოორდინატები.  $z$  კოორდინატი პროექცირდება ცხადია  $-d$  წერტილში.

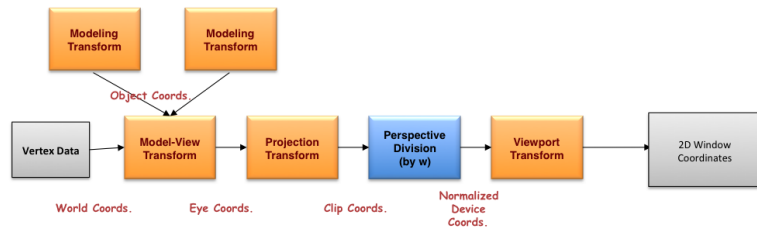
<sup>2</sup>შემდგომში მივუთითებთ როგორც „ენჯ.“

მნიშვნელოვანი პარამეტრია კამერის ხედვის კუთხე. ეს ახასიათებს ობიექტის მაქსიმალურ სიმაღლეს რომელიც პროეცირდება კამერის ფირფიტაზე და იკავებს ფირფიტის მაქსიმალურ სიმაღლეს. იხ.(ენჯ. სურ. 1.16. გვ 41)

$$tg(\theta/2) = h/(2d)$$

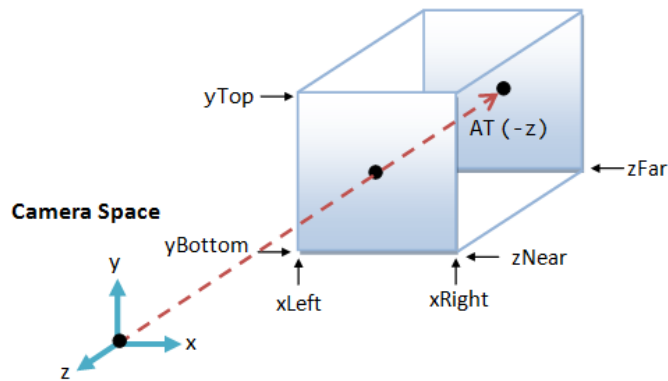
ამ კამერას აქვს რამდენიმე ნაკლი. ვინაიდან ნახვრეტი საიდანაც შედის სხივი უნდა იყოს რაც შეიძლება მცირე (იმისათვის რომ შევიდეს მხოლოდ ერთი სხივი და წერტილი აისახოს წერტილში) განათებულობა მცირეა. ამავე დროს შეუძლებელია ხედვის კუთხის შეცვლა. ეს ორი პრობლემა მოხსნილია ტანამედროვე ფოტოაპარატებში, ვინაიდან გამოყენებულია ლინზები, რაც საშუალებას იძლევა მიიღოს მეტი ენერგია(ობიექტის ზომის შეცვლით) ხოლო ფოკუსური მანძილის ცვლილებით შეიძლება შეიცვალოს ხედვის კუთხე.

3D გამოსახულება პროეცირდება ეკრანზე შემდეგი სქემის გავლით:



სურ 1.3: 3D გამოსახულების ეკრანზე გამოტანის "გზა"

პროექციული გარდაქმნა შეიძლება იყოს ორი სახის: ორთოგრაფიული და პერსპექტივაში:



Orthographic Projection: Camera positioned infinitely far away at  $z = \infty$

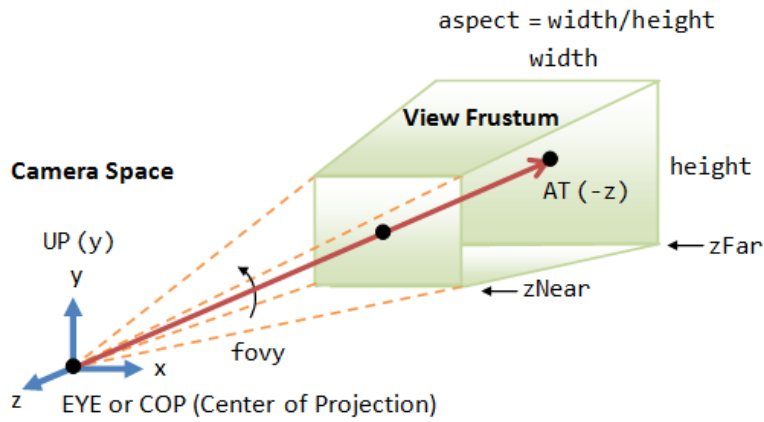
სურ 1.4: ორთოგრაფიული პროექცია

ის ობიექტები რომლებიც ხვდება  $xLeft$ ,  $xRight$ ,  $zNear$ ,  $zFar$ ,  $yBottom$ ,  $yTop$ , კუბში ორთოგრაფიული პროექციის შემტხვევაში აისახება ეკრანზე.

ის ობიექტები რომლებიც ხვდება  $zNear$ ,  $zFar$ , წაკვეთილ პირამიდაში პექსპექტიული პროექციის შემტხვევაში აისახება ეკრანზე.

„eye“-„თვალი“ ან რაც იგივეა პროექციის ცენტრი არის ის წერტილები მსოფლიო საკოორდინატო სისტემაში საიდანაც ვუყურებთ 3D "სცენას".

„fovy“- ესაა ე.წ. ხედვის კუთხე. მნიშვნელოვანია შეფარდება  $width/height$  ეს პროპორცია დაცული უნდა იყოს იმისათვის რომ ეკრანულ კოორდინატებზე ასახული სურათში დაცული იყოს სცენის/ობიექტის პროპორციები.



**Perspective Projection:** The camera's view frustum is specified via 4 view parameters: *fovy*, *aspect*, *zNear* and *zFar*.

სურ 1.5: პერსპექტიული პროექცია

კამერის სურათიდან ჩანს რომ  $z$  მიმართულია ხედვის მიმართულების საპირისპიროდ. ანუ ხედვა ხდება კამერიდან უარყოფითი მიმართულებით. კამერის საკოორდინატო სისტემა „მარცხენა“.

თუ როგორ მუშაობს ხედვა და პროექცია. გაუშვით პროგრამები `matrixModelView`, `matrixProjection` და/ან ნეიტ რობინსის სასწავლო პროგრამები : `transformation`, `projection`.





## გარდაქმნები

### 2.1 2D გარდაქმნები

გარდაქმნები შეიძლება იყოს შემდეგი:

1. გადატანა (ტრანსლიაცია)
2. მობრუნება
3. არეკვლა
4. სკალირება
5. გაჭიმვა

გადატანა

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix} \quad (2.1)$$

ერთგვაროვანი სკალირება:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \alpha \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha x \\ \alpha y \end{pmatrix} \quad (2.2)$$

არაერთგვაროვანი სკალირება:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha x \\ \beta y \end{pmatrix} \quad (2.3)$$

მობრუნება:

ჩავთვალოთ რომ დადებითი მიმართულებაა საათის ისრის საწინააღმდეგო მიმართულება. ანუ თუ მობრუნება ხდება  $\alpha$  კუთხეზე მობრუნება გვაქვს საათის ისრის მოძრაობის საწინააღმდეგო მიმართულებით.

$$z_0 = d = 0$$

$$z_{n+1} = z_n^2 + c$$

$$x = r \cos \phi; y = r \sin \phi \quad (2.4)$$

$$x' = r \cos(\alpha + \phi); y' = r \sin(\alpha + \phi) \quad (2.5)$$

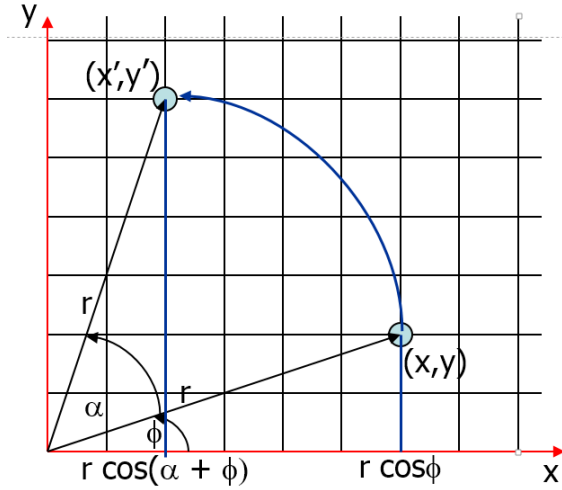
$$x' = r \cos \alpha \cos \phi - r \sin \alpha \sin \phi \quad (2.6)$$

$$= x \cos \alpha - y \sin \alpha \quad (2.7)$$

$$y' = r \sin \alpha \cos \phi + r \cos \alpha \sin \phi \quad (2.8)$$

$$= x \sin \alpha + y \cos \alpha \quad (2.9)$$

მიღებული შედეგი შეიძლება ჩაიწეროს მატრიცული სახით



სურ 2.1: 2D მობრუნება

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.10)$$

როგორც ვხედავთ გადატანისას თითოეულ კოორდინატს ემატება რიცხვი. სკალირებისას მრავლდება რიცხვზე და მობრუნებისას კოორდინატები მრავლდება მატრიცაზე. ანუ თუ გვექნება რაიმე გეომეტრიული ობიექტი და ობიექტი რიომელსაც შეგვიძლია ვუწოდოთ გარდაქმნა მოცემული ობიექტს უნდა ჰქონდეს გარდაქმნის ობიექტიც. გარდაქმნის ობიექტში უნდა იყოს სკალირების, მობრუნების და გადატანის წევრები და ფუნქციები.

მაგრამ რა ხდება როცა რამდენჯერმე ვასრულებთ გადატანას, მობრუნებას, სკალირებას ან ამ გარდაქმნების ნებისმიერ კომბინაციას?

გადატანა, სკალირების შემთხვევაში ადვილია, უბრალოდ შევცვლით შესაბამის მნიშვნელობებს. მობრუნების შემთხვევაშიც ყოველთვის შეიძლება ავაგოთ ახალი მატრიცა მოცემული ახალი კუთხისთვის, თუ წინასწარ ვიცით ეს კუთხე. მაგრამ არის შემთხვევები როცა ობიექტი გადის გარდაქმნის რამდენიმე ეტაპს და საჭიროა არა მარტო პირველადი მდგომარეობის ცოდნა არამედ ყველა ეტაპზე მდგომარეობის ცოდნაც. კითხვაზე პასუხი შემდეგია: მობრუნებების შემთხვევაში მრავალჯერადი მობრუნებებისას შესაბამისი მატრიცები უნდა გამრავლდეს. მაგ. თუ  $R_1(\alpha_1), R_2(\alpha_2), R_3(-\alpha_3)$  მობრუნებებია მიღებული შედეგი იქნება:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R_3(-\alpha_3)R_2(\alpha_2)R_1(\alpha_1) \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.11)$$

ანუ

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R(\beta) \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.12)$$

სადაც

$$R(\beta) = R_3(-\alpha_3)R_2(\alpha_2)R_1(\alpha_1) \quad (2.13)$$

მობრუნებების მატრიცების ნამრავლია. ანუ ჩვენ შეგვიძლია რამდენჯერმე კი არ გარდავქმნათ ობიექტი არამედ ავაგოთ ერთი მობრუნების მატრიცა რომელზეც გამრავლება მოგვცემს საჭირო შედეგს.

მაგრამ ისევ ცალკე გვაქვს გადატანები და სკალირებები. თუკი რომელიმე მობრუნებებს შორის მოხდა გადატანა და ან სკალირება ესეც გატვალისწინებული უნდა იყოს. მაგ. თუ 1 და მეორე

მობრუნების მერე მოხდა გადატა  $T_1$  და  $T_2$  მაშინ:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = R_3(-\alpha_3)T_2R_2(\alpha_2)T_1R_1(\alpha_1) \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.14)$$

და ყველაფერი სწორია, მაგრამ ასეთი მიდგომით შეუძლებელი იქნება გარდაქმნების (უფრო სწორედ საბოლოო შედეგის) ერთი მატრიცით წარმოდგენა.

თუმცა ასეთი მიდგომაც სწორია. მაგალითად თუ წერთ გეომეტრიის რედაქტორს მოცემული ობიექტს ექნება შესაბამისი მობრუნების, სკალირების, გადატანის წევრები. თქვენ ცვლით ამ წევრებს და ყოველი შეცვლისას ახატინებთ გრაფიკულ ძრავას ობიექტს. ამ შემთხვევაში მნიშვნელობა აქვს გარდაქმნების თანმიმდევრობას. ხშირად იღებენ ასეთ ტანმიმდევრობას:

- 1) სკალირება
- 2) მობრუნება
- 3) გადატანა.

ანუ თუ ჩავთვლით რომ რედაქტორში შექმნილი ან გარე ფაილიდან ცატვირთული ობიექტი არის კოორდინატის სათავეში, ჯერ ხდება ამ ობიექტის სკალირება, შემდეგ მობრუნება და მხოლოდ ამის შემდეგ გადატანა.

მოცემული მიდგომა არ იძლევა საშუალებას, მაგალითად ჯერ მოვახდინოთ გადატანა და შემდეგ მობრუნება მოცემული საკოორდინატო ღერძის/ან ღერძების გარშემო. ცხადია იგივე ეფექტი შეიძლება მიღწეულ იქნას ზემოთ მოყვანილი მეთოდითაც

ანუ შეიძლება მოინახოს ისეთი გადატანა და მობრუნება რომ ჯერ მოვაბრუნოთ ობიექტი საკოორდინატო სატავეში და შემდეგ გადავიტანოთ სასურველ ადგილას.

მაგრამ არსებობს სხვა მიდგომაც რომელიც საშუალებას იძლევა განვიხილოთ ობიექტის გარდაქმნა ერთი გარდაქმნის მატრიცის საშუალებით. ამ მატრიცაში „შენახულია“ როგორც მობრუნება ასევე გადატანაც და სკალირებაც.

## 2.2 ერთგვაროვანი კოორდინატები

გტქვათ გარდაქმნა მოიცემა შემდეგი სახით:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.15)$$

მაშინ თუ გარდაქმნის მატრიცის ელემენტებს ჩავწერთ როგორც მობრუნების ელემენტებს, გვექნება მობრუნების მატრიცა.

თუ გვსურს სკალირება მაშინ:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.16)$$

ანუ მობრუნება და სკალირება შეიძლება გვექონდეს ერთ მატრიცაში.

მაგრამ შეუძლებელია 2D გადატანა მოცემულ იყოს როგორც  $2 \otimes 2$  მატრიცაზე გამრავლება:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} dx \\ dy \end{pmatrix} \quad (2.17)$$

იმისათვის რომ გადატანაც ჩართული იყოს გარდაქმნის მატრიცაში და წარმოდგენილ იქნას როგორც მატრიცაზე გამრავლება შემოაქვთ ე.წ. ერთგვაროვანი კოორდინატები.

გუმატებთ სივრცეს კიდევ ერთ განზომილებას მესამე კოორდინატით  $w$  (ანუ 2D შემთხვევისათვის იქნება 3 განზომილება) მაშინ მოცემული წერტილი შეგვიძლია წარმოვიდგინოთ როგორც

$$(wx, wy, w)^T, w \neq 0$$

ან თუ დავანორმირებთ

$$(x, y, 1)^T, w = 1$$

ანუ 2D  $(x, y)$  წერტილი ერთგვაროვან 3D კოორდინატებში ესაა წრფე რომელიც გადის საკოორდინატო სისტემის სათავეზე, მაგრამ არ შეიცავს მას (პირობა  $w \neq 0$ )

ერთგვაროვან კოორდინატებში გარდაქმნა შეიძლება ჩაიწეროს შემდეგი სახით:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \quad (2.18)$$

მაგრამ რა სახე ექნება გარდაქმნის მატრიცას გადატანის, სკალირების, მობრუნების შემთხვევებში?

ადვილად შეიძლება შემოწმდეს რომ მატრიცა

$$T = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix}$$

გვაძლევს გადატანას ფორმ.(2.17)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.19)$$

ხოლო სკალირებას მოგვცემს მატრიცა

$$S = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.20)$$

შესაბამისად  $\alpha$  კუთხეზე მობრუნებას მოგვცემს მატრიცა

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

მობრუნებას ექნება სახე:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.21)$$

ანუ ზოგადად გვაქვს მატრიცა

$$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \quad (2.22)$$

სადაც  $a, e > 1$  იძლევა სკალირებას, ხოლო  $c, f$  გადატანას

ერთგვაროვან კოორდინატებში  $x$  გაჭიმვას (ეს არაა კარგი ტერმინი. ინგლისურად უწოდებენ „sharing“) გვაძლევს მატრიცა

$$SH_x = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x + ay \\ y \\ 1 \end{pmatrix}$$

ხოლო  $y$  გასწვრივ

$$SH_y = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y + bx \\ 1 \end{pmatrix}$$

#### დავალება:

1. ააგეთ მატრიცა რომელიც შედგება მობრუნება სკალარებისგან
2. ააგეთ მატრიცა რომელიც შედგება მობრუნება გადატანა მობრუნება
3. აჩვენეთ რომ მობრუნება გადატანა არაა იგივე რაც გადატანა მობრუნება
4. რა სახე ექნება არეკვლის მატრიცებს?
5. ვთქვათ ობიექტის კოორდინატებია  $P_x, P_y, P_z$  ააგეთ მატრიცა რომელიც შედგება შემდეგი გარდაქმნებისგან:
  - ა) გადაგვაქვს ობიექტი საკოორდინატო სისტემის სათავეში. T(-P)
  - ბ) ვაბრუნებთ რაღაც კუთხით  $R(\phi)$ .
  - გ) გადაგვაქვს უკან T(P);
6. ვთქვათ ობიექტის კოორდინატებია  $P_x, P_y, P_z$  ააგეთ მატრიცა რომელიც შედგება შემდეგი გარდაქმნებისგან:
  - ა) გადაგვაქვს ობიექტი საკოორდინატო სისტემის სათავეში. T(-P)
  - ბ) ვაკეთებთ სკალირებას S(a,b).
  - გ) გადაგვაქვს უკან T(P);

### 2.3 3D გარდაქმნები

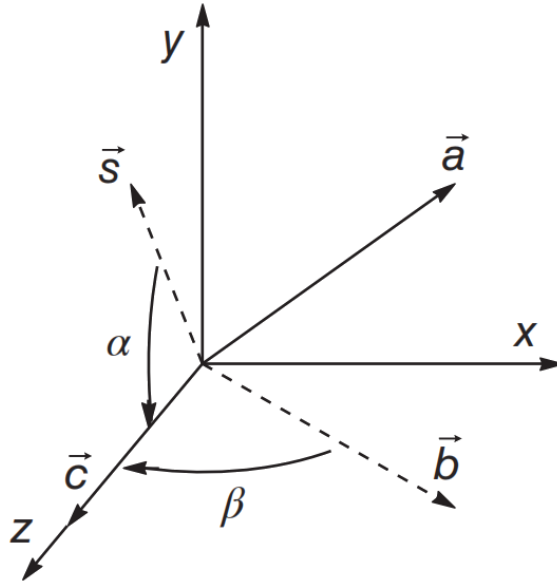
2D დან 3D გასავლისას გვექნება

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3D გარდაქმნებისთვის გვექნება  $4 \times 4$  მატრიცა: გადატანა:

$$T = \begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

სკალირება



სურ 2.2: მობრუნება

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

მობრუნება

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

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

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

3D ზოგად გარდაქმნის მატრიცას აქვს სახე:

$$T = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \quad (2.23)$$

## 2.4 ვექტორის გარშემო მობრუნება

ზემოთ მოყვანილ იყო მობრუნების ფორმულები საკოორდინატო დერძების გარშემო. ესლა განვიხილოთ მობრუნება ნებისმიერი  $\vec{a}$  ვექტორის გარშემო. ვუთქვათ  $\vec{a}$  არის ერთეულოვანი ვექ-

ტორი სურ.**2.2**. მოვაბრუნოთ ვექტორი  $\vec{x}$  ღერძის გარშემო საათის ისრის მოძრაობის საწინააღმდეგოდ  $\alpha$  კუთხით ისე რომ დაიღოს  $zx$  სიბრტყეში. მივიღებთ  $\vec{b}$  ვექტორს.  $\vec{a}$ -ს გვეგმილი  $zy$  სიბრტყეში აღვნიშნოთ  $\vec{s}$ . მოვაბრუნოთ  $\vec{b}$   $y$  გარშემო ისე რომ დამთხვევს  $z$  ღერძს. მიღებული ვექტორი ავლენიშნოთ  $\vec{c}$ . ეს ორი მობრუნება ადგენს ე.წ. შედგენილ გარდაქმნას რომელსაც  $\vec{a}$  გადაჰყავს  $\vec{c}$ . ჩვენი ამოცანაა ამ გარდაქმნის მატრიცის პოვნა.  $\vec{s} = (0, a_y, a_z)$ ,  $|\vec{s}| = g = \sqrt{a_y^2 + a_z^2}$

მაშინ

$$\cos\alpha = \frac{\vec{e}_z \cdot \vec{s}}{1 \cdot g} = \frac{a_z}{g} \quad (2.24)$$

შესაბამისად

$$\sin\alpha = \frac{a_y}{g} \quad (2.25)$$

მაშინ მობრუნების მატრიცას აქვს სახე:

$$M_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{a_z}{g} & -\frac{a_y}{g} \\ 0 & \frac{a_y}{g} & \frac{a_z}{g} \end{pmatrix} \quad (2.26)$$

$\vec{b}$  კომპონენტებია  $(v_x, 0, g)$  და  $|\vec{b}| = 1$  (რატომ?) ესლა უნდა დავწეროთ მობრუნების მატრიცა  $\vec{b} \Rightarrow \vec{c}$   $y$  ღერძის გარშემო. ვინაიდან:

$$\cos\beta = \frac{\vec{e}_z \cdot \vec{b}}{1 \cdot |\vec{b}|} = g \quad (2.27)$$

და

$$\sin\beta = \pm\sqrt{1-g^2} = \pm a_x \quad (2.28)$$

ამ შემთხვევაში უნდა ავიღოთ - ნიშანი ვინაიდან მობრუნება ხდება საათის ისრის მიმართულებით. მობრუნების მატრიცას ექნება სახე

$$M_y(\beta) = \begin{pmatrix} g & 0 & -a_x \\ 0 & 1 & 0 \\ a_x & 0 & g \end{pmatrix} \quad (2.29)$$

გარდაქმნის მატრიცა  $\vec{a} \Rightarrow \vec{c}$  იქნება

$$T(\vec{P}) = M_y(\beta) \cdot M_x(\alpha) \vec{P} \quad (2.30)$$

შებრუნებული გარდაქმნისთვის კუთხეებს ეცვლება ნიშანი  $\alpha \rightarrow -\alpha$  და  $\beta \rightarrow -\beta$

თუ გვსურს მობრუნება ნებისმიერ  $\theta$  მაშინ მოვაბრუნებთ წერტილს მიღებული  $\vec{c}$  ღერძის გარშემო  $\theta$  კუთხით ხოლო შემდეგ დავგვირდებთ შებრუნებული გარდაქმნები რომ  $\vec{c}$  ისევ დაემთხვეს  $\vec{a}$ . ანუ  $\vec{a}$  ვექტორის გარშემო ნებისმიერ  $\theta$  კუთხეზე მობრუნებას ექნება სახე

$$M = M_x^{-1}(\alpha) M_y^{-1}(\beta) M_z(\theta) M_y(\beta) M_x(\alpha) \quad (2.31)$$

თუ გადავამრავლებთ მატრიცებს და გავითვალისწინებთ რომ  $c = \cos\theta$ ,  $s = \sin\theta$  მივიღებთ

$$M = \begin{pmatrix} c + (1-c)a_x^2 & (1-c)a_x a_y - s a_z & (1-c)a_x a_z + s a_y \\ (1-c)a_x a_y + s a_z & c + (1-c)a_y^2 & (1-c)a_y a_z - s a_x \\ (1-c)a_x a_z - s a_y & (1-c)a_y a_z + s a_x & c + (1-c)a_z^2 \end{pmatrix} \quad (2.32)$$

ზემოთ მოყვანილი ფორმულა გამოვიყვანეთ როცა  $\vec{a}$  მიმართული იყო პირველ მეოთხედში (მსოფლიო საკოორდინატო სისტემის მიმართ). მიღებული ფორმულა სამართლიანია  $\vec{a}$ -ს სხვა ორიენტირებისათვისაც ოღონდ გათვალისწინებულ უნდა იქნას შემდეგი:

თუ კუთხეების  $\beta$  და  $\theta$  გვაქვს საათის ისრის მიმართულებით ან მეტია  $\pi/2$  მაშინ კოსინუსის და სინუსის ნიშნები იცვლება

$M_x(\alpha)$  მატრიცისათვის  $a_x$  გავლენა არ აქვს. კოსინუსს და სინუსს აქვს იგივე ნიშანი რაც  $a_z$  და  $a_y$ . თუ შევამოწმებთ ვნახავთ მიღებული ფორმულა კვლავ სამართლიანია.

ანალოგიურად  $M_x(\beta)$  თვის  $a_x$  შეიძლება იყოს უარყოფითი, მაგრამ თუ  $a_x$  უარყოფითია ეს ნიშნავს რომ ვექტორი მიმართულია x უარყოფითი მიმართულებით და  $yx$  მობრუნება უნდა მოხდეს საათის ისრის საწინააღმდეგო მიმართულებით  $y$  ღერძის გარშემო (ზემოთ კი გვექონდა საათის ისრის მიმართულებით), რაც შეცვლის სინუს ფუნქციის ნიშანს და მატრიცაც შესაბამისად შეიცვლება ვინაიდან  $a_x$  უარყოფითია. ანუ ჩვენს მიერ მიღებული პასუხი ზოგადაა და არაა დამოკიდებული  $\vec{u}$ -ს მიმართულებაზე.

ზემოთ მოყვანილი მობრუნების მატრიცა დაგვევიწყებოდა, როცა ავაგებთ ობიექტს რომელიც მაუსის მოძრაობით აბრუნებს ობიექტს ან სცენას. უფრო ზუსტად მაუსის კოორდინატების ცვლილებით ეკრანზე ავაგებთ ზემოთ მოყვანილ მობრუნების მატრიცას.

## 2.5 არეკვლა სიბრტყის მიმართ

ზემოთ მოყვანილი მეთოდი შეიძლება გამოყენებულ იქნას არა მარტო ნებისმიერი ღერძის გარშემო მობრუნებისათვის არამედ არეკვლისათვის სიბრტყის მიმართ.

$z$  ღერძის მიმართ არეკვლა მოიცემა მატრიცით

$$M_z(\text{არეკვლა}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (2.33)$$

ვთქვათ მოცემულია რაღაც სიბრტყე რომელიც გადის კოორდინატთა სათავეზე და რომლის ნორმალურია  $\vec{n}$  მაშინ ამ სიბრტყის მიმართ არეკვლის მატრიცა აიგება შემდეგი წესით:

მოვაბრუნოთ  $\vec{n}$  ისე რომ დაემთხვეს ერთ ერთ ღერძს. მოვახდინოთ არეკვლა დავაბრუნოთ ნორმალური უკან.

ზემოთ უკვე გავარჩიეთ როგორ მოვაბრუნოთ ვექტორი ისე რომ დაემთხვეს  $z$  ღერძს.  $\vec{a}$  ნაცვლად უნდა ავიღოთ  $\vec{n}$ .

**დავალება:** ააგეთ არეკვლის მატრიცა.

## 2.6 ნორმალის გარდაქმნა

ზშირად საჭიროა ობიექტის გარდაქმნებთან ერთად ნორმალის გარდაქმნაც. არის მეორე გზა გარდაქმნათ ობიექტი და გარდაქმნილი ობიექტისთვის გადავითვალოთ ნორმალური, მაგრამ თუ ნორმალური უკვე გამოთვლილი გვაქვს და გარდაქმნით ობიექტს, მაშინ არაა აუცილებელი ნორმალურების თავიდან გამოთვლა თუ ვიცით მათი გარდაქმნის წესი.

ვატყვათ ზედაპირი მოცემულია რაიმე სახით (მართკუთხედი, სამკუთხედი) და ვიცით ნორმალური მოცემულ წერტილში  $\vec{n}$  ვთქვათ ზედაპირზე მოცემულია ორი წერტილი  $\vec{P}, \vec{Q}$ . მაშინ

$$\vec{v} = \vec{P} - \vec{Q}$$

და

$$\vec{n} \cdot \vec{v} = \vec{n}^T \vec{v} = [n_1 n_2 n_3] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0$$

ვთქვათ  $M$  არის  $\vec{v}$  გარდაქმნის მატრიცა. საძიებელია გარდაქმნის მატრიცა  $L$  ისეთი რომ

$$L\vec{n} \cdot M\vec{v} = 0$$

ანუ

$$(L\vec{n})^T M\vec{v} = 0$$



ეს ტოლობა ძალაში იქნება ნებისმიერი ნორმალისთვის და ვექტორისთვის მაშინ როცა

$$(L\vec{n})^T M\vec{v} = \vec{n}^T L^T M\vec{v}$$

$$L^T M = 1$$

და

$$L = (M^{-1})^T$$

ორთოგონალური მატრიცებისათვის  $L = M$

## 2.7 საკოორდინატო სისტემა, წერტილი, ვექტორი

### 2.7.1 ვექტორი სხვადასხვა ბაზისში

ვექტორები ადგენენ წრფივ სივრცეს მასზე განმარტებული ოპერაციებით რაც რამდენჯერმე შეგვხვდა სხვა კურსებშიც და არ გავიმეორებთ. ყურადღებას მივაქცევთ იმ საკითხებს რის შესახებაც არ იყო საუბარი.

ვიცით რომ ნებისმიერი ვექტორი შეგვიძლია გავშალოთ როგორც საბაზისო ვექტორების რაღაც კომბინაცია.

$$\vec{v} = \sum_{i=1}^N \vec{e}_i c_i \quad (2.34)$$

აღსანიშნავია ორი რამ:

1. ზემოთ მოყვანილ გარდაქმნებში ვგულისხმობდით რომ გარდაიქმნება კოორდინატები

$$\vec{v} \rightarrow \vec{v}' = \vec{e}'_i M_{ij} c_j$$

სადაც  $\vec{e}'_i$  სტრიქონ ვექტორია<sup>1</sup>

2. არ გვიხსენებია ე.წ. ათვლის წერტილი და საკოორდინატო სისტემა, რაც ნისანვს რომ არავითარი აუცილებლობა არაა საბაზისო ვექტორები მოდებულ იყვნენ ერთ წერტილში.

ზემოთ მოყვანილ ტოლობას შეგვიძლია შევხედოთ როგორც საკოორდინატო ბაზისის გარდაქმნას

$$\mathcal{L}([\vec{e}]^t) = [\vec{e}]^t \cdot M$$

სადაც  $[\vec{e}]^t$  არის არა ერთი ვექტორი არამედ ბაზისი. 3D შემთხვევისათვის გვექნება:

$$[\mathcal{L}(\vec{e}_1), \mathcal{L}(\vec{e}_2), \mathcal{L}(\vec{e}_3)] = [\vec{e}_1, \vec{e}_2, \vec{e}_3] \cdot \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$$

ვთქვათ მოცემული გვაქვს ვექტორი  $\vec{v} = [\vec{b}]^t c$  ბაზისში  $[\vec{b}]^t$ , და გავქვს ბაზისი  $[\vec{a}]^t$  რომელიც თავის მხრივ მოიცემა  $[\vec{b}]^t$  ბაზისის გარდაქმნით

$$[\vec{a}]^t = [\vec{b}]^t \cdot M$$

ანუ

$$[\vec{a}]^t \cdot M^{-1} = [\vec{b}]^t$$

მაშინ მოცემული  $\vec{v}$  შეგვიძლია ჩავწეროთ როგორც

$$\vec{v} = [\vec{b}]^t c = [\vec{a}]^t \cdot M^{-1} c$$

ეს არაა გარდაქმნის ფორმულა, უბრალოდ ერთი და იგივე ვექტორი ცაწერილია სხვადასხვა ბაზისში. საკოორდინატო ვექტორი  $c$  წარმოადგენს  $\vec{v}$  ვექტორს  $[\vec{b}]^t$  საკოორდინატო ბაზისის მიმართ და  $M^{-1}c$  საკოორდინატო ვექტორი წარმოადგენს იგივე ვექტორს ოღონდ  $[\vec{a}]^t$  ბაზისის მიმართ.

<sup>1</sup> ალენიშნავთ სტრიქონს

მაგალითი: ვთქვათ ვექტორი მოცემულია მსოფლიო ბაზისის  $[\vec{b}]^t$  მიმართ, და  $[\vec{a}]^t$  ობიექტთან დაკავშირებული ბაზისი ემთხვეოდა მსოფლიო ბაზისს. შემდეგ მოვაბრუნეთ ობიექტი ანუ მოვაბრუნეთ მასთან დაკავშირებული ბაზისი. მაშინ იგივე წერტილს გარდაქმნილი (მობრუნებული) ბაზისის მიმართ ექნება კოორდინატები  $M^{-1}c$ , სადაც  $M$  არის მობრუნების მატრიცა

### 2.7.2 წერტილები და საკოორდინატო სისტემა

როდესაც ვსაუბრობთ ვექტორებზე შეგვიძლია განვასხვავოთ ორი შემთხვევა. ვუყუროთ ვექტორს როგორც რაღაც წერტილის რადიუს ვექტორს, მაშინ ის გვიჩვენებს მდებარეობას სივრცეში და ვუყუროთ ვექტორს როგორც წანაცვლებას ერთი წერტილიდან მეორისკენ. მაშინ ვექტორი გვიჩვენებს წანაცვლებას ორ წერტილს შორის.

წერტილებს ავლენიშნავთ დიდი ასო ნიშნებით ხოლო ვექტორებს პატარაით. მაშინ თუ გვაქვს ორი წერტილი  $\vec{Q}$  და  $\vec{P}$

$$\vec{P} - \vec{Q} = \vec{v} \quad (2.35)$$

და

$$\vec{Q} + \vec{v} = \vec{P}$$

როგორც ზემოთ ვნახეთ იმისათვის რომ აღვწეროთ გადატანა და მობრუნება ერთი მატრიცით და გვჭირდა დამატებითი კოორდინატის შემოტანა და შესაბამისად მატრიცის განზომილების გაზრდა.

გარდაქმნებს რომლებიც აღწერენ გადატანას და მობრუნებას როგორც ერთს აფინური გარდაქმნები ეწოდებათ. მიღებულ სივრცეს აფინური სივრცე. აფინურ სივრცეში წერტილი აღიწერება როგორც საკოორდინატო სათავეს  $\vec{O}$  დამატებული ვექტორების რაღაც წრფივი კომბინაცია.

$$\vec{Q} = \vec{O} + c_i \vec{e}_i$$

თუ დავუშვებთ რომ  $1 \cdot \vec{O} = \vec{O}$

$$[\vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{O}] \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} = \vec{f}^t c$$

სადაც  $\vec{f}^t$  აღნიშნავს არა უბრალოდ საბაზისო ვექტორებს არამედ საბაზისო ვექტორებს დამატებული კიდევ ერთი წერტილი (ათვლის წერტილი) ანუ საკოორდინატო სისტემას („ფრეიმს“, *frame*). სტრიქონს

$$[\vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{O}]$$

აფინურ საკოორდინატო სისტემას უწოდებენ. და ვთვლით რომ საბაზისო ვექტორები მოდებულია ათვლის წერტილში. ეს სტრიქონი ბაზისის მსგავსია, მაგრამ დამატებით დამატებული აქვს ერთი წერტილი — საკოორდინატო სათავე.

იმისათვის რომ აღვწეროთ წერტილი აფინურ სივრცეში ჩვენ ვიყენებთ საკოორდინატო 4-ვექტორ სვეტს სადაც პირველი სამი სიდიდე არის წერტილის კოორდინატები, ხოლო მეოთხე სიდიდე ტოლია ერთისა.

იმისათვის რომ აღვწეროთ ვექტორი აფინურ სივრცეში ჩვენ ვიყენებთ 4-ვექტორ სვეტს სადაც პირველი სამი სიდიდე არის ვექტორის პროექციები, ხოლო მეოთხე სიდიდე ტოლია ნულისა იხ. განტ (2.35).

მაშინ ზოგადი გარდაქმნას (2.23) აფინური სივრცისთვის ექნება სახე

$$T = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.36)$$

წერტილის აფინური გარდაქმნა იქნება

$$[\vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{O}] \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} \Rightarrow [\vec{e}'_1, \vec{e}'_2, \vec{e}'_3, \vec{O}'] \cdot \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} \quad (2.37)$$

ანუ

$$\vec{f}^t c = \vec{f}^t M c \quad (2.38)$$

თუ ჩავწერთ წერტილის კოორდინატებისათვის საკოორდინატო სისტემის გარეშე გვექნება

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.39)$$

ხოლო თუ გვინდა ჩავწეროთ ბაზისის გარდაქმნები მაშინ გვჭირდება საკოორდინატო სათავის გათვალისწინება. ვინაიდან აფინური გარდაქმნებისას ზოგადად გარდაიქმნება საკოორდინატო სათავეც (გადატანა), რაც მოცემულია ქვემოთ

$$[\vec{e}'_1, \vec{e}'_2, \vec{e}'_3, \vec{O}'] = [\vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{O}] \cdot \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.40)$$

გველისხმობთ რომ  $0 \cdot \vec{O} = \vec{0}$  ანუ

$$\vec{f}^t \Rightarrow \vec{f}^t M \quad (2.41)$$

## 2.8 კოორდინატები სხვადასხვა სისტემაში

შეიძლება გვექონდეს რამდენიმე საკოორდინატო სისტემა და ყურადღებით უნდა ვიყოთ თუ რომელი სისტემის მიმართ ხორციელდება გარდაქმნა

გარდაქმნის მატრიცის მოცემა ჯერ კიდევ არ ნიშნავს ობიექტის მდებარეობის და ორიენტაციის განსაზღვრას. საჭიროა საკოორდინატო სისტემა თუ რის მიმართ ხდება ესა თუ ის გარდაქმნა და/ან მოცემულია რაღაც კოორდინატები.

ვტყვით მოცემულია წერტილი  $\vec{P}$  და გარდაქმნა

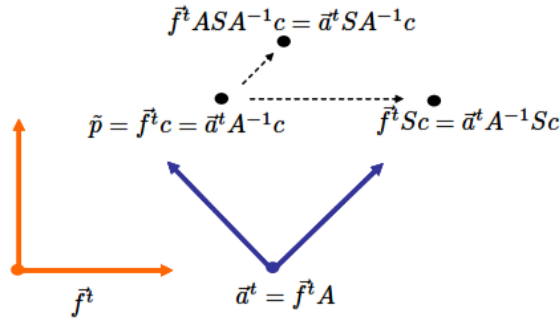
$$S = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.42)$$

ვთქვათ მოცემულია რაღაც საკოორდინატო სისტემა  $\vec{f}^t$  რომლის მიმართ მოცემულია კოორდინატები. მაშინ  $\vec{P} = \vec{f}^t c$ . თუ გამოვიყენებთ ზემოთ მოყვანილ გარდაქმნას  $\vec{P} \Rightarrow \vec{P}'$  და  $\vec{f}^t c \Rightarrow \vec{f}^t S c$ . რაც ნიშნავს რომ მოცემულ ბაზისში ხდება ობიექტის სკალირება  $x$  გასწვრივ.

ვთქვათ ავირჩიეთ სხვა საკოორდინატო სისტემა  $\vec{a}^t$  დავუშვათ ეს ორი საკოორდინატო სისტემა დაკავშირებულია გარდაქმნით  $\vec{a}^t = \vec{f}^t A$ . მოცემული წერტილი შეგვიძლია გამოვსახოთ ამ ახალ საკოორდინატო სისტემაში როგორც

$$\vec{P} = \vec{f}^t c = \vec{a}^t d$$

სადაც  $d = A^{-1} c$  ეს არის კოორდინატები  $\vec{a}^t$  საკოორდინატო სისტემის მიმართ. ეხლა თუ გარდაქ-



სურ 2.3: სკალირება სხვადასხვა სისტემის მიმართ

მნას გამოვიყენებთ  $\vec{a}^t$  სისტემის მიმართ გვექნება

$$\vec{a}^t d \Rightarrow \vec{a}^t S d$$

ამ შემთხვევაში მოხდა იგივე  $\vec{P}$  წერტილის სკალირება  $x$  გასწვრივ ოღონდ  $\vec{a}^t$ -ს მიმართ. ეს ცხადია განსხვავებული გარდაქმნაა. იხ. სურ. 2.3

აღსანიშნავია შემდეგი წესი (ე.წ. მარცხენა წესი): წერტილი გარდაქმნილია იმ საკოორდინატო სისტემის მიმართ რომელიც დგას გარდაქმნების მარცხნივ. ამ წესის გამოყენებით ქვემოთ მოყვანილი ტოლობა

$$\vec{P} = \vec{f}^t c \Rightarrow \vec{f}^t S c$$

წაიკითხება როგორც:

„ $\vec{P}$  გარდაქმნილია  $S$  გარდაქმნით  $\vec{f}^t$  ბაზისის მიმართ.“ ხოლო ტოლობა:

$$\vec{P} = \vec{a}^t A^{-1} c \Rightarrow \vec{a}^t S A^{-1} c$$

„ $\vec{P}$  გარდაქმნილია  $S$  გარდაქმნით  $\vec{a}^t$  ბაზისის მიმართ.“

იგივე წესი შეგვიძლია გამოვიყენოთ საკოორდინატო სისტემის მიმართაც.

$$\vec{f}^t \Rightarrow \vec{f}^t S$$

„ $\vec{f}^t$  გარდაქმნილია  $S$  გარდაქმნით  $\vec{f}^t$  ბაზისის მიმართ.“

ხოლო

$$\vec{f}^t = \vec{a}^t A^{-1} \Rightarrow \vec{a}^t S A^{-1}$$

„ $\vec{f}^t$  გარდაქმნილია  $S$  გარდაქმნით  $\vec{a}^t$  ბაზისის მიმართ.“

## 2.9 გარდაქმნები შუალედური საკ. სისტემის გამოყენებით

ხშირად არის შემთხვევა, როცა  $M$  მატრიცით მოცემული სისტემა  $\vec{f}^t$  გვსურს გარდავქმნათ რაღაც სხვა  $\vec{a}^t$  სისტემის მიმართ. მაგალითად ვიყენებთ დედამიწასთან დაკავშირებულ სისტემას და შემდგომში გვსურს ამ სისტემის მოძრაობა მზის გარშემო.

ამ ამოცანის შესრულება ადვილია თუ ვიცით როგორ არის დაკავშირებული ეს ორი სისტემა. ანუ

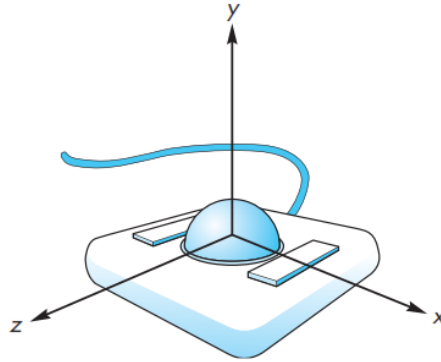
$$\vec{a}^t = \vec{f}^t A$$

გარდაქმნილი სისტემა შეგვიძლია გამოვსახოთ როგორც

$$\vec{f}^t \tag{2.43}$$

$$= \vec{a}^t A^{-1} \tag{2.44}$$

$$\Rightarrow \vec{a}^t M A^{-1} \tag{2.45}$$



სურ 2.4

$$= \vec{f}^t A M A^{-1} \quad (2.46)$$

გამოვსახეთ  $\vec{f}^t$   $\vec{a}^t$ -ს საშუალებით. შემდეგ ვიყენებთ ე.წ. მარცხენა წესს ანუ ვაკეთებთ სისტემის გარდაქმნას  $\vec{a}^t$ -ს მიმართ უკანასკნელ განტოლებაში კი უბრალოდ დროებითი სისტემის ნაცვლად კვლავ ვუბრუნდებით აღნიშვნებს ძირითად სისტემის მიმართ.

ანუ მაგალითად თუ  $\vec{f}^t$  არის მზესტან დაკავშირებული სისტემა  $\vec{a}^t$  დედამიწასთან, მაშინ  $\vec{a}^t M A^{-1}$  იქნება მაგ. მთვარის მოძრაობა დედამიწის გარშემო(დედამიწის სისტემის მიმართ), ხოლო  $\vec{f}^t A M A^{-1}$  მთვარის მოძრაობა მზის გარშემო.

როცა გვაქვს საკოორდინატო სისტემის მრავალჯერადი გარდაქმნები

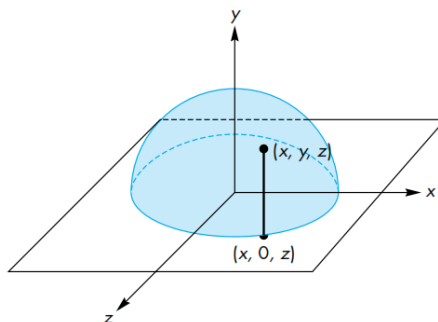
$$\vec{f}^t \Rightarrow \vec{f}^t R_1 R_2 R_3 \quad (2.47)$$

ინტერპრეტირება შეიძლება ორგვარად. თუ ვკითხულობთ მარცხნიდან მარჯვნივ ეს გარდაქმნები ნიშნავს: „ $\vec{f}^t$  გარდაქმნილია  $R_1$  გარდაქმნით  $\vec{f}^t$  ბაზისის მიმართ და ყოველი ახალი გარდაქმნა ხდება ახალი ბაზისის მიმართ.“ თუ წავიკითხავთ მარცხნიდან მარჯვნივ მაშინნიშნავს რომ გარდაქმნები შესრულებულია გლობალური საკოორდინატო სისტემის მიმართ.

## 2.10 მაუსით ბრუნვა

განვიხილოთ მაუსის საშუალებით სცენის(ან ობიექტის) ბრუნვა და გადატანა. ტრეკბოლის მოწყობილობას აქვს შემდეგი სახე სურ.2.4

წარმოვიდგინოთ, რომ ეკრანზე(უფრო სწორედ პროგრამის ხედვის პორტზე) გვაქვს ერთეულოვანი ნახევარსფერო და გვაქვს ეკრანის კოორდინატების ასახვა სფეროზე, როგორც ეს წარმოდგენილია სურათზე 2.5. ქვემოთ აღწერილ ალგორითმს ეწოდება ვირტუალური ტრეკბოლის ალგორითმი.



სურ 2.5: ეკრანის კოორდინატების ასახვა ვირტუალურ სფეროზე

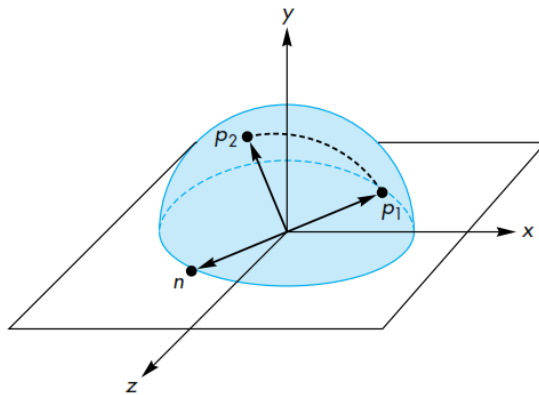
მაშინ ასეთ წარმოდგენაში გვექნება

$$x^2 + y^2 + z^2 = 1 \quad (2.48)$$

და მოცემული წერტილისათვის სიბრტყეზე კოორდინატებით  $(x, 0, z)$  სფეროზე შესაბამისი წერტილი იქნება  $(x, y, z)$  სადაც

$$y = \sqrt{1 - x^2 - z^2} \quad (2.49)$$

მაუსის მცირე გადაადგილებას ეკრანზე შეესაბამება ახალი წერტილი. შესაბამისად გვაქვს „ძველი“ (გადაადგილებამდე) წერტილი და ახალი წერტილი სფეროზე. წარმოვიდგინოთ, რომ მაუსის მოძრაობისას ბრუნავს სფერო, რაც ნიშნავს რომ სფერო მობრუნდა ისე, რომ „ძველი“ (გადაადგილებამდე) წერტილი სფეროზე გადავიდა ახალ წერტილში. ამოცანაა ვიპოვნოთ რა ღერძის მიმართ და რა კუთხით მობრუნდა სფერო? სურ.2.6 ჩანს, რომ სფეროზე მოცემული  $\vec{P}_1$  და  $\vec{P}_2$



სურ 2.6: სფეროს მობრუნება

წერტილებისათვის შეგვიძლია განვმარტოთ ვექტორი რომელიც  $\vec{OP}_1\vec{P}_2$  სიბრტყის მართობული იქნება:

$$\vec{n} = \vec{p}_1 \times \vec{p}_2 \quad (2.50)$$

$\vec{n}$  მოდული კი ტოლი იქნება ამ ვექტორებს შორის კუთხის სინუსის, ვინაიდან სფერო ერთეულრვანია.

ანუ ნაპოვნი გვაქვს კუთხე და ვექტორი რომლის მიმართაც მოხდა მობრუნება. ამის შემდეგ ავაგებთ მობრუნების მატრიცას და მივაწვდით მას *OpenGL* ძრავას.

თუ გვსურს რაიმე მონიშნული ობიექტის მობრუნება მაშინ მიღებული მატრიცა იქნება ობიექტის მობრუნების მატრიცა.

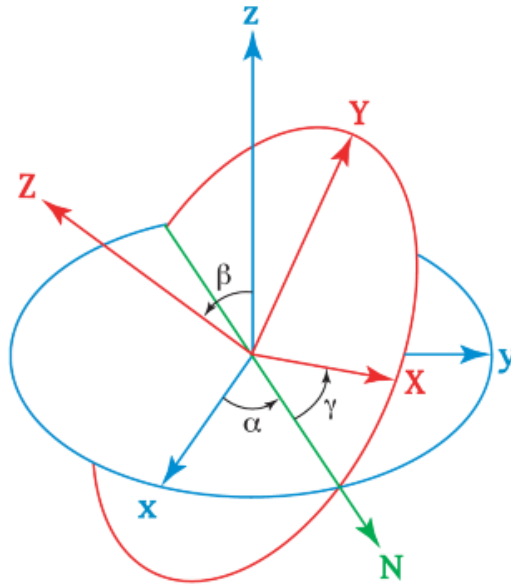
იმის მიხედვით თუ როგორ ხდება მობრუნების კუთხის გამოთვლა და ეკრანის წერტილების ასახვა სფეროზე გვაქვს რამდენიმე ალგორითმი (შემოტანილ იქნა სხვადასხვა ავტორების მიერ).

ცხადია ეკრანზე იქნება ისეთი კოორდინატებიც რომლებიც ვერ აისახება სფეროზე. ზოგიერთ იმპლემენტაციაში ასეთი წერტილებისათვის  $y$  კოორდინატს იღებენ ნულს. ზოგიერთ შემთხვევაში კი იმ არეში სადაც სფერო ვერ ფარავს ეკრანის სიბრტყეს სფეროს "აგრძელებენ" როგორც ჰიპერბოლოიდს და  $y$  იღებენ ჰიპერბოლაზე (ჩენის ტრაკბოლის ალგორითმი).

## 2.11 მობრუნების პარამეტრიზაცია

ცნობილია რომ მობრუნების პარამეტრიზაცია შესაძლებელია ე.წ. ეილერის კუთხეებით, მაგრამ ეილერის კუთხეები არ იზღვევა საშუალებას წარმოვადგინოთ მობრუნება როგორც „ელემენტარული მობრუნებების“ ერთობლიობა, როგორც მაგალითად ერთი წერტილიდან მეორეში გადატანა შეიძლება წარმოვადგინოთ როგორც ელემენტარული გადატანების ერთობლიობა. უფრო მეტიც ეილერის კუთხეებს აქვს ნაკლი რომელიც ცნობილია "Gimbal Lock" სახელით (ბრუნვის ერთი თავისფლების ხარისხის „დაკარგვა“. იხ. [https://en.wikipedia.org/wiki/Gimbal\\_lock](https://en.wikipedia.org/wiki/Gimbal_lock) და

<https://www.youtube.com/watch?v=zc8b2Jo7mno>) ზემოთ მოყვანილი ტრეკბოლის ალგორითმიც არ იძლევა ამის საშუალებას. 1985 წელს კენ შოგმაკემ SIGGRAPH (Special Interest Group



სურ 2.7: ეილერის კუთხვები

on Computer Graphics and Interactive Techniques) კონფერენციაზე წარმოადგინა მეთოდი როგორ შეიძლება აღიწეროს მობრუნებები კვატერნიონების საშუალებით.

### 2.12 კვატერნიონი

კვატერნიონი შეიძლება განმარტებულ იქნას სხვადასხვა გზით. ჰამილტონმა კვატერნიონი სემიოტანა როგორც კომპლექსური რიცხვის განზოგადება შემდეგი სახით:  $w + ix + jy + kz$ , სადაც  $i^2 = j^2 = k^2 = -1$  და  $i \cdot j = k = -j \cdot i$ .  $w, x, y$  და  $z$  ნამდვილი რიცხვებია. (ჰამილტონის პატივსაცემად მათემატიკაში კვატერნიონები აღინიშნება როგორც  $H$ ). კვატერნიონი ასევე შეიძლება შემოტანილ იქნას როგორც ოთხი რიცხვის ერთობლიობა რომელსაც აქვს გამრავლების, გაყოფის და ა.შ. საკუთარი წესები, მაგრამ ადვილი შესამჩნევია რომ ამ ოთხი რიცხვიდან სამი ბუნებრივად ჯგუფდება ერთად და მას ჰამილტონმა უწოდა ვექტორი, ხოლო მეოთხე ( $w$ ) არის ცალკე—მას ჰამილტონმა უწოდა სკალარი. მოგვიანებით გიბსმა ჰამილტონის კვატერნიონების თეორიიდან ცალკე გამოყო ობიექტი ვექტორი და განავითარა ვექტორული არითმეტიკა. დღეს გამოყენებული ე.წ. ვექტორული და სკალარული ნამრავლის ცნებები ვექტორებისათვის შემოტანილია სწორედ გიბსის მიერ.

ბუნებრივია, რომ კვატერნიონი განმარტებულ იქნას როგორც ვექტორის და სკალარის ერთობლიობა  $(w, \vec{v})$  (სადაც  $\vec{v}(x, y, z)$ ). მაშინ ნადვილი რიცხვი შეიძლება წარმოვადგინოთ როგორც  $(w, \vec{0})$  და ვექტორი შეგვიძლია რამოვადგინოთ როგორც  $(0, \vec{v})$

### 2.13 კვატერნიონის თვისებები

**ნამრავლი** განმარტებულია როგორც

$$q_1 q_2 = (w_1, \vec{v}_1)(w_2, \vec{v}_2) = (w_1 w_2 - \vec{v}_1 \cdot \vec{v}_2, w_1 \vec{v}_2 + w_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2) \quad (2.51)$$

**შეუღლებული**

$$q^* = (w, -\vec{v}) \quad (2.52)$$

**ნორმა**

$$\|q\| = q q^* = q^* q = w^2 + \vec{v} \cdot \vec{v} \quad (2.53)$$

შებრუნებელი  $q^{-1}q = 1$ . თუ გამოვიყენებთ შეუღლებული კვატერნიონის განმარტებას

$$q^{-1} = q^* / \|q\| \quad (2.54)$$

**გაყოფა** გაყოფა შეგვიძლია ვუწოდოთ შებრუნებულზე გამრავლებას:

$$q_1 / q_2 = q_1 q_2^{-1} \quad (2.55)$$

თუ კვატერნიონის სკალარული ნაწილი ნულია მას „წმინდა კვატერნიონი“ ეწოდება (წმინდა წარმოსახვითის მსგავსად). კვატერნიონს რომლის ნორმა ერთია ერთეულოვანი კვატერნიონი ეწოდება. ერთეულოვანი კვატერნიონისთვის  $q^{-1} = q^*$  რაც კომპონენტების ენაზე ნიშნავს

$$w^2 + \vec{v} \cdot \vec{v} = 1 \quad (2.56)$$

ანუ ერთეულოვანი კვატერნიონი შეგვიძლია წარმოვიდგინოთ, როგორც

$$(\cos\theta, \sin\theta\vec{u}) \quad (2.57)$$

სადაც  $\vec{u}$  ერთეულოვანი ვექტორია.

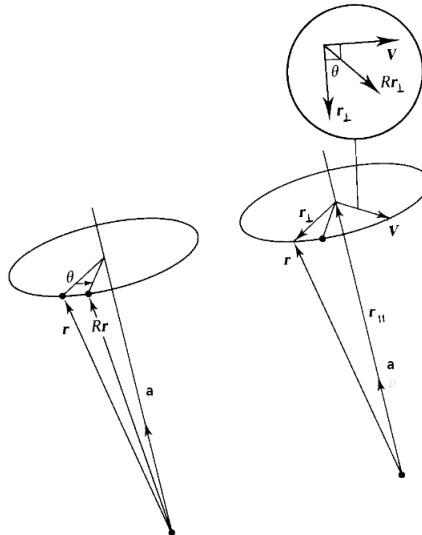
ზემოთ გამოვიყვანეთ მოცემული ვექტორი გარშემო ნებისმიერ კუთხეზე მობრუნების მატრიცა. მიღებული მატრიცის სახე მოსახერხებელია პროგრამაში შესატანად. ეს მატრიცა ავსავთ როგორც ორი დამხმარე მობრუნება შემდეგ მობრუნება საჭირო კუთხეზე და ორი დამხმარე მობრუნების შებრუნებელი მობრუნება, რომ მობრუნების ღერძი კვლავ გადაგვეტანა საწყის მდგომარეობაში.

სანამ ვაჩვენებდეთ, რომ ერთეულოვანი კვატერნიონით შესაძლებელია მობრუნების გარდაქმნის აგება გამოვიყვანოთ (2.32) ფორმულა განსხვავებული გზით.

## 2.14 ვექტორის გარშემო მობრუნება. კინემატიკური გზა

ვთქვათ წერტილი ბრუნავს მყისიერი  $\vec{a}$  ღერძის გარშემო რადიუს ვექტორი იქნება  $\vec{r}$ . გავშალოთ ორ მდგენელად: ერთი ბრუნვის ღერძის პარალელური  $r_{\parallel}$

$$r_{\parallel} = (\vec{a} \cdot \vec{r})\vec{a} \quad (2.58)$$



სურ 2.8: მობრუნება ვექტორის გარშემო

მეორე პერპენდიკულარული  $r_{\perp}$ :

$$r_{\perp} = \vec{r} - (\vec{a} \cdot \vec{r})\vec{a} \quad (2.59)$$



ცხადია პარალელური მდგენელი ბრუნვისას არ იცვლება. იცვლება მხოლოდ პერპენდიკულარული მდგენელი (ბრუნავს წერტილზე).  $R$  მობრუნებისას  $\vec{r}$  გადადის  $R\vec{r}$  და  $\vec{r}_\perp \rightarrow R\vec{r}_\perp$ . ვიპოვოთ  $\vec{r}_\perp$  მართობული ვექტორი ბრუნვის სიბრტყეში.

$$\vec{V} = \vec{a} \times \vec{r}_\perp = \vec{a} \times \vec{r} \quad (2.60)$$

მაშინ მობრუნებული ვექტორი  $R\vec{r}_\perp$  შეგვიძლია გავშალოთ ორ მდგენელად  $\vec{r}_\perp$  გასწვრივი და  $\vec{V}$  გასწვრივი:

$$R\vec{r}_\perp = \cos\theta \cdot \vec{r}_\perp + \sin\theta \cdot \vec{V} \quad (2.61)$$

ზემოთ მოყვანილი ფორმულების გათვალისწინებით მობრუნებული რადიუს ვექტორისთვის  $R\vec{r}$  გვექნება

$$\begin{aligned} R\vec{r} &= R\vec{r}_\parallel + R\vec{r}_\perp \\ &= R\vec{r}_\parallel + \cos\theta \cdot \vec{r}_\perp + \sin\theta \cdot \vec{V} \\ &= (\vec{a} \cdot \vec{r})\vec{a} + \cos\theta \cdot (\vec{r} - (\vec{a} \cdot \vec{r})\vec{a}) + \sin\theta \cdot \vec{a} \times \vec{r} \\ &= \cos\theta \cdot \vec{r} + (1 - \cos\theta)(\vec{a} \cdot \vec{r})\vec{a} + \sin\theta \cdot \vec{a} \times \vec{r} \end{aligned} \quad (2.62)$$

## 2.15 მობრუნება და კვატერნიონი

როგორც ავღნიშნეთ ვექტორი შეგვიძლია განვიხილოთ როგორც „წმინდა კვატერნიონი“  $p(0, \vec{r})$ . განვიხილოთ ნამრავლი (უფრო ზუსტი ტერმინი იქნებოდა გარდაქმნა)

$$R_q(p) \equiv qpq^{-1} \quad (2.63)$$

სადაც  $q$  ერთეულოვანი კვატერნიონია  $p(w, \vec{v})$ . გვექნება:

$$R_q(p) = (0, (w^2 - \vec{v} \cdot \vec{v})\vec{r} + 2\vec{v}(\vec{v} \cdot \vec{r}) + 2w\vec{v} \times \vec{r}) \quad (2.64)$$

რადგან  $q$  ერთეულოვანი კვატერნიონია, შეგვიძლია წარმოვადინოთ როგორც (2.57) მაშინ გვექნება:

$$\begin{aligned} R_q(p) &= (0, (\cos^2\theta - \sin^2\theta)\vec{r} + 2\sin^2\theta\vec{a}(\vec{a} \cdot \vec{r}) + 2\cos\theta\sin\theta\vec{a} \times \vec{r}) \\ &= (0, \cos 2\theta\vec{r} + (1 - \cos 2\theta)(\vec{a} \cdot \vec{r})\vec{a} + \sin 2\theta\vec{a} \times \vec{r}) \end{aligned} \quad (2.65)$$

თუ შევადარებთ ამ ფორმულას (2.62) ფორმულას ვნახავთ, რომ განსხვავება არის მხოლოდ კუთხის მნიშვნელობაში რაც ნიშნავს, რომ (2.63) წარმოადგენს მობრუნებას  $2\theta$  კუთხით. ანუ  $\vec{a}$  ვექტორის გარშემო მობრუნება  $\theta$  კუთხით ეს იგივეა რაც (2.63) ოპერაცია ერთეულოვანი კვატერნიონით  $(\cos\frac{1}{2}\theta, \sin\frac{1}{2}\theta\vec{a})$ .

კვატერნიონების ენაზე მობრუნება შეგვიძლია დავახასიათოთ ოთხი პარამეტრით:  $\cos\frac{1}{2}\theta, \sin\frac{1}{2}\theta a_x, \sin\frac{1}{2}\theta a_y, \sin\frac{1}{2}\theta a_z$

## 2.16 მაუსით ბრუნვა კვატერნიონების გამოყენებით

მაუსით მობრუნების აღწერა კვატერნიონების ენაზე ზუსტად ისევე ხდება როგორც ზემოთ აღწერილ ტრეკბოლის მეთოდში. ოღონდ მობრუნების მატრიცის ნაცვლად უნდა ავაგოთ შესაბამისი ვტარნიონი. და „ძველი“ (მობრუნებამდე) კვატერნიონი უნდა გავამრავლოთ ახალზე. შემდეგ ეს ახალი კვატერნიონი უნდა გამოვიყენოთ კვატერნიონული მობრუნების მატრიცის ასაგებად და უნდა გავითვალისწინოთ ეკრანული კოორდინატების საშუალებით მიღებულ კუთხეს შეესაბამება ორჯერ მეტ კუთხეზე მობრუნება. ამ მეთოდს უწოდებენ მაუსის მართვას arcball საშუალებით.