

ინგა გაბისონია, სოფია შენგელია

# კომპიუტერული გრაფიკა

| ნაწილი



ინგა გაბისონია

სოფია შენგელია

# კომპიუტერული გრაფიკა

## I ნაწილი

თბილისი 2017

„კომპიუტერული გრაფიკის“ პირველ ნაწილში განხილულია შესაბამისი დარგის თეორიული საფუძვლები, მოყვანილია მისი მათემატიკური აპარატი, განხილულია ფერთა მოდელები და რასტრული გრაფიკის ძირითადი პრინციპები. პრაქტიკულ ნაწილში მოყვანილია სხვადასხვა ტიპის პრაქტიკული სავარჯიშოები და დავალებები გრაფიკული რედაქტორის Adobe Photoshop-ის (ვერსია CS6 ან მისი ანალოგი) გამოყენებით. მოყვანილი მასალა სასარგებლო იქნება ბაკალავრიატის სტუდენტებისათვის და ყველასთვის, ვისაც აინტერესს კომპიუტერული გრაფიკის მრავალფეროვანი სამყარო.

რეცენზენტი - სოხუმის სახელმწიფო უნივერსიტეტის

მათემატიკისა და კომპიუტერულ მეცნიერებათა ფაკულტეტის

ასოცირებული პროფესორი, ინფორმატიკის დოქტორი ნანა ზენიძე

ISBN 978-9941-0-9598-6 (ყველა ნაწილის)

ISBN 978-9941-0-9599-3 (პირველი ნაწილის)

## კომპიუტერული გრაფიკის ძირითადი ცნებები

კომპიუტერული გრაფიკა გულისხმობს გრაფიკული ფორმატის ფაილების შექმნას, რედაქტირებას და ინტერპრეტაციას კომპიუტერის მეშვეობით, გრაფიკული რედაქტორების დახმარებით. თავდაპირველად, გრაფიკული ფორმატის ფაილების შექმნა ხდებოდა სიმბოლოების საშუალებით, ფურცელზე ამობეჭდვის დროს გამოსახულება იქმნებოდა სხვადასხვა სიმბოლოების: ვარსკვლავების, წერტილების, ასოების და ა.შ. მეშვეობით. ამ შემთხვევაში იქმნებოდა მოზაიკის მაგვარი გამოსახულებები. ასე ქმნიდნენ ფუნქციების გრაფიკებს, სითხეების და აირების მოძრაობის ტრაექტორიებს, ელექტრული და მაგნიტური ველების ნახაზებს, ზოგიერთი ცნობილი სურათის რეპროდუქციებს. მაშინ ასეთ გამოსახულებებს მანქანურ ფერწერას ეძახდნენ. შემდგომში შეიქმნა სპეციალური მოწყობილობები გრაფიკების აგებისათვის - პლოტერები. ასეთი მოწყობილობები სპეციალური კალმითა და მელნით ქაღალდზე გამოსახავდნენ ფუნქციების გრაფიკებს, დიაგრამებს, ტექნიკურ სურათებს და ა.შ. ნამდვილი გრაფიკული რევოლუცია მოხდა, როცა გამოიგონეს ე.წ. გრაფიკული რეჟიმის მქონე ეკრანები. ანუ გრაფიკულ ეკრანზე შეიძლება იგივე ტიპის გამოსახულებების შექმნა, რაც შეიძლება დავხატოთ ფანქრით, სახაზავი მოწყობილობებითა და საღებავების საშუალებით.

სიტყვა "გრაფიკა", ტრადიციული აზრით, აღნიშნავს რაიმე გამოსახულების შექმნას ხაზების, წერტილებისა და მრუდების მეშვეობით. კომპიუტერულ გრაფიკაშიც არსებობს გამოსახულებების შექმნის ორი ძირითადი მეთოდი: ვექტორული (გამოსახულება ხაზოვანია) და რასტრული (გამოსახულება შედგება წერტილთა ერთობლიობისაგან) გრაფიკა. ხანდახან ამბობენ, რომ ვექტორული გრაფიკით გეომეტრიული მოდელები იქმნება, ხოლო რასტრული გრაფიკით კი - ციფრული მოდელები. კომპიუტერული გრაფიკის განვითარება უშუალოდ დაკავშირებულია კომპიუტერული ტექნიკის პროგრესთან.

პროგრამული უზრუნველყოფის თვალსაზრისით არსებობს სხვადასხვა ტიპისა და სირთულის გრაფიკული რედაქტორები; განასხვავებენ რასტრულ, ვექტორულ და ფრაქტალურ გრაფიკას. პარალელურად, არსებობს სხვა ტიპის კლასიფიკაციაც: ორგანოზომილებიანი (2D) და სამგანზომილებიანი (3D) გრაფიკა. აქ D სწორედ განზომილებას ("Dimension") ნიშნავს. 2D გამოსახულებები სიბრტყეზე ძევს, ხოლო ასეთი გამოსახულებების შერწყმით მიიღება 3D გამოსახულებები. 3D ანუ სივრცითი მოდელები სპეციალური გრაფიკული რედაქტორების მეშვეობით იქმნება, ისინი შედგებიან მარტივი და შედარებით რთული გეომეტრიული ფორმებისაგან

("პრიმიტივებისაგან"), შემდეგ ამ პრიმიტივების ერთობლიობას ერთვის რაიმე ფაქტურა ("ტექსტურა"), მოდელს აფერადებენ ფერით, ანიჭებენ გამჭვირვალობას ან დაბურულ ტონს, მის ირგვლივ მოძრაობს ვირტუალური კამერა, სივრცეში განთავსდება განათების წყაროები და იქმნება ე.წ. სამგანზომილებიანი სცენა. ამ მეთოდით გამოსახულებებს ქმნიან სარეკლამო რგოლებში, ზოგიერთ კომპიუტერულ თამაშში და ა.შ. რასტრულ გრაფიკაში არსებობს დისკრეტული სიბრტყისა და სივრცის ცნებები, ხოლო ვექტორულ გრაფიკაში აგებენ ბრტყელ და მოცულობით გეომეტრიულ მოდელებს. გრაფიკულ პროგრამულ უზრუნველყოფაში რასტრულ და ვექტორულ გრაფიკულ რედაქტორებად დაყოფა მართებული არ არის, რადგან რასტრულ რეჟიმში მომუშავე გრაფიკულ რედაქტორებში აუცილებლად არსებობს ვექტორული გრაფიკის ინსტრუმენტები და პირიქით, ვექტორულ გრაფიკულ რედაქტორებში აუცილებლად არსებობს რასტრული გამოსახულების ინსტრუმენტების გამოყენების შესაძლებლობები.

საკმაოდ მრავალფეროვანია კომპიუტერული გრაფიკის გამოყენების სფეროები. ასე, მაგალითად, კომპიუტერული გრაფიკის გამოყენების ერთ-ერთ პირველად სფეროს წარმოადგენს სამეცნიერო ვიზუალიზაცია, ანუ მეცნიერების რაიმე დარგის კვლევის ობიექტების თვალნათლივ გამოსახვა, რაიმე გათვლების შედეგების გრაფიკული დამუშავება, გამოთვლითი ექსპერიმენტების შედეგების თვალსაჩინოდ ილუსტრირება. არსებობს საქმიანი გრაფიკის ცნებაც, როცა იქმნება დაწესებულების მუშაობისთვის აუცილებელი თვალსაჩინო ილუსტრაციები: გეგმის შესრულების დიაგრამები, აღრიცხვის მასალები, სტატისტიკური გრაფიკები. ყველა ეს მასალა შეიძლება წარმოდგენილი იყოს სხვადასხვა ტიპის გრაფიკული დიაგრამების სახით. ინჟინერ-კონსტრუქტორები იყენებენ სპეციალურ გრაფიკას. ასეთი გრაფიკა ავტომატური პროექტირებისათვის სისტემებისათვისაა განკუთვნილი და ამ სისტემების სავალდებულო ნაწილს წარმოადგენს. დასაპროექტებელი მოწყობილობების ტექნიკური ნახაზები და სპეციალური გამოთვლები ერთობლივად ოპტიმალური კონსტრუქციის მოძებნის შესაძლებლობას იძლევა. ამ ტიპის გრაფიკაში შესაძლებელია როგორც ბრტყელი ნახაზების (კვეთა, პროექცია), ასევე სამგანზომილებიანი მოდელების მიღება. კომპიუტერული გრაფიკა გამოიყენება პოლიგრაფიაში, როცა ტექსტური ან გრაფიკული ინფორმაციის მრავლობითი სახით ბეჭდება ხდება. ამ დარგში მომუშავე გრაფიკოსს ესაჭიროება როგორც უშუალოდ გრაფიკული რედაქტორების ცოდნა, ასევე დაკაბადონოებისა და დასაბეჭდი მანქანების შექმნა. კომპიუტერული გრაფიკა ფართოდ გამოიყენება ვებ-დიზაინში, ვებ-გვერდის გრაფიკული ელემენტების შექმნისას, გვერდის სტრუქტურის შექმნისა და ნავიგაციის დროს. მულტიმედიაში გამოიყენება ინტერაქტიული გრაფიკული

შესაძლებლობები. გრაფიკა გამოიყენება ელექტრონული ენციკლოპედიების, საცნობარო სისტემების, სასწავლო პროგრამებისა და მათი ინტერფეისის შექმნის დროს.

განვიხილოთ სამაგიდო საგამომცემლო სისტემის ცნება (Desktop Publishing). ასეთი სისტემის ფუნქციონირებისათვის აუცილებელია სამი ურთიერთ დაკავშირებული სისტემის გამართული მუშაობა. პირველი მათგანი აპარატულ დონეს აღნიშნავს (Hardware level) და წარმოადგენს ინფორმაციის შესატან, დასამუშავებელ, შესანახ, გადასაცემ და გამოსატან მოწყობილობათა ერთობლიობას. მეორე მათგანი პროგრამულ დონეს აღნიშნავს (software level) და წარმოადგენს პროგრამების(რედაქტორების) ერთობლიობას, რომლებიც უზრუნველყოფენ ტექსტური და მხატვრული ინფორმაციის დამუშავებას, აპარატული მოწყობილობების მართვას. მესამე დონეს მომხმარებელთა დონეს (brainware level) უწოდებენ. მომხმარებლებში შედის შემოქმედებითი პიროვნებები, მაღალი კლასის სპეციალისტები და ჩვეულებრივი მომხმარებლები, რომლებიც თავისი შესაძლებლობების ინტეგრაციას ახდენენ.

უფრო დეტალურად განვიხილოთ თითოეული მათგანი. პირველ, აპარატულ დონეზე ხდება ინფორმაციის შესატანი მოწყობილობებით (Input devices) სხვადასხვა ინფორმაციის მატარებელზე ნებისმიერი სახით მოცემული ინფორმაციის ციფრული სახით წარმოდგენა, აღნიშნული პროცესი უზრუნველყოფს შემდგომში ინფორმაციის ციფრული სახით დამუშავების პერსპექტივას. შემდგომ განიხილავენ ინფორმაციის დამუშავების, შენახვის და გადაცემის მოწყობილობებს (process, storage and transfer devices) - აპარატული დონის ძირითადი ინსტრუმენტები, ბოლოს უნდა გამოვყოთ ინფორმაციის გამოტანის მოწყობილობები (output devices) - ამ მოწყობილობებს გრაფიკული ინფორმაცია ციფრული სახიდან გადაჰყავს ადამიანისათვის გასაგებ ვიზუალურ ფორმაში. ინფორმაციის შეტანის მოწყობილობებს წარმოადგენს კლავიატურა (keyboard), მაუსი (mouse), ტრეკბოლი (trackball), სკანერი (scanner), ციფრული კამერა (digital camera), გრაფიკული პლანშეტი (graphics tablet), დიჯიტაიზერი (digitizer) და ა.შ. ინფორმაციის დამუშავებისათვის გამოიყენება პროცესორი, ვიდეოპროცესორი. ხოლო ინფორმაციის შენახვის საშუალებებს მიეკუთვნება: ოპერატიული მეხსიერება (RAM), ვიდეობარათის მეხსიერება ("ვიდეოკარტა"), მაგნიტური მატარებლები (HDD, FDD), ოპტიკური მატარებლები (CD, DVD და სხვა), მაგნიტო-ოპტიკური მატარებლები (MO, MOD drive), გადასატანი გარე დისკები და მატარებლები (flash-მოწყობილობები), ლენტისანი მოწყობილობები

(სტრიმერი, mini DV) და ა.შ. ინფორმაციის გადაცემისთვის გამოიყენება სხვადასხვა სახის პორტები.

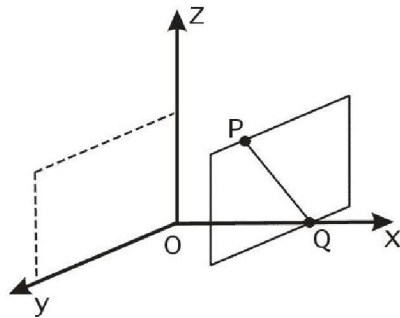
ინფორმაციის გამოტანის მოწყობილობების წინაშე დგას შეტანის მოწყობილობების მიმართ დაყენებული ამოცანის შებრუნებული ამოცანა: ანუ ინფორმაციის ციფრული ფორმიდან ადამიანისთვის გასაგებ ვიზუალურ ფორმაში გადმოტანა. ვიზუალიზაციის მეთოდის მიხედვით გამოყოფენ მოწყობილობების ორ კლასს: 1) ელექტრონული ვიზუალიზაციის მოწყობილობები (მონიტორები და პროექტორები), 2) ინფორმაციის გამოტანის მოწყობილობები (ჭავჭავი, ლაზერული და ფოტო პრინტერები, გრაფიკების ამგები მოწყობილობები, ფოტოამწყობი ავტომატები).

ცნობილია, რომ კომპიუტერი სპეციალური პროგრამების(რედაქტორების) გარეშე ვერანაირ მოქმედებას ვერ ასრულებს. ცხადია, რომ ყველა მოქმედება მკაცრად გაწერილია მომუშავე ალგორითმებში. გრაფიკულ ფაილებთან მუშაობის დროს გამოიყენება შემდეგი ტიპის პროგრამული უზრუნველყოფა: რასტრული (იგივე პიქსელური) გრაფიკის რედაქტორები, ვექტორული გრაფიკის რედაქტორები, ფრაქტალური გრაფიკის რედაქტორები, სამგანზომილებიანი გრაფიკული რედაქტორები, დაკაზადონობის ინსტრუმენტები, აგრეთვე სისტემური და სპეციალური პროგრამები: ოპერაციული სისტემა, უტილიტები, plug-in პროგრამები (ზოგიერთი შესაძლებლობის გაფართოების მიზნით ჩასმული მოდულები), viewer პროგრამები (დამათვალიერებლები), კონვერტორები, ბრაუზერები, არქივატორები და სხვა). მომხმარებელი მისი ესთეტიკური და მხატვრული უნარებით წარმოადგენს საგამომცემლო სისტემის მესამე შემადგენელ ნაწილს.

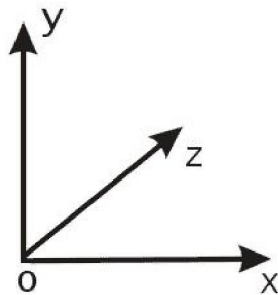
## **კოორდინატთა სისტემები, გრაფიკული გამოსახულებების გარდაქმნები**

იმისათვის, რომ კომპიუტერის მონიტორზე შეიქმნას გრაფიკული გამოსახულება, აუცილებელია ობიექტი მათემატიკურად იყოს აღწერილი ორ ან სამგანზომილებიან სივრცეში. სწორედ ამ წარმოდგენებზე გავამახვილოთ ახლა ყურადღება. ცნობილია, რომ ჩვენს ირგვლივ სამყარო წარმოადგენს სამგანზომილებიან ევკლიდეს სივრცეს. ასეთ სივრცეში ობიექტის აღწერა ნიშნავს დროის ყოველ მომენტში ამ ობიექტის ნებისმიერი წერტილის ადგილმდებარეობის ცოდნას. სამგანზომილებიან სივრცეში წერტილების მდებარეობის

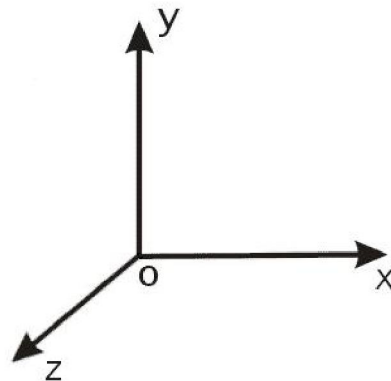
განსაზღვრისათვის მოსახერხებელია დეკარტის კოორდინატა სისტემის გამოყენება. სამგანზომილებიან სივრცეში დეკარტის კოორდინატა სისტემა წარმოადგენს ერთ წერტილში (კოორდინატა სათავეში) გადამკვეთ სამ წრფეს (ღერძებს), რომლებიც ერთ სიბრტყეზე არ მდებარეობენ. ღერძებზე უნდა იყოს მოცემული მიმართულება და საზომი ერთეული (მასშტაბი). სივრცეში ყოველი წერტილის ადგილმდებარეობა განისაზღვრება თითოეული ღერძზე წერტილის პროექციასა და კოორდინატა სათავეს შორის მანძილით. სამი ღერძისთვის სამი პროექცია და მანძილის აღმნიშვნელი სამი რიცხვი მიიღება.



ზემოთ აღწერილი კოორდინატა სისტემა ზოგადია, თუმცა ყველაზე ხშირად გამოიყენება მართკუთხა (ორთოგონალური) კოორდინატა სისტემა, როცა ღერძები ერთმანეთის მიმართ ურთიერთ პერპენდიკულარულია. ე.ი. ნებისმიერი  $P$  წერტილი იწერება, როგორც  $P(x, y, z)$ . ღერძების ურთიერთგანლაგების მიხედვით არსებობს მარცხენა და მარჯვენა კოორდინატა სისტემა (წარმოიდგინეთ მარჯვენა და მარცხენა ხელის თითების მეშვეობით).



მარცხენა კოორდინატა სისტემა



მარჯვენა კოორდინატა სისტემა



დეკარტის კოორდინატთა სისტემა აღწერს სტატიკურ (უძრავ ობიექტებს) სივრცეში. მოძრაობის მოდელირების დროს გამოიყენება დამატებითი მათემატიკური ობიექტები. ერთ-ერთ ასეთ ობიექტს წარმოადგენს რადიუს-ვექტორი.

რადიუს-ვექტორს იგივე თვისებები აქვთ, რაც ჩვეულებრივ ვექტორს, მაგრამ მისი საწყისი წერტილი ყოველთვის კოორდინატთა სათავეში მდებარეობს, ხოლო ბოლო სივრცის რაიმე წერტილშია. რადიუს-ვექტორის ეს თვისება იძლევა საშუალებას, ავაგოთ ურთიერთცალსახა ასახვა სივრცის ყველა წერტილსა და რადიუს ვექტორებს შორის. შემდეგ ეტაპზე განვიხილავთ წრფეებისა და სიბრტყეების აღწერის მეთოდს სამგანზომილებიან სივრცეში წერტილის კოორდინატებისა და რადიუს-ვექტორების გამოყენებით. წრფის აღწერის დროს უნდა განვსაზღვროთ, ეკუთვნის თუ არა წერტილი მოცემული კოორდინატებით რაიმე წრფეს. ანუ ვიპოვოთ წრფის განტოლება. წრფის ორ წერტილს  $P_1(x_1, y_1, z_1)$  და  $P_2(x_2, y_2, z_2)$  შევუსაბამოთ ორი რადიუს ვექტორი  $p_1$  და  $p_2$ .  $P_1$  წერტილიდან  $P_2$  წერტილამდე გავავლოთ ახალი ვექტორი  $p^* = p_2 - p_1$ . მაშინ  $p$  რადიუს-ვექტორი, რომელიც განსაზღვრავს რაიმე წერტილს წრფეზე, მივიღებთ  $p_1$  ვექტორისა და  $p^*$  ვექტორის რაიმე  $\mu$  რიცხვზე ნამრავლის შეკრებით. ანუ  $p = p_1 + \mu p^*$ . ფაქტიურად ჩვენ უკვე მივიღეთ წრფის განტოლება, რომელიც ჩაწერილია არა ორი წერტილის კოორდინატების მეშვეობით, არამედ ჩვენ მივიღეთ ჩანაწერი, რომელშიც გამოყენებულია ორი ვექტორი: საბაზისო  $p_1$  რადიუს-ვექტორი და მიმმართველი  $p^*$  რადიუს-ვექტორი. ახლა დავიყვანოთ მიღებული განტოლება ისეთ სახეზე, რომ განტოლებაში გამოყენებული იქნეს მხოლოდ საწყისი  $p_1$  და  $p_2$  ვექტორების კოორდინატები.

$$p = p_1 + \mu p^* = p_1 + \mu(p_2 - p_1) \Rightarrow p - p_1 = \mu(p_2 - p_1)$$

ამ ვექტორული განტოლებიდან ვღებულობთ სამ საკოორდინატო განტოლებას:

$$\begin{cases} x - x_1 = \mu(x_2 - x_1) \\ y - y_1 = \mu(y_2 - y_1) \\ z - z_1 = \mu(z_2 - z_1) \end{cases}$$

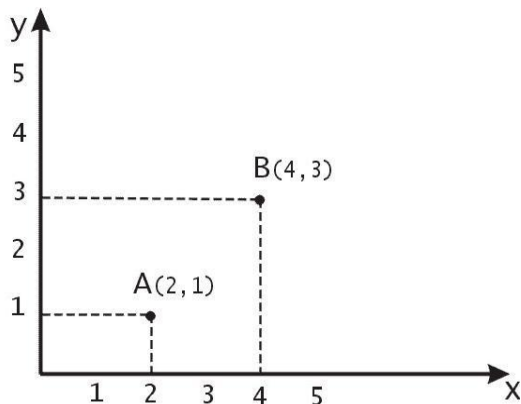
წყვილ-წყვილად გავყოთ ეს განტოლებები ერთმანეთზე იმისათვის, რომ მოვიცილოთ  $\mu$  კოეფიციენტი. საბოლოოდ, ვღებულობთ შემდეგ განტოლებათა სისტემას, რომელიც განსაზღვრავს წრფეს სივრცეში:

$$\begin{cases} (x - x_1)(y_2 - y_1) = (x_2 - x_1)(y - y_1) \\ (y - y_1)(z_2 - z_1) = (y_2 - y_1)(z - z_1) \\ (z - z_1)(x_2 - x_1) = (z_2 - z_1)(x - x_1) \end{cases}$$

კომპიუტერულ გრაფიკაში აუცილებელია გამოსახულების მართვა ეკრანზე, გამოსახულების ფორმის, განლაგების, ორიენტაციისა და ზომის ცვლილება. სწორედ ამ მიზნებისათვის გამოიყენება სპეციალური გეომეტრიული გარდაქმნები, რომლებიც ზემოთ ჩამოთვლილი ცვლილებების განხორციელების საშუალებას გვაძლევს.

## ორგანზომილებიანი მატრიცული გარდაქმნები

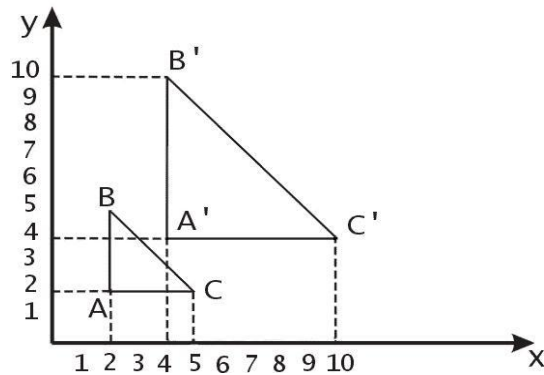
განვიხილოთ წერტილების კოორდინატების გარდაქმნები სიბრტყეზე. სურათზე  $A$  წერტილი  $B$  წერტილში გადადის.



$A$  წერტილის  $B$  წერტილში გადატანის ოპერაცია ანუ ტრანსლაცია: გეომეტრიულად ეს ოპერაცია შეიძლება აღვწეროთ გადატანის  $\vec{AB}$  ვექტორის საშუალებით. ვთქვათ,  $\vec{R}$  რადიუს-ვექტორია, რომელიც გადატანის  $\vec{AB}$  ვექტორს შეესაბამება. მაშინ  $A$  წერტილის  $B$  წერტილში გადატანა შეესაბამება ვექტორულ ჩანაწერს  $\vec{B} = \vec{A} + \vec{R}$ . აქედან ვასკვნით, რომ წერტილის ახალ წერტილში გადატანისათვის აუცილებელია თავდაპირველი წერტილის კოორდინატებს დავუმატოთ რიცხვები, რომლებიც გადატანის ვექტორის კოორდინატებს წარმოადგენენ:

$$\vec{B} = \vec{A} + \vec{R} = [A_x + R_x, A_y + R_y, A_z + R_z]$$

**ობიექტის მასშტაბირებას** უწოდებენ ობიექტის გაჭიმვას კოორდინატა სათავის მიმართ შესაბამისი საკოორდინატო ღერძების გასწვრივ. ასეთი ოპერაცია გამოიყენება ობიექტის თითოეული წერტილის მიმართ. ამიტომ ჩვენ შეგვიძლია წერტილის მასშტაბირების ცნება განვიხილოთ. თუმცა, რა თქმა უნდა, ამ შემთხვევაში ჩვენ არ განვიხილავთ წერტილის ზომების ცვლილების ოპერაციას. მასშტაბირების ოპერაცია ნიშნავს წერტილის კოორდინატების რაიმე მუდმივ რიცხვებზე გამრავლებას. იმ შემთხვევაში, როცა ყველა კოორდინატი ერთი და იმავე რიცხვზე მრავლდება, მასშტაბირებას ერთგვაროვანს უწოდებენ. სურათზე მოცემულია სამკუთხედის ერთგვაროვანი მასშტაბირების მაგალითი.



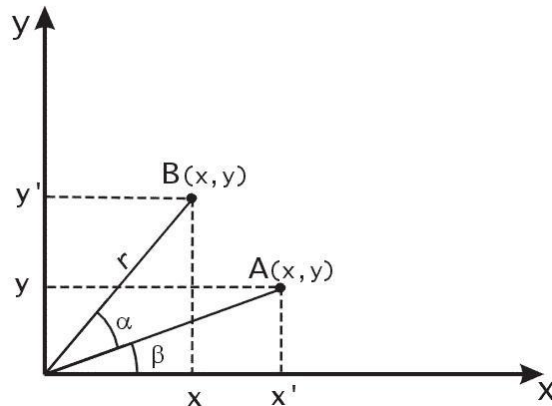
სურათზე მოცემულია სამკუთხედის ერთგვაროვანი მასშტაბირება კოეფიციენტით 2, როცა მოცემული  $ABC$  სამკუთხედი გადადის  $A'B'C'$  სამკუთხედში. მასშტაბირების მატრიცას ვუწოდებთ შემდეგ მატრიცას:

$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

ორი  $A$  და  $A'$  წერტილებისათვის მასშტაბირების ოპერაცია მატრიცული სახით შემდეგნაირად გამოიყურება:

$$[x', y'] = [x, y] \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

ახლა კი განვიხილოთ წერტილის რაიმე კუთხით მობრუნების ოპერაცია კოორდინატთა სათავის მიმართ. სურათზე  $A = (x, y)$  წერტილი გადადის  $B = (x', y')$  წერტილში ისე, რომ მობრუნება რაიმე  $\alpha$  კუთხით ხდება.



სურათზე მოცემულია  $A$  წერტილის  $\alpha$  კუთხით მობრუნება. ახლა ჩავწეროთ  $A$  წერტილის კოორდინატების  $B$  წერტილის კოორდინატებში გარდაქმნა.  $\beta$  კუთხით აღვნიშნოთ კუთხე  $\vec{A}$  რადიუს ვექტორსა და  $Ox$  ღერძს შორის,  $r$ - რადიუს ვექტორის სიგრძეა, მაშინ

$$x' = r \times \cos(\alpha + \beta) = r(\cos \alpha \cos \beta - \sin \alpha \sin \beta),$$

$$y' = r \times \sin(\alpha + \beta) = r(\sin \alpha \cos \beta + \cos \alpha \sin \beta)$$

რადგან  $\cos \beta = \frac{x'}{r}$  და  $\sin \beta = \frac{y'}{r}$ , მაშინ ჩავსვათ ეს გამოსახულებები  $x'$  და  $y'$  -ის გამოსახულებებში, მივიღებთ:

$$x' = x \times \cos \alpha - y \times \sin \alpha$$

$$y' = x \times \sin \alpha + y \times \cos \alpha$$

$A$  წერტილის  $\alpha$  კუთხით მობრუნება მატრიცული სახით გამოიყურება შემდეგნაირად:

$$[x', y'] = [x, y] \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

## ერთგვაროვანი კოორდინატები და ორგანზომილებიანი გარდაქმნების მატრიცული წარმოდგენები

ზემოთ, ჩვენ გვქონდა წარმოდგენილი სიბრტყის წერტილების სამი სახის გარდაქმები. ამ გარდაქმნების ორი სახეობა - მობრუნების ოპერაცია და მასშტაბირების ოპერაცია - ჩაიწერება როგორც მატრიცისა და ვექტორის ნამრავლი, ხოლო მესამე - გადატანის ოპერაცია ჩაიწერება ორი ვექტორის ჯამის სახით. იმ შემთხვევაში, როცა მობრუნებისა და მასშტაბირების ოპერაციები გარკვეული თანმიმდევრობით სრულდება, შედეგი შეიძლება მარტივად ჩავწეროთ შესაბამისი გარდაქმნებით - მატრიცების ნამრავლის სახით. ამ შემთხვევაში მივიღებთ მობრუნებისა და მასშტაბირების საბოლოო მატრიცას. ცხადია, ასეთი საბოლოო მატრიცის გამოყენება უფრო მოსახერხებელია, ვიდრე ყოველ ჯერზე მატრიცების ნამრავლის ძებნა. მაგრამ ამ ხერხით ვერ მივიღებთ საბოლოო მატრიცას, თუ

გარდაქმნებს შორის ერთ-ერთი ოპერაცია გადატანაა. მატრიცულ ნამრავლებს კომპიუტერულ გრაფიკაში კომპოზიციას უწოდებენ. კარგი იქნებოდა, თუ ასეთი მათემატიკური აპარატი უზრუნველყოფდა სამივე ტიპის ოპერაციების კომპოზიციას. ეს მოგვცემდა გათვლების კარგ სისწრაფეს. ერთგვაროვანი კოორდინატების მეთოდი წარმოადგენს სწორედ სამი ტიპის გარდაქმნებისათვის შესაფერის მათემატიკურ აპარატს. განვიხილოთ ეს მეთოდი უფრო ვრცლად. ორგანზომილებიანი ვექტორი  $(x, y)$  ერთგვაროვან კოორდინატებში ჩაიწერება შემდეგი სახით  $(wx, wy, w)$ , სადაც  $w \neq 0$ . ამ  $w$  რიცხვს უწოდებენ მასშტაბურ თანამამრავლს. იმისათვის, რომ ერთგვაროვანი კოორდინატებით ჩაწერილი ვექტორისაგან მივიღოთ ჩვეულებრივ კოორდინატებში ჩაწერილი ვექტორი, აუცილებელია პირველი ორი კოორდინატის მესამეზე გაყოფა:

$$(wx/w, wy/w, w/w) \rightarrow (x, y, 1)$$

თუ განვიხილავთ ზოგად შემთხვევას, აქ ხდება  $n$ -განზომილებიანი სივრციდან  $(n+1)$ -განზომილებიან სივრცეზე გადასვლა. ცხადია, ამ გარდაქმნისათვის არსებობს შექცეული გარდაქმნა. შექცეულ გარდაქმნას უწოდებენ ერთგვაროვანი კოორდინატების პროექციას.

ერთგვაროვან კოორდინატებში ცენტრალური პერსპექტივის ოპერაციას მატრიცული სახით აღწერენ, ის შემდეგნაირად გამოიყურება:

$$\begin{bmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & k \end{bmatrix} = P$$

ვაჩვენოთ, რომ ეს მატრიცა განსაზღვრავს ერთგვაროვან კოორდინატებში მოცემული ობიექტის წერტილის გარდაქმნას პერსპექტიული პროექციის წერტილში (აგრეთვე ერთგვაროვანი კოორდინატებით ჩაწერილს). ვთქვათ,  $p(x, y, z)$  სამგანზომილებიანი

სივრცის წერტილია. მისი ერთგვაროვანი წარმოდგენაა  $v = (wx, wy, wz, w)$ .  $v$  გავამრავლოთ  $P$ -ზე:

$$vP = [wkw, wky, 0, w(z+k)] = [kx/(z+k), ky/(z+k), 0, 1]$$

ამის შედეგად, ორგანზომილებიანი სივრცის წერტილები აღიწერება სამ ელემენტური ვექტორ-სტრიქონების სახით. ამიტომ გარდაქმნების მატრიცები, რომლებზეც გამრავლდება წერტილის ვექტორი, იქნება 3x3-ზე ზომის. ჩავწეროთ ახლა გადატანის ოპერაციის მატრიცული გარდაქმნა ერთგვაროვანი კოორდინატებისათვის:

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}$$

$$\text{ან } p' = pT(D_x, D_y), \text{ სადაც } T(D_x, D_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}$$

$p$  წერტილის  $p'$ -ში, ხოლო შემდეგ  $p^*$  თანმიმდევრული გადატანის დროს ჯამური გადატანის ვექტორის კომპონენტები წარმოადგენენ გადატანის თანმიმდევრული ვექტორების შესაბამისი კომპონენტების ჯამებს. განვიხილოთ, თუ რას წარმოადგენენ ჯამური გადატანის მატრიცის ელემენტები.

ვთქვათ,  $p' = pT(D_x, D_y)$ ,  $p'' = p'T(D'_x, D'_y)$ . ჩავსვათ პირველი განტოლება მეორეში, მივიღებთ:  $p'' = pT(D_x, D_y)p'T(D'_x, D'_y)$ . მატრიცული ნამრავლი, ე.ი. ჯამური გადატანა, უდრის შესაბამისი გადატანის მატრიცების ნამრავლს:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D'_x & D'_y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x + D'_x & D_y + D'_y & 1 \end{bmatrix}$$

მასშტაბირების ოპერაციის მატრიცულ სახეს შემდეგნაირად წერენ:

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

მასშტაბირების მატრიცა შემდეგი სახისაა:  $S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$

შევნიშნოთ, რომ თანმიმდევრული გადატანები ადიციური ოპერაციებია. ვაჩვენოთ, რომ თანმიმდევრული მასშტაბირებები იქნება მულტიპლიკაციური.

$$S(S_x, S_y) \times S(S'_x, S'_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S'_x & 0 & 0 \\ 0 & S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_x \times S'_x & 0 & 0 \\ 0 & S_y \times S'_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

მობრუნების ოპერაციის მატრიცული სახე შემდეგია:

$$[x', y', 1] = [x, y, 1] \times \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ახლა განვსაზღვროთ მობრუნების მატრიცა  $R(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$

წინა ორი შემთხვევის ანალოგიურად, ვაჩვენოთ, რომ მობრუნების მატრიცა იგივე რჩება (ინვარიანტულია) თანმიმდევრული მობრუნებების დროს.



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

აქედან გამომდინარე, ორი თანმიმდევრული მობრუნება (და ე.ი. თანმიმდევრული მობრუნებების ნებისმიერი რაოდენობა) შეიძლება ჩაიწეროს ჯამური მობრუნების ერთი მატრიცის სახით.

შეიძლება აგრეთვე შევნიშნოთ, რომ ოპერაციების ნებისმიერი თანმიმდევრობა, ერთგვაროვან კოორდინატებში გადატანის, მასშტაბირების და მობრუნების ჩათვლით, შეიძლება იყოს წარმოდგენილი ერთი მატრიცის სახით, რომელიც მოცემული მატრიცების ნამრავლს წარმოადგენს.

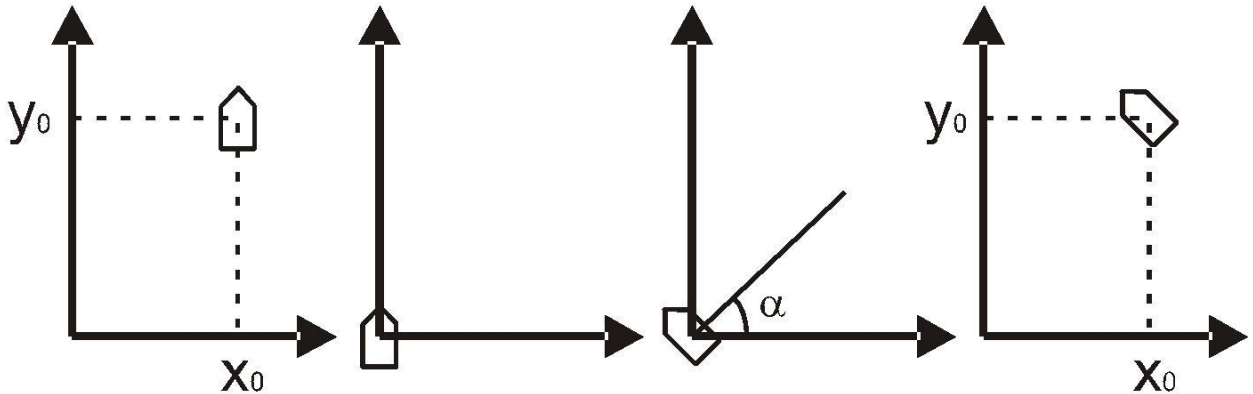
განხილოთ ახლა, თუ როგორ შეიძლება მივიღოთ ერთი საბოლოო გარდაქმნა მატრიცული გარდაქმნების კომპოზიციის მეშვეობით. ამისათვის უნდა გამოვიყენოთ მატრიცები  $T, S$  და  $R$ . გამოთვლების თვალსაზრისით გაცილებით მარტივია და სწრაფია, უკვე მზა გარდაქმნის მატრიცის გამოყენება, ვიდრე ეს გარდაქმნები გამოვიყენოთ რიგ-რიგობით, ერთმანეთის მიმდევრობით. წერტილის მიმართ უფრო ეფექტურია გამოვიყენოთ ერთი საბოლოო გარდაქმნა, ვიდრე ერთმანეთის მიმდევრობით რამდენიმე გარდაქმნა.

მაგალითისათვის განვიხილოთ ობიექტის სიბრტყეზე მობრუნების ამოცანა რაიმე  $p_0$  წერტილის მიმართ. ჯერ-ჯერობით, ჩვენ მხოლოდ განვიხილეთ ობიექტების მობრუნება მხოლოდ კოორდინატთა სათავის მიმართ. მაგრამ შესაძლებელია ეს ამოცანა დავყოთ რამდენიმე ბიჯად, როცა თითოეულ ბიჯზე შესრულდება მხოლოდ ელემენტარული ოპერაცია: გადატანა, მასშტაბირება ან მობრუნება. ჩამოვთვალოთ ეს ელემენტარული გარდაქმნები:

1. გადატანა, როცა  $p_0$  წერტილი გადადის კოორდინატთა სათავეში.

2. მოზრუნება მოცემული კუთხით.

3. გადატანა, როცა წერტილი კოორდინატა სათავიდან ბრუნდება პირვანდელ  $p_0$  წერტილში.



გარდაქმნათა თანმიმდევრობა, როცა ობიექტის მოზრუნება ხდება მოცემული  $p_0$  წერტილის მიმართ  $\alpha$  კუთხით. პირველი გადატანა ხდება  $[-x_0, -y_0]$  ვექტორზე, ხოლო შექცეული გადატანა  $[x_0, y_0]$  ვექტორზე.

## სამგანზომილებიანი მატრიცული გარდაქმნები

იმის მსგავსად, რომ ორგანზომილებიანი გარდაქმნები აღიწერება  $3 \times 3$ -ზე ზომის მატრიცებით, სამგანზომილებიანი გარდაქმნები შეიძლება იყოს წარმოდგენილი  $4 \times 4$ -ზე ზომის მატრიცებით. მაშინ სამგანზომილებიანი  $(x, y, z)$  წერტილი ერთგვაროვან კოორდინატებში იწერება, როგორც  $(wx, wy, wz, w)$ , სადაც  $w \neq 0$ . დეკარტის კოორდინატების მისაღებად პირველი სამი ერთგვაროვანი კოორდინატი უნდა გავყოთ  $w$ -ზე. გარდაქმნების მატრიცებს ჩვენ ჩავწერთ მარჯვენა კოორდინატა სისტემაში, თანაც დადებითი მოზრუნება განისაზღვრება

შემდეგნაირად. თუ შევხედავთ მობრუნების ღერძის დადებითი ნაწილიდან (მაგალითად, z ღერძიდან), კოორდინატთა სათავისაკენ, მაშინ 90°-იანი მობრუნება საათის ისრის საწინააღმდეგოდ გადაიყვანს ერთ დადებით ნახევარღერძს მეორეში (x ღერძს გადაიყვანს y ღერძზე, ციკლური გადანაცვლების წესების მიხედვით).

შევნიშნოთ, რომ პრაქტიკაში უფრო მოსახერხებელია მარცხენა კოორდინატთა სისტემის გამოყენება, რადგან ამ შემთხვევაში სწორადაა ინტერპრეტირებული ის ფაქტი, რომ z-ის დიდი მნიშვნელობა განლაგებულია მომხმარებლისგან უფრო შორს. ჩავწეროთ ახლა სამგანზომილებიანი გადატანის მატრიცა, ორგანზომილებიანი შემთხვევის ანალოგიურად.

$$T(D_x, D_y, D_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}, \text{ სადაც}$$

$$[x, y, z, 1] \cdot T(D_x, D_y, D_z) = [x + D_x, y + D_y, z + D_z, 1]$$

მასშტაბირების ოპერაცია:

$$S(S_x, S_y, S_z) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[x, y, z, 1] \cdot S(S_x, S_y, S_z) = [x \cdot S_x, y \cdot S_y, z \cdot S_z, 1]$$

გადავიდეთ ახლა მობრუნების ოპერაციაზე. რადგან ორგანზომილებიანი მობრუნების დროს xy სიბრტყეზე z კოორდინატები უცვლელი რჩება, ამიტომ z ღერძის ირგვლივ მობრუნება იწერება შემდეგნაირად:

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

x ღერძის მიმართ მობრუნების მატრიცას აქვს სახე:

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

და y ღერძის მიმართ:

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

მიაქციეთ ყურადღება კუთხის სინუსის უარყოფით ნიშანს y ღერძის ირგვლივ მობრუნების მატრიცაში. ამ მატრიცების სისწორეში შეიძლება დარწმუნდეთ, თუ რომელიმე ღერძს მოვაბრუნებთ  $90^\circ$ -ით, თანაც ღერძი გადადის ისევ ღერძზე.

შექცეული გარდაქმნები გამოისახება შებრუნებული მატრიცებით. გადატანის ოპერაციისათვის უნდა შევცვალოთ მხოლოდ გადატანის ვექტორის კომპონენტები საპირისპირო მნიშვნელობებით:

$$T^{-1}(D_x, D_y, D_z) = T(-D_x, -D_y, -D_z);$$

მასშტაბირების ოპერაციისათვის - შებრუნებული მნიშვნელობებით:

$$S^{-1}(S_x, S_y, S_z) = s(1/S_x, 1/S_y, 1/S_z);$$

მობრუნებისათვის - მობრუნების უარყოფითი კუთხის არჩევით:

$$R^{-1}(\alpha) = R(-\alpha).$$

რამდენიმე თანმიმდევრული მობრუნების შედეგი იქნება მატრიცა:

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

აქ ზედა მატრიცას  $3 \times 3$ -ზე ზომით უწოდებენ ორთოგონალურს. მისი ერთი-ერთი საინტერესო თვისებაა ის, რომ მისი შებრუნებული მატრიცა არის მისი ტრანსპონირებული მატრიცა, ანუ  $B^{-1} = B^T$ . ეს თავისებურება თავისთავად მოსახერხებელია იმიტომ, რომ გამოთვლების დროს საკმარისია ინდექსების ადგილების შეცვლა და შექცეული გარდაქმნა ავტომატურად მიიღება.

$T, S, R$  მატრიცების გადამრავლების შედეგად მიღებულ მატრიცას ყოველთვის ექნება სახე:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

აქ  $3 \times 3$ -ზე ზომის ზედა ნაწილი განსაზღვრავს ჯამურ მობრუნებას და მასშტაბირებას, ხოლო ბოლო სტრიქონის 3 კოეფიციენტი განსაზღვრავს ჯამურ გადატანას.

## ფერთა მოდელები

ფერის ფიზიკური შინაარსი მდგომარეობს შემდეგში: ფერი წარმოადგენს გარკვეული სიგრძის ტალღების ერთობლიობას, რომელიც რაიმე საგნიდან არის არეკლილი ან რაიმე გამჭვირვალე საგანს გადის. თუმცა ჩვენ არ გვინტერესებს საკითხი, თუ რას წარმოადგენს ფერი, რა არის მისი ფიზიკური

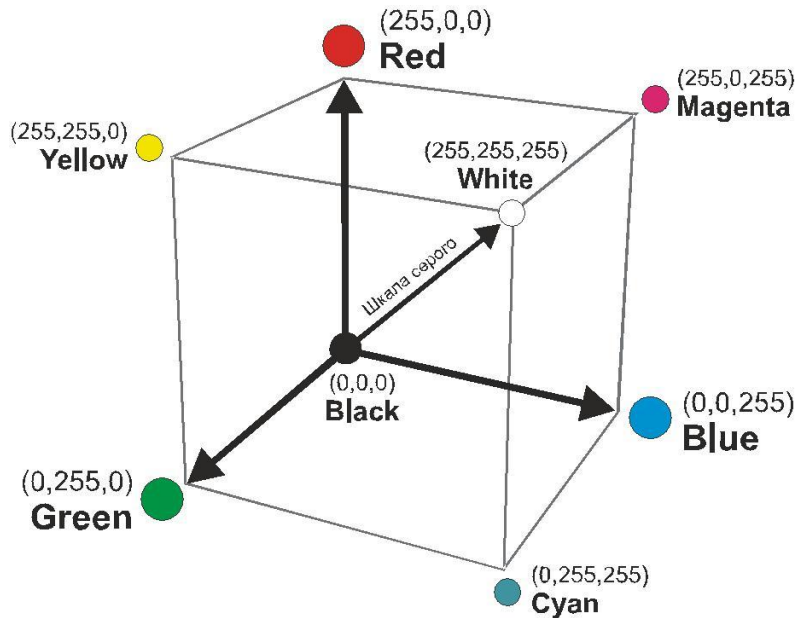
შინაარსი. ჩვენ გვინტერესებს, თუ როგორ უნდა მივიღოთ, პრაქტიკული თვალსაზრისით, ესა თუ ის ფერი.

ბუნებაში იშვიათად გვხვდება სუფთა, იდეალური ფერები. ფერების უმრავლესობა შერევის შედეგად მიიღება. ასე, მაგალითად, წითელისა და ლურჯის ნარევი იძლევა იძლევა მეწამულ ფერს (purple), ხოლო ლურჯი და მწვანეს შერევა იძლევა ცისფერს. მაშასადამე, მცირე რაოდენობის მარტივი (სუფთა) ფერების შერევით მიიღება შერეული (რთული) ფერების საკმაოდ დიდი პალიტრა. ამიტომ ფერთა აღწერისათვის შემოტანილია ფერთა მოდელის ცნება. ფერთა მოდელი კომპიუტერულ გრაფიკაში ერთ-ერთი ძირითადი ცნებაა და წარმოადგენს მარტივი შემადგენელი კომპონენტების საშუალებით ფერთა პალიტრის მიღების მეთოდს. ცნობილია რამდენიმე ათეული ფერთა მოდელი. ჩვენ ჯერ-ჯერობით განვიხილავთ მხოლოდ სამ ძირითად მოდელს.

**RGB მოდელი.** ფერთა მრავალფეროვნებას ადამიანის თვალი აღიქვამს იმიტომ, რომ განათებული ობიექტები მათ ასხივებენ. გამოსხივების შედეგად მიღებულ ასეთ ფერებს მიეკუთვნება ტელევიზორის, მონიტორის ან პროექტორის ეკრანიდან მიღებული ფერები. შერევის შედეგად უამრავი ფერი მიიღება, მაგრამ სამი მათგანი (ლურჯი, მწვანე და წითელი) წარმოადგენს ძირითად (პირველად) ფერებს. ორი ძირითადი ფერის შერევის დროს მიღებული ფერი უფრო ღიაა, ვიდრე თავდაპირველი ფერები. წითელი და მწვანე იძლევა მოყვითალო ფერს, მწვანე და ლურჯის ნარევი ცისფერია, ლურჯი და წითელი იძლევა მეწამულს. სამი ძირითადი ფერის შერევა იძლევა თეთრ ფერს. ასეთ ფერთა მოდელს ადიციურს უწოდებენ.

ფერთა მოდელს, რომლის ძირითად ფერებს წარმოადგენს ლურჯი, მწვანე და წითელი, უწოდებენ RGB მოდელს. სახელწოდება წარმოადგენს აბრევიატურას Red (წითელი), Green (მწვანე), Blue (ცისფერი). ამ მოდელში

გამოიყენება სამი ერთმანეთისაგან დამოუკიდებელი მნიშვნელობა, ამიტომ მისი წარმოდგენა შეიძლება სამგანზომილებიანი კოორდინატთა სისტემის სახით.



ყოველი კოორდინატი აღნიშნავს კონკრეტულ ფერში შეტანილ "თავის წვლილს" და ეს კოორდინატი შეიძლება იცვლებოდეს ნულიდან მაქსიმალურ მნიშვნელობამდე. ასეთი წარმოდგენის შედეგად მიიღება კუბი, რომლის შიგნით მოთავსდება ფერთა მოდელით მიღებული ყველა შესაძლო ფერი. შეიძლება ვთქვათ, რომ ასე ვლებულობთ RGB მოდელის ფერთა პალიტრას. ცხადია, ასეთი წარმოდგენის დისკრეტულობის გამო შესაძლო ფერთა რაოდენობა სასრულია. ასეთი კუბის "მოცულობა", ანუ ციფრულ ფერთა რაოდენობა ითვლება შემდეგნაირად: ყოველ ღერძზე თუ ავიღებთ 256 მნიშვნელობას, მაშინ  $256^3$  ანუ  $2^{24}$  გვაძლევს 16 777 216 ფერს.

განვიხილოთ ახლა ამ ფერთა მოდელის ზოგიერთი განსაკუთრებული წერტილი.

**კოორდინატთა სათავე.** ამ წერტილში ყველა შემადგენელი კომპონენტი ნულის ტოლია, ანუ გამოსხივება არ არსებობს, ეს კი ფერის თვალსაზრისით სიბნელეს ნიშნავს, მაშასადამე, კოორდინატთა სათავე - შავი ფერის წერტილია. **წერტილი,**

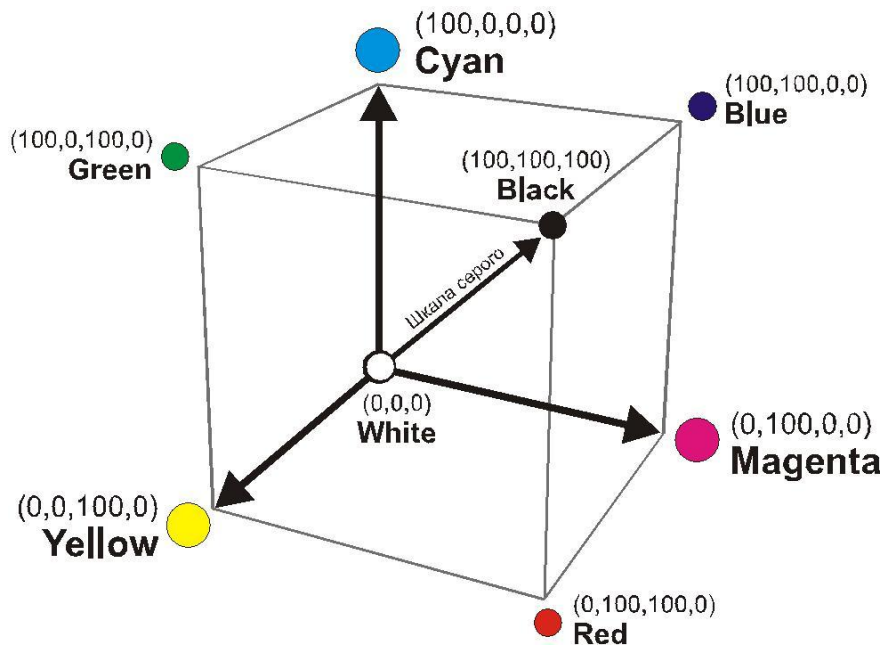
სადაც ყველა კომპონენტი მაქსიმალურ მნიშვნელობას ღებულობს (ნახაზზე უახლოესი წერტილი), აღნიშნავს აბსოლუტურად თეთრ ფერს. **კუბის დიაგონალზე**, რომელიც აერთებს კოორდინატთა სათავესა და უახლოეს წერტილს, ანუ აბსოლუტურად თეთრ ფერს, განლაგებულია ნაცრისფერი ფერები - შავიდან თეთრამდე. ეს კი ხდება იმიტომ, რომ ასეთი წერტილების სამივე კომპონენტი ერთნაირია და მათი მნიშვნელობები იცვლება დიაპაზონში ნულიდან მაქსიმალურ მნიშვნელობამდე. ამ დიაპაზონს ანუ დიაგონალს უწოდებენ ნაცრისფერ სკალას (Grayscale). კომპიუტერულ ტექნოლოგიებში ყველაზე ხშირად გამოიყენება ნაცრისფრის 256 გრადაცია (ელფერი ანუ ტონი). თუმცა არსებობს სკანერები, რომლებიც ნაცრისფრის 1024 ან მეტი გრადაციის კოდირების შესაძლებლობა აქვთ. კუბის სამი წვერო აღნიშნავს "სუფთა" საწყის ფერებს, ხოლო დანარჩენი სამი წვერო წარმოადგენს ორ-ორი სუფთა ფერის ნარევს.

RGB მოდელი წარმოადგენს სკანირების პროცესის თეორიულ ბაზას და მას ემყარება მონიტორის ეკრანზე გამოსახულების ვიზუალიზაცია.

**CMYK მოდელი.** ეს მოდელი აღწერს არეკლილ ფერებს. არეკლილი ფერები არის ის ფერები, რომლებიც თვითონ არანაირ შუქს არ ასხივებენ, მაგრამ იყენენ თეთრ შუქს და თეთრი შუქიდან რაიმე ფერის გამოკლებით მიიღებიან, ასეთ ფერებს სუბტრაქტულებს (გამოკლების შედეგად მიღებულებს) უწოდებენ. მათი მიღება შესაძლებელია თეთრი შუქიდან ძირითადი ადიციური ფერების გამოკლებით. არსებობს სამი ძირითადი სუბტრაქტული ფერი: ცისფერი, მეწამული და ყვითელი. მათ პოლიგრაფიულ ტრიადას უწოდებენ. რთული მექანიზმის შედეგად ამ ფერებით ბეჭდვის დროს ხდება წითელის, მწვანეს და ლურჯის შთანთქმა ისეთნაირად, რომ შესაძლებელია ხილვადი ფერების სპექტრის უდიდესი ნაწილის რეპროდუცირება ქალაღზე, ანუ თეთრ ფონზე. ორი სუბტრაქტული ფერის შერევის დროს მიღებული ფერი მუქდება, ხოლო სამი სუბტრაქტული ფერის შერევის დროს უნდა მივიღოთ შავი ფერი. საღებავის არ არსებობის შემთხვევაში



რჩება ფონის თეთრი ფერი ანუ თეთრი ქალაქი. ზემოთ თქმულის საფუძველზე გამოდის, რომ შემადგენელი სუბტრაქტული ფერების ნულოვანი მნიშვნელობები უნდა გვაძლევდეს თეთრ ფერს, მათი მაქსიმალური მნიშვნელობები უნდა გვაძლევდეს შავ ფერს, ხოლო სამივე კომპონენტის ერთნაირი მნიშვნელობები - ნაცრისფერ ტონებს. გარდა ამისა, არსებობს სუფთა სუბტრაქტული ფერები და მათი ორ-ორ ფერად ნარევი. წარმოდგენილი მოდელის ძირითადი პრობლემა მდგომარეობს შემდეგში: რადგანაც ეს მოდელი გამიზნულია რეალურად არსებულ პოლიგრაფიული საღებავების აღწერის მიზნით, საღებავები კი სულაც არ არის ისე იდეალურად შექმნილი, როგორც ეს ფერადი შუქის სხივის შემთხვევაში ხდება, საღებავებს აქვთ მინარევი, მათ არ შეუძლიათ სრულად გადაფარონ ფერების მთელი დიაპაზონი, ეს კი თავის მხრივ, იწვევს ფერების დამახინჯებას.



მაგალითად, სამი ძირითადი ფერის შერევა უნდა გვაძლევდეს სუფთა შავ ფერს, მაგრამ პრაქტიკულად, გაურკვეველი ჭუჭყიანი მუქი ფერი მიიღება. თუ მიღებულ ფერს დავაკვირდებით, ის უფრო მუქი ყავისფერია, ვიდრე ღრმა შავი ფერი. ამ ნაკლის კომპენსაციის მიზნით ძირითად პოლიგრაფიულ საღებავებს კიდევ დაამატეს შავი

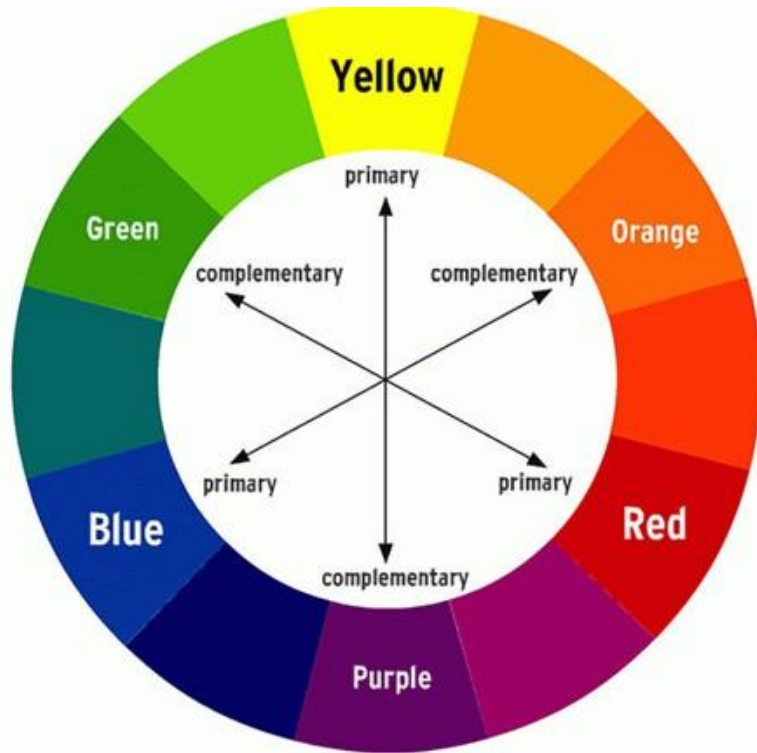
ფერი. ამიტომაც ფერთა მოდელს არა CMY არამედ CMYK უწოდეს, ე.ი. C-Cyan (ცისფერი), M-Magenta (მეწამული), Y-Yellow (ყვითელი), K-Black (შავი), ანუ ბოლო სიტყვაში აღებულია ბოლო ასო, რადგან B უკვე Blue-ს აღნიშნავს. თუმცა ზოგიერთ წყაროში ამ საკმარის ცნობილი მოდელის შესახებ წერენ, რომ K - Key color-ს აღნიშნავს, თანაც ამ Key color-ის როლში ნებისმიერი ფერი შეიძლება გამოვიყენოთ.

ფერთა ეს ორი მოდელი RGB და CMYK, ერთმანეთთან მჭიდრო კავშირშია, მაგრამ ერთი მოდელიდან მეორეში და პირიქით გადასვლის (ანუ კონვერტაციის) დროს ფერების ნაწილის დაკარგვა აუცილებლად ხდება (ანუ ინფორმაციის გარკვეული ნაწილი იკარგება, რაც ნებისმიერი ტიპის კონვერტაციისთვისაა დამახასიათებელი), რადგან ამ ორ მოდელის ფერთა დიაპაზონი სხვადასხვაა, სასურველია კონვერტაციის დროს დანაკარგები მინიმუმამდე დავიყვანოთ. ამ მიზეზით ხდება რამდენიმე მოწყობილობის ერთმანეთთან ურთიერთშეთანხმება და ზუსტი პარამეტრების მითითება (ყარგონულად ამ პროცესს "კალიბრაცია" ჰქვია). ურთიერთ შეთანხმებულად უნდა მუშაობდეს სკანერი (გამოსახულების შეტანის მოწყობილობა), მონიტორი (მის მეშვეობით განსაზღვრავენ ფერებს და ახდენენ გამოსახულების კორექციას), გრაფიკული ინფორმაციის გამოსატანი მოწყობილობა (ის ქმნის ორიგინალს პოლიგრაფიული ბეჭდვისათვის), პოლიგრაფიული საბეჭდი დანადგარები და დაზგები (საბოლოო ბეჭდვის მოწყობილობები).

იმის გამო, რომ აპარატურასთან ურთიერთობა ძალიან მოუხერხებელია, ამიტომ შეიქმნა ე.წ. პერცეპციული (ინტუიტიური) ფერთა მოდელი. ეს მოდელები ემყარება ფერადობისა და სიკაშკაშის ცნებების განცალკევებას.

**HSB მოდელი.** აუცილებლად უნდა აღვნიშნოთ, რომ RGB მოდელი ყველაზე მოსახერხებელია კომპიუტერისათვის, CMYK მოდელი გამოყენებადი

ტიპოგრაფიაში, ხოლო HSB მოდელი მოსახერხებელია ადამიანისთვის.



წარმოვიდგინოთ ფერთა წრე.

ფერთა წრეში RGB და CMY ფერები ერთმანეთთან შემდეგ ურთიერთობაში არიან: ყოველი ძირითადი ფერი ზუსტად 180 გრადუსით არის დაშორებული თავის მოპირდაპირე დამატებით (კომპლემენტარულ) ფერთან. თანაც ეს ძირითადი ფერი განლაგებულია იმ ორ ფერს შორის, რომლებსგანაც მისი მიღებაა შესაძლებელი. მაგალითად, მწვანე და წითელის შეკრებით მიიღება ყვითელი ფერი. იმისათვის, რომ რომელიმე ფერი გავაძლიეროთ, გავზარდოთ მისი ინტენსივობა, უნდა შევასუსტოთ მისი მოპირდაპირე დამატებითი ფერი. მაგალითად, თუ გვინდა ჩვენს სურათზე გავზარდოთ ლურჯი ფერების ინტენსივობა, მაშინ უნდა შევასუსტოთ წითელი ფერის შემადგენლობა. ფერთა წრის საზღვარზე განლაგებულია ეგრეთ წოდებული სპექტრული ფერები ანუ ფერთა ტონები (Hue); თითოეული ფერთა ტონი განისაზღვრება იმ შუქის ტალღის სიგრძით, რომელიც აირეკლება არაგამჭვირვალე ობიექტიდან ან გაივლის გამჭვირვალე ობიექტში. ფერის ტონი ხასიათდება თავისი მდებარეობით ფერთა წრეზე და განისაზღვრება კუთხის ზომით

დიაპაზონში 0-დან 360 გრადუსამდე. ამ ფერებს გააჩნიათ მაქსიმალური ინტენსივობა, ანუ მაგალითად ლურჯი ისეთი ლურჯია, რომ მასზე მეტი ინტენსივობის ლურჯი არ არსებობს.

შემდეგ პარამეტრს წარმოადგენს ფერის ინტენსივობა (Saturation). ეს არის ფერის მახასიათებელი, რომელიც განსაზღვრავს ამ ფერის სისუფთავეს. ფერის ინტენსივობის შემცირება ნიშნავს ამ ფერის "შეთეთრებას", ანუ ის ხდება უფრო პასტელის, უფრო, განზავებული. ერთნაირი ინტენსივობის ფერები ფერთა წრეზე განლაგებული არიან კონცენტრულ წრეებზე. ანუ ჩვენ შეგვიძლია ვილაპარაკოთ, მაგალითად, მწვანე და მეწამული ფერების ერთნაირ ინტენსივობაზე, თანაც რაც უფრო ახლოსაა ფერი წრის ცენტრისაკენ, მით უფრო განზავებული ფერი მიიღება. წრის ცენტრში ფერი მაქსიმალურად განზავებულია და ბოლოს თეთრ ფერად იქცევა.

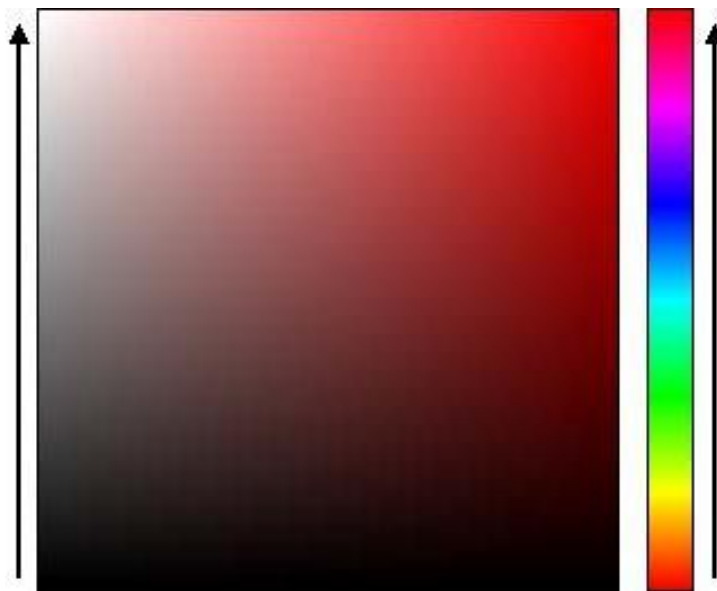
ამიტომ Saturation პარამეტრთან მუშაობის დროს შეიძლება წარმოვიდგინოთ, რომ ჩვენ სუფთა სპექტრალურ ფერს ვამატებთ თეთრი საღებავის გარკვეულ პროცენტს.

მორიგი პარამეტრი ფერის სიკაშკაშე (Brightness) - ფერის ეს პარამეტრი აჩვენებს, თუ რამდენად "განათებულია" ან "ჩაბნელებული" ფერი. ფერის სიკაშკაშის დაკლება ნიშნავს მის "ჩაშავებას" ანუ მასში შავი ფერის ჩამატებას. ამიტომ სიკაშკაშის პარამეტრთან მუშაობა ნიშნავს სუფთა სპექტრალურ ფერში გარკვეული პროცენტული რაოდენობით შავი საღებავის დამატებას.

ზემოთ თქმულიდან გამომდინარე, ნებისმიერი ფერი შეიძლება მივიღოთ სპექტრალური ფერიდან მასში გარკვეული პროცენტული რაოდენობით თეთრი ან შავი "საღებავის" დამატებით. ე.ი. ფაქტიურად ნაცრისფერი ფერის ჩამატებით.

აღნიშნული მიდგომა და ამ მიდგომაზე დამყარებული ფერთა მოდელი უკვე ფერთან მუშაობის ტრადიციულ პრინციპებს იზიარებს. შესაძლებელია, თავიდან ფერის ტონი (Hue) , ხოლო შემდეგ ინტენსივობა (Saturation) და სიკაშკაშე

(Brightness) განვსაზღვროთ. აღწერილმა ფერთა მოდელმა მიიღო თავისი სახელწოდება პარამეტრების პირველი ასოების მეშვეობით - HSB. მარცხნიდან მარჯვნივ იცვლება ინტენსივობა, ხოლო ქვევიდან ზევით იცვლება სიკაშკაშე. მარჯვნივ მოთავსებულია ფერის ტონის სკალა. ეს ფერთა მოდელი კარგად აღქმადია ადამიანისათვის, ფერის ტონი შუქის ტალღის სიგრძის ექვივალენტურია, ინტენსივობა- ტალღის ინტენსივობას აღნიშნავს, სიკაშკაშე კი - შუქის რაოდენობას აღნიშნავს.



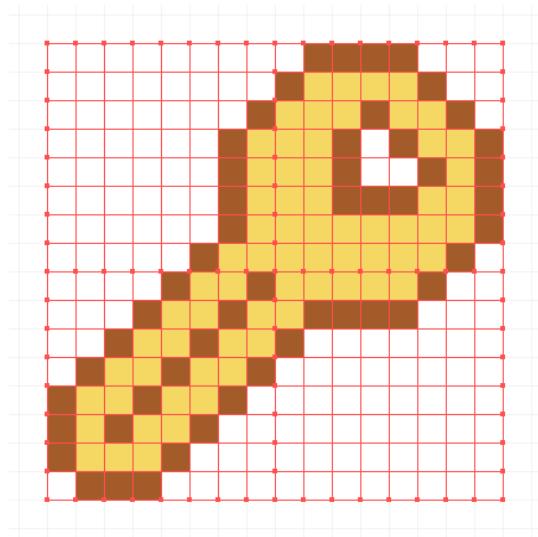
ამ მოდელის ნაკლს წარმოადგენს შემდეგი ფაქტი: მონიტორის ეკრანზე გამოსახვის დროს გამოსახულება უნდა გარდავექმნათ RGB მოდელში, ხოლო პოლიგრაფიული ანაბეჭდის მისაღებად - CMYK მოდელში.

## რასტრული ანუ პიქსელური გრაფიკის ცნება

რასტრულ გრაფიკას იყენებენ ელექტრონულ პუბლიკაციებში და პოლიგრაფიული გამოცემებისათვის. რასტრული გრაფიკის მეშვეობით მიღებული ილუსტრაციები ძალიან იშვიათად იხატება ადამიანის მიერ, ან კომპიუტერული პროგრამების დახმარებით. უფრო ხშირად გამოიყენება მხატვრულად ქაღალდზე

გაფორმებული ილუსტრაცია, რომელიც სკანერით ელექტრონულ ფორმაში არის გადაყვანილი, ან ფოტოსურათები. უკანასკნელ დროს რასტრული გამოსახულებების კომპიუტერში ელექტრონული სახით მოთავსებისათვის გამოიყენება ციფრული ფოტო და ვიდეოკამერები. სწორედ ამ მოსაზრებების გამო რასტრული გრაფიკული რედაქტორების უმრავლესობა ორიენტირებულია არა გამოსახულების შექმნაზე, არამედ მათ დამუშავებაზე.

რასტრი ეწოდება წერტილთა მართკუთხა ბადეს, რომელიც ქმნის გამოსახულებას კომპიუტერის ეკრანზე. რასტრის მეშვეობით ხდება ეკრანზე გამოსახულების ფორმირება. რასტრის ყოველი წერტილი ხასიათდება შემდეგი პარამეტრებით: თავისი ადგილმდებარეობით ეკრანზე და თავისი ფერით მონიტორის ფერად რეჟიმში ან სიკაშკაშით, თუ მონიტორი მონოქრომულ (შავ-თეთრ) რეჟიმში მუშაობს. რასტრულ გრაფიკაში გამოსახულება წარმოდგენილია რიცხვების მასივის სახით. მარტივად რომ ვთქვათ, მრავალჯერადი გადიდების დროს გამოსახულებები მოზაიკასავით გამოიყურება და პატარა ხვრელებიანი ბადის მსგავსია.



თვითონ ბადეს უწოდებენ რასტრულ ბარათს, ხოლო ელემენტებს, რისგანაც ფორმირებულია გამოსახულება, უწოდებენ პიქსელებს.

პიქსელი (pixel) კომპიუტერის მიერ ეკრანზე ასახული გამოსახულების უწვრილესი ელემენტია. თვითონ სიტყვა პიქსელი წარმოიშვა picture element (გამოსახულების ელემენტი) ფრაზის შემოკლებით.

პიქსელი ხასიათდება ერთგვაროვნებით - ყველა პიქსელი ზომებით ერთნაირია; გაუყოფელობით - პიქსელის შიგნით არ შეიძლება იყოს უფრო წვრილი ელემენტები. თუ პიქსელების ზომა საკმაოდ წვრილია, მაშინ ადამიანის თვალი აღიქვამს პიქსელურ მოზაიკას, როგორც ერთიან მთელ გამოსახულებას. მასშტაბირების დროს რასტრულ გამოსახულებაში წარმოიქმნება საფეხურები - (aliasing) ხარვეზები.



პირველ შემთხვევაში გამოსახულება ნატურალურ ზომებშია, მეორე შემთხვევაში - გადიდებული ზომებითაა, მესამე შემთხვევაში გადიდების დროს გამოყენებულია anti-aliasing ეფექტი. რასტრულ რედაქტორებში ეს პიქსელიზაციის ეფექტი შეიძლება ნაწილობრივ დავმალოთ სპეციალური ოპერაციის მეშვეობით. ამ ალგორითმს anti-aliasing ეფექტი ეწოდება.

რასტრული გრაფიკის რეზოლუცია (resolution-გადაწყვეტა) - დისკრეტული ელემენტების რაოდენობა სიგრძის ერთეულზე.

სიგრძის ერთეულად მიღებულია დიუიმი (ინჩი), რომელიც 25,4 მმ-ის ტოლია. როგორც ზემოთ იყო ნაჩვენები, დისკრეტული ელემენტი პიქსელია. მამასადამე რეზოლუცია შეიძლება განვსაზღვროთ, როგორც დიუიმში პიქსელების რაოდენობა, ჩვეულებრივ აღინიშნება როგორც ppi , რაც არის pixels per inch ( პიქსელები ყოველ დიუიმზე).

რეალურ ცხოვრებაში რეზოლუციასთან (გადაწყვეტასთან) დაკავშირებული ტერმინოლოგია ცალსახად არ არის განსაზღვრული. სხვადასხვა მოწყობილობებისათვის, რომელზეც ხდება გამოსახულების გამოტანა, შესაძლებელია რეზოლუციის საზომი სხვადასხვა სახის ერთეულების გამოყენება:

1. spi (sample per inch) - ელემენტები დუიმზე;
2. dpi (dot per inch) - წერტილები დიუიმზე;
3. ppi (pixel per inch) – პიქსელები დიუიმზე;
4. lpi (line per inch)-ხაზები დიუიმზე.

## რეზოლუციის სახეობები

კომპიუტერულ გრაფიკაში რეზოლუციის ცნებასთან დაკავშირებით არ არის მიღებული ერთიანი ტერმინოლოგია და ცნებები, რაც ხშირად გაუგებრობას იწვევს. ეს ხდება იმ მიზეზით, რომ საქმე გვაქვს სხვადასხვა ობიექტების რამდენიმე თვისებასთან ერთდროულად. მკაცრად უნდა განვასხვავოთ ეკრანის რეზოლუცია, საბეჭდი მოწყობილობის რეზოლუცია, გამოსახულების რეზოლუცია. ყველა ეს ცნება სხვადასხვა ობიექტს შეეხება. ერთმანეთთან ეს რეზოლუციები დაკავშირებული არ არის მანამ, სანამ არ დაგვჭირდება გავიგოთ, თუ რა ფიზიკური ზომის იქნება სურათი მონიტორის ეკრანზე, რა ზომის იქნება ის ქაღალდზე და რა ზომის იქნება მიღებული ფაილი თუნდაც მყარ დისკზე.

ეკრანის რეზოლუცია - ეს არის კომპიუტერის მონიტორზე და ვიდეობარათზე დამოკიდებული სიდიდე, ის აგრეთვე დამოკიდებულია ოპერაციულ სისტემაზე (ანუ სისტემაში დაყენებულ პარამეტრებზე). ეკრანის რეზოლუცია (რომელსაც ზოგჯერ ეკრანის გარჩევადობასაც უწოდებენ) იზომება პიქსელებში და ის გვიჩვენებს მთლიან ეკრანზე გამოსახულების ზომებს.

პრინტერის რეზოლუცია- ეს არის პრინტერის თვისება. ის აჩვენებს, თუ რამდენი წერტილი შეიძლება დავბეჭდოთ ერთეულოვანი სიგრძის მონაკვეთზე. ის იზომება

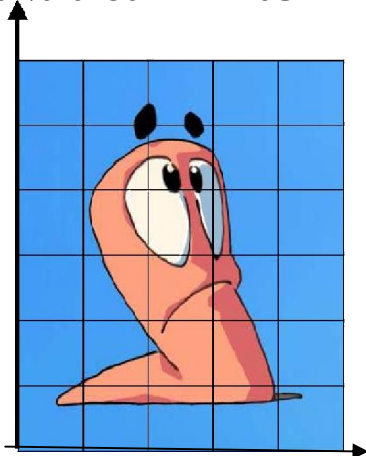


dpi (წერტილები დიუიმზე) და ის განსაზღვრავს გამოსახულების ზომას გამოსახულების წინასწარ ცნობილი ხარისხის შესაბამისად. შეიძლება იყოს პირიქით სიტუაცია, ანუ რეზოლუცია აღნიშნავდეს გამოსახულების ზომას წინასწარ ცნობილი სურათის ზომისათვის.

გამოსახულების რეზოლუცია - ეს არის თვითონ გამოსახულების თვისება. ის იზომება პიქსელებში დიუიმზე და წინასწარ უნდა იყოს მითითებული ფაილის შექმნის დროს გრაფიკულ რედაქტორში ან სკანირების დროს. ფაილის რეზოლუცია ინახება თვით ფაილში და დამოკიდებულია ფაილის ფიზიკურ ზომებზე. გამოსახულების ფიზიკური ზომები (ეს არ არის მარტო მოცულობა ბაიტურ სისტემაში) იზომება პიქსელებში, ან სიგრძის საზომ ერთეულებში: მილიმეტრებში, სანტიმეტრებში და დიუიმებში.

**გამოსახულების კოდირების პრინციპები.** განვიხილოთ ახლა გამოსახულების კოდირების პროცესი (გამოსახულების ციფრული სახით წარმოდგენა).

გამოსახულება (ანუ ორიგინალი) იყოფა დისკრეტულ ელემენტებად (პიქსელებად) და რადგანაც ეს ხდება დეკარტის კოორდინატთა სისტემაში, ელემენტებს მართკუთხა ფორმა აქვს.



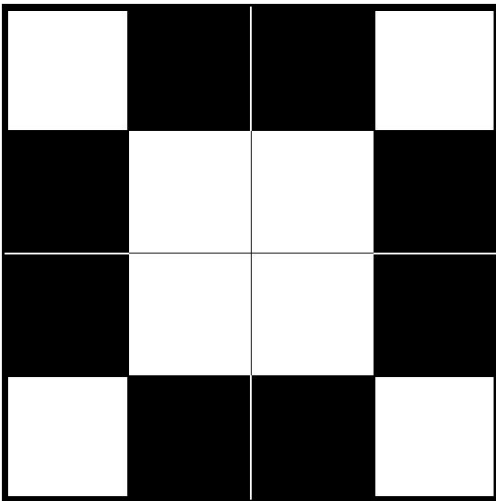
ორიგინალი დეკანტის კოორდინატებში თავსდება, მის ქვედა ნაწილში  $ox$  ღერძია განლაგებული, ხოლო მარცხენა მხარეს -  $oy$  ღერძია განლაგებული. ორივე ღერძზე უნდა მიეთითოს ერთნაირი საზომი ერთეულები.

დისკრეტიზაციის ბადაე წარმოადგენს მატრიცას, მისი ელემენტები გადანომირილია და მათში იწერება მნიშვნელობები. მატრიცას, რომელიც იქმნება

გრაფიკული პროგრამის საშუალებებით, უწოდებენ `map` (ბარათი), რადგან მასში იწერება ინფორმაციის ბიტები, მან მიიღო `bitmap` (ბიტური ბარათის) დასახელება.

ახლა განვიხილოთ ბიტური ბარათის შევსების მექანიზმი. ყველაზე მერტივ შემთხვევაში ვიხილავთ ორფეროვან (მონოქრომულ შავ-თეთრ) შემთხვევას, როცა ხდება ორიგინალის კოდირება ორი მდგომარეობით. იქ, სადაც დისკრეტულ უჯრებს თეთრი ფერი აქვს, მატრიცის (ბიტური ბარათის) შესაბამის ელემენტებში ნულები იწერება, ხოლო სადაც შავი ფერი აქვს - ერთიანები. ყველა უჯრის მთლიანად შევსების შემდეგ ორიგინალის კოდირება დასრულებულია და ცხრილი შეიძლება ციფრულ მოწყობილობაში მოვათავსოთ.

შავ-თეთრი გამოსახულება და მისი ბიტური ანალოგი.



0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

აღნიშნული კოდირების სისტემა მხოლოდ ერთი კერძო შემთხვევაა, მონოქრომული გამოსახულებების კოდირების დროს. რაც შეეხება ფერად გამოსახულებას, სხვადასხვა ფერთა მოდელებისა და კოდირების სხვადასხვა სისტემაში განსხვავებულ მექანიზმთან გვიწევს მუშაობა. რეალურ სამყაროში საკმაოდ მრავალფეროვან სიტუაციასთან გვაქვს საქმე. ანუ შეიძლება გამოვყოთ კლასიკური შავ-თეთრი ფოტო სურათები, მხატვრული ტექნიკა - ფანქრით ხატვა, აკვარელი, ზეთის საღებავებით ხატვა, გრაფიურების ტექნიკა და სხვა.

თავდაპირველად ფერადი გამოსახულებებისათვის გამოიყენეს ფერის (ტონის) გრადაციების რაოდენობის ზრდის მექანიზმი. ეს ზრდა დაიწყო 16 გრადაციიდან, შემდეგ - 64, მერე - 256, და ამჟამად 1024 გრადაციაც კი გვხვდება.

რადგან ბაიტი=8 ბიტს (რვა თანრიგს), ამიტომ ტონის გრადაციების რაოდენობა  $2^8=256$  რიცხვთანაა დაკავშირებული.

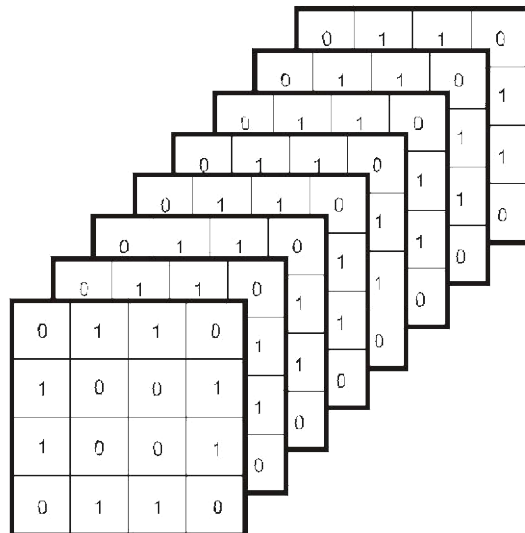
აქედან გამომდინარე, კვანტური ცხრილი, რომელიც ემსახურება ფერის ტონის უწყვეტ ცვლილებას, შედგება 256 სტრიქონისაგან ანუ 256 ორობითი კოდისაგან, ამ კოდების დიაპაზონი იცვლება 00000000-დან 11111111-მდე. საწყის კოდს, ანუ 0-ს (ათობით სისტემაში) შეუსაბამებენ შავ ფერს, ხოლო ბოლო კოდს - 255-ს (ათობით სისტემაში) შეუსაბამებენ თეთრ ფერს. შუაში 254 კოდი ნაცრისფერის გრადაციებია, დაწყებული მუქი, თითქმის შავიდან ღია ნაცრისფერამდე, თითქმის თეთრამდე.

ახლა დავუბრუნდეთ ვირტუალურ ციფრულ მატრიცას - ბიტურ ბარათს. გავიხსენოთ, რომ ყოველ უჯრაში თითო ბიტი იწერება, 0 ან 1. რადგან ფერად გამოსახულებაში ფერთა ტონები უნდა იყოს ჩაწერილი, ამიტომ თითოეულ პიქსელში უნდა ჩაიწეროს არა ერთი ორობითი რიცხვი, არამედ რვა ბიტი (1 ბაიტი). ანუ თითოეული თანრიგის ნაცვლად კიდევ დამატებით შვიდი თანრიგი უნდა ჩაისვას. ეს პრობლემა გადაჭრილია ბიტური ბარათების დამატებით (საჭირო თანრიგების რაოდენობით). დამატებითი ბიტური ბარათები, ასე ვთქვათ, სივრცეში ანუ სიღრმეშია განლაგებული. ბიტური ბარათების რაოდენობა (თანრიგების ანუ ორადი ციფრების რაოდენობა) განსაზღვრავს კვანტური ცხრილის სიღრმეს. ამ პარამეტრს color dept უწოდეს, რაც პირდაპირი მნიშვნელობით ფერის სიღრმეს ნიშნავს. ფერის სიღრმე - ეს არის ბიტების (თანრიგების) ის რაოდენობა, რომელიც ერთ პიქსელს შეესაბამება (bpp). ფერის სიღრმე აღნიშნავს, თუ რამდენი ბიტია გამოყენებული გარკვეული ფერის ტონის კოდირების დროს.

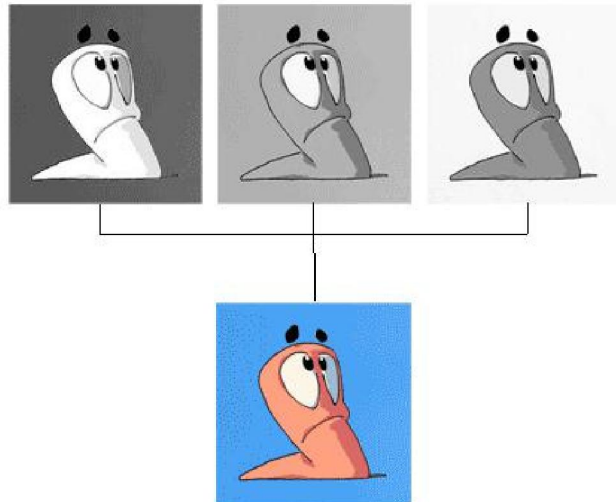
ფერის სიღრმე საკმაოდ ინფორმატიული მახასიათებელია. მას გააჩნია საზომი ერთეული. ფერის სიღრმეს ზომავენ ერთი პიქსელისათვის გამოყოფილ თანრიგთა რაოდენობით. ე.ი. ბიტური ბარათების საერთო რაოდენობა არის ფერის სიღრმის საზომი ერთეული. თუ გამოყენებულია ტონური (grayscale) გამოსახულება, მაშინ მისი ფერის სიღრმე ტრადიციულად რვა ბიტის ტოლია, მას 8-bit image-ს უწოდებენ, ხანდახან grayscale გამოსახულებას ეძახიან.

მრავალფეროვანი გამოსახულებებისათვის ტექნიკური სისტემები იყენებენ შემდეგ მეთოდს. ფერადი ინფორმაციის ციფრული წარმოდგენისათვის (და შენახვისათვის)

იყენებენ ფერად ფილტრაციას: ფერად ნაკადს უშვებენ სამ (წითელ, მწვანე და ლურჯ) ფილტრებში. ამ პროცედურის შედეგად იქმნება სამი გამოსახულება ნაცრისფერის გრადაციებში (grayscale). ამ სამ გამოსახულებაში ყოველი პიქსელი აღიწერება რვა ორობითი თანრიგის მეშვეობით, ჯამში გამოდის 24 ბიტი ანუ ფერადი გამოსახულებების აღწერის ერთ-ერთი მეთოდი არის 24-bit image. ფერის ამ სამ კომპონენტს ფერის არხებს უწოდებენ, ყოველი არხი შეიცავს ნაცრისფერი ფერის გრადაციებში (სიმკვეთრე/სიკაშკაშე) იცვლება 0-დან 255-ის ჩათვლით) წარმოდგენილ გამოსახულებას.



ასე ხდება გამოსახულების არხებად წარმოდგენა.



სურათზე ნაჩვენებია ცალკეული არხებიდან ფერადი გამოსახულების აწყობა პროგრამაში Adobe Photoshop.

## ფერთა პალიტრების შესახებ

**ფერთა პალიტრა** - ეს არის მონაცემთა ცხრილი, რომელშიც ინახება ინფორმაცია, თუ რომელი კოდით არის წარმოდგენილი ესა თუ ის ფერი. ეს ცხრილი იქმნება და ინახება გრაფიკულ ფაილთან ერთად. კომპიუტერისათვის ფერის კოდირების ყველაზე "მოსახერხებელ" მეთოდს წარმოადგენს 24-თანრიგიანი True Color რეჟიმი. ამ რეჟიმში ყოველი შემადგენელი ფერისათვის გამოყოფილია თითო ბაიტი (8 ბიტი). ე.ი. R-8, G-8, B-8 სულ 24 ბიტია. სამი კომპონენტიდან ყოველის სიმკვეთრე იცვლება 0-დან 255-ის ჩათვლით, მიღებული 16,5 მილიონი ფერიდან ნებისმიერი კომპიუტერს შეუძლია ამ სამი კოდის მეშვეობით გამოსახოს. ამ შემთხვევაში ფერთა პალიტრა არ არის საჭირო, რადგან თითოეული პიქსელისათვის 3 ბაიტით ფერის წარმოდგენა ისედაც საკმარის ინფორმაციას შეიცავს.

**ინდექსირებული პალიტრა.** თუ გამოსახულებას მხოლოდ 256 ფერი გააჩნია, კოდირება თითო ბაიტით ხდება. ამ შემთხვევაში ყოველი ფერის ტონი

მხოლოდ ერთი რიცხვით არის ჩაწერილი, ეს რიცხვი აღნიშნავს არა პიქელის ფერს, არამედ ფერის ინდექსს (მის ნომერს). თვითონ ფერი უნდა მოიძებნოს ნომრით ფერთა პალიტრაში, პალიტრა უშუალოდ ფაილს მოჰყვება. ასეთ ფერთა პალიტრებს ინდექსირებულ პალიტრებს უწოდებენ. სხვადასხვა გამოსახულებებს სხვადასხვა ფერთა პალიტრები მოჰყვება. ერთ სურათში ინდექსი 64 შეიძლება აღნიშნავდეს მწვანეს, სხვა სურათში კი - ვარდისფერს.

**ფიქსირებული პალიტრები.** ზოგიერთ შემთხვევაში ფერი შეიძლება კოდირებული იყოს 2 ბაიტით (High Color), ეკრანზე შესაძლებელია 65 ათასი ფერის გამოსახვა. ეს არ არის ყველა შესაძლო ფერი, არამედ True Color უწყვეტ ფერთა რეჟიმის მხოლოდ 1/256 ნაწილია. ასეთ წარმოდგენაში არ გამოიყენება ინდექსების პალიტრა, რადგან ფაილის ზომა ასეთი პალიტრით ძალიან დიდი იქნებოდა. ამ შემთხვევაში ფიქსირებულ პალიტრას იყენებენ. ის არ ემატება გრაფიკულ ფაილს, რადგან ერთი და იგივე კოდი ასეთი ტიპის კოდირებაში ერთი და იგივე ფიქსირებულ ფერს შეესაბამება.

**უსაფრთხო პალიტრა.** ვებ-დიზაინში და კერძოდ, ვებ-გრაფიკაში გამოიყენება უსაფრთხო პალიტრის ცნება. რადგან ინტერნეტში მონაცემთა გადაცემის სიჩქარე ჯერ კიდევ არც თუ ისე დიდია, ვებ-გვერდების გაფორმებისას ფერების წარმოდგენის დროს არ გამოიყენება 8 თანრიგზე მეტი კოდირების სისტემა. ამ შემთხვევაში ჩნდება კლასიკური პრობლემა: ვებ-დიზაინერს არანაირი ინფორმაცია არ გააჩნია, თუ რომელი ტიპის კომპიუტერში, რომელი ოპერაციული სისტემით და რა პროგრამაში დაათვალიერებენ მის ვებ-გვერდს. ამიტომ მიიღეს შემდეგი გადაწყვეტილება: ვებ-ბრაუზერების უმრავლესობაში წინასწარ არის დაფიქსირებული ერთი სტანდარტული ფერთა პალიტრა. თუ ვებ-გვერდის შემქმნელი გამოიყენებს მხოლოდ ამ პალიტრას, მაშინ ის დარწმუნებული იქნება, რომ მის ილუსტრაციას მთელს მსოფლიოში სათანადოდ დაინახავენ. ასეთ პალიტრაში არ არის 256 ფერი, მასში მხოლოდ 216 ფერია. ეს დაკავშირებულია კომპიუტერული

პლატფორმების არათავსებადობის პრობლემასთან. ამ პალიტრას უსაფრთხო პალიტრა უწოდეს.

# *Adobe Photoshop*



პრაქტიკული სავარჯიშოები Adobe Photoshop-ში



Adobe Photoshop გრაფიკული რედაქტორის მენიუს მოკლე მიმოხილვა

File/ New - გამოსახულების შექმნა (ახალი გრაფიკული ფაილის შექმნა)

File / Open - გამოსახულების გახსნა (არსებული გრაფიკული ფაილის გახსნა)

File / Save - გამოსახულების შენახვა (გრაფიკული ფაილის შენახვა)

File / Exit - ფოტოშოპიდან გამოსვლა



Edit / Cut - ამოჭრა

Edit/ Copy- კოპირება

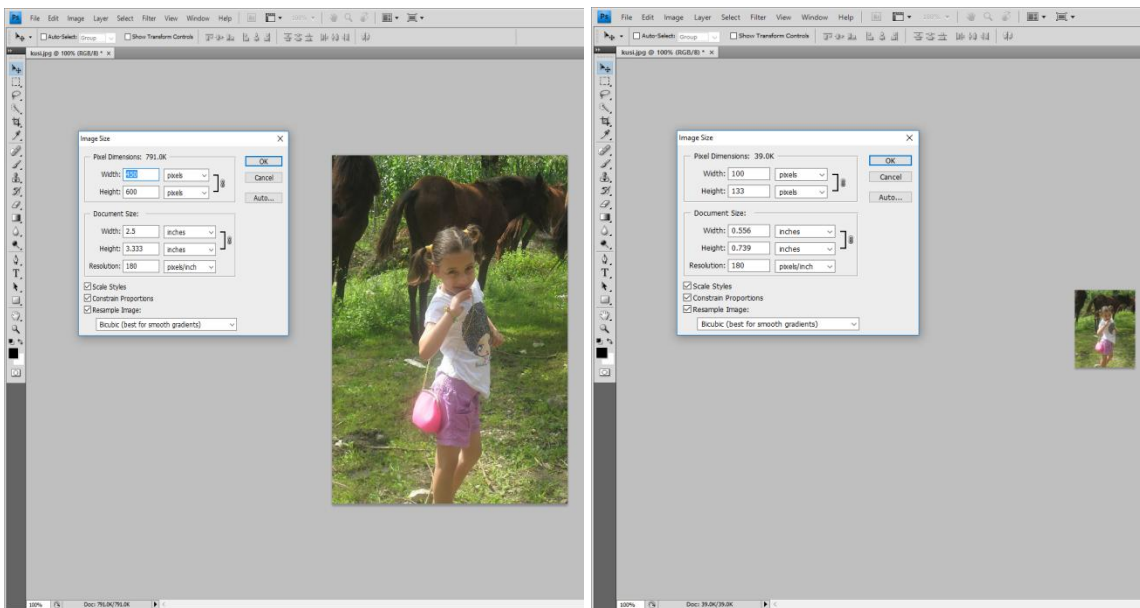
Edit/ Paste- ჩასმა

Edit / Undo - ერთი მოქმედებით უკან დაბრუნება

Edit/Redo - ერთი მოქმედებით წინ წასვლა



### Image / Image Size - ფოტოს ზომების ცვლილება



### Image / Rotation /180° - ფოტოს შემობრუნება 180°

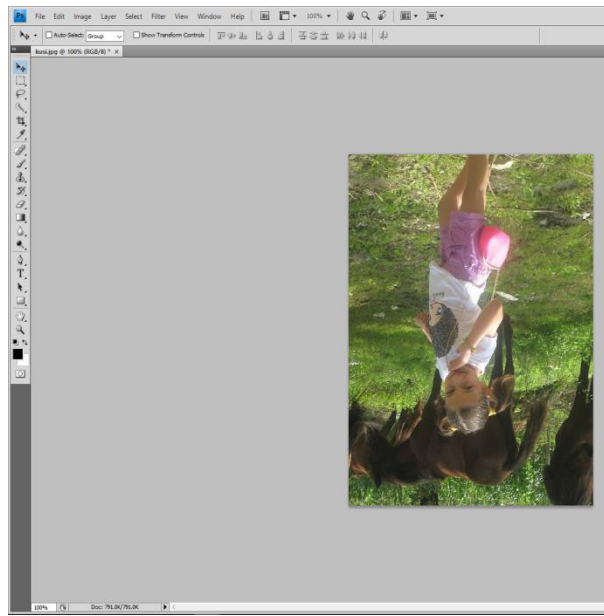
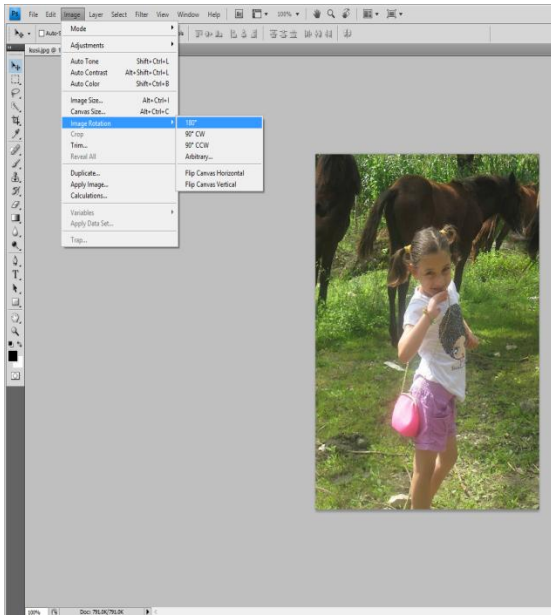
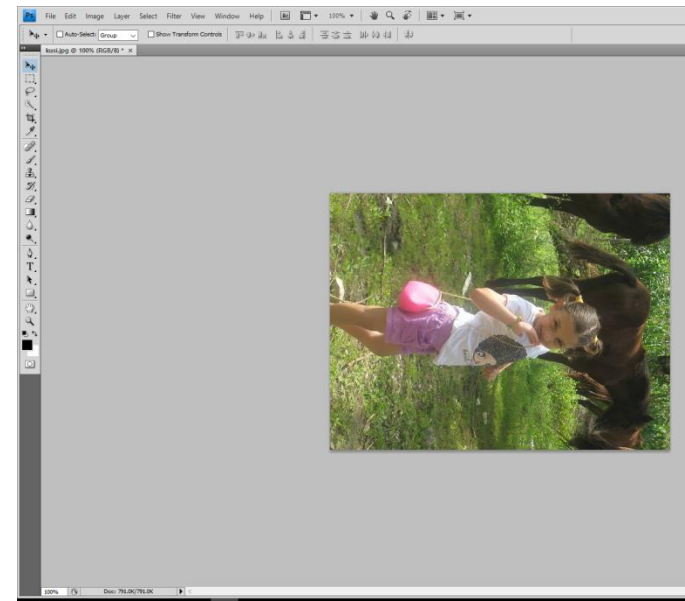
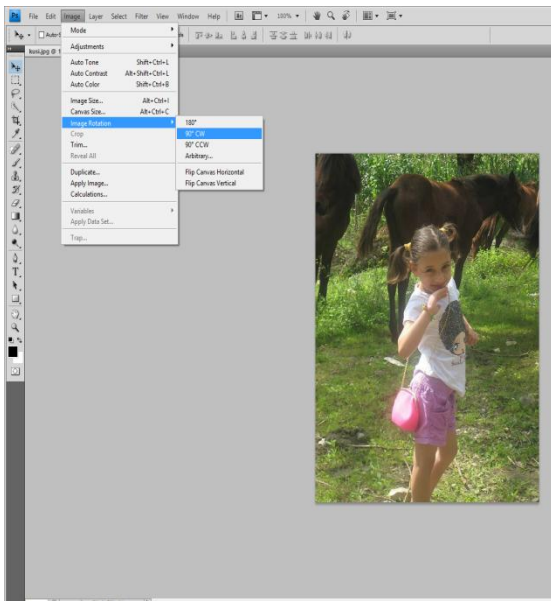
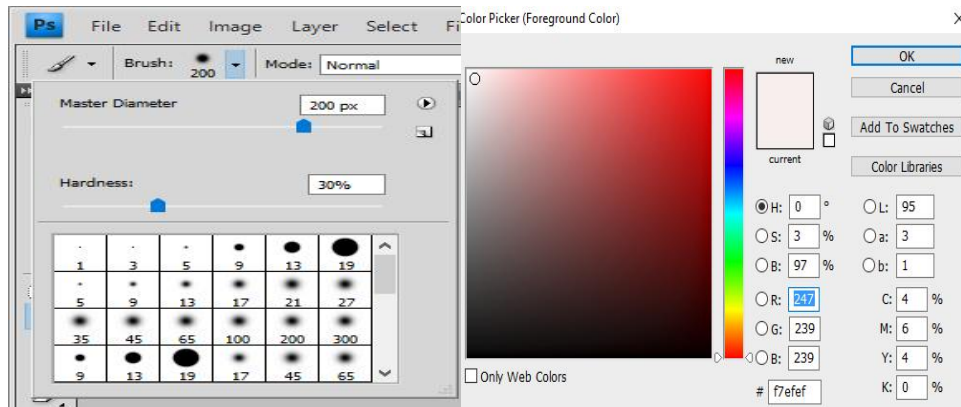


Image / Rotation / 90° - ფოტოს შემობრუნება 90°



### თემა 1 Brush Tool ინსტრუმენტის გამოყენება

გახსენით სასურველი ფოტო File /open /foto1.jpg;  
აირჩიეთ **Brush Tool(B)** ინსტრუმენტი;  
შეცვალეთ პარამეტრები : Brush / Master Diameter: 200px / Hardness: 30% /Color  
Picker\_აირჩიეთ თეთრი ფერი;  
ჩასვით მთვარე და ვარსკლავები;



საწყისი ფოტო



შედეგი


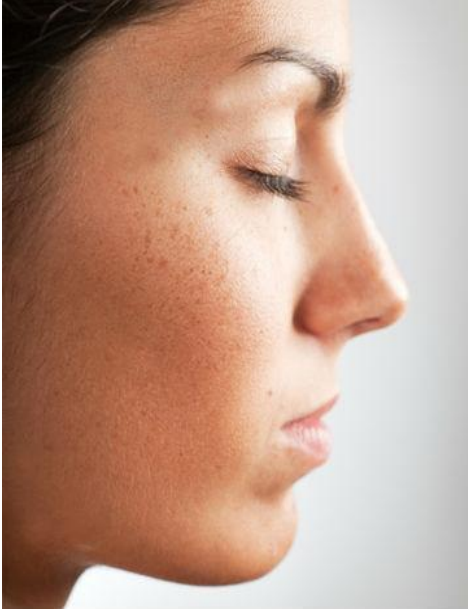


# დავალება 1

შეასრულეთ დავალება Brush Tool ინსტუმენტის გამოყენებით.

საწყისი ფოტო	შედეგი
	

თემა 2 სახეზე ლაქების გაქრობა

<p>გახსენით სასურველი ფოტო File /open /foto2.jpg;          აირჩიეთ <b>Clone&amp; Stamp Tool (S) + Alt</b> ;          გადაატარეთ ლაქების ახლოს მდებარე ადგილზე;  <b>Alt</b> ღილაკის გარეშე შტამპით გადაღებთ სახეზე ლაქები;</p>	
საწყისი ფოტო	შედეგი
	

დავალება 2.

შეასრულეთ დავალება Clone& Stamp Tool ინსტრუმენტის გამოყენებით.

საწყისი ფოტო	შედეგი
	

### თემა 3 ნაოჭების გასწორება

გახსენით სასურველი ფოტო File /open /foto3.jpg ;

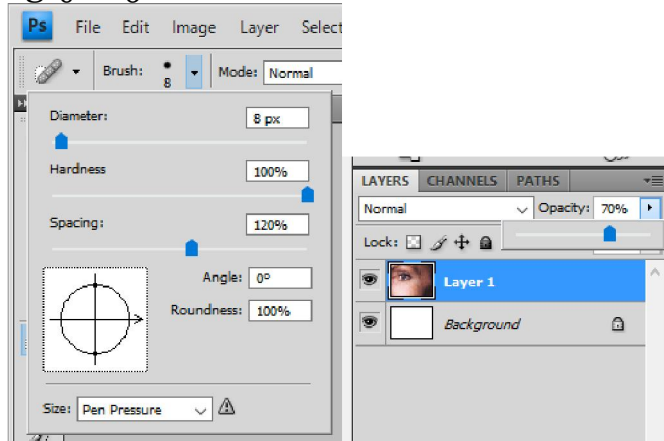
აირჩიეთ **Spot Healing Brush Tool (J)** ;

შეცვალეთ პარამეტრები: Diameter: 8 px; Spacing :120% ;

Spot Healing Brush Tool + Alt დააკლიკეთ ნაოჭების ახლოს მდებარე ადგილას ;

Alt ღილაკის გარეშე გადაატარეთ ფუნჯი ნაოჭებზე. მონიშნეთ დამუშავებული ფოტო და ჩააკოპირეთ ახალ ფენაში;

შეცვალეთ Opacity პარამეტრი/ Opacity : 70 % ამით გამოსახულება გახდება უფრო ბუნებრივი;



საწყისი ფოტო

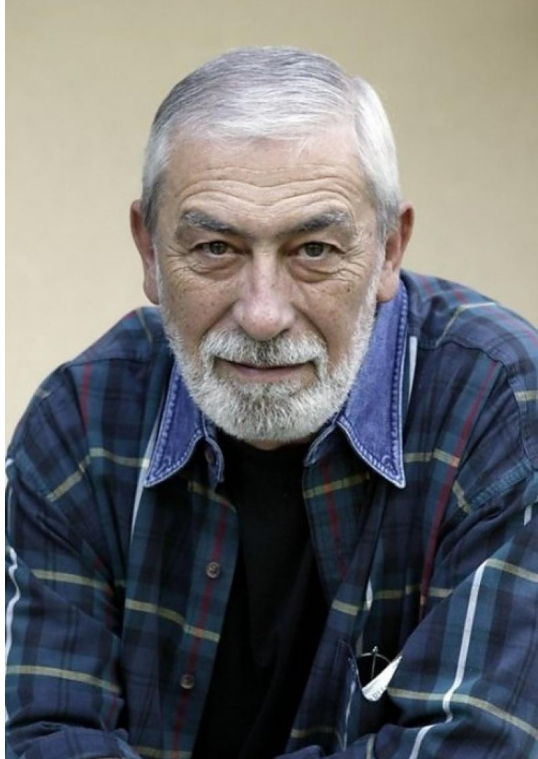
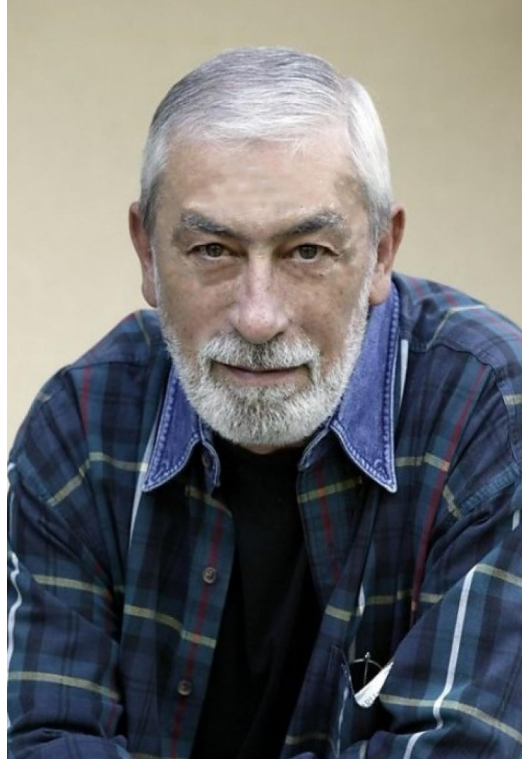


შედეგი



### დავალება 3

შეასრულეთ დავალება Spot Healing Brush Tool და Patch Tool ინსტრუმენტების გამოყენებით.

საწყისი ფოტო	შედეგი
	



თემა 4 სავიზიტო ბარათი

გახსენით სასურველი ფოტო File /open /foto4.jpg;  
აირჩიეთ **Horizontal Type Tool (T)** ინსტრუმენტი და კლავიატურიდან აკრიფეთ ტექსტი;



დავალება 4

შეასრულეთ დავალება Rectangular Marquee Tool , Paint Bucket Tool , Horizontal Type Tool , ინსტრუმენტების გამოყენებით.



თემა 5 ობიექტის ამოჭრა და ჩასმა

გახსენით სასურველი ფოტო File /open /foto5.jpg;  
აირჩიეთ **Lasso Tool (L)**;  
მონიშნეთ ობიექტი რომლის გადატანა გვსურთ;  
Ctrl +C დააკოპირეთ და ჩასვით სასურველ ადგილას;  
ახალ ფენაზე **Eraser Tool (E)** საშლელით წაშალეთ ზედმეტი არე, რაც კოპირების დროს გადმოყვა;

საწყისი ფოტო

შედეგი



## დავალება 5

შეასრულეთ დავალება Lasso Tool და Eraser Tool ინსტრუმენტების გამოყენებით.



თემა 6 ობიექტის გაქრობა

გახსენით სასურველი ფოტო File /open /foto6.jpg ;  
აირჩიეთ **Polygonal Lasso Tool (L)**; მონიშნეთ ობიექტი რომლის გაქრობაც გსურთ;  
**Ctrl +X** ღილაკების კომბინაციით ამოჭერით ობიექტი;  
**Clone& Stamp Tool (S) + Alt** ღილაკით ამოჭრილი ობიექტის ახლოს გადაატარეთ;  
**Clone& Stamp Tool** ინსტრუმენტის გამოყენებით შეავსეთ ამოჭრილი არე;

საწყისი ფოტო



შედეგი



## დავალება 6

შეასრულეთ დავალება Polygonal Lasso Tool და Clone& Stamp Tool ინსტრუმენტების გამოყენებით.



თემა 7 სარეკლამო ბანერი

გახსენით სასურველი ფოტო File /open /foto7.jpg ;

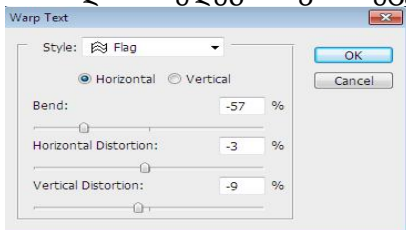
შექმენით ახალი ფენა , სადაც **Gradient Tool (G)** ინსტრუმენტის გამოყენებით გააფორმებთ მოცემულ ფენას;

**Magic Tool (W)** ღილაკით ფოტოში მონიშნეთ სასურველი არე და გადაიტანეთ თქვენს მიერ შექმნილ ახალ ფენაზე;

**Warp Text** -ში შეცვალეთ პარამეტრები: **Style: Flag; Bend : -57%; Horizontal Distortion : -3%; Vertical Distortion : -9%;**

აკრიფეთ ტექსტი: **Flowers ;**

საბოლოო შედეგისთვის შეცვალეთ ფილტრები: **Filter/ Stylize / Glowing Edges;**



საწყისი ფოტო


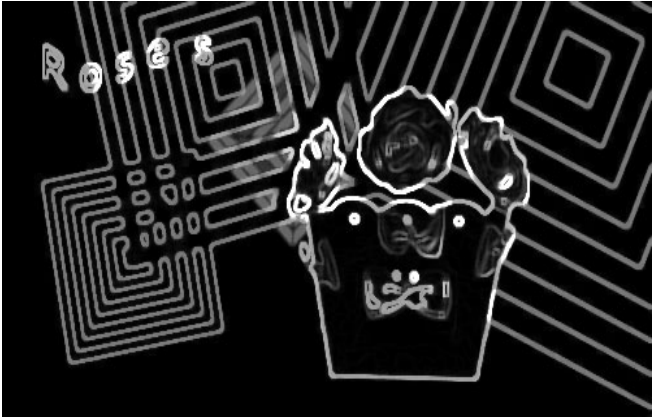


შედეგი



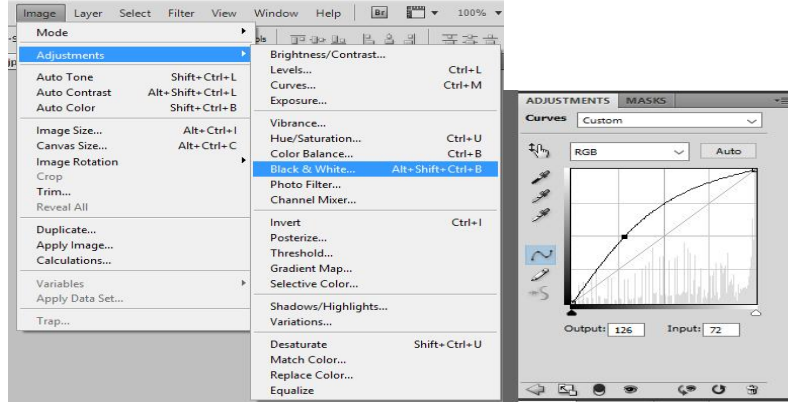
## დავალება 7

შეასრულეთ დავალება Gradient Tool , Magic Tool და Horizontal Type Tool ინსტრუმენტების გამოყენებით.

საწყისი ფოტო	შედეგი
	

**თემა 8 საბავშვო გასაფერადებელი**

გახსენით სასურველი ფერადი ფოტო File /open /foto8.jpg;  
 ფოტო გადააქციეთ შავ - თეთრ ფოტოდ Image/ Ajustment / Black& White /ok;  
 შავ- თეთრი ფოტო გააფერმკრთალეთ ;  
 Curves / Custom -ში შევცვალოთ პარამეტრები Output: 126 ; Input: 72;



საწყისი ფოტო

შედეგი





## დავალება 8

შეასრულეთ დავალება Black& White და Filter გამოყენებით.

საწყისი ფოტო

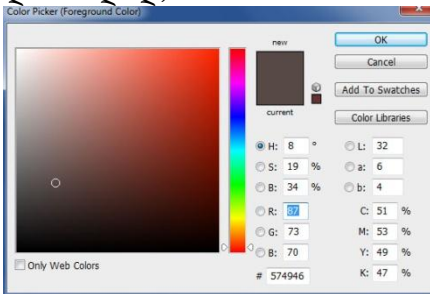


შედეგი



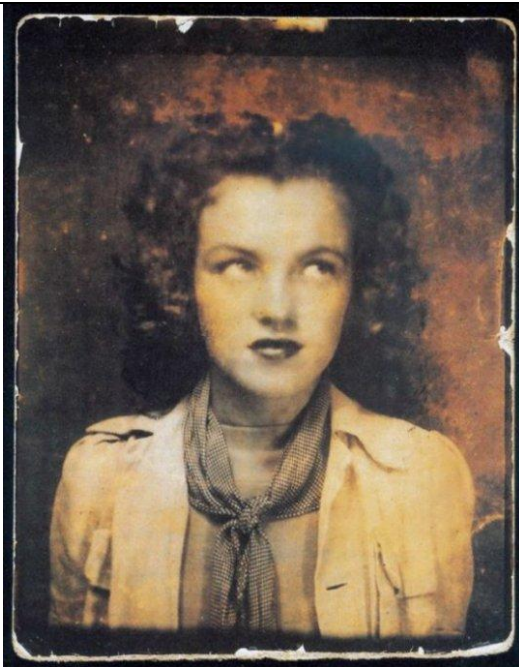
**თემა 9 ძველი ფოტოს აღდგენა**

გახსენით სასურველი ფოტო File /open /foto9.jpg  
File/ New ბრძანებით შექმენით ახალი ფოტო , რომელიც გამოდგება ფონის აღსადგენად;  
შეარჩიეთ სასურველი ფონის ფერი და **Paint Bucket Tool** ინსტრუმენტით გააფერადეთ ახალი ფოტო;  
სასურველი ფერი მიიღება **Color Picker** ფანჯარაში პარამეტრების ცვლილებით ;  
**Lasso Tool** ინსტრუმენტით მონიშნეთ გოგონა და გადაიტანეთ ახალ ფოტოზე;  
ზედმეტი ნაწილები წაშალეთ **Eraser Tool**-ის გამოყენებით;  
**Clone& Stamp Tool** -ით აღადგინეთ დაზიანებული არეები - გოგონას თმებზე და ტანსაცმელზე;  
**Blur Tool** ინსტრუმენტი გადაატარეთ პრობლემურ არეებზე, ხარვეზების დასამალად;



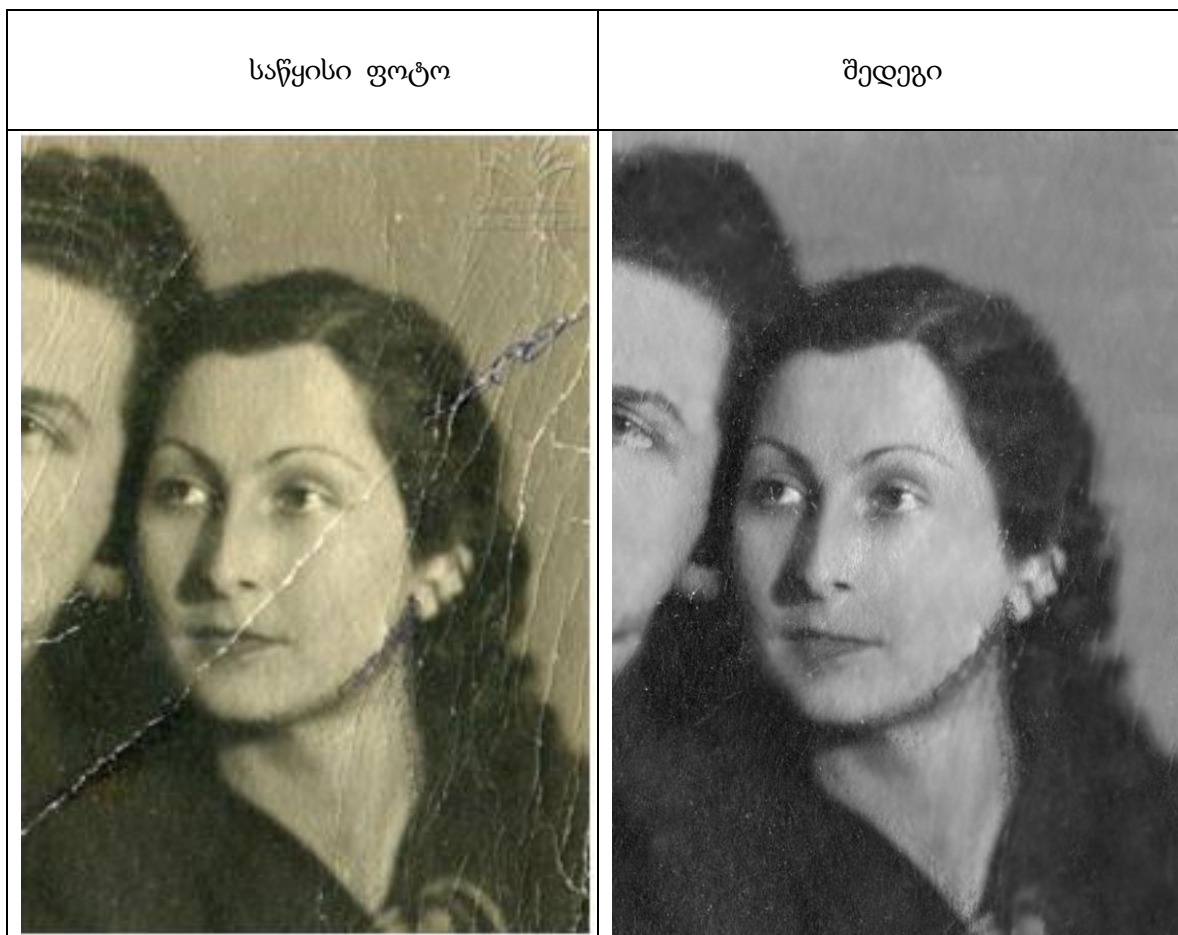
საწყისი ფოტო

შედეგი



## დავალება 9

შეასრულეთ დავალება Clone& Stamp Tool და Blur Tool ინსტუმენტების გამოყენებით.



თემა 10 მოსაწვევი ბარათი

გახსენით სასურველი ფოტო File /open /foto10.jpg სასურველია ფოტო იყოს სილუეტების; File/ New ბრძანებით შექმენით ახალი ფოტო;  
**Rectangle Tool** ინსტრუმენტით ჩახაზეთ სამი ერთი ზომის მართკუთხედი;  
**Gradient Tool** -ის გამოყენებით შეავსეთ მართკუთხედების შიდა არეები;  
აირჩიეთ **Brush Tool / Brush** განსხვავებული ფუნჯი და გააფორმეთ მართკუთხედების არეები;  
სასურველი სილუეტი მონიშნეთ **Polygonal Lasso Tool** ინსტრუმენტით და დააკოპირეთ მართკუთხედებზე;  
**Filter/ Stylize/ Diffuse.../** ბრძანების შესრულებით მიიღებთ საბოლოო შედეგს;

საწყისი ფოტო



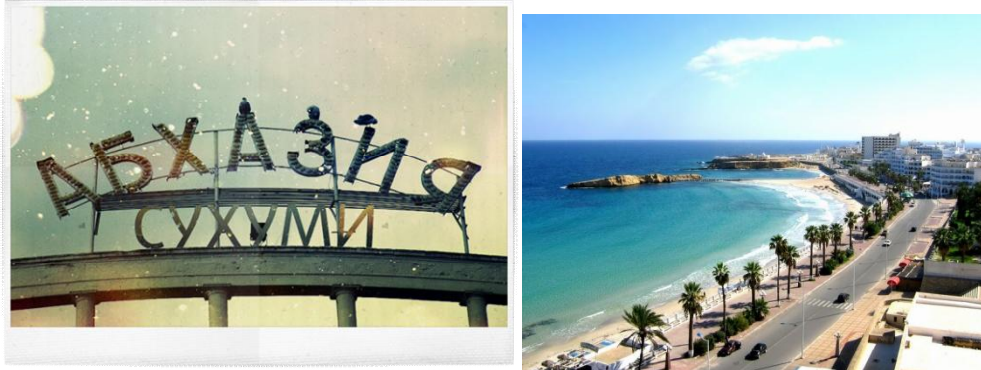
შედეგი



## დავალება 10

შეასრულეთ დავალება Rectangle Tool, Gradient Tool, Polygonal Lasso Tool და Brush Tool ინსტრუმენტების გამოყენებით.

საწყისი ფოტო

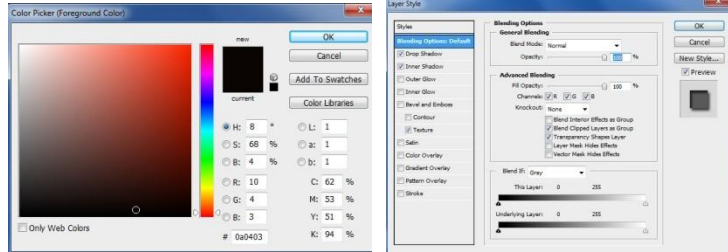


შედეგი



**თემა 11 წიგნის ყდის გაფორმება**

გახსენით სასურველი ფოტო File /open /foto11.jpg ძველი გაზეთი;  
 გახსენით ასევე ფოტო ძველი ხის ტექსტურით;  
**Move Tool** ინსტრუმენტით გადაიტანეთ ტექსტურა ძველი გაზეთის ფოტოზე;  
**Ctrl+ T** ღილაკების კომბინაციით დააყენეთ სასურველ ზომაზე;  
 შეცვალეთ პარამეტრები Layers / Opacity: 50% იმისათვის, რომ ზედა ფენა იყოს უფრო გამჭვირვალე;  
 გახსენით ახალი ფოტო , რომელზეც არის გამოსახული პალმა. **Polygonal Lasso Tool** ინსტრუმენტით მონიშნეთ პალმის არე, ამოჭერით და ჩასვით ძველი გაზეთის ფოტოზე;  
 აირჩიეთ **Paint Bucket Tool** ინსტრუმენტი და პალმის შიდა არე გააფერადეთ;  
 გააფორმეთ სხვადასხვა ზომის წარწერებით (ტექსტი: „სოხუმი“);  
**Layer Style** დიალოგურ ფანჯარაში გაააქტიურეთ **Drop Shadow** და **Inner Shadow**, ამით შეიცვლება ტექსტის სტილი; **Blur Tool** ინსტრუმენტით დაამუშავეთ ფოტო;



საწყისი ფოტო



შედგენი



დავალება 11

შეასრულეთ დავალება Move Tool , Polygonal Lasso Tool , Paint Bucket Tool და Blur Tool ინსტუმენტების გამოყენებით.

საწყისი ფოტო

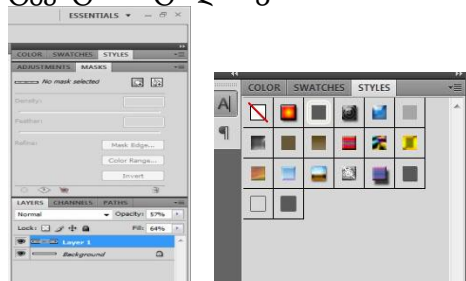


შედეგი

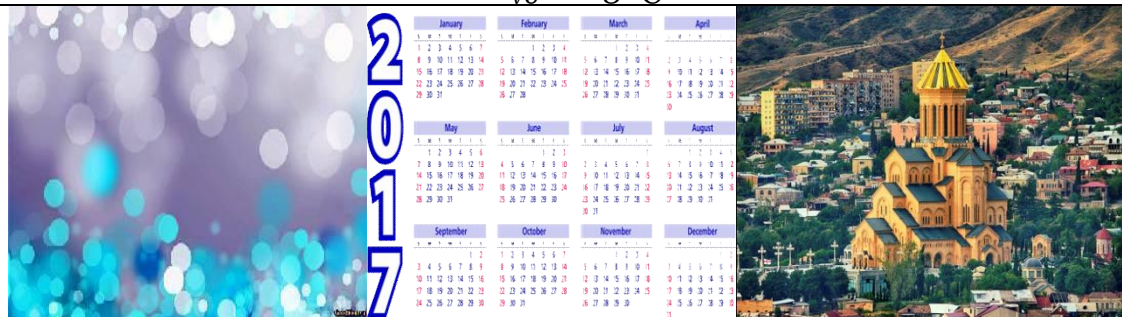


თემა 12 კალენდარი

გახსენით სასურველი ფოტო File /open /foto12.jpg;  
 გახსენით ასევე 2017 წლის კალენდარი და **Move Tool** ინსტრუმენტით გადაიტანეთ foto 12 - ზე;  
**Ctrl+T** კლავიატურის კომბინაციით კალენდარი დააყენეთ თქვენთვის სასურველ ზომაზე და მოათავსეთ ფოტოს მარჯვენა მხარეზე;  
 საბოლოო ეფექტისთვის დაამატეთ ტექსტურის ახალი ფენა;  
**Styles/ Masks** შეცვალეთ პარამეტრები **Opacity: 57%; Fill: 64%** ფენა არის უფრო მკრთალი და კარგად იკითხება კალენდარი;  
**Eraser Tool** ინსტრუმენტით წაშალეთ ტამარზე ზედმეტი ლაქები, რომელიც ჰქონდა ტექსტურას;  
 ფოტოს მარჯვენა ზედა კუთხეში დააწერეთ 2017 **Horizontal Type Tool** -ით;  
 ტექსტის სტილი ავირჩიოთ **Double Ring Glow**;



საწყისი ფოტო



შედეგი





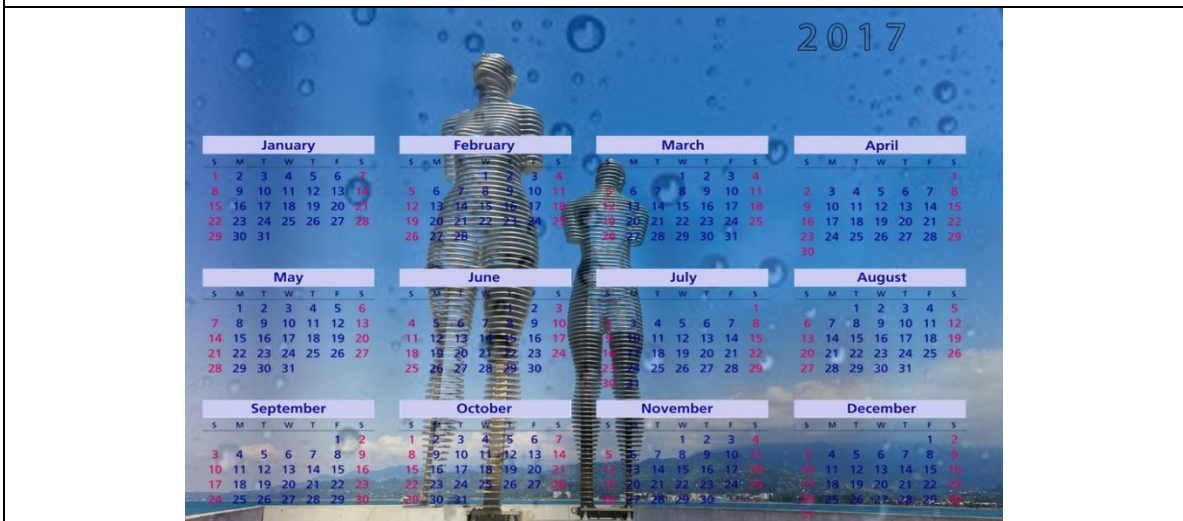
## დავალება 12

შეასრულეთ დავალება Move Tool და Horizontal Type Tool ინსტრუმენტების გამოყენებით.

### საწყისი ფოტო

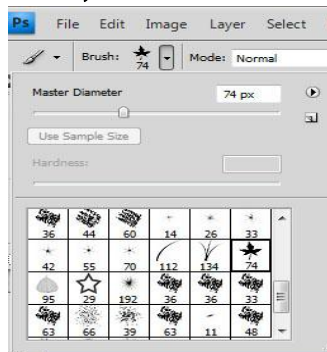


### შედეგი



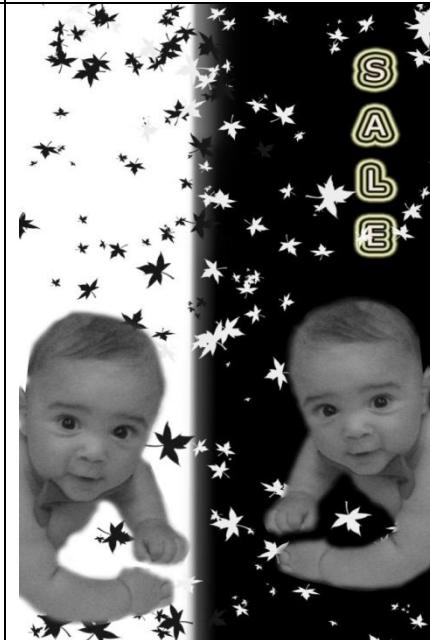
**თემა 13 ფასდაკლება ბანერზე**

გახსენით სასურველი ფოტო File /open /foto13.jpg;  
 გახსენით ახალი ფოტო File / New ბრძანებით და **Gradient Tool** ინსტრუმენტით გააფერადეთ ფოტოს ერთი ნაწილი შავ ფერში, ხოლო მეორე ნაწილი თეთრ ფერში; გაააქტიურეთ **Brush Tool** ინსტრუმენტი და აირჩიეთ ფოტოლის ფორმის ფუნჯი; სხვადასხვა ზომის ფოთლებით გააფორმეთ ფონი; შემდეგ შეცვალეთ ფუნჯის ფერი და გააფორმეთ დარჩენილი ნაწილი;  
 Foto13 -ზე მონიშნეთ ბავშვი **Lasso Tool** ინსტრუმენტით და გადაიტანეთ ახალ ფოტოზე; **Image / Image Rotation/ Flip Canvas Horizontal** ბრძანების შესრულებით შემოაბრუნეთ;  
**Vertical Type Tool** ინსტრუმენტით აკრიფეთ ტექსტი “SALE”, სტილი **Double Ring Glow** ;



საწყისი ფოტო

შედეგი



### დავალება 13

შეასრულეთ დავალება Gradient Tool , Brush Tool, Lasso Tool და Horizontal Type Tool ინსტუმენტების გამოყენებით.

საწყისი ფოტო



შედეგი



თემა 14 3D ფიგურები

File / New ბრძანებით გახსენით ახალი ფოტო;  
**3D / New Shape From Layer/ Wine Bottle** ბრძანებით შექმენით  
სამგანზომილებიანი ღვინის ბოთლი;  
თეთრ მართკუთხედზე **Horizontal Type Tool** ინსტრუმენტის გამოყენებით  
დაბეჭდეთ წარწერა „სოხუმი“.

შედეგი



*დავალება 14*

3D-ს გამოყენებით შექმენით სამგანზომილებიანი თუნუქის ბოთლი.

საწყისი ფოტო



შედეგი



*თემა 15 პიქსელური აფეთქება ფოტოზე*

გახსენით სასურველი ფოტო File /open /foto15.jpg;  
ახალი ფუნჯის შექმნა: File/ New გახსენით ახალი დოკუმენტი, **Rectangular Marquee Tool** ჩახაზეთ მართკუთხედი და **Shift+D** დილაკებით მონიშნეთ მართკუთხედი **Paint Bucket Tool** ინსტრუმენტით, გააფერადეთ შავ ფერში;  
**Define Brush Preset / Ok** ბრძანების შესრულებით დაემატა ახალი ფუნჯი;  
ახალი ფუნჯით სხვადასხვა ფერის მართკუთხედები ჩახაზეთ;  
**Magic Tool** ინსტრუმენტით მონიშნეთ ფერადი მართკუთხედები და გააქტიურეთ ინსტრუმენტი **Filter / Distort / Glass** ;  
მიიღებთ დაშლილი მართკუთხედების ეფექტს;

საწყისი ფოტო



შედეგი



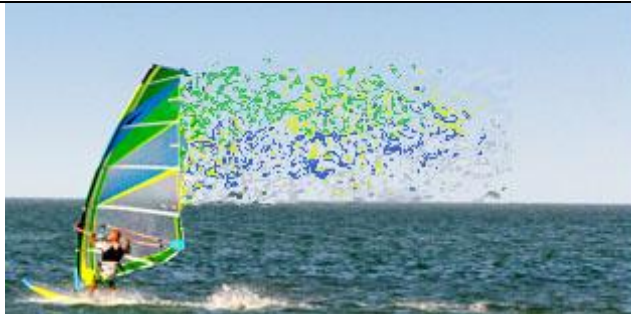
## დავალება 15

შეასრულეთ დავალება Gradient Tool, Paint Bucket Tool და Magic Tool ინსტრუმენტების გამოყენებით.

საწყისი ფოტო



შედეგი



## ლიტერატურა

1. <https://photoshop-master.ru/lessons/practice/>
2. [http://esate.ru/article/cg/osnovnye\\_vidy\\_kompyuternoy\\_grafiki/](http://esate.ru/article/cg/osnovnye_vidy_kompyuternoy_grafiki/)
3. Дональд Херн, М. Паулин Бейкер. Компьютерная графика и стандарт OpenGL = Computer Graphics with OpenGL. — 3-е изд. — М.: «Вильямс», 2005. — С. 1168. — ISBN [5-8459-0772-1](#).
4. Эдвард Энджел. Интерактивная компьютерная графика. Вводный курс на базе OpenGL = Interactive Computer Graphics. A Top-Down Approach with Open GL. — 2-е изд. — М.: «Вильямс», 2001. — С. 592. — ISBN [5-8459-0209-6](#).
5. Сергеев Александр Петрович, Кущенко Сергей Владимирович. Основы компьютерной графики. Adobe Photoshop и CorelDRAW - два в одном. Самоучитель. — М.: «Диалектика», 2006. — С. 544. — ISBN [5-8459-1094-3](#).
6. [What is Computer Graphics?](#), Cornell University Program of Computer Graphics. Last updated 04/15/98. Accessed November 17, 2009.
7. <https://web.archive.org/web/20070405172134/http://accad.osu.edu:80/~waynec/history/lessons.html>



# სარჩევი

კომპიუტერული გრაფიკის ძირითადი ცნებები.....	3
კოორდინატთა სისტემები, გრაფიკული გამოსახულებების გარდაქმნები...6	
ორგანზომილებიანი მატრიცული გარდაქმნები.....	9
ერთგვაროვანი კოორდინატები და ორგანზომილებიანი გარდაქმნების მატრიცული წარმოდგენები.....	12
სამგანზომილებიანი მატრიცული გარდაქმნები.....	17
ფერთა მოდელები.....	20
რასტრული ანუ პიქსელური გრაფიკის ცნება.....	28
რეზოლუციის სახეები.....	31
ფერთა პალიტრების შესახებ.....	36
პრაქტიკული სავარჯიშოები.....	39

ISBN 978-9941-0-9599-3



9 789941 095993



სამბროვლის მაცხოვრებელი