



Digitalna obrada slike

Lekcija III

Histogram

Operacije nad tačkom

Geometrijske transformacije

Interpolacija

HDR slika i Bayerov mozaik

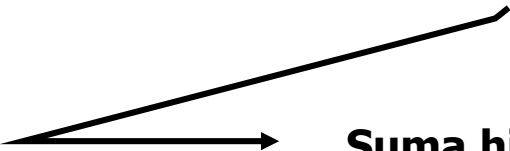


Histogram

- **Histogram** je jednostavna (ali veoma korisna) statistika slike.

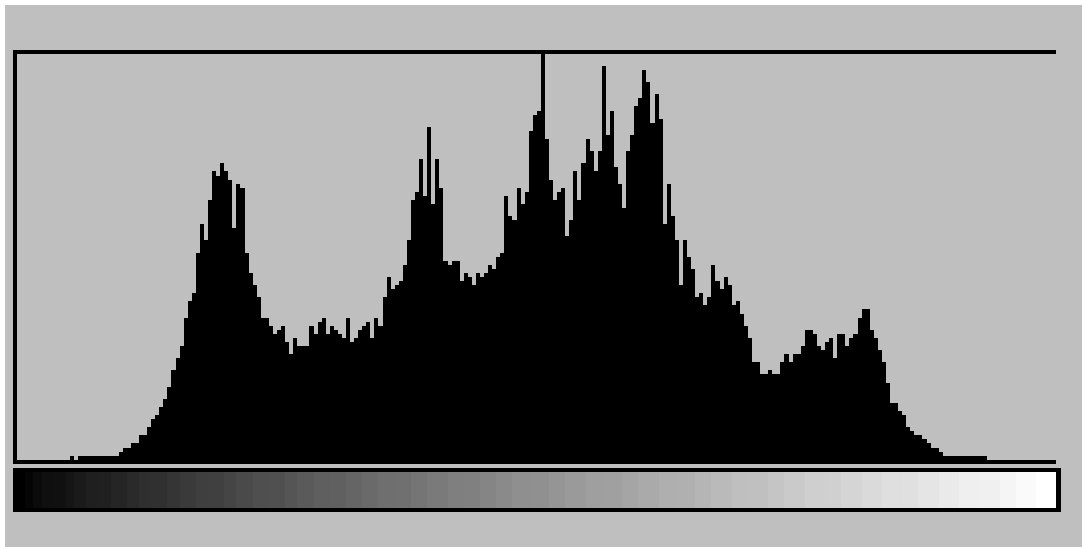
$H(X)$ = broj tačaka slike sa osvjetljajem X

$$\sum_{X=0}^{255} H(X) = M \times N$$



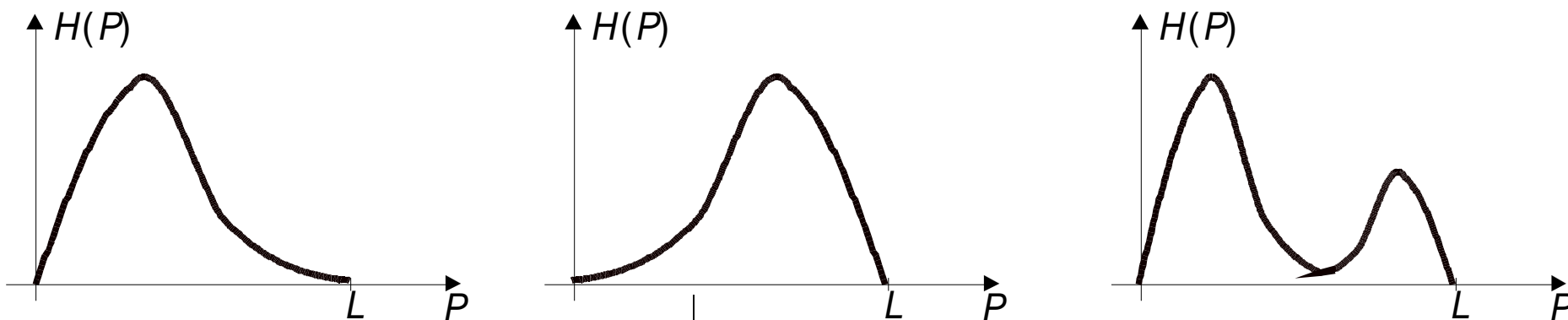
Suma histograma slike za sve osvjetljaje (ovdje je dato za sivoskaliranu sliku sa 8 bita po pikselu) jednaka je dimenziji slike u pikselima.

Primjer histograma



Histogram ima brojne primjene. Često se primjenjuje kod metoda koji su zasnovani na probabilističkim modelima koji zahtjevaju funkciju gustine raspodjele slike (ili barem procjenu ove veličine). **Kako se histogram može dovesti u vezu sa funkcijom gustine raspodjele?**

Histogram – Uobičajeni izgled



**unipolarni histogrami
tamne i svijetle slike**

**bipolarni histogram može da se
koristi za određivanje praga
prilikom dobijanja binarne slike
(kako?)**



Širenje histograma

- Optički senzori često sabiju sliku u neprirodno uzak nivo osvjetljaja.
- Stoga se softverski (a ne hardverski) vrše ispravke.
- Ispravke se vrše na osnovu histograma.
- Neka je slika sabijena u dio osvjetljaja $[A, B]$ (procjena vrijednosti A i B se obavlja na osnovu histograma).
- Ako pretpostavi da sliku želimo da proširimo na zonu $[0, 255]$ relacija je:

osvjetljaj slike sa
proširenim
histogramom

$$\longleftarrow f(X) = \frac{255}{B-A} X - \frac{255A}{B-A}$$

osvjetljaj
originalne slike

Širenje histograma - Primjer

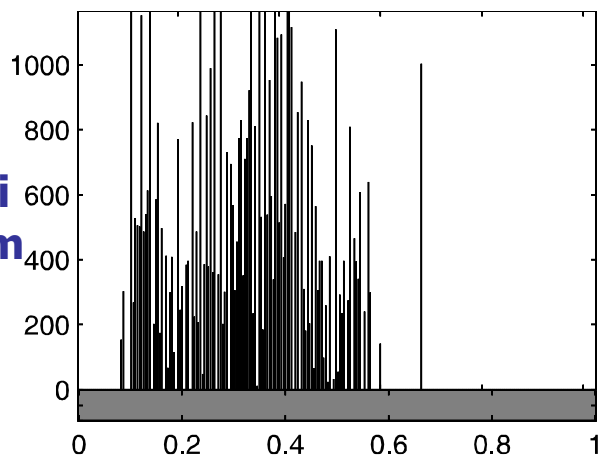
original



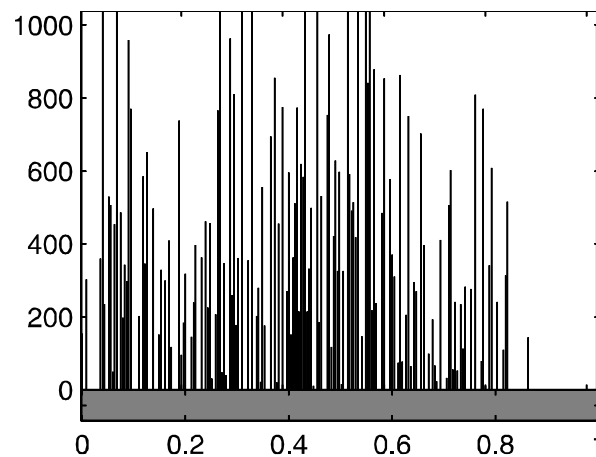
Nakon
širenja
histograma



originalni
histogram



histogram
poslije
širenja

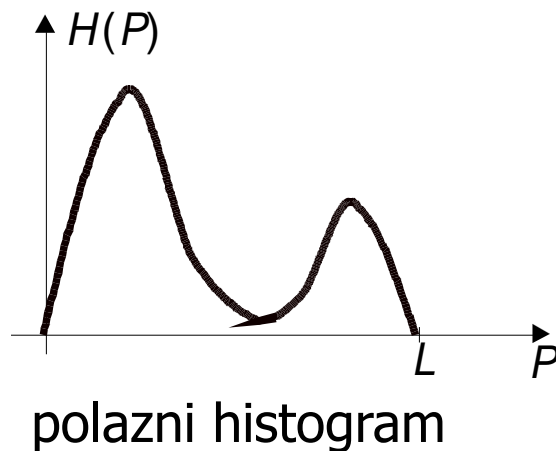




Ekvilizacija histograma

- Ekvilizacija (ujednačavanje) je česta operacija koja se primjenjuje kod histograma.
- Kod ekvilizirane slike histogram je približno ravan odnosno cilj je postići (ako je to moguće) da približno isti broj piksela uzima svaku moguću vrijednost osvjetljaja.
- Dakle, treba transformisati histogram proizvoljne slike da bude približno uniforman.
- Ekvilizirane slike imaju veoma dobar kontrast što je i razlog za primjenu ove operacije.

Ekvilizacija histograma



Ovo se može shvatiti kao sledeći problem: Imamo slučajnu promjenljivu sa funkcijom gustine raspodjele $f_x(x)$ (ovo ćemo procjenjivati na osnovu histograma ulazne slike). Traži se transformacija $y=g(x)$ koja daje funkciju gustine raspodjele $f_y(y)$. U našem slučaju to je proporcionalno ekviliziranom histogramu (odnosno konstanti).



Ekvilizacija histograma

- Iz teorije vjerovatnoće slijedi da je:

$$f_y(y) = \frac{f_x(x_1)}{|g'(x)|_{x=x_1}} + \frac{f_x(x_2)}{|g'(x)|_{x=x_2}} + \dots + \frac{f_x(x_N)}{|g'(x)|_{x=x_N}}$$

gdje su (x_1, x_2, \dots, x_N) realni korjeni jednačine $y=g(x)$.
Pretpostavimo da je rješenje jedinstveno što je moguće kada je funkcija $g(x)$ monotona u datoj oblasti.

$$f_y(y) = \frac{f_x(x_1)}{|g'(x)|_{x=x_1}}$$



Ekvilizacija histograma

- Kako je $f_y(y)$ konstanta to znači da je $|g'(x_1)|$ proporcionalno $f_x(x_1)$.
- Ako pretpostavimo dalje da je $g(x)$ uniformno rastuća dobijamo praktično da je $g'(x)=c f_x(x)$.



konstanta proporcionalnosti

- Odaberimo $c=1$ (izlazna slika će imati isti opseg osvjetljaja kao i ulazna) pa dobijamo:

$$g(x) = \int_{-\infty}^x f_x(x_1) dx_1$$

Integral funkcije gustine raspodjele je funkcija raspodjele koja je monotona u oblasti definisanosti.



Ekvilizacija histograma

- Pošto je slika diskretan a ne kontinualan podatak to je situacija ovdje malo komplikovanija jer nemamo kontinualnu funkciju gustine raspodjele već njenu diskretnu procjenu (histogram).
- MATLAB realizacija je ipak izuzetno jednostavna:

```
I=imread('pout.tif');  
a=imhist(I);  
g=cumsum(a)/sum(a);  
J=uint8(255*g(I));
```

→ učitavanje originalne slike

→ estimacija funkcije transformacije g

→ slika nakon transformacije

Ekvilizacija histograma - Primjer

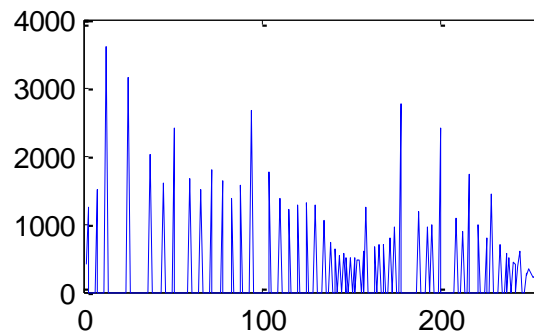
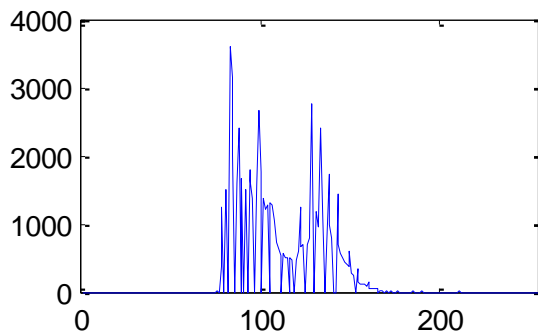


originalna slika

ekvilizirana slika
(znatno popravljen
kontrast)



**odgovarajući
histogrami**



Dobijena
raspodjela nije
idealno
uniformna zbog
diskretne
prirode slike



Podešavanje histograma

- Histogram se kod ekvilizacije pokušavao podesiti da produkuje uniformnu gustinu raspodjele.
- Histogram se može na sličan način koristiti da podesi sliku na bilo koju željenu funkciju gustine raspodjele.
- Procedura je potpuno ista kao u određivanju transformacije kod ekviliziranog histograma sve dok je dobijena funkcija $g(x)$ uniformna (opadajuća ili rastuća) što je kod ekvilizacije obezbjeđeno.
- U suprotnom, najbolji način rada je izdijeliti funkciju na monotone segmente i obaviti operaciju na njima.



Primjena histograma

- Mnoge metode zahtjevaju procjenu funkcije gustine raspodjele (histogram je tu nezamjenjiv).
- Povećanje kontrasta slike (ekvilizacija).
- Podešavanje histograma.
- Različite intervencije na bojama slike.
- Zapamtite da se histogram često primjenjuje lokalno. Npr. slikate izuzetno svijetao objekat koji zauzima mali dio slike. Ostatak slike će vam ostati neprirodno taman. Operacije podešavanja histograma i druge slične se mogu obaviti na dijelu slike (npr. na tamnom dijelu).
- Postoji mnogo situacija od koji ćemo se sa nekim upoznati gdje se histogram koristi lokalno (računa za djelove slike).



Negativ slike

- Postoji čitav niz operacija koje se obavljaju sa osvjetljajem slike. Neke od njih se primjenjuju na svaki piksel slike pojedinačno bez obzira na osvjetljaj susjednih piksela. Te operacije se nazivaju **operacijama nad tačkom**.
- Jedna od njih je i određivanja negativa slike (ili pozitiva ako imamo negativ slike).
- Operacije se provodi na različite načine u zavisnosti od formata slike.



Negativ slike

logička operacija negacija

- Binarna slika:

$$\text{Negativ}(n,m) = \sim \text{Original}(n,m)$$

broj bita korišćen za
memorisanje piksela slike

- Sivoskalirana slika

$$\text{Negativ}(n,m) = 2^k - 1 - \text{Original}(n,m)$$

- RGB slika

Operacija koja je opisana za sivoskaliranu sliku provodi se na svakom kanalu slike posebno



Odsjecanje boja

- Odsjecanje boja (**color clipping** – koje se ne smije pomiješati sa geometrijskim odsjecanjem koje ćemo naknadno upoznati) predstavlja operaciju kojom se ostavljaju u originalu boje u jednom dijelu osvjetljaja dok se ostale limitiraju na granice osvjetljaja:

$$b(i, j) = \begin{cases} c_{\max} & a(i, j) > c_{\max} \\ a(i, j) & c_{\max} \geq a(i, j) \geq c_{\min} \\ c_{\min} & a(i, j) < c_{\min} \end{cases}$$

Posvjetljivanje (potamnjenje) slike

- Postoji više strategija da se slika posvijetli.
- Npr. $f(n,m)=g(n,m)+r$ će posvijetliti sliku ako je $r>0$ a za $r<0$ potamniti.
- Drugi način je: $f(n,m)=g(n,m)\times r$ gdje se slika posvijetljava ako je $r>1$ odnosno potamnjuje ako je $0<r<1$.
- Prethodne dvije strategije izobličuju osvjetljaj slike i stvaraju neke loše efekte. Stoga se češće provodi sledeća procedura:

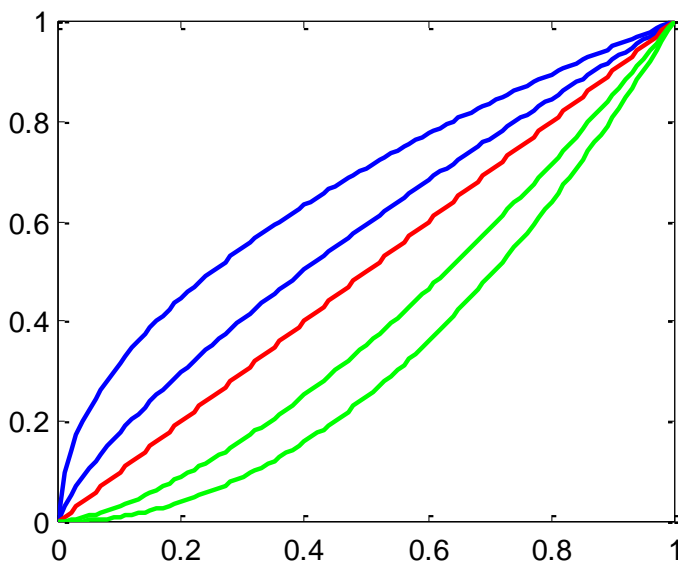
$$f(n,m) = [2^{k-1}] \{g(n,m) / [2^{k-1}]\}^\gamma$$

broj bita za zapis piksela

$\gamma > 1$ slika postaje tamnija $\gamma < 1$ slika postaje svetlija

Promjena osvjetljaja slike

- Funkcija transformacije histograma:



— posvjetljivanja

— potamnjivanje

Ovakav prikaz operacija nad tačkom je prihvaćen od većine softvera za obradu slike (Npr. Photoshopa).



Rezultat operacija nad tačkom

- Operacije nad tačkom kao i druge operacije koje se provode nad slikom ne moraju nužno da rezultuju cijelim brojevima na datom intervalu.
- Stoga se nakon operacija a prije zapisivanja slike u fajl ili prikaza na monitor mora izvršiti vraćanje u format slike.
- Koraci u ovoj proceduri su za sivoskaliranu sliku u 256 nivoa:
 - Zaokružiti ili odsjeći necjelobrojni dio (bolje je zaokružiti ali se odsjecanje lakše sprovodi);
 - Ako je dobijeni osvjetljaj manji od 0 postaviti ga na nula a ako je veći od 255 postaviti ga na 255.



Operacije sa tačkom - Zaključak

- Operacije nad tačkom, koje se najčešće obavljaju na osnovu uvida u histogram, izuzetno se često provode.
- Mi smo podrazumjevali da piksel odredišne slike zavisi samo od osvjetljaja istog piksela izvorne slike.
- Moguće je definisati i složenije funkcije gdje piksel zavisi od više drugih piksela i slično.
- Neke od ovih varijanti vam ostavljamo za vježbu.



Geometrijske transformacije

- Kod geometrijskih transformacija imamo slučaj da se piksel sa pozicije (x, y) pomjera na poziciju (x_1, y_1) . Ovdje je suština u transformaciji koordinata slike što se može zapisati kao:

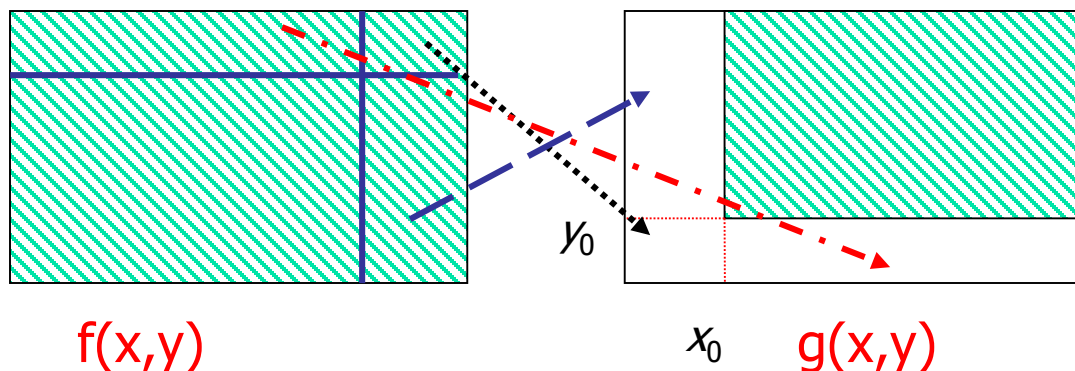
$$\mathbf{X}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = g(\mathbf{X}) = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Najjednostavnija od svih operacija je translacija kada se čitava slika pomjera za dati vektor (x_0, y_0) .

Translacija

$$\mathbf{X}_1 = \mathbf{X} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

$$g(x, y) = f(x - x_0, y - y_0)$$



U ovom slučaju su zadržane iste granice odredišne slike kao i polazne s time što je u dio koji je "nastao" pomjeranjem upisana bijela boja (može i neka druga boja). Ova strategija se naziva odsjecanjem. Takođe, jedna od strategija (koja zavisi od naših potreba je da se promjeni veličina slike odnosno da se poveća kako bi čitava slika stala.

Moguće je i ciklično pomjeranje da ono što nije stalo u sliku prebaci se na početak slike kako pokazuju strelice.



Odsjecanje

- Odsjecanje ili cropping je operacija kojom se dio originalne slike odsjeca i stvara nova slika manjih dimenzija od polazne.
- Npr. neka je data originalna slika $f(x,y)$ dimenzija (M,N) i neka se želi iz nje odsjeći zona: od (M_1,N_1) do (M_2,N_2) gdje je $0 < M_1 < M_2 < M$ i $0 < N_1 < N_2 < N$.

$$g(x - M_1 + 1, y - N_1 + 1) = f(x, y)$$

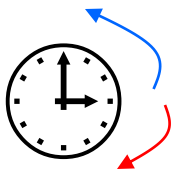
za $x \in [M_1, M_2]$ i $y \in [N_1, N_2]$. Kolike su dimenzije odredišne slike?

Rotacija

- Transformacija koordinata kod rotacije ima oblik:

$$\mathbf{X}' = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

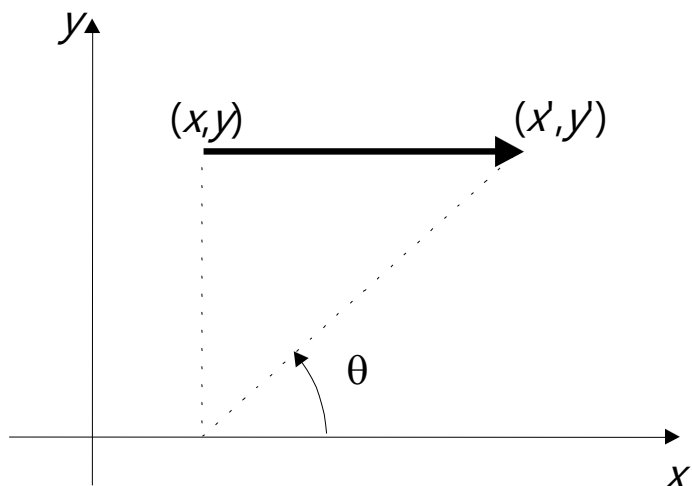
- Dobijena slika je: $g(x,y)=f(x\cos\alpha+y\sin\alpha,-x\sin\alpha+y\cos\alpha)$.
- Pretpostavljeno je da je rotacija koordinata obavljena oko koordinatnog početka što se rijetko radi kod digitalne slike. Kako bi izgledala rotacija oko proizvoljne tačke (x_0, y_0) ?
- Pozitivan smjer rotacije je u smjeru suprotnom kazaljki na satu.



pozitivan smjer suprotan kazaljki na satu

negativan smjer kazaljke na satu

Smicanje - distorzija



Ilustracija smicanja koje se po pravilu provodi duž jedne ose – koordinate.

Transformacija koordinata:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -\cot \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$g(x, y) = f(x - y \cot \theta, y)$$

Razradite smicanje paralelno sa proizvoljnom pravom linijom $y = ax + b$.



Skaliranje

- Transformacija koordinata kod skaliranja ima oblik:

$$\mathbf{X}' = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \mathbf{X}$$

Odrediti funkciju izlazne slike od ulazne slike.

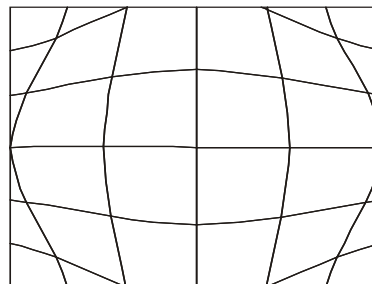
Da li se originalna slika povećava ili smanjuje u zavisnosti od parametara **a** i **b**?

Data je transformacija koja skalira sliku duž **x** i **y** ose da li je i kako moguće skalirati sliku duž drugih pravaca?

Da li se refleksije u odnosu na koordinatne ose ili koordinatni početak može predstaviti preko skaliranja?

Nelinearne transformacije

- Daleko od toga da su transformacije koje se primjenjuju na slici uvijek linearne.
- Baš suprotno one su češće nelinearne.
- Evo jedan prosti primjer:
 $g(x,y)=f(x+A\sin by, y+A\sin bx)$.
- Važan primjer ("gdje je sve počelo") je ispravljenja **fish-eye** zakrivljenosti.





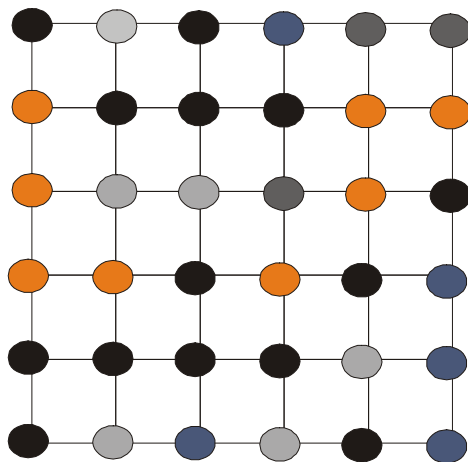
Fish-eye zakrivljenost

- Posledica je zakrivljenosti i konačnih (relativno malih) dimenzija sočiva što dovodi do toga da se objekti koji su udaljeni od centra slike prikazuju znatno umanjenim u odnosu na one u centru.
- Ponekad fotografi žele da simuliraju ili pojačavaju ovaj efekat i tada moraju da kupuju objektivne sa posebno zakrivljenim sočivima.
- Probajte sami da simulirate fish-eye transformaciju a zatim predložite postupak za njeno ispravljanje.

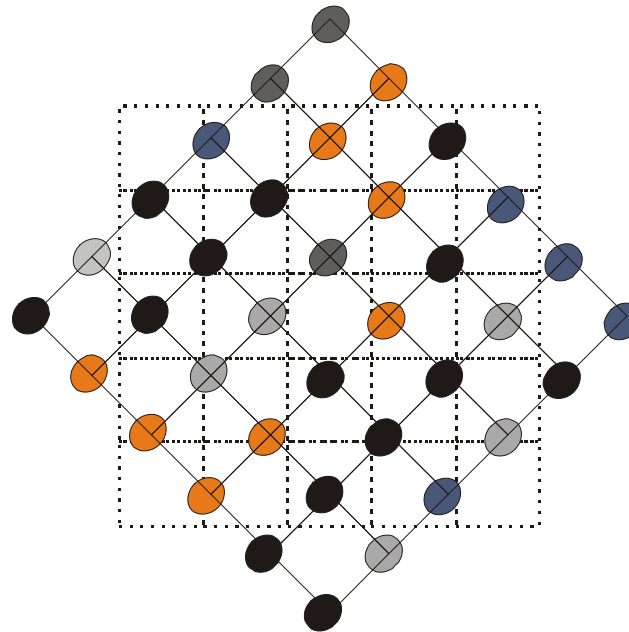


Geom. transf. - Problem

Originalna slika u čvorovima rešetke su pikseli.



**Slika rotirana za 45 stepeni.
Pikseli iz prethodne slike se
više ne nalaze u čvorovima
rešetke.**





Interpolacija - potreba

- Samo mali broj geometrijskih transformacija preslikava piksele originalne slike u čvorove mreže ciljne slike.
- Postavlja se pitanje kako odrediti vrijednosti u čvorovima rešetke za odredišnu sliku.
- Ove strategije se nazivaju interpolacijama i o njima su napisane knjige.
- Ovdje ćemo ukratko opisati nekoliko strategija.
- Za ostale metodologije pogledajte knjigu, Internet, a dodatna se materija može dobiti i od predmetnog nastavnika.

Najbliži susjedi

- Tehnika najbližih susjeda (**nearest neighbor**) je najjednostavnija.
- Za vrijednost čvora u rešetki se usvaja vrijednost najbližeg piksela transformisane slike.
- Kod najbližih susjeda međutim postoji znatan problem u kvalitetu.

**originalni
pravougaonik**



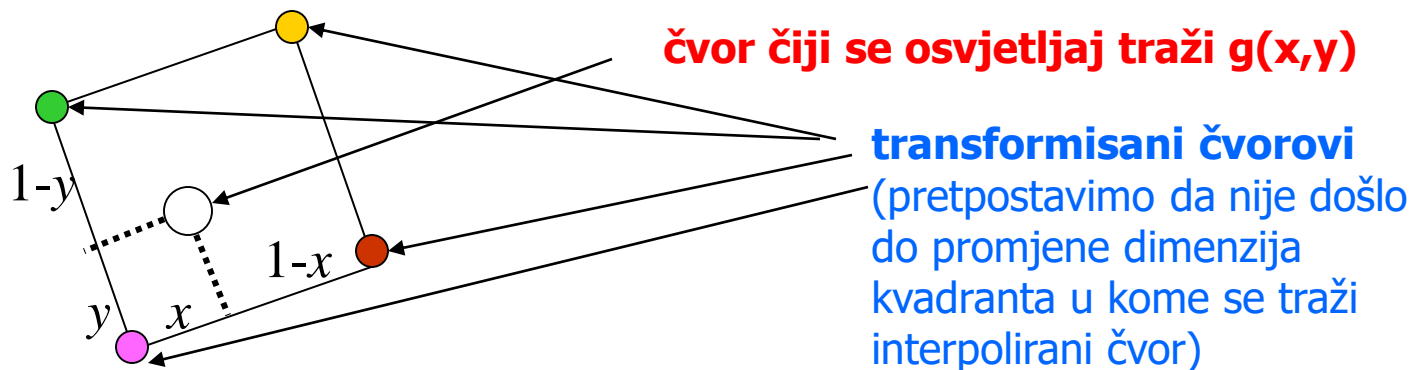
**nakon rotacije za 5 stepeni uz primjenu
interpolacije na najbliži susjed**



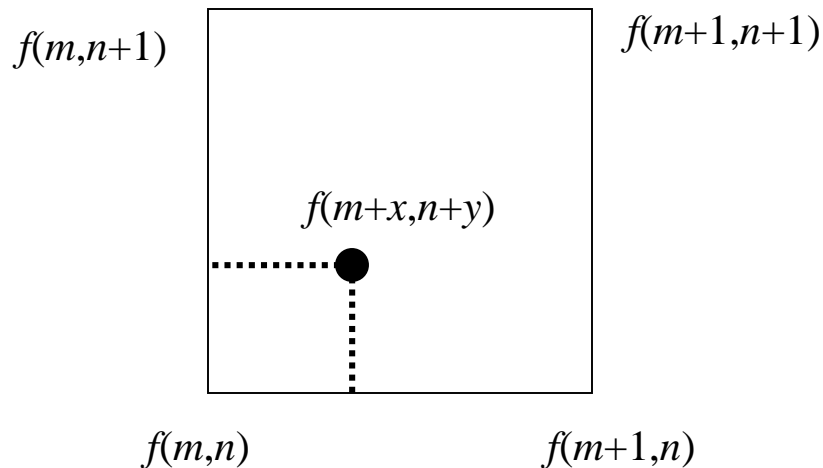
**Isprekidane i izlomljene stranice
ivice, i oštećenja malih detalja slike
su posljedica primjenjene
interpolacije.**

Bilinearna interpolacija

- Strategija bilinearne interpolacije je nešto bolja sa stanovišta kvaliteta a nešto sporija sa stanovišta brzine izračunavanja (ovo danas obično nije problem).
- Neka je čvor slike u susjedstvu četiri piksela transformisane slike.



Bilinearna interpolacija



Radi lakšeg računa vratimo se u zarotirani koordinatni sistem.

Bilinearna transformacija određuje vrijednost osvjetljava u tački $(m+x,n+y)$ kao:

$$f(m+x,n+y)=axy+bx+cy+d$$

konstante a , b , c i d treba odrediti



Bilinearna interpolacija

- Konstante se mogu odrediti iz graničnih uslova:

$$f(m,n)=a\times 0\times 0+b\times 0+c\times 0+d \Rightarrow d=f(m,n)$$

$$f(m+1,n)=a\times 1\times 0+b\times 1+c\times 0+d \Rightarrow b=f(m+1,n)-f(m,n)$$

$$f(m,n+1)=a\times 0\times 1+b\times 0+c\times 1+d \Rightarrow c=f(m,n+1)-f(m,n)$$

$$f(m+1,n+1)=a+b+c+d \Rightarrow$$

$$a=f(m+1,n+1)+f(m,n)-f(m+1,n)-f(m,n+1)$$

Posmatrajte sledeći primjer. Ne vrši se geometrijska transformacija već samo želite da promijenite broj piksela u slici (npr. umjesto $N\times M$ da dobijete $kN\times lN$ gdje su k i l cijeli brojevi veći od nule $k>1$ i $l>1$. Odrediti relaciju koja povezuje početnu i odredišnu sliku a koja je zasnovana na bilinearnoj transformaciji? Napominjem brojevi k i l ne moraju biti cijeli.

Ova operacije se naziva image resize.



Rotacija – MATLAB program

```
function b=rotacija_i_interpolacija(a, theta, x0, y0)  
%Realizacija je obavljena u okviru funkcije gdje je a polazna slika  
%theta je ugao za koji se vrši rotacija a x0 i y0 centar oko koga se rotacija vrši
```

```
[M, N] = size(a); %Veličina sivoskalirane slike
```

```
b = zeros(M, N); %Odredišna slika  
%pretpostavka je da je odredišna slika istih dimenzija kao i polazna  
%kako se vrši rotacija to ne mora biti slučaj a ono što preteče biće odsječeno
```

```
for xp = 1 : M  
  for yp = 1 : N %Prolazimo kroz sve piksele odredišne slike
```

```
    x = (xp - x0) * cos(theta) - (yp - y0) * sin(theta) + x0;
```

```
    y = (xp - x0) * sin(theta) + (yp - y0) * cos(theta) + y0;
```

```
%određujemo odakle potiče piksel koji se preslikava na poziciju (xp,yp)  
%odnosno gdje se on nalazio u polaznoj slici (inverzna transformacija)
```



Rotacija – MATLAB program

```
if ((x >= 1) & (x <= M) & (y >= 1) & (y <= N))  
%Da li je piksel u granicama polazne slike?
```

```
xd = floor(x); xg = ceil(x);  
yd = floor(y); yg = ceil(y);    %(xd,yd) donji lijevi ugao  
%pravougaonik za interpolaciju; (xg,yg) gornji desni ugao
```

```
D = double(a(xd, yd));  
B = double(a(xg, yd)) - double(a(xd, yd));  
C = double(a(xd, yg)) - double(a(xd, yd));  
A = double(a(xg, yg)) + double(a(xd, yd))...  
- double(a(xd, yg)) - double(a(xg, yd));  
%Određivanje koeficijenata
```

```
b(xp, yp) = A*(x-xd)*(y-yd)+B*(x-xd) +C* (y-yd)+D;  
%Vrijednost odredišne slike
```



Rotacija – MATLAB program

```
        end
    end
end
b = uint8(b);      %%%%Kraj programa (kraj if selekciji i dvije for petlje)
%%%%%%%%%%%%%%%%i na kraju vraćanje slike u odgovarajući format
```

Napisati program za distorziju.

Napisati program za rotaciju i najbliži susjed.

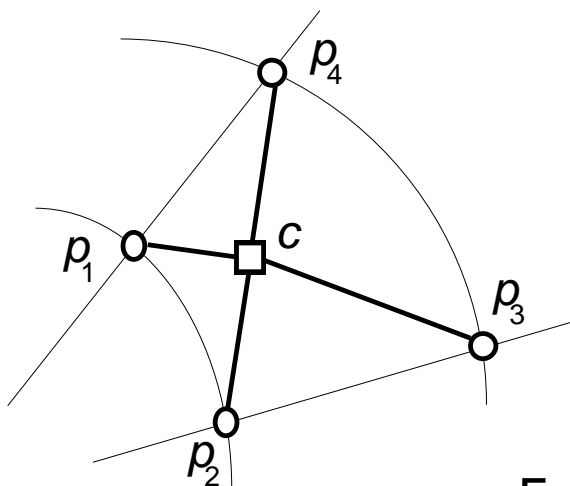
Izvršiti rotaciju za 5 stepeni pomoću najbližeg susjeda i to dva puta a zatim rotirati za -5 stepeni dva puta. Ovo odraditi i za bilinearnu interpolaciju. Uporedite rezultate.



Polarni u pravougaoni raster

- U medicini veliki broj slika nastaje aksijalnim snimanjem objekata pod različitim uglovima. Ovo se odnosi na različite tipove skenera.
- Dobijena slika je kružnog oblika sa rezultatima koje želimo da vizuelizujemo dobijenim duž prečnika kruga.
- Slične slike se dobijaju kod radara, sonara i nekih drugih uređaja.
- Ove slike imaju polarni raster (raspored piksela).
- Kako monitori imaju pravougaoni raster to treba izvršiti odgovarajuću interpolaciju.

Polarni u pravougaoni raster



p_1, p_2, p_3, p_4

pikseli polarnog rastera

c piksel pravougaonog rastera

Forma bilinearne interpolacije koja bi se mogla primjeniti ovdje:

$$f(c) = \frac{f(p_1)/p_1 + f(p_2)/p_2 + f(p_3)/p_3 + f(p_4)/p_4}{1/p_1 + 1/p_2 + 1/p_3 + 1/p_4}$$

p_i su udaljenosti između tačke p_i i c dok je $f()$ osvjetljaj u odgovarajućoj tački



Bikubična interpolacija

- Postoji više formi bikubične interpolacije kod koje se primjenjuju polinomi trećeg reda. Jedna forma je objašnjena u udžbeniku u okviru sekcije sa zadacima. Ovdje ćemo demonstrirati jednu nešto češću formu.
- Ponovo je polinom bikubičan trećeg reda

$$f(n+x, m+y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

- Međutim, sada se pretpostavlja da su pored vrijednosti u tačkama $f(n,m)$, $f(n+1,m)$, $f(n,m+1)$, $f(n+1,m+1)$ poznati i izvodi u ovim tačkama

$$\begin{aligned} f_x(n+x, m+y) &= \frac{\partial f(n+x, m+y)}{\partial x} & f_{xy}(n+x, m+y) &= \frac{\partial^2 f(n+x, m+y)}{\partial x \partial y} \\ f_y(n+x, m+y) &= \frac{\partial f(n+x, m+y)}{\partial y} \end{aligned}$$



Bikubična interpolacija

- Ovo nam daje 16 jednačina sa 16 nepoznatih:
 $f(0,0)=a_{00}$, $f(1,0)=a_{00}+a_{10}+a_{20}+a_{30}$, $f(0,1)=a_{00}+a_{01}+a_{02}+a_{03}$,
 $f(1,1)=a_{00}+\dots+a_{33}$, $f_x(0,0)=a_{10}$, $f_x(1,0)=a_{10}+2a_{20}+3a_{30}$,
 $f_x(0,1)=a_{10}+a_{11}+a_{12}+a_{13}$, $f_x(1,1)=\dots$
- Formirajte sami sistem jednačina i odredite rješenje.
- U slučaju da izvodi nisu poznati što je čest slučaj u praksi oni se mogu aproksimirati putem **konačnih razlika**. Pogledajte npr. odrednicu "**finite difference**" na wikipedia-i.
- Postoje načini putem tzv. **spline procedura** da se izbjegne ovakav način rada i predmetna procedura sa 16 jednačina i 16 nepoznatih.
- Potražite više podataka o spline procedurama kod slike.



Druge forme interpolacije

- Često postoji problem da senzori geometrijski deformišu sliku (npr. skeneri). Na sreću način na koji to rade je poznat proizvođaču. Stoga se može koristiti interpolacija pomoću mreže (grida).
- Procedura za interpolaciju je sledeća: skenira se pravougaona mreža i proizvođač pogleda gdje se piksel koji se nalazi u čvorovima mreže preslikao. Na osnovu toga kreira inverznu transformaciju koju softverski primjenjuje na svaku sliku koja se skenira.



Druge forme interpolacije

- Najčešće se primjenjuju polinomijalni algoritmi od kojih bi se možda najjednostavnije mogao primjeniti Lagranžev.
- Postoje dobro razvijeni algoritmi za interpolaciju koji pokušavaju da čuvaju neke karakteristike slike (npr. ivice).
- Danas se često koriste **spline** tehnike za interpolaciju (za poznavanje ovih tehnika treba znati nešto o waveletima).
- Interpolacija se može vršiti i pomoću Fourierove transformacije ali o tome kasnije.



HDR slika

- Digitalni senzori i dalje imaju uži dinamički opseg (mogućnost prikaza različitih nijansi u slici) u odnosu na analogne senzore i npr. film.
- Digitalnim sensorima nije jednostavno da podese vrijeme ekspozicije – osvjetljavanja da bi postigli najbolji mogući odnos između najsvjetlijih i najtamnijih nijansi koje se mogu prikazati u slici (obično mogu prikazati samo tamne ili svijetle nijanse).
- Moguće rješenje ovog problem je kreiranje digitalnih slika sa visokim dinamičkim rasponom (High Dinamic Range – **HDR**) tako što se naprave nekoliko digitalnih slika sa različitim ekspozicijama koje se zatim kombinuju u jednu cjelinu na osnovu podataka o ekspoziciji pojedinačnih slika.



HDR slika

- Savremene digitalne kamere snimaju podatke u EXIF formatima koji posjeduju podatke o samoj kameri uključujući trajanju ekspozicije.
- HDR slika zahtjeva slike sa najmanje tri ekspozicije: standardnu, tamniju (kraća ekspozicija) i svjetliju (duža ekspozicija).
- HDR se posebno razvija zbog velikog napretka u konstrukciji hardvera koje mogu da formiraju ovakve slike u realnom vremenu.
- Pretpostavimo da imamo P fotografija iste scene sa jedinom razlikom u pogledu vremena ekspozicije navedene slike. Označimo te slike kao $f_i(n,m)$, $i=1,\dots,P$. Neka su sve slike istih dimenzija $n=1,\dots,N$, $m=1,\dots,M$.



HDR slika

- Ovdje ćemo demonstrirati postupak za jedan kanal slike u boji (suštinski ovakav algoritam je jedino smislen za slike u boji) ali se ista procedura primjenjuje za sve kanale. Neka je vrijeme ekspozicije za svaku sliku poznato ili se može dobiti iz EXIF slike i neka iznosi e_i .
- Prvi korak u algoritmu je normalizacija vremena ekspozicije tako da suma normalizovanih vremena ekspozicije bude jednaka 1:

$$e'_i = \frac{e_i}{\sum_{i=1}^P e_i}$$

- Relativna ekspozicija se dobija kao: $w_i = 2^{e'_i}$



Algoritam za HDR sliku

- Sintetička slika koja je jednaka broju normalno osvjetljenih piksela slike:

$$p(n,m) = \sum_{i=1}^P 1[f_i(n,m) > t_{\min} \wedge f_i(n,m) < t_{\max}]$$

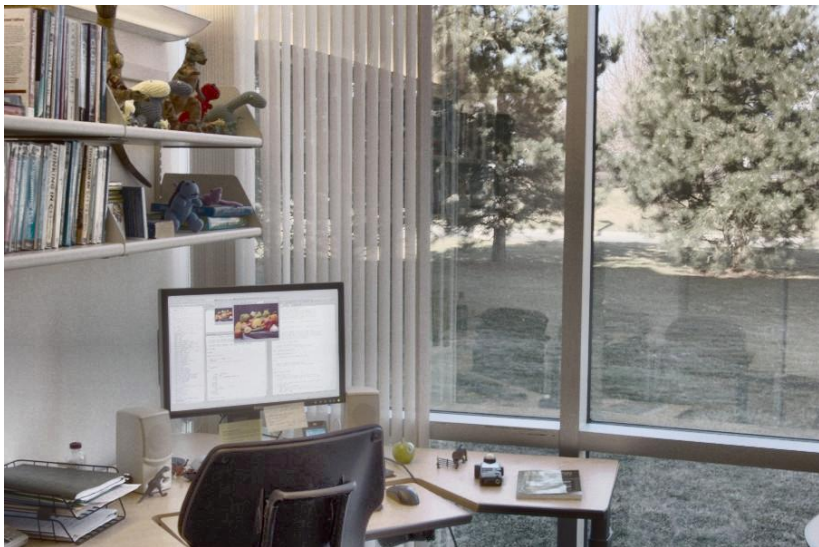
gdje je $1[]$ funkcija koja vraća 1 ako joj je argument tačan i 0 ako nije. Podrazumijevamo da je normalno osvjetljenje slike u granicama (t_{\min}, t_{\max}) a ako nemate bolju vrijednost često se usvaja $t_{\min}=5$ ili $t_{\max}=250$ za 8-bitnu sliku. Sada se HDR slika dobija kao:

$$h(n,m) = \frac{\sum_{i=1}^P f_i(n,m) 1[f_i(n,m) > t_{\min} \wedge f_i(n,m) < t_{\max}]}{w_i p(n,m)}$$

Ovdje podrazumijevamo da postoji makar jedan “normalno” osvjetljen piksel slike za svaku poziciju.

HDR slika - primjer

- Ako su svi pikseli tamni ili svijetli usvaja se tamna ili svijetla vrijednost a ako su svi loše osvijetljeni ali neki tamni a neki svijetli onda se po pravilu koristi interpolacija.



Dobra standardna slika



HDR verzija slike

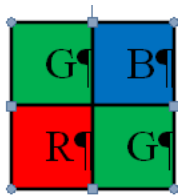


Bayerov mozaik model

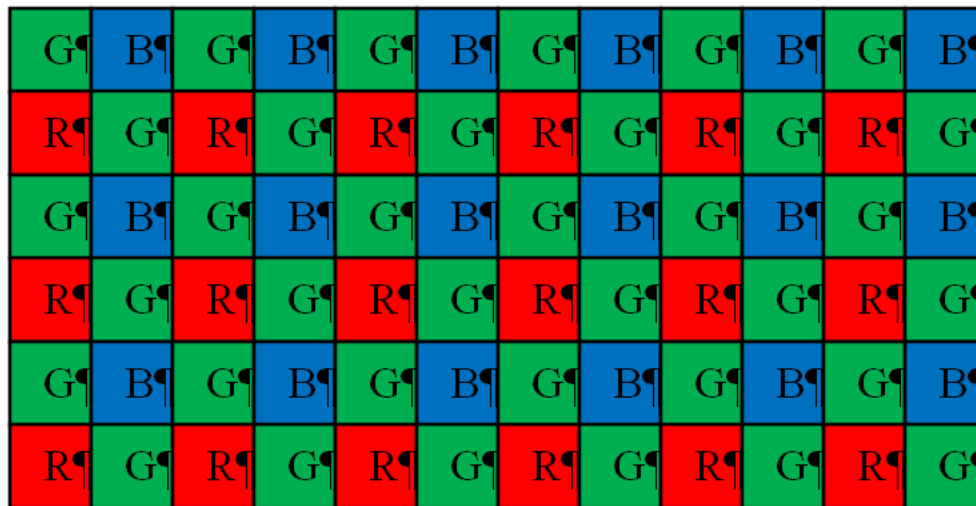
- Činjenica je da kamere obično raspolažu samo sa jednim senzorom za akviziciju odnosno sa jednom matricom.
- Kako kamera koja suštinski posjeduje jednu matricu može da posluži za snimanje slike u RGB kanalima?
- Najčešće se iznad CCD ili nekog drugog senzora postavlja optički filter koji filtrira pojedine boje tako da je svaka ćelija CCD senzora (odnosno piksel slike) osvjetljena sa jednom od osnovnih boja.
- Jedna od najpopularnijih je Bayerova – mozaik šema.

Bayerovala – mozaik šema

- Bayerov model i mozaik:



(a)



- Pojedini kanali se sada dobijaju interpolacijom.



Za samostalan rad

- Napišite sopstveni program za računanje histograma slike.
- Napišite program kojim se vrši širenje histograma tako da je donja i gornja granice koje se uzimaju u operacijama dopuštaju da 5% najsvjetlijih odnosno 5% najtamnijih piksela ostanu na granicama posmatranog područja osvjetljaja.
- Kako se određuje negativ slike koja je memorisana preko kolorne palete?
- Kako se može odrediti negativ slike za slučaj kolornih modela različitih od RGB?
- Napisati programe za računanje negativa slike.
- Formirati odredišnu sliku na osnovu izvorne tako što se u izvornoj slici formiraju šestougaona polja veličine 2-4-2 i svi piskeli odredišne slike u toj zoni postavljaju na srednju vrijednost piksela iz polazne slike.
- Realizovati sopstvene funkcije za sve varijante translacije.



Za samostalni rad

- Napisati transformaciju koordinata oko proizvoljne tačke za proizvoljni ugao.
- Napisati transformaciju koordinata koja obavlja distorziju paralelnu sa proizvoljnom pravom linijom $y=ax+b$.
- Odrediti funkciju izlazne slike od ulazne slike za sve oblike geometrijskih transformacija koje su definisane na predavanjima.
- Da li se originalna slika povećava ili smanjuje u zavisnosti od parametara a i b kod skaliranja?
- Data je transformacija koja skalira sliku duž x i y ose da li je i kako moguće skalirati sliku duž drugih pravaca?
- Da li se refleksije u odnosu na koordinatne ose ili koordinatni početak mogu predstaviti preko skaliranja?
- Realizovati sve naučene linearne transformacije koordinata.
- Realizovati transformaciju koordinata: $g(x,y)=f(x+A\sin by, y+A\sin xb)$. Eksperimentišite sa parametrima A i b .



Za samostalni rad

- Napisati program za image resize zasnovan na bilinearnoj transformaciji. Program treba da barata i sa necjelobrojnim vrijednostima faktora k i l kao i sa mogućnošću da k i l budu manji od 1.
- Napisati program za distorziju.
- Napisati program za rotaciju i najbliži susjed.
- Izvršiti rotaciju za 5 stepeni pomoću najbližeg susjeda i to dva puta a zatim rotirati za -5 stepeni dva puta. Ovo odraditi i za bilinearnu interpolaciju. Uporedite rezultate.
- Provjeriti da li je bilinearna intepolacija kod preslikavanja polarnog u pravougaoni raster suštinski ista kao i osnovna forma bilinearne interpolacije.



Projekat

- Omogućiti korisniku interaktivni izbor osvjetljaja i boja uključujući zadavanje boja pomoću krive transformacije koja je prikazana na slajdovima i omogućiti da korisnik zada više tačaka kroz koje kriva treba da prođe pa region između tačaka interpolirati preko Lagranževih množilaca.
- Napisati program kojim se vrši fish-eye transformacija kao i program koji sliku koja je fish-eye transformisana vraća približno u originalni oblik.



Projekat

- Podsjetite se Lagranževe formule a zatim je iskoristite da dobijete polinomijalnu interpolaciju pomoću mreže.
- Pročitajte resurse o interpolaciji do kojih možete doći putem Interneta i napišite o njima seminarski uz odgovarajuće simulacije.



Projekti

- Realizujte HDR sistem i proučite postojeći hardver za HDR sisteme. Uporedite vašu realizaciju sa onom iz MATLAB-a.
- Realizujte Bayerovu interpolaciju. Proučite alternativne modele mozaika koji se koriste za istu namjenu.