VEŽBA 8 – Histogram i poboljšanje kontrasta u slici

Potrebno predznanje

- Poznavanje programskog jezika C
- 2D signali
- RGB i YUV prostori boja i konverzija između ova dva prostora

Šta će biti naučeno tokom izrade vežbe

Nakon urađene vežbe naučićete na koji način se izračunava histogram slike. Saznaćete na koji način histogram slike opisuje nivo kontrasta u slici i na koji način se modifikacijom histograma slike može poboljšati njen kvalitet. Videćete kakve rezultate poboljšanja daje prosto skaliranje histograma na pun opseg vrednosti, a kakve primena algoritma za ujednačavanje histograma.

Naučićete na koji način je moguće vršiti kontrolu nivoa boja u slici bez promene ukupnog osvetljaja pojedinačnih piksela.

Motivacija

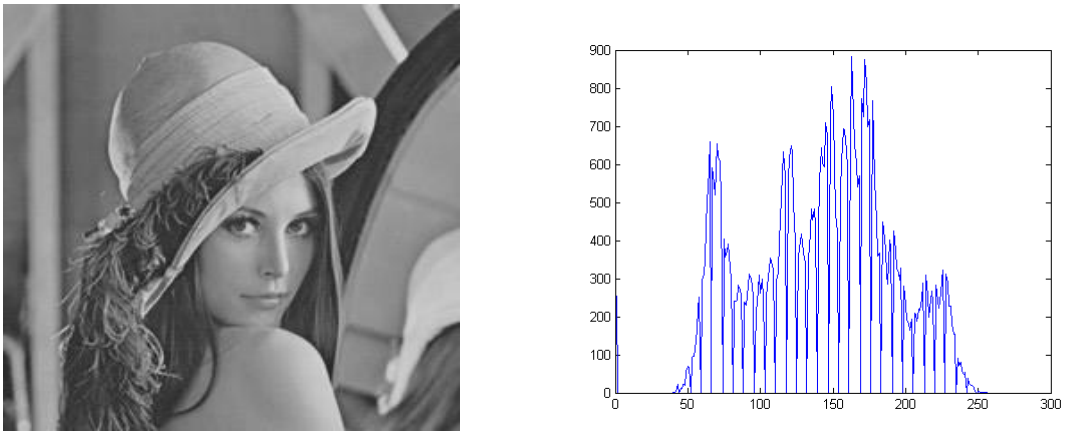
Potreba za poboljšanjem kvaliteta signala slike ili videa je svakodnevna. Neki od problema koji se često javljaju jestu neadekvatan kontrast unutar slike, nivo boja ili osvetljaj. Ovi problemi mogu se javiti prilikom snimanja sadržaja ili fotografisanja usled neadekvatnog osvetljenja ili fizičkih ograničenja uređaja korišćenih za snimanje. Takođe često se javljaju prilikom prikazivanja sadržaja na različitim ekranima, kada je potrebno prilagoditi kontrast, nivo boje i osvetljaja osobinama ekrana. U toku izrade ove vežbe upoznaćete se sa nekim od mogućih rešenja ovih problema.

Teorijske osnove

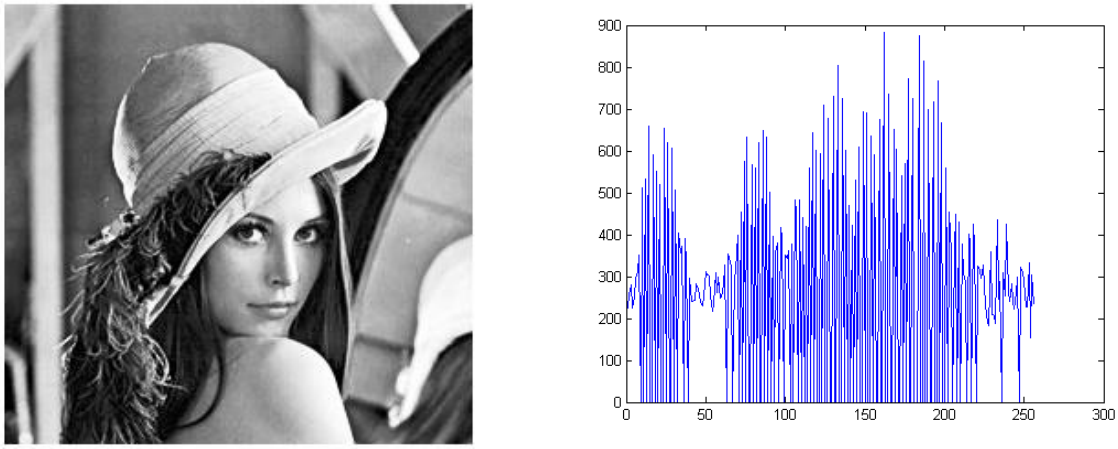
Histogram

Histogram slike predstavlja frekvenciju pojavljivanja različitih nivoa osvetljaja. Histogram se najčešće predstavlja kao grafikon na kome vrednosti X ose predstavljaju moguće vrednosti intenziteta osvetljaja u slici (kod YUV formata 0-255), dok je na Y osi predstavljen broj piksela u slici koji imaju tu vrednost osvetljaja. Histogram se koriste profesionalni fotografi prilikom podešavanja osvetljaja i kontrasta prilikom fotografisanja. Takođe histogram slike pomaže prilikom određivanja praga kod algoritama za detekciju objekata unutar slike.

Primer slike i njenog histograma dat je na slikama 1 i 2.



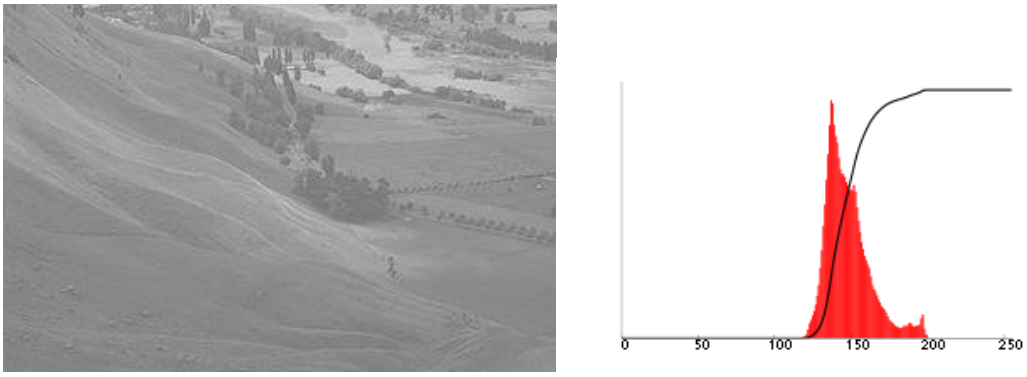
Slika 1 - Histogram slike sa niskim kontrastom



Slika 2 – Histogram slike sa poboljšanim kontrastom

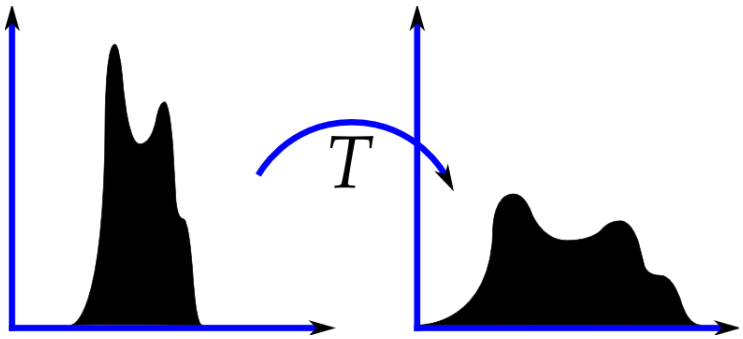
Ujednačavanje histograma

Ujednačavanje histograma je metod obrade slike kojim se podešava kontrast slike korišćenjem funkcije određivanja histograma. Ovaj metod se najčešće koristi za povećanje globalnog kontrasta u slici, naročito ukoliko slika sadrži piksele približnih vrednosti. Na slici 3 prikazan je primer slike sa lošim kontrastom i njen histogram.



Slika 3 – Slika sa lošim kontrastom i njen histogram

Postupak ujednačavanja histograma podrazumeva da oblasti nižeg kontrasta u slici dobiju veći kontrast tako što se rasprostiru vrednosti piksela koje se najčešće javljaju. Ujednačavanje histograma rezultuje pojavom da tamni pikseli u slici izgledaju još tamniji a svetli još svetliji.



Slika 3 – Transformacija u uniformni histogram

Treba napomenuti da se operacija ujednačavanja histograma ne vrši nad samim histogramom. Za razliku od transformacije slike u spektralni domen, gde je vršena transformacija slike, obrada u spektralnom domenu pa rekonstrukcija slike, u slučaju histograma vrši se izračunavanje histograma ulazne slike, a zatim se na osnovu tog histgrama i samih vrednosti ulazne slike vrši izračunavanje vrednosti izlazne slike koja će imati poboljšani histogram.

Algoritam za ujednačavanje histograma se primenjuje nad Y komponentom slike. Pre primene algoritma potrebno je izvršiti konverziju slike iz RGB u YUV prostor boja. Prvi korak u algoritmu jeste izračunavanje samog histograma slike. Potrebno je proći kroz čitavu sliku, odrediti broj ponavljanja za svaku moguću vrednost intenziteta osvetljaja:

$$hist(i) = n_i, \qquad 0 \leq i \leq L$$

Gde je *hist* histogram slike, *L* broj mogućih vrednosti piksela, a n_i broj piksela u slici čiji je intenzitet jednak *i.*

Ako posmatramo monohromatsku sliku, kod koje je vrednost svakog piksela predstavljena sa 8 bita, ta vrednost nalazi se u opsegu [0, 255]. Histogram ovakve slike se može predstaviti kao niz dužine 256 elemenata, gde svaki od elemenata predstavlja broj ponavljanja piksela sa vrednošću koja je jednaka indeksu datog elementa. U programskom jeziku C, za monohromatsku sliku predstavljenu matricom *image*, dimenzija `[xSize]*[ySize]`, histogram se može izračunati na sledeći način:

```
for(i = 0; i < xSize; i++)
    for(j = 0; j < ySize; j++)
        histogram[input[j * xSize + i]]++;
```

Pre samog računanja neophodno je sve elemente niza *histogram* inicijalizovati na 0;

Nakon toga potrebno je izračunati vrednost kumulativne funkcije raspodele za zadati histogram.

$$cdf(i) = \sum_{j=0}^{i} hist(j), \;\; 0 \leq i \leq L$$

Gde je *cdf* kumulativna funkcija raspodele.

U prgoramskom jeziku C, kumulativna funkcija raspodele može se predstaviti na isti način kao i histogram (niz od 256 elemenata). Računanje funkcije raspodele na osnovu izračunatog histograma prikazano je na narednom primeru. Promenljiva *sum* predstavlja privremenu promenljivu čija je inicijalna vrednost 0.

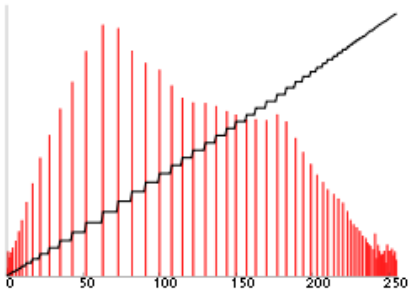
```
for(i = 0; i < 256; i++)
{
    sum += histogram[i];
    cdf[i] = sum;
}
```

Kada je izračunata kumulativna funkcija raspodele, potrebno je proći kroz čitavu ulaznu sliku i za svaki piksel izračunati vrednost izlaznog piksela koristeći njegovu vrednost i vrednost funkcije rapspodele na sledeći način:

$$y = h(x)$$

$$h(k) = round\left(\frac{cdf(k) - cdf_{min}}{(M \cdot N) - cdf_{min}} \cdot (L-1)\right)$$

Gde *y* predstavlja vrednost izlaznog piksela, *x* vrednost ulaznog piksela sa istim koordinatama kao *y*, M i N dimenzije slike a cdf_{min} minimalnu vrednost funkcije *cdf*(ne računajući nule). Na slici 4 prikazane su slika sa ujednačeni histogramom (levom) koristeći opisani algoritam i graf koji sadrži njen histogram (crvenom bojom) i kumulativnu funkciju raspodele (crnom bojom).



Slika 4 – Slika posle poboljšanja kontrasta.

Primer ujednačavanja histograma

U datom preomeru matrica *X* predstavlja ulaznu sliku, *hist* njen histogram, *cdf* kumulativnu funkciju raspodele dok je *Y* izlazna slika sa poboljšanim histogramom. U svim matricama zelenom bojom je obeležen piksel za koji je dat primer računanja izlazne vrednosti.

X =

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

hist =

i	n_i	*I*	n_i	*i*	n_i	*i*	n_i	*i*	n_i
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	183	1	109	1		

cdf =

i	n_i	*I*	n_i	*i*	n_i	*i*	n_i	*i*	n_i
52	1	64	19	72	40	85	51	113	60
55	4	65	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	183	49	109	59		

$$h(k) = round\left(\frac{cdf(k) - 1}{(8 \cdot 8) - 1} \cdot 255\right)$$

$$h(78) = round\left(\frac{46 - 1}{63} \cdot 255\right) = 182$$

Y =

0	12	53	93	146	53	73	166
65	32	12	215	235	202	130	158
57	32	117	239	251	227	93	166
65	20	154	243	255	237	146	130
97	53	117	227	247	210	117	146
190	85	36	146	178	117	20	170
202	154	73	32	12	53	85	194
206	190	130	117	85	174	182	219

Podešavanje nivoa boje u slici

Jedan od načina za poboljšanje kvaliteta slike jeste promena zasićenja boja u slici (eng. *Colour Saturation*). Ova operacija podrazumeva promenu nivoa boje u svakom od piksela slike bez promene ukupnog osvetljaja tog piksela.

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} (1-s)\cdot 0.299 + s & (1-s)\cdot 0.587 & (1-s)\cdot 0.114 \\ (1-s)\cdot 0.299 & (1-s)\cdot 0.587 + s & (1-s)\cdot 0.114 \\ (1-s)\cdot 0.299 & (1-s)\cdot 0.587 & (1-s)\cdot 0.114 + s \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Parametar *S* predstavlja nivo zasićenja. Njegova vrednost nalazi se u opsegu [0.0, 2.0]. Vrednost 0 odgovara slici bez boja (monohromatska slika), 1.0 originalnim vrednostima boja a 2.0 maksimalnoj saturaciji.

Konstante u datoj jednačini jednake su konstantama koje se koriste prilikom izračunavanja Y komponente prilikom konverzije iz RGB u YUV prostor boja. Kontrolu saturacije je moguće izvršiti prateći naredne korake:

1. Izvršiti konverziju iz RGB u YUV prostor boja
2. Za svaki piksel izlazne slike izračunati vrednost sa:

$$R_{out} = R \cdot s + Y \cdot (1 - s)$$

$$G_{out} = G \cdot s + Y \cdot (1 - s)$$

$$B_{out} = B \cdot s + Y \cdot (1 - s)$$

U nastavku sledi primer realizacije algoritma za modifikaciju zasićenja boja koristeći programski jezik C. U funkciji je prikazano računanje izlazne vrednosti samo za R komponentu (crvena boja) RGB slike.

```
void modifySaturation(const uchar inputRGB[], const uchar inputY[], int xSize, int ySize,
uchar outputRGB[], double S)
{
    for(int j = 0; j < ySize; j++ )
    {
        for(int i = 0; i < xSize; i++ )
        {
            out[j*xSize*3 + i*3] = inRGB[j*xSize*3 + i*3]*S + inY[j*xSize+i] * (1-S);
            out[j*xSize*3 + i*3+1] = inRGB[j*xSize*3 + i*3+1]*S + inY[j*xSize+i] * (1-S);
            out[j*xSize*3 + i*3+2] = inRGB[j*xSize*3 + i*3+2]*S + inY[j*xSize+i] * (1-S);
        }
    }
}
```

Zadaci

Zadatak 1:

Realizovati funkciju:

- `void simpleContrastImprovement (uchar input[], int xSize, int ySize, uchar output[])`

Funkcija vrši sakliranje vrednosti piksela na pun opseg 0-255. Potrebno je pronaći minimalnu vrednost piksela *min*, maksimalnu vrednost piksela *max* i zatim za svaki ulazni piksel izračunati vrednost izlaznog piksela po formuli:

$$Y = \frac{(X - min) * 255}{\max - min}$$

Zadatak 2:

Realizovati funkciju:

- `void calculateHistogram(const uchar input[], int xSize, int ySize, int histogram[])`

Funkcija vrši određivanje histograma slike kao što je to opisano u teorijskim osnovama. Ulazna slika data je sa parametrom *input*, dimenzije slike sa *xSize* i *ySize*. Izračunatu vrednost histograma smestiti u niz *histogram*.

Zadatak 3:

Realizovati funkciju:

- `void equalizeHistogram(const uchar input[], int xSize, int ySize, uchar output[])`

Funkcija treba da vrši poboljšanje kontrasta u slici korišćenjem algoritma za ujednačavanje histograma slike datog u okviru teorijskih osnova. Ulazna slika data je sa parametrom *input*, dimenzije slike sa *xSize* i

ySize, a izlazna slika sa *output*. Uporediti izgled obrađenih slika i njihovih histograma koristeći obrade iz zadatka 1 i zadatka 3.

Zadatak 4:

Realizovati funkciju:

- `void modifySaturation(const uchar inputRGB[], const uchar inputY[], int xSize, int ySize, uchar outputRGB[], double S);`

U okviru ove funkcije potrebno je realizovati podešavanje nivoa zasićenja boje u slici. Kao ulazni parametri su dati ulazna slika u RGB formatu, Y komponenta ulazne slike u YUV formatu, dimenzije slike i memorijski niz za smeštanje izlazne slike u RGB formatu. Parametar *S* predstavlja nivo zasićenja.