

Multimedije

Uvodno predavanje

Igor Jovančević mail: igorjovan@gmail.com

Danas...

- Informacije o organizaciji kursa
- Informacije o polaganju ispita
- Informacije o sadržaju kursa

- Uvodno predavanje
- Digitalna 2D slika

Termin predavanja i vježbi

Ponedjeljak 16h

Par pravila

- Termin za konsultacije
 - Petkom od 10h do 12h (kancelarija preko puta sale 210)
 - Uz najavu makar dan prije, mail-om
- Mailu koji se tiče svih **mora** da prethodi vaš dogovor
- Slanje tuđeg rada je strogo zabranjeno i kažnjivo za obje strane

Bodovi za ocjenu

- 40%
 - prisustvo i aktivno učešće na vježbama koje će vas uz malo dodatnog rada kod kuće dovesti do projekta koji ćete braniti
 - Vodite evidenciju svega što radite na vježbama, rezimirajte zaključke i šta ste naučili kroz izvještaj koji šaljete (Google Drive) poslije svakih vježbi
- 30%
 - seminarski rad i njegova odbrana
- 30%
 - završni teorijski ispit



Projekat/Vježbe

- Grupni rad
- Na kraju semestra se predaje:
 1. **JEDAN** pdf dokument (Izvještaj)
 2. **Spakovan** folder sa programskim kodom
 3. Slajdovi koji će biti prezentovani pred profesorom
- Ako ste koristili ideju/kod/koncept, iz članka/knjige/naučnog rada/linka/git repozitorijuma, referencirajte ga (pomenite ga u reportu i navedite punu referencu na kraju)
- Primjer reporta

Fusce venenatis ornare dui nec maximus. Sed egestas elit eget libero dignissim, ut vestibulum nisi sollicitudin. Nam aliquam in turpis sit amet pharetra. Vestibulum eu interdum nunc. Integer ac diam quis erat tincidunt ullamcorper. Curabitur pellentesque ullamcorper leo et pellentesque. Proin vel ante mi.

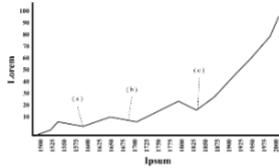


Fig. 1. Tempor nec feugiat nisl pretium fusce id velit. (a) Sed egestas egestas fringilla phasellus. (b) Turpis tincidunt id aliquet risus feugiat in. (c) Dolor morbi non arcu risus quis varius.

Vivamus mi quam, dapibus in magna non, interdum fringilla orci. Maecenas dapibus augue faucibus ligula gravida, eu tincidunt neque placerat. Curabitur mi est, viverra vel aliquet ac, tristique vel nisl. Interdum et malesuada fames ac ante ipsum primis in faucibus. In nibh dui, molestie a egestas at, condimentum non lectus. Curabitur vestibulum venenatis mauris, lobortis mattis libero ullamcorper a. Nullam nec vestibulum mauris, ut placerat urna.

Seminarski rad

- Grupni rad
- Na kraju semestra se predaje:
 1. **JEDAN** pdf dokument (Izvještaj)
 2. **Spakovan** folder sa programskim kodom (ako ga ima)
 3. Slajdovi koji će biti prezentovani pred kolegama
- Ako ste koristili ideju/kod/koncept, iz članka/knjige/naučnog rada/linka/git repozitorijuma, referencirajte ga (pomenite ga u reportu i navedite punu referencu na kraju)
- Primjer reporta

Dvije odbrane na kraju semestra

- Odbrana projekta
 1. Poslati izvještaj + kod + slajdove
 2. Prezentacija pred profesorom (samo jedna grupa prisutna) u zakazanom terminu (svaki član grupe prezentuje makar jedan dio slajdova)
- Odbrana seminarskog rada
 1. Poslati izvještaj + kod (ako ga ima) + slajdove
 2. javna prezentacija (15min + 5min) u zakazanom terminu (svaki član grupe prezentuje makar jedan dio slajdova)

Izvori ilustracija

- *Digital Image Processing 2nd Edition (Rafael Gonzales, Richard Woods)*
- *Slajdovi profesora Đurovića od prethodnih godina*
- *Ostali izvori će biti navedeni na slajdovima*

Definicija multimedija

- Bilo koji sadržaj koji predstavlja kombinaciju svih ili nekih od sljedećih formi informacije:
 - 1. Slika**
 - 2. Video**
 - 3. Audio**

Definicija multimedija

- Bilo koji sadržaj koji predstavlja kombinaciju svih ili nekih od sljedećih formi informacije:
 1. **Slika**
 2. **Video**
 3. **Audio**
 4. **3D**

Definicija multimedija

- Bilo koji sadržaj koji predstavlja kombinaciju svih ili nekih od sljedećih formi informacije:
 1. **Slika**
 2. **Video**
 3. **Audio**
 4. **3D**
 5. **Animacije**
 6. **Ultrazvuk**
 7. **X-ray zraci**

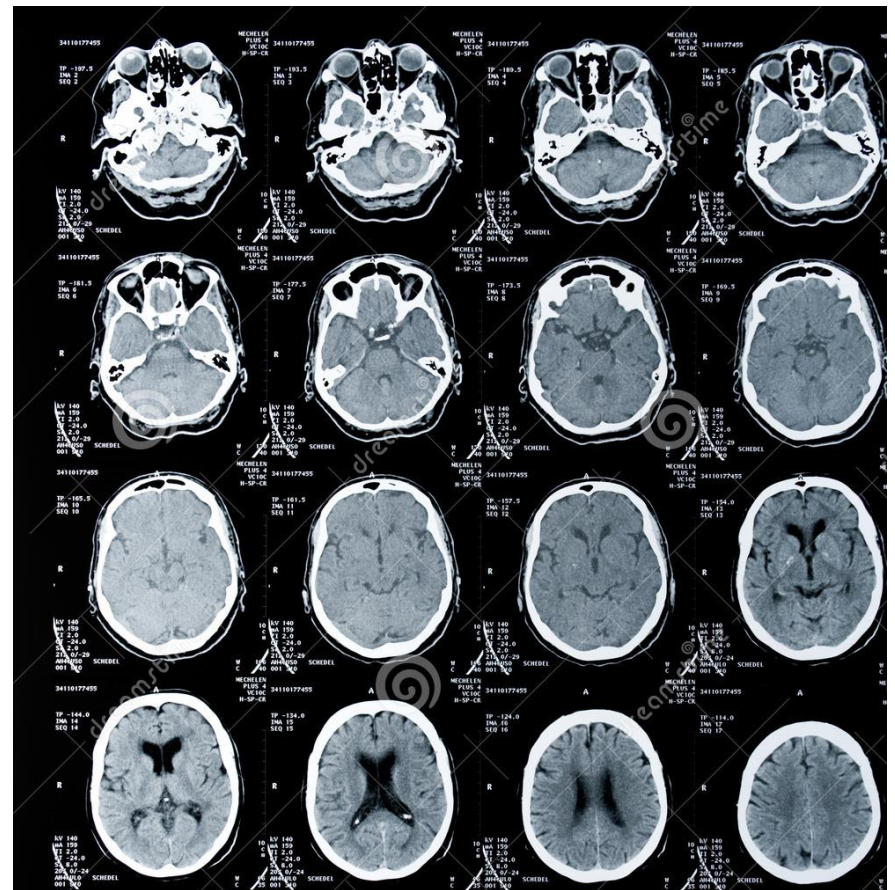
Neke od primjena

- Medicina



Neke od primjena

- Medicina



Neke od primjena

- Obrazovanje



Neke od primjena

- Virtuelna i proširena stvarnost (VR i AR) i sve njihove primjene (npr virtuelne obuke u industriji ili medicini, treninzi vojnika, virtuelne posjete muzejima, industrijska inspekcija)



Neke od primjena

- Industrijska inspekcija



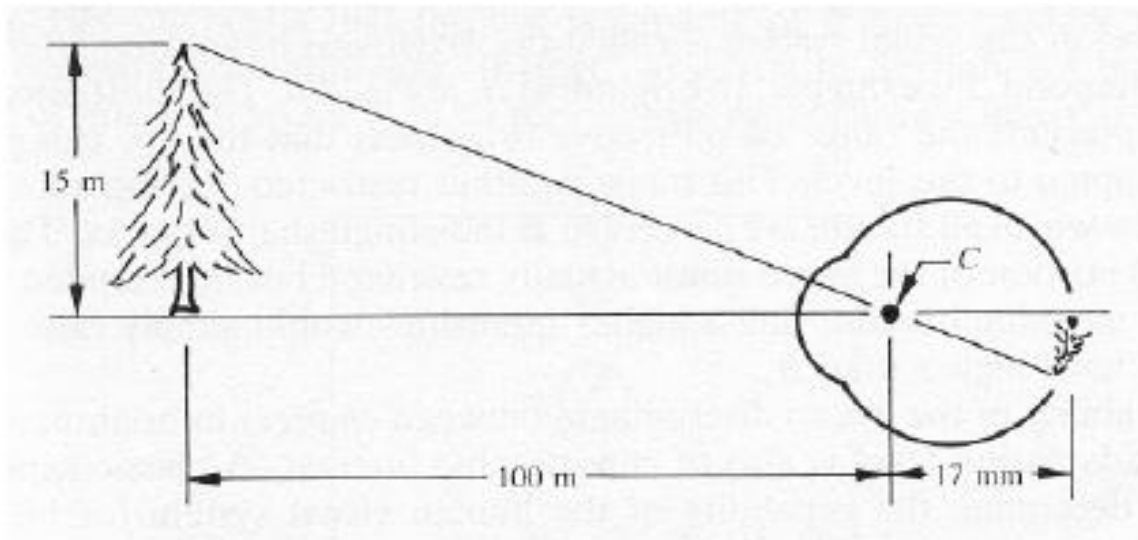
Struktura kursa

- 2D digitalna slika
- 3D podaci: oblaci tačkica (point cloud), 3D mreže (meshes)
- Video
- Audio signali
- Osnove digitalne obrade signala i Fourier-ove (Fourijeove) analize

Digitalna slika

Kako vidimo?

- Sve što vidimo je proizvod reflektovane svjetlosti sa objekata koje gledamo.
- Ljudski vidni senzor je oko



$$\frac{15}{100} = \frac{x}{17}$$

$$x = 2.55 \text{ mm}$$

Digitalna 2D slika

- Akvizicija (ili nastanak) digitalne slike pomoću senzora

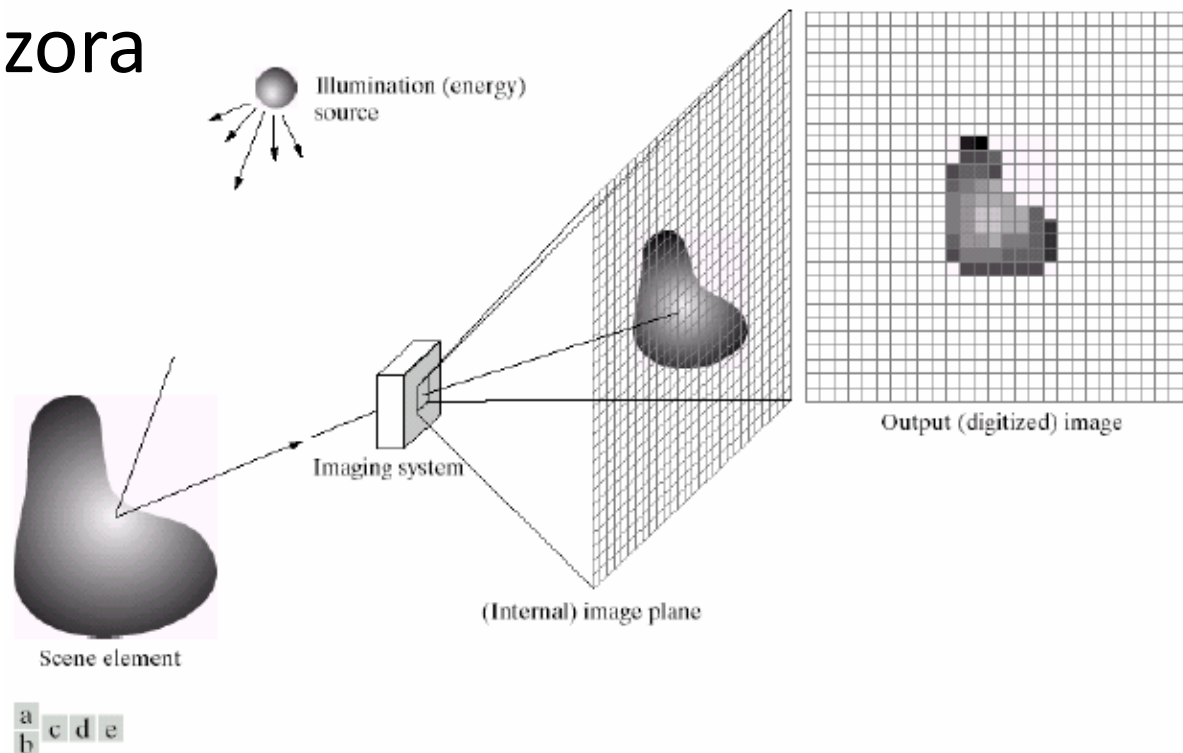
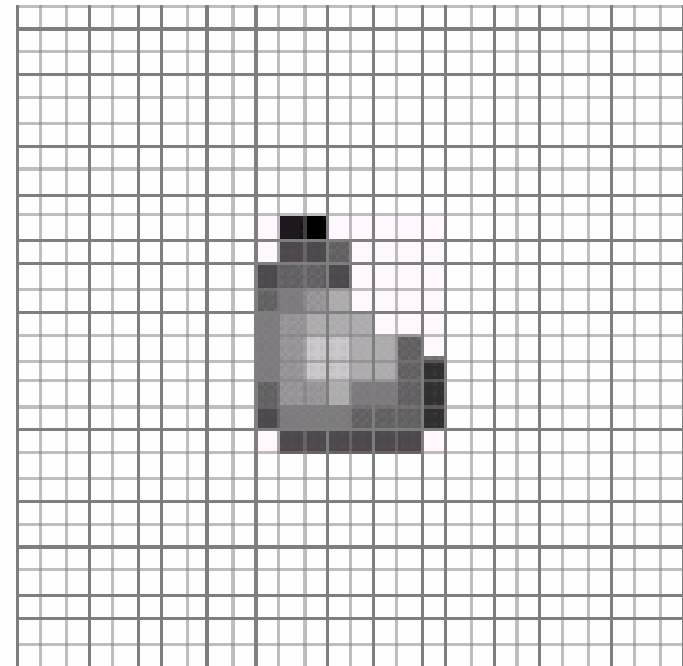


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Digitalna 2D slika

- Za računar i njegovu memoriju, slika je matrica dimenzija $w \times h$.
(w – širina, h – visina)
- Svaki elemenat matrice odgovara jednom pikselu (elementarnoj ekranskoj tački)
- Vrijednost elementa matrice je osvjetljenost (brightness ili intenzitet) piksela.



Output (digitized) image

Digitalna 2D slika

- Digitalna slika se može razumjeti i kao funkcija od 2 promjenjive

$$f = \begin{bmatrix} f(1, 1) & f(1, 2) & \dots & f(1, N) \\ f(2, 1) & f(2, 2) & \dots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \dots & f(M, N) \end{bmatrix}$$

- $f(x,y) = I$; $0 < f(x,y) < \infty$

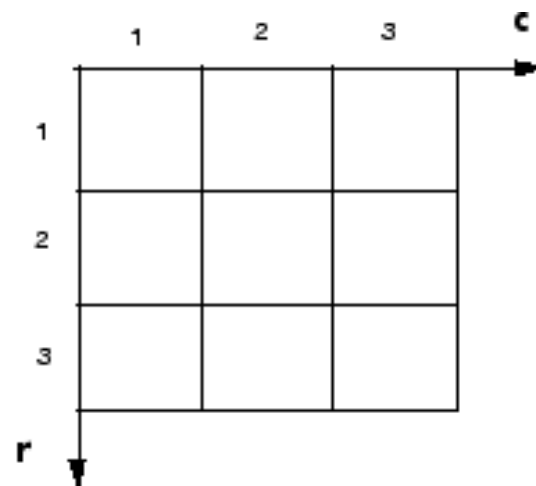
- Mi ćemo raditi sa opsegom:

$$0 < f(x,y) < 255 \text{ ili } 0 < f(x,y) < 1$$

x, y, I - diskretne konačne veličine

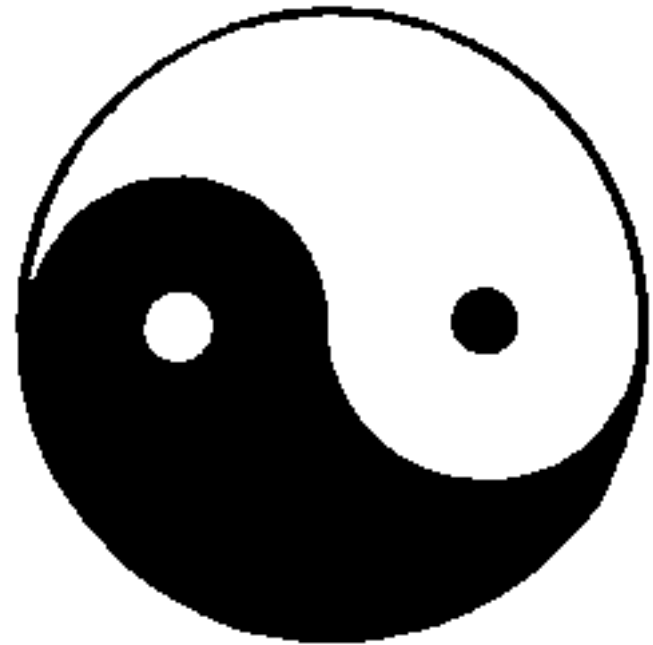
x, y - prostorne koordinate

I – intenzitet piksela (brightness)



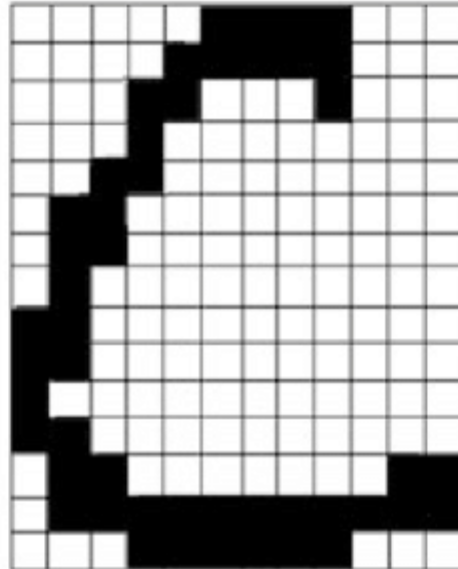
Binarna slika

- sastoji se samo od crnih i bijelih piksela
- važna u industrijskim aplikacijama kada senzori rade sa najmanjom mogućom količinom informacija.



Binarna slika

- sastoji se samo od crnih i bijelih piksela
- važna u industrijskim aplikacijama kada senzori rade sa najmanjom mogućom količinom informacija.



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

Sivoskalirana slika (grayscale)

- Moguće vrijednosti intenziteta su u opsegu $0 - 2^k - 1$. k je najčešće 8 .



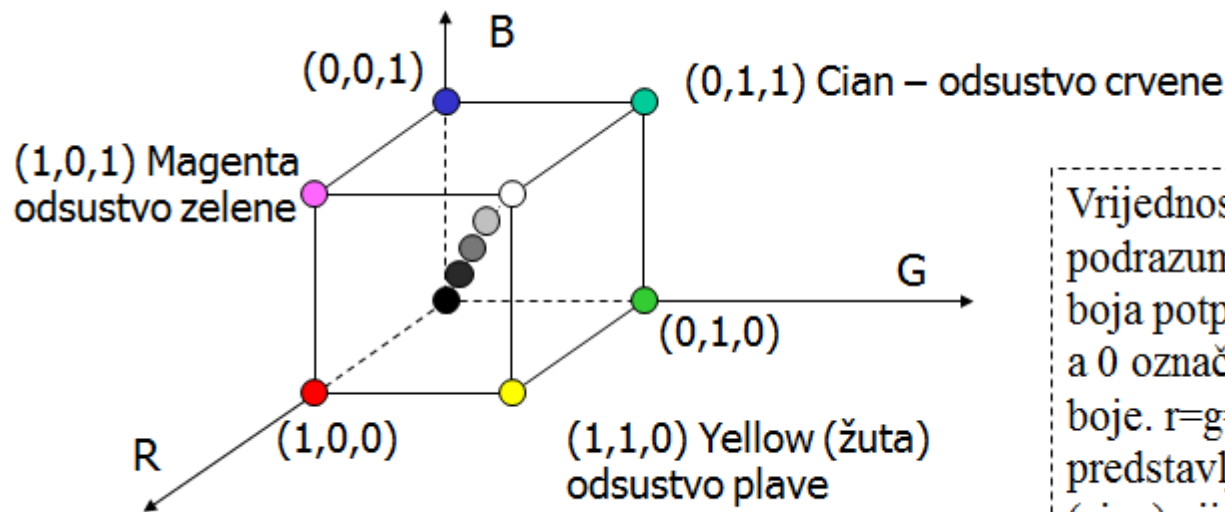
Sivoskalirana slika (grayscale)

- **crno** je predstavljano sa brojem **0** dok je svaka nijansa sive boje predstavljena kao jedan k -bitni broj. Čisto **bijeloj** boji odgovara 2^k-1 .



Slika u boji

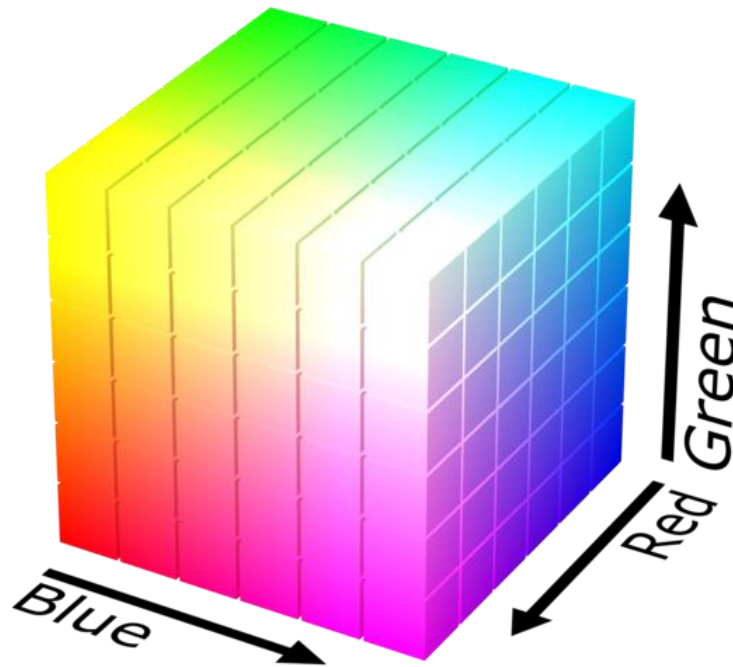
- Young je još 1802.-ge uočio da se svaka boja vidljivog spektra može prikazati preko tri nezavisne boje. Najčešće su nezavisne boje **R** (red-crvena), **G** (green-zelena) i **B** (blue-plava). Ove boje se obično smiještaju u uglove kocke u Dekartovim koordinatama.



Vrijednost 1 podrazumijeva da je boja potpuno uključena a 0 označava odsustvo boje. $r=g=b$ prava predstavlja ahromatske (sive) nijanse.

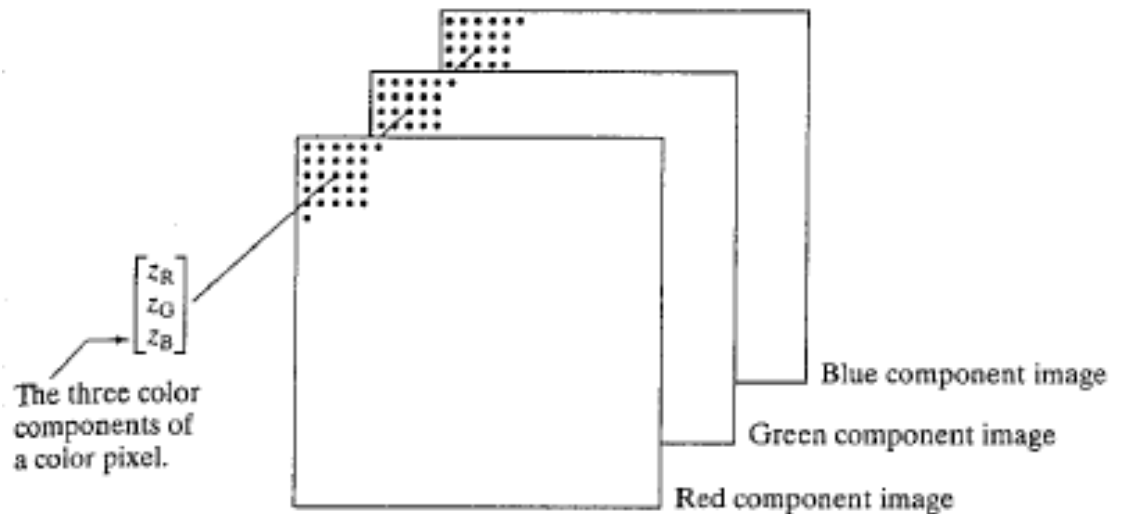
Slika u boji

- Young je još 1802.-ge uočio da se svaka boja vidljivog spektra može prikazati preko tri nezavisne boje. Najčešće su nezavisne boje **R** (red-crvena), **G** (green-zelena) i **B** (blue-plava). Ove boje se obično smiještaju u uglove kocke u Dekartovim koordinatama.



Slika u boji

- RGB slika se predstavlja kao 3D niz (ili stek) od **3 matrice ili 3 sivoskalirane slike**
- Dimenzije tog niza su $w \times h \times 3$
- Postoje crvena (R), plava (B) i zelena (G) komponenta (ili kanal)
- Za zapis slike dimenzija $w \times h$ u RGB je potrebno $w \times h \times 3 \times k$ bita (k je obično jednako 8).
- Intenzitet piksela je vektor ili uređena trojka (r, g, b) a sve 3 vrijednosti imaju vrijednost između 0 i 1



Konverzija RGB -> sivoskalirana slika

- Obično se primjenjuje formula:

$$g \approx (R + G + B) / 3$$



Konverzija sivoskalirana slika -> binarna slika

- Thresholding (binarizacija) – rasporedjivanje SVIH piksela u 2 klase (tamnu i svijetlu)
- Svaki pixel postaje ili 0 ili 1



Konverzija sivoskalirana slika -> binarna slika

- Binarizacija se izvodi po formuli

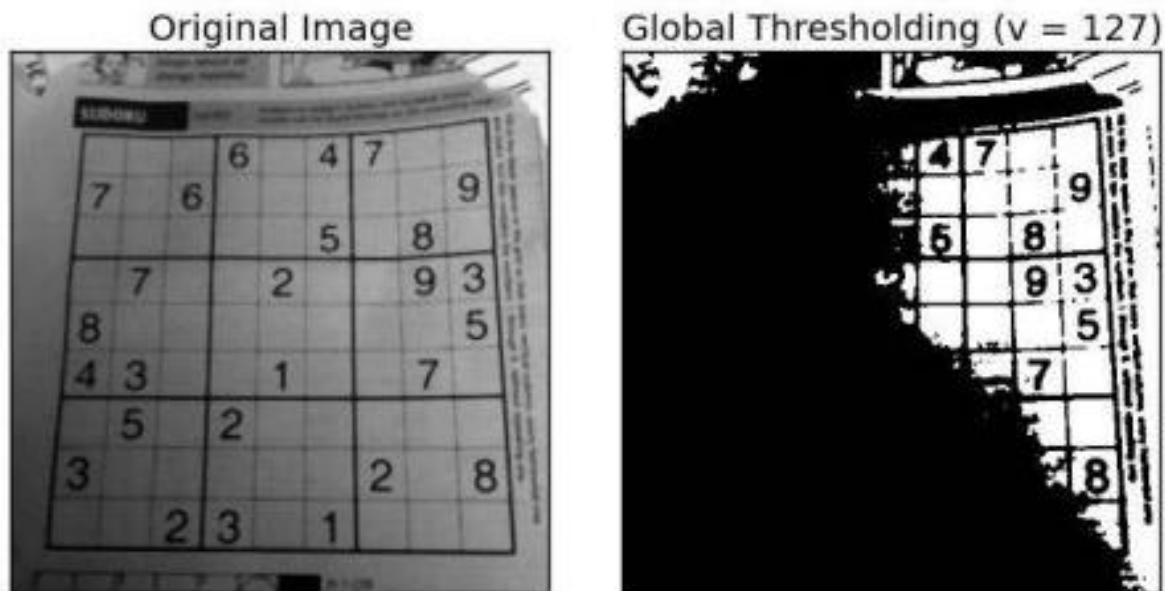
$$g(n, m) = \begin{cases} 1 & f(n, m) > T \\ 0 & f(n, m) \leq T \end{cases}$$

Prag je ovdje **globalan** (isti za cijelu sliku) ali može biti i **lokalan** odnosno oblika $T(n, m)$.



Konverzija sivoskalirana slika -> binarna slika

- Binarizacija nije lak problem



Konverzija sivoskalirana slika -> binarna slika

- Binarizacija nije lak problem



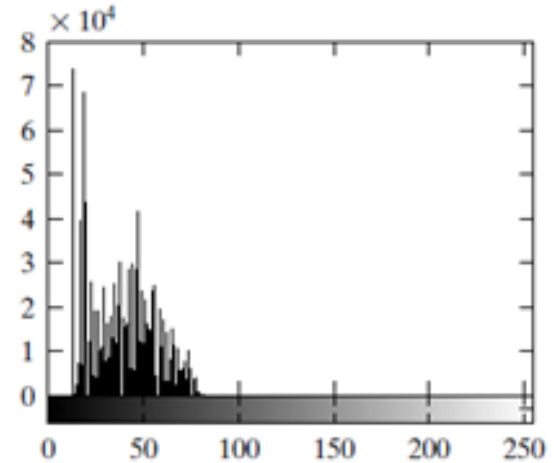
Konverzija sivoskalirana slika -> binarna slika

- Binarizacija nije lak problem

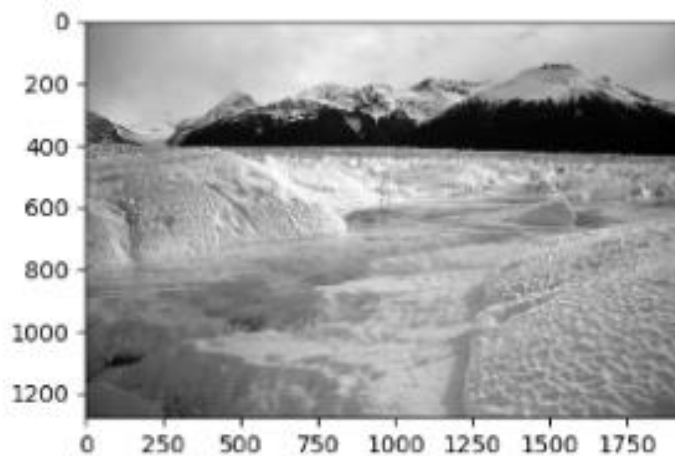


Statistike slike

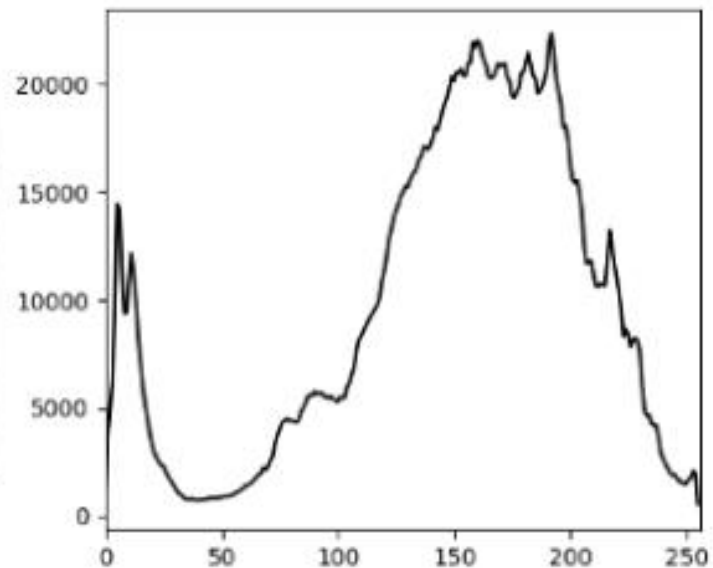
- Najpoznatija statistika slike je Histogram
- $h(i) = n_i$, $i = 0..255$
- n_i - broj piksela u slici koji imaju intenzitet i
- $p(i) = h(i)/n$ – normalizovani histogram
– (vjerovatnoća pojavljivanja piksela intenziteta i)



Histogram slike



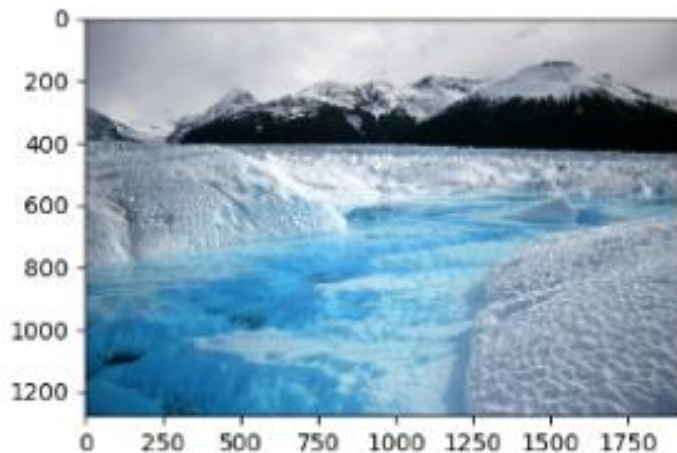
Sivoskalirana slika



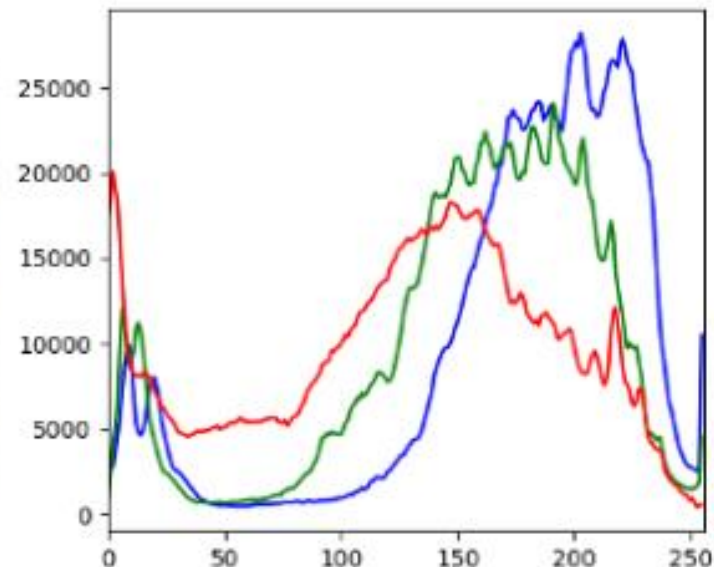
Njen histogram

Histogram slike

- 1-dimenzionalni histogram za RGB slike
- Po jedan 1-dimenzionalni histogram za svaki kanal



RGB slika



Njena 3 histograma

Sa digitalnom slikom nastavljamo sljedeći čas

- Literatura:

*Digital Image Processing 2nd Edition (Rafael
Gonzales, Richard Woods)*

Alati za vježbe

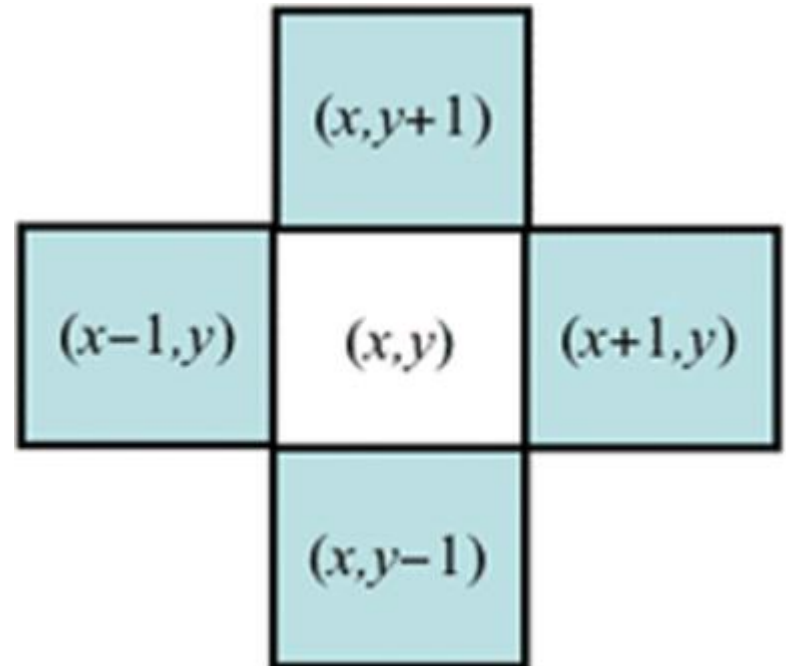
- Visual Studio Code kao okruženje
- Python
- OpenCV biblioteka
- Open3D biblioteka
- Point Cloud Library (PCL)
- 3D senzori koje imamo na raspolaganju:
 - 2 Realsense kemere (ima Python SDK),
 - <https://dev.intelrealsense.com/docs/python2>
 - 1 Microsoft Azure Kinect DK (SDK?)
 - (<https://azure.microsoft.com/en-us/services/kinect-dk/#overview>)
 - 1 ORBBEC (SDK?)
 - <https://orbbe3d.com/develop/>
- Lista nije konačna

Vježba 1

- Napisati i pokrenuti „Hello World“ primjer u Python-u
- Instalirati OpenCV pomoću *pip*-a
- Učitati „sliku Lene“
 - <http://optipng.sourceforge.net/pngtech/img/lena.html>
- Prikazati je sa `imshow` funkcijom
- Izdvojiti kanale R, G i B
- Konvertovati sliku u sivoskaliranu (grayscale)
- Konvertovati sivoskaliranu sliku u binarnu sliku tako što se pixeli sa vrijednošću ispod 127 setuju na 0(crna) a oni sa vrijednošću ≥ 127 se setuju na 255(bijela)
- Prikazati sivoskaliranu i binarnu sliku

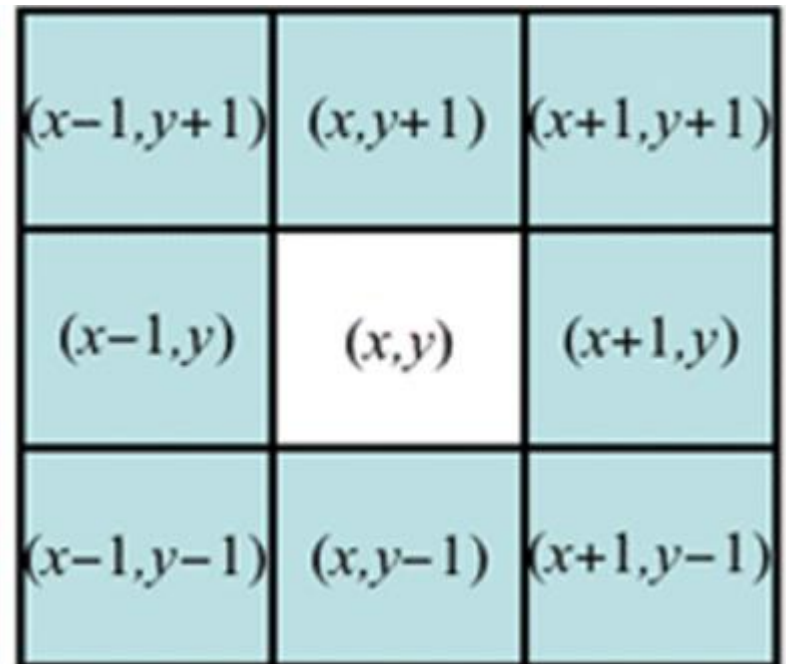
Susjedstvo u digitalnoj slici

- Pikel p na lokaciji (x,y) u slici ima 2 **vertikalna** i 2 **horizontalna** susjeda
- Ova 4 susjeda imaju koordinate: $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, $(x,y-1)$
- Ovaj skup od 4 piksela se zove **4-susjedstvo** (4-neighborhood) piksela p i označava se sa $N_4(p)$



Susjedstvo u digitalnoj slici

- 4 **dijagonalna** susjeda piksela p se nalaze na koordinatama: $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y+1)$, $(x-1,y-1)$
- Ovaj skup označavamo sa $N_D(p)$
- Zajedno $N_4(p) + N_D(p)$ čine **8-susjedstvo** piksela p (8-neighborhood) $N_8(p)$



Putanja

- Putanja od piksela p na lokaciji (x,y) do piksela q na lokaciji (s,t) je skup različitih piksela (x_0,y_0) , (x_1,y_1) , ..., (x_n,y_n) takvih da važi:

$$(x_0,y_0)=(x,y), (x_n,y_n)=(s,t) \text{ i}$$

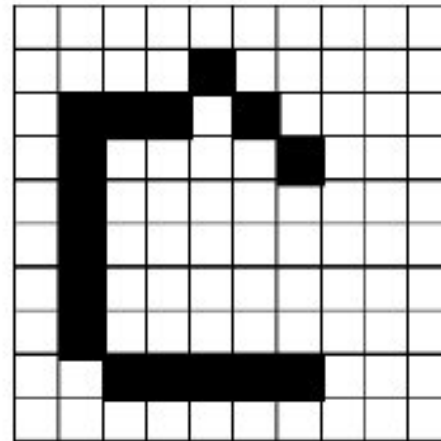
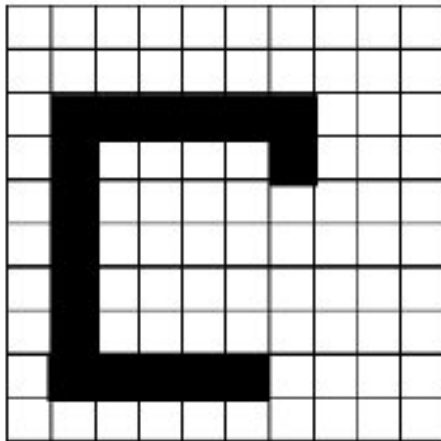
$$(x_i,y_i) \text{ je } \mathbf{susjed} (x_{i-1},y_{i-1}) \text{ za } 1 \leq i \leq n$$

n – dužina putanje

Ako važi $(x_0,y_0)=(x_n,y_n)$, kažemo da je putanja **zatvorena**.

Putanja

- Zavisno od toga kako definišemo susjedstvo, postoje 4-putanje i 8-putanje

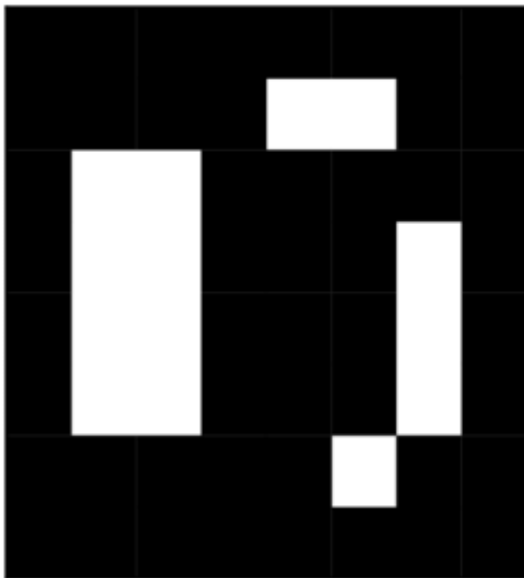


Povezanost piksela

- *eng. connectivity*
- Neka je S skup piksela u slici i neka $p, q \in S$.
- Kažemo da je q **povezan** sa p u S ako postoji putanja od p do q sastavljena **isključivo** od piksela koji pripadaju S .

Region

- Za svaki piksel p u skupu S , skup svih piksela povezanih sa p zovemo **povezana komponenta** (connected component)
- Ako skup S ima **samo jednu** povezanu komponentu, S zovemo **povezani skup** (connected set) ili **region**.

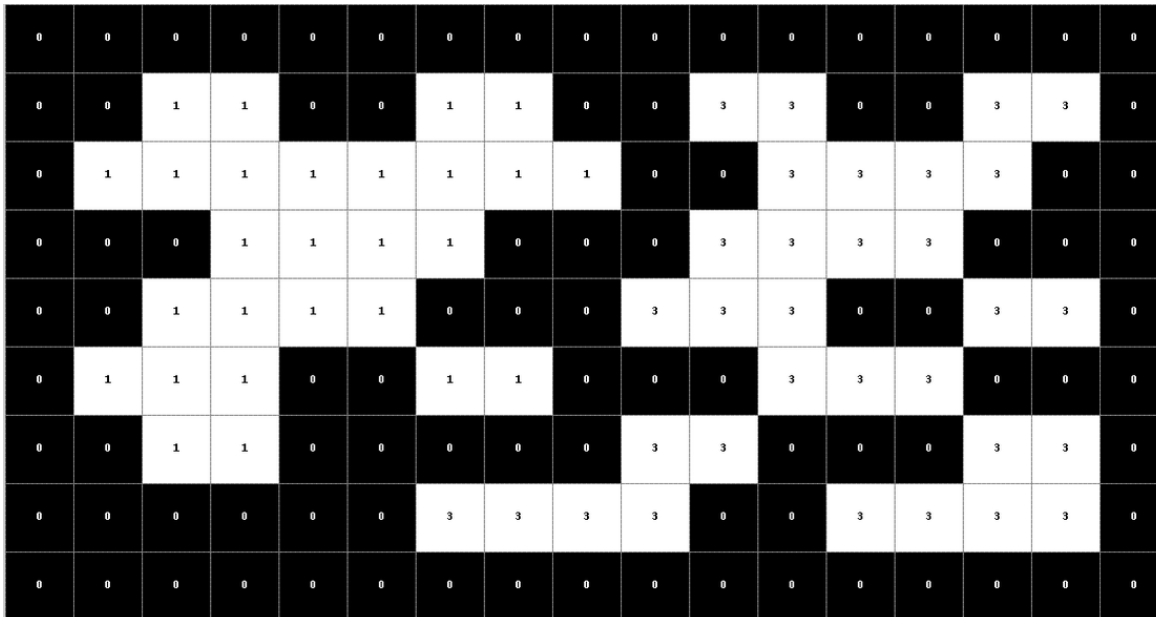


0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	0	0	2	2	0	0
0	1	1	0	0	0	0	0
0	1	1	0	0	0	3	0
0	1	1	0	0	0	3	0
0	1	1	0	0	0	3	0
0	0	0	0	0	4	0	0
0	0	0	0	0	0	0	0

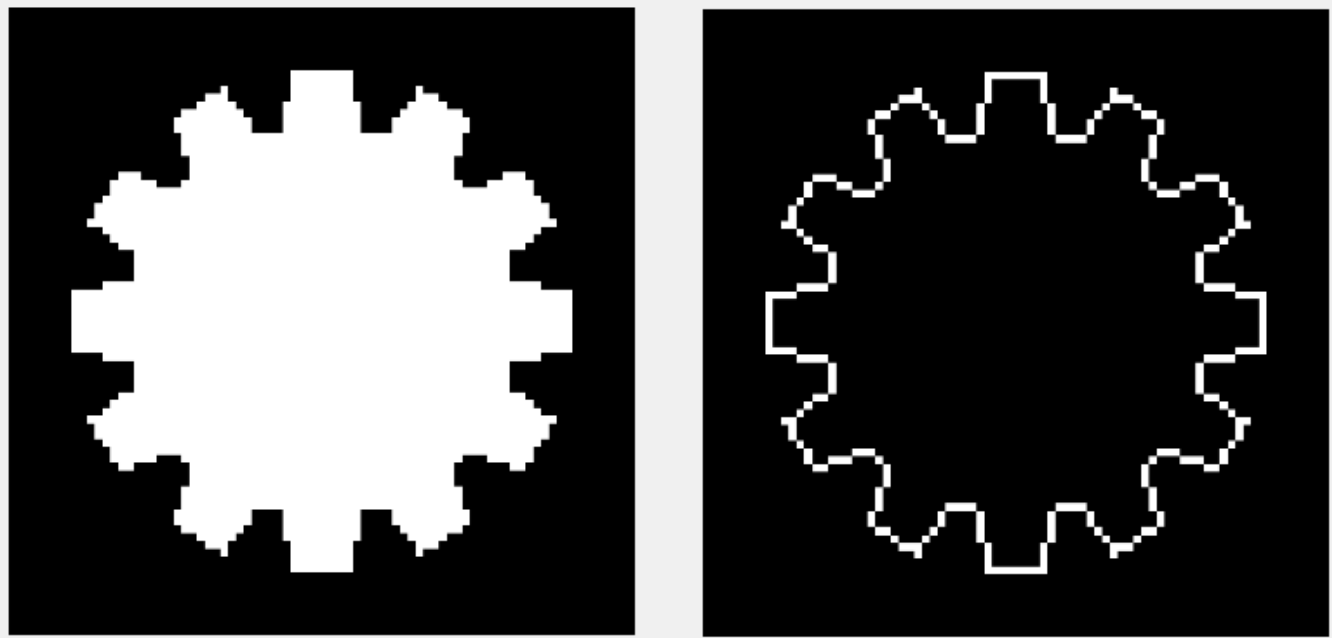
Region

- Za svaki piksel p u skupu S , skup svih piksela povezanih sa p zovemo **povezana komponenta** (connected component)
- Ako skup S ima **samo jednu** povezanu komponentu, S zovemo **povezani skup** (connected set) ili **region**.



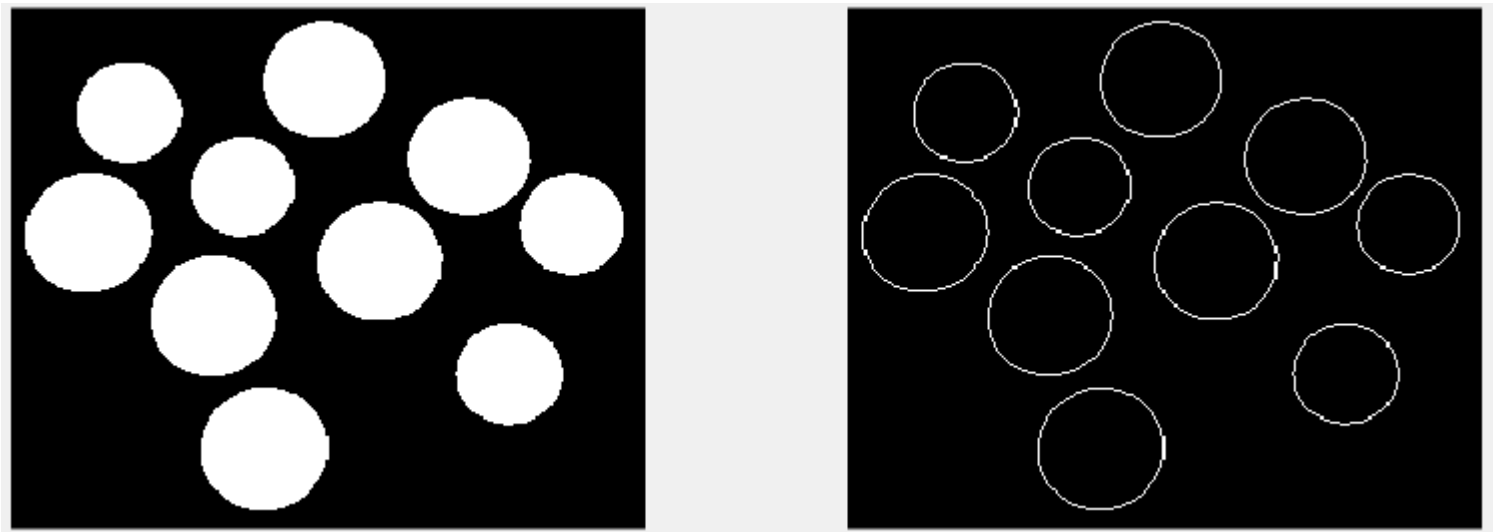
Granica/kontura/ivica

- Granica/kontura/ivica (eng. boundary/border/contour) regiona R je skup piksela iz R koji imaju **najmanje jednog** susjeda koji nije iz R .



Granica/kontura/ivica

- Granica/kontura/ivica (eng. boundary/border/contour) regiona R je skup piksela iz R koji imaju **najmanje jednog** susjeda koji nije iz R .



2 vrste operacija na slikama

- Transformacija intenziteta

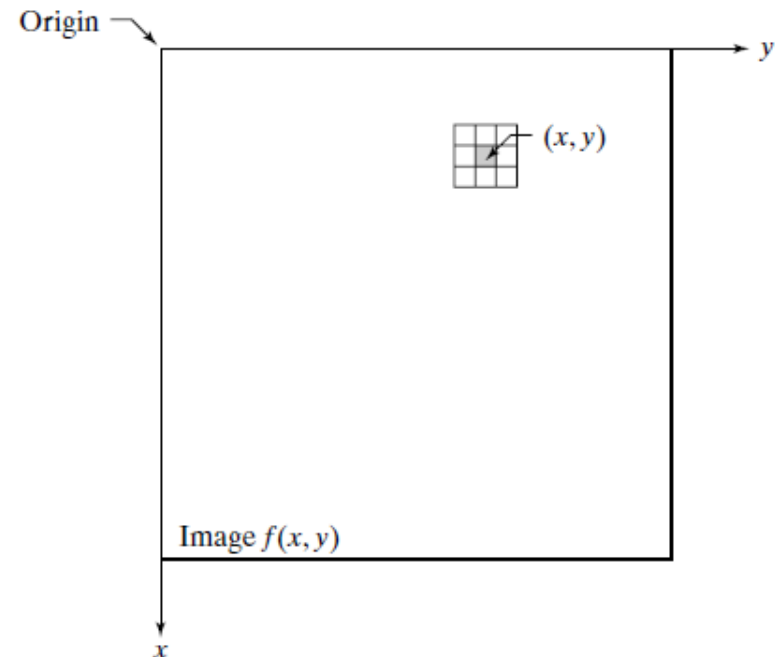
(lokacija piksela nema uticaj na novu vrijednost)

$$T(I) = I_{new}$$

- Prostorno filtriranje

(susjedi određuju novu vrijednost piksela)

(eng. spatial filtering)



Transformacije intenziteta

- Npr. negativ (oduzimanje od potpuno bijele slike)

Sivoskalirana



Binarna

Lampent used
Shadow Ball!

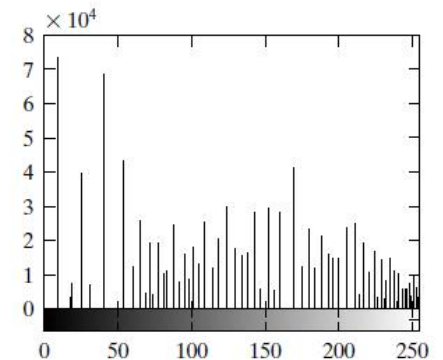
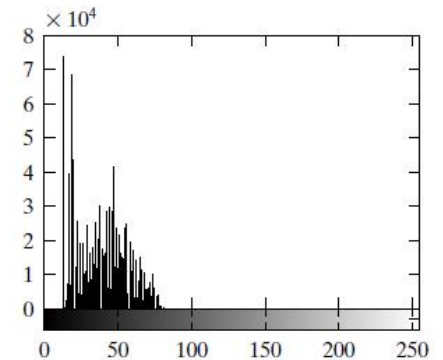
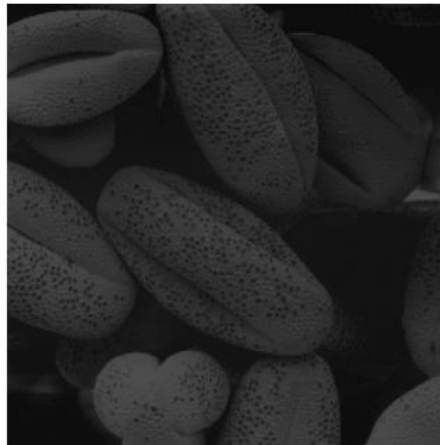
Lampent used
Shadow Ball!

Transformacije intenziteta

$$s = T(r) = \int_0^r p_r(w) dw$$

- Izjednačavanje histograma (histogram equalization)
- Poslije ove transformacije:
 1. Opseg intenziteta pokriva cijeli gray-level opseg
 2. Svi intenziteti imaju *sličniji* broj pojavljivanja

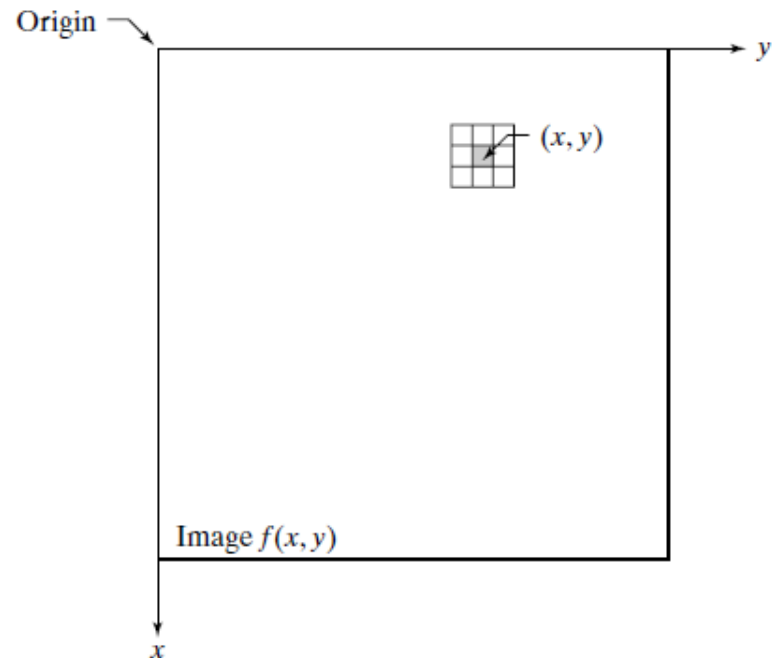
Krajnji efekat: veći kontrast



Prostorno filtriranje (spatial filtering)

susjedi određuju novu vrijednost
piksela

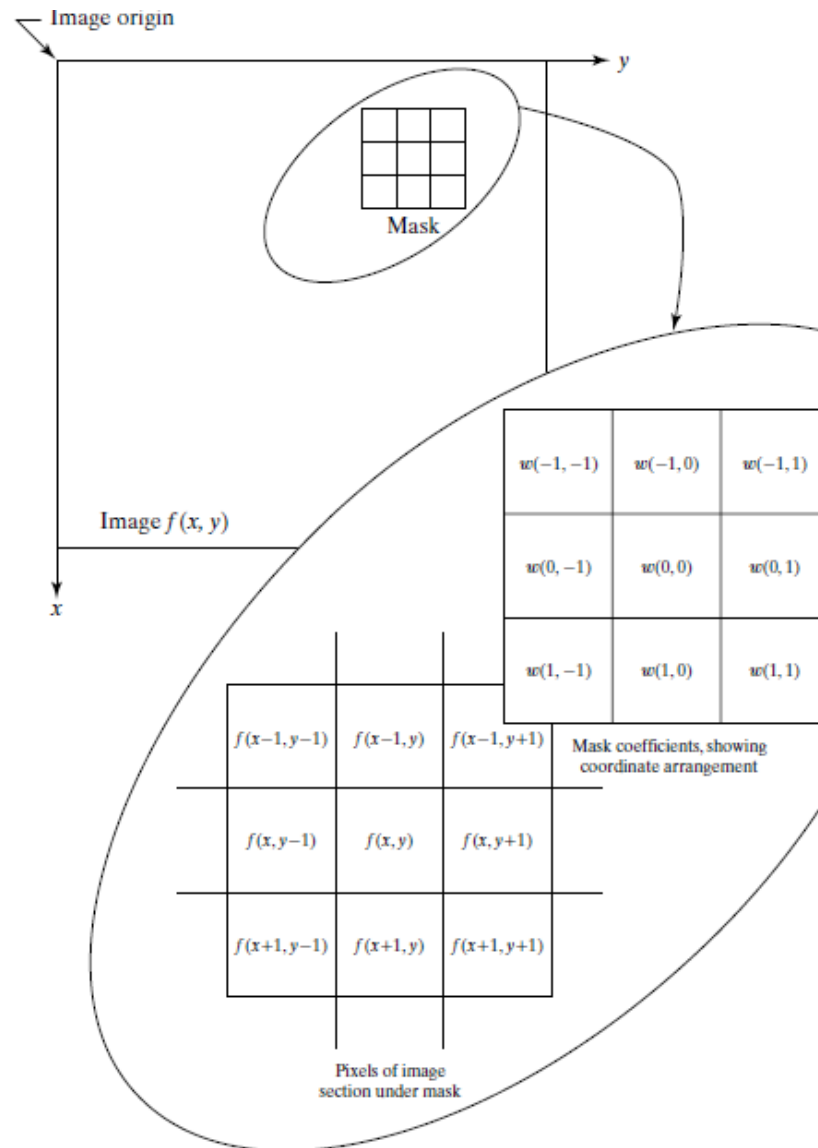
1. Linearno prostorno filtriranje
2. Nelinearno prostorno filtriranje



Linearni filteri

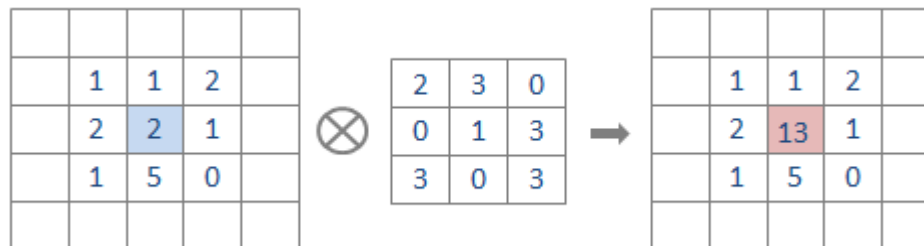
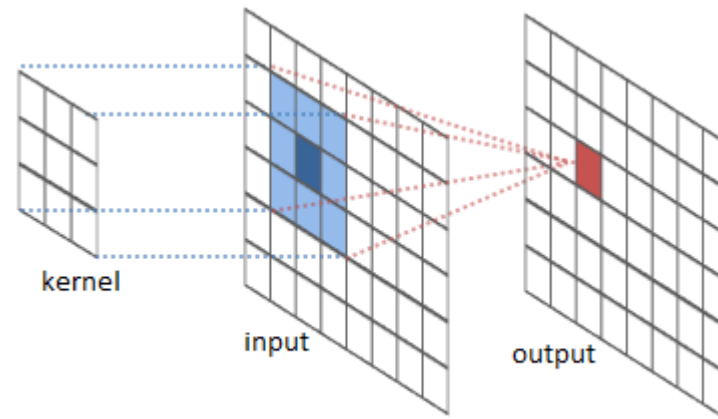
- Nazivaju se i **konvolucioni**
- Primjena linearnog filtera -
(množenje filtera i piksela slike
element-wise - **konvolucija**)

Alternativni termini za filter: *maska*, *kernel*,
filter mask, *template*, *window*, *sliding window*



Linearni filteri

- Konvolucija



Linearni filteri - primjene

- Pokretna sredina (moving average filter)
 - Ovaj filter se koristi za uklanjanje šuma
 - Nažalost, njegova primjena uzrokuje i zamagljivanje slike



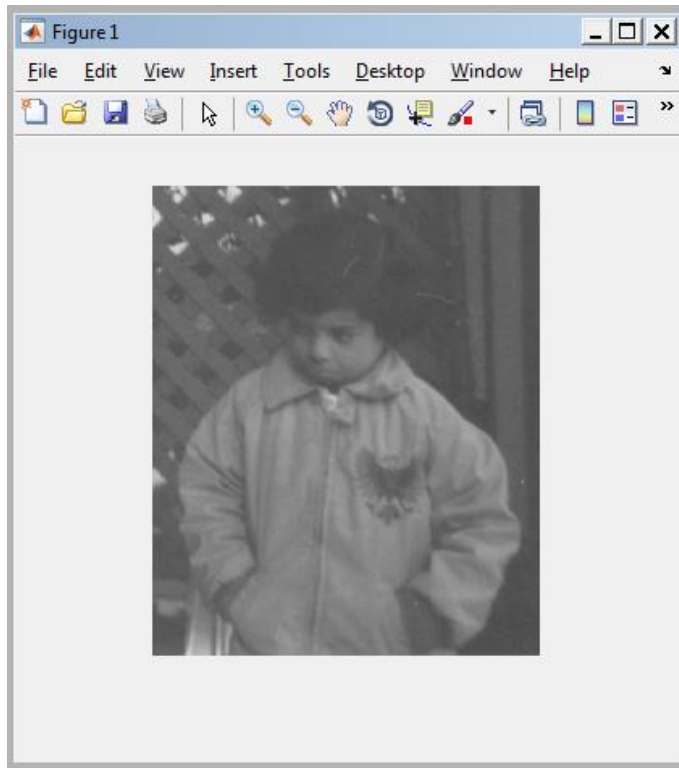
$$\frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} =$$



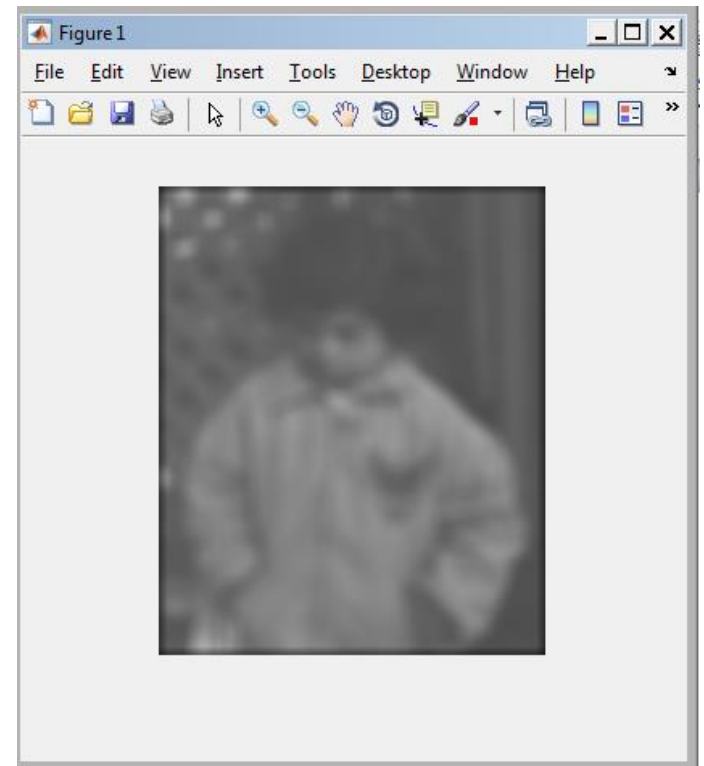
Filter veličine 3x3

Linearni filteri - primjene

- Pokretna sredina (moving average filter)



Originalna slika



Poslije primjene moving average filtera veličine **11x11**

Linearni filteri - primjene

- Gausovsko zamućenje (Gaussian blur)

$1/16$

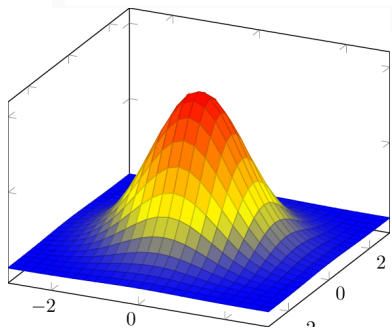
1	2	1
2	4	2
1	2	1

$1/273$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

$1/1003$

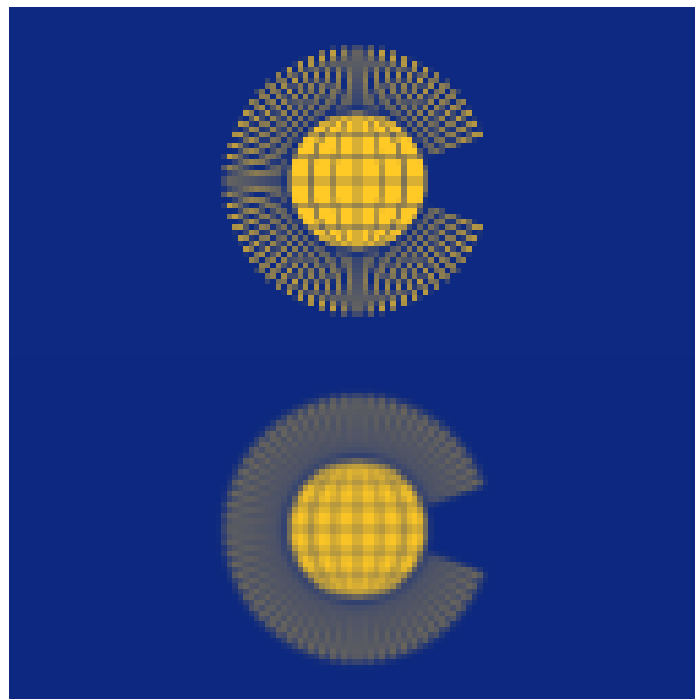
0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0



Gausovi filteri različitih veličina

Linearni filteri - primjene

- Gausovsko zamućenje (Gaussian blur)



Linearni filteri - primjene

- **Gaussian** filter (otklanjanje šuma ali i zamućivanje ako se izabere preveliki filter)
- uticaj veličine filtera



Linearni filteri - primjene



Laplacian filter (izoštavanje)

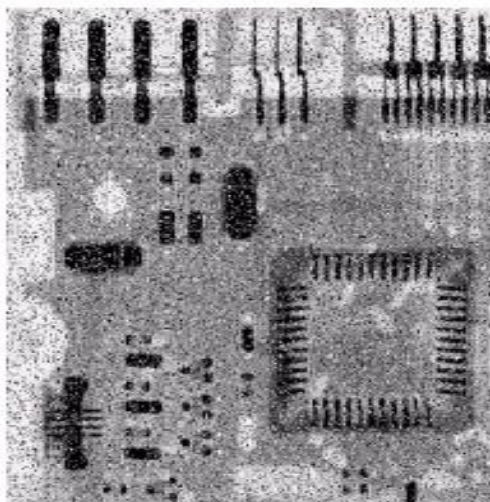
Mane konvolucionih filtera

- Brzi zbog svoje linearnosti ali se ne mogu koristiti za uklanjanje impulsnog šuma
- Impulsni šum se rijetko pojavljuje ali ima izuzetno velike vrijednosti. Konvolucionni filtri ove vrijednosti samo razmazuju po slici šireći njihov uticaj.
- Impulsni šum se često pojavljuje kod digitalne slike što zbog načina akvizicije što zbog prenosa slike komunikacionim kanalom gdje se mogu pojaviti ovakve smetnje.

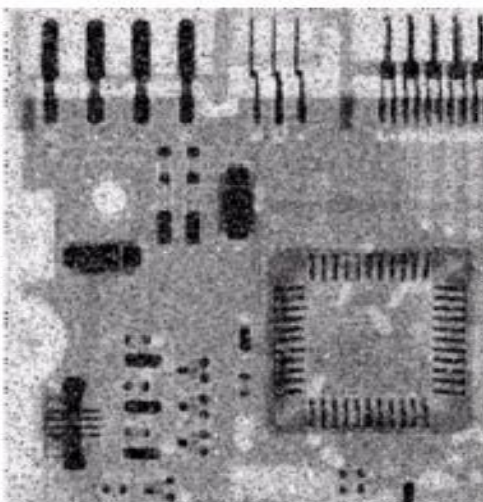
Nelinearno prostorno filtriranje

- Na pikselima slike koji su „ispod“ filtera, se izvršava neka nelinearna operacija, npr. max, min, mean, median
- Najpoznatiji predstavnik je **median** filter -> uklanja „salt and pepper“ šum (impulsni šum)
- Ne zamagljuje sliku (posebno ivice) ali može da ošteti sitne detalje

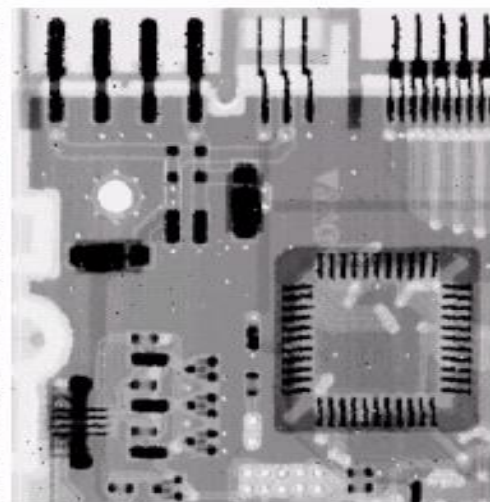
Originalna slika



Pokretna sredina

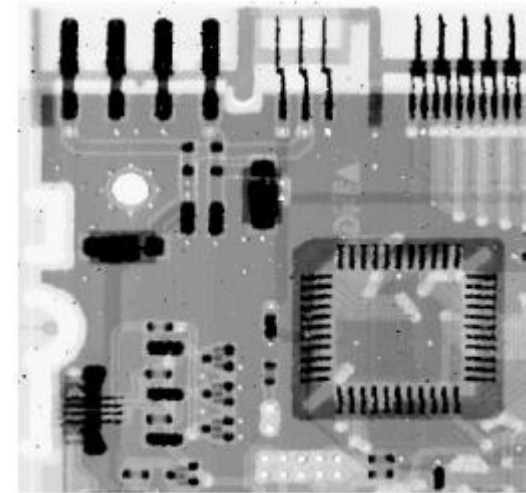
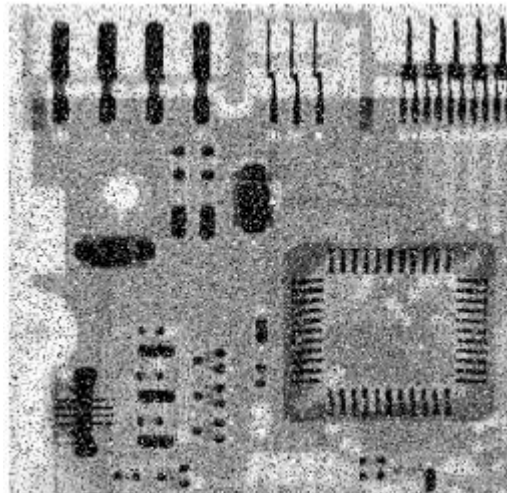


Median filter

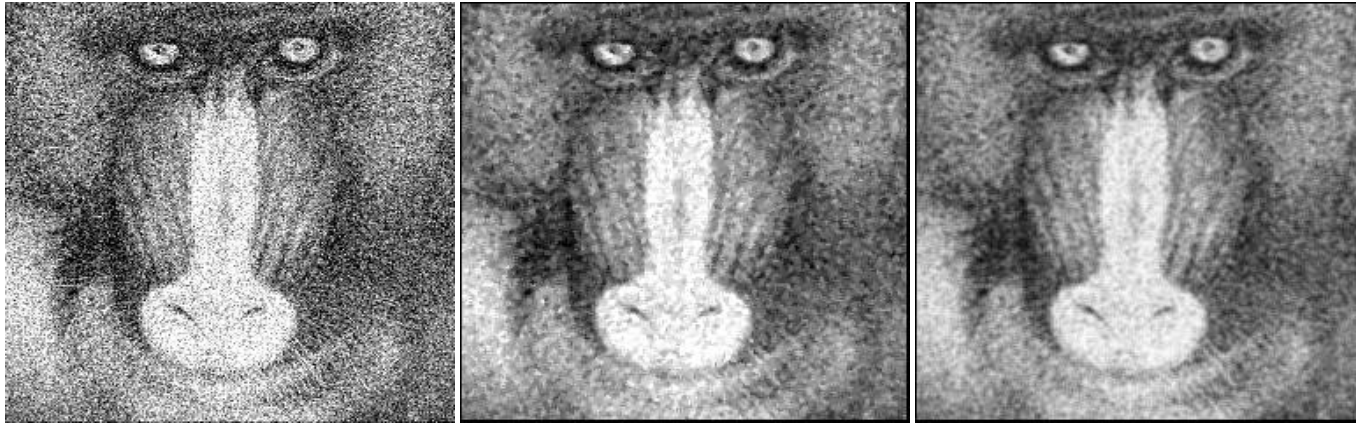


Nelinearno prostorno filtriranje

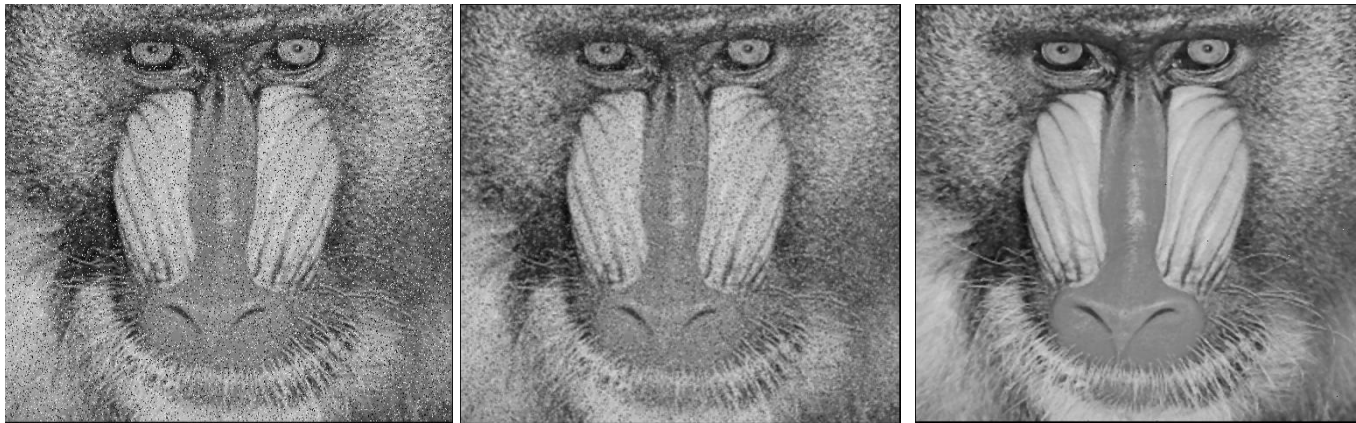
- Kako radi median filter?
 - ne postoje elementi filtera
 - računa medijanu sekvence koja je u filteru
- Npr. ako je sortirana sekvenca 1 3 5 7 21, mediana je 5. Kod sekvence sa parnim brojem elemenata usvaja se poluzbir srednja dva
- Problem: računaska složenost jer se svaki put radi sortiranje lokalnog susjedstva



Poređene median – pokretna sredina



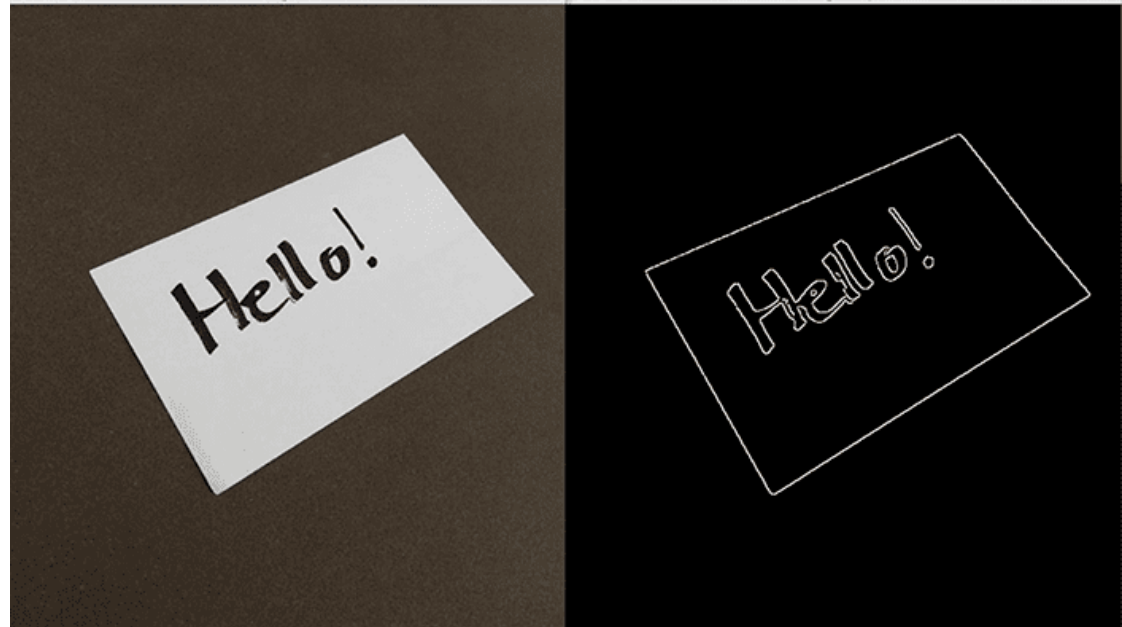
Slika zahvaćena Gausovim šumom i filtrirana filtrom sa pokretnom sredinom i median filtrom. Kod filtriranih slika se ne uočava bitna razlika u kvalitetu.



Slika zahvaćena Laplasovim šumom i filtrirana filtrom sa pokretnom sredinom i median filtrom. Median ima znatno bolji kvalitet.

Detekcija ivica

- Jedna od najvažnijih karakteristika slike je **ivica (edge)**.
- Definiše se kao nagli prelaz sa jednog na drugi osvjetljaj.
- Često se na osnovu polazne slike formira specijalna binarna slika $e(n,m)$ koja uzima vrijednost 1 (bijelo) ako je piksel (n,m) polazne slike ivica a suprotno uzima vrijednost 0 (crno).
- Slika desno se zove *edge map* ili *edge image*



Source:

<https://www.pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/>

Detekcija ivica

- Ivica se može shvatiti kao pozicija na kojoj **prvi izvod** postiže maksimum. Stoga ivicu možemo tražiti kao maksimum apsolutne vrijednosti izvoda slike po pojedinim pravcima.
- Problem je što je slika diskretna veličina a ne kontinualna što vodi ka tome da umjesto izvoda moramo koristiti konačne razlike duž x i y ose.

Prvi izvod funkcije

- Prvi izvod (gradient) – vektor duž koga je promjena intenziteta maksimalna
- Gradient ima:
 - intenzitet/magnitudu
 - koliko je jaka promjena
 - pravac/orijentaciju
 - duž kog pravca se dešava najveća promjena intenziteta

131	162	232	84	91	207
104	93	139	101	237	109
243	26	252	196	135	126
185	135	230	48	61	225
157	124	25	14	102	108
5	155	116	218	232	249

Prvi izvod – ručno računanje

- Koristićemo vrijednosti ovih piksela

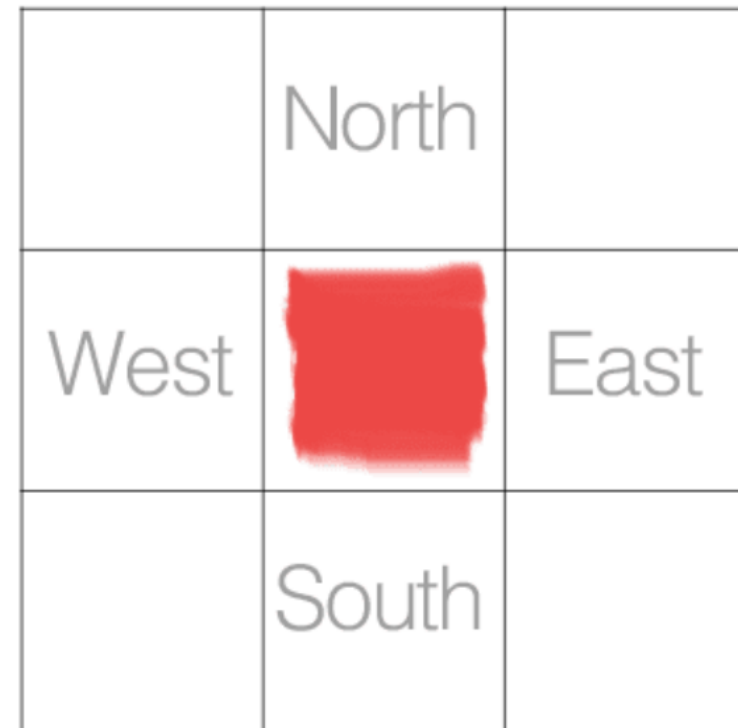
North: $I(x, y - 1)$

South: $I(x, y + 1)$

East: $I(x + 1, y)$

West: $I(x - 1, y)$

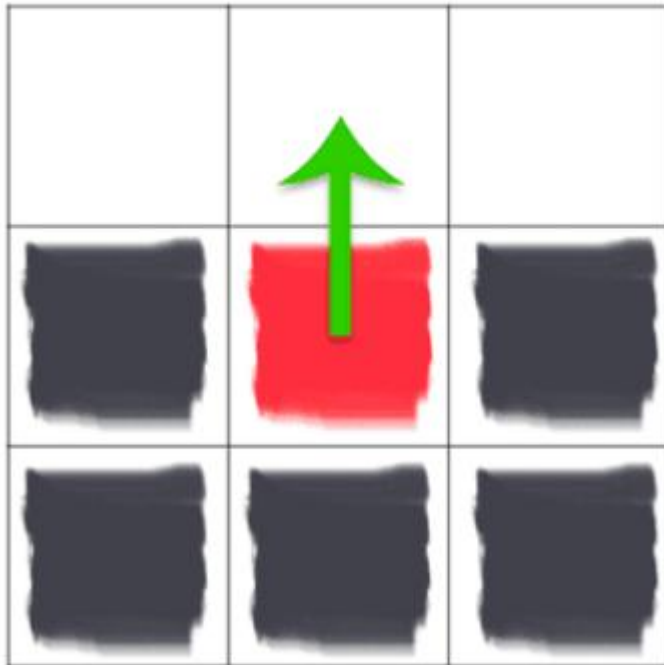
- $G_y = I(x, y + 1) - I(x, y - 1)$
- $G_x = I(x + 1, y) - I(x - 1, y)$



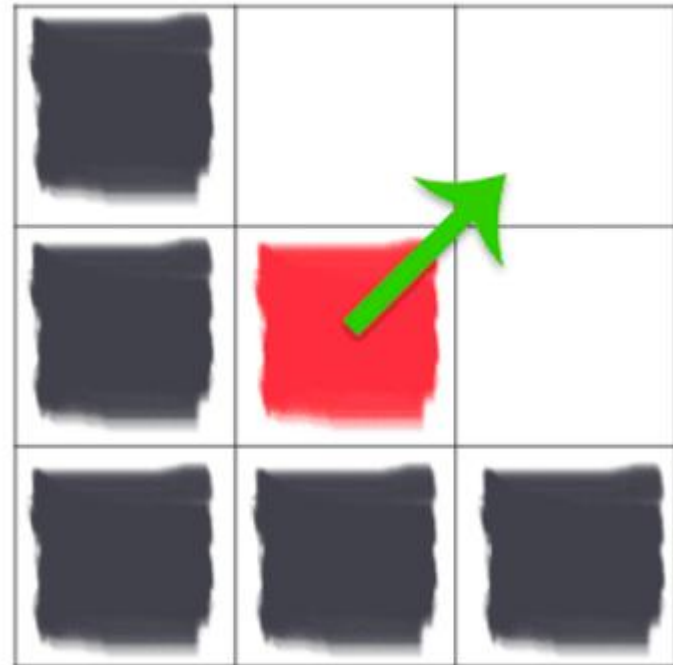
Prvi izvod – ručno računanje

- Orijehtacija gradijenta

$\theta = -90^\circ$



$\theta = -45^\circ$

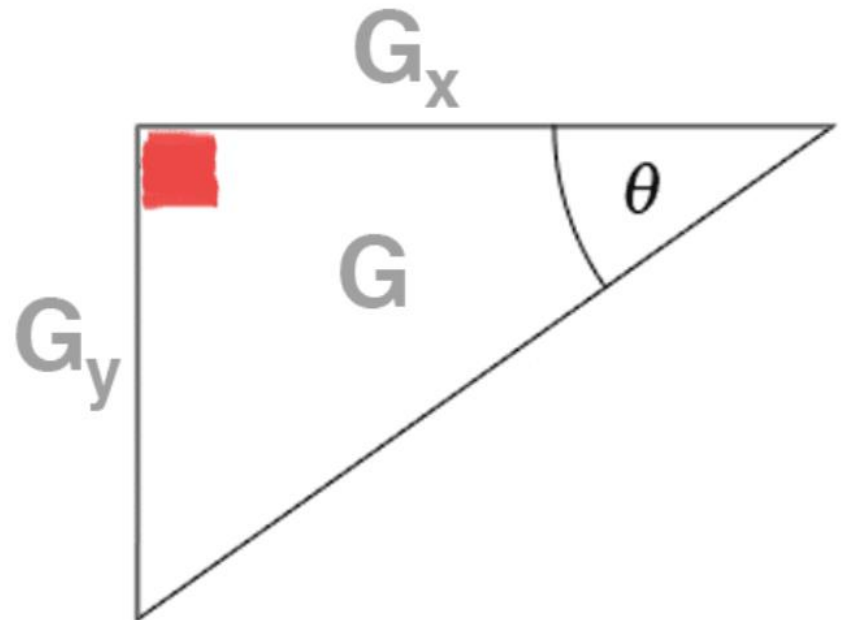


Prvi izvod – ručno računanje

- Magnituda i orijentacija gradijenta

$$G = \sqrt{G_x^2 + G_y^2}$$







$$\theta = \arctan2(G_y, G_x) \times \left(\frac{180}{\pi} \right)$$



Prvi izvod – ručno računanje

- Primjer
- $G_x?$ $G_x?$ $G?$ $\theta?$

255	255	255
0		0
0	0	0







Prvi izvod – ručno računanje

- Primjer

- $G_x = 0 - 0 = 0$

$$G_y = 0 - 255 = -255$$

255	255	255
0		0
0	0	0







Prvi izvod – ručno računanje

- Primjer

$$G = \sqrt{0^2 + (-255)^2} = 255$$

$$\theta = \arctan2(-255, 0) \times \left(\frac{180}{\pi}\right) = -90^\circ$$





255	255	255
0		0
0	0	0


		
		

Prvi izvod – ručno računanje

- Još jedan primjer
- $G_x?$ $G_x?$ $G?$ $\theta?$

255	0	0
255		0
255	255	255







Prvi izvod – ručno računanje


- Primjer

- $G_x = 0 - 255 = -255$

$$G_y = 255 - 0 = 255$$

255	0	0
255		0
255	255	255







Prvi izvod – ručno računanje

- Primjer

$$G = \sqrt{(-255)^2 + 255^2} = 360.62$$

$$\theta = \arctan2(255, -255) \times \left(\frac{180}{\pi}\right) = 135^\circ$$

255	0	0
255		0
255	255	255

Detekcija ivica

- Najčešći (najjednostavniji) način koji se u obradi koristi za detekciju ivica je na osnovu **maske**.
- Maske su matrice dimenzija 3×3 sa kojima se slika konvoluirala.
- Obično dolaze u paru i jedna se koristi za detekciju ivica duž horizontale a druga (rotirana ili transponovana) za detekciju ivica duž vertikale.
- Najpoznatije maske su: Prewittova i Sobelova. Njihove maske su:

$$\begin{bmatrix} 1 & K & 1 \\ 0 & 0 & 0 \\ -1 & -K & -1 \end{bmatrix}$$

Za ivice u
x-pravcu

$$\begin{bmatrix} -1 & 0 & 1 \\ -K & 0 & K \\ -1 & 0 & 1 \end{bmatrix}$$

Za ivice u
y-pravcu

Prewittova i Sobelova matrica

- Za $K=1$ slijedi Prewittova matrica a za $K=2$ slijedi Sobelova.
- **Sobelov** detektor se u ovoj klasi najčešće koristi:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

- Pored ovih postoje i drugi detektori ivica (Canny-jev, Laplasov, Gaus-Laplasov, Kumani, Robertsov itd.).

Sobelov filter

- Primjer

$$I_{i,j} = \begin{bmatrix} 93 & 139 & 101 \\ 26 & 252 & 196 \\ 135 & 230 & 18 \end{bmatrix}$$

$$G_x = \sum \begin{bmatrix} -1 \times 93 & 0 \times 139 & 1 \times 101 \\ -2 \times 26 & 0 \times 252 & 2 \times 196 \\ -1 \times 135 & 0 \times 230 & 1 \times 18 \end{bmatrix} = \sum \begin{bmatrix} -93 & 0 & 101 \\ -52 & 0 & 392 \\ -135 & 0 & 18 \end{bmatrix} = 231$$

$$G_y = \sum \begin{bmatrix} -1 \times 93 & -2 \times 139 & -1 \times 101 \\ 0 \times 26 & 0 \times 252 & 0 \times 196 \\ 1 \times 135 & 2 \times 230 & 1 \times 18 \end{bmatrix} = \sum \begin{bmatrix} -93 & -278 & -101 \\ 0 & 0 & 0 \\ 135 & 460 & 18 \end{bmatrix} = 141$$

$$G = \sqrt{231^2 + 141^2} = 270.63 \quad \text{and} \quad \theta = \arctan2(141, 231) \times \frac{180}{\pi} = 31.4^\circ$$

Sharr filter

- Alternativa za Sobel filter
- Sobel se koristi češće

$$G_x = \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

Vježba 2

- Učitati slike
 - Source:
 - <https://www.sampleforms.com/admission-forms.html>
 - <https://guillaumejaume.github.io/FUNSD/download/>
 - 1. Primijeniti Sobel filter
 - 2. Izračunati magnitudu i orijentaciju
 - 3. Binarizovati sliku magnitude da dobijemo ivice
 - 4. Komentarisati rezultat
-
- Ukoliko su slike u boji ili sivoskalirane, konvertovati ih u binarne (testirati adaptive thresholding i Otsu thresholding)
 - Ponoviti 1,2,3 i 4
-
- Koristićemo OpenCV funkcije za binarizaciju i detekciju kontura i pokušati da detektujemo pravogaonike na slici
 - Bonus za danas: pokrenuti 2D ili 3D stream (ili oba) sa 3D kamere

Funkcije distance

- ...ili metrike
- Def. Za piksele p, q, z sa koordinatama (x, y) , (s, t) , (u, v) D je funkcija distance ili metrika ako važi:
 1. $D(p, q) \geq 0$ ($D(p, q) = 0$ iff $p = q$)
 2. $D(p, q) = D(q, p)$ i
 3. $D(p, z) \leq D(p, q) + D(q, z)$

Euklidsko rastojanje

- eng. *Euclidean distance*

$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{1/2}$$

- Tačke (pikseli) koji su udaljeni na distancama manjim ili jednakim od r od tačke (piksela) (x,y) formiraju **disk** sa poluprečnikom r i centrom u (x,y) .

D_4 rastojanje

- eng. D_4 distance ili *city-block distance*

$$D_4(p,q) = |x-s| + |y-t|$$

- Tačke (pikseli) koji su udaljeni na distancama manjim ili jednakim od r od tačke (piksela) (x,y) formiraju **dijamant** oko piksela (x,y)
- Npr pikseli na rastoj

2
2 1 2
2 1 0 1 2
2 1 2
2

Koji skup čine pikseli sa $D_4 = 1$?

D_8 rastojanje

- eng. D_8 distance ili *chessboard distance*

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

- Tačke (pikseli) koji su udaljeni na distancama manjim ili jednakim od r od tačke (piksela) (x,y) formiraju **kvadrat** oko piksela (x,y)
- Npr pikseli na rastojanju $D_8 \leq 2$ od p :

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Koji skup čine pikseli sa $D_8 = 1$?

Image enhancement

- Naglašavanje/pojačavanje slike
- Obrada slike tako da je rezultat pogodniji nego originalna slika za **specifičnu aplikaciju**

- **Metodi u prostornom domenu**

$$g(x, y) = T[f(x, y)]$$

- Metodi u frekvencijskom domenu

Metodi u prostornom domenu

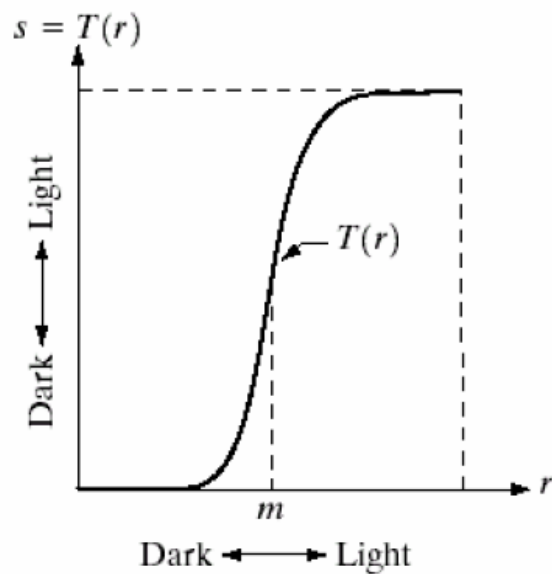
1. Prostorno filtriranje (vidjeli smo ga prošli čas)
2. Transformacije intenziteta (*eng. gray-level transformation ili point processing*)

Transformacije intenziteta

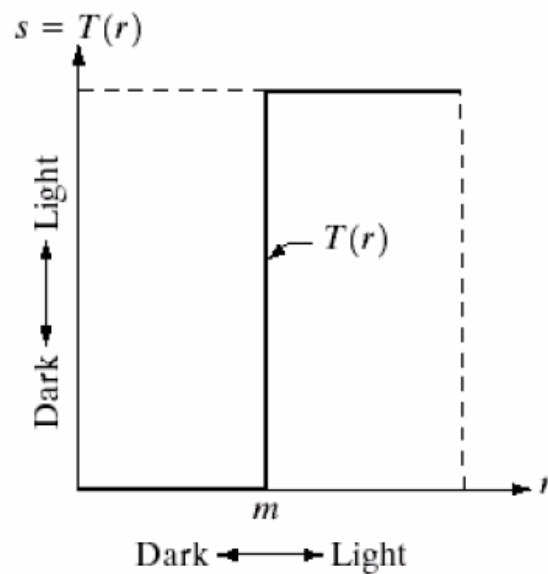
$$s = T(r)$$

- Nova vrijednost piksela se računa **samo** na osnovu njegove vrijednosti, susjedi ne učestvuju
 - Stoga ime *point processing*
- r je vrijednost piksela na poziciji (x,y) u originalnoj slici $f(x,y)$
 - dakle r je vrijednost **prije** obrade (transformacije T)
$$r = f(x,y)$$
- s je vrijednost istog piksela u obrađenoj slici $g(x,y)$ – dakle s je vrijednost **poslije** obrade (transformacije T)
$$s = g(x,y)$$

Primjer: naglašavanje kontrasta



Contrast Stretching



Thresholding

a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

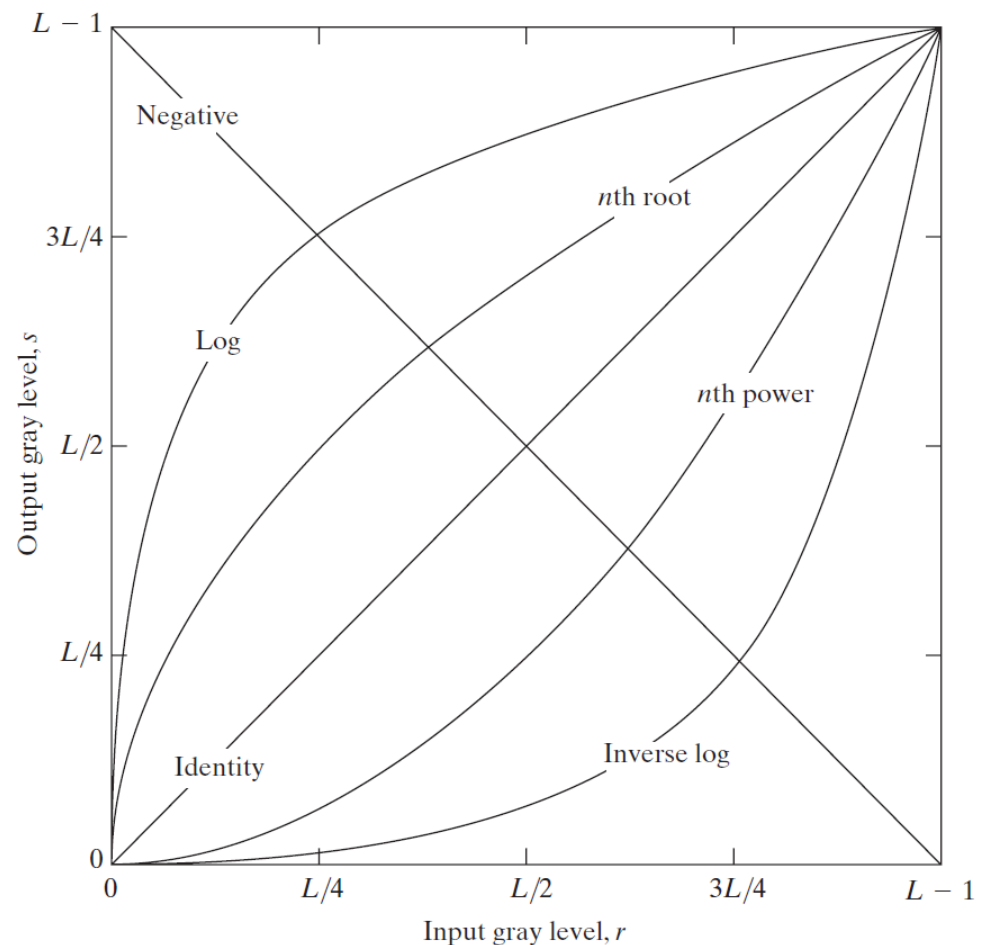
Transformacije intenziteta

- Implementiraju se pomoću look-up tabela (LUT)
- Za 8-bitnu sliku, LUT je 1-D niz koji će sadržati 256 vrijednosti

Transformacije intenziteta

- 3 osnovna tipa transformacija

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



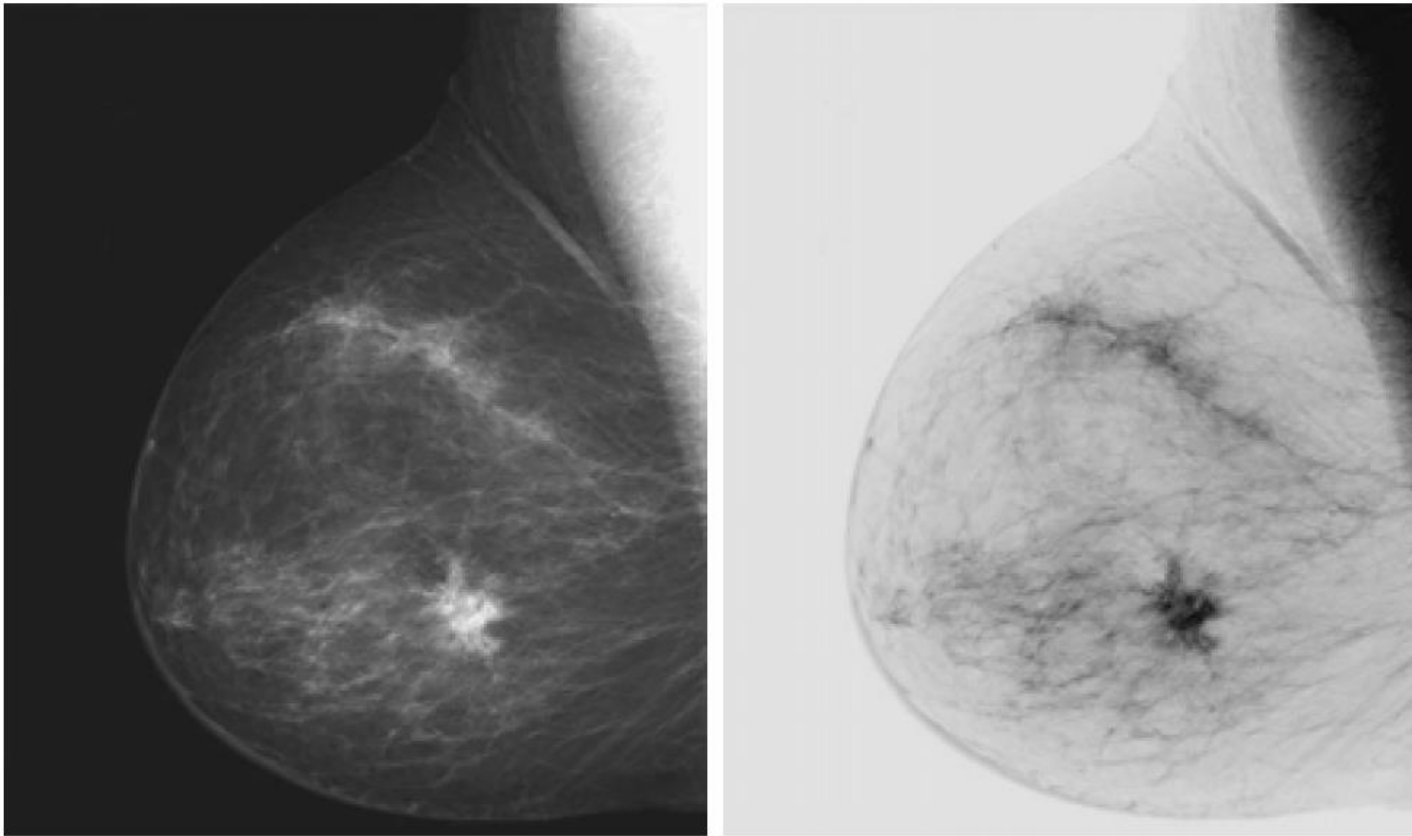
1. Linearne: identitet, negativ
2. Logaritamske: log i inverzna
3. Stepen i korijen konstantnom

Negativ

- Efekat: naglašavanje svjetlijih (bijelih, sivih) regiona „uronjenih“ u tamne regione
 - posebno značajno kada su tamni regioni znatno veće veličine od svjetlijih
 - Koristi se puno u domenu *medical imaging*
- Ako su intenziteti u originalnoj slici u opsegu $[0, L-1]$ (npr 0 - 255) negativ slike se dobija transformacijom

$$s = L - 1 - r$$

Negativ - primjer



a b

FIGURE 3.4

(a) Original digital mammogram.

(b) Negative image obtained using the negative transformation in Eq. (3.2-1).

(Courtesy of G.E. Medical Systems.)

Logaritamske transformacije

- Mapira **uzak** opseg niskih (tamnih) intenziteta u ulaznoj slici u **širi** izlazni opseg sivih intenziteta.
- Sa višim (svjetlijim) intenzitetima radi **suprotno** od opisanog.

$$s = c \log(1+r)$$

c: constant

- Koristi se da se proširi opseg u **tamnim** regionima i time poveća kontrast u tim regionima a „na štetu“ **visokih** intenziteta koji se ovom transformacijom kompresuju.
- Inverzna funkcija radi suprotno od opisanog

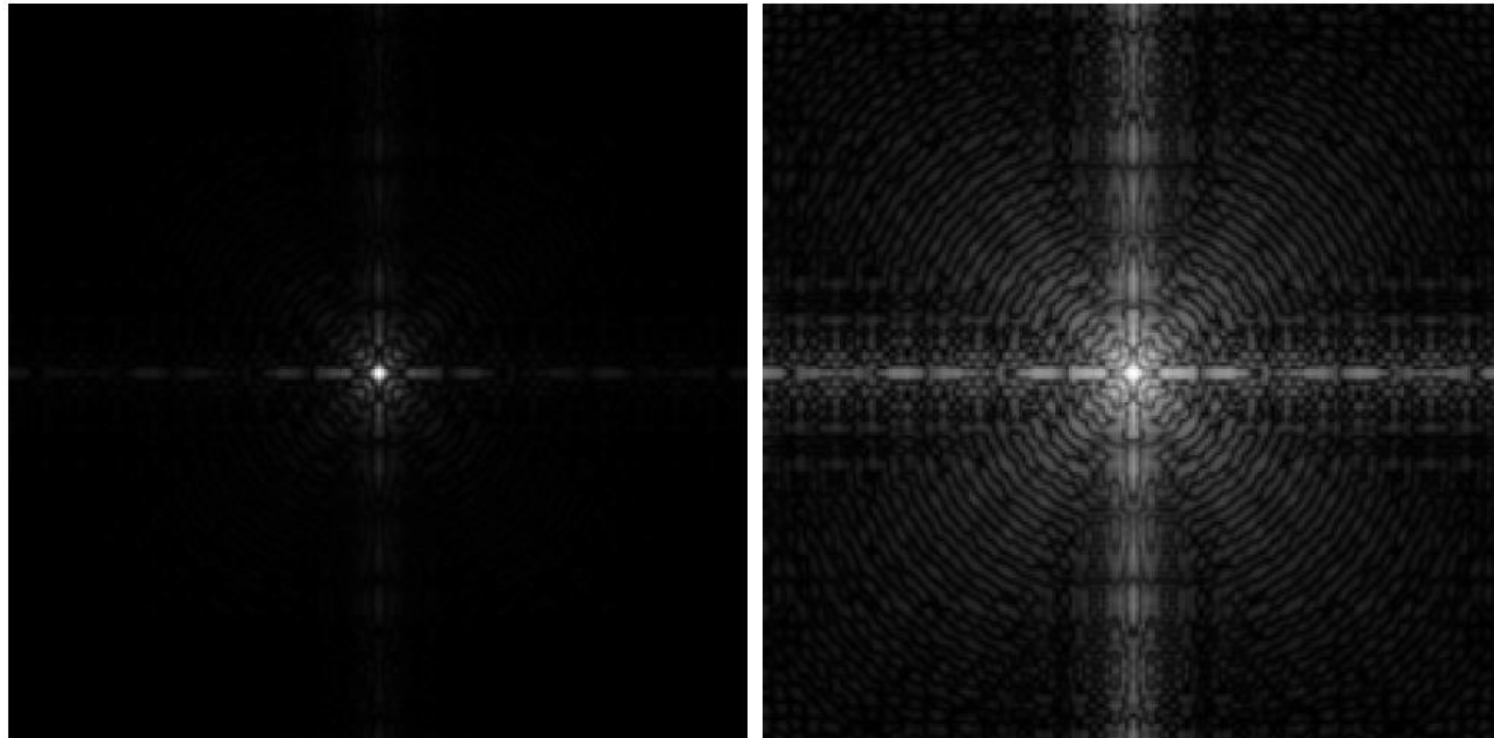
Logaritamske transformacije

- Naglašen kontrast u tamnim regionima

a b

FIGURE 3.5

(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Power-law transformacije

- Stepenovanje konstantom

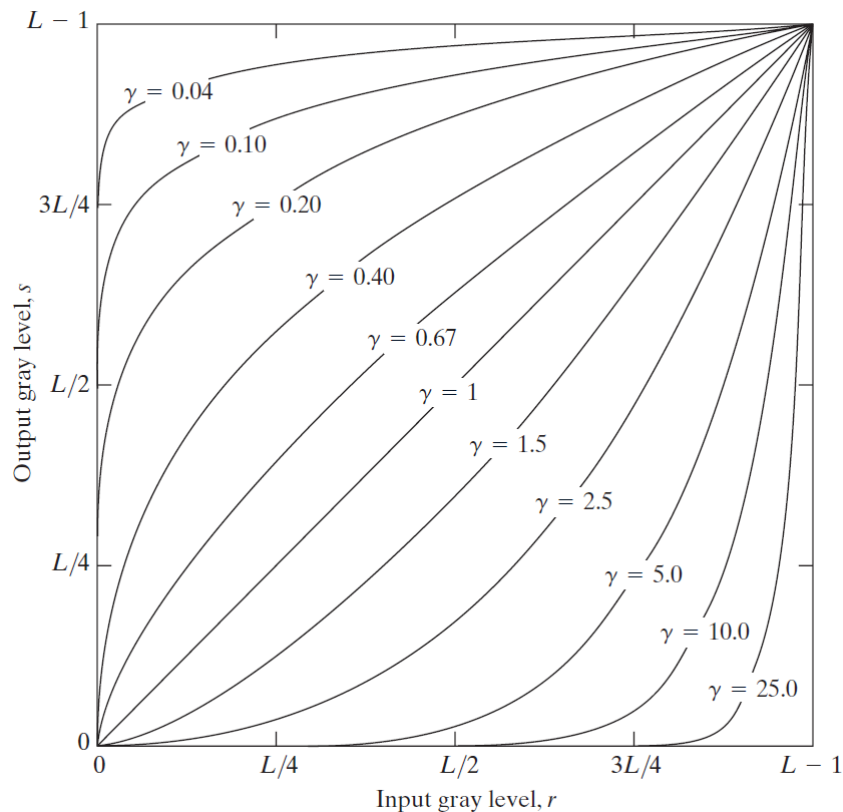


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

$$s = cr^\gamma$$

C, γ : positive constants

Power-law transformacije

- Kada je $\gamma < 1$, efekat je sličan log transformaciji
- Kada je $\gamma > 1$, efekat je suprotan
- γ pruža fleksibilnost za razliku od log

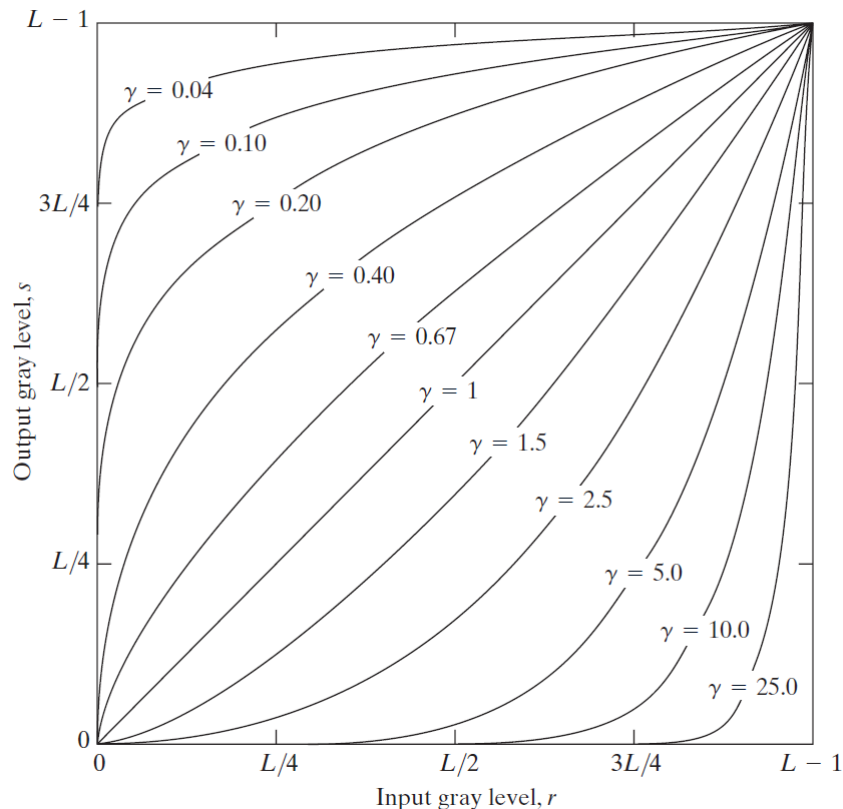


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

$$s = cr^\gamma$$

C, γ : positive constants

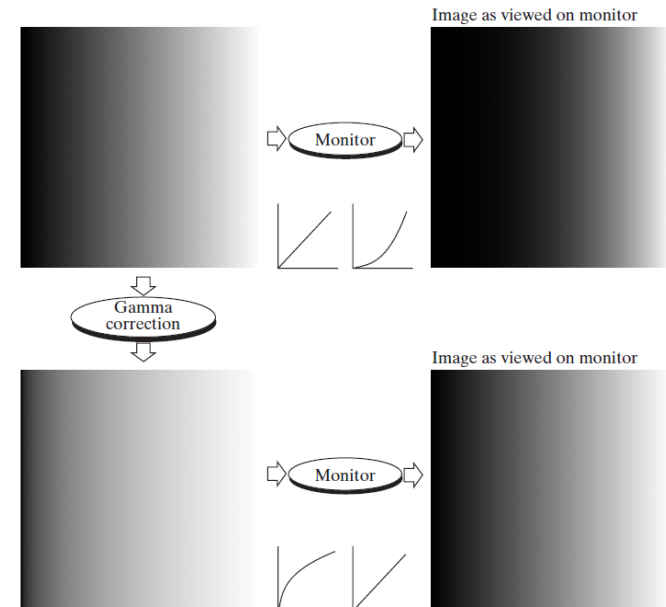
Power-law transformacije

- Jedna primjena:
- Mnogi uređaji za generisanje (capture), štampanje i prikaz slika reaguju tako što transformišu sliku po power law funkciji
 - Ovo je često neželjen efekat
 - Npr CRT uređaji prikazuju sliku tamnijom nego što bi trebalo (često sa $\gamma = 2.5$)
- Proces ispravljanja ovog efekta se zove **Gama korekcija** (u slučaju CRT uređaja korekcija se radi sa $\gamma = 1 / 2.5 = 0.4$)

- Dobra praksa je da se za γ koristi srednja vrijednost svih γ vrijednosti uređaja za koje se očekuje da će prikazivati ili štampati sliku

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



Power-law transformacije

- Još jedna primjena: povećavanje kontrasta

a b
c d

FIGURE 3.9

(a) Aerial image.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and $5.0,$ respectively. (Original image for this example courtesy of NASA.)



Piecewise-linearne transformacije

- Prednost je što možemo imati kompleksne oblike funkcija
- Mana je što za definisanje ovih transformacija je potrebno puno informacija od korisnika

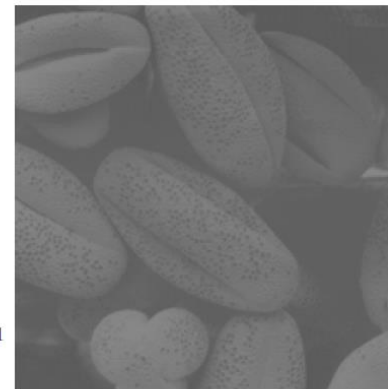
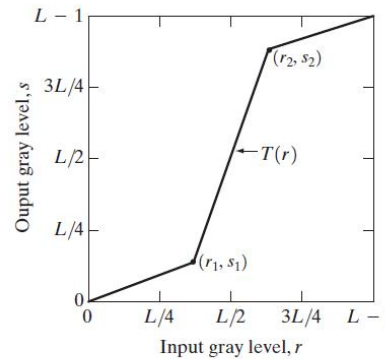
Piecewise-linearne transformacije

- Primjer: Contrast stretching
- Slika polena uvećana 700 puta
- Uzroci niskog kontrasta: senzor sa malim dinamičkim opsegom, slabo svijetlo, loše postavljeno sočiv

- Vrijednosti za r_1 , s_1 , r_2 i s_2 određuju funkciju

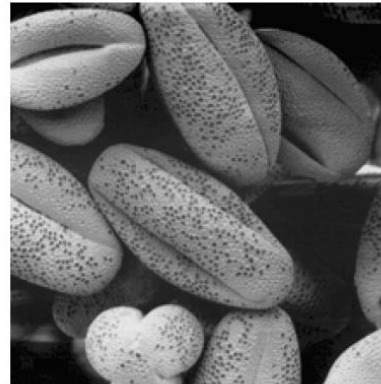
- Granični slučajevi:

1. Ako $r_1 = s_1$ i $r_2 = s_2$, funkcija je
2. Ako $r_1 = r_1$ i $s_1 = 0$ i $s_2 = L-1$



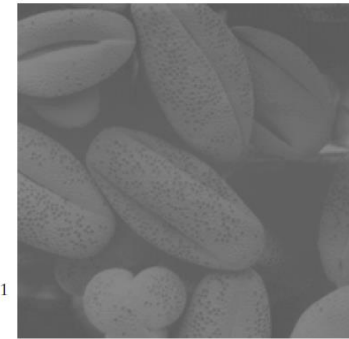
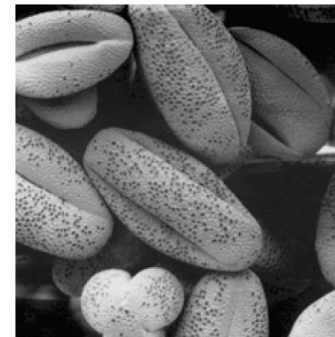
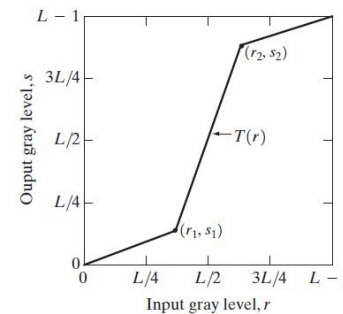
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



Piecewise-linearne transformacije

- Primjer: Contrast stretching
- Vrijednosti za r_1 , s_1 , r_2 i s_2 određuju funkciju
- Granični slučajevi:
 1. Ako $r_1 = s_1$ i $r_2 = s_2$, funkcija je linearna, nema promjene intenziteta
 2. Ako $r_1 = r_1$ i $s_1 = 0$ i $s_2 = L-1$, funkcija postaje thresholding funkcija a izlazna slika je binarna
 3. Srednje vrijednosti ovih parametara generišu transformacije različitih nivoa proširenosti opsega pa time i kontrasta
- Jedino ograničenje: $r_1 \leq r_2$ i $s_1 \leq s_2$

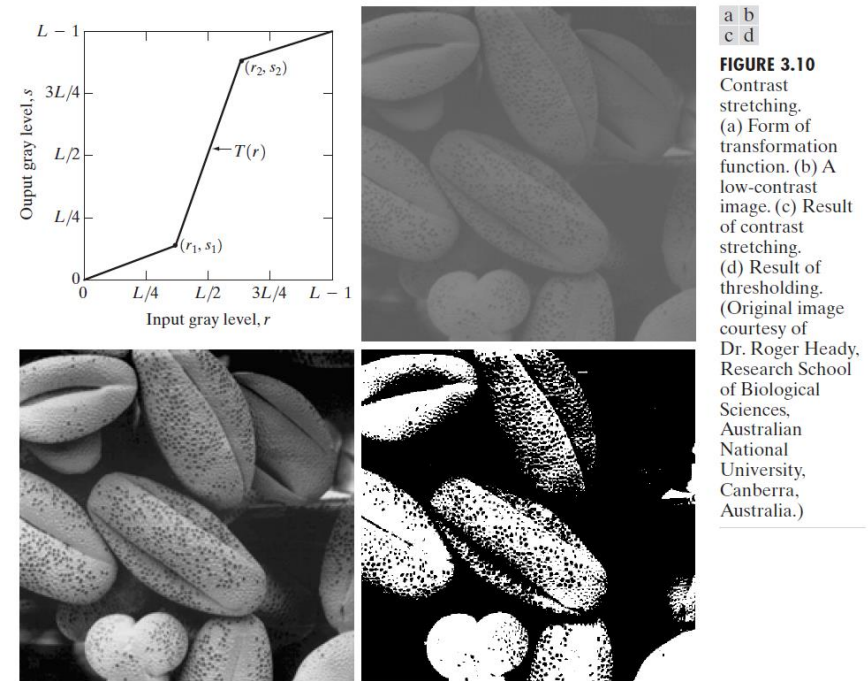


a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

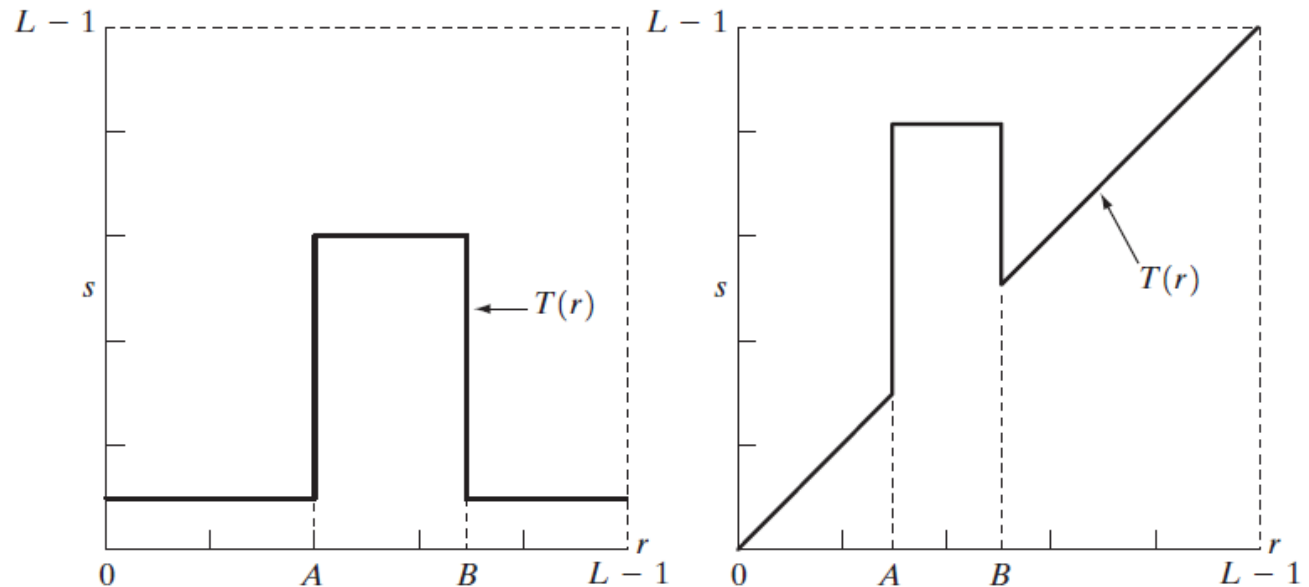
Piecewise-linearne transformacije

- Primjer: Contrast stretching
- Rezultat pod c) dobijen parametrima: $(r_1, s_1) = (r_{\min}, 0)$ i $(r_2, s_2) = (r_{\max}, L-1)$
 - Cijeli opseg slike razvučen na cijeli 8-bitni opseg
- Rezultat pod d) dobijen binarizacijom $r_1 = r_2 = m$
 - m = srednja vrijednost intenziteta u originalnoj slici



Piecewise-linearne transformacije

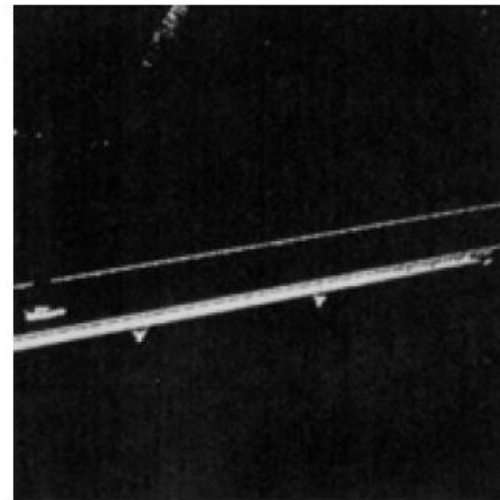
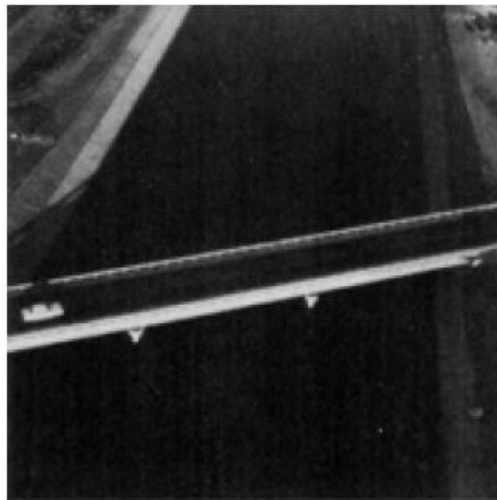
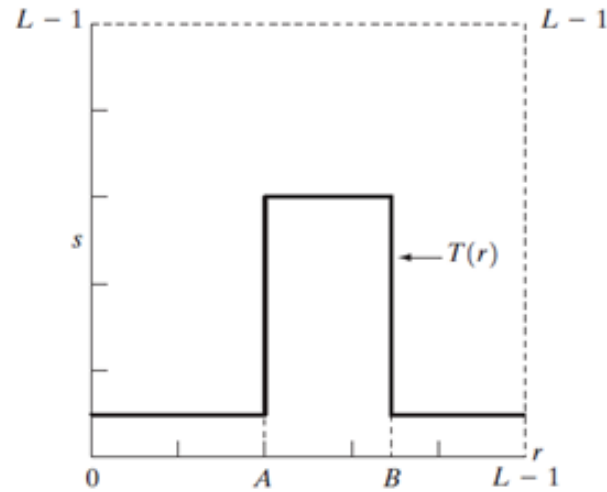
- Primjer: Gray-level slicing
- Naglašavanje određenog opsega i samo njega
- Primjene:
 - naglašavanje regiona, npr. masa vode u satelitskim snimcima
 - naglašavanje mana in X-ray slikama
- 2 pristupa:



- Kakvu sliku proizvodi prvi pristup a kakvu drugi ?

Piecewise-linearne transformacije

- Efekat



Vježba 3

- Učitati slike slabog kontrasta
- Ispitati i testirati i uporediti funkcije za naglašavanje kontrasta koje nude biblioteke OpenCV i scikit-image
- Implementirati funkcije koje nisu ugrađene u biblioteku
- Bonus: generisati RGBXYZ i RGBD informaciju iz inputa senzora i sačuvati jedan screenshot na disk u odgovarajućem formatu (Open3D)