

Seminarski rad

Image Halftoning (polutoniranje slike)

Sadržaj

- Uvod
 - Konstantni prag
 - Binarizacija slučajnim šumom
- Određeno zamućivanje (*Ordered dither*)
 - Prikaz pomoću grozda tački (*Clustered dot screens*)
 - Prikaz pomoću raspršivanja tački (*Dispersed dot screens*)
- Greška širenja (*Error diffusion*)

Digitalno polutoniranje

- Halftoning -polutoniranje, dvonivoska kvantizacija slike, furniranje slike crno bijelim elementima
- Mnoge tehnologije za renderisanje slike imaju samo binarni izlaz. Na primjer, štampači mogu "isprintati tačku" ili ne.
- Halftoning je metoda za kreiranje iluzije o neprekidnosti nijanse sive boje na izlaznom binarnom uređaju.
- Efikasan digitalni halftoning može znatno poboljšati kvalitet renderisane slike po minimalnoj cijeni

Prag

- Pretpostavimo da pikseli sa slike padaju u rangu 0-255.
 - Primjenimo prostorno promjenjiv prag, $T(i, j)$.
- $$b(i, j) = \begin{cases} 255, & \text{ako je } X(i, j) > T(i, j) \\ 0, & \text{u suprotnom} \end{cases}$$
- Šta je $X(i, j)$?
 - Svjetlost - veće => svjetlije
- koristi se za prikaz
 - Koeficijent apsorpcije - veća vrijednost => tamnije
- koristi se za printanje
 - $X(i, j)$ će generalno biti u jedinicama svjetlosti

Konstantni prag

- Pretpostavimo da pikseli sa slike padaju u rangu 0-255.
- $0 \Rightarrow$ Crna i $255 \Rightarrow$ Bijela
- Kvantizator minimalne kvadratne greške je jednostavan prag

$$b(i, j) = \begin{cases} 255, & \text{ako je } X(i, j) > T \\ 0, & \text{u suprotnom} \end{cases}$$

gdje je $T=127$.

- Ovo proizvodi loš kvalitet renderisanja neprekidnog tonaliteta slike.

Konstantni prag - primjer

- Postavljamo prag za svaki piksel
 - piksel ≤ 127 pucamo tintu
 - piksel > 127 ne radimo ništa

POGLEDATI KOD
ISKUCAN U MATLAB-U
U SEMINARSKOM
RADU

Originalna slika



Thresholded slika



Binarizacija slučajnim šumom

- Svakom pikselu na sivoj slici $f(i, j)$, dodaje se slučajan broj n u rangu $[-A, A]$
- Pogledati kod u dokumentu (seminarski rad)

Originalna slika

Binarizacija slučajnim šumom

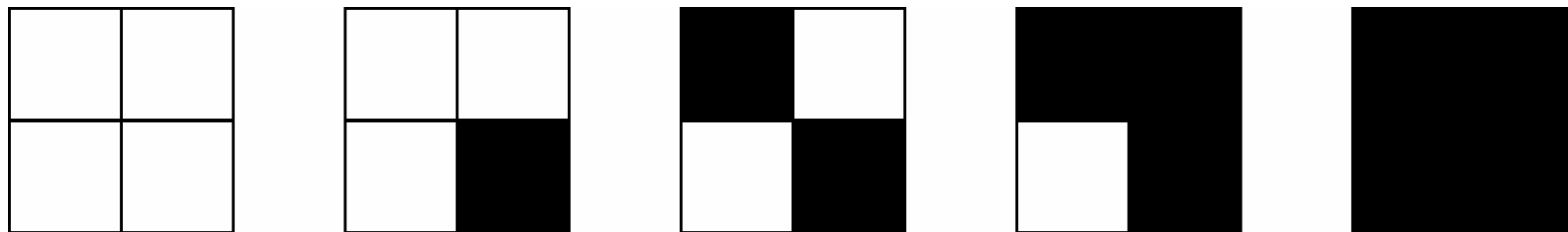


Određeno zamućivanje

- Za konstantni nivo sivog poboljšanja, uključimo piksel "on" u određenom redu.
- Ovo proizvodi dojam o neprekidnoj promjeni sive boje.
- $N \times N$ indeks matrica predviđa kojim redom ćemo piksele koristiti.

$$I_2(i, j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

- Pikseli se uključuju po sljedećem poretku



Primjena određenog zamuć. preko praga

- Indeks matrica može biti konvertovana u "prag matricu" ili "prikaz" korištenjem sljedeće operacije:

$$T(i, j) = 255 \frac{I(i, j) + 0,5}{N^2}$$

- $N \times N$ matrica tada može biti "popločana" preko slike koristeći periodične zamjene.

$$T((i \bmod N)+1, (j \bmod N) +1)$$

- Određen dither algoritam je tad primjenjiv preko praga.

$$b(i, j) = \begin{cases} 255, & \text{ako je } X(i, j) > T((i \bmod N)+1, (j \bmod N)+1) \\ 0, & \text{u suprotnom} \end{cases}$$

Prikaz pomoću grozda tački

- Definicija: Ako su uzastopni pragovi locirani u prostornoj blizini, tada ovo zovemo "klaster tačkasti prikaz".
- Primjer za 8×8 matricu:

62	57	48	36	37	49	58	63
56	47	35	21	22	38	50	59
46	34	20	10	11	23	39	51
33	19	9	3	0	4	12	24
32	18	8	2	1	5	13	25
45	31	17	7	6	14	26	40
55	44	30	16	15	27	41	52
61	54	43	29	28	42	53	60

Prikaz pomoću grozda tački

			3	0			
			2	1			

			21	22				
			20	10	11	23		
		19	9	3	0	4	12	24
32	18	8	2	1	5	13	25	
		31	17	7	6	14	26	
			30	16	15	27		
				29	28			

			10	11			
			9	3	0	4	12
			8	2	1	5	13
				7	6	14	
				16	15		

			48	36	37	49		
			47	35	21	22	38	50
46	34	20	10	11	23	39	51	
33	19	9	3	0	4	12	24	
32	18	8	2	1	5	13	25	
45	31	17	7	6	14	26	40	
			44	30	16	15	27	41
				43	29	28	42	

Prikaz pomoću grozda tački - primjer

- Pogledati kod u seminarskom radu
- Primjer za 2×2 i 4×4 matricu:

Prikaz veličine 2×2



Prikaz veličine 4×4

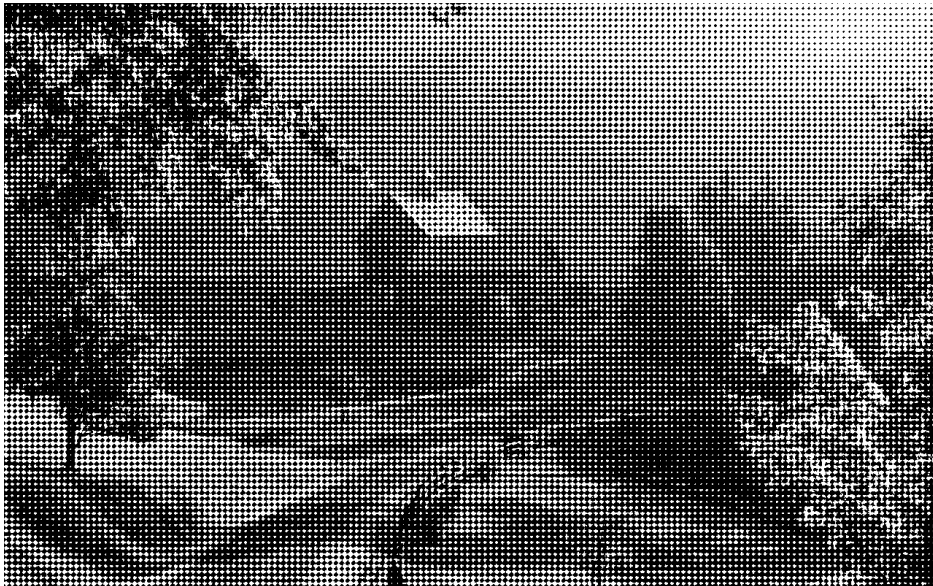


- Redom podržavaju samo 5 i 17 nivoa sive

Prikaz pomoću grozda tački - primjer

- Pogledati kod u seminarskom radu
- Primjer za 8×8 i 16×16 matricu:

Prikaz veličine 8×8



Prikaz veličine 16×16



- Redom podržavaju 65 i 257 nivoa sive

Osobine Prikaza pomoću grozda tački

- Zahtijeva razmjenu između broja sivih nivoa i rezolucije
- Relativno vidljive teksture
- Relativno slabi detalji interpretacije slike
- Uniformne teksture preko čitave sive skale
- Snažna izvedba na ne-idealnom izlaznom uređaju
 - ne postoji preklapanje tački
 - mrlja u mrlju promjenjivost
 - šum

Prikaz pomoću raspršivanja tački

- Bayer optimalne index matrice su definisane rekurzivno pomoću formule:

$$I_2(i, j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \quad I_{2^n}(i, j) = \begin{bmatrix} 4 * I_n + 1 & 4 * I_n + 2 \\ 4 * I_n + 3 & 4 * I_n \end{bmatrix}$$

- Primjeri za 2×2, 4×4 i 8×8 :

1	2
3	0

5	9	6	10
13	1	14	2
7	11	4	8
15	3	12	0

21	37	25	41	22	38	26	42
53	5	57	9	54	6	58	10
29	45	17	33	30	46	18	34
61	13	49	1	62	14	50	2
23	39	27	43	20	36	24	40
55	7	59	11	52	4	56	8
31	47	19	35	28	44	16	32
63	15	51	3	60	12	48	0

- Proizvodi ljepšu kvantizaciju nad većim površinama
- Zadržava dobro izvođenje detalja na malim površinama

Prikaz pomoću raspršivanja tački - primjer

- Pogledati kod u seminarskom radu
- Bayer prikaz pomoću indeks matrica veličina 2×2 i 4×4

Prik. uz ind. matr. vel. 2×2

Prik. uz ind. matr. vel. 4×4



- Ponovo, imamo samo 5 i 65 nivoa sive nijanse

Prikaz pomoću raspršivanja tački - primjer

- Pogledati kod u seminarskom radu
- Bayer prikaz pomoću indeks matrica veličina 8×8 i 16×16



- Ova dva slučaja se ne razlikuju puno jedno od drugog
- Nema razmjene između rezolucije i broja sivih nijansi

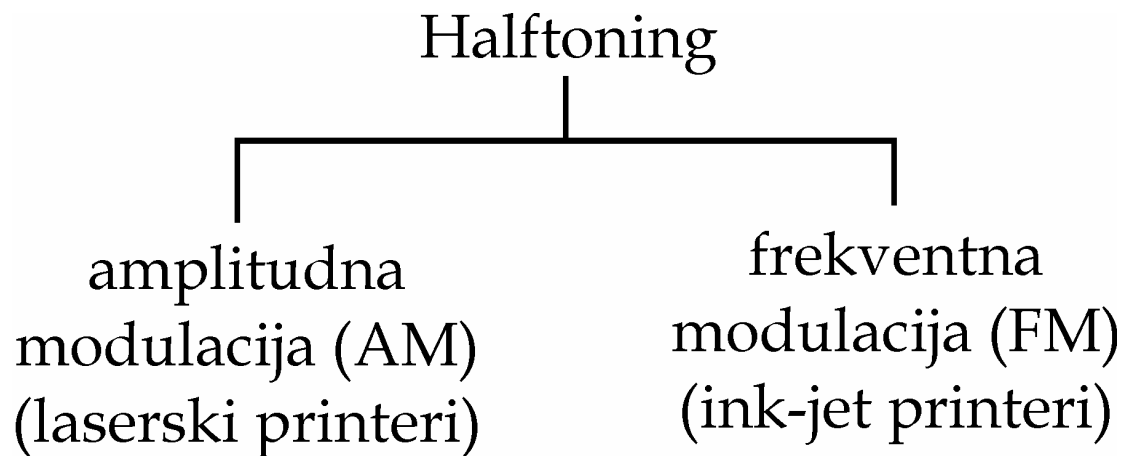
Osobine prikaza pomoću raspršivanja tački

- Uklanja razmjenu između nivoa sivih nijansi i rezolucije
- Unutrašnjost bilo kojeg područja koje sadrži K tački, K prag treba biti distribuiran uniformno što je moguće tačnije
- Teksture koje predstavljaju pojedinu nijansu sive imaju slabu vidljivost
- Poboľjšano je tumačenje detalja
- Prijelaz između tekstura koje odgovaraju različitim nijansama sive su obično vidljivije

Osobine prikaza pomoću raspršivanja tački

- Tehnika nije puno snažna na ne-idealnim izlaznim uređajima
 - zahtjeva stabilno formiranje izoliranih tački.

- Napomena:

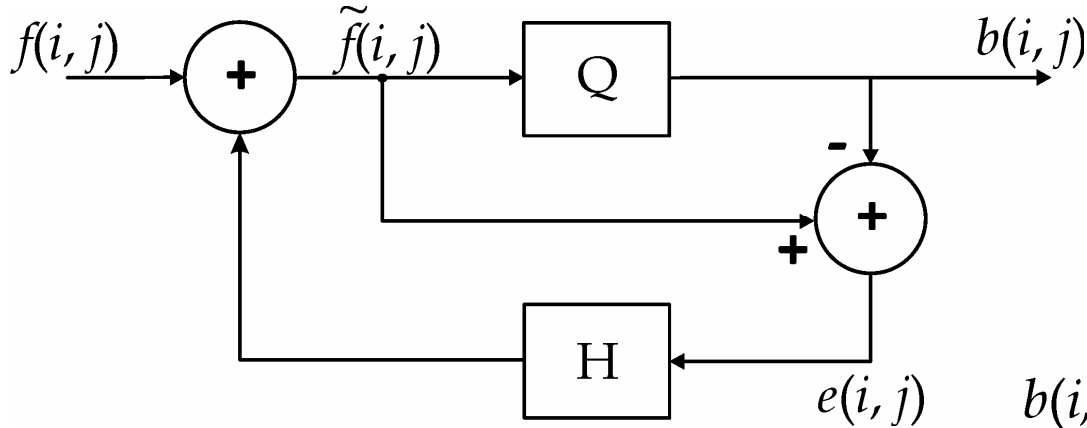


- Cluster dot screening (AM)
- Dispersed dot screening (FM)

Greška širenja

- Kvantizira svaki piksel koristeći češće operacije susjedstva, nego korištenja jednostavnih tačkastih operacija
- Prolazi kroz sliku u rastr uređenju, kvantizirajući rezultat, i gurajući grešku naprijed
- Može proizvesti puno kvalitetnije slike nego što je to moguće sa prikazima (tehnikom određenog zamučivanja)

Šematski prikaz filtera greške širenja



$$e(i, j) = \tilde{f}(i, j) - b(i, j)$$

$$b(i, j) = \begin{cases} 255, & \text{ako je } \tilde{f}(i, j) > T \\ 0, & \text{u suprotnom} \end{cases}$$

- Parametri:

$$\tilde{f}(i, j) = f(i, j) + \sum_{k, l \in S} h(k, l) e(i-k, j-l)$$

- prag je obično $T = 127$.

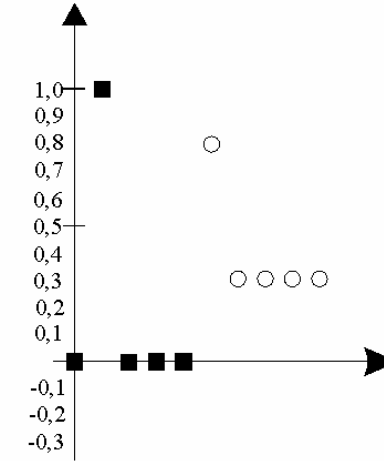
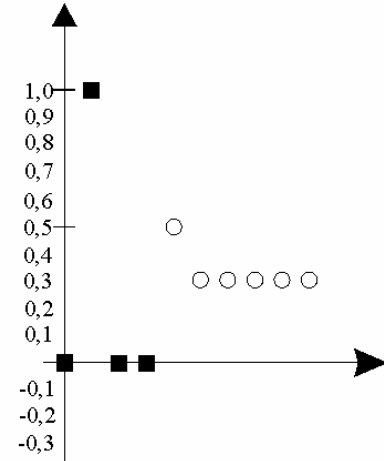
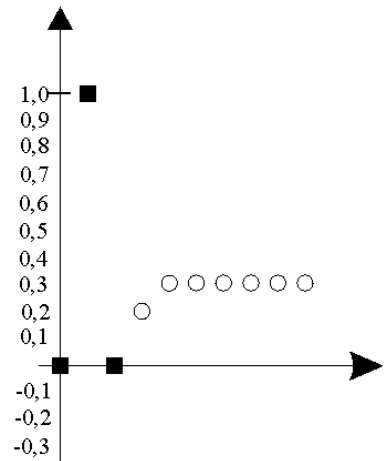
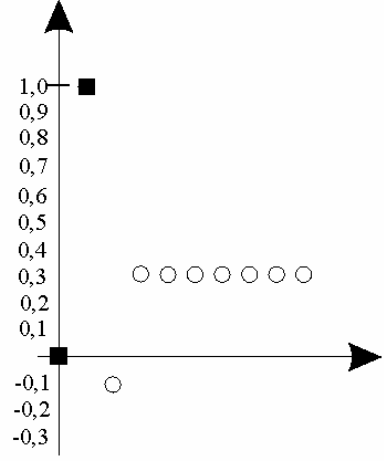
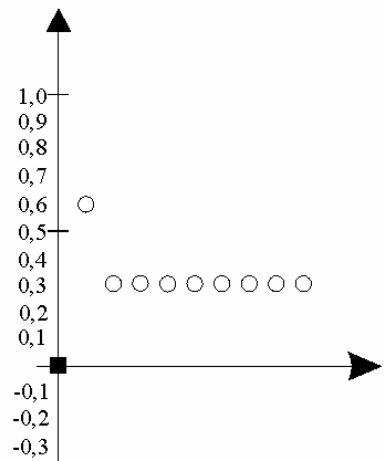
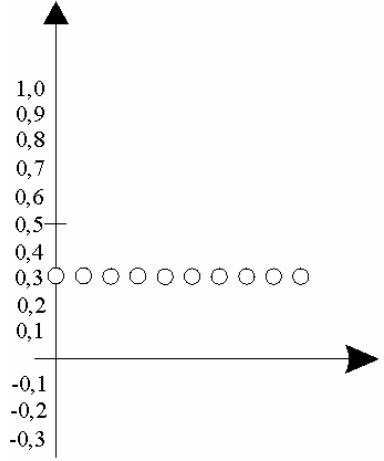
- $h(k, l)$ tipično je izabrano da bude pozitivno i da mu suma po k i l -ovima iznosi 1

Primjer greške širenja za 1-D

● $\tilde{f}(i, j)$
krugovi

● $b(i, j)$
kvadrati

● $T = 0.5$



Dva pogleda na grešku širenja

- Dva matematički jednaka pogleda na grešku širenja:
 - vađenje greške sprijeda
 - guranje greške naprijed
- Vađenje greške sprijeda
 - ima prednosti za analizu
- Guranje greške naprijed
 - originalni pogled na grešku širenja
 - lakše može biti produžen na važne slučajeve kad težina površine varira u zavisnosti od vremena/prostora

Vađenje greške sprijeda

1. Za svaki piksel na slici (u raster poretku)

(a) Izvadi grešku sprijeda
$$\tilde{f}(i, j) = f(i, j) + \sum_{k, l \in S} h(k, l) e(i-k, j-l)$$

(b) Izračunaj binarni izlaz

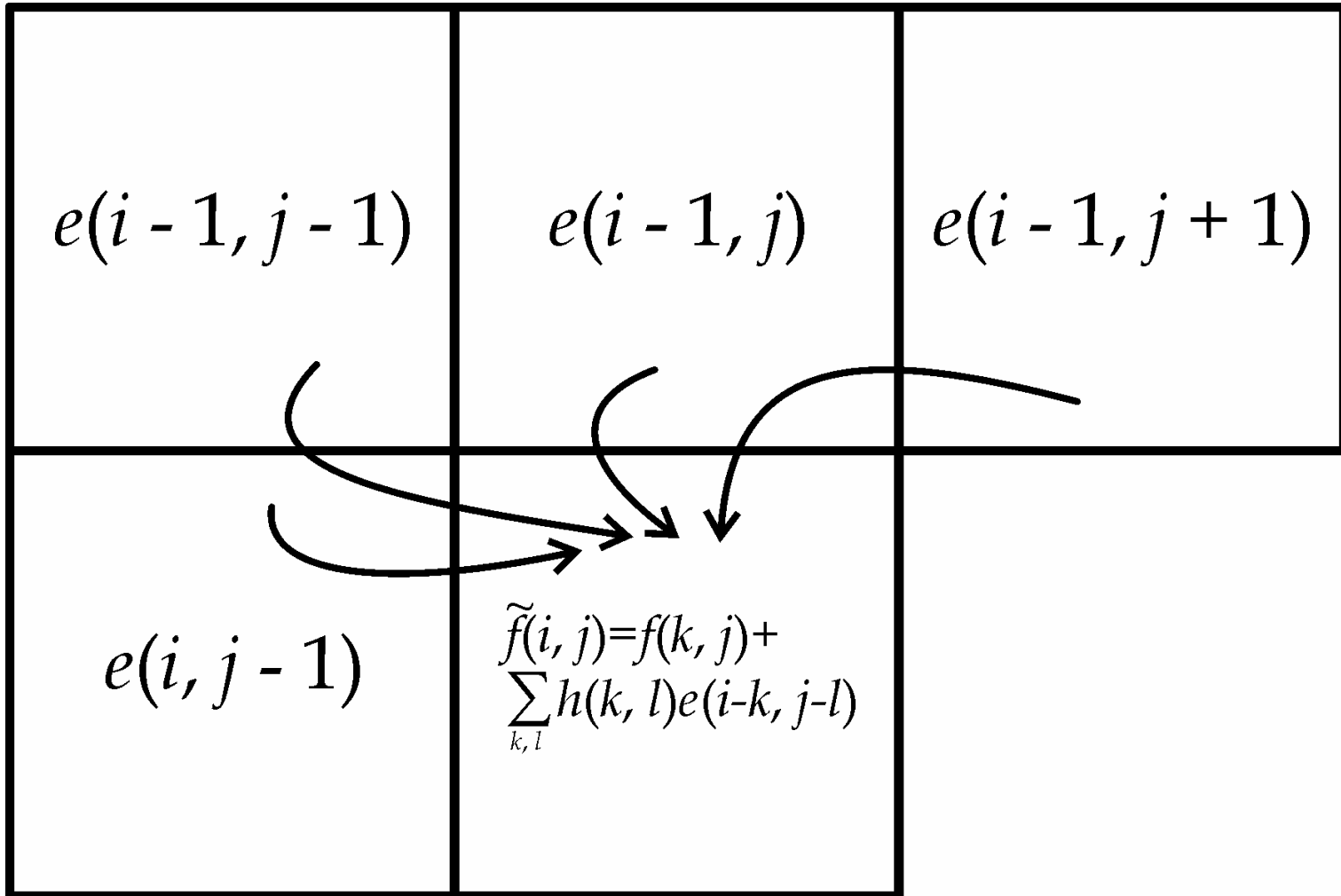
$$b(i, j) = \begin{cases} 255, & \text{ako je } \tilde{f}(i, j) > T \\ 0, & \text{u suprotnom} \end{cases}$$

(c) Izračunaj grešku piksela

$$e(i, j) = \tilde{f}(i, j) - b(i, j)$$

2. Prikaži binarnu sliku $b(i, j)$

Vađenje greške sprijeda



Guranje greške naprijed

1. Inicijaliziraj $\tilde{f}(i, j) \leftarrow f(i, j)$
2. Za svaki piksel na slici (u raster poretku)

(a) Izračunajte

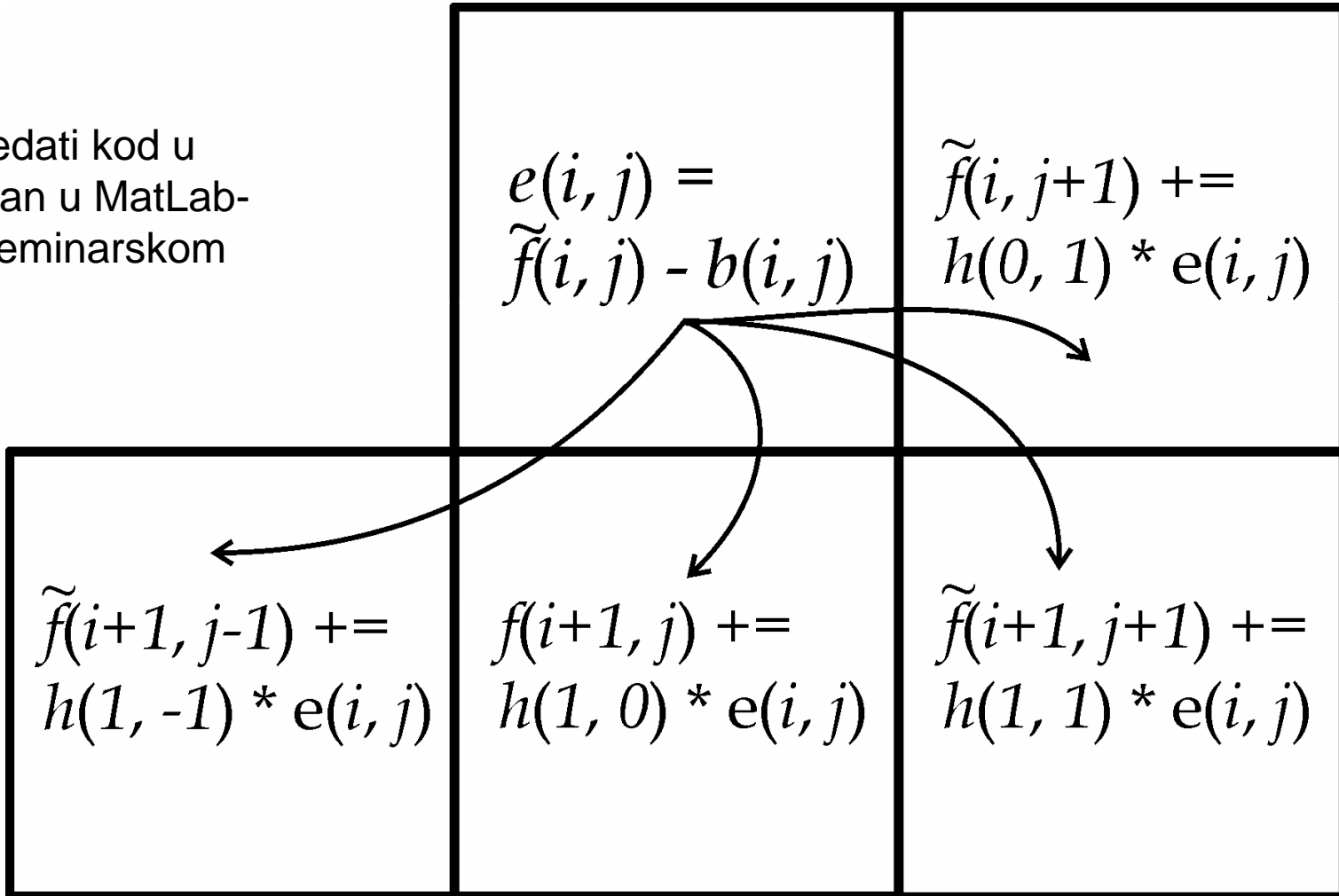
$$b(i, j) = \begin{cases} 255, & \text{ako je } \tilde{f}(i, j) > T \\ 0, & \text{u suprotnom} \end{cases}$$

(c) Raširite grešku naprijed koristeći šemu prikazanu na sljedećem slajdu

3. Prikaži binarnu sliku $b(i, j)$

Guranje greške naprijed

Pogledati kod u
 iskucan u MatLab-
 u u seminarskom
 radu



Često korištene težine za grešku širenja

- Floyd i Steinberg (1976)

		7/16
3/16	5/16	1/16

- Jarvis, Judice and Ninke (1976)

			7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

Često korištene težine za grešku širenja

- Pikseli obrađeni pomoću "greške širenja" i rezultirajuća kvantizirana slika

Originalna slika



Floyd i Steinberg-ova sl. gr.šir.



Literatura

- [1] <https://engineering.purdue.edu/~bouman/ece637/>
<https://engineering.purdue.edu/~bouman/ece637/lectures07/>
Video predavanja pod rednim brojem 36, 37 i 38
<https://engineering.purdue.edu/~bouman/grad-labs/lab7/>
<https://engineering.purdue.edu/~bouman/ece637/notes/>
- [2] R. C. Gonzales, R. E. Woods, S. L. Eddins, "Digital Image Processing Using Matlab", Pearson 2004, str. 108-143