

1. Pseudocolorarea este o tehnica punctuala de imbunatatire prin care se pun in evidenta, prin culori diferite, anumite detalii/obiecte, astfel schimbându-se harta originala a imaginii. Se renunta la tabela de culoare standard, se fac modificari la anumite intrari si le fac colorate astfel incat unele componente din imagine se vor vedea mai bine, fiind colorate. Orice pata de culoare pe imagine cu niveluri de gri are o vizibilitate mai buna.

Daca avem nevoie sa identificam o anumita gama de gri cu ochiu liber, este aproape imposibil. Insa, daca modific tabela initiala de niveluri de gri si modific aduc pixelii din acel interval la o anumita culoare, se va putea identifica mai usor acea portiune din imagine.

Prin urmare, nu se va mai mai folosi un tabel de culoare care va folosi doar nuante de gri, ci modific valorile de gri conform unor reguli impuse de aplicatia pe care o utilizam.

Ochiului uman ii este mult mai usor sa distinga culori diferite, fata de niveluri de gri diferite. Astfel, daca dorim sa evidentiem niste obiecte dintr-o imagine, folosim tehnica pseudocolorarii.

Cu ajutorul pseudocolorarii, putem modifica in cateva linii de cod componenta imaginii initiale, oferind niste efecte vizibile mari. Insa, in cele mai multe cazuri, desi schimbarea e evidenta, nu inseamna neaparat ca aceasta e necesara aplicatiei pe care o utilizam.

Aceste afirmatii sunt facute de catre Kenneth Castleman in cartea sa Digital Image Processing. Acesta afirma faptul ca, daca ar fi sa analizeze aplicatiile puse la dispozitie, foarte putine dintre ele necesita tehnica de pseudocolorare.

Eu gasesc o utilizare eficienta a pseudocolorarii in aplicatiile de imagistica medicala. Imagistica medicala este o tehnica ce face posibila vizualizarea unor parti ale corpului uman ce ajuta la analiza in detaliu a acestora. Este extrem de utila in cazul in care medicul doreste sa vizualizeze o anumita zona si sa puna un diagnostic. In imagistica medicala moderna, se gasesc multe echipamente ce fac posibila producerea de imagini medicale concrete. Cu ajutorul acestor tehnici moderne se pot pune diagnostice intr-un timp scurt.

Exista mai multe aplicatii medicale ce folosesc pseudocolorarea cu ajutorul unor computere si a unor echipamente moderne, ce stocheaza imaginea medicala, care ulterior este analizata de specialisti. In unele din aceste echipamente moderne se regasesc CT-urile, RMN-urile, PET-urile etc.

In cazul pseudocolorarii in domeniul medicinei, aceasta este folosita atunci cand imaginea cu o anumita zona anatomica este stocata pe un computer, iar cu ajutorul unuia dintre instrumentele mentionate mai sus, este vizualizata in amanunt. Pseudocolorarea ajuta la vizionarea mai rapida si mai usoara a unor zone de interes pentru cadrul medical, iar acesta poate sa isi faca o impresie dupa aceste analize amanuntite. Fiecare dispozitiv are un scop diferit, iar pacientul este indrumat sa ingereze anumite lichide preparate in laborator (si recomandate de medicul specialist), care se numesc substante de contrast, astfel incat atunci cand este supus analizelor sa se observe zonele de interes ale medicului care il are pe pacient sub observatie.

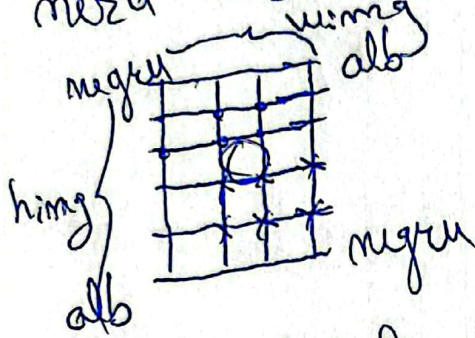
2. Din punct de vedere theoretic, histogramele acestor doua imagini ar trebui sa fie egale pentru ca au aceleasi 2 valori in componenta lor, ce sunt distribuite in mod egal, chiar daca sunt plasate diferit. Histograma este desnitarea de probabilitate a pixelilor din imagine.

2) Vom numi imaginea a1 si b1

Pentru imaginea a1, fereastra de mediere poate ge-

nera $W \cdot [H/2]$ valori

Numarul de valori e egal cu
numarul de puncte astfel incat
se genereaza o valoare noua.



Valorile cele mai apropiate de centrul histo-
gramii cu L valori de gri intre care nu exista
alte valori noi sunt:

$$q_1 = \lfloor W \cdot H/2 \rfloor \text{ si } q_2 = \lfloor W \cdot H/2 + 1 \rfloor$$

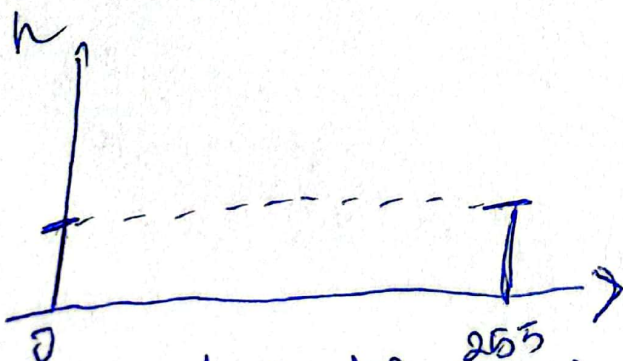
Nr. de pixeli negri din fereastra de mediere arit-
metica

$$\text{Nr pixeli negri} = j \cdot i + (H-i)(W-j)$$

Nr. pixeli albi din fereastra de mediere aritmetica
se respecta cazel :

$$\text{Nr. pixeli albi} = i(W-j) + j(H-i)$$

Histograma dupa mediere va fi armatata ca
cu histograma din b1.



Noile valori diferite de 0 in cazul imaginii b1

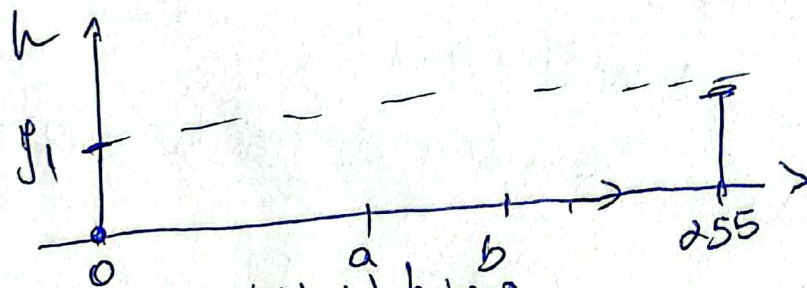
pentru numărul de linii H .

$$f_{MA} = \frac{1}{W-H} \left[\begin{matrix} 1 & 1 & \dots & 1 \\ \vdots & & & \\ 1 & \dots & \dots & 1 \end{matrix} \right] \cdot H$$

$$\frac{1}{W \cdot H} (255 \cdot H \cdot \lceil W/2 \rceil + 0 \cdot H \cdot (\lceil W/2 \rceil + 1)) = \frac{1}{W} (255 \cdot \lceil W/2 \rceil + 1)$$

$$\frac{1}{W \cdot H} (\lceil W/2 \rceil + 1) + 0 \cdot \lceil W/2 \rceil$$

$$\frac{1}{W} [255 \cdot i + 0 \cdot (W-1)] \cdot H; \quad i = 1, W-1 \quad \text{pt } [a] \leq a$$



$$g_1 = \frac{0,5 \cdot (W-1) \cdot \text{himg}}{\text{himg} \cdot W \cdot \text{himg}} \quad \text{valoare maximă}$$

$$a = \frac{1}{W} (255 \cdot \lceil W/2 \rceil + 0 \cdot (\lceil W/2 \rceil + 1))$$

$$b = \frac{1}{W} (255 \cdot (\lceil W/2 \rceil + 1) + 0 \cdot \lceil W/2 \rceil)$$

Pentru imaginea b

$$h(i) = \frac{\text{himg} - \lceil H/2 \rceil}{W \cdot \text{himg}} \quad \text{fără bordare și } \frac{\text{himg}}{W \cdot \text{himg} \cdot \text{himg}}$$

cu bordare, dacă i e regăsit în noua histogramă

Pentru imaginea a

$$h(i) = \frac{2H}{W \cdot \text{himg} \cdot \text{himg}} \quad \text{cu bordare}$$

3. Filtrare liniară:

$$g(l, c) = \sum_{(m,n) \in V} w_{m,n} f(m+l, n+c)$$

$$w_{m,n} \in \mathbb{R}$$

Se presupune:

$$\sum_{(m,n) \in V} w_{m,n} = 1; w_{m,n} > 0.$$

> borduri

$$\begin{pmatrix} 112 \\ 114 \end{pmatrix}$$

$$114$$

$$\begin{array}{rrrr} 100 & 20 & 20 & 20 \\ 100 & 100 & 100 & 100 \\ 20 & 20 & 100 & 100 \\ 80 & 80 & 100 & 100 \\ \hline 80 & 80 & 100 & 100 \end{array}$$

$$\sum \text{pixeli} = 840.$$

$$\frac{220}{h}; \frac{160}{h}; \frac{160}{h}; \frac{320}{h}; \frac{320}{h}; \frac{400}{h}; \frac{140}{h}; \frac{220}{h};$$

$$\frac{400}{h}; \frac{320}{h}; \frac{340}{h}; \frac{400}{h} \rightarrow \text{valori după aplicarea}$$

filtrării

$$\sum \text{valori} = 845.$$

$$w_{00} = \frac{1}{2}; w_{10} = \frac{1}{4}; w_{01} = \frac{1}{4}$$

$$g(l, c) = \frac{1}{2} f(l, c) + \frac{1}{4} f(l, c) + \frac{1}{4} f(l, c)$$

```

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Teme facultate\PI\LABORATOR 3\fil_de_netzeire.py

63 sigma = 10
64 N = np.random.normal(0, sigma, (h,w))
65
66 # adun zgomotul in imagine si afisez
67 img_noise = img + N
68 plt.figure(), plt.imshow(img_noise, cmap='gray')
69 plt.show()
70
71 # dim ferestrei si numarul de linii si coloane care se pierd pt fiecare tip de fereastră
72 window_size = 3
73 capat = math.floor(window_size / 2)
74
75 # initializeaza masca de filtrare pentru media ponderata
76 [[0.075,0.124,0.075],[0.124,0.2,0.124],[0.075,0.124,0.075]]
77 mask_ponderat = np.array([[0.075,0.124,0.075],[0.124,0.2,0.124],[0.075,0.124,0.075]])
78
79 # pt aritmetic fereastră e tot 3x3, toti coef sunt egali cu 1/9
80 mask_aritmetic = np.ones((3,3))/9
81
82 b = time.time()
83
84 # aplic functia de filtrare liniara scrisa mai sus pentru cele 2 cazuri (medie arit si ponder)
85 img_filt_ponderat = filt_mediere_arit(img_noise,h,w,caput,mask_ponderat)
86 img_filt_aritmetic = filt_mediere_arit(img_noise,h,w,caput,mask_aritmetic)
87
88 sum1=0
89 sum2=0
90 for i in img:
91     sum1+=sum(i)
92 for j in img_filt_aritmetic:
93     sum2+=sum(j)
94 print('sum1:', sum1)
95 print('sum2:', sum2)
96
97 # MSE între imaginea originală și imaginea zgomotoasă înainte să aplicăm filtrarea
98 # capatul pt ca în cazul imaginii originale nu am pierdut încă linii și coloane, urmează la
99 print('Mse înainte de filtrare: ', MSE(img, img_noise, h, w, 0))
100

```

Name	Type	Size	Value
img_noise	Array of float64	(512, 512)	[34.77647951 ... 92.6 ...]
mask_aritmetic	Array of float64	(3, 3)	[[0.11111111 0.11111111 0.11111111] ...]
mask_ponderat	Array of float64	(3, 3)	[[0.075 0.124 0.075] ...]
N	Array of float64	(512, 512)	[[-7.43419516 -9.79986296 18.18988197 ... -4.87841921] ...]
sigma	int	1	10
sum1	int32	1	55659
sum2	float64	1	55504.39600898845
w	int	1	512
window_size	int	1	3

```

In [6]: runfile('C:/Teme facultate/PI/LABORATOR 3/filt_de_netzeire.py', wdir='C:/Teme facultate/PI/LABORATOR 3')
uint8
sum1: 55659
sum2: 55504.39600898845
Mse înainte de filtrare: 100.42241146397193
Mse pentru filtrarea ponderata inainte de filtrare: 35.285960848007875
Mse pentru filtrarea aritmetica inainte de filtrare: 41.338550467191844
timp executie: 2.9463322162628174

In [7]:

```

```

Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Teme facultate\PI\LABORATOR 3\fil_de_netzeire.py

60 # adunug zgomot in imagine si afisez imaginea zgomotoasa
61 # zgomot de medie 0 dispersie sigma
62 sigma = 10
63 N = np.random.normal(0, sigma, (h,w))
64
65 # adun zgomotul in imagine si afisez
66 img_noise = img + N
67 plt.figure(), plt.imshow(img_noise, cmap='gray')
68 plt.show()
69
70 # dim ferestrei si numarul de linii si coloane care se pierd pt fiecare tip de fereastră
71 window_size = 3
72 capat = math.floor(window_size / 2)
73
74 # initializeaza masca de filtrare pentru media ponderata
75 [[0.075,0.124,0.075],[0.124,0.2,0.124],[0.075,0.124,0.075]]
76 mask_ponderat = np.array([[0.075,0.124,0.075],[0.124,0.2,0.124],[0.075,0.124,0.075]])
77
78 # pt aritmetic fereastră e tot 3x3, toti coef sunt egali cu 1/9
79 mask_aritmetic = np.ones((3,3))/9
80
81 b = time.time()
82
83 # aplic functia de filtrare liniara scrisa mai sus pentru cele 2 cazuri (medie arit si ponder)
84 img_filt_ponderat = filt_mediere_arit(img_noise,h,w,caput,mask_ponderat)
85 img_filt_aritmetic = filt_mediere_arit(img_noise,h,w,caput,mask_aritmetic)
86
87 sum1=0
88 sum2=0
89 for i in img:
90     sum1+=sum(i)
91 for j in img_filt_aritmetic:
92     sum2+=sum(j)
93 print('sum1:', sum1)
94 print('sum2:', sum2)
95
96 sum3=0
97 for k in img_noise:

```

Name	Type	Size	Value
j	Array of float64	(510,)	[41.33909506 41.34213122 44.57964901 ... 91.53261089 94.5379195]
k	Array of float64	(512,)	[46.55850569 34.88500308 54.88140282 ... 94.96419794 97.9463275]
mask_aritmetic	Array of float64	(3, 3)	[[0.11111111 0.11111111 0.11111111] ...]
mask_ponderat	Array of float64	(3, 3)	[[0.075 0.124 0.075] ...]
N	Array of float64	(512, 512)	[[18.1179145 -13.60591048 5.04575898 ... -5.78534891] ...]
sigma	int	1	10
sum1	int32	1	55659
sum2	float64	1	55447.809851223654
sum3	Array of float64	(512,)	[22309216.02544017 226085872.83831653 22539567.98214789 ...]
w	int	1	512

```

In [9]: runfile('C:/Teme facultate/PI/LABORATOR 3/filt_de_netzeire.py', wdir='C:/Teme facultate/PI/LABORATOR 3')
uint8
sum1: 55659
sum2: 55447.809851223654
sum3: 15606955418.40372
Mse înainte de filtrare: 99.91902822686156
Mse pentru filtrarea ponderata inainte de filtrare: 35.099186471272574
Mse pentru filtrarea aritmetica inainte de filtrare: 41.223447503861344
timp executie: 3.1728880465426025

In [10]:

```

Sum1= Suma pixeli imagine initiala

Sum2= Suma pixeli imagine finala