

Universitatea POLITEHNICA din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Efficient Block Matching for Removing Impulse Noise

Proiect PAIC

Coordonator:
Prof. Dr. Ing. Constantin VERTAN

Masterand TAID:
Gheorghe-Iulian CHIVU

Anul 2022

Cuprins

Lista figurilor	4
Lista tabelelor	5
Lista acronimelor	6
1. Descrierea si implementarea algoritmului de filtrare BMLUT	7
1.1. Identificarea pixelilor zgomotosi	7
1.2. Crearea LUT	10
1.3. Indepartarea zgomotului	13
2. Rezultate experimentale	14
Concluzii	16
Concluzii generale	16
Dezvoltări viitoare	16
Bibliografie	17

Lista figurilor

1.1.	Histograma 2D de vecinatate pe planul verde de culoare pe imaginea Lena cu 30% zgomot impulsiv	7
1.2.	Sectiunea pe axa X a histogramei 2D de vecinatate in X=150	8
1.3.	(a) Canalul de verde afectat de zgomot impulsiv 30%. (b) Masca de clasificare a pixelilor zgomotosi $\delta=20$, prag=1	8
1.4.	Masca de clasificare a pixelilor zgomotosi pentru prag=1 si: (a) $\delta=20$, (b) $\delta=40$, (c) $\delta=60$	9
1.5.	Masca de clasificare a pixelilor zgomotosi pentru $\delta=40$ si: (a)prag=1, (b)prag=3, (c)prag=6	9
1.6.	Masca de clasificare a blocurilor 3x3 initiala	10
1.7.	Masca de clasificare a blocurilor 3x3 completa	11
1.8.	Masca de clasificare a blocurilor 3x3 completa + zoom	11
1.9.	BMLUT	12
1.10.	Impartirea in 4 noi blocuri a unui bloc de tip zgomot	13
2.1.	(a) Canalul de verde al imaginii originale afectat de zgomot impulsiv 30%, (b) Imaginea filtrata BMLUT	14
2.2.	(a) Imaginea originala afectata de zgomot impulsiv 30%, (b) Imaginea filtrata BMLUT	14

Lista tabelelor

2.1.	Rezultate experimentale - Lena	15
2.2.	Rezultate experimentale - Mushroom	15
2.3.	Rezultate experimentale - Rose	15
2.4.	Rezultate experimentale - Brasov	15
2.5.	Rezultate experimentale - Cooper	15

Lista acronimelor

LUT = LookUp Table

BMLUT = Block Matching by a LookUp Table

Capitolul 1

Descrierea si implementarea algoritmului de filtrare BMLUT

In acest capitol este prezentata partea teoretica a algoritmului de filtrare BMLUT [1] (Block Matching by a LookUp Table) si implementarea acestuia in limbajul de programare Python. Algoritmul a fost definit pentru a filtra zgomot de tip impulsiv intrucat functioneaza ca un filtru de tip median care se foloseste si de informatia aflata in alte blocuri similare din poza, nu doar de vecinatatea pixelului curent procesat.

1.1 Identificarea pixelilor zgomotosi

Primul pas al algoritmului de filtrare BMLUT consta in indentificarea pixelilor zgomotosi din poza de intrare. Acest pas este realizat folosind metoda Homogeneity Level Information [2] pentru fiecare pixel din poza in parte.

Aceasta metoda consta in crearea unei histograme 2D $H(x,y)$ (Figura 1.1) de dimensiune 256×256 care contine informatie asupra valorilor vecinitatii fiecarui pixel (O valoare discreta $H(i,j)$ reprezinta de cate ori un pixel cu valoarea j a fost gasit in vecinatatea unui pixel cu valoarea i).

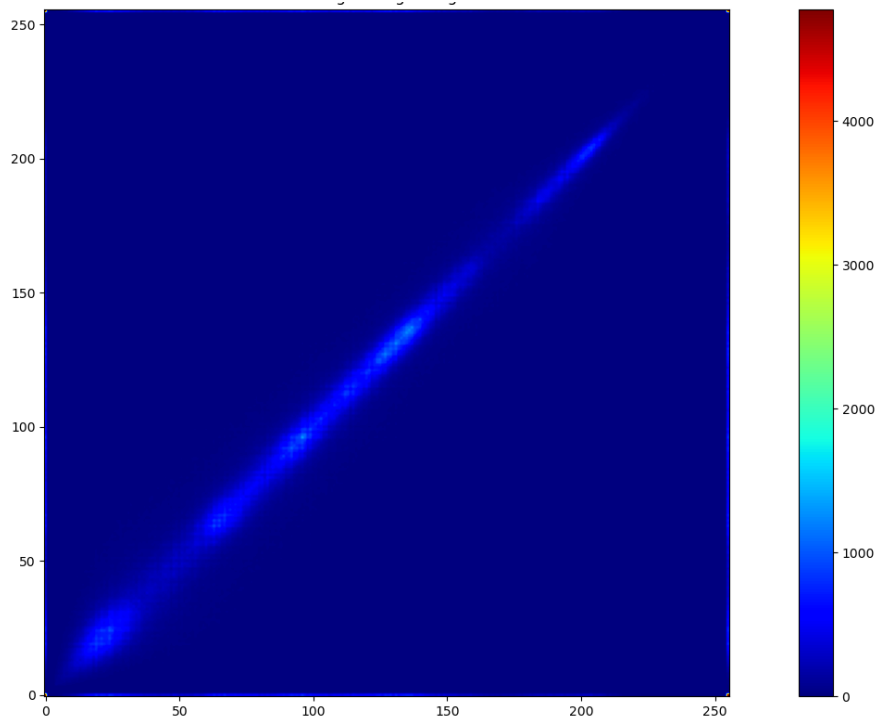


Figura 1.1: Histograma 2D de vecinatate pe planul verde de culoare pe imaginea Lena cu 30% zgomot impulsiv

Pentru crearea histogramei 2D de vecinatate am initializat o matrice numpy cu valori de 0 si am parcurs imaginea cu o fereastră de 3x3 cu pas 1, iar pentru fiecare vecin al centrului ferestrei am indexat valoare din matrice la indexul [centru, vecin].

Fiecare sectiune pe axa X a histogramei 2D de vecinatate intr-o valoare discreta (Exemplu: 150) (Figura 1.2) reprezinta distrubutia tuturor valorilor unui pixel [0-255] aflate in vecinatatea acelei valori.

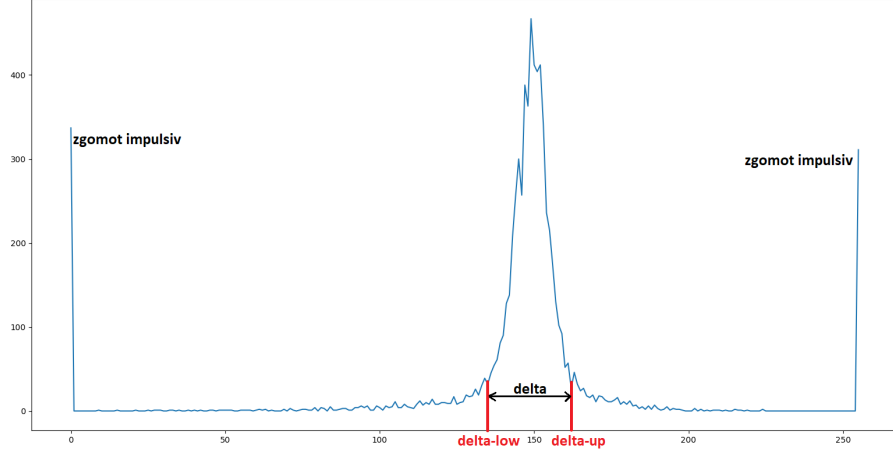


Figura 1.2: Sectiunea pe axa X a histogramei 2D de vecinatate in X=150

In continuare determinam pentru fiecare pixel pragurile δ -low si δ -up care definesc intervalul de omogenitate δ folosind ecuatia (1.1). Unde δ reprezinta un parametru reglabil in intervalul [0,100].

$$\delta = 100 \cdot \frac{\sum_{j=\delta-low}^{\delta-up} H(i, j)}{\sum_{j=0}^{255} H(i, j)} \quad (1.1)$$

Am ales un interval $\delta = 20$ (Figura 1.3), pragurile au fost initializate cu valoare pixelului curent, δ -low a fost decrementat, iar δ -up a fost incrementat pana egalitatea din ecuatia (1.1) a fost indeplinita.

Apoi se defineste un pixel (P-zgomot) afectat de zgomot folosind o fereastră glisanta de dimensiune 3x3, iar pentru fiecare pixel vecin (P-vecin) daca P-zgomot este inclus in intervalul de omogenitate δ al pixelului P-vecin un contor este incrementat. Dupa ce toata vecinatatea pixelului P-zgomot a fost parcursa, daca contorul are o valoare mai mica de un prag (parametru reglabil in intervalul [1,8] - in experimentele realizate am setat acest parametru cu valoarea 1 (Figura 1.5) pentru a incerca sa scot cat mai mult din clasificarea eronata a conturilor), atunci pixelul P-zgomot este considerat pixel afectat de zgomot.

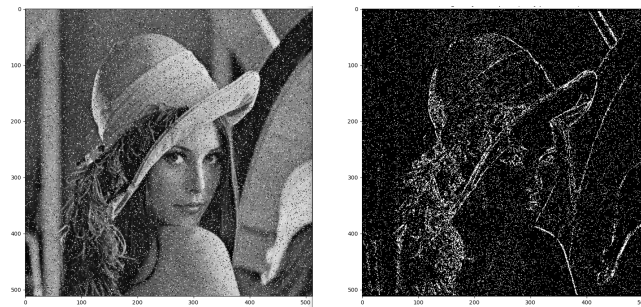


Figura 1.3: (a) Canalul de verde afectat de zgomot impulsiv 30%. (b) Masca de clasificare a pixelilor zgomotosi $\delta=20$, prag=1

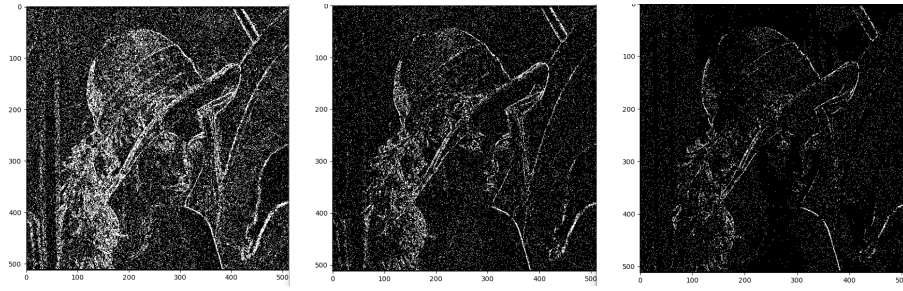


Figura 1.4: Masca de clasificare a pixelilor zgomotosi pentru $\text{prag}=1$ si: (a) $\delta=20$, (b) $\delta=40$, (c) $\delta=60$

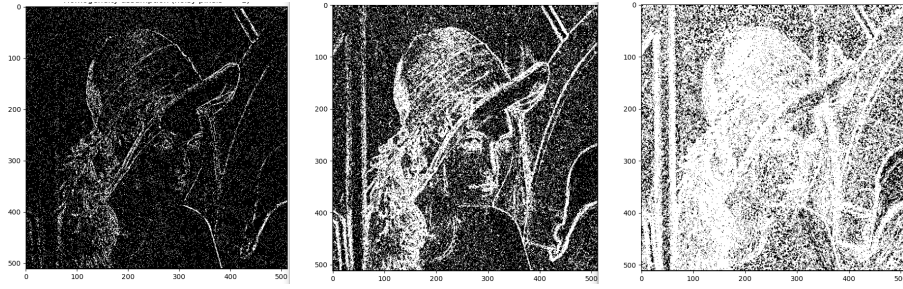


Figura 1.5: Masca de clasificare a pixelilor zgomotosi pentru $\delta=40$ si: (a) $\text{prag}=1$, (b) $\text{prag}=3$, (c) $\text{prag}=6$

In urma acestui pas obtinem o masca de clasificarea pixelilor zgomotosi. Putem observa in Figurile (1.3) (1.4) (1.5) ca zgomotul impusiv este detectat prin aceasta metoda, dar avem si detectii false in jurul conturilor si delaliilor fine din poza (frecvente inalte).

1.2 Crearea LUT

Cu ajutorul mastii de clasificare a pixelilor zgomotosi determinata din pasul anterior putem defini urmatoarele tipuri de blocuri 3x3 din imaginea de intrare:

- **Matching blocks**
- **Target blocks**
- **Noisy blocks**
- **Noisy non-overlapping blocks**

Pentru identificarea fiecarui tip de bloc am creat o alta masca cu 4 valori posibile: 0-matching block, 1-target block, 2-noisy block, iar 3-noisy non-overlapping blocks.

Aceasta masca reprezinta o matrice de dimensiune $(h-2)(w-2)$ a carei valoare determina clasa de apartenta a respectivului bloc, unde h -inaltimea imaginii, w -latimea imaginii.

Masca de clasificare a blocurilor de dimensiune 3x3 este initializata cu valori de 0 (matching blocks) si este parcursa cu pasul 1 pentru a identifica blocurile cu zgomot (noisy blocks). Un bloc este clasificat noisy block atunci cand fereastra 3x3 contine mai multi pixeli zgomotosi decat un prag stabilit. In lucrarea [1] a fost folosit un prag = 5 pixeli.

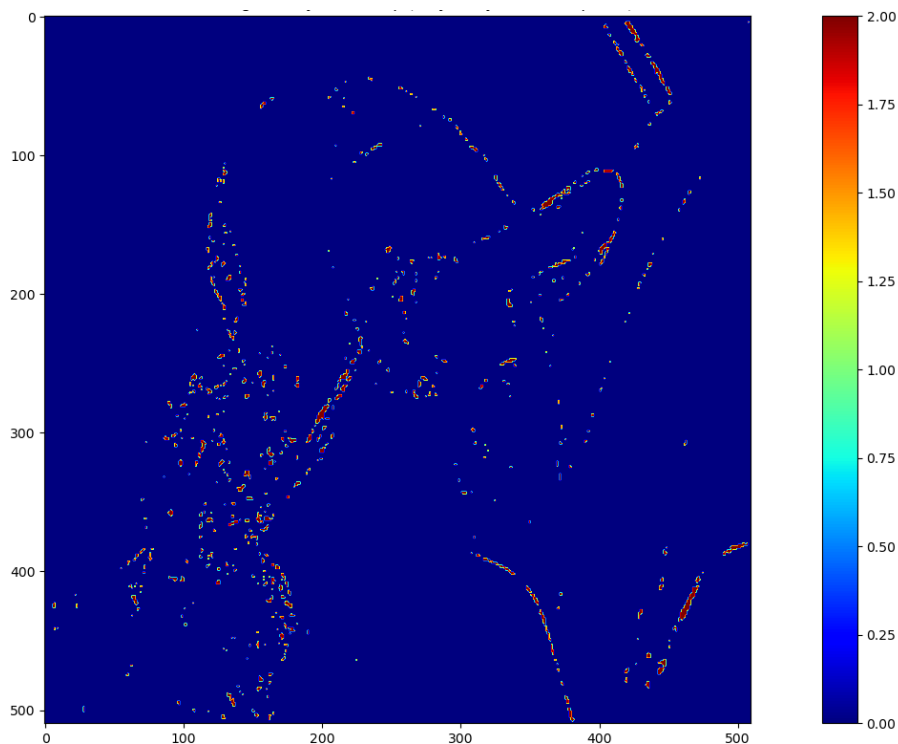


Figura 1.6: Masca de clasificare a blocurilor 3x3 initiala

In urma parcurgerii mastii de clasificare a pixelilor obtinem masca din (Figura ??) in care avem doar valori de 0-matching blocks respectiv 2-noisy blocks.

Pentru a clasifica blocurile de tip target si noisy non-overlapping masca de clasificare a pixelilor zgomotosi este parcursa din nou cu un filtru de dimensiune 3×3 cu un pas de 3 pixeli. Blocurile de tip target reprezinta orice matching-block care contine cel putin un pixel de tip zgomot, acest tip de bloc urmand sa fie filtrat cu ajutorul restului de blocuri de tip matching. In urma acestei clasificari obtinem forma completa a mastii de clasificare a blocurilor.

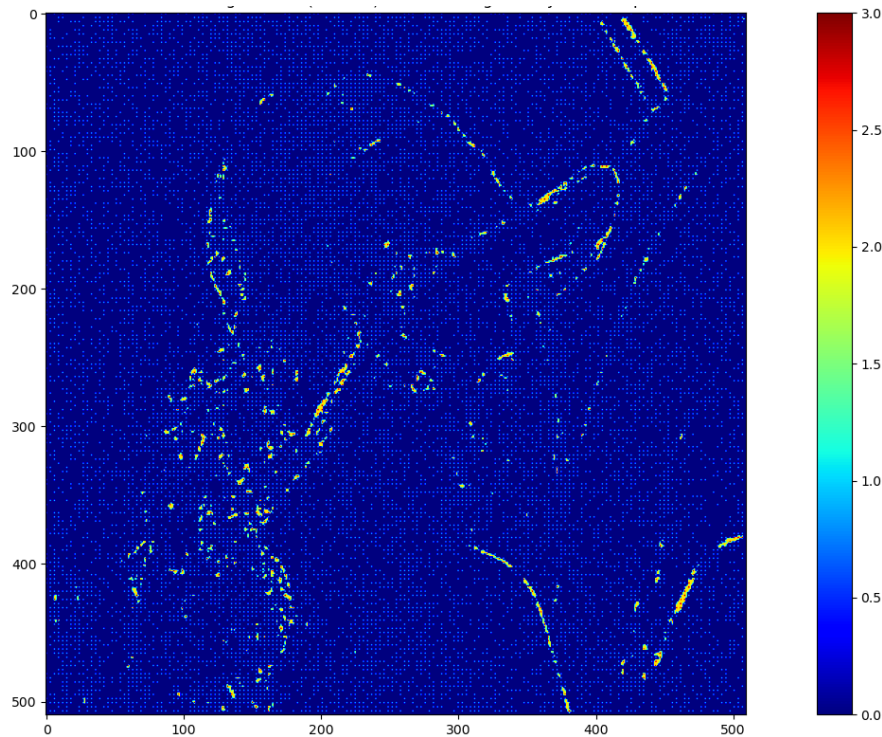


Figura 1.7: Masca de clasificare a blocurilor 3×3 completa

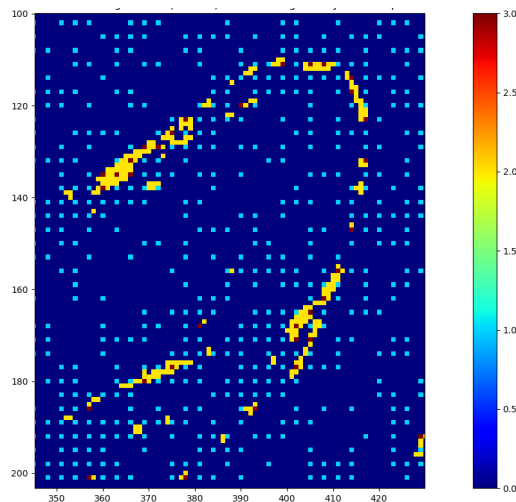


Figura 1.8: Masca de clasificare a blocurilor 3×3 completa + zoom

Acest algoritm de filtrare consta in filtrarea blocurilor de tip target folosind informatiile din restul de blocuri de tip matching. Pentru a reduce timpul de rulare al algoritmului locatiile blocurilor de tip matching sunt salvate intr-un LUT.

Acest LUT are următoarea structura (Figura 1.9). Acesta a fost implementat ca o structura 3D de dimensiuni $[9][256][\text{variabil}]$ cu ajutorul listelor. Prima dimensiune $[0.. 8]$ este data de pozițiile fiecărui pixel dintr-un bloc de tip matching, a doua dimensiune $[0.. 255]$ reprezintă valorile posibile ale unui pixel pe 8 biti, iar ultima dimensiune este variabila deoarece în această sunt salvate coordonatele fiecărui pixel de la poziția $[a]$ cu valoare $[b]$ dintr-un matching block. Coordonatele (i,j) ale pixelului pe imaginea de intrare sunt reprezentate de o singură valoare $= i*w + j$, unde: w -latimea imaginii.

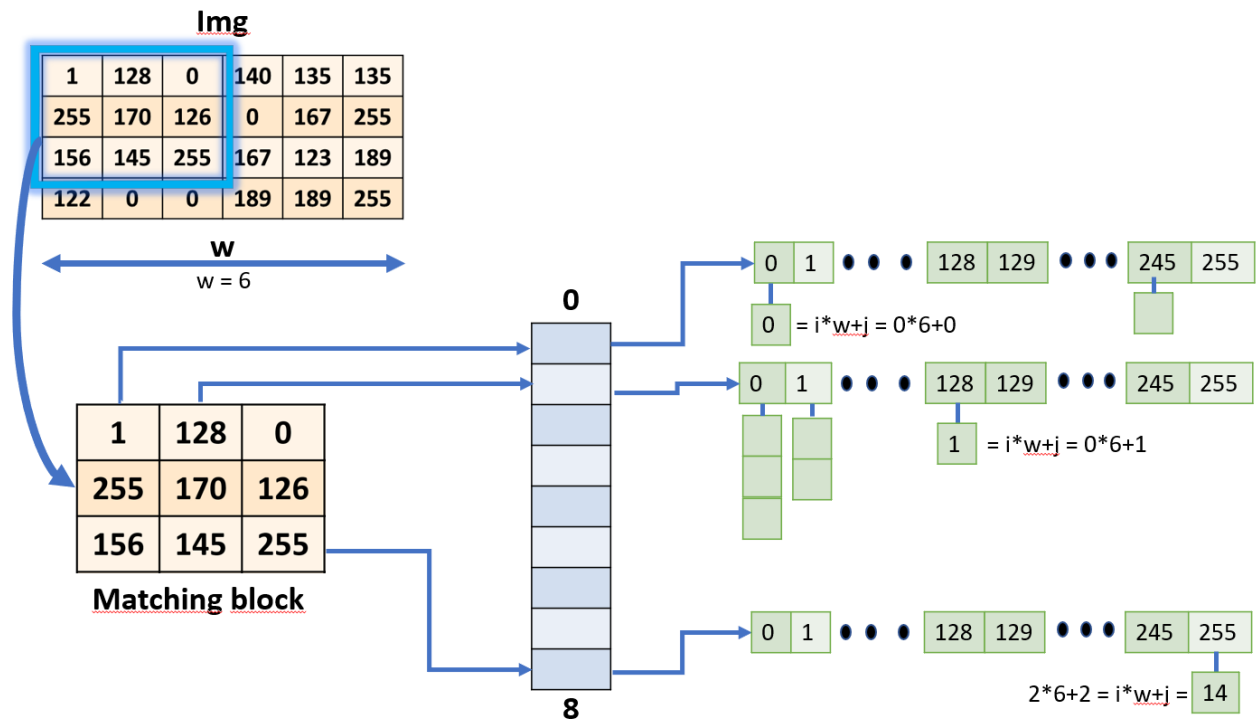


Figura 1.9: BMLUT

1.3 Indepartarea zgomotului

Pentru a indeparta zgomotul dintr-un bloc de tip target cautam in LUT un numar de $\#S$ (parametru reglabil - experiminte $\#S == 3$) blocuri matching care au cel putin τ (parametru reglabil - experiminte $\tau == 3$) pixeli in comun. Pixel in comun == pixel care nu reprezinta zgomot + aceasi pozitie in fereastra 3×3 + diferenta dintre cei doi pixeli nu este mai mare decat o toleranta d . Aceasta toleranta pleaca de la o valoare mica si este incrementata pe parcursul algoritmului de indepartare a zgomotului. Intervalul tolerantei d reprezinta un parametru reglabil, in experimintele realizate am plecat de la o toleranta $d=1$ pana la $d=20$.

Dupa ce au fost gasite $\#S$ matching blocks pentru blocul target curent, sunt calculate mediile pixelilor fara zgomot pentru fiecare pozitie din fereastra 3×3 a celor $\#S$ blocuri matching, iar aceste medii inlocuiesc valorile pixelilor zgomotosi din target block. Daca nu mai exista pixeli de tip zgomot in target block, acesta este clasificat matching block, iar pixelii care au fost filtrati sunt trecuti in imagine.

Pentru a indeparta zgomotul din blocurile clasificate noisy-blocks, acestea sunt impartite in 4 blocuri (Figura 1.10), iar daca numarul de pixeli zgomotosi din cele noile blocuri formate respecta pragul setat in sectiunea (1.2), acestea devin target block.

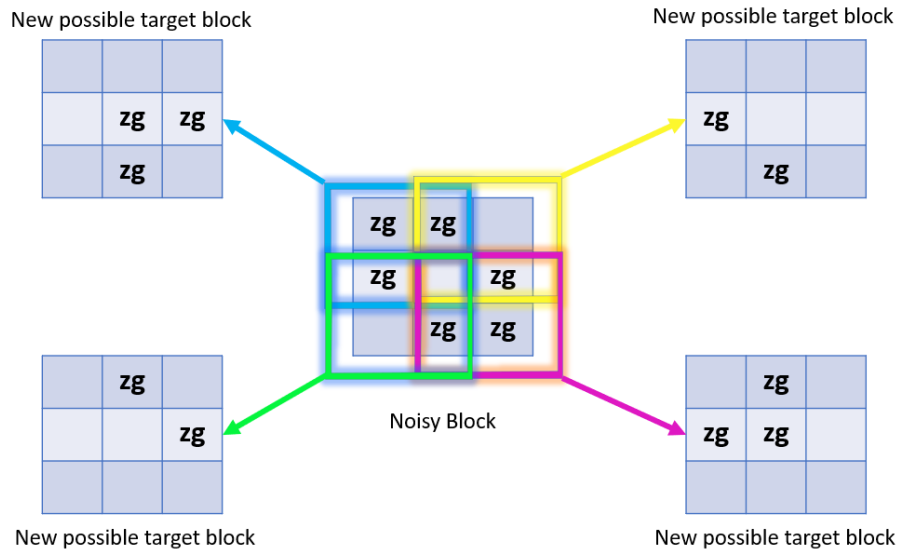


Figura 1.10: Impartirea in 4 noi blocuri a unui bloc de tip zgomot

Acest algoritm se repeta atat timp cat mai avem blocuri de tip target si toleranta d nu a ajuns in pragul maxim setat.

Daca toleranta d a ajuns la pragul maxim setat si exista in continuare blocuri de tip target, atunci intreg algoritmul este repetat pentru blocuri de dimensiune 2×2 , iar noua definitie a blocurilor de tip target este aceea a unui bloc care contine doar un pixel de tip zgomot.

Capitolul 2

Rezultate experimentale

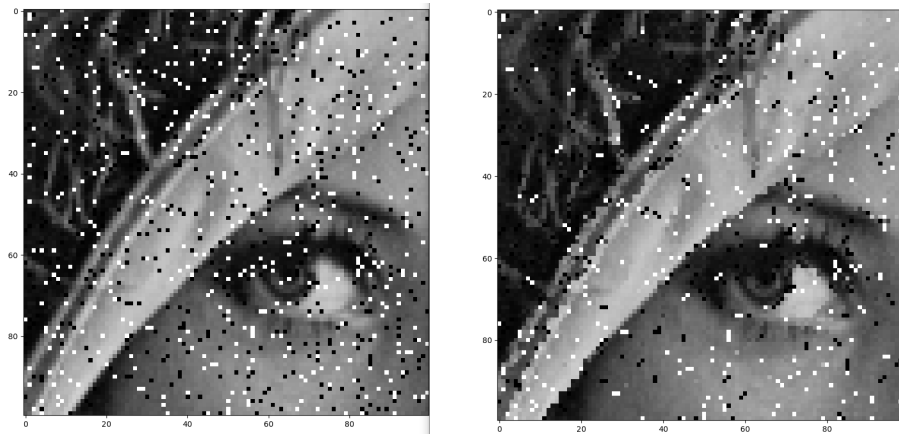


Figura 2.1: (a) Canalul de verde al imaginii originale afectat de zgomot impulsiv 30%, (b) Imaginea filtrata BMLUT

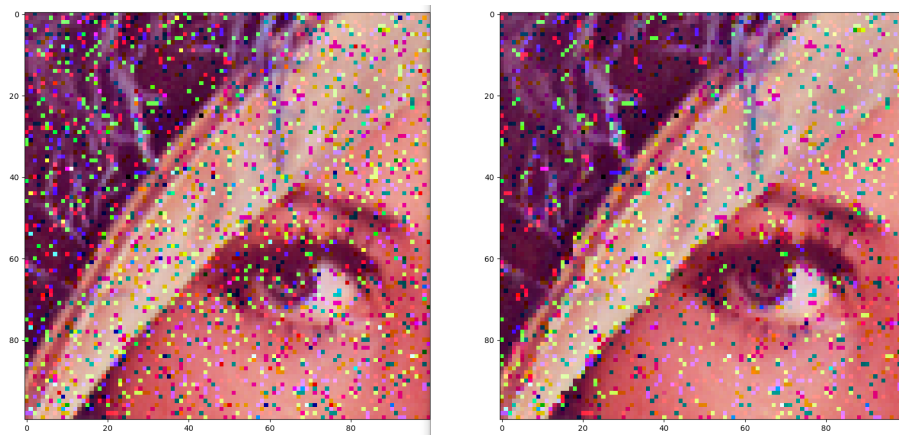


Figura 2.2: (a) Imaginea originală afectată de zgomot impulsiv 30%, (b) Imaginea filtrată BMLUT

Timp de rulare = 27 secunde pentru o imagine RGB de 100x100 pixeli

Poza	Filtru	Zgomot	MAE zgomot	MAE filtrat	PSNR zgomot	PSNR filtrat
Lena	Medie aritm	Impulsiv 10%	4.3	8.3	42.5	26.4
		Gaussian d=10	8	6.4	28	29.1
	Median	Impulsiv 10%	4.3	4	42.5	31
		Gaussian d=10	8	6	28	29.6
	BMLUT	Impulsiv 10%	4.3	11.2	42.5	38.6
		Gaussian d=10	8	118.3	28	30.2

Tabela 2.1: Rezultate experimentale - Lena

Poza	Filtru	Zgomot	MAE zgomot	MAE filtrat	PSNR zgomot	PSNR filtrat
Mushroom	Medie aritm	Impulsiv 10%	3.3	14.7	42.5	22.4
		Gaussian d=10	7.8	13.1	28.1	23.4
	Median	Impulsiv 10%	3.3	10.8	42.5	23.9
		Gaussian d=10	7.8	12.1	28.1	23.7
	BMLUT	Impulsiv 10%	3.3	10.4	42.5	38.3
		Gaussian d=10	7.8	118.6	28.1	30.2

Tabela 2.2: Rezultate experimentale - Mushroom

Poza	Filtru	Zgomot	MAE zgomot	MAE filtrat	PSNR zgomot	PSNR filtrat
Rose	Medie aritm	Impulsiv 10%	3	8.3	44	25.5
		Gaussian d=10	6.8	6.6	29	28.3
	Median	Impulsiv 10%	3	3.7	44	29.9
		Gaussian d=10	6.8	5.9	29	28.9
	BMLUT	Impulsiv 10%	3	7.2	44	40
		Gaussian d=10	6.8	114	29	31

Tabela 2.3: Rezultate experimentale - Rose

Poza	Filtru	Zgomot	MAE zgomot	MAE filtrat	PSNR zgomot	PSNR filtrat
Brasov	Medie aritm	Impulsiv 10%	3.4	15.5	42.7	21.7
		Gaussian d=10	8	14.1	28.1	22.5
	Median	Impulsiv 10%	3.4	12.3	42.7	22.6
		Gaussian d=10	8	13.5	28.1	22.5
	BMLUT	Impulsiv 10%	3.4	10.8	42.7	38.15
		Gaussian d=10	8	118.2	28.1	30.1

Tabela 2.4: Rezultate experimentale - Brasov

Poza	Filtru	Zgomot	MAE zgomot	MAE filtrat	PSNR zgomot	PSNR filtrat
Cooper	Medie aritm	Impulsiv 10%	4.7	5	42.7	29.1
		Gaussian d=10	8	3.1	28.1	35.8
	Median	Impulsiv 10%	4.7	1.1	42.7	41.8
		Gaussian d=10	8	3.7	28.1	34.6
	BMLUT	Impulsiv 10%	4.7	7.2	42.7	41.8
		Gaussian d=10	8	116	28.1	30.4

Tabela 2.5: Rezultate experimentale - Cooper

Concluzii

Concluzii generale

Dupa cum se poate observa din figurile 2.1 si 2.2, implementarea filtrului BMLUT realizata nu a reusit sa filtreze intreaga cantitate de zgomot din imaginea de intrare, intrucat in pasul din sectia 1.1 nu am reusit sa reproduc o masca fidela zgomotului impulsiv. Iar pe tot parcursul algoritmului exista multi parametrii reglabili care pot fi optimizati.

Dezvoltări viitoare

- Optimizarea si Eliminarea conturilor din masca de clasificare a pixelilor.
- Cautare exhaustiva (grid search) a parametrilor reglabili.

Bibliografie

- [1] G.Pok K.H.Ryu. Efficient block matching for removing impulse noise. In *IEEE Signal Processing Letters*, VOL. 25, NO. 8, pages 1176–1180, 2018.
- [2] G.Pok J.C.Liu, A.S.Nair. Selective removal of impulse noise based on homogeneity level information. In *IEEE Trans. Image Process.*, vol. 12, no. 1, pages 85–92, 2003.