Proiect PS

Teodoroiu Vlad-Mihail, 341C2, Automatica si Calculatoare

Restaurarea imaginilor folosing denoising iterativ si proiectii inverse

Prezentare Problema

In procesarea grafica, restaurarea imaginilor este una dintre cele mai dificile sarcini, avand nevoia sa stim ce operatie precisa sa facem si care este nivelul de degradare, ca sa avem un raspuns apt. Acest proiect se axeaza pe denoising, eliminarea “zgomotului” artificial din imagine, degradare obtinuta in urma a mai multor factori, cum ar fi interferente in trimiterea sau stocarea imaginii.

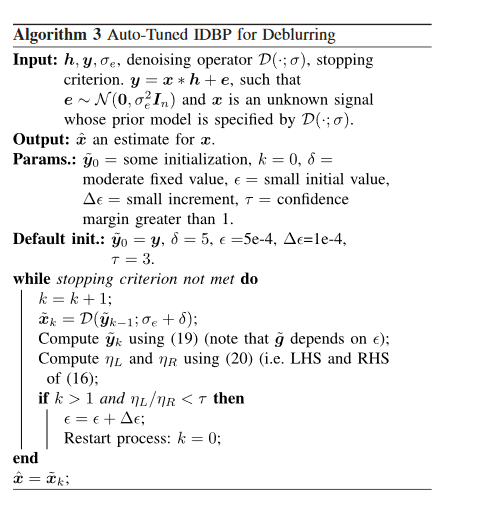
Solutie:

Cum datele initiale ale unui pixel sunt partial pierdute prin acest proces, noise-ul avand gradiente de severitate, de obicei se aplica operatori care estimeaza valoarea pixelului prin luarea in calcul a pixelilor vecini, criteriile de aplicare ale acestor valori vecine diferiind de la operator la operator.

In [paper-ul](https://ieeexplore.ieee.org/abstract/document/8489894) cercetat pentru acest proiect, se prezinta un algoritm de aplicare ai acestor operatori, cu rezultate exceptionale, iar programul rezultat din cercetarea data incearca sa oglindeze acele rezultate.

Algoritmul in sine:

Din pagina 5 a paper-ului cercetat reiese următorul algoritm:



Care tradus pe românește, variabilele de input, output si parametrii înseamnă:

Intrare: h, y, σe, operatorul de denoisificare D(·; σ), criteriu de oprire. y = x \* h + e

Cu e cum ar fi e ~ N(0, σ e 2 In) si x un semnal necunoscut al cărui model anterior e specificat de D(·; σ)

Ieșire: x^, o estimata a lui x.

Parametrii: y ^ 0 = ceva initializare, k = 0; δ = o valoare moderata fixata, ϵ = o valorare mica initializata, Δϵ = valoare mica de increment, τ = margine de toleranta mai mare ca 1

Initializari: y ^ 0= y, δ = 5; ϵ = 3e-3, Δϵ = 1e – 4; τ = 3

Semnificatii variabile:

Criteriul de oprire(stopping criterion): numarul de iteratii pe care vrem sa le parcurgem, presupunem 30

h: matrice de degradare m \* n

y: imaginea degradata

x: imaginea originala

e(din y = x \* h + e): este un vector de identic variabile independente aleatoare Gaussiene distribuite cu medie zero și abaterea standard de σe

x^ : aproximarea imaginii x, i.e. imaginea restaurata

D(·; σ): operator de denoisificare(denoising operator), cu · fiind imaginea de intrare si σ fiind o variabila, pentru acest proiect, folosim functia python convolve() folosind ca parametrii imaginea de input ·, filtrul gausian, modul constant, cval = 0, si σ dat [convolve(·, gaussian\_filter, mode='constant', cval=0.0, sigma= σ)], alternativ putem folosi BM3D

Testare rezultate:

Pentru testare vom folosi database-ul BSD68 valabil la <https://github.com/clausmichele/CBSD68-dataset>

Analiza rezultatelor: calculam PSNR-ul intre imaginea cu zgomot si imaginea originala si comparam cu PSNR-ul imaginii rezultat cu imaginea originala. Cu atat numarul e mai mare cu atat e mai “clara”  
apoi comparam folosind SSIM intre imaginea rezultata si imaginea originala prin impartire. Cu atat SSIM-ul e mai aproape de 1 cu atat cele doua imagini sunt mai apropiate.

Sintetizare Rezultate:

Din paginile 8 si 11 ale paper-ului, rezultatele obtinute de ei pentru SSIM este in raza de 0.8 +- 0.1

Rezultatele obtinute de algoritmul facut:

noisy5 Average SSIM: 0.85

noisy10 Average SSIM: 0.73

noisy15 Average SSIM: 0.69

noisy25 Average SSIM: 0.59

noisy35 Average SSIM: 0.52

noisy50 Average SSIM: 0.45

Dupa cum se poate observa, pentru o valoare de zgomot redusa, restaurarea imaginii este relativ buna, dar cu atat se adauga mai mult zgomot cu atat se pierde din ce in ce mai multa informatie.

Concluzii:

Metoda prezentata in paper are potential, reiesind din testele lor, dar, neavand acess neobfuscat la operatorii de denoisificare sau valorile exacte de variabile folosite, rezultatele sunt mai slabe decat cele din paper.

Bibliografie:

<https://ieeexplore.ieee.org/abstract/document/8489894>

<https://github.com/clausmichele/CBSD68-dataset>