

Metode Numerice - Tema 2

Aspecte legate de implementare

Pentru rezolvarea acestei teme, am urmat pas cu pas explicatiile din schelet si din enunt, care nu puteau fi mai bune de atat. Foarte clar enuntul, foarte clare explicatiile.

Un aspect important pe care trebuie sa-l mentionez este ca am evitat cat se poate de mult folosirea de for-uri sau while-uri, m-am indepartat cat se poate de mult de gandirea in stilul C, si am apelat la vectorizarile care fac MATLAB / Octave un limbaj atat de special, cand vine vorba de calcule matematice complexe (de fapt nu am folosit niciun for sau while).

Pentru operatiile mai dificile, precum descompunerea svd, calcularea de valori si vectori proprii, sortarea unui vector, suma elementelor unui vector, m-am folosit de functiile built-in Octave, **svd**, **eig**, **sort**, **sum**, etc, iar pentru remodelarea unei matrici intr-un vector, sau invers, am folosit **reshape**.

La calcularea lui μ , am facut operatiile direct pe matrice, avand ca rezultat un vector de μ -uri, fiecare corespunzator unei anumite linii / coloane.

Cel mai bine se vede in cod, prin simplitatea lui.

Observatii asupra imaginilor

Imi place faptul ca in Octave este foarte usor sa vizualizezi ce ai facut, cu functiile **imshow** si **imread** este foarte tare sa vezi rezultatul imediat. Ma pasioneaza procesarea de imagini, si chiar am incercat sa inteleg tehnica, fiind prima oara cand am auzit de ea.

Este interesant de observat cum la valori mici ale lui k se vad diferente enorme, iar de la un k incolo, nu se mai disting diferentele intre poza originala si cea comprimata. La valori foarte mici ale lui k , imaginea se vede ca o secventa de linii colorate, care pastreaza cele mai importante nuante / culori din imaginea originala, de altfel. Cu cat creste k -ul, cu atat se pastreaza mai multe culori. Pentru anumite valori intermediare, intre valorile care fac poza sa fie o colectie de linii, si valorile care redau foarte bine originalul, sunt niste valori care fac ca poza sa para blurata.