

MINISTERUL EDUCAȚIEI NAȚIONALE
UNIVERSITATEA OVIDIUS DIN CONSTANȚA
ȘCOALA DOCTORALĂ DE ȘTIINȚE APLICATE

TEZĂ DE DOCTORAT
REZUMAT

Studiul unor tehnici de reconstrucție
algebrică cu aplicații în reconstrucția și
prelucrarea imaginilor

Conducător științific
Prof. univ. dr. Constantin Popa

Doctorand
Lăcrămioara (Liță) Grecu

Constanța
2013

Introducere

Odată cu evoluția tehnologică problemele legate de reconstrucția imaginilor și procesarea imaginilor (recunoașterea formelor/fețelor) au căpătat o răspândire din ce în ce mai mare și beneficiază de un real interes. Problemele legate de reconstrucția de imagini apar în domenii ca: medicină (tomografia computerizată), geologie (geotomografia electromagnetică), etc. Recunoașterea fețelor este o problemă de mare interes în domenii ca: siguranță și probleme de supraveghere, probleme medico-legale (criminalistice), interfața om-calculator, comunicații multimedia, și așa mai departe.

Lucrarea de față tratează aceste două direcții de actualitate:

1. reconstrucția de imagini, mai exact demonstrarea, pe o cale de demonstrație complet diferită de cea din [28], a convergenței algoritmului **Simultaneous Algebraic Reconstruction Technique** (SART) prin încadrarea acestui algoritm într-o clasă generală de metode proiective;
2. recunoașterea fețelor, mai exact propunem doi algoritmi noi care să rezolve problema recunoașterii fețelor. Primul algoritm este de tip Principal Component Analysis (PCA) (bazat pe proiectarea bazei de date pe un subspațiu de dimensiune mai mică) și este o personalizare a algoritmului prezentat în [6] și [7]. Cel de-al doilea algoritm este bazat pe tensori și este indicat pentru baze de date de dimensiuni mari, în care numărul de poze din baza de date este mai mare decât rezoluția unei imagini.

Teza este structurată pe cinci capitole după cum urmează.

În *primul capitol* sunt prezentate noțiuni și concepte de bază de algebră liniară și analiză numerică folosite pe parcursul tezei. Noțiuni despre norme vectoriale și matriceale, descompunerea în valori singulare, proiecții ortogonale și proiecții oblice, probleme în sensul celor mai mici pătrate. Este de asemenea prezentată și problema reconstrucției algebrice a imaginilor.

În acest prim capitol este prezentat și un algoritm de generare a matricei de scanare în Geotomografia Electromagnetică (EGT). Este descris algoritmul de obținere a matricei de scanare pentru diverse tipuri de scanare: “well to well”, “surface to well” și scanare totală. Pentru toate cele trei tipuri de scanare există câte un rezultat în ceea ce privește numărul de elemente nenule ale matricei de scanare ce trebuie obținută. Mai mult, este descris și un procedeu de optimizare a algoritmilor de generare a matricei de scanare, pentru a reduce timpii de calcul ai algoritmilor.

În *capitolul al doilea* sunt introduse Tehnicile de Reconstrucție Algebrică a imaginilor (TRA), tehnici de reconstrucție algebrică bazate pe proiecții succesive și tehnici de reconstrucție algebrică bazate pe proiecții simultane. Mai mult, se studiază convergența tehnicilor de reconstrucție algebrică bazate pe proiecții succesive.

Tot în acest capitol se studiază convergența unor extensii ale tehnicilor de reconstrucție algebrică la probleme inconsistente și la probleme cu constrângeri. Acest studiu se face prin prisma unei metode generale proiective, introdusă în [38], care înglobează metodele de reconstrucție algebrică cunoscute: Kaczmarz, Cimmino, etc.

Capitolul al treilea se ocupă de studiul convergenței tehnicilor de reconstrucție algebrică bazate pe proiecții oblice simultane, mai exact studiul convergenței algoritmului SART. Algoritmul SART a fost introdus de A. H. Andersen și A. C. Kak în 1984 (a se vedea [1]) ca o alternativă simultană la algoritmul Kaczmarz care este bazat pe proiecții ortogonale succesive.

Algoritmul SART se bazează pe proiecții oblice simultane, iar termenii de corectare a erorilor se aplică simultan pentru toate razele unei anumite proiecții.

Studiul convergenței acestui algoritm este făcut prin integrarea algoritmului SART la metoda generală proiectivă introdusă în capitolul doi, adică o cale nouă de demonstrație a convergenței. Încadrarea algoritmului SART la metoda generală proiectivă permite aplicarea constrângerilor asupra acestui algoritm, ceea ce va conduce la reconstrucții mai bune.

În *capitolul patru* sunt prezentate clase de algoritmi pentru recunoașterea formelor. Pe parcursul capitolului sunt descriși algoritmi bazați pe analiza liniară, adică proiecția pe un subspațiu de dimensiune mai mică. La începutul capitolului este prezentat algoritmul “fețelor proprii” (eigenfaces) care deși este una dintre cele mai vechi metode (a se vedea [26], [29] și [40]) folosite în recunoașterea fețelor, are încă rezultate satisfăcătoare.

PCA (Principal Component Analysis) este un algoritm matematic prin care se obține dintr-un set de date inițial, de dimensiune mare, un alt set de date, de dimensiune mai mică, cu anumite proprietăți, date numite componente principale. Această metodă este definită în așa fel încât componentele principale sunt alese în ordinea importanței. Prima componentă principală obținută corespunde celei mai mari valori proprii, cea de-a doua componentă corespunde următoarei valori proprii (ca mărime) și tot așa. În același timp se ține cont de restricția ca fiecare componentă principală să fie ortogonală pe toate componentele precedente.

Apoi, este propus un algoritm nou pentru recunoașterea fețelor, algoritmul COD-A1. Acest algoritm este o variantă personalizată a algoritmului prezentat în [6] și [7]. Pornind de la ideea de a aproximar din [6] și [7], și de la ideea de a reprezenta toate imaginile din baza de date într-un subspațiu de dimensiune mai mică, propunem acest algoritm COD-A1 care, după cum se poate vedea din Capitolul cinci, capitolul de experimente, are o rată de recunoaștere mai mare decât cea a algoritmului PCA.

Nu în ultimul rând este propus un nou algoritm bazat pe tensori, algoritmul A2. Acest algoritm este mult mai potrivit pentru cazul în care mulțimea de date este bine organizată și numărul de poze din baza de date este mai mare

decât rezoluția unei imagini, lucru care nu a fost luat în considerare până acum deoarece bazele de date disponibile erau de dimensiuni mici (numărul imaginilor dintr-o bază de date era mic). Acest algoritm bazat pe tensori are o rată de recunoaștere mai mare decât cea a algoritmului PCA.

Motivația pentru care tensorii sunt utilizați în recunoașterea fețelor este faptul că de multe ori datele sunt stocate sub formă de tensor. În loc de a stoca toate datele sub formă de matrice, toate fotografiile aparținând aceleiași persoane se pot stoca într-o matrice, iar toate matricele corespunzătoare tuturor persoanelor formează un tensor. Acest lucru conduce la o organizare mai bună a imaginilor din baza de date.

Capitolul cinci este rezervat experimentelor numerice. În acest capitol sunt prezentate teste realizate cu algoritmul de generare a matricei de scanare în Geotomografia Electromagnetică din Capitolul unu; comparații între rezultatele obținute cu algoritmul SART (prezentat în Capitolul trei) și cele obținute cu algoritmul Kaczmarz, algoritm standard pentru clasa algoritmilor de tip TRA, experimentele fiind efectuate pe două “fantome” frecvent utilizate în literatura de specialitate (a se vedea [24]); și experimente cu toți algoritmii descriși pe parcursul Capitolului patru, testele fiind efectuate pe patru baze de date, două baze de date proprii (CTOVF și CTOVD) și alte două baze de date cunoscute în literatura de specialitate (ORL și ExtYaleB).

Rezultatele originale prezentate în această teză sunt cuprinse în lucrările:

- [16] Popa C., Grecu L., *Constrained SART algorithm for invers problems in image reconstruction*, Inverse Problems and Imaging, 1(7) (2013), 199-216. (Scor Relativ de Influență = 2.13, Factor de Impact = 1.074)
- [18] Grecu L., *On the construction of the scanning matrix in Electromagnetic Geotomography*, Buletinul Științific al Universității ”Politehnica” din Timișoara, Seria Matematică - Fizică, 1 (2012), 42-56.
- [17] Grecu L., Nicola A., *Some results on simultaneous algebraic techniques in image reconstruction from projections*, ROMAI Journal, 2 (2009), 79-96, <http://rj.romai.ro/>
- [41] Pelican E., Grecu L., *Comparison Between Some Matrix Methods with Applications in Pattern Recognition*, prezentată la Applied Linear Algebra Conference, May 24-28, 2010, Novi Sad, Serbia.
- [20] Grecu (Liță) L. , Pelican E., *Customized Orthogonalization via Deflation Algorithm with Applications in Face Recognition*, trimisă spre publicare la Carpathian Journal of Mathematics, 2013.
- [21] Grecu (Liță) L. , Pelican E., *A Low-Rank Tensor-Based Algorithm for Face Recognition*, trimisă spre publicare la Pattern Recognition Letters, 2013.

- [43] Pelican E., Grecu (Liță) L., *Solving the Pattern Recognition Problem with some Low-Rank Approximation Based Algorithms*, prezentată la XIème Colloque Franco-Roumain de Mathématiques Appliquées, 24-30 August, 2012, București.
- [19] Grecu (Liță) L. , Pelican E., *Systematic and comparative experiments with some algorithms for pattern recognition*, acceptată pentru publicare în volumul conferinței Ninth Workshop on Mathematical Modelling of Environmental and Life Sciences Problems, November 1-4, 2012, Constanța, Romania.
- [42] Pelican E., Grecu L., *Low-Rank Matrix Methods in Pattern Recognition*, prezentată la Balkan Conference on Operational Research (BALCOR), 2009, publicată electronic cu ISBN: 973-86979-9-9.

Constanța, martie 2013

Lăcrămioara (Liță) Grecu

1 Preliminarii

1.1 Noțiuni introductive

Prezentăm în acest capitol câteva noțiuni și concepte de bază de algebră liniară și analiză numerică folosite pe parcursul lucrării: norme vectoriale și matriceale, descompunerea în valori singulare (SVD), proiecții în $\mathbb{R}^n$, probleme liniare în sensul celor mai mici pătrate.

Pe parcursul lucrării vom folosi următoarele notații: A_i linia i din matricea A , A^j coloana j din matricea A , A_{ij} elementul de pe poziția (i, j) din matricea A , I_n matricea identitate de ordin n , A^T transpusa matricei A , $\mathcal{N}(A) = \{x \in \mathbb{R}^n, Ax = 0\}$, $\mathcal{R}(A) = \{y \in \mathbb{R}^m, \exists x \in \mathbb{R}^n, y = Ax\}$. Notăția $S \oplus T$ este folosită pentru suma ortogonală directă a subspațiilor S și T , adică $E = S \oplus T \Leftrightarrow$

$$\begin{aligned} (i) & \forall x \in E, \exists s \in S, t \in T \text{ astfel încât } x = s + t; \\ (ii) & S \cap T = \{0\}; \\ (iii) & \forall s \in S, t \in T, \langle s, t \rangle = 0. \end{aligned} \tag{1}$$

1.2 Reconstrucția algebrică a imaginilor

Formularea matematică a problemei reconstrucției algebrice a imaginilor constă în aflarea unei funcții de două variabile atunci când se cunosc proiecțiile sale, adică integralele sale curbilinii pentru orice dreaptă din plan. Există două formulări: problema directă și problema inversă.

Problema directă presupune calculul integralei unei funcții f (cunoscute) pe o dreaptă C din plan, notată prin

$$I_C(f) = \oint_C f(x, y) dI. \tag{2}$$

Pentru problema inversă se presupune cunoașterea valorii integralei $I_C(f)$ din (2), adică se pot determina valorile funcției de absorbție $f(x, y)$ în orice punct (x, y) din domeniu.

Legea lui Lambert stabilește că există o dependență logaritmică între intensitatea de transmisie (I_S) a razei și cea de recepție (I_R). În acest caz, pentru o rază SR avem

$$\ln \left(\frac{I_R}{I_S} \right) = \oint_{SR} f(x, y) dI. \tag{3}$$

Pentru rezolvarea algebrică a acestei probleme de reconstrucție a imaginilor se consideră două nivele de discretizare: discretizarea mulțimii valorilor funcției de absorbție f (discretizarea imaginii) și discretizarea dreptelor SR din plan (discretizarea scanării). Discretizarea imaginii presupune împărțirea acesteia în elemente de imagine numite pixeli. Discretizarea scanării presupune o mulțime finită de raze de scanare, $SR \in \mathbb{R}^2$.

Dacă folosim un număr de m raze și o discretizare a imaginii în n pixeli, matricea de scanare A va fi de dimensiune $m \times n$. Matricea A este rară (a se vedea Lema 1), nu este de rang plin, are nucleul nenul și este rău condiționată.

Măsurând intensitatea surselor de emisie (S_i) și recepție (în detectorii R_j) pentru raza X cu numărul i , obținem componenta b_i a termenului liber b din modelul discret al problemei de reconstrucție. Această tehnică care reduce reconstrucția imaginilor la rezolvarea problemei în sensul celor mai mici pătrate: să se determine $x \in \mathbb{R}^n$ a.î.

$$\|Ax - b\| = \min \{ \|Az - b\|, z \in \mathbb{R}^n \}, \quad (4)$$

sau pe scurt

$$\|Ax - b\| = \min! \quad (5)$$

se numește **reconstrucție algebrică**. Pentru a rezolva problema (5) a fost dezvoltată o clasă de metode iterative bazate pe proiecții, numită **Tehnici de Reconstrucție Algebrică (TRA)**.

1.3 Un algoritm de generare a matricei de scanare în geotomografia electromagnetică

Rezultatele din această secțiune au fost publicate în lucrarea [18] L. Grecu (Liță), *On the construction of the scanning matrix in Electromagnetic Geotomography*, unde este prezentat un algoritm de generare a matricei de scanare. Procedura de scanare a fost mai întâi folosită în Tomografia Computerizată (CT) (a se vedea [24]). Această idee a fost utilizată cu succes și în Geotomografia Electromagnetică (EGT).

În geotomografie, procedura de scanare constă în forarea a două puțuri de forare (“well”) paralele, și scanarea secțiunii transversale dintre ele, folosind emițători și receptori amplasați de-a lungul celor două sonde. Această metodă oferă informații despre fisurile apărute în urma cutremurelor sau fisuri ale conductelor subterane. După scanarea secțiunii dorite, este obținută o matrice de scanare. Această matrice este folosită de algoritmi de reconstrucție a imaginilor (a se vedea [11]).

Procesul de scanare din geotomografie implică un set de emițători (surse) ($S_1, S_2, \dots, S_p$) și un set de receptori ($R_1, R_2, \dots, R_q$) localizați în cele două puțuri de forare paralele. Din fiecare emițător este trimisă o rază de scanare către fiecare receptor. Aceasta este scanarea “well to well”.

Modelul discret al problemei reconstrucției algebrice este (4) (a se vedea [11], [24]), unde A este matricea de scanare și fiecare element A_{ij} este obținut măsurând lungimea segmentului determinat de intersecția razei i cu pixelul j . Cu ajutorul intensităților surselor de emisie ($S_1, S_2, \dots, S_p$) și de recepție (din receptorii $R_1, R_2, \dots, R_q$) pentru raza i , se obține componenta b_i a termenului liber b a modelului discret (4). Vom nota prin x_{LS} soluția de normă minimă a

problemei în sensul celor mai mici pătrate (4), dată de $x_{LS} = A^+b$, unde A^+ este pseudoinversa Moore–Penrose a matricei A (a se vedea [15]).

O imagine de dimensiune $\alpha \times \alpha$, este împărțită în pixeli și notăm pixeli cu $P_1, P_2, \dots, P_n$. Obținem matricea de scanare A , de dimensiune $m \times n$, unde $m = \text{numărul de emițători} \times \text{numărul de receptori}$ și $n = \text{numărul de pixeli}$. Matricea A se construiește “linie cu linie”, fiecare linie corespunzând unei raze. Mai mult, vom presupune ca pixeli sunt normalizați, adică au muchia egală cu 1.

1.3.1 Scanarea “well to well”

Pentru această metodă de generare a matricei de scanare, fără a restrânge generalitatea, am presupus că numărul de emițători și numărul de receptori este egal și că emițătorii/receptorii sunt poziționați în mijlocul muchiei pixelului.

Din fiecare emițător, o rază de scanare este trimisă spre fiecare receptor. Fie $P_{j_1}, \dots, P_{j_q}$ pixelii care intersectează raza i . Ecuația razei de scanare i este obținută cu ajutorul coordonatelor punctelor corespunzătoare emițătorului și receptorului razei respective. Apoi, din ecuația razei de scanare, este calculată lungimea segmentului determinat de intersecția razei i cu pixelul P_j .

Linia i din matricea A corespunde razei i . Dacă raza i intersectează pixelul j , atunci elementul A_{ij} va fi egal cu lungimea intersecției dintre raza i și pixelul j ; dacă raza i nu intersectează pixelul j , $A_{ij} = 0$. Așa cum vom demonstra în Lema 1, matricea A este rară.

Observația 1 *Aceste considerații pot fi făcute și pentru o imagine dreptunghiulară de dimensiune $\alpha \times \beta$. Atunci numărul de pixeli este $n = \alpha \times \beta$. Metoda de generare a matricei de scanare nu se modifică în acest caz.*

Lema următoare (prezentată fără demonstrație în [32]) afirmă că matricea de scanare A este rară.

Lema 1 [32] *Pentru scanarea “well to well” matricea de scanare A are cel mult $m + n - 1$ elemente nenule pe o linie, pentru o imagine de dimensiune $n \times m$.*

Demonstrația a acestei leme, care ne va ajuta în construcția algoritmică a matricei de scanare.

1.3.2 Scanarea “surface to well”

Pentru a obține rezultate mai bune în reconstrucția de imagini putem folosi de asemenea emițători la suprafață. În acest caz, receptorii sunt poziționați în ambele sonde și razele sunt emise de la suprafață spre ambele părți. Acest tip de scanare este numit scanare “surface to well”. În cazul acestui tip de scanare sunt obținute informații suplimentare despre zona din subteran care se analizează.

Lema 2 Pentru cazul de scanare “surface to well”, matricea de scanare A are cel mult $m + n - 1$ elemente nenule pe o linie, pentru o imagine de dimensiune $n \times m$.

1.3.3 Scanare totală

Procedura de scanare totală include scanarea “well to well” cât și scanarea “surface to well”. Această procedură de scanare oferă cele mai bune rezultate.

Și în cazul scanării totale linia i din matricea A corespunde razei i , care poate fi o rază “well to well” sau o rază “surface to well”. Structura densității matricei corespunzătoare este dată de lema următoare care rezultă din Lemele 1 și 2.

Lema 3 În cazul scanării totale, matricea de scanare A are cel mult $m + n - 1$ elemente nenule pe o linie, pentru o imagine de dimensiune $n \times m$.

1.3.4 Optimizarea algoritmului

Pentru construcția matricei de scanare, trebuie determinată “banda” minimă care este intersectată de o anumită rază. Nu este necesar a se verifica dacă raza intersectează toți pixelii din imagine, ci numai cei din zona respectivă. În acest fel este redus timpul de calcul.

Pentru a găsi zona de pixeli prin care raza este cel mai probabil să treacă trebuie luate în considerare rândurile de imagine pe care sunt poziționați emițătorul și receptorul, și să se verifice dacă raza intersectează pixelii de pe aceste rânduri și pe cei dintre aceste rânduri. Acest lucru poate fi realizat având în vedere numărul de emițători/receptori dintr-un pixel (în cazul în care există mai mult de un emițător/receptor într-un pixel, aceștia sunt poziționați echidistanți). Deoarece emițătorii/receptorii sunt plasați într-o anumită ordine și pentru fiecare rază se cunoaște emițătorul și receptorul corespunzător, se pot deduce rândurile de imagine în care este poziționat emițătorul/receptorul.

Algoritmul pentru aflarea “benzii” de imagine pe care raza o străbate este următorul (a se vedea Figura 1, $\alpha \times \alpha$ este rezoluția unei imagini și presupunem că $\alpha_i > \alpha_j$):

Pasul 1. Se determină intervalul i pe muchia AD unde este poziționată sursa $S_i(0, \alpha_i)$, $i = \alpha - \text{floor}(\alpha_i)$ cu $\alpha_i \in (0, \alpha)$;

Pasul 2. Se determină intervalul j pe muchia BC unde este poziționat receptorul $R_j(\alpha, \alpha_j)$, $j = \alpha - \text{floor}(\alpha_j)$ cu $\alpha_j \in (0, \alpha)$;

Pasul 3. Se obține banda $[i, j]$;

*Pasul 4. Pixelii din bandă sunt cei cu indicii $(i - 1) * \alpha + 1, \dots, j * \alpha$.*

unde $\text{floor}(\alpha)$ este partea întreagă a lui α .

Apoi se determină pixelii din bandă intersectați de raza SR.

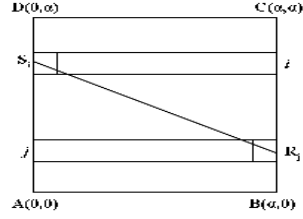


Figura 1: Optimizarea algoritmului

Pentru scanarea “surface to well” este mai ușor de determinat porțiunea de imagine care este cel mai probabil să fie intersectată de rază. Cu fiecare linie din imagine, numărul de pixeli care sunt mai probabil a fi intersectați de rază scade cu 1. Deci, se obține un triunghi de pixeli, care s-ar putea intersecta cu raza. Algoritmul pentru a determina acești pixeli este următorul (avem $S(i, \alpha)$ și $R(0, j)$):

*Pentru fiecare linie de imagine, $r = 1, 2, \dots, j$;
 pixelii care prezintă interes sunt: $(r - 1)\alpha + 1, \dots, (r - 1)\alpha + j - (r - 1)$.*

2 Metode iterative bazate pe proiecții ortogonale pentru rezolvarea problemei reconstrucției algebrice a imaginilor

2.1 Metodele Kaczmarz și Cimmino

Metoda Kaczmarz apare pentru prima dată în 1937 în lucrarea matematicianului polonez S. Kaczmarz [30], dar un studiu teoretic riguros al proprietăților sale este făcut de abia în 1971 de matematicianul japonez K. Tanabe (a se vedea [50]). Metoda Kaczmarz este cunoscută și sub numele de metoda proiecțiilor datorită modalității particulare de construcție a aproximațiilor $x^1, x^2, \dots$. Definim aplicațiile $f_i(b; \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $i = \overline{1, m}$ și $F(b; \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ prin

$$f_i(b; x) = x - \frac{\langle x, A_i \rangle - b_i}{\|A_i\|^2} A_i = P_{H_i}(x), \quad (6)$$

și

$$F(b; x) = (f_1 \circ f_2 \circ \dots \circ f_m)(b; x). \quad (7)$$

Metoda Kaczmarz se scrie atunci astfel

$$x^{k+1} = F(b; x^k), \quad k \geq 0, \quad (8)$$

unde $x^0 \in \mathbb{R}^n$ este aproximația inițială.

Observația 2 O iterație (8) a metodei Kaczmarz constă în proiecții (ortogonale) succesive pe toate hiperplanele $H_m, H_{m-1}, \dots, H_1$ asociate ecuațiilor problemei (4), într-o anumită ordine fixată a priori.

Metoda standard în acest caz Tehnicilor de Reconstrucție Algebrică (TRA) bazate pe proiecții simultane este algoritmul propus de G. Cimmino în [5]. Autorul consideră în loc de proiecțiile ortogonale din metoda lui Kaczmarz, simetricile unei aproximații $x^k \in \mathbb{R}^n$ față de hiperplanele

$$H_i = \{x \in \mathbb{R}^n, \langle x, A_i \rangle = b_i\}.$$

Avem

$$S_i(x) = x - 2 \frac{\langle x, A_i \rangle - b_i}{\|A_i\|^2} A_i. \quad (9)$$

Metoda Cimmino se scrie atunci

$$x^{k+1} = \sum_{i=1}^m \frac{\omega_i}{\omega} S_i(x^k) = x^k - \frac{2}{\omega} \sum_{i=1}^m \omega_i \frac{\langle x^k, A_i \rangle - b_i}{\|A_i\|^2} A_i, \quad (10)$$

unde $\omega_i > 0$ și $\omega = \sum_{i=1}^m \omega_i$.

Astfel x^{k+1} este o combinație convexă a simetricilor $\{S_i(x^k), i = 1, \dots, m\}$.

Vom analiza convergența metodelor Kaczmarz (8) și Cimmino (10) prin intermediul unei abordări generale prezentate în lucrarea [38]. Autorii consideră acolo un algoritm general de forma: $x^0 \in \mathbb{R}^n$, pentru $k = 0, 1, \dots$ definim șirul

$$x^{k+1} = Tx^k + Rb, \quad (11)$$

unde $T : n \times n$, $R : n \times m$, pentru matricele T și R se impun următoarele proprietăți

$$I - T = RA, \quad (12)$$

$$\forall y \in \mathbb{R}^m, Ry \in \mathcal{R}(A^T), \quad (13)$$

$$\text{dacă } \tilde{T} = TP_{\mathcal{R}(A^T)} \text{ atunci } \|\tilde{T}\| < 1. \quad (14)$$

Observația 3 Din (14) obținem $T = TP_{\mathcal{R}(A^T)} + TP_{\mathcal{N}(A)} = \tilde{T} + TP_{\mathcal{N}(A)}$, $P_{\mathcal{N}(A)}\tilde{T} = P_{\mathcal{N}(A)}TP_{\mathcal{R}(A^T)} = 0$, și $\tilde{T}P_{\mathcal{N}(A)} = TP_{\mathcal{R}(A^T)}P_{\mathcal{N}(A)} = 0$.

Propoziția 1 Metoda Kaczmarz (6)-(8) se poate scrie sub forma (11), cu $T = P_1P_2 \dots P_m$, unde $P_i(x) = x - \frac{\langle x, A_i \rangle}{\|A_i\|^2} A_i$, și $R^i = \frac{1}{\|A_i\|^2} P_1P_2 \dots P_{i-1}(A_i)$, unde R^i este coloana i a matricei R .

Metoda Cimmino (10) se poate scrie sub forma (11), cu $T = \sum_{i=1}^m \frac{\omega_i}{\omega} S_i$, unde

$$S_i = I - 2 \frac{A_i A_i^T}{\|A_i\|^2}, \text{ și } R = \sum_{i=1}^m \frac{\omega_i}{\omega} \frac{1}{\|A_i\|^2} A_i.$$

Propoziția 2 ([38]) *Dacă au loc relațiile (12)-(14) atunci*
(i)

$$\text{Dacă } x \in \mathcal{N}(A) \text{ atunci } Tx = x \in \mathcal{N}(A), \quad (15)$$

și

$$\text{dacă } x \in \mathcal{R}(A^T) \text{ atunci } Tx \in \mathcal{R}(A^T). \quad (16)$$

(ii) *Matricea $I - \tilde{T}$ este inversabilă și matricea $G : n \times m$ definită de*

$$G = (I - \tilde{T})^{-1} R, \quad (17)$$

satisface

$$AGA = A \quad (18)$$

și

$$GP_{\mathcal{R}(A)}(b) = x_{LS}. \quad (19)$$

(iii) *Matricea T are proprietățile*

$$\|Tx\| = \|x\| \Leftrightarrow x \in \mathcal{N}(A) \quad (20)$$

și

$$\|T\| \leq 1. \quad (21)$$

(iv) *Pentru șirul $(x^k)_{k \geq 0}$ generat de (11) avem*

$$P_{\mathcal{N}(A)}(x^k) = P_{\mathcal{N}(A)}(x^0), \quad \forall k \geq 0. \quad (22)$$

Teorema 1 ([38]) *Dacă au loc relațiile (12)-(14), șirul $(x^k)_{k \geq 0}$ definit de (11), cu $x^0 \in \mathbb{R}^n$ converge, și*

$$\lim_{k \rightarrow \infty} x^k = P_{\mathcal{N}(A)}(x^0) + Gb, \quad (23)$$

unde G este dat de (17).

2.2 Extinderi la probleme cu constrângeri și la probleme inconsistente

În lucrarea [32] se consideră o funcție de constrângere de forma $C : \mathbb{R}^n \rightarrow \mathbb{R}^n$, cu $Im(C) \subset \mathbb{R}^n$ închisă și cu următoarele proprietăți:

$$\|Cx - Cy\| \leq \|x - y\|, \quad (24)$$

$$\text{dacă } \|Cx - Cy\| = \|x - y\| \text{ atunci } Cx - Cy = x - y, \quad (25)$$

$$\text{dacă } y \in Im(C) \text{ atunci } Cy = y. \quad (26)$$

Un exemplu de astfel de aplicație îl constituie proiecția ortogonală pe o mulțime convexă și închisă în $\mathbb{R}^n$.

Exemplul 1 $C : \mathbb{R}^n \rightarrow [a, b] = [a_1, b_1] \times \dots \times [a_n, b_n]$, dată de

$$(Cx)_i = \begin{cases} a_i, & \text{dacă } x_i < a_i \\ x_i, & \text{dacă } x_i \in [a_i, b_i] \\ b_i, & \text{dacă } x_i > b_i \end{cases} . \quad (27)$$

Lema 4 *Avem $C = P_{[a,b]}$, deci satisface condițiile (24)-(26).*

2.2.1 Metoda proiectivă generală cu constrângeri

Pentru metoda proiectivă generală (11) considerăm varianta cu constrângeri:

$$x^0 \in \mathbb{R}^n, \quad x^{k+1} = C(Tx^k + Rb), \quad k \geq 0. \quad (28)$$

Dacă ξ este limita șirului $(x^k)_{k \geq 0}$ din (11) (a se vedea (23)), pornind cu $x^0 \in \mathcal{R}(A^T)$, și folosind (19) rezultă $\xi = Gb = GP_{\mathcal{R}(A)}(b) + GP_{\mathcal{N}(A^T)}(b) = x_{LS} + \delta$, unde $\delta = P_{\mathcal{N}(A^T)}(b)$. Definim mulțimea $\mathcal{V} = \{y \in \text{Im}(C), y - \delta \in \text{LSS}(A; b)\}$ și presupunem că $\mathcal{V} \neq \emptyset$.

Observația 4 *Dacă problema (4) este consistentă, atunci $\delta = 0$, și*

$$\mathcal{V} = S(A; b) \cap \text{Im}(C). \quad (29)$$

Lema următoare demonstrează că imaginea unui vector $h \in \text{Im}(C)$ prin aplicația (28) este mai aproape de $\mathcal{V}$ decât vectorul h .

Lema 5 [38] *Fie C cu proprietățile (24)-(26). Dacă $h \in \text{Im}(C)$ și*

$$g = C(Th + Rb), \quad (30)$$

atunci, pentru orice $y \in \mathcal{V}$

$$\|g - y\| \leq \|h - y\| \quad (31)$$

și, sau

$$\|g - y\| < \|h - y\| \quad (32)$$

sau

$$g = h \in \mathcal{V}. \quad (33)$$

Teorema 2 [38] *Fie C cu proprietățile (24)-(26). Dacă $x^0 \in \text{Im}(C)$ și $(x^k)_{k \geq 0}$ este definit de (28), atunci $\lim_{k \rightarrow \infty} x^k$ există și aparține lui $\mathcal{V}$.*

2.2.2 Metoda proiectivă generală extinsă

În lucrarea [38] se consideră următoarea variantă extinsă a metodei proiective generale (11). Fie matricele U și S de dimensiuni $m \times m$, respectiv $m \times n$, similare matricelor T și R din (11), pentru sistemul

$$A^T y = 0. \quad (34)$$

Algoritmului iterativ de forma (11) devine în acest caz

$$y^0 \in \mathbb{R}^m, \quad y^{k+1} = Uy^k + S \cdot 0 = Uy^k, \quad \forall k \geq 0. \quad (35)$$

Pentru algoritmul (35) avem următorul rezultat de convergență (corespunzător lui (23)): pentru orice $y^0 \in \mathbb{R}^m$, șirul $(y^k)_{k \geq 0}$, generat cu algoritmul (35), converge și

$$\lim_{k \rightarrow \infty} y^k = P_{\mathcal{N}(A^T)}(y^0). \quad (36)$$

Dacă $y^0 = b$, din (36) obținem $\lim_{k \rightarrow \infty} y^k = P_{\mathcal{N}(A^T)}(b)$, deci $\lim_{k \rightarrow \infty} (b - y^k) = P_{\mathcal{R}(A)}(b)$.

Metoda proiectivă generală extinsă

Fie $x^0 \in \mathbb{R}^n$, $y^0 = b$; pentru $k = 0, 1, \dots$ calculează

$$y^{k+1} = Uy^k, \quad (37)$$

$$b^{k+1} = b - y^{k+1}, \quad (38)$$

$$x^{k+1} = Tx^k + Rb^{k+1}. \quad (39)$$

Pentru algoritmul (37)-(39), în [38] se demonstrează următoarea proprietate de convergență.

Teorema 3 [38] *Dacă matricele T și R satisfac proprietățile (12)-(14), $\forall x^0 \in \mathbb{R}^n$, șirul $(x^k)_{k \geq 0}$ generat de (37)-(39) converge și*

$$\lim_{k \rightarrow \infty} x^k = P_{\mathcal{N}(A)}(x^0) + x_{LS} \in LSS(A; b). \quad (40)$$

2.2.3 Metoda proiectivă generală extinsă cu constrângeri

În lucrarea [38] se consideră următorul algoritm.

Metoda proiectivă generală extinsă cu constrângeri

Fie $x^0 \in \text{Im}(C)$, $y^0 = b$; pentru $k = 0, 1, \dots$ calculează

$$y^{k+1} = Uy^k, \quad (41)$$

$$b^{k+1} = b - y^{k+1}, \quad (42)$$

$$x^{k+1} = C[Tx^k + Rb^{k+1}]. \quad (43)$$

Dacă mulțimea $\mathcal{V}^*$ definită de

$$\mathcal{V}^* = LSS(A; b) \cap Im(C) \quad (44)$$

este nevidă, se arată că șirul $(x^k)_{k \geq 0}$ generat de (41)-(43) converge la un element din $\mathcal{V}^*$.

Lema 6 [38] Pentru orice $y \in \mathcal{V}^*$ avem egalitățile

$$(I - T)y = (I - T)x_{LS} = Rb_A, \quad (45)$$

cu

$$b_A = P_{\mathcal{R}(A)}(b). \quad (46)$$

Lema 7 [38] Fie $k \geq 0$ fixat și presupunem că aplicația C satisface ipotezele (24) și (26). Atunci, pentru orice $y \in \mathcal{V}^*$ avem

$$\|x^{k+1} - y\| \leq \|T(x^k - y) - R\tilde{U}^{k+1}b_A\|. \quad (47)$$

Mai mult, șirul $(x^k)_{k \geq 0}$ generat de (41)-(43) este mărginit.

Din Lema 7, obținem că șirul $(x^k)_{k \geq 0}$ generat de (41)-(43) are un subșir convergent $(x^{k_s})_{s \geq 0}$, i.e.

$$\lim_{s \rightarrow \infty} x^{k_s} = u \in Im(C). \quad (48)$$

Lema 8 [38] În ipotezele Lemei 7, elementul u din (48) aparține mulțimii $\mathcal{V}^*$ din (44).

Lema 9 [38] Dacă șirul $(x^k)_{k \geq 0}$ din (41)-(43) are un subșir convergent $(x^{k_s})_{s \geq 0}$, a cărui limită este $u \in \mathcal{V}^*$, atunci orice alt subșir convergent, $(x^{\bar{k}_s})_{s \geq 0}$, converge la aceeași limită $u \in \mathcal{V}^*$.

Toate lemele de mai sus contribuie direct la demonstrația teoremei următoare.

Teorema 4 [38] Dacă $\mathcal{V}^*$ din (44) este nevidă și au loc (24), (26) și matricea A are toate liniile și coloanele nenule, atunci pentru orice $x^0 \in \mathbb{R}^n$ șirul $(x^k)_{k \geq 0}$ generat cu algoritmul (41)-(43) converge la un element din $\mathcal{V}^*$.

3 Studiul convergenței tehnicilor de reconstrucție algebrică bazate pe proiecții oblice simultane

În lucrarea [16] L. Grecu, C. Popa, *Constrained SART algorithm for inverse problems in image reconstruction* este prezentată încadrarea algoritmului SART la metoda generală (11)-(14), care permite apoi aplicarea constrângerilor asupra algoritmului SART.

În lucrarea [16] am obținut convergența algoritmului SART printr-o altă cale de demonstrație complet diferită și mult mai generală față de rezultatul obținut de autori în lucrările [27] și [28]. Integrând algoritmul SART în metoda generală iterativă din [38] (prezentată în Capitolul doi) obținem nu doar convergența lui, chiar mai mult, putem aplica algoritmului SART constrângeri și putem demonstra convergența algoritmului SART cu constrângeri.

Algoritmul SART a fost introdus de Andersen și Kak în 1984 în lucrarea [1] ca o alternativă simultană la algoritmul Kaczmarz care este bazat pe proiecții ortogonale succesive. Algoritmul SART se bazează pe proiecții oblice simultane. Prima demonstrație de convergență a algoritmului SART a fost făcută în lucrarea [28].

Algoritmul SART: $x^0 \in \mathbb{R}^n$

$$x^{k+1} = x^k + \lambda_k V^{-1} A^T W (b - Ax^k), \quad k \geq 0, \quad (49)$$

unde V și W sunt matrice diagonale pozitiv definite

$$V = \text{diag}(V_{11}, \dots, V_{nn}), \quad W = \text{diag}(W_{11}, \dots, W_{mm}), \quad (50)$$

cu

$$V_{jj} = \sum_{i=1}^m |A_{ij}|, \quad j = 1, \dots, n, \quad \frac{1}{W_{ii}} = \sum_{j=1}^n |A_{ij}|, \quad i = 1, \dots, m. \quad (51)$$

Vom nota cu $\langle \cdot, \cdot \rangle_V$, $\langle \cdot, \cdot \rangle_W$, $\|\cdot\|_V$, $\|\cdot\|_W$ produsele scalare energetice definite pe $\mathbb{R}^n$, $\mathbb{R}^m$ de matricele V , respectiv W și respectiv normele corespunzătoare. Pentru $S \subset \mathbb{R}^n$, $T \subset \mathbb{R}^m$ subspații vectoriale vom nota cu P_S^V , P_T^W proiecțiile ortogonale pe S , T în raport cu $\langle \cdot, \cdot \rangle_V$, respectiv $\langle \cdot, \cdot \rangle_W$, unde $\langle x, y \rangle_V = \langle Vx, y \rangle$, $\|x\|_V = \sqrt{\langle x, x \rangle_V}$, $\|A\|_{V,W} = \sup_{\|x\|_V=1} \|Ax\|_W$. Notăția

$S \oplus_{\perp_V} T$ este folosită pentru suma ortogonală directă a subspațiilor S și T în raport cu produsul scalar energetic $\langle \cdot, \cdot \rangle_V$, adică $E = S \oplus_{\perp_V} T \Leftrightarrow$

$$\begin{aligned} (i) \quad & \forall x \in E, \exists s \in S, t \in T \text{ astfel încât } x = s + t; \\ (ii) \quad & S \cap T = \{0\}; \\ (iii) \quad & \forall s \in S, t \in T, \langle s, t \rangle_V = 0. \end{aligned} \quad (52)$$

3.1 Studiul convergenței algoritmului SART

Considerăm matricea A și adjuncta sa A^τ ca aplicații liniare

$$A : (\mathbb{R}^n, \langle \cdot, \cdot \rangle_V) \rightarrow (\mathbb{R}^m, \langle \cdot, \cdot \rangle_W), \quad A^\tau : (\mathbb{R}^m, \langle \cdot, \cdot \rangle_W) \rightarrow (\mathbb{R}^n, \langle \cdot, \cdot \rangle_V) \quad (53)$$

cu A^τ unic definită de

$$\langle Ax, y \rangle_W = \langle x, A^\tau y \rangle_V, \quad \forall x \in \mathbb{R}^n, y \in \mathbb{R}^m. \quad (54)$$

Pentru algoritmul SART (49) considerăm problema în sensul celor mai mici pătrate: căutăm $x \in \mathbb{R}^n$ astfel încât

$$\|Ax - b\|_W = \min \{ \|Az - b\|_W, z \in \mathbb{R}^n \}. \quad (55)$$

Fie $LS_{V,W}(A; b)$ mulțimea soluțiilor pentru (55), iar $x_{LS}^{V,W}$ soluția de normă $\|\cdot\|_V$ minimă. Problema

$$\left\| \frac{1}{\rho} Ax - \frac{1}{\rho} b \right\|_W = \min! \quad (56)$$

este echivalentă cu (55), iar dacă ρ satisface

$$\rho \leq \min \left\{ \frac{1}{\|A\|_{V,W}}, \frac{1}{\|A^\tau\|_{W,V}} \right\} \quad (57)$$

obținem

$$\left\| \frac{1}{\rho} A \right\|_{V,W} \leq 1. \quad (58)$$

Lema 10 *Adjuncta A^τ a matricei A din (53) este dată de*

$$A^\tau = V^{-1} A^T W. \quad (59)$$

Mai mult, avem descompunerea

$$\mathbb{R}^n = \mathcal{N}(A) \oplus_{\perp_V} \mathcal{R}(A^\tau), \quad \mathbb{R}^m = \mathcal{N}(A^\tau) \oplus_{\perp_W} \mathcal{R}(A), \quad (60)$$

unde $\oplus_{\perp_V}$ și $\oplus_{\perp_W}$ sunt sumele ortogonale directe corespunzătoare produselor scalare $\langle \cdot, \cdot \rangle_V$ și respectiv $\langle \cdot, \cdot \rangle_W$.

Pentru

$$\lambda_k = \lambda, \quad \forall k \geq 0, \quad (61)$$

scriem algoritmul SART (49) sub forma (11), cu $T : n \times n$ și $R : n \times m$ definite astfel

$$T = I - \lambda V^{-1} A^T W A, \quad R = \lambda V^{-1} A^T W. \quad (62)$$

Lema 11 Matricele T și R din (62) au următoarele proprietăți (similare cu (12)-(14))

$$I - T = RA, \quad (63)$$

$$\forall y \in \mathbb{R}^m, Ry \in \mathcal{R}(A^\tau), \quad (64)$$

$$\text{dacă } 0 < \lambda < 2 \text{ și } \tilde{T} = T \cdot P_{\mathcal{R}(A^\tau)}^V, \text{ atunci } \|\tilde{T}\|_{V,V} < 1, \quad (65)$$

$$\text{unde } \|\tilde{T}\|_{V,V} = \sup_{\|x\|_V \leq 1} \|\tilde{T}x\|_V.$$

Propoziția 3 La fel ca în [38], dar conform cu descompunerea (60) avem

$$T = \tilde{T} \oplus_{\perp_V} P_{\mathcal{N}(A)}^V. \quad (66)$$

Teorema 5 Dacă (63)-(65) au loc, sunt adevărate următoarele afirmații
(i)

$$\text{Dacă } x \in \mathcal{N}(A) \text{ atunci } Tx = x, \quad (67)$$

și

$$\text{dacă } x \in \mathcal{R}(A^\tau) \text{ atunci } Tx \in \mathcal{R}(A^\tau). \quad (68)$$

(ii) Matricea $I - \tilde{T}$ este inversabilă și G definită astfel

$$G = (I - \tilde{T})^{-1} R, \quad (69)$$

satisface

$$AGA = A \quad (70)$$

și

$$GP_{\mathcal{R}(A)}^W(b) = x_{LS}^{V,W}. \quad (71)$$

(iii) Matricea T are proprietățile

$$\|Tx\|_V = \|x\|_V \Leftrightarrow x \in \mathcal{N}(A). \quad (72)$$

(iv) Pentru șirul $(x^k)_{k \geq 0}$ definit de (49) avem

$$P_{\mathcal{N}(A)}^V(x^k) = P_{\mathcal{N}(A)}^V(x^0), \quad \forall k \geq 0. \quad (73)$$

(v) Dacă (63)-(65) au loc atunci $(x^k)_{k \geq 0}$ definit de (49), cu $x^0 \in \mathbb{R}^n$ converge,
și

$$\lim_{k \rightarrow \infty} x^k = P_{\mathcal{N}(A)}^V(x^0) + x_{LS}^{V,W}, \quad (74)$$

unde matricea G este definită de (69).

Observația 5 Forma simultană a metodei SART permite paralelizarea completă a acestui algoritm. Considerații privind acest aspect sunt prezentate în Capitolul 5.

3.2 Algoritmul SART cu constrângeri

Încadrarea algoritmului SART la metoda generală (11)-(14) permite aplicarea constrângerilor asupra acestui algoritm, obținând algoritmul SART cu constrângeri (CSART). Acest rezultat nu este obținut în [27] sau [28].

Fie $C : \mathbb{R}^n \rightarrow \mathbb{R}^n$ o aplicație (în general neliniară) cu $\text{Im}(C) \subset \mathbb{R}^n$ mulțime închisă. Vom înlocui relațiile (24)-(26) cu

$$\|Cx - Cy\|_V \leq \|x - y\|_V, \quad (75)$$

$$\text{dacă } \|Cx - Cy\|_V = \|x - y\|_V \text{ atunci } Cx - Cy = x - y, \quad (76)$$

$$\text{dacă } y \in \text{Im}(C) \text{ atunci } Cy = y. \quad (77)$$

Definim algoritmul SART cu constrângeri ca și în [32].

Algoritmul CSART

Fie $x^0 \in \mathbb{R}^n$, pentru $k = 0, 1, \dots$

$$x^{k+1} = C(Tx^k + Rb). \quad (78)$$

Din (23), pentru $x^0 \in \mathcal{R}(A^\tau)$ avem $\lim_{k \rightarrow \infty} x^k = x_{LS}^{V,W}$ deci, conform lucrării [32] definim mulțimea $\mathcal{S} \subseteq \mathbb{R}^n$ prin

$$\mathcal{S} = \text{Im}(C) \cap LSS_{V,W}(A; b), \quad (79)$$

și presupunem că $\mathcal{S} \neq \emptyset$.

Teorema 6 *Presupunem că C satisface relațiile (75)-(77). Dacă $x^0 \in \text{Im}(C)$ și șirul $(x^k)_{k \geq 0}$ este dat de (78), atunci $\lim_{k \rightarrow \infty} x^k$ există și aparține mulțimii $\mathcal{S}$.*

Propoziția 4 *Aplicația $C : \mathbb{R}^n \rightarrow [a, b] = [a_1, b_1] \times \dots \times [a_n, b_n]$, dată de*

$$(Cx)_i = \begin{cases} a_i, & \text{dacă } x_i < a_i \\ x_i, & \text{dacă } x_i \in [a_i, b_i] \\ b_i, & \text{dacă } x_i > b_i \end{cases}.$$

este o aplicație de constrângere și satisface (75)-(77).

Observația 6 *Aplicația de constrângere definită anterior este folositoare deoarece în realitate intervalele $[a_i, b_i]$ se cunosc (a se vedea [24] pentru reconstrucția de imagini în medicină).*

Propoziția 5 *Aplicația $C : \mathbb{R}^n \rightarrow S = [0, \infty) \times \dots \times [0, \infty)$, definită astfel*

$$(Cx)_i = \begin{cases} x_i, & \text{dacă } x_i \geq 0 \\ 0, & \text{dacă } x_i < 0 \end{cases}.$$

este o aplicație de constrângere și satisface (75)-(77).

4 Clase de algoritmi pentru recunoașterea formelor

În lucrările [20] L. Grecu (Liță), E. Pelican, *Customized Orthogonalization via Deflation Algorithm with Applications in Face Recognition* și [21] L. Grecu (Liță), E. Pelican, *A Low-Rank Tensor-Based Algorithm for Face Recognition* este tratată problema recunoașterii fețelor, o subproblemă a problemei recunoașterii formelor.

Pentru o mai clară expunere a algoritmilor descriși în această lucrare, vom stabili mai întâi problema pentru care vom aplica algoritmi. Fiind dată o bază de date cu imagini aparținând unor P persoane, toate imaginile sunt transformate în vectori $\{\Gamma_1, \Gamma_2, \dots, \Gamma_N\}$. Aceste N imagini sunt împărțite în două submulțimi disjuncte: submulțimea de antrenare (training) și submulțimea de testare (testing). Problema luată în considerare este următoarea: fiind dată o imagine Γ (imaginea unei persoane) din mulțimea de testare, vrem să aflăm dacă algoritmul identifică în mod corect persoana folosind imaginile din mulțimea de training.

4.1 Algoritmul “fețelor proprii” (PCA)

“Fețele proprii” sunt componentele principale ale unei mulțimi de fețe, sau echivalent, vectorii proprii ai matricei de covarianță a unei mulțimi de fețe (a se vedea [51] și [52]). Acești vectori proprii sunt numiți “fețe proprii” (eigen-faces) pentru că atunci când sunt reprezentați, ei seamănă cu fețele umane. O mulțime de fețe proprii poate fi obținută printr-un procedeu matematic numit Analiza Componentelor Principale (Principal Component Analysis - PCA), pe o mulțime de dimensiuni mari, alcătuită din imagini reprezentând diferite fețe. Acești vectori proprii sunt aleși în ordinea descrescătoare a importanței lor, dată de mărimea valorilor proprii asociate. Prima componentă principală corespunde celei mai mari valori proprii, cea de-a doua componentă corespunde următoarei valori proprii ca mărime, și așa mai departe. În același timp, se ia în considerare limitarea ca fiecare componentă principală să fie ortogonală în raport cu toate componentele principale anterioare.

Componentele principale sunt date de vectorii proprii ai matricei de covarianță. Prima componentă principală este vectorul propriu corespunzător celei mai mari valori proprii, cea de-a doua componentă principală este vectorul propriu corespunzător următoarei valori proprii ca mărime și așa mai departe.

Ideea de a folosi componentele principale pentru a reprezenta fețele umane a fost dezvoltată de Sirovich și Kirby (a se vedea [31] și [49]) și folosită de către Turk și Pentland (a se vedea [51] și [52]) pentru detectarea și recunoașterea fețelor.

Pentru o bază de date cu N imagini, fiecare imagine trebuie să aibă aceeași rezoluție $M = n_1 \times n_2$ și este transformată într-un vector Γ_i de dimensiune

$M \times 1$. Apoi se calculează vectorul medie $\Psi = \frac{1}{N} \sum_{i=1}^N \Gamma_i$, care se scade din toți vectorii $\varphi_i = \Gamma_i - \Psi$, $i = 1, \dots, N$ adică, în terminologia statistică, datele sunt “centrate”. După cum se menționează în [13] (pagina 116) prima imagine singulară, care este definită de perechea de vectori singulari principali, seamănă foarte mult cu vectorul medie. Astfel, se pare că nu există nici un beneficiu prin scăderea vectorului medie din toți ceilalți. Aceeași remarcă se poate face pentru algoritmul COD-A1. Pentru algoritmul “fețelor proprii” urmărim în continuare ideea din [51].

Algoritmul PCA este aplicat unei mulțimi de vectori de dimensiuni mari $\varphi_1, \varphi_2, \dots, \varphi_N$. Se caută o mulțime de vectori ortonormali $u_1, u_2, \dots, u_N$ care descriu cel mai bine tiparele apărute în baza de date.

Lema 12 [51] [53] *Vectorul u_k este ales astfel încât cantitatea*

$$\lambda_k = \frac{1}{N} \sum_{i=1}^N (u_k^T \varphi_i)^2 \quad (80)$$

este maximă ținând cont de restricția

$$u_i^T u_k = \delta_{ik} = \begin{cases} 1, & \text{dacă } i = k \\ 0, & \text{altfel} \end{cases} \quad (81)$$

unde vectorii u_k și scalarii λ_k sunt vectori proprii și valori proprii ai matricei de covarianță $C = \frac{1}{N} AA^T$ pentru $A = [\varphi_1 \varphi_2 \dots \varphi_N]$.

Dimensiunea matricei de covarianță C este $M \times M$, unde M este rezoluția unei imagini. Deoarece, în practică, numărul M este foarte mare, efortul computațional pentru a determina M valori proprii și M vectori proprii pentru matricea C este foarte mare. În acest caz se dorește reducerea dimensiunii și deci a volumului de calcul. Fie $L = A^T A$, o matrice de dimensiune $N \times N$. De obicei, N numărul de imagini din baza de date este mult mai mic decât dimensiunea unui vector, M , și este mult mai ușor să se calculeze N valori proprii și N vectori proprii pentru o matrice de dimensiune $N \times N$.

Pentru matricea L considerăm vectorii proprii v_i , adică $A^T A v_i = \mu_i v_i$. Și obținem $AA^T A v_i = \mu_i A v_i$, și obținem că $A v_i$ este vector propriu pentru matricea C . Căutăm N vectori proprii, v_i , pentru matricea L . Din cei N vectori proprii obținuți, sunt păstrați doar primii K , corespunzători celor mai mari K valori proprii, care sunt suficienți pentru a caracteriza mulțimea inițială de date. Ceilalți $N - K$ vectori, corespunzători celor mai mici valori proprii sunt înlăturați deoarece informația asociată cu ei este mai puțin semnificativă. Prin urmare, baza de vectori ortonormați din $\mathbb{R}^M$ (care vor fi folosiți pentru a obține imagini/vectori): $u_1, u_2, \dots, u_N$ va fi trunchiată la $u_1, u_2, \dots, u_K$, cu $K \ll N$.

Pentru a alege nivelul de trunchiere K , putem folosi una dintre cele trei metode menționate în [10]. În experimente vom folosi prima metodă unde sunt eliminați aproximativ 90% din ultimii vectori proprii.

Pentru a identifica o nouă imagine Γ , aceasta este reprezentată folosind vectorii proprii $\{u_1, u_2, \dots, u_K\}$. Avem $\omega_i = u_i^T (\Gamma - \Psi)$, $i = 1 : K$. Coeficienții ω_i formează vectorul $\Omega^T = [\omega_1, \omega_2, \dots, \omega_K]$. Vectorul Ω descrie contribuția fiecărei “fețe proprii” la reprezentarea imaginii Γ și este folosită pentru a clasifica imaginea Γ .

4.2 Algoritmul Nearest Neighbour (NN)

În această secțiune vom prezenta succint un alt algoritm folosit în recunoașterea formelor. În Capitolul de experimente și în lucrarea [19] Grecu (Liță) L., Pelican E., *Systematic and comparative experiments with some algorithms for pattern recognition* sunt prezentate comparații între rezultatele obținute cu acest algoritm și cele obținute cu ceilalți algoritmi prezentați în cadrul acestui capitol.

Cea mai simplă metodă care poate fi folosită în recunoașterea fețelor/formelor este căutarea secvențială, adică găsirea celui mai apropiat “vecin”, în cazul nostru persoana/cifra care “seamănă” cel mai bine cu persoana/cifra căutată. Așadar fiind dată o imagine Γ (imaginea unei persoane sau a unei cifre) vrem să găsim cea mai apropiată imagine din baza de date $\{\Gamma_1, \Gamma_2, \dots, \Gamma_N\}$ de imaginea respectivă. Căutarea secvențială (algoritmul Nearest Neighbour, a se vedea [36]) presupune compararea imaginii Γ cu fiecare Γ_i și găsirea unui indice i_0 astfel încât

$$\|\Gamma - \Gamma_{i_0}\| = \min_{1 \leq i \leq N} \|\Gamma - \Gamma_i\|. \quad (82)$$

Așadar indicele i_0 ne va da imaginea aflată la cea mai mică distanță față de imaginea căutată.

4.3 Algoritmul COD-A1

În lucrarea [20] L. Grecu (Liță), E. Pelican, *Customized Orthogonalization via Deflation Algorithm with Applications in Face Recognition* propunem un algoritm pentru recunoașterea fețelor.

Pentru a îmbunătăți (ca rată de recunoaștere), algoritmul PCA (un algoritm bazat de SVD trunchiat), trebuie căutată o bună aproximare de rang redus pentru matricea A . Chiar dacă algoritmi bazați pe SVD trunchiat dau cele mai bune aproximări de rang redus atât în norma Frobenius cât și în norma L2, pentru matricele de dimensiuni mari, obținerea descompunerii SVD poate fi costisitoare. O opțiune se bazează pe metoda Lanczos și este potrivită pentru lucrul cu matrice de dimensiuni mari. Se știe că metoda Lanczos are

două potențiale dezavantaje. În primul rând, în timp ce este de așteptat să dea estimări bune pentru cele mai mari și cele mai mici valori proprii (valori singulare), are o dificultate în a estima valorile intermediare. În al doilea rând, pe măsură ce crește numărul de iterații, vectorii singulari calculați pot pierde proprietatea de ortogonalitate. Acest ultim aspect este deja rezolvat prin re-ortogonalizare. În lucrările [4] și [14], autorii au utilizat deja metoda Lanczos pentru aproximarea de rang redus. Dar, după cum este raportat în lucrările respective rata de recunoaștere a metodelor propuse nu o depășește pe cea a PCA.

O altă variantă pentru a calcula rangul k de trunchiere al descompunerii SVD este algoritmul propus în [6] și [7]. În aceste lucrări, pentru o matrice $A : M \times N$, $M \geq N$, se generează o secvență de matrice $A_1, A_2, \dots, A_{k+1}$, pentru care

$$A_{k+1} = A_k - \tilde{\sigma}_k \tilde{u}_k \tilde{v}_k^T = A - \sum_{j=1}^k \tilde{\sigma}_j \tilde{u}_j \tilde{v}_j^T = A - \tilde{U}_k \tilde{D}_k \tilde{V}_k^T = A - \tilde{B}_k, \quad (83)$$

unde $\tilde{u}_k, \tilde{v}_k$ și $\tilde{\sigma}_k$ sunt date de (84), (85), și respectiv (86), $\tilde{U}_k = [\tilde{u}_1 \ \tilde{u}_2 \ \dots \ \tilde{u}_k]$, $\tilde{V}_k = [\tilde{v}_1 \ \tilde{v}_2 \ \dots \ \tilde{v}_k]$, $\tilde{D}_k = \text{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \dots, \tilde{\sigma}_k)$, și $\tilde{B}_k = \tilde{U}_k \tilde{D}_k \tilde{V}_k^T$. Matricea $\tilde{B}_k$ servește drept aproximare de rang redus a matricei A .

Fie $\hat{u}_k \in \mathcal{R}(A_k)$ și $\hat{v}_k \in \mathcal{R}(A_k^T)$ o pereche arbitrară de vectori care satisface $\hat{u}_k^T A_k \hat{v}_k > 0$. Avem

$$\tilde{u}_k = A_k \hat{v}_k / \|A_k \hat{v}_k\|_2, \quad (84)$$

$$\tilde{v}_k = A_k^T \hat{u}_k / \|A_k^T \hat{u}_k\|_2, \quad (85)$$

și

$$\tilde{\sigma}_k = (\|A_k \hat{v}_k\|_2 \|A_k^T \hat{u}_k\|_2) / (\hat{u}_k^T A_k \hat{v}_k). \quad (86)$$

Teorema 7 (a se vedea [6] și [7]) Fie matricele $\hat{U}_k \in \mathbb{R}^{M \times k}$, $\hat{V}_k \in \mathbb{R}^{N \times k}$, și $\hat{D}_k \in \mathbb{R}^{k \times k}$ definite de relațiile

$$\hat{U}_k = [\hat{u}_1 \ \hat{u}_2 \ \dots \ \hat{u}_k], \ \hat{V}_k = [\hat{v}_1 \ \hat{v}_2 \ \dots \ \hat{v}_k] \text{ și } \hat{D}_k = (\hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_k), \quad (87)$$

unde $\hat{\sigma}_j = \hat{u}_j^T A \hat{v}_j$, pentru $j = 1, \dots, r = \text{rank}(A)$. Atunci, relațiile următoare au loc pentru $k = 1, \dots, r$:

$$\mathcal{R}(A_{k+1}) \subseteq \mathcal{R}(A_k), \ \mathcal{R}(A_{k+1}^T) \subseteq \mathcal{R}(A_k^T), \quad (88)$$

$$\mathcal{R}(\hat{U}_k) \subseteq \mathcal{R}(A), \ \mathcal{R}(\hat{V}_k) \subseteq \mathcal{R}(A^T), \quad (89)$$

$$\hat{U}_k^T A_{k+1} = 0, \ A_{k+1} \hat{V}_k = 0, \quad (90)$$

$$\hat{U}_k^T \hat{U}_k = I, \quad \hat{V}_k^T \hat{V}_k = I. \quad (91)$$

Deci, pentru $k = r$ coloanele matricelor $\hat{U}_r$ și $\hat{V}_r$ constituie bază ortonormată pentru $\mathcal{R}(A)$ și $\mathcal{R}(A^T)$, respectiv. Prin urmare,

$$A_{r+1} = 0 \quad (92)$$

și

$$A = \tilde{U}_r \tilde{D}_r \tilde{V}_r^T \quad (93)$$

Pornind de la ideea de a aproxima matricea A cu matricea B_k , și de la ideea de a reprezenta toate imaginile într-un subspațiu de dimensiune mai mică, în cele ce urmează propunem o versiune personalizată a algoritmului (83)-(86), algoritmul COD-A1 (Customized Orthogonalization via Deflation Algorithm). Personalizarea constă într-o alegere adecvată la fiecare iterație pentru $\hat{u}_{i+1} \in \mathcal{R}(A_i)$ și $\hat{v}_{i+1} \in \mathcal{R}(A_i^T)$. Am testat și alte variante pentru aceste inițializări (pentru fiecare iterație), dar rezultatele obținute nu au fost satisfăcătoare.

Cu acest algoritm încercăm să aproximăm matricea U din descompunerea în valori singulare a matricei A , $A = U\Sigma V^T$ cu matricea $\tilde{U}_k$ din (83), care va servi drept bază ortonormată pentru $\mathcal{R}(A)$. Apoi urmărim procesul de clasificare de la algoritmul “fețelor proprii”: vom compara noua expresie a imaginii căutate cu toate coloanele din noua matrice obținută (baza de date) și căutăm imaginea din baza de date, care este cel mai aproape (într-o anumită metrică) de imaginea căutată.

Algoritmul propus este următorul.

Algoritmul COD-A1:

Pasul 1. Se inițializează $\hat{u}_1$, $\hat{u}_1 = \hat{u}_1 / \|\hat{u}_1\|$, $\hat{v}_1$, $\hat{v}_1 = \hat{v}_1 / \|\hat{v}_1\|$ și $A_1 = A$.

Pasul 2. for $i = 1, 2, \dots, k$

$$\tilde{u}_i = A_i \hat{v}_i / \|A_i \hat{v}_i\|_2$$

$$\tilde{v}_i = A_i^T \hat{u}_i / \|A_i^T \hat{u}_i\|_2$$

$$\tilde{\sigma}_i = (\|A_i * \hat{v}_i\|_2 \|A_i^T \hat{u}_i\|_2) / (\hat{u}_i^T A_i \hat{v}_i)$$

$$A_{i+1} = A_i - \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^T$$

se reinițializează $\hat{u}_{i+1}$, $\hat{u}_{i+1} = \hat{u}_{i+1} / \|\hat{u}_{i+1}\|$ și $\hat{v}_{i+1}$, $\hat{v}_{i+1} = \hat{v}_{i+1} / \|\hat{v}_{i+1}\|$

end

Pasul 3. Fie $U = [\tilde{u}_1 \ \tilde{u}_2 \ \dots \ \tilde{u}_k]$ și $B = A_{k+1}$.

Pasul 4. Se obține $\Omega_i^T = [\omega_1^i, \omega_2^i, \dots, \omega_k^i]$ unde $col_i A = \sum_{j=1}^k \omega_j^i \tilde{u}_j$.

Pasul 5. Fiind dată o imagine Γ , se obține $\Gamma = \sum_{j=1}^k \omega_j \tilde{u}_j$.

Pasul 6. Se reprezintă Γ ca fiind $\Omega^T = [\omega_1, \omega_2, \dots, \omega_k]$.

Pasul 7. Se caută $i_0 \in \{1, \dots, k\}$ care satisface $\|\Omega - \Omega_{i_0}\| = \min_{1 \leq i \leq k} \|\Omega - \Omega_i\|$.

4.4 Descompunerea în valori singulare pentru tensori

În lucrarea [21] L. Greu (Liță), E. Pelican, *A Low-Rank Tensor-Based Algorithm for Face Recognition* propunem un algoritm pentru recunoașterea fețelor bazat pe descompunerea în valori singulare pentru tensori.

Motivația pentru care tensorii sunt utilizați în recunoașterea fețelor este faptul că de multe ori datele sunt stocate sub formă de tensor. În loc de a stoca toate datele sub formă de matrice, toate imaginile aparținând aceleiași persoane se pot stoca într-o matrice, iar toate matricele corespunzătoare tuturor persoanelor formează un tensor. Acest lucru conduce la o organizare mai bună a imaginilor din baza de date. În această organizare, fețele sunt clasificate într-un număr de grupuri de diferite “expresii”. Această clasificare se referă la unghiuri fotografice (de exemplu, profil stânga, profil dreapta, portret), condițiile de iluminare (întuneric, luminat, etc) sau expresii faciale (fericit, trist, furios, etc), și așa mai departe.

În continuare prezentăm o generalizare a teoremei SVD de la matrice pentru tensori (a se vedea [8] și [13]).

Fie $A \in \mathbb{R}^{l \times m \times n}$, $U \in \mathbb{R}^{l_0 \times l}$ și $A \times_1 U$ un tensor de dimensiune $l_0 \times m \times n$, avem următorul mod de înmulțire (modul-1 de înmulțire tensor-matrice)

$$(A \times_1 U)(j, i_2, i_3) = \sum_{k=1}^l u_{j,k} a_{k,i_2,i_3}. \text{ Pentru } A \in \mathbb{R}^{l \times m \times n}, U \in \mathbb{R}^{m_0 \times m} \text{ și } A \times_2 U$$

un tensor de dimensiune $l \times m_0 \times n$ avem (modul-2 de înmulțire tensor-matrice)

$$(A \times_2 U)(i_1, j, i_3) = \sum_{k=1}^m u_{j,k} a_{i_1,k,i_3}, \text{ și pentru } A \in \mathbb{R}^{l \times m \times n}, U \in \mathbb{R}^{n_0 \times n} \text{ și } A \times_3 U$$

un tensor de dimensiune $l \times m \times n_0$ avem (modul-3 de înmulțire tensor-matrice)

$$(A \times_3 U)(i_1, i_2, j) = \sum_{k=1}^n u_{j,k} a_{i_1,i_2,k}. \text{ Modul-}i \text{ și modul-}j \text{ de înmulțire comută}$$

dacă $i \neq j$, $i, j \in \{1, 2, 3\}$: $(A \times_i U) \times_j V = (A \times_j V) \times_i U = A \times_i U \times_j V$.

Un tensor poate fi desfășurat într-o matrice, $A_{(i)} = \text{unfold}_i(A)$, deci:

$$A_{(1)} = \text{unfold}_1(A) = (A(:, 1, :), A(:, 2, :), \dots, A(:, m, :)),$$

$$A_{(2)} = \text{unfold}_2(A) = \begin{pmatrix} A(:, :, 1)^T & A(:, :, 2)^T & \dots & A(:, :, n)^T \end{pmatrix},$$

$$A_{(3)} = \text{unfold}_3(A) = \begin{pmatrix} A(1, :, :)^T & A(2, :, :)^T & \dots & A(l, :, :)^T \end{pmatrix}.$$

Pentru toate desfășurările, linia i din $A_{(j)}$ conține toate elementele tensorului A care au indexul j egal cu i . Operația inversă desfășurării este înfășurarea $\text{fold}_i(\text{unfold}_i(A)) = A$.

Folosind aceste desfășurări obținem: $A \times_1 U = \text{fold}_1(U \cdot \text{unfold}_1(A))$, $A \times_2 U = \text{fold}_2(U \cdot \text{unfold}_2(A))$, și $A \times_3 U = \text{fold}_3(U \cdot \text{unfold}_3(A))$. Produsul scalar $\langle A, B \rangle$ a doi tensori $A, B \in \mathbb{R}^{l \times m \times n}$, este definit astfel $\langle A, B \rangle =$

$$\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n A(i, j, k) B(i, j, k).$$

Teorema 8 (HOSVD, a se vedea [8] și [13]) Tensorul $A \in \mathbb{R}^{l \times m \times n}$ poate fi scris ca

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \quad (94)$$

unde $U^{(1)} \in \mathbb{R}^{l \times l}$, $U^{(2)} \in \mathbb{R}^{m \times m}$, $U^{(3)} \in \mathbb{R}^{n \times n}$ sunt matrice ortogonale. Matricele $U^{(i)}$ sunt obținute din $A_{(i)} = U^{(i)} \Sigma^{(i)} (V^{(i)})^T$, $A_{(i)} = \text{unfold}_i(A)$, fără ca matricele $V^{(i)}$ să fie formate explicit. S este un tensor de aceeași dimensiune ca și A și satisface

- oricare două “felii” diferite fixate în același mod sunt ortogonale
 $\langle S(i, :, :), S(j, :, :)\rangle = 0$, $i \neq j$, $\langle S(:, i, :), S(:, j, :)\rangle = 0$, $i \neq j$,
 $\langle S(:, :, i), S(:, :, j)\rangle = 0$, $i \neq j$.
- normele “feliilor” în orice mod sunt ordonate, de exemplu pentru primul mod avem $\|S(1, :, :)\| \geq \|S(2, :, :)\| \geq \dots \geq 0$.
- descompunerea (94) nu este unică (a se vedea pentru detalii [13] și [8])

Fie $A \in \mathbb{R}^{n_i \times n_e \times n_p}$ un tensor reprezentând baza de date (mulțimea de training) și fie un vector din $\mathbb{R}^{n_i}$ reprezentând o poză din mulțimea de testing. Aici n_i este rezoluția unei imagini ($n_i = M$ din secțiunile anterioare), n_p este numărul de persoane din baza de date ($n_p = P$ din secțiunile anterioare), și n_e este numărul de expresii pentru o persoană ($n_e \cdot n_p = N$ din secțiunile anterioare). Vrem să vedem dacă algoritmul o identifică corect. În [13], pentru algoritmi de la paginile 173-174 se presupune că $n_i \gg n_e n_p$. Pentru o bază de date mare (de exemplu baza de date ExtYaleB), reducând dimensiunea imaginilor, se obține $n_i \ll n_e n_p$. În urma experimentelor cu algoritmul din [13] pentru baza de date ExtYaleB am obținut o rată de recunoaștere mai mică decât pentru algoritmul PCA.

Pentru a soluționa această problemă propunem un nou algoritm pentru recunoașterea fețelor, pentru cazul în care $n_i \ll n_e n_p$.

Pentru problema în discuție, folosim următoarea formă a teoremei HOSVD $A = C \times_e G$, $C = S \times_i F \times_p H$, unde $\times_i = \times_1$, $\times_e = \times_2$, $\times_p = \times_3$ și $F = U^{(1)} \in \mathbb{R}^{n_i \times n_i}$, $G = U^{(2)} \in \mathbb{R}^{n_e \times n_e}$, $H = U^{(3)} \in \mathbb{R}^{n_p \times n_p}$ din teorema HOSVD.

Pentru o anumită persoană p avem: $A(:, :, p) = C(:, :, p) \times_e G$.

Tensorii $A(:, :, p)$ și $C(:, :, p)$ sunt de fapt matrice, notate cu A_p și, respectiv C_p . Deci $A_p(:, e)$ este imaginea unei persoane p în expresia e și coloanele matricei C_p formează o bază de vectori pentru persoana p . Obținem că $A_p = C_p G^T$, $p = 1, 2, \dots, n_p$.

Fie $G^T = (g_1 \dots g_{n_e})$, atunci $A_p(:, e) = C_p g_e$. Prin urmare g_e , $e = 1, 2, \dots, n_e$ sunt coordonatele imaginii $A_p(:, e)$ a persoanei p în expresia e în baza menționată anterior.

Fie $z \in \mathbb{R}^{n_i}$ o imagine din mulțimea de testing. Vrem să vedem dacă imaginea este identificată corect. Pentru acest lucru, trebuie să rezolvăm $\min_{\alpha_p} \|C_p \alpha_p - z\|_2$. Pentru fiecare imagine z trebuie să rezolvăm n_p probleme în sensul celor mai mici pătrate cu $C_p \in \mathbb{R}^{n_i \times n_e}$. Din $C = S \times_i F \times_p H$ obținem că $C_p = F B_p$, unde $B_p \in \mathbb{R}^{n_e n_p \times n_e}$, $B_p = (S \times_p H)(:, :, p)$.

Putem trunchia tensorii și matricele astfel încât să obținem o descompunere HOSVD trunchiată pentru tensorul A . Fie $F_k = F(:, 1:k)$ și obținem $\hat{C} = (S \times_p H)(1:k, :, :) \times_i F_k$. Deci, trebuie să rezolvăm $\min_{\alpha_p} \|\hat{C}_p \alpha_p - z\|_2$. Algoritmul propus este următorul, pentru $n_i \ll n_e n_p$.

Algoritmul pentru tensori - A2:

```

Fie  $z$  imaginea pe care o căutăm
for  $p = 1, 2, \dots, n_p$ 
    Fie  $\hat{C} = (S \times_p H)(1:k, :, :) \times_i F_k$ 
    Se rezolvă  $\min_{\alpha_p} \|\hat{C}_p \alpha_p - z\|_2$ 
    for  $e = 1, 2, \dots, n_e$ 
        dacă  $\|\alpha_p - g_e\|_2 < tol$ , atunci este persoana  $p$  și ne oprim
    end
end

```

Observația 7 Nivelul de trunchiere k poate fi ales ca și în [42] sau [10] sau [14] și nu empiric cum se sugerează în [13] (paginile 116 și 173).

5 Experimente

În acest capitol sunt prezentate experimente și rezultate obținute cu algoritmi prezențați în Capitolele 1, 3 și 4. Acest capitol cuprinde următoarele secțiuni.

5.1 Experimente cu generarea matricei de scanare în geotomografia electromagnetică

5.2 Experimente cu algoritmul SART

5.2.1 Paralelizarea algoritmului SART

5.3 Experimente cu clase de algoritmi pentru recunoașterea formelor

5.3.1 Experimente cu clase de algoritmi pentru recunoașterea fețelor

5.3.2 Studiu comparativ al algoritmilor NN, PCA și COD-A1 în recunoașterea formelor

Bibliografie

- [1] Andersen A. H., Kak A. C., *Simultaneous Algebraic Reconstruction Techniques (SART): a superior implementation of the ART algorithm*, Ultrasonic imaging, **6** (1984), 81–94.
- [2] Björk A., *Numerical methods for least squares problems*, SIAM Philadelphia, 1996.
- [3] Censor Y., Stavros A. Z., *Parallel optimization: theory, algorithms and applications*, Numer. Math. and Sci. Comp. Series, Oxford Univ. Press, New York, 1997.
- [4] Chen J., Saad Y., *Lanczos Vectors versus Singular Vectors for Effective Dimension Reduction*, IEEE Transactions on Knowledge and Data Engineering, **21(8)** (2009), 1091-1103.
- [5] Cimmino G., *Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari*, Ric. Sci. progr. tecn. econom. naz. **1** (1938), 326-333.
- [6] Dax A., *Orthogonalization Via Deflation: A Minimum Norm Approach for Low-Rank Approximations of a Matrix*, Journal of Computational and Applied Mathematics, **234(11)** (2010), 3091-3103.
- [7] Dax A., *A minimum norm approach for low-rank approximations of a matrix*, SIAM Journal on Matrix Analysis and Applications, **30(1)** (2008), 236-260.
- [8] De Lathauwer L., de Moor B., Vandewalle J., *A multilinear singular value decomposition*, SIAM Journal Matrix Anal. Appl, **21** (2000), 1252-1278.
- [9] Delac K., Grgic M., Grgic S., *Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set*, International Journal of Imaging Systems and Technology, **15(5)** (2006), 252-260.
- [10] Delac K., Grgic M., Liatsis P., *Appearance-based Statistical Methods for Face Recognition*, Proceedings of the 47th International Symposium ELMAR-2005 focused on Multimedia Systems and Applications, Zadar, Croatia, 2005, 151-158.
- [11] Dines K. A., Lytle R. J., *Computerized geophysical tomography*, Proc. IEEE, **67** (1979), 1065-1073.
- [12] Duda R.O., Hart P.E., Stork D.G., *Pattern Classification, second ed.*, Wiley-Interscience, New York, 2001.
- [13] Elden L., *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, Philadelphia, 2007.

- [14] Glineur F., Lu L., Van Dooren P., Wang X., *Extended Lanczos Bidiagonalization for Dimension Reduction in Information Retrieval*, acceptată pentru publicare în Proceedings of the 8th International Conference on Natural Computation (ICNC'12), Maui, Hawaii, 2012.
- [15] Golub G., Van Loan C., *Matrix Computation, 3rd Edition*, The John Hopkins University Press, Baltimore 1996.
- [16] Grecu L., Popa C., *Constrained SART algorithm for invers problems in image reconstruction*, Inverse Problems and Imaging, 1(7) (2013), 199-216.
- [17] Grecu L., Nicola A., *Some results on simultaneous algebraic techniques in image reconstruction from projections*, ROMAI Journal, 2 (2009), 79-96, <http://rj.romai.ro/>
- [18] Grecu (Liță) L., *On the construction of the scanning matrix in Electromagnetic Geotomography*, Buletinul Științific al Universității "Politehnica" din Timișoara, Seria Matematică - Fizică, 1 (2012), 42-56.
- [19] Grecu (Liță) L. , Pelican E., *Systematic and comparative experiments with some algorithms for pattern recognition*, acceptată pentru publicare în volumul conferinței Nineth Workshop on Mathematical Modelling of Environmental and Life Sciences Problems, November 1-4, 2012, Constanța, Romania.
- [20] Grecu (Liță) L. , Pelican E., *Customized Orthogonalization via Deflation Algorithm with Applications in Face Recognition*, trimisă spre publicare la Carpathian Journal of Mathematics, 2013.
- [21] Grecu (Liță) L. , Pelican E., *A Low-Rank Tensor-Based Algorithm for Face Recognition*, trimisă spre publicare la Pattern Recognition Letters, 2013.
- [22] Guo G.-D., Zhang H.-J., Li S.Z., *Pairwise Face Recognition*, Proc. of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, 2 (2001), Vancouver, Canada, 282-287.
- [23] Guo G.-D., Zhang H.-J., *Boosting for Fast Face Recognition*, Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, RATFG-RTS'01, Vancouver, Canada, 96-100.
- [24] Herman G. T., *Image reconstruction from projections. The fundamentals of computerized tomography*, Academic Press, New York, 1980.
- [25] Horn R. A., Johnson C. R., *Matrix analysis*, Cambridge University Press, New York, 1990.

- [26] Hotelling H., *Analysis of a complex of statistical variables into principal components*, J. Educ. Psychol., **24** (1933), 417-441, 498-520.
- [27] Jiang M., Wang G., *Convergence of the simultaneous algebraic reconstruction technique (SART)*, in Proc. 35th Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA, 2001, 360-364.
- [28] Jiang M., Wang G., *Convergence studies on iterative algorithms for image reconstruction*, IEEE Trans. on Medical Imaging, **22** (2003), 569-579.
- [29] Jolliffe I.T. , *Principal Component Analysis*, Springer-Verlag, Second Edition, New York, 2002.
- [30] Kaczmarz S., *Angenaherte Auflosung von Systemen linearer Gleichungen*, Bull. Acad. Polonaise Sci. et Lettres **A** (1937), 355-357.
- [31] Kirby M., Sirovich L., *Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **12**(1) (1990), 103-108.
- [32] Koltracht I., Lancaster P., Smith D., *The structure of some matrices arising in tomography*, Linear algebra and its applications, **130** (1990), 193-218.
- [33] Koltracht I., Lancaster P., *Constraining Strategies for Linear Iterative Processes*, IMA Journal of Numerical Analysis, **10** (1990), 555-567.
- [34] Martinez A.M., Kak A.C., *PCA versus LDA*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **23**(2) (2001), 228-233.
- [35] Meyer C. D., *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
- [36] Murty M. Narasimha, Devi V. Susheela, *Pattern Recognition. An Algorithmic Approach*, Springer, First Edition, London, 2011.
- [37] Natterer F., *The Mathematics of Computerized Tomography*, John Wiley and Sons, New York, 1986.
- [38] Nicola A., Petra S., Popa C. and Schnörr C., *On a general extending and constraining procedure for iterative methods*, Intern. Journal of Computer Mathematics, **89**(2) (2012), 231-253.
- [39] Pan X., Sidky E. Y., Vannier M., *Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?*, Inverse Problems, **25**(12) (2009).
- [40] Pearson K., *On lines and planes of closest fit to systems of points in space*, Phil. Mag., **2**(6) (1901), 559-572.

- [41] Pelican E., Grecu L., *Comparison Between Some Matrix Methods with Applications in Pattern Recognition*, prezentată la Applied Linear Algebra Conference, May 24-28, 2010, Novi Sad.
- [42] Pelican E., Grecu L., *Low-Rank Matrix Methods in Pattern Recognition*, prezentată la Balkan Conference on Operational Research (BALCOR), 2009, publicată electronic cu ISBN: 973-86979-9-9.
- [43] Pelican E., Grecu L., *Solving the Pattern Recognition Problem with some Low-Rank Approximation Based Algorithms*, prezentată la XIème Colloque Franco-Roumain de Mathématiques Appliquées, 24-30 August, 2012, Bucureşti.
- [44] Pentland A., Starner T., Etcoff N., Masoiu A., Oliyide O., and Turk M., *Experiments with Eigenfaces*, prezentată la Looking at People Workshop, IJCAI '93, Franţa, 1993.
- [45] Popa C., *Projection algorithms, classical results and developments. Applications to image reconstructions*, Lambert Academic Publishing - AV Akademikerverlag GmbH & Co. KG, Saarbrücken, Germany, 2012.
- [46] Popa C., *Constrained Kaczmarz extended algorithm for image reconstruction*, Linear Algebra and its Applications, **429** (2008), 2247-2267.
- [47] Popa C., *Extensions of block-projections methods with relaxation parameters to inconsistent and rank-deficient least-squares problems*; B I T, **38(1)** (1998), 151-176.
- [48] Popa C., *A hybrid Kaczmarz-Conjugate Gradient algorithm for image reconstruction*, Mathematics and Computers in Simulation, **80** (2010), 2272-2285.
- [49] Sirovich L., Kirby M., *Low-dimensional Procedure for the Characterization of Human Faces*, Journal of the Optical Society of America A - Optics, Image Science and Vision, **4(3)** (1987), 519-524.
- [50] Tanabe K., *Projection Method for Solving a Singular System of Linear Equations and its Applications*, Numer. Math., **17** (1971), 203-214.
- [51] Turk M., Pentland A., *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, **1** (1991), 71-86.
- [52] Turk M., Pentland A., *Face Recognition using Eigenfaces*, Computer Vision and Pattern Recognition Proceedings CVPR '91, 1991, 586-591.
- [53] http://www.math.ucf.edu/~xli/SVD_PCA.pdf