Revision History

Date/Reviser	Comment
Mon, 2 Jan 2012 Alexandru Mihail	*Inceput. Problema recunoasterii imaginilor*
Mon, 23 Jan 2012 Alexandru Mihail	*Prima versiune gata de printare cu sectiunile: Abstract, Nevoia, Normalizarea, Clasificarea si descompunerea imaginilor, Parametrarea aparatului.*

Contents

Revision History 1
1 ABSTRACT 2
2 NEVOIA 2
2.1 Imaginea ca limbaj 3
2.2 Recunoasterea imaginilor 3
2.3 O situatie tipica: inregistrarea unei carti de identitate 3
2.4 Corpus 4
3 PRELIMINARII 6
3.1 Imagine raster, definitie 6
3.2 Spatiul de culoare RGB 6
3.3 Semnal si zgomot in imagini 7
3.4 Imaginea vectoriala ca populatie de studiu statistic 7
4 METODA PROPUSA PENTRU NORMALIZAREA IMAGINILOR 8
4.1 Eliminarea ramei absolut negre 8
4.2 Iteratorul de pixeli 8
4.3 Eliminarea punctelor albe si reductiunea celor negre 9
4.4 Identificarea culorii dominante 9
4.5 Vectorizarea imaginii 11
4.6 Rezultatul vectorizarii 11
4.7 Estimarea inclinatiei imaginii 11
4.8 Iterator convergent spre unghiul de inclinatie al imaginii 16
4.9 Determinarea noilor dimensiuni ale imaginii 17
4.10 Aducerea la scara fixa a imaginii vectoriale. 17
4.11 Codul de Deskew 18
5 METODA PROPUSA PENTRU CLASIFICAREA SI DESCOMPUNEREA IMAGINILOR 21
5.1 Disimilaritatea imaginilor 21
5.2 Clasificatori de imagine 22
5.3 Masti de decupaj 22
5.4 Algoritmul de constituire a claselor optime 23
5.5 Exemplu de constituire a ierarhiei claselor 28
6 PARAMETRAREA APARATULUI 30
7 ANEXE 31
8 BIBLIOGRAFIE 31

1 ABSTRACT

In acest document este prezentat un dispozitiv de indreptare, clasificare si descompunere a imaginilor digitale provenite din scanare sau din fotografiere. Aparatul isi are aplicabilitatea in extragerea textului prin opto-caractero-recunoastere din documente de business de tipul contract, factura, aviz de insotire a marfii, document de transport, sau documente de identificare a persoanei de tip buletin sau pasaport. Procedeul specifica o serie de pasi necesari intre admisia documentelor in coada de asteptare pana la decupajul final dupa sabloanele corespunzatoare fiecarui tip de document. Intr-o prima faza aparatul separara fiecare imagine in prim-plan si fundal prin detectia tonului dominant folosind tehnici statistice de clasificare. Se vectorizeaza apoi partea de prim plan a imaginii cu pastrarea valorii aproximative a tonului de fundal. Se trece la faza de stabilizare a imaginii prin detectia unghiului de cea mai probabila inclinatie definind si utilizand un estimator statistic si o metoda de estimare iterativa convergenta. Se roteste imaginea potrivit cu unghiul detectat – faza cunoscuta ca *deskewing*, se determina volumul de selectie al imaginii normate si centrate cu eliminarea marginilor de gradul 2 - neconcludente. Intr-o faza de antrenare a aparatului in care se acumuleaza statistici referitoare la geometriile imaginilor corpus, se determina clasele in care documentele de intrare pot fi cel mai bine separate dupa criterii configurabile precum geometria paginii, raportul culorilor, prezenta sau absenta codurilor de bara sau alte detalii extrase din imagine. In faza de configurare se creaza mastile de decupaj dupa care, in ultima faza, imaginile sunt descompuse recursiv in subimagini pana la factorii primi care sa fie dati in analiza etajelor inferioare de OCR sau de detectie faciala.

2 NEVOIA

Una din ambitiile erei digitale a fost eliminarea hartiei din raporturile interumane si inlocuirea acesteia cu documente electronice mai versatile si mai transmisibile. Victima clara a fost desigur scrisoarea. In business s-a urmarit crearea birourilor “green” fara hartie unde comunicarea sa fie integral electronica. S-a constatat ulterior ca acest “paperless office” genereaza de cateva ori mai multa hartie tiparita decat predecesorul sau. Care este asadar situatia hartiei in momentul de fata? Paradoxal, intreprinderile si institutiile continua sa genereze si sa stocheze in arhive cantitati mari de documente pe suport hartie. Motivul este cel al asumarii raspunderii de catre persoane prin semnatura asupra continutului acestora, fapt care confera hartiei valoare de proba. Intre un furnizor si un client nu este suficienta transmiterea marfii daca aceasta nu este insotita de avizul sau factura semnata si contrasemnata care sa legitimeze schimbul de proprietate sau posesie asupra bunurilor. Comertul este in continuare un mare producator de hartie tiparita de toate tipurile. La fel este si statul in raporturile sale cu institutiile, cu intreprinderile private si cu populatia, comunicand exclusiv prin documente scrise de tip lege, ordin, decizie, dar si bilant, situatie financiara, audit, etc.

Documentul de tip imagine bitmap, tiff, jpeg, pdf, etc, este astazi forma generalizata a scrisorii de hartie de cele mai multe ori nascandu-se ca atare, sau provenind din scanarea unor originale de hartie. Mari producatori de imagini au devenit sistemele electronice insele care genereaza si transmit utilizatorilor informatie in format imagine. Sistemul informatic este astazi clasa cea mai importanta de producator de documente, motivat de nevoia de a-si proteja modelul datelor in spatele prezentarii si de a transmite utilizatorului doar forma datelor nu si structura lor. Hartia a metamorfozat asadar in forme electronice, dar ramane conceptual aceeasi iar rolul sau social nu s-a schimbat. Destinatarul se confrunta cu aceeasi problema ca si pana acum cu hartia scrisa, aceea a intelegrii formei si a interpretarii continutului, compusa insa acum dramatic de diversitatea si de volumul acestor documente. Dupa acelasi model in care sistemele electronice insele sunt cele mai mari producatoare de documente, este firesc ca rolul intelegerii acestora sa le revina tot lor.

2.1 Imaginea ca limbaj

O transmisiune generala completa intre expeditor si destinatar porneste cu alegerea de catre expeditor a unei portiuni din reteaua sa semantica si serializarea/depunera acestuia intr-un mediu de comunicare intr-o anume forma de prezentare convenita cu destinatarul, urmata la celalalt capat de ridicarea/deserializarea acestei prezentari de catre destinatar din mediul de comunicare si recunoasterea/acceptarea mesajului in propria sa semantica:

modelul expeditorului ⟶ prezentare ⟶ transmisie ⟶ recunoastere ⟶ modelul destinatarului

Transmisiunea intre doi poli mai poate fi definita si in sistemul limbajelor formale ca secventa: generare, sir, compilare. unde sirul generat, acelasi cu banda de intrare a compilatorului sunt simboluri apartinand unui mediu de transmisie ales. Suntem obisnuiti in domeniul limbajelor cu o banda de iesire/intrare textuala – alfabet. Generalizam notiunea de banda de intrare sa includa si imaginea. Aparatul prezentat este preporcesorul de imagini al oricarui compilator, acel etaj al intelegerii care transforma datele crude desprinse din imagini in simboluri asimilabile si reductibile de catre compilator.

2.2 Recunoasterea imaginilor

In lucrarea de fata ne vom ocupa de problema recunoasterii imaginilor in format electronic si vom prezenta un aparat logico-matematic si algoritmul corespunzator implementabil pe orice calculator cu procesor, memorie si interfata de intrari-iesiri care reduce imaginile de intrare la sabloane prestabilite si/sau determinate printr-un proces de invatare.

Responsabilitatile acestui preprocesor sunt multiple. Principala sa problema, este descompunerea imaginii in bucati mai mici dupa un criteriu de separare. Acest proces este cunoscut in teoria compilatoarelor drept analiza lexiacala si are ca rezultat formarea multimii de simboluri – tokens - pe care etajul superior al compilatorului sa-l poata analiza potrivit cu o gramatica data. Daca token-izarea in banda textuala este o chestiune elementara: orice nu e blank apartine simbolului curent, dupa blanc incepe alt simbol, token-izarea in imagine devine o problema deosebit de convoluta si de alambicata tehnic ce tine prelucrarea semnalelor si de statistica necesare pentru identificarea frontierelor de separare intre simboluri. La originea problemei sta caracterul nevectorial al imaginii si saturarea acesteia cu puncte de fundal care trebuiesc eliminate in faza de preprocesare. Totodata, imaginea este in general denaturata de roto-translatiile scanarii si trebuie normalizata. Imaginea are o componenta de zgomot adaugat sub forma de spary de pixeli sau de contururi false care trebuie de asemenea identificat si eliminat.

In ultima faza a aparatului prezentat se constituie clasele de echivalenta in care imaginile prezentate sunt ierarhizate. Imaginile adunate in aceste clase sunt indivizii cu deviatie minima fata de individul sablon care prin reprezentare le poate descompune in subimagini-camp. Campurile sunt rezultatul ultim al acestei preprocesari, acestea putand fi tratate in continuare printr-un motor de opto-caractero-recunoastere care sa extraga textul coresounzator.

2.3 O situatie tipica: inregistrarea unei carti de identitate

Filialele de banca tin in fiset dosare de hartie cu date despre clientii lor. Documentul cheie, cap de dosar, este copia scanata dupa buletinul clientului obtinuta la deschiderea contului. La fiecare vizita, clientul se legitimeaza cu buletinul iar banca verifica in baza de date a bancii numele si CNP-ul pentru a localiza contul. Imaginile buletinelor raman in arhiva fizica pentru consultare ocazionala.

Contul unui client nou se deschide printr-un efort manual de data-entry in care operatorul preia datele de identificare si populeaza campurile unei structuri prestabilite in baza de date.



$\rho \longrightarrow$

```
struct BULETIN
{
   Char SERIA[2];
   Int NR;
   Char CNP[13];
   Char*Nume;
   Char*Prenume;
   Char*Tata;
   Char*Mama;
   Char Sex;
   Char*Loc nastere;
   Char*Domiciliu;
   Char*Emisa de;
   DateTime Valabilitate[2];
   Char C1[3];
   Char C2;
   Char C3[5];
   Char*TEXT;
   Void*PIC;
};
```

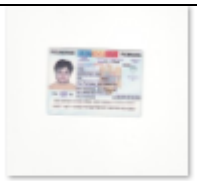
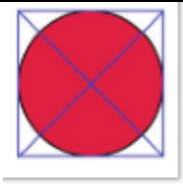
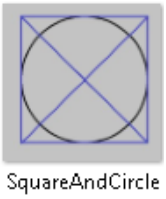
Situatia este similara in Sanatate, unde orice raport intre pacient si institutie incepe prin identificarea cu buletinul. Probleme deosebite incep sa se puna si legat de validitatea unui asemenea document care poate fi in unele situatii chiar... fals. Problema documentelor de identitate se pune la adevarata sa amploare la punctele de control de frontiera unde sistemele electronice trebuie sa inteleaga si sa clasifice cele peste 150 de tipuri de ID folosite in Uniunea Europeana.

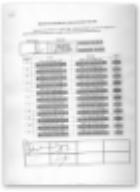
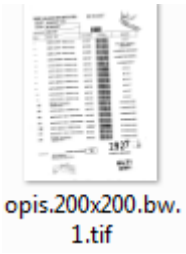
Daca in cazul deschiderii unui cont la banca functia ρ era indeplinita de catre operator “a mano”, pentru alte aplicatii este necesara automatizarea acesteia. Obiectivul aparatului prezentat este tocmai automatizarea acestei functii de recunoastere **ρ**.

2.4 Corpus

Pe langa multitudinea de formate in care imaginile trebuiesc admise in sistem, toate pana la urma reductiblie la bitmap necomprimat RGB, sunt posibile o serie de moduri ale acestora care le indeparteaza sau apropie de o forma ideala. Ieala este imaginea asa cum a produs-o emitentul. In practica, insa imaginea este alterata de transmisie si retransmisie.

Matricea T a imaginilor de test – introdusa in scopul de a rationaliza referintele acestui document – demonstreaza unele adversitati pe care aparatul propus va trebui sa le circumvina:

T	1	2	3	4
1	 Buletin Alexander 2.bmp	 Buletin Alexander 0.bmp	 Buletin Anda0.bmp	 Buletin Anda.tif
2	 Buletin Mamo_Col.tif	 Circle24red.bmp	 SquareAndCircle.bmp	 SquareAndCircle 24.bmp

3	 av14s300.tif	 av14s400.tif	 AVIZ ORIGINAL.tif	 fact_header.300x300xq1_bad.tif
4	 Decizie.400x400xg_unrecognized.tif	 Self 1.bmp	 opis.200x200.bw.1.tif	 CircleAndLine24red.bmp

Buletinul T_{11} ar fi un caz ideal din punct de vedere geometric pentru ca este scanat in conditii optime de aliniament la marginile scannerului. Imaginea are exact limitele buletinului fara vreo rama alba sau neagra. Problema acestei imagini este insa rezolutia mare de culoare cu zeci de mii de nuante in spatiul RGB. T21 este un buletin ortonormat, care de asemenea sufera de adancime mare de culoare. O problema importanta, dupa cum vom vedea, este ca nuanta dominanta de fundal difera de la T11 la T21, si in general de la o imagine la alta. Nuanta de fundal nu poate fi prezumata ci trebuie cautata. Parte din metoda prescrisa este, asadar, detectia polilor de culoare ai imaginii.

T12 este o situatie grava in care pe langa geometria inclinata, adancimea mare de culoare si tonul de fundal diferit, imaginea mai are si rama neagra care trebuie exclusa din calcule. Fiind inclinata, imaginea are si rama inclinata. Unghiul de inclinatie al pozei nu este cel al ramei din T12 dupa cum demonstreaza T14. Detectia si eliminarea ramei absolut negre este primul pas al prezentei metode.

La polul opus este imaginea T14 care este o copie la a doua speta a unui buletin. T14 este un scan dupa un scan, situatie pe care orice sistem electronic de recunoastere o va refuza din start, dar pe care noi o vom accepta cu mandrie. imaginea are colturi total negre provenite de la dispozitivul optic de scanare.

T13 este o situatie tipica de copie de buletin in care operatorul a centrat oarecum originalul in scanner dar nu a putut asigura ortogonalitatea ideala. T13 demonstreaza necesitatea identificarii ramei constant albe a imaginii pe care o consideram zgomot.

Imaginile T31, T32, T33, T34 sunt avize sau facturi emise de societatea RomSoft SA in doua perioade istorice diferite. T31 si T32 fiind documente “noi” iar T33 si T34 documente “vechi”. T41 este un act administrativ al societatii RomSoft SA. Documentul are geometrie diferita de facturi, este inclinat si a fost scanat in conditii de rezolutie prea bune astfel incat fundalul imaginii a fost asimilat ca un dégradé de griuri centrate in jurul unui ton de culoare altul decat alb. Imaginea este un exemplu de efect advers al sacanrii imaginilor la rezolutie inalta.

Imaginile T22, T23, T24, T44 sunt produse de sinteza folosite in testele de deskewing ale aparatului prezentat. T22 demonstraza invarianta la roto-proiectie. Datorita simetriei, T23 si T24 prezinta maxime multiple in algoritmul de cautare a unghiului de inclinatie. Imaginile sunt utile in a demonstra tehnica de separare a imaginii in cele 3 multimi cautate: background, edge si bulk.

T42, in fine, este o fotografie folosita in testarea si explicarea algoritmului convergent de calcul al nuantei dominante intr-o imagine.

3 PRELIMINARII

3.1 Imagine raster, definitie

O definitie exacta a imaginii nu poate fi gasita in literatura. Unele obiecte precum lista, arborele, stiva, heapul, sunt definite mai degraba prin algoritmii care le utilizeaza decat printr-o formula analitica. Imaginea este in esenta o proiectie 2D a unei realitati pe placa fotografica. Mai precis,imaginea este structura de legatura intre planul euclidian si spectrul luminos care asociaza oricarui punct dintr-o arie rectangulara de proiectie o frecventa sau un cod de culoare.

Imaginea raster poate fi definita ca o structura

$$P: <w,h,C>,\ h,w\in N,\ C=\{c_1,c_2,...c_n\},\ n\in N,\ n=h*w,\ c_i\in RGB,\ RGB\subset N^3 \qquad (1)$$

unde w este latimea imaginii, h este inaltimea, iar c_i sunt vectori intr-un spatiu de culoare de de componente rosu, verde, si albastru. Rezulta din definitie urmatoarele:

1. Imaginea este o insiruire de vectori c_i intr-un spatiu de culoare RGB.
2. Exista o aplicatie de indexare $p:N^2->C$ astfel incat exista unic x,y cu $c_i = p(x,y)$, i=1,n. Aceasta aplicatie asociaza fiecarui punct c al imaginii puncte p din plan.
3. Imaginea este un sector compact, neexistand $1 \le i,j \le n$ pentru care $p(i,j) \notin C$.
4. Imaginea este un subset discret al spatiului de culoare RGB.
5. Imaginea este un mediu izotropic.

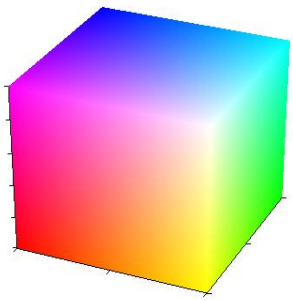
Imaginea P poate fi adusa la forma vectoriala ca multime de vectori in spatiul $R^2 \times RGB$ astfel:

$$V= \{(x,y,c),\ p(x,y)=c,\ x\in R,\ y\in R,\ c\in RGB\} \qquad (2)$$

Reprezentarea vectoriala a unei imagini este considerabil mai scumpa in ce priveste consumul de memorie intre cele doua reprezentari raportul fiind de 3*1 la 5*8. Cu unele conditii, reprezentarea vectoriala este insa preferabila pentru ca va permite aplicarea operatorilor liniari de roto-translatie, este o multime neordonata care permite reuniunea si disjungerea de multimi. Neordonarea acestei multimi face posibila descompunerea imaginilor vectoriale in sub imagini si recompunerea lor.

3.2 Spatiul de culoare RGB

Spatiul RGB este codat pe 3 directii rosu verde si albastru pe 256 de nivele de intensitate corspunzand intervalului real [0,1] de la negru la rosu, la verde si la albastru. Cubul unitate RGB este prezentat dinspre coltul de coordonate (1,1,1) reprezentand culoarea alba inspre originea (0,0,0) reprezentand culoarea neagra, invizibila in proiectie. Cubul RGB a fost produs in Matlab:



```
vertices = [0 0 0; 1 0 0; 0 1 0; 0 0 1; 1 0 1; 0 1 1; 1 1 0; 1 1 1];
faces = [1 2 7 3; 1 3 6 4; 1 4 5 2; 8 7 3 6; 8 7 2 5; 8 5 4 6; 2 3 4 2;];
alpha = [ 1; 1; 1; 1; 1; 1; 1; ];
patch('Faces', faces,...
      'Vertices', vertices,...
      'FaceColor','interp',...
      'FaceVertexCData', vertices,...
      'FaceAlpha', 'flat',...
      'AlphaDataMapping', 'none',...
      'EdgeColor', 'none',...
      'FaceVertexAlphaData',1);
view(70,25);
grid on;
set(gca,'XTickLabel',{},'YTickLabel',{},'ZTickLabel',{})
```

3.3 Semnal si zgomot in imagini

Imaginea poate fi considerata ca fiind un semnal RGB esantionat de w*h ori la rezolutie de 256 pe componentele de culoare. Esantioanele c_i pot fi corelate cu vectori in cubul unitate al spatiului RGB printr-o functie de mapare c:C⟶RGB. Purtatoarea semnalului este functia index p(x,y). Semnalul este p•c:N^2⟶RGB compunerea celor doua functii. Semnalul imaginii este suma dintre functia p de index cu rol de sync la grila de rasterizare a imaginii si un kernel de selectie in spatiul RGB. O discutie lamuritoare despre esantionarea si reconstituirea semnalelor prin modularea unui kernel cu o functie puls de sync este data in sectiunea “Sampling Theory” (James D. Foley, 1990) p623-642.

Postulam ca in fiecare imagine exista un semnal util care se evidentiaza din suma dupa un criteriu de utilitate care trebuie determinat. Semnalul inutil este zgomot. O imagine blanck este in intregime semnal inutilizabil - zgomot. In termeni de imagine, blanck-ul este culoarea de fundal. Neputand apriori specifica acest criteriu de utilitate a semnalului in imagini, spunem ca daca intr-o imagine o culoare se distinge ca fiind dominanta, atunci ea formeaza zgomot, iar semnalul util este tot ce ramane dupa eliminarea acesteia. Cunoscand culoarea dominanta c_0, putem recoda imaginea P ca multime de vectori in spatiul R^2×RGB astfel:

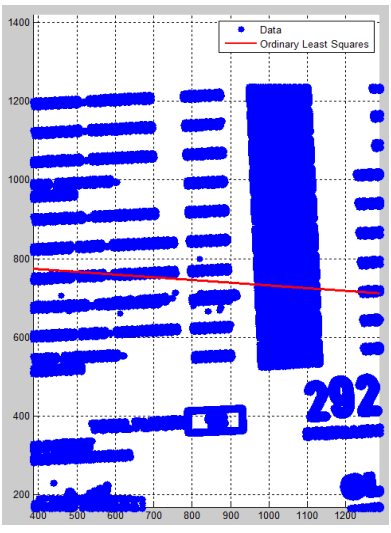
$$U= \{(x,y,c),\ p(x,y)=c,\ c \neq c_0,\ x \in R,\ y \in R,\ c \in RGB\} \qquad (3)$$

Multimea U este reprezentare vectoriala *utila* a imaginii P. V-U este asadar zgomot. Criteriul de utilitate ales poate fi inversat: utila este numai componenta de imagine care are culoarea dominanta, orice punct de alta culoare este inutil. In procedeul de fata, dorim sa pastram cat mai mult posibil din original intre limitele resurselor disponibile. Acest criteriu de utilitate se aplica cu succes imaginilor alb-negru, insa in cazul buletinelor T11, T12, etc. a renunta la toate punctele de culoare altele decat cele negre de interes in OCR ar insemna sa se renunte la fotografie. In metoda de fata, zgomot este culoarea dominanta. Eficienta in cazul imaginilor alb-negru se atinge prin parametrarea adecvata a nivelelor de tresholding pe care le vom discuta ulterior.

3.4 Imaginea vectoriala ca populatie de studiu statistic

Multimea U este o populatie de n indivizi care ocupa planul euclidian si spatiul de culoare. Indivizii x din U au proprietatile x,y,r,g,b, variabile in sens statistic. Pe aceasta populatie devin posibile studii de dispersie in E^2 sau in RGB, studiul covariantei, analiza factoriala, si regresia liniara.

Un exercitiu prealabil de logica sugereaza ca desi analiza componentelor principale ar putea explica corelatiile dintre x,y si culorile imaginii, nu poate raspunde la problema esentiala a identificarii unghului de rotatie al imaginii. Motivul este independenta inertiei obiectului fata de sistemul de referinta, inertia fiind o masura intrinseca a naturii masei si a geometriei obiectului.



Un alt demers plauzibil este cel al regresiei liniare asupra norului de puncte individ din U. S-a luat in studiu in Matlab si Statistica imaginea T43 si s-a obtinut linia de regresie cu o inclinatie de -3.9° care nu reflecta ungiul de inclinatie al paginii ci este de asemenea o masura intrinseca a obiectului. Elementul de “seductie” in aceasta abordare a constat in posibilitatea de a fi trecut prin aparat o serie de documente ideale pentru a le afla dreapta de regresie, urmand ca orice alt document sa devieze de la prototip prin ungiul de inclinatie in pagina. Esecul abordarii consta in dependenta dreptei de regresie de datele din document nu de tipul acestuia. Daca acest opis are 11 randuri iar altul ar avea 5, dreptele lor de regresie vor fi diferite.

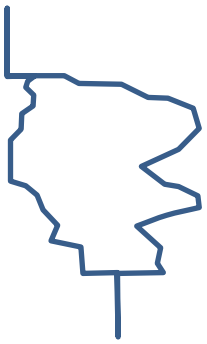
4 METODA PROPUSA PENTRU NORMALIZAREA IMAGINILOR

Analiza imaginilor se face in urmatorii pasi:

1. Admisia si deschidera imaginii.
2. Vectorizarea si normarea imaginii.
    a. Identificarea si eliminarea ramei absolut negre din imaginile C11, C14, C21, C22 cu formarea *membranei*.
    b. Determinarea culorilor folosite in imagine prin utilizarea histografului predeterminat.
    c. Determinarea culorilor dominante in imagine.
    d. Vectorizarea imaginii.
    e. Gasirea unghiului de inclinatie al imaginii.
    f. Rotirea imaginii vectoriale cu unghiul gasit
    g. Determinarea noilor dimensiuni ale imaginii.
    h. Aducerea la scara fixa a imaginii vectoriale.
3. Inchiderea imaginii originale.
4. Clasificarea imaginii vectoriale:
    a. cautarea disimilaritatii minime fata de clasificatorii configurati.
    b. reducerea imaginii la clasa proxima.
    c. analiza clusterilor optimi
5. Aplicarea grilelor de decupaj specificate per clasa in etajele inferioare ale OCR-ului textual.

Algoritmul este monitorizat in punctele sale cheie prin diverse mijloace de diagnostic incluzand crearea de matrici temporare care pot fi exportate catre Statistica si Matlab.

4.1 Eliminarea ramei absolut negre



Procesarea incepe prin presupunerea ca spatiul pixelilor este dat de dimensiunile declarate ale imaginii. Dreptunghiul pixel_space este deschis la maximum intre punctele (0,0) si (w,h) ale imaginii, apoi este progresiv restrans pe masura ce sunt identificate criterii noi de clipping. Imaginile T12, T14, T31 si T32 au rama neagra sau colturi negre care trebuiesc eliminate.
Iterand peste imagine, de sus in jos, de jos in sus, de la stanga la dreapta si de la dreapta la stanga se numara liniile negre, daca acestea exista, si se reduc progresiv limitele dreptunghiului pixel_space pana ce rama neagra este exclusa. Imaginea ajunge acum intr-o situatie de tip T31 cu colturi triunghiulare negre. Acestea se elimina configurand vectorii SL si SR (Size limit Left si Right) care masoara si retin pozitia primilor pixeli de culoare alta decat negru de la stanga si respectiv dreapta. Volumul de selectie este marginit de segmentul neregulat, potential concav [SL,SR] din plan numit *membrana* a imaginii. Dreptungiul pixel_space este mai departe redus sa incadreze minim aceasta figura.

4.2 Iteratorul de pixeli

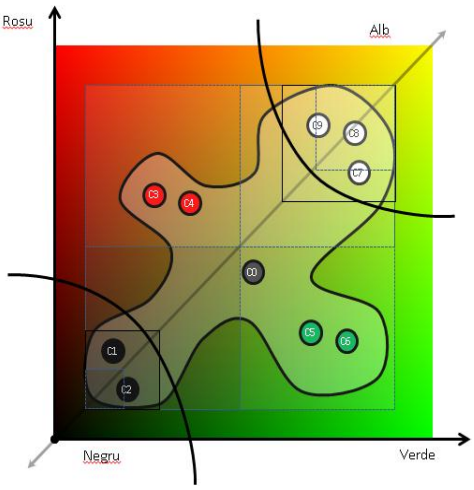
Calul de bataie al procesarii pana la obtinerea imaginii vectoriale este iteratorul de pixeli. Este o bucla dubla pe verticala si-apoi pe orizontala intre limitele membranei care permite abstractizarea operatiilor de cliping atat in spatiul XY cat si in spatiul RGB. Rutina poate face o eliminare triviala a culorii albe sau aflate intr-o raza configurabila de alb, si poate deasemenea marca prin aliasare pixelii negri sau intr-o proximitate configurabila a culorii negre. Rutina ascunde totodata detalii privind reducerea rezolutiei de culoare a imaginii. Este utilizata in fazele de preprocesare dupa modelul:

```
FOR_PIXELS
 pixel_count++;
END_FOR_PIXELS
```

```
#define FOR_PIXELS          \
 for (p.y=pixel_space.A.y;      \
    p.y<=pixel_space.B.y;       \
    p.y++)              \
 {                      \
  int L=max(SL[p.y],pixel_space.A.x);       \
  int R=min(SR[p.y],pixel_space.B.x);       \
  for (p.x=L;p.x<=R;p.x++)      \
  {                     \
   int argb=GET_PIX(p.x,p.y);\
   point3D color=RGB2POINT(argb);\
   color.x&=color_mask;                     \
   color.y&=color_mask;    \
   color.z&=color_mask;    \
   if (color.x>brightness&&
            color.y>brightness
            &&color.z>brightness) \
    continue;           \
   if (color.x<darkness&&
            color.y<darkness&&
            color.z<darkness) \
    color=black;        \
   argb=POINT2RGB(color);
#define END_FOR_PIXELS      \
  } }
```

4.3 Eliminarea punctelor albe si reductiunea celor negre

Acest pas este esential in separarea semnalului util de zgomot si de determinare a unui criteriu de diferentiere in imagine. Pana la separare, nu putem spune despre corpul din imagine mai mult decat ca ocupa segmentul de forma [SL,SR] daca are rame negre, sau omogen, intreg spatiul incadrat de dreptunghiul pixel_space. Obiectul are insa forma in spatiul RGB prin culorile pe care le utilizeaza. Ilustram o sectiune rosu-verde in cubul unitar RGB:



La originea axelor este negru iar in coltul (1,1,0) este galben. In coltul (1,1,1) al cubului este alb. Cele doua coarde din jurul punctului de negru si celui de alb sunt tolerantele configurabile de intuneric si lumina.

Punctele de lumina vor fi asimilate albului absolut iar cele de intuneric vor fi asimilate negrului. Punctele intermediare rosii/verzi/albastre vor ramane nealterate.

Punctele de alb absolut vor fi eliminate din imagine ca zgomot. Punctele de negru sunt pastrate si marcate ca prim-plan.

Imaginea se descompune astfel in trei sub-multimi: Bg, Fg, si Mg, background, foreground si middleground. din care se retin ultimele doua. Separarea nu se face efectiv ci doar la nivelul iteratorului care va sari peste Fg pana la vectorizarea imaginii.

Aceasta metoda elimina rapid fundalul alb insa nu rezolva problema generala cand fundalul nu este alb ci gri cum este cazul imaginii T41, albastruie cum e cazul buletinelor T21 si T12, sau a imaginilor de sinteza in care culoarea dominanta este rosie sau gri 50%. Daca s-ar largi pragul de luminozitate sa acopere aceste nuante, atunci s-ar reduce in mod inacceptabil toate culorile care intra in volumul de taiere pierzandu-se detalii potential importante ale imaginii. Vom pastra eceste praguri de intuneric si lumina la valori minime pentru a nu compromite imaginile color.

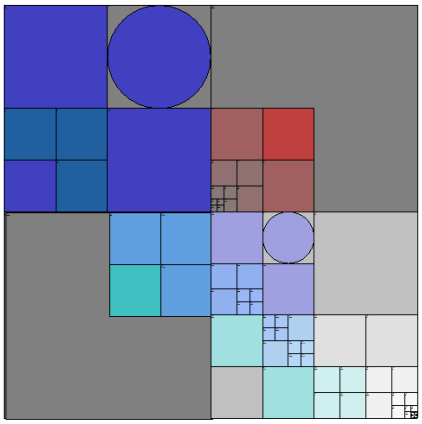
4.4 Identificarea culorii dominante

Separarea semnal/zgomot se poate face numai dupa ce s-au contorizat contributiile populatiei de puncte de imagine la culorile spectrului. Analiza spectrala se face prin doua metode succesive cu performante si rezultate diferite.

Prima metoda este un clasificator cu histograma care parcurge imaginea o singura data si duce fiecare individ in galeata potrivita. Algoritmul are complexitatea O(aria([SL,SR])*$\mathcal{C}$(C)) unde $\mathcal{C}$(C) este cardinalitatea multimii culorilor utilizate in poza. Maximum acestei cardinalitati este insasi rezolutia imaginii, adica aria([SL,SR]). In cazul cel mai nefavorabil, algoritmul are complexitatea patratului numarului de puncte din imagine. Algoritmul are o complexitate timp neglijabila pentru imagini cu numar mic de culori, dar se degradeaza simtitor odata cu cresterea adancimii de culoare. Problema sa este si complexitatea de memorie care pentru imagini mari color devine inacceptabila. Imagine buletinului T13, de exemplu are 1573*1489 = 2342197 puncte in spatiul RGB, toate potential diferite, prcatic in jur de 80,000 dupa cum arata testele. Pentru aceasta imagine trebuie rezervate 80,000 de galeti de contorizare iar complexitatea timp fata de procesarea unei imagini cu adancime de culoare de 16 creste cu un factor de 5000!. Pentru gestionarea unui numar nedeterminat de culori algoritmul foloseste o lista inlantuita de perechi (culoare,contor) alocata pe stiva. Pentru fiecare lista se parcurge lista in cautarea culorii. In cazul fericit cand culoarea este gasita, se incrementeaza contorul. Lista este mentinuta ordonata top down, astfel ca la fiecare

asemenea increment, algorimul urca pe lista sa verifice daca nodul nu cumva trebuie dus mai sus in fata unui dren de culori cu contor egal cu valoarea avuta inainte de increment. In cazul cel mai nefericit, lista este testata exhaustiv si un nod nou este creat si legat in coada pentru culoarea ultimului pixel. Daca histograful este abandonat inainte de final, rezultatele deja acumulate sunt neconcludente chia si la nivel de tendinta, neputandu-se spune despre vre-o culoare ca ar fi dominanta. Este totusi posibil ca la jumatatea buclei sa se decida ca numai prima culoare are sansa de a fi dominanta, si este prin urmare cea dominanta. S-ar reduce numarul de calcule la ½ cu pretul nefinalizarii contorizarilor, obtinandu-se un castig nesemnificativ in timp. Acest algoritm poate fi rulat insa cu o limita de 256 de culori care, intr-un caz favorabil cu mai putine culori, sa produca statistici complete de culoare, cu riscul de a iesi prematur cu invalidarea datelor.

Daca primul algoritm esueaza, se lanseaza cel de-al doilea analizor de culoare dominanta. Acesta poate fi parametrat sa caute primele k, k>0 culori dominante in imagine. Tehnica pe care aparatul o aplica estea cea a divizarii cubului RGB in octane si recursiunea pe fiecare dintre cele 8 partitii. La fiecare nivel de recursiune se parcurg pixelii imaginii si se contorizeaza cei a caror culoare cade in partitia RGB curenta. Pentru fiecare punct de interior adaugat se verifica daca partitia ramane monotona. Daca partitia este formata dintr-o singura culoare, situatie mrcata cu cercuri in schema de diagnostic produsa, atunci aceasta poate fi mai mica su mai mare cu o vloare globala. Sectoarele monotone cu numar de indivizi sub valoarea de maxim global sunt regiuni terminale in care algoritmul nu intra in recursiune. La fel si pentru zonele cu mai multe culor al caror contor de populatie este sub valoarea global maxima la care s-a ajuns. Daca contorul unei zone monotone este mai mare decat cea globala, el devine noul maxim si se retine culoarea ultimei zone. Fara a intra in recursiune, se trece la celalalte octane de pe palier cu acelasi rationament. Daca insa octanul curent este policrom si contorul punctelor incidente este mai mare decat maximul global, atunci nu se poate spune nimic despre zona curenta si aceasta trebuie divizata in continuare si algoritmul intra in recursiune. Diviziunea se incheie pentru cuburi de muchie 1 care nu pot fi decat monotone.



In figura alaturata este arborele de diganostic al acestui algoritm ruland pentru imaginea T42 cu cromatica variata. Se observa ca o nuanta de albastru si una de bleu au fost gasite fara vecini in spectru. Algoritmul insista pana la nivel de cub indivizibil in varful alb al spectrului. Intr-adevar, algoritmul incepe cu octanul dinspre alb al cubului RGB cu minimum global la 0 si testeaza multe sectoare adiacente, in spirala in volume din ce in ce mai mari pentru care efortul depus este din ce in ce mai mic. La ficeare nivel de recursiune imaginea este iterata de pana in 8 ori. Algoritmul converge catre culoarea dominanta in cateva zeci-sute de iteratii peste imagine cu o complexitate timp mai mica decat cea a histografului prezentat anterior. Problema sa este necesitatea de a fi reluat pentru cele k-1 culori dominante investigate.

Cei doi algoritmi conlucreaza sa gaseasca primele k culori dominante din imagine.
Odata identificate, in jurul lor este centrat cubul de excluziune bg_box. Daca k=1 sau daca culorile dominante sunt mai departe unele de altele decat diametrul configurat al zonei de zgomot, atunci cubul de excluziune este centrat in jurul culorii dominante unice. Imaginile T au fost filtrate cu un kernel bg_box de diametru 36 fata de 256 al cubului RGB, cu eliminarea uneori chiar a 90% din punctele de fundal si retinerea de detaliu de imagine la nivel satisfacator.

4.5 Vectorizarea imaginii

Odata identificat kernelul de filtru de culoare, se poate trece la vectorizarea efectiva a imaginii. Se mai parcurge imaginea odata pentru a se calcula ultimul contor N de puncte dupa eliminarea fundalului si se recalibreaza limitele pixel_space si color_space care acum sunt minime.
Se creaza matricea U a punctelor utile, matrice de Nx3 sau de N*5 cu componente reale. Se parcurge imaginea o ultima oara pentru a se incarca matricea cu coordonatele x si y si vectorul de culoare codat sau explicit. Punctele de negru gasite in original sau cele fortate spre negru de filtrul de reductiune configurat sunt marcate cu o valoare speciala negativa, pe care instrumentele de rendering ale acestei matrici o vor utiliza la substitutia culorii de prim-plan. Matricea este in principiu comparabila din punctul de vedere al consumului de memorie cu imaginea originala necomprimata RGB 24 bpp.

4.6 Rezultatul vectorizarii

Imaginea T41 a pus probleme deosebite si a fost motivarea algoritmilor de detectie de fundal. In acest caz deosebit, atat semnalul cat si zgomotul de fundal sunt intr-o plaja larga de culoare. Negrul nu e negru ci toata paleta de griuri spre albastru spre rosu si spre verde aproape in intreg octanul de intuneric al cubului RGB. Fundalul este de asemenea imrastiat intre toate nuantele de gri din partea superioara a spectrului, centrate in jurul a doua valori de alb-gri. Doua culori dominante au fost detectate si au redus poza cu circa 80%. Imagine a putut fi totusi filtrata in multimile Bg, Fg si Mg.



Odata identificat, prim planul (Fg) a putut fi substituit cu rosu, iar bg cu vrde. Se remarca doua dare verticale gri apartinand fundalului dar ramse nefiltrate

4.7 Estimarea inclinatiei imaginii

Premiza fundamentala este ca toate imaginile perceptibile de catre om au afinitate orizontala. Motivatia este de origine biologica legata de viziunea binoculara si de postura bipeda. Am putea considera ca orice fiinta inzestrata cu simtul vederii indiferent de postura isi stabileste in cursul unui proces de invatare un sistem de referinta fata de care va urmari sa raportez imaginile asimilate. In procesul nostru de educare, toate observatiile vizuale se raporteaza la orizontala si verticala. Pentru ilustrare: incercati sa urmariti un film la televizor din pozitie orizontala, privind televizorul la 90°.

Textul tiparit este un caz particular de orizontalitate absoluta, (sau verticalitate in cazul Kanji). Orice document scanat trebuie asadar orizontalizat inainte de se pretinde intelegerea continutului. Dupa detectia culorii dominante, a doua mare problema in analiza imaginilor este detectia unghiului de inclinare fata de orizontala a continutului acesteia.

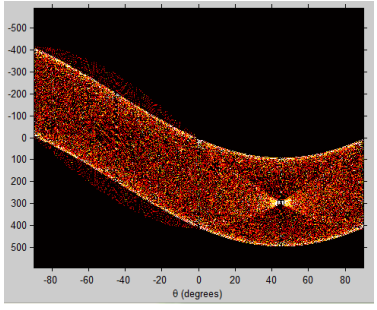
4.7.1 Transformarea Hough

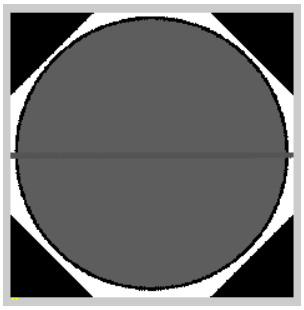
Problema orizontalizarii imaginilor este rezolvata in sistemele de imaging prin transformarea Hough prin care imaginea este transformata din coordonate planare in coordonate polare unde similaritatile de inclinatie in imagine devin imediat aparente. Lucrarea lui Paul V. C. Hough consta in brevetul US3069654 (Hough V, 1962) aplicat in televiziune, generalizat apoi de brevetul US5063604 (Weiman, 1989). Metoda prescrisa identifica si acumuleaza liniile gasite in imagine si poate fi adaptata sa culeaga cea mai importanta linie din imagine dupa care sa se faca rotirea.
Ilustram principiul utilizand in Matlab transformarea Hough pentru imaginea sintetica T44.

Programul de analiza este:

```
Iorg= imread('CircleAndLine24red.bmp');
if (size(Iorg,3)==1)
    I = mat2gray(Iorg);
else
    if (size(Iorg,3)==3)
        I = rgb2gray(Iorg);
    end
end
rotI = I;
fig1 = imshow(rotI);
BW = edge(rotI,'canny');
figure, imshow(BW);
[H,theta,rho] = hough(BW,'RhoResolution',0.1,'Theta',-90:0.1:89.9);
figure, imshow(imadjust(mat2gray(H)),[],'XData',theta,'YData',rho,...
        'InitialMagnification','fit');
xlabel('\theta (degrees)'), ylabel('\rho');
axis on, axis normal, hold on;
colormap(hot)
P = houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
x = theta(P(:,2));
y = rho(P(:,1));
plot(x,y,'s','color','black');
lines = houghlines(BW,theta,rho,P,'FillGap',5,'MinLength',7);
figure, imshow(rotI), hold on
max_len = 0;
index_longest=0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
      index_longest=k;
   end
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red');
angle=lines(index_longest).theta-90;
if (angle<-45 || angle>45)
    angle=angle-180;
end
I=imrotate(I,angle,'crop');
imshow(I);
```

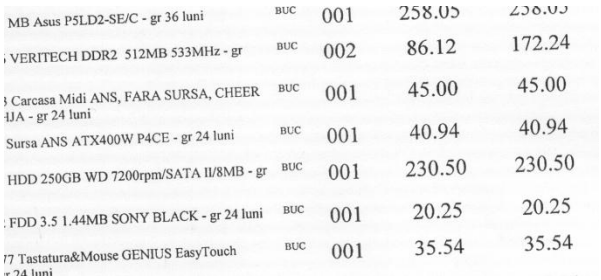
Programul produce plotul acumulatorului Hough, determina ungiul de inclinatie a singurei linii din imagine ca fiind de -45° cu care se roteste poza:

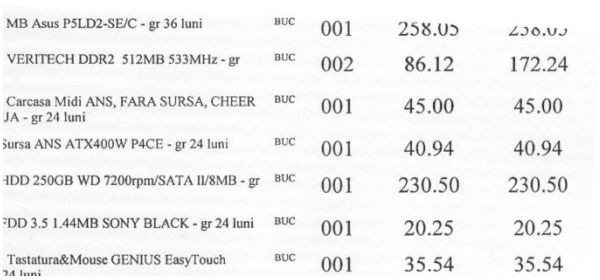




Implementarea Matlab este extrem de costisitoare, calculul durand 5 minute pentru o imagine mica.

Aceasta tehnica poate fi aplicata in detectia directiei textului in imagini dupa cum se arata in "A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents" (G. Louloudis). O adaptare a algoritmului gaseste pentru un decupaj textual din mijlocul imaginii T34 ca ungiul de inclinatie al textului este de -2.27° si reuseste sa ortogonalizeze imaginea:





Procedeul normalizeaza de asemenea cu succes T13 si T32 dar pentru T12 rezultatul este eronat, cu o inclinatie in sensul arcelor de ceasornic de 0.43 grade fata de pozitia corecta:



inainte -1.6°



dupa Hough +0.43°

Problema imaginilor de tip buletin este slaba diferentiere intre prim-planul textului si fundal.

4.7.2 Deskew prin rotoproiectie XY

Ideea acestei metode (proprii) este relativ simpla: o imagine cu afinitate fata de orizontala va avea proiectia cea mai "stransa" pe axa verticala OY. Criteriul orizontalitatii perfecte a unei imagini este ca densitatea proiectiei sa fie maxima iar dispersia acesteia sa fie minima.

In aceasta faza se ignora componentele de culoare ale indivizilor nourului U obtinut la (3) si notam populatia cu **x** variabila independenta individ.

Se va construi un vector acummulator **e** cu rol de histograf in galetile caruia se vor pune punctele imaginii a caror proiectie rotunjita la intreg corespunde cu indexul acumulatorului. Vectorul de contoare astfel obtinut indica densitatea obiectului din imagine pe sectiuni orizontale facute la intervale intregi si este o variabila dependenta de model. Se calculeaza produsul scalar σ=**e•e**. Un iterator Φ roteste imaginea cu un pas de increment si retinand unghiul pentru care σ este maxim.

Se cauta asadar:

$$\alpha' = \mathbf{arg\ max}_{\alpha}\ \Phi(\mathbf{x},\mathbf{e},\alpha) \qquad (4)$$

Rotatiile sunt scumpe in ce priveste calculul asa ca vom roti numai in intervalul $[L,R) \subset [-\pi/2,\ \pi/2)$, proiectand imaginea pe axele OX si OY simultan.
Pasii necesari:

1. Se centreaza norul de puncte individ U in jurul centrului de greutate G al imaginii.

2. Se calculeaza dispersia maxima a indivizilor - raza r a cercului de centru in G care contine toate punctele **x** ale imaginii.
3. Se formeaza un vector **e** de dimensiune 4*r.
4. Se roteste imaginea la unul din capetele intervalului de unghi intre care este cautat maximul.
5. Pentru fiecare individ de coordonate (x,y) se incrementeaza contoarele $e_{r+x|z}$ si $e_{3r+x|z}$. Vectorul e unifica proiectiile pe axele OX si OY intr-o proietie XY unificata.
6. Se calculeaza produsul scalar σ=**e•e** care reprezinta suma patratelor valorilor contoarelor, cantitate care are natura unei distante. Aceasta valoare se imparte la numarul contoarelor ne-nule ale vectorului **e**. Cantitatea obtinuta este dispersia maxima a valorilor lui **e**.
7. Se roteste imaginea cu un pas angular inspre celalalt capat al intervalului.
8. Se bucleaza la pasul 5 pana la capatul intervalului angular.

Functia Φ o vom numi rotoproiector.

Constructia lui e la dimensiuni 4*r are scopul de a conserva izotropia planului euclidian al imaginii, conditie necesara ca semnificatia statisticilor acumulate in histograf sa nu fie alterate de unghi.

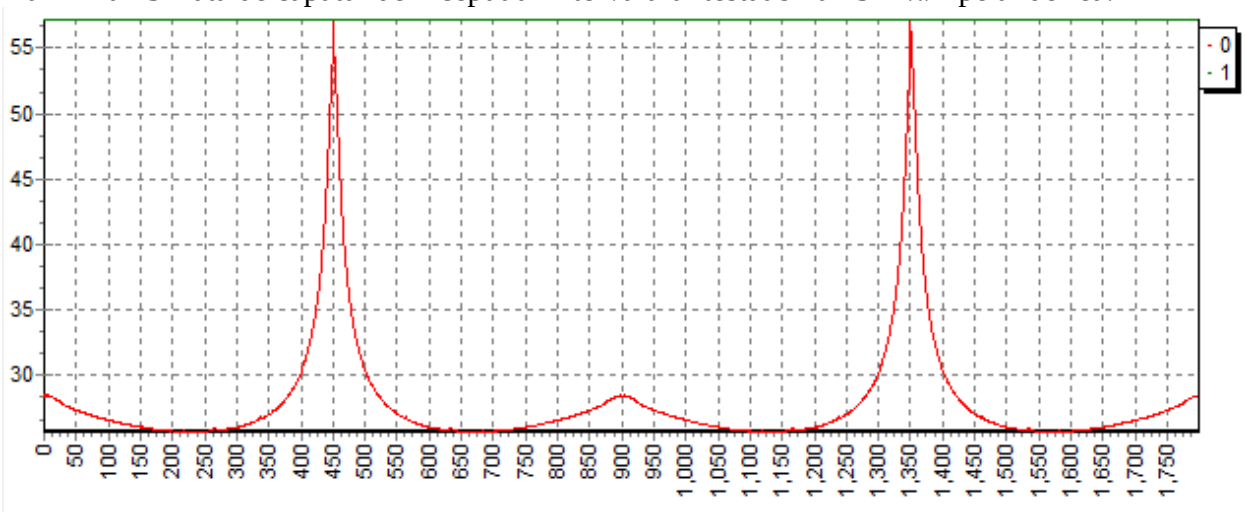
Un test pentru imaginea T44:

```
Dragging CircleAndLine24red.bmp...Thread Delete Node running.
Thread Delete Node created.
Thread Delete Node terminating.
Starting analysis...

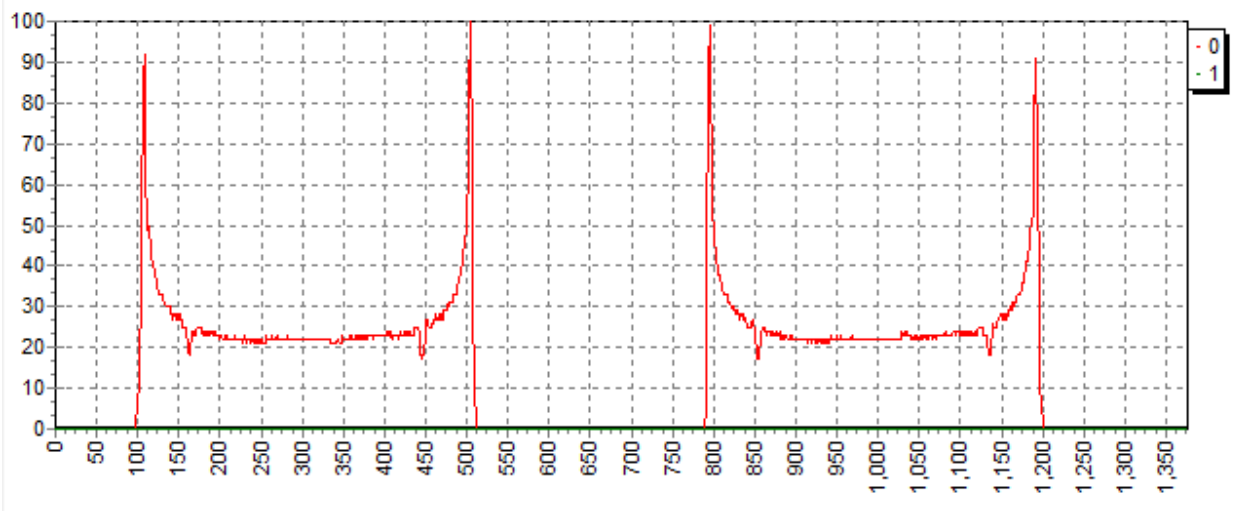
Determining the color space...
Color space: [(0,0,0),(255,255,255)]
Classification:
        6174 pixels below limit 1
        45771 pixels above limit 100
        124035 pixels within limits
Stream CircleAndLine24red.bmp width=419, height: 420, count:175980
Determining the pixel space...
Pixel space is: [(0,4),(418,415)].
Pixel count is 130121
Determining the dominant color...
        Color0 E02040 with count 119294 and contribution %91.6793  is starting the background blob.
Image loops: 4, max recursion depth: 1, pixels tested: 119294.
        Dispersion of 1 topmost colors is %13.2812 in a region of [(54,22,214),(74,42,234)] with a total count of 119294 and a
contribution of %91.6793.
        Background colour centroid is e02040 with a count of 119294 and a contribution of %91.6793.
Re-calculating the color and pixel spaces...
        Contribution of 119294 background points is %91.6793 with a dispersion of %15.1111 in a color sub-space of
[(54,22,214),(74,42,234)].
        Contribution of 6174 foreground points is %4.74481 with a dispersion of %0.444444 in a color sub-space of [(0,0,0),(1,1,1)].
        Contribution of 4653 normal points is %3.5759 with a dispersion of %0 in a color sub-space of [(204,72,63),(204,72,63)].
        Retained pixel population is 10827 in a XY space of [(4,4),(414,415)].
        Image weight centre in pixel space is :(246,248).
        Image weight centre in color space is :(133,47,41).
Creating vector image...
Preparing image rotor in the [-90,90) interval with step 1800 and recursion depth of 2...
        Creating histograph...
        Creating estimation matrix...
        Pass 0:. optimal angle is 45.
        Pass 1: exit.
Optimal angle after 1 iterations is 45.
Centering the pixel population...
New bounding box is [(-245,-242),(167,167)] with center (-39,-37).
Recognition signature: 1.00756.
Image analysis completed in 4.494 secs.
Analysis finished.
```

In cursul analizei aparatul poate produce o matrice de diagnostic A cu valorile σ (vertical) fata de unghiul α (orizontal). Continutul matricii A obtinuta pentru imaginea T44 dupa 1800 de iteratii cu pasul 0.1° in intervalul [-π/2, π/2) rezulta ca functia obiectiv Φ este periodica, desi imaginea T44 nu

este simetrica. Periodicitatea vine din simetria planului euclidian insusi. Functia isi gaseste un maxim la 45° fata de capatul de inceput al intervalului testat si la 45°+ $\pi/2$ pe al doilea.



Vectorul e, estimatorul functiei obiectiv Φ se stabilizeaza la gasirea ungiului optim de 45° la urmatorul continut, cu proiectiile simultane ale imaginii pe X si Y.

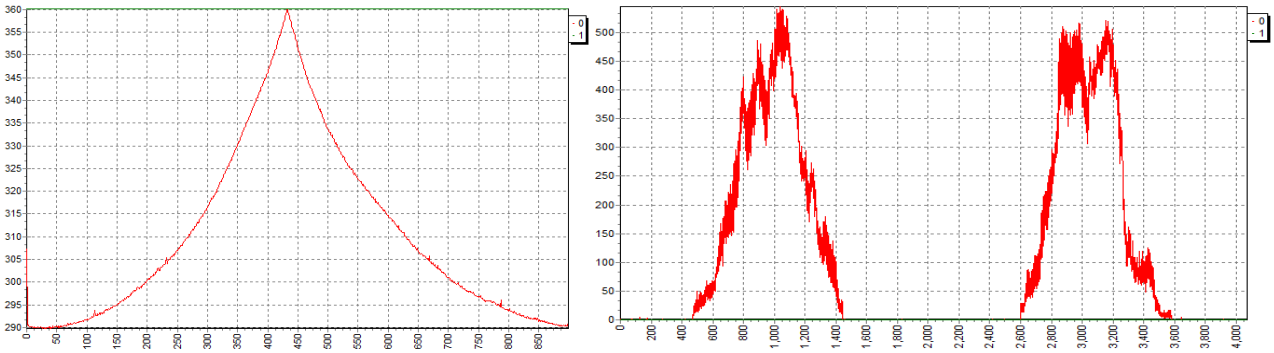


Un test pentru imaginea T13 rotoproiectata in intervalul [-45°, 45°):

Starting analysis...
Determining the color space...
Color space: [(0,0,0),(255,255,255)]
Classification:
0 pixels below limit 0
0 pixels above limit 255
2342197 pixels within limits
Stream Buletin Anda0.tif width=1573, height: 1489, count:2342197
Determining the pixel space...
Pixel space is: [(0,4),(1572,1484)].
Pixel count is 2317765
Determining the dominant color...
Color0 F9FBFB with count 330022 and contribution %14.2388 is starting the background blob.
Image loops: 104, max recursion depth: 8, pixels tested: 15504130.
Dispersion of 1 topmost colors is %23.8281 in a region of [(221,221,219),(256,256,256)] with a total count of 330022 and a contribution of %14.2388.
Background colour centroid is f9fbfb with a count of 330022 and a contribution of %14.2388.
Re-calculating the color and pixel spaces...
Contribution of 2082547 background points is %89.8515 with a dispersion of %13.8322 in a color sub-space of [(221,221,219),(256,256,256)].
Contribution of 0 foreground points is %0 with a dispersion of %0 in a color sub-space of [(0,0,0),(0,0,0)].
Contribution of 235218 normal points is %10.1485 with a dispersion of %100 in a color sub-space of [(0,0,0),(255,255,255)].
Retained pixel population is 235218 in a XY space of [(25,12),(1526,1309)].
Image weight centre in pixel space is :(784,878).
Image weight centre in color space is :(187,179,184).
Creating vector image...
Preparing image rotor in the [-45,45) interval with step 900 and recursion depth of 2...
Creating histograph...

```
        Creating estimation matrix...
        Pass 0:..... optimal angle is -1.7.
        Pass 1: exit.
Optimal angle after 1 iterations is -1.7.
Centering the pixel population...
New bounding box is [(-527,-429),(423,439)] with center (-52,5).
Recognition signature: 1.09589.
Image analysis completed in 32.398 secs.
Analysis finished.
```

In evolutia functiei obiectiv Φ se remarca un maxim cert la ungiul de inclinatie de -1.7°:



Se observa pe intreg corpusul T ca functia obiectiv are un singur maxim pentru imaginile asimetrice.

4.8 Iterator convergent spre unghiul de inclinatie al imaginii

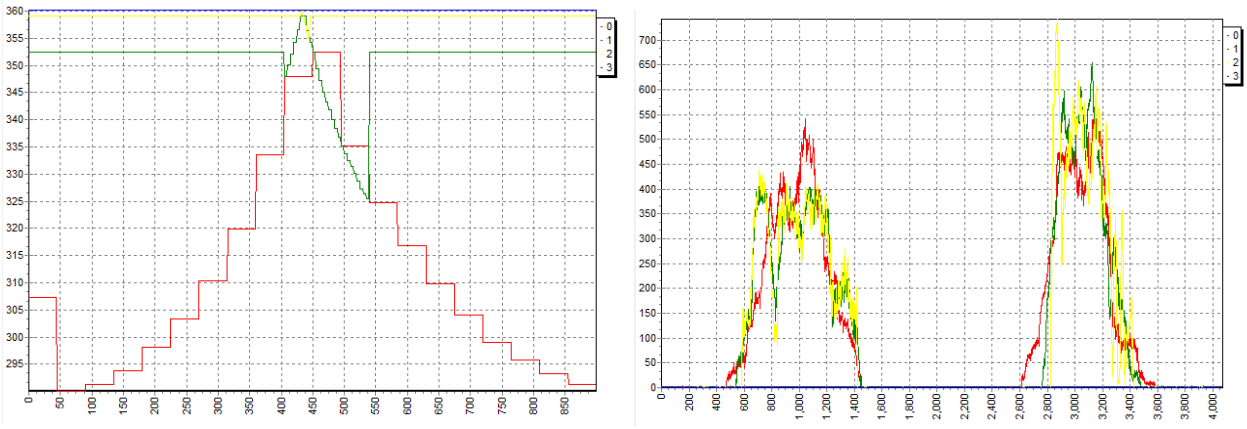
Performanta algoritmului de deskew poate fi imbunatatita de la O(r) la O(log(r)), unde r este rezolutia intervalului [L,R) in care se cauta unghiul de inclinatie. Se va itera cu pasi mai putini si mai mari pentru o prima aproximare a unghiului de skew. Intr-una din diviziuni se va constata un maxim care apoi va vi subdivizat pana la limita rezolutiei intervalului L,R. Pentru **e** si **a** se va trece de la vectori la matricile E si A de latime egala cu numarul de etape de imbunatatire a preciziei log(|R,L|)/log(n). Analiza pentru T13 devine:

```
Starting analysis...
Determining the color space...
Color space: [(0,0,0),(255,255,255)]
Classification:
        0 pixels below limit 0
        0 pixels above limit 255
        2342197 pixels within limits
Stream Buletin Anda0.tif width=1573, height: 1489, count:2342197
Determining the pixel space...
Pixel space is: [(0,4),(1572,1484)].
Pixel count is 2317765
Determining the dominant color...
        Color0 F9FBFB with count 330022 and contribution %14.2388  is starting the background blob.
Image loops: 104, max recursion depth: 8, pixels tested: 15504130.
        Dispersion of 1 topmost colors is %23.8281 in a region of [(221,221,219),(256,256,256)] with a total count of 330022 and a
contribution of %14.2388.
        Background colour centroid is f9fbfb with a count of 330022 and a contribution of %14.2388.
Re-calculating the color and pixel spaces...
        Contribution of 2082547 background points is %89.8515 with a dispersion of %13.8322 in a color sub-space of
[(221,221,219),(256,256,256)].
        Contribution of 0 foreground points is %0 with a dispersion of %0 in a color sub-space of [(0,0,0),(0,0,0)].
        Contribution of 235218 normal points is %10.1485 with a dispersion of %100 in a color sub-space of [(0,0,0),(255,255,255)].
        Retained pixel population is 235218 in a XY space of [(25,12),(1526,1309)].
        Image weight centre in pixel space is :(784,878).
        Image weight centre in color space is :(187,179,184).
Creating vector image...
Preparing image rotor in the [-45,45) interval with step 10 and recursion depth of 4...
        Creating histograph...
        Creating estimation matrix...
        Pass 0:..... optimal angle is 0.
        Pass 1:..... optimal angle is -1.5.
        Pass 2:..... optimal angle is -1.7.
        Pass 3: exit.
```

```
Optimal angle after 3 iterations is -1.7.
Centering the pixel population...
New bounding box is [(-434,-229),(403,367)] with center (-15,69).
Recognition signature: 1.40845.
Image analysis completed in 6.349 secs.
Analysis finished.
```

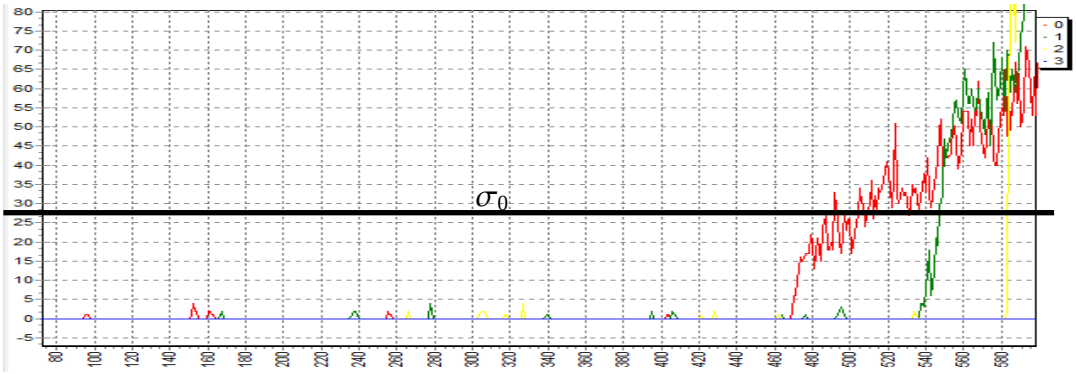
Performanta algoritmului convergent este acum de 6.35s fata de cel “flat” care dura 32.3s.

Matricile A si E reflecta acum cele trei etape de imbunatatirea a preciziei unghiului:



4.9 Determinarea noilor dimensiuni ale imaginii

Dupa procesarile necesare in eliminarea ramei negre, a colturilor, si dupa deskew, dimensiunile w si h (1) originale ale imaginii nu mai corespund. Aparatul utilizeaza vectorul **e**, sau ultima coloana a matricii E in modul convergent al algoritmului, intr-o noua analiza a dreptunghiului de incadrare a imaginii normalizate. In **e** se observa “ridicarea” si “caderea” abrupta a densitatii imaginii.



Pe graficul functiei obiectiv Φ se poate alege un prag σ_0 care sa taie zgomotul pixelilor izolati. Cele doua segmente maxime care taie graficul functiei Φ la inaltimea σ_0 sunt noile dimensiuni w’ si h’ ale imaginii rotite U’.

4.10 Aducerea la scara fixa a imaginii vectoriale.

In fine, ultima faza a pre-procesorului de deskew este aducerea imaginii la scara fixa. Odata transformata in multime de vectori, imaginea devine mai usor de manipulat prin operatori liniari. Toate imaginile de tip “populatie” (x,y,r,g,b) sunt aduse la o scara fixa, cu pastrarea raportului laturilor, in care latimea w este aceeasi pentru toate imaginile. Intre imagini variaza doar inaltimea.

Rezulta un principiu important al aparatului prezentat: **imaginile sunt clasificabile dupa inaltime**, idee pe care sectiunea dedicata clasificarii urmeaza sa o dezvolte.

4.11 Codul de Deskew

Extract din nucleul sistemului ANNAK5,
X:\Sources.DEV\SATURN.OS\MORPH\ABSTRACT\Process.cpp, NODE::GetSkewAngle:

```
  int optimal_angle=0;
  int step=(RAngle-LAngle)/2;
  int max_pass=0;
  if (RAngle>LAngle)
  {
   K5CHECKPOINT(Deskew,Histographs,2)
   max_pass=ceil(log(RAngle-LAngle)/log(resolution))+1;
   DBG((trace_flags,"Preparing image rotor in the [%g,%g) interval with step %d and recursion depth
of %d...\n",
    (double)LAngle/10,(double)RAngle/10,resolution,max_pass
   ));
   DBG((trace_flags,"\tCreating histograph...\n"));
   ObjectClient proxy_m(m);
   MatrixGroup*tEst=CAT_ALGEBRA_Type(self)->FindCreateMS(0,4*radius+1,max_pass,
&CAT_ALGEBRA_Type(self)->Relations[CAT_ALGEBRA::trReal],true);
   self=(NODE*)proxy_self.GetServer();
   Matrix*Est=CAT_ALGEBRA_Type(self)->CTR_Matrix("LinVect",tEst,0);
   self=(NODE*)proxy_self.GetServer();
   DBG((trace_flags,"\tCreating estimation matrix...\n"));
   MatrixGroup*tRot=CAT_ALGEBRA_Type(self)->FindCreateMS(0,1*(RAngle-
LAngle),max_pass,&CAT_ALGEBRA_Type(self)->Relations[CAT_ALGEBRA::trReal],true);
   self=(NODE*)proxy_self.GetServer();
   Matrix*Rot=CAT_ALGEBRA_Type(self)->CTR_Matrix("rot",tRot,0);
   self=(NODE*)proxy_self.GetServer();
   tRot=(MatrixGroup*)Rot->GetTypeMaster();
   m=(Matrix*)proxy_m.GetServer();
   mtype=(MatrixGroup*)m->GetTypeMaster();
   tEst=(MatrixGroup*)Est->GetTypeMaster();
   bits=(char*)HOST_OS_GDI::GetBitmapPixelArray((void*)self-
>get_Handle(),width,height,scan_width,bpp,image_type);
   double*est=(double*)((int)Est+sizeof(Matrix));
   mdata=(double*)((int)m+sizeof(Matrix));

   Est->Insert(m,TREE::NORMAL);
   Rot->Insert(m,TREE::NORMAL);

   double max_sharpness=0;
   optimal_angle=LAngle;
   int LA=optimal_angle;
   int RA=0;
   int vangle=0;
   int pass;
   for (pass=0;pass<max_pass;pass++)
   {
     DBG((trace_flags,"\tPass %d:",pass));
     for (int i=0;i<RAngle-LAngle;i++)
      Rot->GetSetCellValue(i,pass,+1,&max_sharpness);
     vangle+=LA-RA;
     K5CHECKPOINT(Deskew,RotationReset,2)
     //Rotate(m,m,LA-RA);
     int iangle=LA-RA;
     double deg2rad = M_PI/180;
     double sina = sin(iangle*deg2rad/ANGLE_RESOLUTION);
     double cosa = cos(iangle*deg2rad/ANGLE_RESOLUTION);
     for (int i=0;i<mtype->Rows;i++)
     {
      double x=mdata[i*mtype->Columns+0];
      double y=mdata[i*mtype->Columns+1];
      mdata[i*mtype->Columns+0]=cosa*x-sina*y;
      mdata[i*mtype->Columns+1]=sina*x+cosa*y;
     }
     RA=optimal_angle+2*step;
     if (RA-LA<resolution)
     {
      DBG((trace_flags," exit.\n"));
      break;
     }
     step=ceil((double)step/resolution);
     sina = sin(step*deg2rad/ANGLE_RESOLUTION);
```

```
  cosa = cos(step*deg2rad/ANGLE_RESOLUTION);
  for (int angle=LA;angle<RA;angle+=step)
  {
   DBG((trace_flags,"."));
   double sharpness=0;
   { // ======================== Sharpness Estimator ================
     K5CHECKPOINT(Deskew,SharpnessEstimation,2)
     for (int i=0;i<4*radius;i++)
      est[pass+tEst->Columns*i]=0;
     for (int i=0;i<mtype->Rows;i++)
     {
      int x=floor(mdata[i*mtype->Columns+0]);
      int y=floor(mdata[i*mtype->Columns+1]);
      x=pass+tEst->Columns*(1*radius+x);
      y=pass+tEst->Columns*(3*radius+y);
      if (x>=4*radius*max_pass)
       FAULT(0);
      if (y>=4*radius*max_pass)
       FAULT(0);
      est[x]++;
      est[y]++;
     }
     int count=0;
     for (int i=0;i<4*radius;i++)
     {
      int v=est[pass+tEst->Columns*i];
      if (!v)
        continue;
      sharpness+=v*v;
      count++;
     }
     sharpness=sqrt(sharpness/count); // Bessel's correction?!
   } // ======================== Sharpness Estimator ================
   vangle+=step;
   // Rotate(m,step);
   K5CHECKPOINT(Deskew,RotationIncrement,2)
   int ix=0;
   int iy=1;
   for (int i=0;i<mtype->Rows;i++,ix+=mtype->Columns,iy+=mtype->Columns)
   {
    double x=mdata[ix];
    double y=mdata[iy];
    mdata[ix]=cosa*x-sina*y;
    mdata[iy]=sina*x+cosa*y;
   }
   for (int i=angle-LAngle;i<min(RAngle-LAngle,angle-LAngle+step);i++)
    for (int p=pass;p<max_pass;p++)
     Rot->GetSetCellValue(i,p,+1,&sharpness);
   if (max_sharpness<sharpness)
   {
    max_sharpness=sharpness;
    optimal_angle=angle;
   }
  }
  if (LA<optimal_angle-step)
   LA=optimal_angle-step;
  else
   LA=optimal_angle;
  DBG((trace_flags," optimal angle is %g.\n",(double)optimal_angle/10));
}
DBG((trace_flags,"Optimal angle after %d iterations is %g.\n",pass,(double)optimal_angle/10));
DBG((trace_flags,"Centering the pixel population...\n"));
pixel_space.SetDegenerate();
double q=max_sharpness/12;
for (int i=0;i<radius;i++)
{
   int x=i-radius;
   int y=i-radius;
   if (est[pass-1+tEst->Columns*(0*(int)radius+i)]>q)
    if (x<pixel_space.A.x)
     pixel_space.A.x=x;
   if (est[pass-1+tEst->Columns*(2*(int)radius+i)]>q)
    if (y<pixel_space.A.y)
     pixel_space.A.y=y;
```

```
      x=radius-i;
      y=radius-i;
      if (est[pass-1+tEst->Columns*(2*(int)radius-i)]>q)
       if (pixel_space.B.x<x)
        pixel_space.B.x=x;
      if (est[pass-1+tEst->Columns*(4*(int)radius-i)]>q)
       if (pixel_space.B.y<y)
        pixel_space.B.y=y;
    }
    if ((flags&(1<<1))==0)
    {
     Est->Free();
     Rot->Free();
    }
   }
   else
   {
    DBG((trace_flags,"Centering the pixel population...\n"));
    pixel_space.SetDegenerate();
    for (int ord=0;ord<mtype->Rows;ord++)
    {
     int x=mdata[ord*mtype->Columns+0];
     int y=mdata[ord*mtype->Columns+1];
     pixel_space.InflateTo(1,&point(x,y));
    }
   }
   K5CHECKPOINT(Deskew,ImageNormalise,2)
   G=pixel_space.GetCenter();
   point S=pixel_space.GetSize();
   for (int ord=0;ord<mtype->Rows;ord++)
   {
    mdata[ord*mtype->Columns+0]=(mdata[ord*mtype->Columns+0]-G.x)/S.x;
    mdata[ord*mtype->Columns+1]=(mdata[ord*mtype->Columns+1]-G.y)/S.y;
   }
   S=point(400,400*S.y/S.x);
   m->V_Size=S;
   self->V_Size=S;
   m->Flags&=~NODE::vaTileable;
   self->Flags&=~NODE::vaTileable;
   DBG((trace_flags,"New      bounding      box      is      [(%d,%d),(%d,%d)]      with      center
(%d,%d).\n",pixel_space.A.x,pixel_space.A.y,pixel_space.B.x,pixel_space.B.y,G.x,G.y));
   DBG((trace_flags,"Recognition signature: %g.\n",(double)S.x/S.y));
   if (optimal_angle<0)
    optimal_angle+=360*ANGLE_RESOLUTION;
   HOST_OS::SetTimeReference(ts);
   DBG((trace_flags,"Image analysis completed in %g secs.\n",(double)ts/1000));
   if ((flags&(1<<1))==0)
   {
    m->Free();
    if (dbg_node)
     dbg_node->Free();
   }
   return optimal_angle;
```

5 METODA PROPUSA PENTRU CLASIFICAREA SI DESCOMPUNEREA IMAGINILOR

In sectiunea 2.3 s-a evidentiat o situatie tipica pentru o banca sau pentru o clinica medicala. Transformarea ρ de la imaginea – acum normalizata - la structura de date BULETIN este facuta de un etaj de clasificare. Rolul acestuia este ca printr-un criteriu de similaritate/disimilaritate sa aduca imaginile de acelasi tip intr-o galeata comuna. Aceste galeti - cozi de asteptare – vor fi procesate mai departe de motoare de recunoastere optica specializate in text, in recunoastere faciala, recunoastere de obiecte, etc., dintre acestea fiind interesati in OCR pentru extragerea textului.

5.1 Disimilaritatea imaginilor

Disimilaritatea este o functie d:E×E ⟶R+, simetrica, pozitiva si 0 doar intre element si sine insusi. Criteriile de disimilaritate intre imagini sunt implementate in functia distanta a prezentei metode de clasificare. Nu toate aceste formule intrunesc conditiile necesare functionarii algoritmului de clasare. Vom arata ca formule de tipul d=min(a,b) nu creaza dendograme cu indici de ierarhie ordonati descrescator. Formula $d=(a-b)^2/(a*b)$ citeste ca disimilaritatea intre doua imagini este in primul rand diferenta in valoare absoluta a dimensiunilor celor doua, ponderata de dimensiunile celor doua. Cu cat diferenta este mai mare, cu atat disimilaritatea este mai mare, dar cu cat dimensiunile celor doua imagini sunt mai mari, cu atat diferenta dintre ele este mai putin relevanta.

```
double Distance(NODE*L,NODE*R)
{
  double a=(double)L->V_Size.y/(double)L->V_Size.x;
  double b=(double)R->V_Size.y/(double)R->V_Size.x;
  switch (ComboBox_Distance->ItemIndex)
  {
    case 0: /* "d=(a-b)^2/(a*b)" */ return (a-b)*(a-b)/(a*b);
    case 1: /* "d=(a-b)^2/(a+b)" */ return (a-b)*(a-b)/(a+b);
    case 2: /* "d=(a-b)^2"       */ return (a-b)*(a-b);
    case 3: /* "d=abs(a-b)"      */ return abs(a-b);
    case 4: /* "d=min(a,b)"      */ return min(a,b);
    case 5: /* "d=max(a,b)"      */ return max(a,b);
    case 6: /* "d=abs(a-b)/(a*b)"*/ return abs(a-b)/(a*b);
  }
}
```

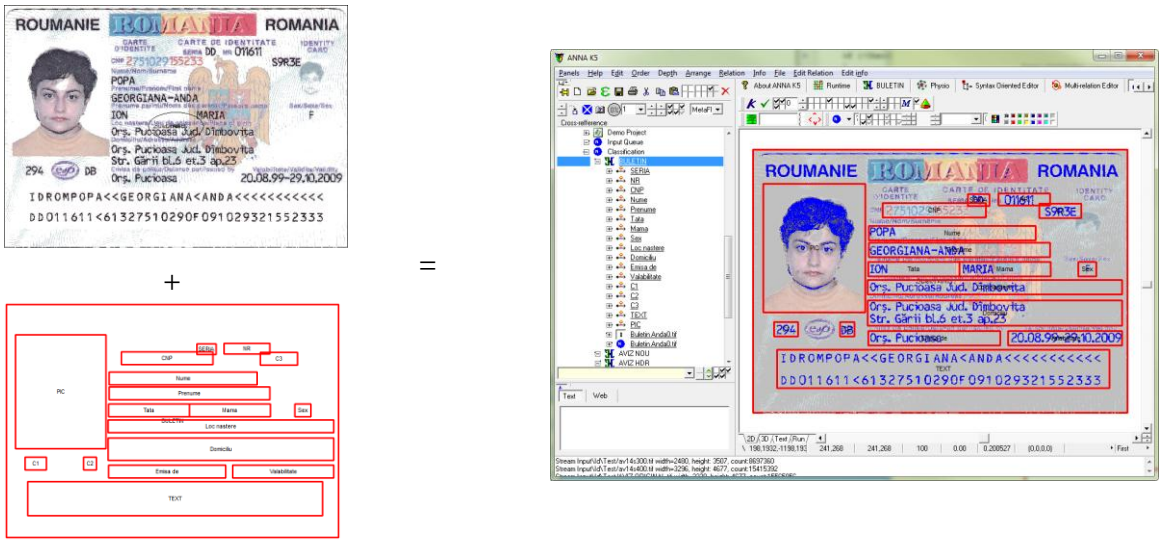
In operanzii acestor formule de disimilaritate intre indivizi pot fi inclusi si altii precum campul de flaguri ale fiecarui nod unde pot fi codate unele modalitati. Atunci trebuie insa definiti si operatorii de adunare si scadere pentru valori nenumerice pe care algoritmul de generare de clase si cel de clasare efectiva ii vor folosi.

5.2 Clasificatori de imagine

Clasificatorul este un prototip pentru indivizii pe care-i poate genera si reduce. In implementare, classificatorii de imagine sunt noduri cu dimensiunea individului mediu al clasei si sunt totodata galetile in care acestia se aduna. Clasificatorul este multimea indivizilor clasati. Sistemul prezentat permite crearea de clasificatori stoc pentru cateva imagini cunoscute precum buletinul si factura.

5.3 Masti de decupaj

In cursul reducerii la prototip, imaginile clasificate iau dimensiunile clasificatorului si ii imprumuta masca de decupaj. Clasificatori noi se pot construi programatic sau manual in editorul schematic prevazut in metoda. In figura urmatoare, unui buletin normalizat i se aplica masca clasificatorului proxim rezultand o suprapunere care poate fi decupata. In imaginea din dreapta se poate de asemenea observa dobandirea controlului asupra fundalului imaginii si asupra culorii negre de "etching" cu care este scris textul:



Ruland algoritmul de clasificare asupra corpusului T, cu toti clasificatorii din stoc si utilizand primul criteriu de disimilaritate $d=(a-b)^2/(a*b)$, unde a si b sunt caracteristicile geometrice ale celor doua imagini comparate – raportul latime/inaltime, se vor obtine urmatoarele clase:

Create stock calssifier | Decizie

Classify | 5 | 5 | Training mode

	BULETIN	AVIZ NOU	AVIZ HDR	DECIZIE
Size.x	439	550	575	600
Population	7	8	3	1
	Input\Id\Test/Buletin Alexander 0.bmp	Input\Id\Test/av14s300.tif	Input\Id\Test/av14s400.tif	Input\Id\Test/fact_header.300x300
	Input\Id\Test/Buletin Alexander 1.bmp	Input\Id\Test/Buletin Anda.tif	Input\Id\Test/AVIZ ORIGINAL.tif	
	Input\Id\Test/Buletin Alexander 2.bmp	Input\Id\Test/Circle24red.bmp	Input\Id\Test/CircleAndLine24red	
	Input\Id\Test/Buletin Anda0.bmp	Input\Id\Test/CircleAndLine24red.bmp		
	Input\Id\Test/Buletin Anda0.tif	Input\Id\Test/Decizie.400x400xg_unreco		
	Input\Id\Test/Buletin Mamo_Col.tif	Input\Id\Test/Self 1.bmp		
	Input\Id\Test/SquareAndCircle.bmp	Input\Id\Test/SquareAndCircle - Copy.bm		
		Input\Id\Test/SquareAndCircle24.bmp		

Se poate remarca o foarte buna clasificare a buletinelor, cu unele exceptia celui misclasat ca AVIZ NOU si care este chiar T14 care necesita o procesare in plus de eliminare a conturului fals.

5.4 Algoritmul de constituire a claselor optime

Clasificatorii de sistem sunt proiectati la design-time in asteptarea unor anume documente cu geometrie cunoscuta. Proiectarea mastilor de OCR este o chestiune delicata care implica mult "trail-and-error" si care trebuie asistata de calculator prin mijloace specifice. Este necesar ca o metoda de pre-procesare a OCR-ului profesionala sa includa un clasificator automat care sa ofere proiectantului o baza de pornire pentru clasele finale.

Vom extinde metoda printr-o retea de perceptroni, antrenabila printr-un corpus de imagini, retea care pastreaza si eventual isi ajusteaza ponderile pentru a realiza la iesire clasificarea si redimensionarea imaginilor.

Perceptronii vor fi clasele insele cu functii de tipul:

$$f(x) = \begin{cases} 1, & daca\ w \cdot x + b > 0 \\ 0, & altfel \end{cases} \qquad (5)$$

Ponderile w vor fi inaltimea sablonului propus de fiecare clasa, iar b va fi un prag de diferentiere intre clase, initial 0, menit sa duca indivizii prea diferiti de prototip intr-o clasa aparte a de "orfani".

In primul rand, dandu-se un corpus de imagini $I_i \in T$ (2.4), reductibile la inaltimea imaginii h_i la care s-a ajuns prin aducerea la scara fixa (4.10), trebuie gasite k clase de maxima disimilaritate care s le grupeze, si sa se calculeze, apoi inaltimea individului mediu care sa devina ponderea perceptronului – clasa.

Pentru algoritmul de identificare a pragurilor de clasare, se va adopta si implementa metoda de ierarhica descrisa in (ENACHESCU, 2009, pp. 127-136, 139-145), cu mentiunea ca vom opera direct pe grafuri, nu pe matrici.

Pasii procedeului sunt urmatorii:

1. Se construieste o ierarhie indexata dupa indicele de disimilaritate calculat intre toti indivizii in baza criteriului selectat din cele predefinite de sistem (5.1).
2. Se formeaza scree-plotul indicilor de ierarhie.
3. Se propune un prag de taiere taiere al arborelui binar al ierarhiei in proximitatea nivelului la care scree-plotul devine constant.
4. Se permite userului sa aleaga pragul k al numarului de clase.
5. Se opereaza taierea in k clase si in acelasi timp se creaza si clasele – perceptron intr-un folder al grafului sistemului de operare. Perceptronii sunt parametrati in termenul w la crearea lor cand deja se cunoaste media h a inaltimii indivizilor de test asociati clasei.

La finalul algoritmului se trece din mod de antrenare in mod "live" si clasele incep sa asimileze indivizii reali, proces deja discutat (5.2).

Extract din UI-ul sistemului ANNAK5,
$X:\Sources.DEV\Saturn.OS/MORPH/Navigator/MorphTypeLib/EditErp.cpp:

```
void ClusterAnalysis()
{
  CAT_ERP*object=(CAT_ERP*)GetObject();
  boolean created;
  NODE*inputq=(NODE*)CAT_ERP_Type(object)->FindCreateUML("Input Queue",0,NODE_Type(object),
       TREE::FIND_EXACT_TYPE|TREE::FIND_CREATE,false,true,created=false);
  if (!inputq)
   return;
  NODE*node;
  if ((node=inputq->get_FirstChild())==0)
```

```
 return;
K5CHECKPOINT(Ocr,ClusterAnalysis,2)
bool has_arcs=false;
int node_count=0;
for (node=inputq->get_FirstChild();node;node=node->GetPrev())
{
 node_count++;
 if (!has_arcs)
  has_arcs=(node->GetLine(NODE::FROM,0)||node->GetLine(NODE::TO,0));
}
if (!has_arcs)
{
  TChartSeries*cs;
  int scount=Chart_Data->SeriesCount();
  while (scount--)
  {
   cs=Chart_Data->Series[scount];
   cs->ParentChart=0;
   cs->Clear();
   delete cs;
  }
  cs=new TLineSeries(0);
  cs->Title="Dissimilarity";
  cs->SeriesColor=clTeeColor;
  cs->Marks->Visible=false;
  //((TLineSeries*)cs)->Stairs=true;
  cs->ParentChart=Chart_Data;
  DBG((DBG_TRACE,"Making dissimilarity graph...\n\r"));
  char buf[100];
  int count_nodes=0;
  int count_lines=0;
  K5CHECKPOINT(Ocr,DissimilarityGraph,2)
  for (node=inputq->get_FirstChild();node;node=node->GetPrev(),count_nodes++)
   for (NODE*n=node;n=n->GetPrev();)
   {
    double d=Distance(node,n);
    sprintf(buf,"%g",d);
    ObjectClient proxy_node(node);
    ObjectClient proxy_inputq(inputq);
    ObjectClient proxy_n(n);
    LINE*l=CAT_ABSTRACT_Type(node)->CTR_LINE(node,n,buf,0,LINE_Type(node),0);
    l->EndpointStyle[LINE::FROM]=LINE::esNone;
    l->EndpointStyle[LINE::TO]=LINE::esNone;
    n=(NODE*)proxy_n.GetServer();
    node=(NODE*)proxy_node.GetServer();
    inputq=(NODE*)proxy_inputq.GetServer();
    count_lines++;
   }
  DBG((DBG_TRACE,"Dissimilarity graph created.\n\r"));
  if (CheckBox_SingleStep->Checked)
   return;
}
NODE**nodes=(NODE**)alloca(node_count*sizeof(NODE*));
DBG((DBG_TRACE,"Starting cluster analysis...\n\r"));
char buf[100];
node=inputq->get_FirstChild();
LINE*sel_line=0;
for (int depth=node_count;depth--;)
{
  NODE*C=node;
  sel_line=0;
  double intra_dissimilarity=MAXDOUBLE;
  for (node=inputq->get_FirstChild();node;node=node->GetPrev())
   for (NODE*n=node;n=n->GetPrev();)
   {
     LINE*l=CAT_ABSTRACT_Type(node)->LINE_Find(true,node,n,0);
     if (!l)
      l=CAT_ABSTRACT_Type(node)->LINE_Find(true,n,node,0);
     double dis=atof(l->Name);
     if (dis<intra_dissimilarity)
     {
      sel_line=l;
      intra_dissimilarity=dis;
     }
```

```
 }
if (!sel_line)
 break;
K5CHECKPOINT(Ocr,PickMinLine,2)
sprintf(buf,"%g",intra_dissimilarity);
if (CheckBox_Binary->Checked)
 C=AbstractKernel->CTR_NODE(inputq,0,buf,NODE_Type(object),0);
else
 C=(NODE*)inputq->FindCreateUML(buf,0,NODE_Type(object),
    TREE::FIND_EXACT_TYPE|TREE::FIND_CREATE,false,true,created=false);
C->Shape=NODE::shSquare;
C->Insert(inputq,TREE::NORMAL);
NODE*A=sel_line->get_From();
NODE*B=sel_line->get_To();
sel_line->Free();
K5CHECKPOINT(Ocr,ClassAggregate,2)
A->Insert(C,TREE::NORMAL);
B->Insert(C,TREE::NORMAL);
LINE*cable;
int linked_nodes=0;
int dirs[]={NODE::FROM,NODE::TO};
for (int dir=0;dir<2;dir++)
 for (int i=0;cable=C->GetLine(dirs[dir],i++);)
 {
  NODE*Q=dir==NODE::FROM?cable->get_To():cable->get_From();
  bool already_in=false;
  for (int i=0;i<linked_nodes;i++)
   if (nodes[i]==Q)
   {
    already_in=true;
    break;
   }
  if (!already_in)
   nodes[linked_nodes++]=Q;
 }
for (int i=0;i<linked_nodes;i++)
{
 K5CHECKPOINT(Ocr,AdjustInterclassInertia,2)
 double dis_cable=0;
 if (cable=AbstractKernel->LINE_Find(true,C,nodes[i],0))
 {
  if (cable->Name)
  {
   dis_cable+=atof(cable->Name);
   sel_line=cable;
  }
  for (LINE*l=(LINE*)cable->get_First();l;l=(LINE*)l->get_Prev())
  {
   dis_cable+=atof(l->Name);
   sel_line=l;
  }
 }
 if (cable=AbstractKernel->LINE_Find(true,nodes[i],C,0))
 {
  if (cable->Name)
  {
   dis_cable+=atof(cable->Name);
   sel_line=cable;
  }
  for (LINE*l=(LINE*)cable->get_First();l;l=(LINE*)l->get_Prev())
  {
   dis_cable+=atof(l->Name);
   sel_line=l;
  }
 }
 dis_cable-=intra_dissimilarity;
 dis_cable/=3;
 sprintf(buf,"%g",dis_cable);
 sel_line->set_Name(buf);
 sel_line->set_From(K5NULL(sel_line));
 sel_line->set_To(K5NULL(sel_line));
 while (cable=AbstractKernel->LINE_Find(true,C,nodes[i],0))
  cable->Free();
 while (cable=AbstractKernel->LINE_Find(true,nodes[i],C,0))
```

```
      cable->Free();
     sel_line->set_From(C);
     sel_line->set_To(nodes[i]);
    }
    point S;
    int count=0;
    for (NODE*n=(NODE*)C->get_First();n;n=(NODE*)n->get_Prev())
    {
     S=S+n->V_Size;
     count++;
    }
    S/=count;
    C->V_Size=S;
    double rdissimilarity=intra_dissimilarity;
    TChartSeries*cs=Chart_Data->Series[0];
    cs->AddXY(depth,rdissimilarity,"",clTeeColor);
    if (CheckBox_SingleStep->Checked)
     break;
  }
  if (sel_line==0)
   DBG((DBG_TRACE,"Cluster analysis finished.\n\r"));
}
```

Procedura ClusterAnalysis constituie arborele binar – dendograma si merge pana la crearea scree-plotului, unde se cere utilizatorului o selectie a pragului k.

Rutina Classify face taierea efectiva a arborelui in cele k clase – perceptron. Algritmul afisaza in grila claselor continutul cozilor perceptronilor, adica imaginile clasate.

Extract din UI-ul sistemului ANNAK5,
$X:\Sources.DEV\Saturn.OS/MORPH/Navigator/MorphTypeLib/EditErp.cpp:

```
void Classify()
{
  CAT_ERP*object=(CAT_ERP*)GetObject();
  boolean created;
  NODE*inputq=(NODE*)CAT_ERP_Type(object)->FindCreateUML("Input Queue",0,NODE_Type(object),
       TREE::FIND_EXACT_TYPE|TREE::FIND_CREATE,false,true,created=false);
  if (!inputq)
   return;
  NODE*classification=(NODE*)CAT_ERP_Type(object)-FindCreateUML("Classification",0,
       NODE_Type(object),TREE::FIND_EXACT_TYPE|TREE::FIND_CREATE,false,true,created=false);
  if (!classification)
   return;
  K5CHECKPOINT(Ocr,Classify,2)
  int classes=Edit_Classes->Text.ToInt();
  double treshold=Edit_DissimilarityTreshold->Text.ToDouble();
  char buf[100];

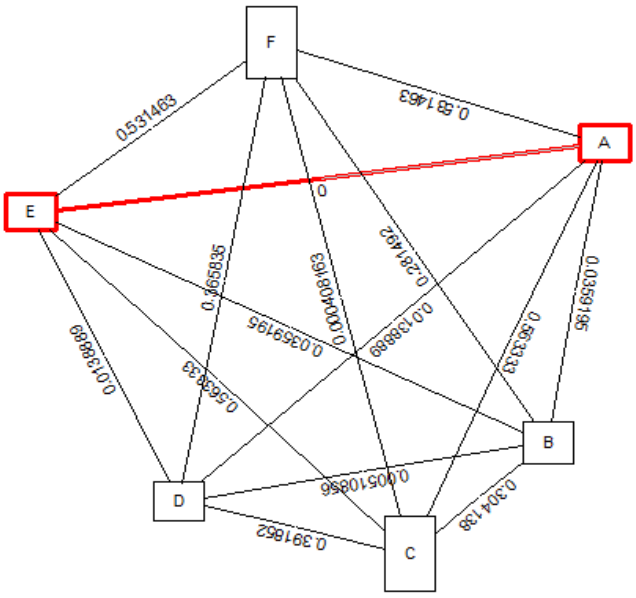
  int FrontierLength=0;
  int all_count=0;
  inputq->GetFrontier(0,all_count,0,FrontierLength,&inputq->T);
  FrontierLength=-FrontierLength;
  NODE**Frontier=(NODE**)alloca(FrontierLength*sizeof(NODE*));
  int iq_count=0;
  TYPE*classifier=0;
  for (NODE*node;node=(NODE*)inputq->get_First();)
  {
    if (!classifier)
    {
     K5CHECKPOINT(Ocr,NewCentroide,2)
     sprintf(buf,"Centroide%2d",iq_count++);
     classifier=(TYPE*)classification->FindCreateUML(buf,0,TYPE_Type(object),
        TREE::FIND_EXACT_TYPE|TREE::FIND_CREATE,false,true,created=false);
     classifier->ClassID=CLSID_NODE;
     classifier->Visibility=TREE::Implementation;
     classifier->Insert(classification,TREE::NORMAL);
     classifier->V_Size=node->V_Size;
     for (NODE*n=classifier->get_FirstChild();n;)
     {
      NODE*p=n->GetPrev();
      if (!n->IsType(RELATION_Type(classifier)))
```

```
      n->Free();
     if (p!=n)
      n=p;
    }
   }
   if (node->Shape==NODE::shBitmap)
   {
    K5CHECKPOINT(Ocr,PromoteLeafC,2)
    node->Insert(classifier,TREE::NORMAL);
    classifier=0;
    continue;
   }
   if (node->Name&&treshold<atof(node->Name))
   {
    NODE*n;
    n=(NODE*)node->get_First();
    if (n->Shape==NODE::shBitmap)
    {
     K5CHECKPOINT(Ocr,PromoteLeafL,2)
     n->Insert(classifier,TREE::NORMAL);
     n=(NODE*)node->get_First();
     n->Insert(inputq,TREE::NORMAL);
     node->Free();
     ListClass(iq_count,classifier);
     classifier=0;
     continue;
    }
    n->Insert(inputq,TREE::NORMAL);
    n=(NODE*)node->get_First();
    if (n->Shape==NODE::shBitmap)
    {
     K5CHECKPOINT(Ocr,PromoteLeafR,2)
     n->Insert(classifier,TREE::NORMAL);
     node->Free();
     ListClass(iq_count,classifier);
     classifier=0;
     continue;
    }
    n->Insert(inputq,TREE::NORMAL);
    node->Free();
    continue;
   }
   K5CHECKPOINT(Ocr,PromoteBranch,2)
   FrontierLength=0;
   node->GetFrontier(0,all_count,0,FrontierLength,&node->T);
   node->GetFrontier(0,all_count,(OBJECT**)Frontier,-FrontierLength,&node->T);
   for (int j=0;j<-FrontierLength;j++)
    Frontier[j]->Insert(classifier,TREE::NORMAL);
   node->Free();
   ListClass(iq_count,classifier);
   classifier=0;
  }
}
```

5.5 Exemplu de constituire a ierarhiei claselor

Pentru acest tur se vor genera noduri pseudoimagine de dimensiuni cunoscute si tractabile.

Nodurile {A,B,C,D,E,F} au inaltimile {2,3,8,2,1,7}, unde A si D au aceeasi inaltime 2. La primul pas al algoritmului ClusterAnalysis se creaza graful complet neorientat (Popescu, 2005) pentru nodurile A,...F.

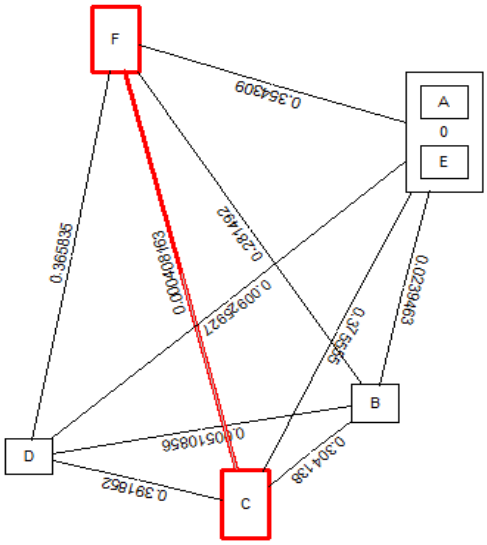


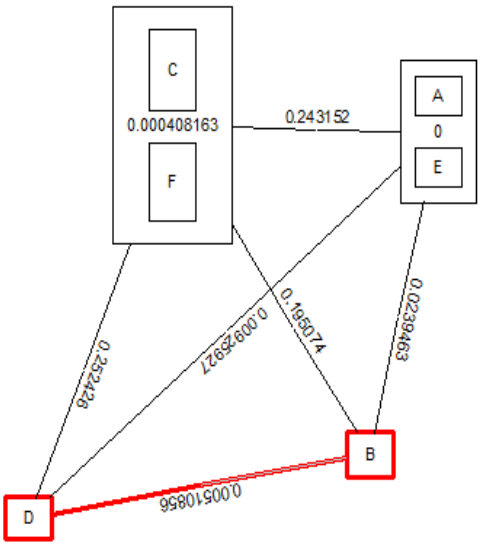
Se executa un pas de iteratie peste acest graf cand se determina ca nodurile A si E sunt cel mai putin disimilare, avand "distanta" 0. Se constituie clasa 0 care le grupeaza. Se itereaza arcele implicate si pentru fiecare se actualizeaza ponderile conform formulei Lance-Wiliams:

$$\delta(k, ij) = \frac{\delta(k,i) + \delta(k,j) - \delta(i,j)}{3} \quad (6)$$

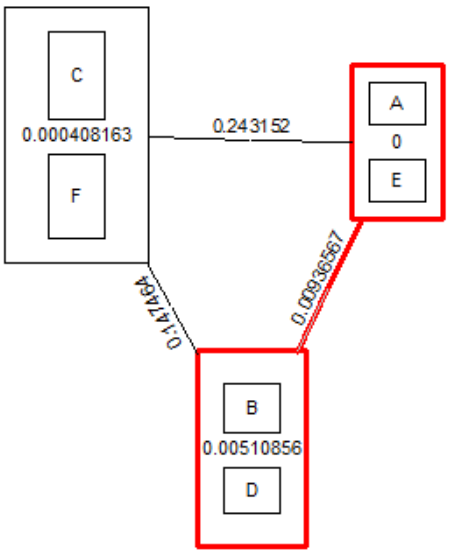
considerand ponderile nodurilor egale. Se distrug apoi toate arcele intra-clasa.

Se itereaza in continuare, observandu-se nodurile C si F ca fiind cel mai putin disimilare si se grupeaza conform celor aratate itr-o clasa 0.0004...

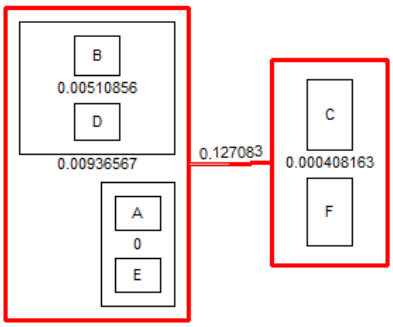




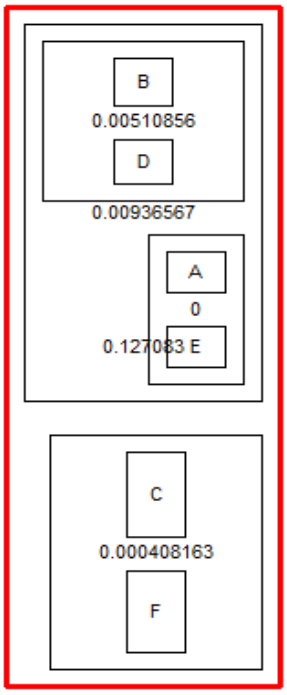
... se grupeaza doua clase gsite cu disimilaritate minima...

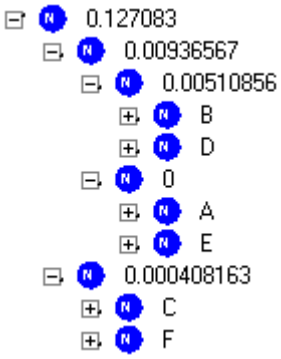


... o ultima grupare de clase...



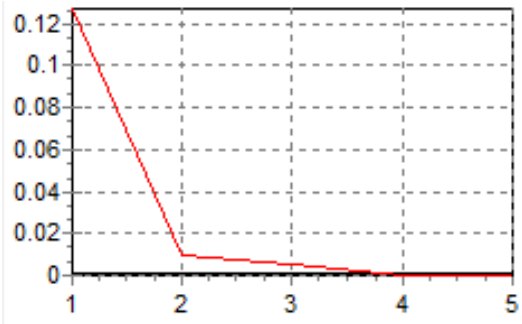
... obtinandu-se arborele binar cautat:

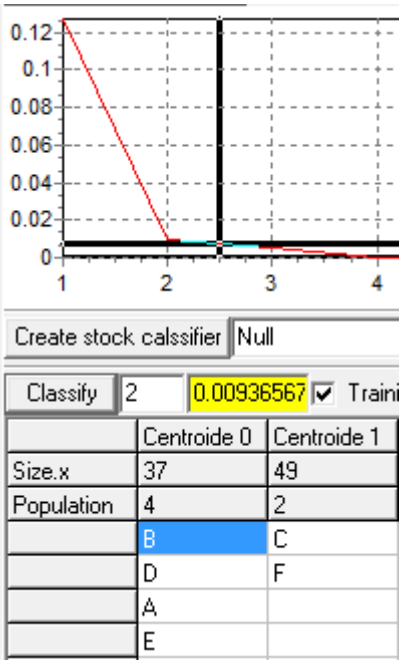




Arborele de clasificare poate fi vizualizat ca dendograma sau ca arbore de tip Eplorer:

Algoritmul produce scree-plotul in care se pot verifica vizual pragurile din ce in ce mai mici la care se naste confuzia. Situatia data sugereaza un prag de taiere de k=3 clase.



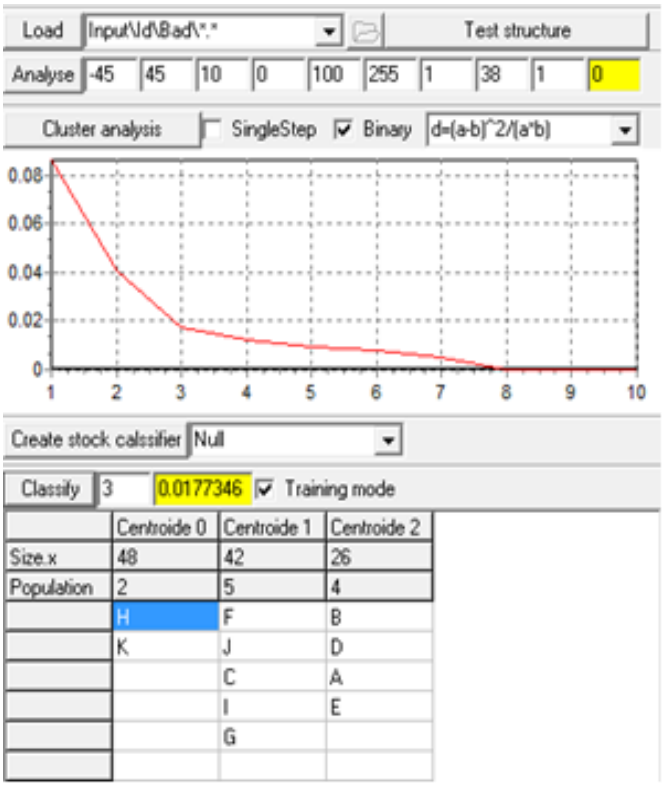


Scree-plotul este parte din interfata de urmarire si control a algoritmului. Userul selecteaza cu cursorul pragul si comanda liniarizarea ierarhiei in cele 2 clase. Intra in actiune algoritmul Classify care formeaza perceptronii centroizi 0 si 1 asupra carora sunt distribuiti indivizii A,B,C,D,E,F.

Intr-un caz concret cu cateva zeci de tipuri de imagini in setul de antrenare, efortul de creare si de manipulare a grafului creste cu patratul numarului de noduri.

6 PARAMETRAREA APARATULUI

Algoritmul este parametrabil atat la nivel de apel an celor 3 functii: GetSkewAngle, ClusterAnalysis si Classify, cat si din interfata grafica.



Prima bara de unelte este pentru incarcarea fisierelor din calea specificata sau pentru creare unei structuri demo pentru testul clasificatorului.

Bara de unelte a analizorului de inclinatie are urmatoarele controale:

1. Definitia coardei pentru care se face rotoproiectia si pasul de increment.
2. Numarul de galeti de acumulare a culorilor in algoritmul direct.
3. Vectorul de intuneric si vectorul de lumina.
4. Rezolutia de culoare.
5. Diametrul kernelului filtru de fundal.
6. Numarul de centri de culoare dominant cautati pentru formarea nuantei de fundal.
7. Parametrii de diagnostic care permit crearea unor matrici precum A si E de urmarire a rotoproiectorului, sau emiterea unei scheme a traseului urmat de algoritmul de determinare a culorii dominante.

Bara de configurare a analizorului de clase:

1. Modul demonstrativ pas cu pas.
2. Modul binar sau N-ar al arborelui de clasificare.
3. Alegerea formulei disimilaritatii.

Scree plot pentru indicele de ierarhie al arborelui de clasare, selectorul pragului de semnificatie.

Bara de creare a clasificatorilor de predefiniti de sistem.

Bara de parametrare a algoritmului de clasare:

1. Numarul de clase dupa cum a fost selectat in scree-plot.
2. Pragul de semnificatie-indicele de ierarhie la care se face taierea arborelui.
3. Modul de invatare sau modul live.

Grila claselor cu listarea pe coloana a continutului clasificatorilor.

7 ANEXE

1	US3069654 Hough.pdf	
2	Use of the Hough transform to detect lines and curves in pictures.pdf	
3	US5063604.pdf	

8 BIBLIOGRAFIE

ENACHESCU, D. (2009). *TEHNICI STATISTICE DE DATA MINING.* Bucharest: Universitatea Bucuresti.

G. Louloudis, B. G. (n.d.). *A Block-Based Hough Transform Mapping for Text Line.* Athens, Greece: Institute of Informatics and Telecommunications.

Hough V, P. C. (1962). *Method and means for recognizing complex patterns* . United States Patent 3069654.

James D. Foley, v. F. (1990). *Computer Graphics: Principles and Practice.* Addison-Wesley.

Popescu, D. (2005). *Combinatorica si teoria grafurilor.* Bucuresti: Societatea de Stiinte Matematice din Romania.

Richard O. Duda, P. E. (1972). *Use of the Hough transformation to detect lines and curves in pictures.* Communications of the Association for Computer Machinery, Vol 15, No.1, pp. 11-15 (January 1972).

Weiman, C. F. (1989). *Method and means for recognizing patterns represented in logarithmic polar coordinates.* United States Patent 5063604 .