Documentul de proiectare a solutiei aplicatiei software

(Software Design Document)

Versiunea 1.0

22 octombrie 2013

Sistem de binarizare a fisierelor

Facultatea de Automatica si Calculatoare, Universitatea Politehnica Bucuresti

10TEAM

Cuprins

[Scopul documentului 3](#_Toc370442370)

[Obiective 3](#_Toc370442371)

[3. Modelul datelor 3](#_Toc370442372)

[3.1 Structuri de date globale 3](#_Toc370442373)

[3.1.1Binarizare 3](#_Toc370442374)

[3.1.2 Votare 3](#_Toc370442375)

[3.2 Structuri de date temporare 4](#_Toc370442376)

[3.2 Formatul fisierelor utilizate 4](#_Toc370442377)

[4. Modelul arhitectural si modelul datelor 5](#_Toc370442378)

[4.1 Descrierea componentelor 5](#_Toc370442379)

[4.2 Interactiunea dintre componente 5](#_Toc370442380)

[5.Elemente de testare 6](#_Toc370442381)

# Scopul documentului

Acest document are rolul de a descrie principalele structuri de date folosite de algoritmul de binarizare si algoritmul de votare precum si modul in care aceste componente comunica intre ele. De asemenea documentul se doreste a fi un ghid pentru echipa in momentul in care se va incepe implementarea efectiva a algoritmului.

# Obiective

Se doreste o implementare care sa ofere rezultate decente intr-un timp cat mai scurt. Se va porni de la o implementare seriala a algoritmului dupa care, in functie de rezultatele obtinute, vom incerca sa optimizam acest algoritm. Optimizarile pot fi atat de paralelizare(OpenMP, Cuda) cat si variatiuni ale algoritmului original.

# 3. Modelul datelor

## 3.1 Structuri de date globale

### 3.1.1Binarizare

Cei trei algoritmi de binarizare vor folosi urmatoarele structuri de date globale :

* Un vector de intregi cu lungimea de 256 unde vom tine histograma pentru imagine. Histograma imaginii contorizeaza numarul de pixeli pentru fiecare valoare de gri din intervalul [0, 255] ;
* 3 matrici de intregi cu lungimea si latimea egale cu dimensiunile imaginii de intrare. O matrice o vom folosi pentru a retine imaginea originala, o alta matrice o vom folosi pentru a retine imaginea rezultata in urma binarizarii, iar ultima matrice o vom folosi pentru a retine confidentele.

### 3.1.2 Votare

* n matrici pentru cele n imagini rezultate in urma binarizarii;
* n matrici pentru cele n fisiere cu confidentele rezultate tot in urma binarizarii;
* 1 matrice rezultat in care se va tine imagine finala.

## 3.2 Structuri de date temporare

Nu se utilizeaza structuri de date temporare care au un consum semnificativ de memorie.

## 3.2 Formatul fisierelor utilizate

Pentru imaginea de intrare se foloseste formatul PGM(Portable Gray Map). Acesta este unul dintre cele mai folosite formate pentru imaginile grayscale.

Un fisier PGM contine una sau mai mai multe imagini PGM. Fiecare imagine PGM contine :

1. Un *numar magic*  pentru a identifica tipul fisierului. *Numarul magic* pentru o imagine pgm este alcatuit din 2 caractere : **P5 ;**
2. Delimitator (spatiu, tab, cr, lf) ;
3. Latimea(ASCII)
4. Spatiu
5. Inaltimea(ASCII)
6. Spatiu
7. Valoarea maxima de gri(ASCII). Trebuie sa fie mai mica decat 65536 si mai mare ca 0 ;
8. Delimitator(de obicei linie noua)
9. Valorile efective pentru pixeli. Daca valoarea maxima de la punctul 7 e mai mica decat 256 atunci fiecare pixel ocupa 1 octet altfel ocupa 2 octeti.

Matricea rezultata in urma binarizarii o vom scrie intr-un fisier ce va avea urmatoarea structura :

1. Lungime
2. Spatiu
3. Latime
4. Newline
5. Matricea efectiva ce va avea valoarea 1 pentru alb si 0 pentru negru.

Pentru confidente folosim o matrice pe care o vom scrie intr-un fisier ce va avea urmatoarea structura :

1. Lungime
2. Spatiu
3. Latime
4. Newline
5. Matricea efectiva cu pixelii separati prin spatiu si newline dupa fiecare linie.

In urma votarii rezulta o imagine pgm cu valori ale pixelilor de 0 sau 255.

# 4. Modelul arhitectural si modelul datelor

## 4.1 Descrierea componentelor

Aplicatia consta din doua module :

* Modulul de citire/scriere va fi responsabil cu citirea imaginilor de intrare si se va asigura de corectitudinea acestora(formatul corect). Tot acest modul va fi responsabil cu scrierea imaginii binarizate (pixeli cu valori de 0 si 1) si a fisierului de confidente( valori de la 0 la 255) ;
* Modulul de prelucrare va aplica algorimul de binarizare asupra matricii cu imaginea initiala si in urma aplicarii va rezulta matricea pentru imaginea binarizata si matricea de confidente ;
* Modulul de votare va analiza imaginile binarizate precum si confidentele si va lua o decizie, rezultand imaginea finala.

## 4.2 Interactiunea dintre componente

Fiecare celula va avea propriul makefile. De asemenea va exista un makefile global care va rula celelalte 4 makefile-uri.

In urma rularii makefile-ului global vor rezulta 4 executabile:

* binarizare1.exe
* binarizare2.exe
* binarizare3.exe
* finalizare.exe

Primele 3 executabile vor primi ca parametru de intrare calea catre imaginea originala si vor rezulta in directorul curent urmatoarele fisiere :

Pt. celula 1 :

* output\_image1 – fisierul cu 1bpp
* output\_image-confidence1 – fisierul cu confidente

Pt. celula 2 :

* output\_image2 – fisierul cu 1bpp
* output\_image-confidence2 – fisierul cu confidente

Pt. celula 3 :

* output\_image3 – fisierul cu 1bpp
* output\_image-confidence3 – fisierul cu confidente

Functia de binarizare va intoarce urmatoarele coduri :

* 0 : executie cu succes ;
* 1 : eroare la citire(nu exista drepturi asupra fisierelor de intrare, nu exista fisierele de intrare, etc.)
* 2 : formatul de intrare nu este acceptat ;
* 3 : imaginea este corupta ;
* 4 : timpul acceptat de rulare a fost depasit ;
* 5 : eroare la scriere(drepturi insuficiente, etc) ;
* 6 : eroare la alocarea de memorie(imaginile au rezolutie prea mare, etc)

Executabilul rezultat pentru votare va primi calea catre fisierele cu imaginile binarizate plus confidentele asociate. Va rezulta in directorul in care se gaseste si executabilul o imagine pgm reprezentand imaginea finala binarizata. De asemenea va rezulta si un fisier de confidente. Rolul sau va fi ca o masura de control pentru imaginea finala.

Pt. votare :

* output\_image-final.pgm
* output\_image\_confidence-final

Functia de votare va intoarce urmatoarele coduri :

* 0 : executie cu succes ;
* 1 : eroare la citire(nu exista fisierele, nu exista minim 2 imagini binarizate pentru a putea face o votare) ;
* 2 : formatul de intrare este gresit
* 3 : imaginea este corupta ;
* 4 : timpul de rulare a fost depasit ;
* 5 : eroare la scriere ;
* 6 : eroare la alocarea de memorie(imaginile au rezolutie prea mare, etc)

# 5.Elemente de testare

Pentru testare vom porni de la o imagine nealterata asupra careia vom aplica diverse efecte(pete, degradeu de culoare). Asupra acestei imagini vom aplica algoritmul de binarizare plus algoritmul de votare.

Vom testa apoi cu OCR atat imaginea nealterata cat si imaginea obtinuta in urma votarii si vom compara rezultatele.