

Univestitatea POLITEHNICA Bucuresti

Facultatea de Automatica si Calculatoare
Specializarea Calculatoare si Tehnologia Informatiei



Voting-Based Image Binarization
README
(Proiect Managementul Proiectelor Software)

Indrumator:

Saca Octavian

Componenta echipei:

Andreea Dragomir	341 C5
Mihai Trifu	341 C5
Dragos Godeanu	341 C5

Bucuresti
2013

Cuprins:

1. Rolul Documentului.....	3
2. Rolurile in echipa.....	3
3. Despre proiect.....	3
4. Algoritm	3
5. Mod de rulare	4
6. Observatii	4

1. Rolul Documentului

Acest document are rolul de a explica

- ✚ modul în care a fost organizată echipa
- ✚ algoritmul folosit
- ✚ modul de rulare al programului

2. Rolurile în echipă

Echipa a fost alcătuită din 3 membrii, fiecare având câte un rol în cadrul dezvoltării proiectului:

- ✚ **Andreea Dragomir (Tester)** – s-a ocupat de realizarea SDD-ului, Gantt și WBS, a căutat imagini pentru testare și s-a ocupat de testarea propriu-zisă
- ✚ **Mihai Trifu (Developer)** – s-a ocupat de alegerea unuia dintre algoritmii propuși de PM și de implementarea acestuia
- ✚ **Dragos Godeanu (Project manager)** – s-a ocupat de organizarea timpului, trasarea sarcinilor, crearea documentelor “activity report”, code review, research în ceea ce privește algoritmul folosit

Deși teoretic fiecare a avut sarcini bine definite, fiecare la rândul său a fost ajutat la realizarea acestora de ceilalți doi membri.

3. Despre proiect

Scopul aplicației este de a obține forma binarizată a unor imagini cu text deteriorat, având o separare cât mai clară între foreground și background, astfel încât pe rezultat să se poată aplica algoritmi OCR.

4. Algoritm

Pentru a obține imaginea binarizată, am folosit algoritmul Bernsen. Modul de funcționare al acestuia este următorul:

- ✚ Imaginii inițiale i se face padding de jur împrejur cu oglinditul ei raportat la latura respectivă
- ✚ Imaginea este împartită în mai multe imagini mai mici, cu latura de dimensiunea paddingului

✚ In fiecare dintre aceste imagini

- se calculeaza care este pixelul cu valoarea maxima si minima
- se calculeaza diferenta si media acestora
- se verifica daca diferenta este mai mare decat un anumit threshold constant setat initial
 - daca da, thresholdul va fi media dintre gri min si gri max
 - daca nu, thresholdul va fi o constanta setata initial
- daca valoarea pixelului current este mai mare decat thresholdul, pixelul binarizat va fi negru, iar daca este mai mica, alb

Matricea de confidenta a fost calculata pentru fiecare imagine mica, dupa urmatoarea formula:

$$\frac{|midGray - pixelCurrent| - minGray}{maxGray} \cdot 255$$

unde midGray este media dintre minGray si maxGray, ce reprezinta minimul, respectiv maximul dintre valorile pixelilor din acea imagine.

5. Mod de rulare

Pentru a rula programul, executabilul “BAM3.exe”, ce poate fi gasit in folderul Exe_file alaturi de libraria FreeImage.dll, va fi apelat din command prompt cu 3 parametrii:

- ✚ numele imaginii de intrare
- ✚ numele imaginii binarizare rezultata
- ✚ numele imaginii de confindenta rezultata

6. Observatii

Nu stiu din ce motive, dar imaginile rezultate difera uneori atunci cand codul este rulat din Visual Studio fata de atunci cand este rulat direct executabilul din cmd. Diferenta apare doar pe ultima coloana, respective ultimul rand de mini-imagini procesate in cadrul celei originale. Aceasta diferenta se poate observa cel mai bine intre imaginile Ex1_VS.TIF si Ex1_cmd.TIF din folderul output, unde numarul de mini-imagini in care se imparte cea originala este mic

Tot in folderul output se gasesc diferite outputuri rezultate in functie de cum au fost setati acei parametrii (dimensiunea paddingului – si implicit a mini-imaginilor, thresholdul constant si constanta de gri). Desi teoretic binarizarea ar functiona mai bine cand paddingul este mare, acest lucru nu se intampla decat pentru imaginile care au o oarecare uniformitate (a se vedea Ex1_30), insa pentru alte imagini, raman multe zone cu zgomot (a se vedea Ex4_30) . Daca thresholdul setat manual are o valoare mai mare, pentru a incerca eliminarea acelor zone, acest lucru va dauna in cazul primului exemplu, deoarece imaginea va fi prea stearsa (a se vedea Ex4_128 respectiv Ex1_128). In concluzie, am preferat sa folosesc un padding mai mic astfel incat binarizarea sa functioneze pe ambele tipuri de imagini, cu rezultate medii.