

Documentul de Proiectare a Soluției Aplicației Software

1. Scopul documentului

Scopul principal al acestui document este să creeze o specificație cât mai realistă a cerințelor și a contrângerilor de implementare pentru un Sistem de Binarizare al Imaginilor. Documentul servește drept ghid unic de construire a soluției pentru echipa de dezvoltare a proiectului.

2. Conținutul documentului

Documentul este format din patru secțiuni esențiale:

- Modelul datelor – prezintă principalele structuri de date folosite.
- Modelul arhitectural și modelul componentelor – prezintă șabloanele arhitecturale folosite, arhitectura sistemului și descrie componentele arhitecturii.
- Modelul interfeței cu utilizatorul – prezintă interfața cu utilizatorul, anume parametrii ce pot fi dați în linia de comandă.
- Elemente de testare – prezintă componentele critice și alternative de proiectare a acestora.

3. Modelul datelor

3.1 Structuri de date globale

Structura de date globală folosită este o instanță a clasei Main (clasa core a aplicației), ce are identificatorul logic. Cu ajutorul acestei structuri de date globale cu rol de intermediar, clasele diferitelor module își pot accesa reciproc structurile de date interne.

3.2 Structuri de date de legătură

Modulul BAM va primi ca argumente fișierele de intrare/ieșire, și va comunica cu modulul VBAM prin intermediul valorii de retur respectiv a outputului din imaginea de "încredere". BAM primește ca input o imagine și întoarce un cod egal cu 0 dacă procesarea a avut succes și unul diferit de 0 altfel. Outputul conține două imagini, prima este chiar imaginea rezultat, iar a doua reprezintă încrederea procesării pentru fiecare pixel în parte. Comunicarea cu VBAM se face în mod direct prin valorile pe care le regăsim în imaginea de încredere și imaginea de output. Practic nu se folosesc structuri de date de legătură.

3.3 Structuri de date temporare

Nu se utilizează structuri de date temporare cu rol important sau presupunând un consum semnificativ de resurse de memorie.

3.4 Formatul fișierelor utilizate

Aplicația va folosi fișiere de tip Netpbm pentru input-ul modulului BAM

(http://en.wikipedia.org/wiki/Netpbm_format), respectiv pentru imaginea procesată, iar pentru matricea de încredere vom folosi fișiere PGM (grayscale), fiind suficiente. Dintre tipurile de fișiere Netpbm se vor folosi tip-urile ce au stocare în mod binar și anume P4 și P5.

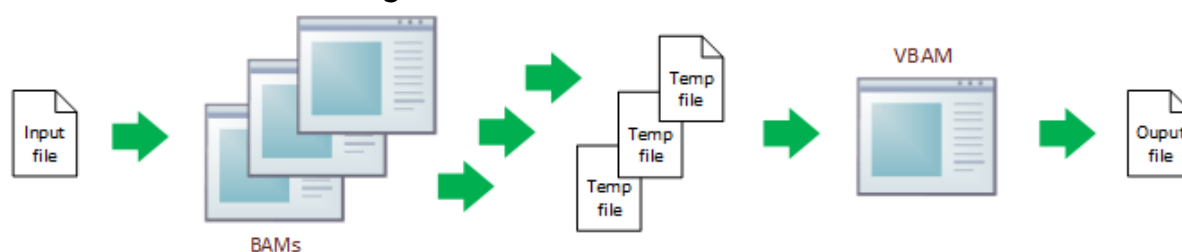
4. Modelul arhitectural și modelul componentelor

4.1 Arhitectura sistemului

4.1.1 Șabloane arhitecturale folosite

Sistemul informatic folosește șablonul arhitectural “pipe and filter”, outputul modulelor BAM este primit ca input în modulul VBAM.

4.1.2 Diagrama de arhitectură



4.2 Descrierea componentelor

BAM - Modulul de BAM se va ocupa de binarizarea imaginilor, apelat de către modulul VBAM.

VBAM - Modulul de VBAM va genera un output bazat pe datele primite de la diversele module de BAM folosind un algoritm de votare.

4.3 Restricțiile de implementare

Pentru a obține cele mai bune rezultate am decis să folosim ca limbaj de programare C/C++.

4.4 Interacțiunea dintre componente

Modulul VBAM este responsabil pentru apelarea modulelor BAM și generarea printr-un algoritm de votare al imaginii binarizate.

5. Modelul interfeței cu utilizatorul

Modulele de binarizare și sistemul de votare vor fi rulate din linia de comandă.

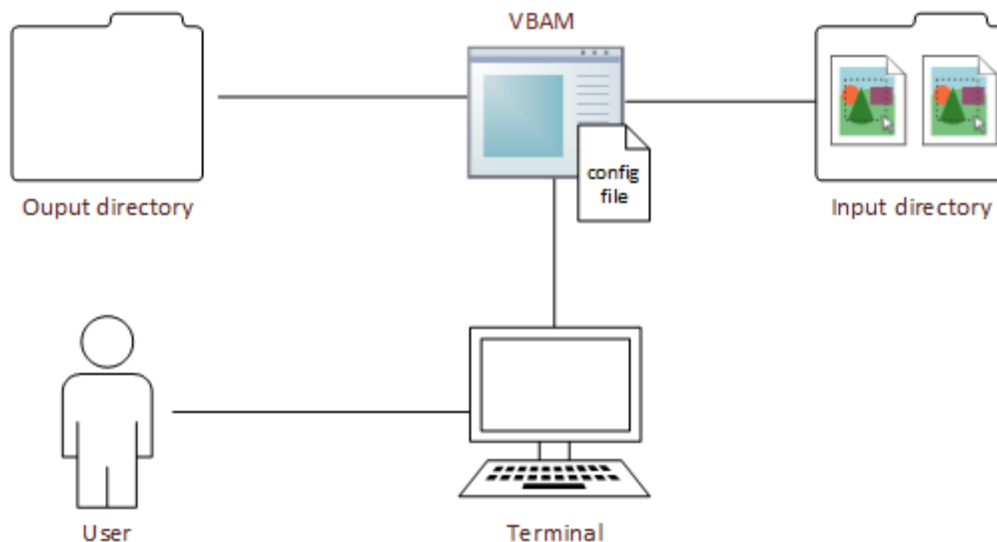
Un modul de binarizare primește la intrare 2 parametri.

- primul parametru reprezintă calea către o imagine ce va fi utilizată.
- cel de-al doilea parametru reprezintă calea către fișierul în care se va salva imaginea în urma algoritmului de binarizare

În urma rulării modulului de binarizare se va întoarce un cod de eroare prin care se poate identifica validitatea execuției:

- 0 indică succes iar imaginea creată poate fi utilizată în cadrul sistemului de votare.
- orice altă valoare indică eroare deci răspunsul nu poate fi utilizat.

Sistemul de votare primește un singur parametru ce reprezintă o cale relativă către un fișier de configurare.



Structura fișierului de configurare urmărește standardul informal INI (https://en.wikipedia.org/wiki/INI_file). Orice secțiune ce nu se numește vbam va fi tratată ca fiind o secțiune de configurare a unui BAM și numele secțiunii va fi folosit pe post de nume pentru acel BAM, adică va fi concatenat la denumirea fișierului temporar de ieșire.

[vbam]

input_folder = <directorul ce conține imaginile ce trebuie procesate de către sistem>

output_folder = <directorul ce va conține imaginile procesate de către sistem>

temp_folder = <directorul în care se stochează imaginile generate de BAM-uri>

[randombam] // modul BAM

path = <cale relativă către executabilul BAM>

flags = <parametri opționali necesari rulării modulului BAM> (opțional)

[expert] // modul BAM

.

[superbam] // numele secțiunii identifică denumirea modulului BAM

.

// utilizat pentru a diferenția imaginile de ieșire.

6. Elemente de testare

6.1 Componente critice

Performanța modulelor de BAM are o influență decisivă asupra performanței globale a aplicației, alături de performanța modului de VBAM. Altfel spus gradul de corectitudine și de confidență a imaginilor binarizate scoase de modulele BAM și a inteligenței modului de VBAM sunt principalele caracteristici ce influențează eficacitatea soluției sistemului.

Timpul necesar rulării modului VBAM depinde direct de timpul necesar rulării modulelor BAM. Evaluarea eficienței unui modul poate începe doar după finalizarea rulării modului respectiv.

6.2 Alternative

În vederea creșterii vitezei de evaluare, modulul VBAM poate executa în paralel evaluarea fiecărui modul BAM prin pornirea mai multor procese sau utilizarea threadurilor. Modulele BAM vor putea folosi GPGPU, SIMD sau alte metode de accelerare a prelucrării.