

Negativ image conversion

Introducere

Formatul de fișier **BMP** (en. Bitmap-hartă de biți), cunoscut de asemenea și ca fișier imagine Bitmap sau formatul de fișier DIB (en. Device Independent Bitmap-„bitmap” independent de dispozitiv) sau pur și simplu Bitmap, este un format de fișier imagine de tip rastru folosit pentru a stoca imagini digitale independent de dispozitivul de afișare (cum ar fi un adaptor grafic) în special pe sistemele de operare Microsoft Windows și OS/2.

Formatul de fișier BMP este capabil de a stoca imagini digitale bidimensionale de lățime, înălțime și rezoluție arbitrare, monocrome sau color, de variate profunzimi ale culorii iar opțional cu comprimare de date, canale alfa și profile de culoare

Descrierea aplicației

Scopul aplicației cerute este conversia culorilor unei imagini primite ca input în culori complementare (negative) și crearea unei noi și salvarea în memorie a acesteia.

Teorie

Conversia culorilor în negativ se realizează prin evaluarea valorilor de tip integer **R,G,B** (roșu, verde și albastru) al fiecărui pixel în parte și înlocuirea acestora cu complementul lor față de 255 (**valoarea nouă = 255 - valoarea veche**). Valoarea maximă pe care un canal de culoare dinter cele trei poate fi maxim 255. Folosirea acestor valori drept tip integer ne ajută la eliminarea dificultății și a problemelor ce pot apărea dacă acestea sunt interpretate drept unsigned byte.

Descrierea aplicației

Pachetul creat special pentru acest proiect este “**negativImage**” care conține module ce efectuează scrierea (**Writer si WriterFilePath**), citirea (**Reader si ReaderFilePath**), interpretarea propriei clase pentru imagini (**imageClass**) folosind metodele prezente în interfața **methodsInterface**, reprezintă punctul de start al aplicației (**Start**) și centrul logic de procesare a imaginii (**Convert**).



Descrierea modulelor

Clasele [Writer](#) și [Reader](#), cat si [WriterFilePath](#) si [ReaderFilePath](#) se ocupă cu scrierea și citirea, în fișierele date alese de la tastatură, respectiv scrierea si citirea in doua destinații predefinite a biților care reprezintă imaginea sub formă unui bitmap din clasă [imgClass](#), [Convert](#) fiind clasă în care se efectuează conversia propriu-zisă a imaginii preluate din obiectul instanțiat de tip [Reader](#) și este trimisă mai departe în obiectul de tip [Writer](#) pentru a fi scrisă, reprezentând centrul de procesare ce manipulează parametrii de intrare și de ieșire ale acestor obiecte prin intermediul metodelor care conțin logica algoritmului de procesare. Clasa [Start](#) este folosită ca punct de start al întregii aplicații, iar interfața [methodsInterfate](#) conține metodele ce sunt folosite de clasa [imgClass](#) pentru a obține valorile înălțimii și a lățimii ([getHeight](#), [getWidth](#)), cât și pentru setarea acestora ([setHeight](#), [setWidth](#)) si dimensiunea totala a imeginii ([getSize](#)).

Performanțe

Performanța algoritmului este liniară, iar, datorită faptului că biții sunt parcurși o singură data pentru a ajunge la rezultat, timpul de procesare este direct proporțional cu dimensiunea imaginii.

Grigoras Teodora
Grupa 334AA
1.14.2024

Concluzii

Algoritmul de conversie este unul simplu și la obiect, rulând într-un timp scurt și cu o complexitate liniară, fără a utiliza metod sau memorii auxiliare. Clasele sunt concepute astfel încât să îndeplinească fiecare o singură responsabilitate cheie.

Bibliografie

<https://ro.wikipedia.org/wiki/Bitmap>

<https://au.mathworks.com/matlabcentral/answers/363802-compute-nega/ve-image-processing-pixel-by-pixel>

<https://www.dyclassroom.com/image-processing-project/how-to-convert-a-color-image-into-negative>

<https://convertio.co/download/eb6c544f7785d1887664d9ff5770adfaa91c09/>

<https://ro.pinterest.com/pin/68746280459/>