Proiect - Logarithmic Gray Level Transform

Aplicatii Web Java

Profesor: Hossu Andrei

Student: Dutulescu Darius-Mihail

Grupa: 333AA

Introducere

**Logarithmic Gray Level Transform**

Exista 3 tipuri de transformare:

1. Lineara
2. Logaritmica
3. Legea puterii

Graficul general al acestor tranziții este prezentat mai jos.

Diagram

Description automatically generated

Transformarea logaritmica este definite de urmatoarea formula: s = c \* log(r+1), unde s și r sunt valorile pixelilor imaginii de ieșire și de intrare și c este o constantă.

Valoarea 1 se adaugă la fiecare din valoarea pixelului imaginii de intrare deoarece dacă există o intensitatea pixelilor de 0 în imagine, apoi log (0) este egal cu infinit. Deci se adaugă 1, a face valoarea minimă cel puțin 1. În timpul transformării jurnalului, pixelii întunecați dintr-o imagine sunt extinși în comparație cu valori mai mari de pixeli. Valorile mai mari ale pixelilor sunt oarecum comprimate transformare. Acest lucru are ca rezultat îmbunătățirea următoarei imagini.

Unul dintre cele mai valoroase puncte pentru îmbunătățirea imaginii la nivel de gri este că procesul este efectuat direct pe pixelul particular al unei imagini în această tehnică. Valoarea de modificare a fiecărui pixel al imaginii procesate depinde de valoarea pixelului original.

În figurile urmatoare, se poate observa transformarea logaritmică cu variația valorii constante de la 0,3 la 0,5. În experimentul efectuat, păstrăm valoarea r de la 0,3 la 0,5. Rezultatul valorii 0,5 este foarte aproape alb, așa cum se arată în figura 3. În timpul cartografierii unei imagini de intrare, nivelul de intensitate se schimbă de la scăzut la mai cuprinzător într-un interval restrâns și de intensitate ridicată spre îngust într-un interval mai larg.

Această tehnică este utilizată pe scară largă atunci când valorile imaginii de intrare sunt excepțional de mari. Variația constantă de la 0,3 la 0,5 a fost procesată, iar rezultatul fiecărei constante este clar și vizibil pentru analiză și comparație. Oproprietate pincipala a acestei transformări logaritmice este că extinde intervalul de gri la amplitudini mai mici de intrare. Acest tip de transformare se potrivește compresiei în intervalul dinamic în care imaginea procesată depășește cu mult aptitudinea dispozitivului de afișare.

A picture containing text

Description automatically generatedA picture containing text

Description automatically generatedA picture containing text, screenshot

Description automatically generated

Imaginea1 Imaginea2 Imaginea3

Descrierea codului sursa pentru procesarea imaginii

Codul sursa pentru procesarea imaginilor cu ajutorul formulei prezentate anterior este alcatuir din 7 clase: Buffer, Clasa Abstracta, Interfata, Producator, Consumator, Pixel si WriterResult si clasa principala, MyMain, in care se face apelul catre obiectele acestor clase.

**Descrierea claselor**

1. **Pixel**

In imaginea urmatoare este prezentata clasa impreuna cu atributele sale.

Text

Description automatically generated

Iar aici este prezentata functia principala din aceasta clasa, care deschide imaginea prin lucrul cu fisiere si seteaza lungimea si latimea acesteia cu ajutorul setter-ilor.

Text

Description automatically generated

1. **Producator**

In aceasta clasa are loc transferul imaginii, pixel cu pixel, prin intermediul scenariului Producator/Consumator, cu ajutor clasei Buffer. Etapele sunt urmatoarele: deschiderea imaginii, citirea pixelilor, contorizarea timpului de executie si trimiterea pixelilor. Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

1. **Consumator**

Aceasta este clasa care primeste, pixel cu pixel, intreaga imagine, provenita din Producator si stocheaza informatiile intr-o matrice temporara. In aceasta clasa are loc si prelucrarea pixelilor cu noile valori cu urmeaza a fi asignate dupa urmatoarele formule:

Text

Description automatically generated

Dupa ce imaginea este procesata, fiecare pixel are o noua valoare, imaginea este trimisa, prin pipe-uri catre un obiect de tip WriteResult, de unde este colectata si creata noua imagine de culoare gri.

Text

Description automatically generated

1. **Buffer**

Reprezinta clasa de legatura din Consumator si Producator, in care sunt definite metodele de put() si get() folosite in clasele mentionate anterior pentru a se putea realiza transmiterea datelor.

Text

Description automatically generated

1. **Interfata**

Aceasta interfata implementeaza functia getDimensions(), folosita in cadrul clasei Pixel.

Graphical user interface, text

Description automatically generated

1. **Clasa Abstracta**

Este derivata din Interfata si implementeaza aceeasi functie, getDimensions().

Text

Description automatically generated

1. **Clasa WriterResult**

In interiorul acestei clase, se redefineste metoda run prin intermediul careia imaginea este obtinuta si prin care se creeaza cea noua. De asemenea, este masurat si timpul de primire pentru fiecare segment din imagine (1/4, 2/4, 3/4, 4/4).A screenshot of a computer

Description automatically generated with medium confidenceText

Description automatically generatedText

Description automatically generated

1. **MyMain**

Aceasta este clasa din care se creaza si se apelezeaza toate obiectele claselor mentionate anterior.

Text

Description automatically generated

Concluzii

Acest studiu reprezintă o tehnica de grayscale a unei imagini implementate în Java. Acesta arată o modalitate mult mai adecvată de a obține rezultatul necesar la nivel de gri. Indiferent de eficacitatea și robustețea tehnicii executate în Java, există o anumită marjă de îmbunătățire, așa cum putem implementa folosind alți algoritmi. În această lucrare, am ilustrat un mod de îmbunătățire a calitatii unei imagini.

Bibliografie

* <https://www.rcet.org.in/uploads/academics/rohini_10935657319.pdf>
* <https://www.tutorialspoint.com/dip/gray_level_transformations.htm>
* <https://www.hindawi.com/journals/mpe/2022/8013474/>
* <https://www.youtube.com/watch?v=bYxlWVsXodI&ab_channel=FranciscoIacobelli>