

Detecția și corectarea ochilor roșii

în imagini digitale (color)

Realizat de:

Shirazi Alexia

Grupa 30235

Universitatea Tehnică din Cluj-Napoca
Procesare de Imagini – Proiect

Structura proiectului

1. Detectarea ochilor (locația lor în imagine)

- Ochii sunt rotunzi, poziționați simetric, în partea superioară a feței
- Ochii roșii prezintă saturație crescută (canalul S din HSV)
- Se realizează conversia imaginii în HSV, iar binarizarea se face pe canalul S
- Se aplică etichetare (BFS), urmată de filtrare pe dimensiune și simetrie

2. Corectarea ochilor roșii

- Se analizează pixelii din jurul centrului fiecărei regiuni detectate
- Dacă R este mult mai mare decât G și B, se aplică: $R \leftarrow (G + B) / 2$
- Corectarea se aplică doar într-o zonă circulară locală

Am ales această metodă deoarece folosirea saturației oferă o detecție mai precisă, evitând fals pozitive. De asemenea, simetria ajută la identificarea corectă a perechii de ochi.

Etapele proiectului

1. Preprocesare imagine

- Citirea imaginii color în format BGR
- Separarea canalelor (B, G, R)
- Conversie în spațiul HSV

2. Detecția regiunilor cu saturație ridicată

- Extragerea canalului S
- Binarizarea imaginii pe baza canalului S
- Eliminarea zgomotului prin filtrare pe dimensiune

3. Filtrarea detecțiilor false

- Se păstrează doar regiunile cu arie potrivită
- Se verifică poziționarea simetrică pe orizontală
- Se folosește un prag de saturație ajustabil

4. Corectarea pixelilor roșii

- Se aplică regula: dacă $R > G + 30$ și $R > B + 30$, atunci $R \leftarrow (G + B)/2$
- Se modifică doar pixelii dintr-un cerc în jurul centrului regiunii

5. Testare și evaluare

- Algoritmul a fost testat pe mai multe imagini
- Rezultatul final a fost comparat vizual cu imaginea originală

Posibile provocări și soluții

Posibile provocări	Soluții implementate
Detecții false în alte zone roșii	Filtrare după dimensiune și simetrie
Imagine întunecată sau blurată	Se poate adăuga preprocesare (ex. egalizare histogramă)
Ochiul nu este perfect roșu	Se folosește canalul de saturație din HSV și un prag flexibil
Se detectează doar un singur ochi	Se caută o pereche simetrică față de centrul imaginii

Implementare – descriere detaliată

Pentru realizarea acestui proiect, am ales o arhitectură modulară. Fiecare etapă este implementată într-o funcție separată, astfel încât codul este ușor de testat și de modificat. Mai jos descriu în detaliu fiecare modul.

1. Preprocesarea imaginii

Am citit imaginea în format BGR și am separat canalele cu `break_channels()`. Apoi, am implementat manual conversia în HSV prin `bgr_2_hsv()`, pentru a înțelege în profunzime cum se calculează Hue, Saturation și Value.

2. Binarizarea pe canalul S

Inițial am încercat detecția ochilor roșii prin comparație directă pe R, G și B. Am observat însă că duce la detecții false. Așa că am trecut la analiza saturației în spațiul HSV și am aplicat binarizarea pe canalul S cu un prag controlabil.

3. Etichetarea regiunilor

Am aplicat algoritmul BFS pentru a identifica regiunile de pixeli negri (care corespund zonelor cu saturație ridicată). Fiecare regiune a primit o etichetă unică, iar rezultatele au fost colorate pentru vizualizare ușoară.

4. Filtrarea regiunilor și identificarea ochilor

Aici am folosit funcția `draw_symmetric_eyes()`. În loc să corectez toate zonele detectate, am folosit o regulă geometrică de simetrie. Ochii, în majoritatea imaginilor frontale, sunt aproape pe aceeași linie orizontală și la distanță relativ egală față de centru. Am calculat centrul fiecărei regiuni și am căutat perechea care respectă cel mai bine aceste condiții. Acest filtru a redus semnificativ riscul de corectare greșită a altor zone.

5. Corectarea ochilor roșii

Folosind funcția `correct_red_eye()`, am parcurs zona circulară din jurul fiecărui ochi și am corectat doar pixelii unde roșul era exagerat. Corectarea se face prin înlocuirea valorii R cu media dintre G și B.