DETECTIA FETELOR CUBULUI RUBIK

*Ma****ş****ca Miron Constantin*

24.05.2017

# Introducere

Tema aleasa pentru acest proiect este rezolvarea unui cub Rubik utilizand procesare de imagini. Mai precis, utilizand camera web a unui PC, se vor identifica fetele cubului Rubik, apoi se vor analiza fetele lui. Astfel se obtine starea curenta a cubului, care va fi transmisa analizata pentru a obtine rezultatul final, acesta fiind un algoritm prin care se poate aduce cubul Rubik din orice stare in cea initiala.

# Studiul bibliografic

Popularitatea cubului Rubik a determinat existenta a multor tentative si metode pentru rezolvarea acestuia. Metodele difera intre ele prin mai multe aspecte, printre acestea fiind:

-modul de captare a fetelor: webcam live/6 poze pentru fiecare fata/2 poze a cate 3 fete

-modul de interactiune a utilizatorului cu aplicatia: poate fi necesara incadrarea fetelor cubului intr-un anumit contur

-algoritmul de detectare a celor 9 patrate de pe fiecare fata: metodele de detectie a contururilor

-algoritmul de calculare a culorilor in diferite spatii de culoare

etc.

# Soluţia propusă

* Diagrama generală a soluţiei propuse
* Prezentarea conceptelor teoretice folosite
* Descrierea fiecărui pas al soluţiei

Algoritmul utilizat in solutia acestui proiect este urmatorul:

1. Se capteaza o imagine utilizand camera web
2. Imaginea se transforma in grayscale, se blureaza si se trece printr-un filtru adaptiv de binarizare
3. Se obtin contururile din imaginea rezultata in urma operatiilor de la pasul 2
4. Se filtreaza aceste contururi in functie de arie si soliditate
5. Daca in urma filtrarii, raman 9 contururi, se verifica daca acestea sunt pozitionate corect, reprezentand cele 9 patrate de pe o fata
6. Daca nu sunt cele 9 patrate de pe o fata, se sare la pasul 1, altfel pasul 7
7. Se calculeaza media culorii de pe fiecare patrat, acestea fiind salvate pentru pasii ulteriori
8. Pasii 1-7 se repeta pentru cele 6 fete. Dupa detectia ultimei fete, se trece la pasul 9
9. Se calculeaza culoarea fiecarui patrat din cele 54 in functie de culorile din mijlocul fiecarei fete, acestea fiind reprezentative pentru fiecare din cele 6 culori
10. Se afiseaza cele 6 imagini luate afisand pentru fiecare patrat culoarea calculata. Aceastea pot fi corectate manual
11. Se utilizeaza o librarie externa pentru a afisa pas cu pas miscarile de efectuat pentru rezolvarea cubului

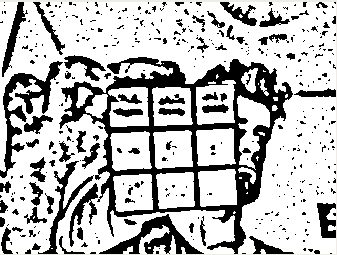
Descrierea detaliata a pasilor:

1. Folosind libraria OpenCV, se capteaza in timp real imaginea de la camera web
2. Transformarea in grayscale implica transformarea imaginii sursa astfel: pentru fiecare pixel de un singur canal din imaginea destinatie se salveaza media aritmetica a celor 3 canale din imaginea sursa.

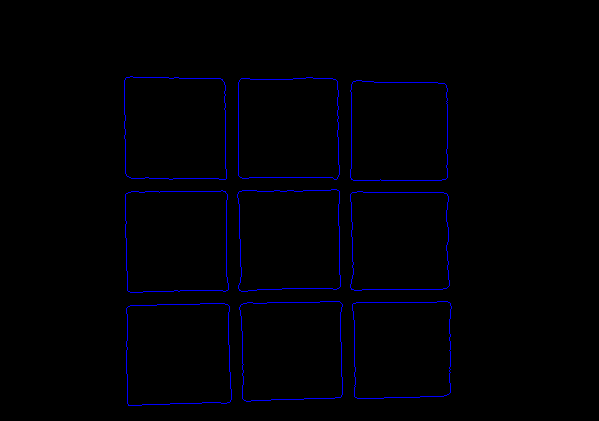
Operatia de blur se efectueaza printr-o convolutie a imaginii cu un kernel normalizat de dimensiune 5x5. Pixelul din mijlocul kernelului primeste ca si valoare media celorlalti pixeli din jurul sau.

Operatia de binarizare adaptiva implica transformarea tuturor pixelilor din imagine din valori in intervalul 0-255 in doar 2 valori: 0 si 255 (negru si alb) utilizand o tehnica adaptiva, adica pragul de definire a valorii finale a unui pixel este determinat de valorile pixelilor din jurul sau (nu se utilizeaza un prag global). Astfel se asigura functionarea aplicatiei in diferite moduri de luminare.

Imaginea in urma operatiilor din pasul 2:



3. Calcularea contururilor dintr-o imagine se obtine scanand o imagine binarizata de sus in jos, stanga-dreapta pana cand se gaseste un pixel de obiect (de culoare neagra, backgroundul fiind alb). De la acest pixel, se verifica vecinatatea si fiecare pixel de pe marginea obiectului este adaugat in lista de puncte ale conturului obiectului.

Imaginea cu contururile calculate:

4. Filtrarea pe baza ariei se face pentru a limita distanta de detectie a cubului din a fi prea departata. O limita superioara si una inferioara in functie de aria totala a imaginii indeplinesc aceasta functionalitate. Soliditatea reprezinta raportul dintre aria unui obiect si “convex hull” al acestuia. Convex hull este un contur aproximativ al unui obiect, acesta aproximand obiectul fara a fi de forma concava. Cu cat soliditatea unui contur este mai apropiata de 1, cu atat obiectul este mai convex, astfel eliminandu-se multe din contururile neregulate din background.

5. Pentru a verifica daca cele 9 contururi reprezinta cele 9 patrate de pe o fata, se calculeaza centroizii fiecaruia si se verifica coliniaritatea lor (cele 3 patrate de pe prima coloana trebuie sa fie coliniare, la fel ca urmatoarele etc. asemenea pentru celelalte patrate de pe randuri)

Conliniaritarea se verifica calculand aria triunghiului format de centroizi. Daca aceasta este relativ mica, semnifica faptul ca punctele sunt coliniare.

7. Pentru fiecare contur, se face media culoriilor pixelilor din interiorul lor. Aceasta medie este considerata culoarea patratului respectiv si este salvata intr-o lista. De asemenea, culoarea patratului din mijloc este salvat cu o anumita eticheta, acesta reprezentand standardul pentru o anumita culoare. In functie de acest standard se vor calcula si restul culorilor.

8. Se cere afisarea fetelor intr-o anumita ordine pentru a putea atasa o culoare reala culorii fetei din mijlocul fetei. Aceasta ordine este urmatoarea: rosu, albastru, portocaliu, galben, alb, verde.

9. Dupa ce toate fetele au fost scanate, se calculeaza distanta euclidiana dintre culorile patratelor si cele 6 culori reprezentative. In functie de aceasta distanta, se eticheteaza fiecare patrat cu o culoare.

10. Datorita luminozitatii ambientale, pot aparea erori in etichetarea culorilor. Aplicatia permite modificarea manuala a acestora, efectuand click pe un anumit patrat, apoi apasand una dintre tastele(r,g,b,o,y,w) pentru a reeticheta culoarea.

# Rezultate experimentale

Solutia implementata ofera rezultate bune in majoritatea cazurilor.

Imaginea rezultat cu etichetarea patratelor:



# Concluzii

Utilizand camera web, solutia ofera un nivel inalt de interactivitate, maximizand sansele de detectie corecta, analizand mai multe imagini pentru fiecare fata pana in momentul in care cubul este pozitionat optim pentru identificarea fetelor.

Partea de preprocesare care implica binarizarea usureaza identificarea contururilor, dar consuma multe resurse.

Calcularea efectiva a culorilor este efectuata in spatiu RGB pentru simplitate. Spatiul LAB ar fi o varianta mai efectiva, datorita independentei de dispozitivul de captura si pentru faptul ca poate separa luminozitatea fata de restul componentelor din care se poate deduce culoarea.

# Bibliografie

<https://stacks.stanford.edu/file/druid:yj296hj2790/Ng_Rubiks%20Cube_Reconstruction_from_Images.pdf>