

Detektovanje stanja rubikove kocke i pronalaženje rešenja

Miloš Miljanić

Fakultet Tehničkih Nauka, Novi Sad

Opis problema

Cilj projekta je olakšavanje dobijanja optimalnog rešenja za rubikovu kocku. Kako bi se dobilo početno stanje kocke, ona se kamerom snima sa svih strana. Nakon toga se rešenje, u vidu simbola koji predstavljaju poteze (R, L, B, F, U... i njihove ‘ varijante) , prikazuje korisniku.

Podaci

Svi video klipovi korišćeni tokom razvoja programa su ručno napravljeni. U procesu razvoja programa, u cilju lakšeg i bržeg testiranja, korišćeni su ranije snimljeni video snimci. Nakon završenog razvoja, program sasvim normalno radi i sa direktnim videom sa kamere.

Methods

Prvi zadatak je pronaći svih 9 kvadrata jedne strane kocke. Ulazni video se analizira frejm po frejm. Svaki frejm se podeli na kanale (red, green, blue). Zatim se primenjuje Gaussian Blur kako bi se smanjio šum na slici, i nakon toga se korišćenjem Canny Edge nalaze ivice oblika koji su boje iz datog spektra.

Od pronađenih oblika se dalje nalaze konture. Konture se ne uzimaju “sirove”, jer bi imale mnogo segmenata i takve nam ne bi bile od koristi. Zbog toga se aproksimiraju. Tim procesom se dovode do nekih oblika sa kojima možemo dalje da radimo. Kako su nam potrebne samo kvadrataste konture, filtriramo sve koje ne ispunjavaju taj uslov, odnosno ostavljamo samo one koje:

- imaju tačno četiri ivice, odnosno stranice
- konveksne su
- stranice su gotovo identične dužine
- svi uglovi imaju približno 90 stepeni

Takođe se filtriraju konture koje imaju premalu ili preveliku površinu kako bi se odstranili objekti van kocke koji bi ispunili ostale kriterijume.

Ovako dobijene konture se preklapaju, jer smo ih nalazili za svaki spektar, i za svaki threshold (npr zelena boja će se naći i u plavom i u zelenom spektru). Kako bismo dobili tačno 9 kontura (ukoliko su pronađene sve), od svake grupe kontura koje se preklapaju, uzimamo samo po jednu.

Zatim se za svaki kvadrat (unutrašnjost konture) pronalazi kojoj boji pripada. To se radi spram unapred odredjenih intervala boja u hsv formatu.

Nakon sto je frame obrađen, dobijene boje za datu stranicu se čuvaju. Kada je ovaj proces uspešno odrađen za svaku stranicu kocke rezultat za rešavanje se prikazuje korisniku.

Rezultati i problemi

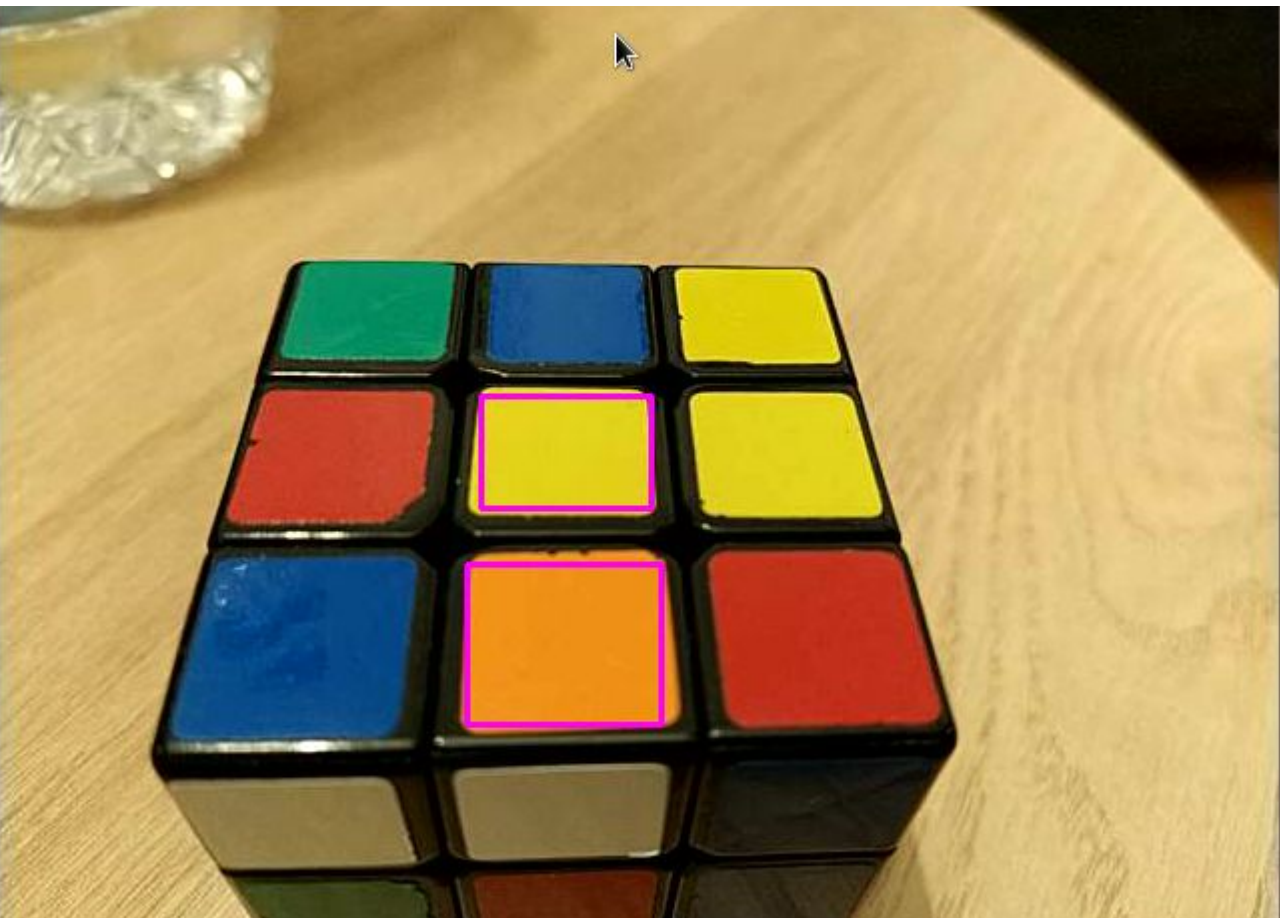
Rezultat obrade jednog frejma, ukoliko je uspešna, je zaokruživanje svih pronađenih kvadrata na frejmu. Ukoliko su pronađeni svi kvadrati, i uspešno je određena boja za svaki od njih, oni su zaokruženi zelenom bojom (Slika 1.). Ukoliko pak nisu pronađeni svi kvadrati, ili nije moguće odrediti boju za neki od njih, oni su zaokruženi ljubičastom bojom (Slika 2., Slika 3)



Jedan od problema koji je uočen prilikom testiranja je razlika u osvetljenju. Na manjim nivoima osvetljenosti, program neretko nije bio u stanju da razazna boje i samim tim nije bio upotrebljiv. Preciznost je takođe zavisila od toplote svetla. Korišćenjem toplijih sijalica cesto je dolazilo do preklapanja crvene žute i narandžaste, kao prve grupe, i zelene i plave, kao druge grupe.



Još jedan od problema koji je primećen je položaj, odnosno ugao, kocke u odnosu na kameru. Ukoliko se kocka nagne previše, iako su ljudskom oku vidljivi, kvadrati nisu bili pronalazeni.



Pravci daljeg razvoja

Kako su trenutni opsezi boja nađeni ručno, i nisu adaptibilni pri promenama boje osvetljenja, te promene mogu loše da uticu na preciznost skeniranja boja.

Jedan od načina za prevazilaženje ovog problema bi bilo dinamičko generisanje raspona svake boje tek nakon potpunog skeniranja kocke. Na taj način program bi bio otporniji na promene svetla (intenziteta, boje...).

Trenutna implementacija daje rešenje u vidu teksta sa potezima koji vode do sklapanja kocke. U daljem razvoju moguće je prikaz resenja proširiti tako da te poteze iscertava na samoj kocki (npr. u vidu strelica).

Reference

- Dokumentacija OpenCV - <https://docs.opencv.org/4.5.0/>