Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času

Jan Šuklje, Peter Peer, Bojan Klemenc
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija
sukljejan@gmail.com, peter.peer@fri.uni-lj.si, bojan.klemenc@fri.uni-lj.si

Rubikova kocka je ena najbolj znanih igrač, tako za mlade kot tudi za starejše ljudi. Za začetnike je precej velik zalogaj, zato smo se odločili narediti aplikacijo, ki jo novincem pomaga rešiti. Primarno namen same aplikacije ni učenje reševanja Rubikove kocke, saj se uporabnik z njeno uporabo ne uči, temveč samo dela gibe, ki jih aplikacija pokaže. Njen cilj je pomagati, saj če se nekomu zaplete pri reševanju kocke, si lahko z aplikacijo pomaga in kocko vrne v prvotno rešeno stanje. Kar izstopa pri tej aplikaciji v primerjavi z drugimi orodji za reševanje Rubikove kocke, je uporaba kamere kot ključnega elementa. Namesto omejevanja uporabnika na 2D ali 3D grafični prikaz kocke, ki je lahko precej zamudna, ta aplikacija izkoristi kamero, ki omogoča uporabo toka videa kot podlago uporabniškega vmesnika. Uporabnik v realnem času vidi navodila za reševanje Rubikove kocke, kar olajša in pospeši celoten proces reševanja.

DOI
https://doi.org/10.18690/um.feri.1.2024.6

ISBN
978-961-286-837-6

Ključne besede:
računalniški vid, Rubikova kocka, obogatena resničnost, Python, OpenCV

Prispevek temelji na:
Šuklje, J. (2022). *Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času:* diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Ljubljana.

Univerzitetna založba Univerze v Mariboru

DOI
https://doi.org/10.18690/um.feri.1.2024.6

ISBN
978-961-286-837-6

Keywords:
computer vision, Rubik's cube, augmented reality, Python, OpenCV

The proceedings is based on:
Šuklje, J. (2022). *Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času*. bachelor's thesis, University of Ljubljana, Faculty Computer and Information Science. Ljubljana.

University of Maribor Press

Computer Vision Application for Solving a Rubik's Cube in Real-Time

Jan Šuklje, Peter Peer, Bojan Klemenc
[1] University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia
sukljejan@gmail.com, peter.peer@fri.uni-lj.si, bojan.klemenc@fri.uni-lj.si

The Rubik's cube is one of the most well-known toys, both for young and older people. For beginners, it can be quite challenging, so we decided to create an application that helps them solve it. The main purpose of the application is not to teach solving the Rubik's cube, as users don't learn from it but only perform the moves shown by the application. Its goal is to assist if the user gets stuck while solving the cube – the user can use the application to return it to the original solved state. What stands out in this application compared to other tools for solving the Rubik's Cube is the use of the camera as a key element. Instead of limiting the user to a 2D or 3D graphical representation of the cube, which can be quite time-consuming, this application utilizes the camera, allowing video stream to be the user interface. The user receives instructions for solving the Rubik's cube in real time, which makes the entire process of solving this challenging game easier and faster.

1 Uvod

Rubikova kocka že dolgo časa navdušuje kot igrača, ki testira naš um in zabava ljudi vseh generacij. Medtem ko nekateri uživajo v zapletenih obratih in vrtenju ploskev kocke, se drugi borijo s prepoznavanjem njenih kompleksnosti. Zavedajoč se potrebe po pomoči pri reševanju te ikonične uganke, smo razvili aplikacijo v Pythonu, ki izkorišča tehnike računalniškega vida in knjižnico *OpenCV*, katere prvo javno različico je podjetje *Intel* dalo v uporabo pod licenco BSD (sedaj Apache) leta 2000 (Kaehler & Bradski, 2016). Aplikacija uporabnikom zagotovi intuitivno in interaktivno rešitev tako, da uporablja kamero za vodenje skozi postopek reševanja.

Namen prispevka je predstaviti razvoj in funkcionalnost inovativnega reševalca Rubikove kocke. Predstavili bomo problemsko domeno, postopek analize vizualnih podatkov v realnem času, ki vključuje zajemanje videa s kamero, zaznavanje kocke in njenih barv ter izračun rešitev Rubikove kocke, kakor tudi implementacijo uporabniškega vmesnika s pomočjo prikaza nad videom. Osredotočili se bomo na tehnične vidike aplikacije, kot so orodja, ki smo jih uporabili za obdelavo vizualnih podatkov, pridobljenih s kamero. Poleg tega bomo predstavili različne funkcije, vgrajene v aplikacijo, ki uporabnikom pomagajo pri procesu reševanja.

2 Pregled področja

2.1 Rubikova kocka

Rubikova kocka je priljubljena 3D uganka, sestavljena iz manjših kock, ki so različne barve. Lahko jo manipuliramo s premikanjem njenih ploskev v različnih smereh. Ko so vsi kvadrati na vseh posameznih ploskvah enake barve, pomeni da je kocka rešena.

Za opisovanje premikov Rubikove kocke smo uporabili najpogosteje uporabljeno notacijo (Ferenc, 2023), v kateri se posameznim stranem Rubikove kocke priredi specifična črka.

2.1 Sorodna dela

Priljubljenost Rubikove kocke je privedla do tega, da jo ljudje rešujejo na različne načine. Medtem ko so nekateri stremeli k hitrim rešitvam, so drugi iskali izzive v nekonvencionalnih metodah, ki presegajo standardne pristope. Eden od pristopov, ki ga je uporabilo podjetje OpenAI, je uporaba robotske roke za reševanje kocke (Akkaya, et al., 2019). S tem so demonstrirali prenosljivost računalniškega modela, treniranega samo v simulaciji, na fizično manipulacijo. Spet drug pristop, ki temelji na globokem spodbujevalnem učenju in iskanju, prikaže algoritem, ki lahko reši kocko v različnih okoljih brez zanašanja na človeško znanje (Agostinelli, McAleer, Shmakov, & Baldi, 2019). Delo, ki je najbolj podobno naši implementaciji, tudi uporablja kamero za prikaz uporabniškega vmesnika, namenjeno pa je le prikazovanju napredka reševanja Rubikove kocke in ne samem reševanju (Ajisaka, et al., 2020). Podoben študentski projekt je tudi diplomska naloga Simona Gerholda (Gerhold, 2014), ki se osredotoča na razvoj interaktivne in vizualizacijo Rubikove kocke.

3 Zajem videa

Za zajemanje videa smo uporabili mobilni telefon, na katerem je bila naložena aplikacija IP Webcam[1]. Nato smo dodali URL toka videa kot parameter naši aplikaciji za reševanje Rubikove kocke. Za zajem slik iz toka videa smo uporabili knjižnico OpenCV in implementirali razred VideoCaptureAsync, ki deluje asinhrono, kar pripomore k hitrosti delovanja. Ta razred omogoča učinkovito branje slik iz videoposnetka, kar izboljša obdelavo videoposnetkov.

Tolerance za oddaljenost in orientacijo kocke so velike, zato ni potrebno natančno nastavljati stojala za mobilni telefon.

4 Zaznavanje Rubikove kocke

Za zaznavanje Rubikove kocke smo uporabili knjižnico OpenCV in Python. S pomočjo knjižnice OpenCV smo obdelali vhodno sliko, izluščili ključne informacije in se osredotočili na prepoznavo devetih posameznih kvadratov na kocki. Zatem

[1] https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en_US

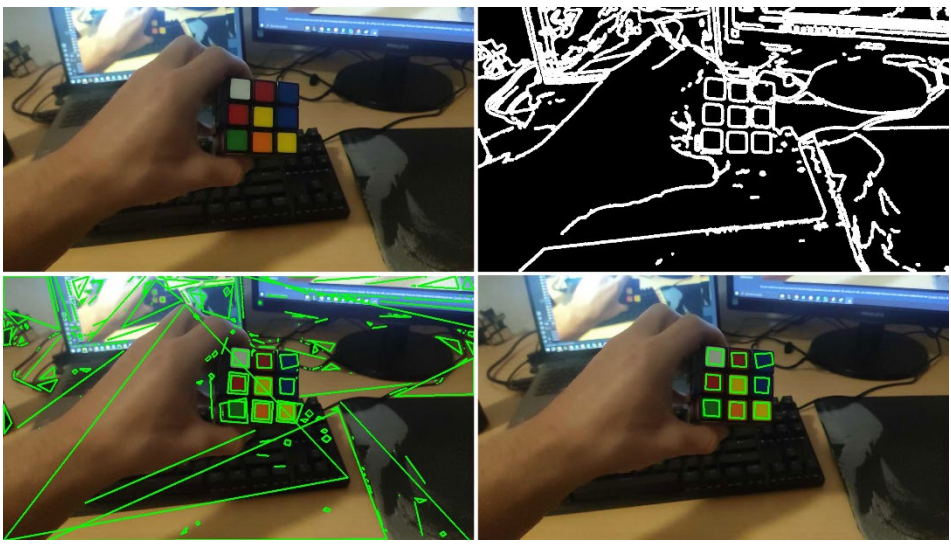
smo poskušali ustvariti zunanji pravokotnik, ki popolnoma obdaja te kvadrate in pove, ali je bila kocka uspešno zaznana. Če je kocka uspešno zaznana, na naslednji sliki video toka uporabimo masko za pospešeno zaznavanje.

4.1 Zaznavanje barvnih kvadratov

Za zanesljivo identifikacijo kvadratov Rubikove kocke je uporabljen niz postopkov, ki vključujejo Gaussovo zameglitev, Cannyjev detektor robov, iskanje kontur, optimizacijo kontur s funkcijo approxPolyDP ter filtriranje kontur glede na število točk, površino, konveksnost in kotne lastnosti.

Gaussova zameglitev se uporablja za zmanjšanje šuma in pridobivanje bolj konsistentnih podob kvadratov. Cannyjev detektor robov omogoča natančno identifikacijo robov v sliki. Iskanje kontur se izvaja s funkcijo cv2.findContours(), nato pa se uporablja funkcija approxPolyDP za optimizacijo kontur z manjšim številom mejnih točk.

Filtriranje kontur se izvaja glede na število točk, površino, konveksnost in kotne lastnosti. Prekrivanje med kvadrati se odstrani, zagotavljajoč, da ostanejo samo relevantni kvadrati za nadaljnjo analizo.



Slika 1: Prikaz postopka zaznavanja barvnih kvadratov
Vir: lasten

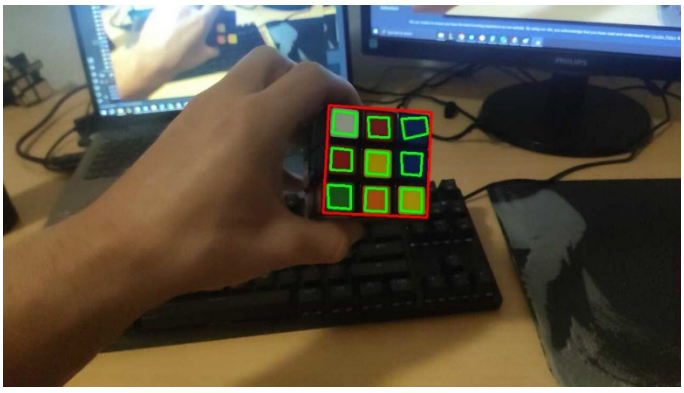
Slika 1 prikazuje začetno sliko, identifikacijo robov z Cannyjevim detektorjem, nato zaznane robove po optimizaciji kontur in na koncu vseh devet zaznanih kvadratov po končni filtraciji. Ta celoten postopek omogoča zanesljivo zaznavanje barvnih kvadratov Rubikove kocke, bolj podroben opis lahko najdete v (Šuklje, 2023).

4.2 Nepravilno število kvadratov

Po izvedbi filtriranja se lahko pojavi situacija, kjer imamo lahko več ali manj kot devet kvadratov, ki so ključni za prepoznavo Rubikove kocke. Če ostane manj kot devet takšnih kvadratov, postopek prekinemo in čakamo na naslednjo sliko. To se običajno zgodi, ko na sliki ni Rubikove kocke ali je prikazan le del nje. Nasprotno, če imamo več kot devet kvadratov, najprej identificiramo srednji kvadrat, obdan s tremi ali več sosednjimi kvadrati na vseh štirih straneh. Nato znotraj slike poiščemo osem najbližjih kvadratov temu srednjemu kvadratu, da pridemo do končnih devetih kvadratov. To se zgodi v primerih, ko je ozadje Rubikove kocke polno drugih kvadratnih objektov, ki so tudi zaznani kot kvadrati.

4.3 Indikator celotne Rubikove kocke

Da bi se prepričali, ali vsi kvadrati tvorijo eno skupno kocko, uporabimo dodaten postopek, kjer jih poskušamo očrtati s še enim kvadratom, primer je razviden na sliki 2. Oglišča tega novega kvadrata določimo glede na položaj manjših štirih kvadratov, ki se nahajajo v ogliščih zaznane kocke.

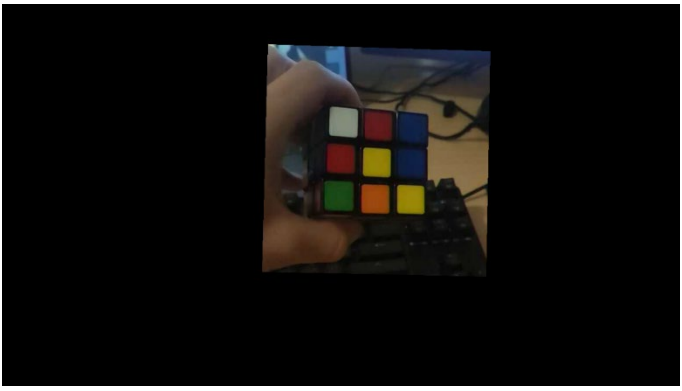


Slika 2: Očrtanje devetih kvadratov
Vir: lasten

Če ima kvadrat z rdečim robom dovolj podobne dolžine vseh štirih stranic ter notranje kote, ki so dovolj podobni pravim kotom, smo lahko skoraj prepričani, da smo zaznali Rubikovo kocko.

4.4 Maskiranje nadaljnjih zaznavanj

Če uspešno zaznamo Rubikovo kocko na eni sliki, lahko pričakujemo, da se bo na naslednji sliki nahajala na približno istem mestu. To omogoča optimizacijo obdelave slik z izbrisom ali zakrivanjem delov slike, ki so dovolj oddaljeni od zaznane kocke. Uporaba strategije maskiranja zmanjša obremenitev računalniških virov, še posebej, če se uporabi natančen Cannyjev detektor robov. Med zajemanjem slik je pomembno upoštevati hitrost premikanja kocke in razdaljo le-te od kamere. Primer maskiranja je prikazan na sliki 3.



Slika 3: Maskiranje izvorne slike
Vir: lasten

5 Zaznavanje barv

Zaznavanje barv je ključno pri reševanju Rubikove kocke, zato je pomembno, da to opravimo natančno in zanesljivo v različnih svetlobnih pogojih. Analiza vsakega kvadrata se izvaja s preverbo, ki ugotovi prevladujočo barvo. V knjižnici OpenCV je uporabljen barvni prostor BGR, vendar za olajšano analizo barv uporabljamo barvni prostor HSV, v katerega prevladujočo barvo tudi pretvorimo. Barvni prostor HSV omogoča ločevanje odtenkov, nasičenosti in svetlosti, kar olajša obdelavo in primerjavo barv med seboj.

S poskušanjem smo prepoznavali, katerim barvam pripadajo vrednosti v modelu HSV. S postopnim testiranjem smo izboljševali pretvorbo in zagotovili, da ni bila preveč občutljiva na različne svetlobne pogoje. Trenutno delovanje sistema za prepoznavanje barv poteka zelo dobro, izjema so zelo slabi svetlobni pogoji.

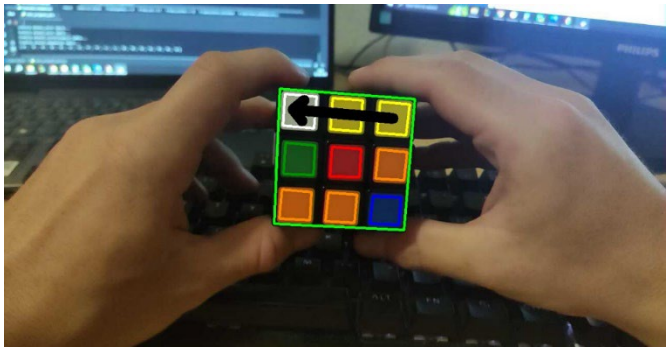
6 Algoritem za reševanje Rubikove kocke

Algoritem Kociemba z dvema fazama (Kociemba, 1992) se izkaže kot izjemno zmogljiva in učinkovita metoda za reševanje zapletene uganke, kot je Rubikova kocka. To inovativno metodo je razvil Herbert Kociemba v zgodnjih 2000-ih letih. Metoda se je hitro uveljavila kot ena izmed najbolj naprednih tehnik za premagovanje izziva sestavljanja te priljubljene ugankarske igrače.

Ena izmed najbolj opaznih prednosti tega algoritma je sposobnost, da kocko reši v manj potezah v primerjavi z mnogimi drugimi metodami. Omogoča hitro reševanje celo najzahtevnejših konfiguracij kocke v razumnem časovnem okviru.

7 Uporabniški vmesnik

Uporabnik na začetku sledi navodilom, ki mu povedo, kako in v katerem vrstnem redu mora prvotno zaznati vse strani Rubikove kocke. S tem pridobimo vse kvadrate in njihove barve, ki jih potrebujemo za iskanje gibov, ki vodijo do rešene kocke. Ko je to urejeno, se uporabniku na videu začnejo prikazovati navodila, kako Rubikovo kocko premikati, da bo prišel do končnega rešenega stanja.



Slika 4: Prikaz puščice za potreben premik
Vir: lasten

Ker smo želeli narediti kar se da enostaven uporabniški vmesnik, smo se odločili, da bo prikaz potrebnih gibov prikazan samo s puščicami na sami kocki v realnem času (slika 4). Tako si lahko uporabnik zlahka predstavlja, kaj mora narediti.

Po vsakem gibu, ki ga naredi uporabnik, se preveri, ali je bil gib pravilno narejen. Če je bil napačen, se celotno zaznavanje prekine, uporabnik pa mora zopet začeti znova in še enkrat pokazati vse strani Rubikove kocke.

Za boljšo predstavo o delovanju je primer uporabe objavljen na omrežju Youtube (Šuklje, 2023)(Šuklje, 2023).

8 Rezultati

Število potez, potrebnih za rešitev Rubikove kocke, je ključna mera, ki je odvisna od učinkovitosti algoritma in od tega, kako dobro je kocka premešana. Algoritem išče rešitve v največ dvajsetih potezah, kar se je izkazalo kot optimalno, saj se rešitev običajno najde hitro. Hitrost in odzivnost aplikacije sta zadovoljivi, brez zaznavnih kjučnih zakasnitev pri prikazu slike. Teste smo izvedli na namiznem računalniku, ki ima procesor AMD Ryzen 7 2700 in na prenosniku z procesorjem AMD Ryzen 5 5500U, v obeh primerih je aplikacija delovala brez problemov, razlike v odzivnosti med njima nismo opazili. Primerjava s hitrostjo prikaza surovega videa kaže minimalno razliko v zakasnitvi, kar kaže na dobro optimizacijo aplikacije. Ker aplikacija omogoča samodejno zaznavanje barv in prikaz navodil na videu kocke, zmanjšuje tveganje za napake, saj uporabniku ni potrebno ročno vnašati vseh 54 barv kvadratov, med reševanjem same kocke pa napačno izveden gib pomeni prekinitev delovanja.

Aplikacijo so preizkusile štiri odrasle osebe in tudi dva otroka stara sedem let, rezultati pa so pokazali, da je aplikacija uporabna tako za starejše kot tudi za mlajše uporabnike. Vsi odrasli so uspešno rešili kocko v nekaj minutah, medtem ko so otroci potrebovali malo dlje, saj so si sprva težko predstavljali katero stran kocke gledajo, ampak ko so to ugotovili, jim je kocko uspelo rešiti.

9 Zaključek

Najzahtevnejši del pri razvoju aplikacije je bilo razvijanje samega sistema za zaznavanje Rubikove kocke. Opravili smo številne iteracije, pri čemer smo vsakič uporabili drugačen pristop k prepoznavanju kocke. Stalnica pri tem je bila uporaba knjižnice OpenCV. Uporaba aplikacije je preprosta in hitrejša v primerjavi z drugimi podobnimi rešitvami, kar smo potrdili z uspešnim preizkusom med družinskimi člani in prijatelji, ki niso znali sestaviti Rubikove kocke. S pomočjo aplikacije so vsi uspeli rešiti Rubikovo kocko v le nekaj minutah.

Med razvojem in testiranjem aplikacije smo zaznali nekaj točk, kjer bi lahko izvedli izboljšave. Ključna izboljšava bi bila povečanje natančnosti pri prepoznavanju Rubikove kocke na kamerah nižje kakovosti. Testiranje z vgrajeno kamero v prenosniku je razkrilo, da se detekcija poslabša v slabših razmerah zaradi omejenega kontrasta slike in večjega šuma. Dodatna izboljšava v prihodnosti bi omogočila uporabnikom, da Rubikovo kocko zaznajo in identificirajo v poljubnem vrstnem redu med prvo fazo zajemanja vseh strani kocke.

Literatura

Agostinelli, F., McAleer, S., Shmakov, A., & Baldi, P. (2019). Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence, 1*, 356–363.

Ajisaka, S., Hara, S., Matsuchi, M., Luo, S., Yoshida, S., Xie, H., & Miyata, K. (2020). Learning rubik's cube through user operation history. *Nicograph International (NicoInt)*, str. 43–46.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., and others. (2019). Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Ferenc, D. (2023). *Rubik's cube notation*. Pridobljeno 1.2.2024: https://ruwix.com/the-rubiks-cube/notation/

Gerhold, S. (2014). Razvoj interaktivne Rubikove kocke: diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Ljubljana.

Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. O'Reilly Media, Inc.

Kociemba, H. (1992). *Rubik's Cube Twophas eSolver*. Pridobljeno 1.2.2024: https://github.com/hkociemba/RubiksCube-TwophaseSolver

Šuklje, J. (2023). Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času: diplomsko delo, Univerza v Ljubljani,Fakulteta za računalništvo in informatiko. Ljubljana.

Šuklje, J. (2023). *Computer vision application for solving a Rubik's cube in real-time*. 1.2.2024: https://www.youtube.com/watch?v=qAvpx65JOVI