

# Počítačové videnie

---

## Rekonštrukcia Rubikovej kocky

Michal Bubnár, Michal Hozza, Matej Kopernický

<https://github.com/mhozza/rubik>

## Zadanie

- Zrekonštruovať Rubikovu kocku z fotografie



## Zadanie

- ▶ Zrekonštruovať Rubikovu kocku z fotografie
- ▶ Získať 3D model kocky s farbami každého políčka na každej stene kocky



## Postup práce

- ▶ Identifikovať pozíciu kocky na obrázku

# Postup práce

- ▶ Identifikovať pozíciu kocky na obrázku
- ▶ Identifikovať pozíciu políčok na kocke

## Postup práce

- ▶ Identifikovať pozíciu kocky na obrázku
- ▶ Identifikovať pozíciu políčok na kocke
- ▶ Priradiť políčka k stene kocky a správne ich zoradiť

## Postup práce

- ▶ Identifikovať pozíciu kocky na obrázku
- ▶ Identifikovať pozíciu políčok na kocke
- ▶ Priradiť políčka k stene kocky a správne ich zoradiť
- ▶ Nájsť farbu každého políčka

## Postup práce

- ▶ Identifikovať pozíciu kocky na obrázku
- ▶ Identifikovať pozíciu políček na kocke
- ▶ Priradiť políčka k stene kocky a správne ich zoradiť
- ▶ Nájsť farbu každého políčka
- ▶ Vizualizovať výsledok



## Postup práce



## Identifikácia kocky

- ▶ Hľadáme význačné charakteristiky kocky

## Identifikácia kocky

- ▶ Hľadáme význačné charakteristiky kocky
- ▶ Prvý pokus – nájsť čiary medzi políčkami pomocou Houghovej transformácie

## Identifikácia kocky

- ▶ Hľadáme význačné charakteristiky kocky
- ▶ Prvý pokus – nájsť čiary medzi políčkami pomocu Houghovej transformácie
- ▶ Bez úspechu

# Identifikácia kocky

- ▶ Hľadáme význačné charakteristiky kocky
- ▶ Prvý pokus – nájsť čiary medzi políčkami pomocu Houghovej transformácie
- ▶ Bez úspechu
- ▶ Hľadanie políčok

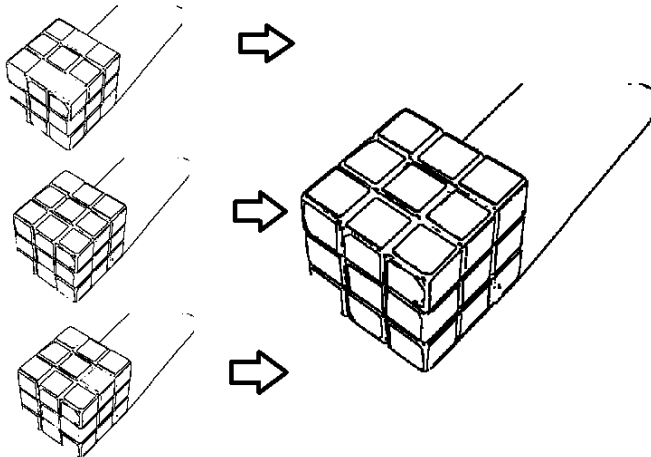
# Identifikácia kocky

- ▶ Hľadáme význačné charakteristiky kocky
- ▶ Prvý pokus – nájsť čiary medzi políčkami pomocu Houghovej transformácie
- ▶ Bez úspechu
- ▶ Hľadanie políčok
- ▶ Nájdeme uzavreté oblasti na binárnom obrázku

# Hľadanie hrán

- ▶  $E$  = zjednotenie hrán zo všetkých farebných zložiek + erózia

## Hľadanie hrán

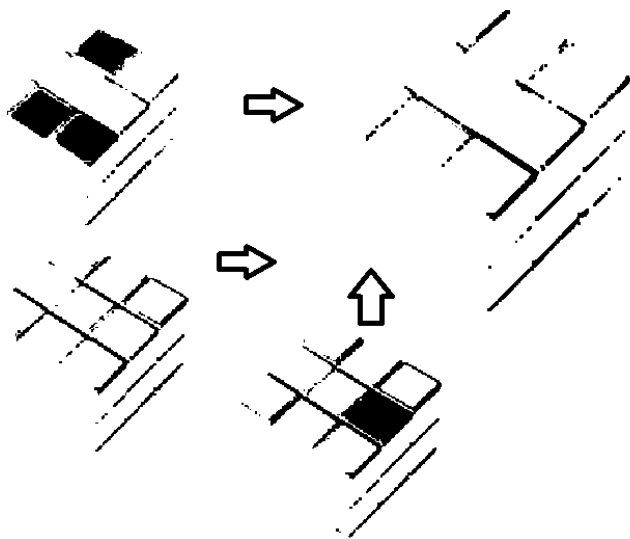




## Hľadanie hrán

- ▶  $E$  = zjednotenie hrán zo všetkých farebných zložiek + erózia
- ▶  $C$  = prienik jednotlivých farebných zložiek + erózia

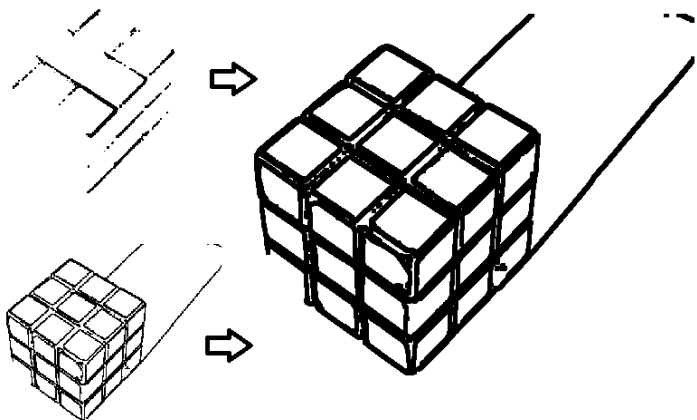
## Hľadanie hrán



# Hľadanie hrán

- ▶  $E$  = zjednotenie hrán zo všetkých farebných zložiek + erózia
- ▶  $C$  = prienik jednotlivých farebných zložiek + erózia
- ▶  $C \cup E$  + erózia = výsledný hranový obrázok

## Hľadanie hrán



## Hľadanie políček

- ▶ Chceme vyseparovať uzavreté oblasti

## Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií

# Hľadanie políček

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku



# Hľadanie políček

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti
  - Pomer hlavnej a vedľajšej osi opísanej elipsy

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti
  - Pomer hlavnej a vedľajšej osi opísanej elipsy
  - Eulerovo číslo oblasti

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti
  - Pomer hlavnej a vedľajšej osi opísanej elipsy
  - Eulerovo číslo oblasti
- ▶ Vylúčenie outlierov

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti
  - Pomer hlavnej a vedľajšej osi opísanej elipsy
  - Eulerovo číslo oblasti
- ▶ Vylúčenie outlierov
  - Vzdialenosť od najväčšieho zhluku

# Hľadanie políčok

- ▶ Chceme vyseparovať uzavreté oblasti
- ▶ Použité množstvo kritérií
  - Veľkosť oblasti
  - Pozícia oblasti na obrázku
  - Solidita oblasti
  - Pomer hlavnej a vedľajšej osi opísanej elipsy
  - Eulerovo číslo oblasti
- ▶ Vylúčenie outlierov
  - Vzdialenosť od najväčšieho zhluku
  - Vzájomné porovnanie veľkostí oblastí

## Hľadanie políčok



## Zaradenie políčok

- Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny



## Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)

## Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)
  - Line sweeping metóda s čiarami pod  $45^\circ$  uhlom

# Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)
  - Line sweeping metóda s čiarami pod  $45^\circ$  uhlom
  - Optimalizácia počítaním skóre pre každý pixel obvodu

## Zaradenie políčok



## Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)
  - Line sweeping metóda s čiarami pod  $45^\circ$  uhlom
  - Optimalizácia počítaním skóre pre každý pixel obvodu
- ▶ Zo vzájomnej pozície rohov vieme získať sklony hrán políčok

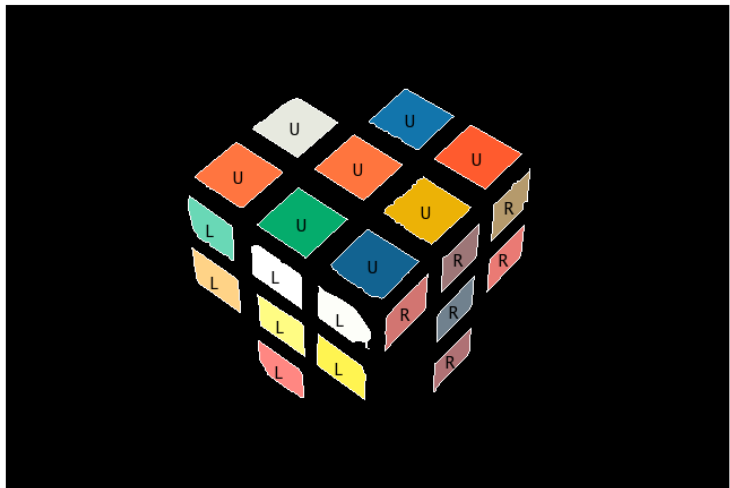
## Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)
  - Line sweeping metóda s čiarami pod  $45^\circ$  uhlom
  - Optimalizácia počítaním skóre pre každý pixel obvodu
- ▶ Zo vzájomnej pozície rohov vieme získať sklony hrán políčok
- ▶ Kritérium pre priradenie k stene kocky

## Zaradenie políčok

- ▶ Políčka chceme priradiť ku stene kocky a zoradiť ich v rámci steny
- ▶ Nájdeme rohy políčok (ľavý horný/dolný, pravý horný/dolný)
  - Line sweeping metóda s čiarami pod  $45^\circ$  uhlom
  - Optimalizácia počítaním skóre pre každý pixel obvodu
- ▶ Zo vzájomnej pozície rohov vieme získať sklony hrán políčok
- ▶ Kritérium pre priradenie k stene kocky
- ▶ Eliminácia zle zaradených políčok na základe vzájomnej polohy

## Zaradenie políčok

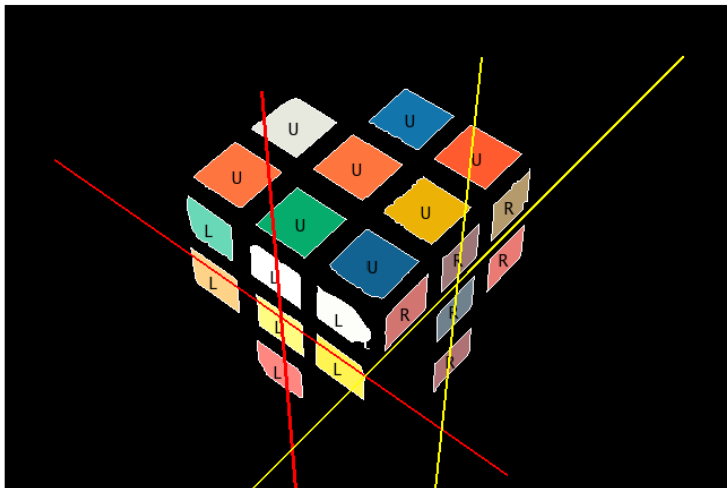




## Chýbajúce políčka

- Nájďme osi bočných strán kocky z hrán políčok

## Chýbajúce políčka



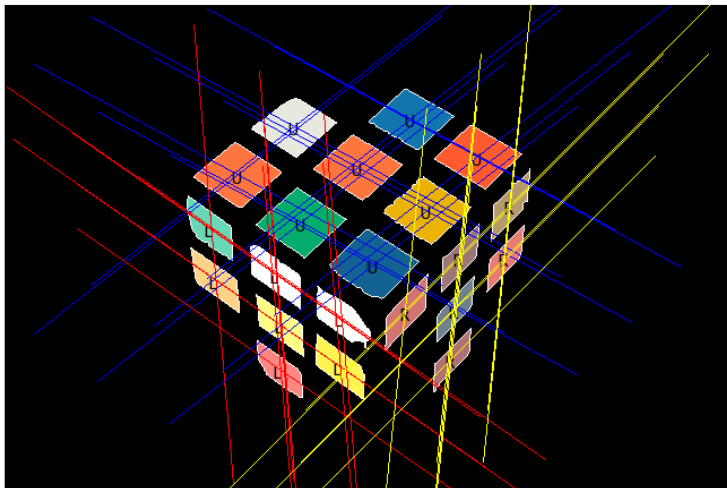
## Chýbajúce políčka

- ▶ Nájďme osi bočných strán kocky z hrán políčok
- ▶ Cez centroidy políčok vedieme osi a získame mriežku

## Chýbajúce políčka

- ▶ Nájdeme osi bočných strán kocky z hrán políčok
- ▶ Cez centroidy políčok vedieme osi a získame mriežku
- ▶ Pre hornú stenu použijeme osi bočných s kompenzáciou perspektívy

## Chýbajúce políčka



## Chýbajúce políčka

- ▶ Priesečníky vedených osí sa stretnú v centroidoch políčok

## Chýbajúce políčka

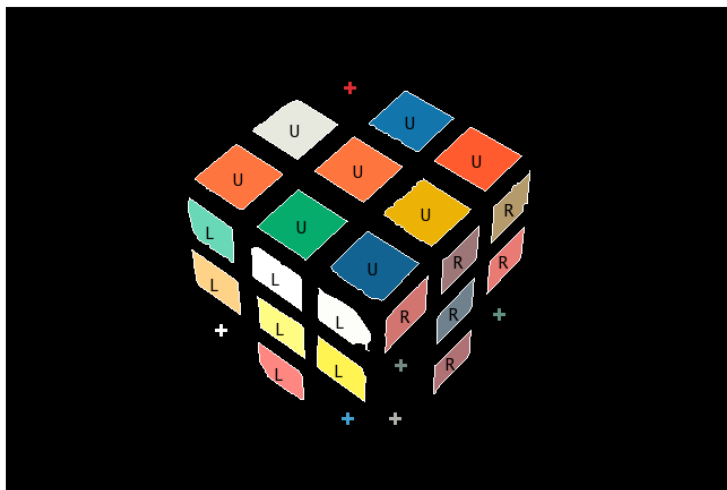
- ▶ Priesečníky vedených osí sa stretnú v centroidoch políčok
- ▶ Priesečníky príliš vzdialené od známych políčok predstavujú chýbajúce

## Chýbajúce políčka

- ▶ Priesečníky vedených osí sa stretnú v centroidoch políčok
- ▶ Priesečníky príliš vzdialené od známych políčok predstavujú chýbajúce
- ▶ Nájdeme zhluky pomocou k-means



## Chýbajúce políčka



## Zoradenie políčok

- Pre vizualizáciu potrebujeme poznať aj poradie políčok v rámci steny

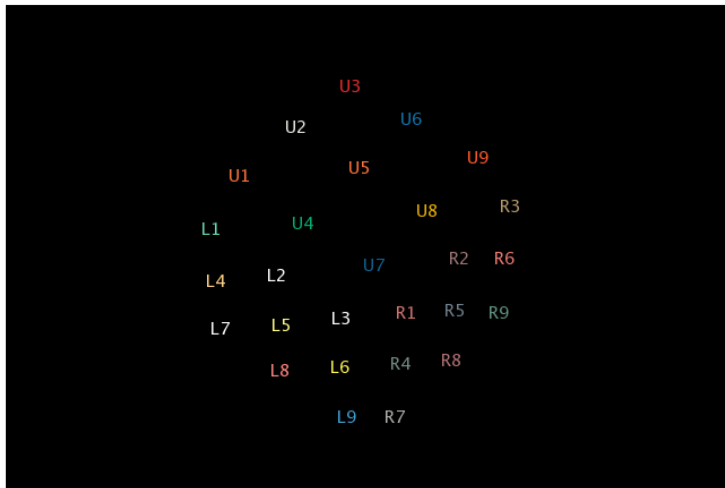
## Zoradenie políček

- ▶ Pre vizualizáciu potrebujeme poznať aj poradie políček v rámci steny
- ▶ Line sweeping pod uhlom osí pre získanie stĺpcov

## Zoradenie políček

- ▶ Pre vizualizáciu potrebujeme poznať aj poradie políček v rámci steny
- ▶ Line sweeping pod uhlom osí pre získanie stĺpcov
- ▶ Políčka v stĺpci zoradíme podľa súradníc

# Zoradenie políčok



## Priradenie nájdených farieb k farbám Rubikovej kocky

- Chceme nájdené farby priradiť k farbám originálnej Rubikovej kocky

## Priradenie nájdených farieb k farbám Rubikovej kocky

- ▶ Chceme nájdené farby priradiť k farbám originálnej Rubikovej kocky
- ▶ Rubikova kocka má 6 základných farieb
  - ▶ Red: (196, 30, 58)
  - ▶ Green: (0, 158, 96)
  - ▶ Blue: (0, 81, 186)
  - ▶ Orange: (255, 88, 0)
  - ▶ Yellow: (255, 213, 0)
  - ▶ White: (255, 255, 255)



## Priradenie nájdených farieb k farbám Rubikovej kocky

- ▶ Chceme nájdené farby priradiť k farbám originálnej Rubikovej kocky
- ▶ Rubikova kocka má 6 základných farieb
  - ▶ Red: (196, 30, 58)
  - ▶ Green: (0, 158, 96)
  - ▶ Blue: (0, 81, 186)
  - ▶ Orange: (255, 88, 0)
  - ▶ Yellow: (255, 213, 0)
  - ▶ White: (255, 255, 255)
- ▶ Jednoduchý algoritmus matchovania farby k farbe





# Priradenie nájdených farieb k farbám Rubikovej kocky

- ▶ Chceme nájdené farby priradiť k farbám originálnej Rubikovej kocky

- ▶ Rubikova kocka má 6 základných farieb

- ▶ Red: (196, 30, 58)
- ▶ Green: (0, 158, 96)
- ▶ Blue: (0, 81, 186)
- ▶ Orange: (255, 88, 0)
- ▶ Yellow: (255, 213, 0)
- ▶ White: (255, 255, 255)



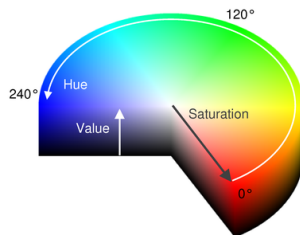
- ▶ Jednoduchý algoritmus matchovania farby k farbe
- ▶ Zložitejší algoritmus zgrupovania a matchovania skupiny k farbe

# Matchovanie jednej farby na farbu Rubikovej kocky

- ▶ Prevod oboch farieb do HSV

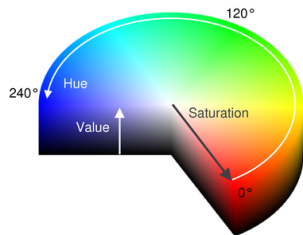
# Matchovanie jednej farby na farbu Rubikovej kocky

- ▶ Prevod oboch farieb do HSV
- ▶ Spočítanie skóre pre všetky farby a následný výber farby s najväčším skóre



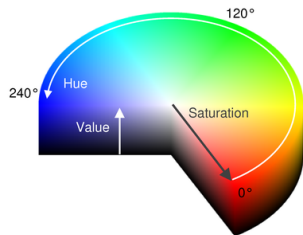
# Matchovanie jednej farby na farbu Rubikovej kocky

- ▶ Prevod oboch farieb do HSV
- ▶ Spočítanie skóre pre všetky farby a následný výber farby s najväčším skóre
  - ▶ rozdiel 2 farieb po zložkách



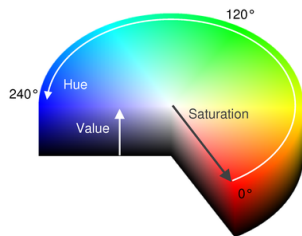
# Matchovanie jednej farby na farbu Rubikovej kocky

- ▶ Prevod oboch farieb do HSV
- ▶ Spočítanie skóre pre všetky farby a následný výber farby s najväčším skóre
  - ▶ rozdiel 2 farieb po zložkách
  - ▶ výsledné skóre je váhovaný súčet zložiek



# Matchovanie jednej farby na farbu Rubikovej kocky

- ▶ Prevod oboch farieb do HSV
- ▶ Spočítanie skóre pre všetky farby a následný výber farby s najväčším skóre
  - ▶ rozdiel 2 farieb po zložkách
  - ▶ výsledné skóre je váhovaný súčet zložiek
  - ▶ pre každú cieľovú farbu sú rôzne váhy

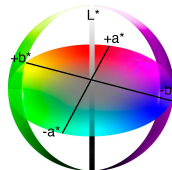


## Zgrupovanie podobných farieb

- ▶ Zgrupenie farieb podľa podobnosti, pre každú stranu zvlášť

# Zgrupovanie podobných farieb

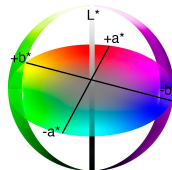
- ▶ Zgrupenie farieb podľa podobnosti, pre každú stranu zvlášť
  - ▶ prevod do *Lab*





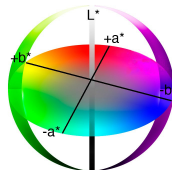
# Zgrupovanie podobných farieb

- ▶ Zgrupenie farieb podľa podobnosti, pre každú stranu zvlášť
  - ▶ prevod do  $Lab$
  - ▶ euklidovská vzdialenosť na zložkách  $a$ ,  $b$ 
$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$



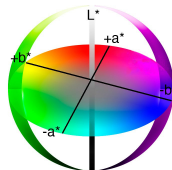
# Zgrupovanie podobných farieb

- ▶ Zgrupenie farieb podľa podobnosti, pre každú stranu zvlášť
  - ▶ prevod do *Lab*
  - ▶ euklidovská vzdialenosť na zložkách *a*, *b*
$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$
  - ▶ treshold na vyhodnotenie podobnosti



# Zgrupovanie podobných farieb

- ▶ Zgrupenie farieb podľa podobnosti, pre každú stranu zvlášť
  - ▶ prevod do  $Lab$
  - ▶ euklidovská vzdialenosť na zložkách  $a$ ,  $b$ 
$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$
  - ▶ threshold na vyhodnotenie podobnosti
- ▶ Priradenie farieb ku grupám - nájsť čo najlepšie ohodnotenie farieb tak, aby žiadne 2 grupy nemali tú istú farbu



# Literatúra



Justin Ng: *Automated Rubik's Cube Recognition* [online].

<[http://www.stanford.edu/class/ee368/Project\\_12/Reports/...](http://www.stanford.edu/class/ee368/Project_12/Reports/...)>.



Andrej Karpathy: *Extracting sticker colors on Rubik's Cube* [online].

<<http://karpathy.ca/portfolio/project525.php>>.



Włodzimierz Kasprzak, Wojciech Szynkiewicz, Łukasz Czajka: *Rubik's cube reconstruction from single view for Service robots*, Machine Graphics & Vision International Journal archive, Volume 15 Issue 3, February 2006, Pages 451-459



Yoann Bourse: *Controle 3D par un Rubik's Cube* [online].

<<http://www.yoannbourse.com/ressources/docs/ens/stereovision.pdf>>.