

# Praktikum zo strojového učenia a umelej inteligencie na vizuálnych dátach

*Andrej Lúčny*

*Katedra aplikovanej informatiky FMFI UK*

*lucny@fmph.uniba.sk*

*[http://dai.fmph.uniba.sk/w/Andrej\\_Lucny](http://dai.fmph.uniba.sk/w/Andrej_Lucny)*

*[www.agentspace.org/praktikum](http://www.agentspace.org/praktikum)*

# Tracker-y

- Detektor spracúva každý obrázok zvlášť, zatiaľ čo tracker pracuje vždy s videom
- Jednoduchý tracker = detector + outlier filtering

# Kalmanov filter

O nejakom systéme vieme z jeho vnútorného stavu predikovať vývoj jeho vnútorného i vonkajšieho stavu, pričom predikcia je zatažená gaussovským šumom

Vonkajší stav systému vieme odmerať, pričom meranie je zatažené gaussovským šumom.

Kalmanov filter dáva optimálny odhad skutočného vnútorného i vonkajšieho stavu systému za týchto okolností.

# Kalmanov filter

Používa sa na vyhladenie pohybu trackovaného objektu (v takom prípade je vnútorným stavom rýchlosť a vonkajším poloha)

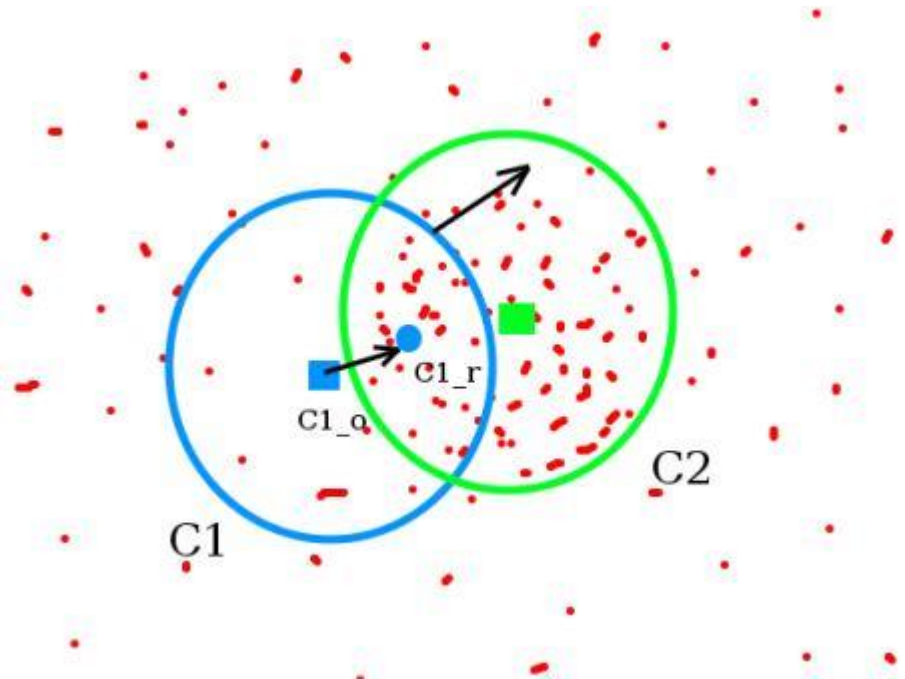
tracker = detektor + Kalmanov filter

Pričom detektor nemusíme volať pre každý snímok, keďže na určitý čas sa môžeme spoľahnúť na predikciu. Kalmanov filter taktiež odstraňuje falošné detekcie

*(Kalmanov filter sa dá využiť aj na filtrovanie šumu na videu - každý pixel sa rieši zvlášť)*

# MeanShift

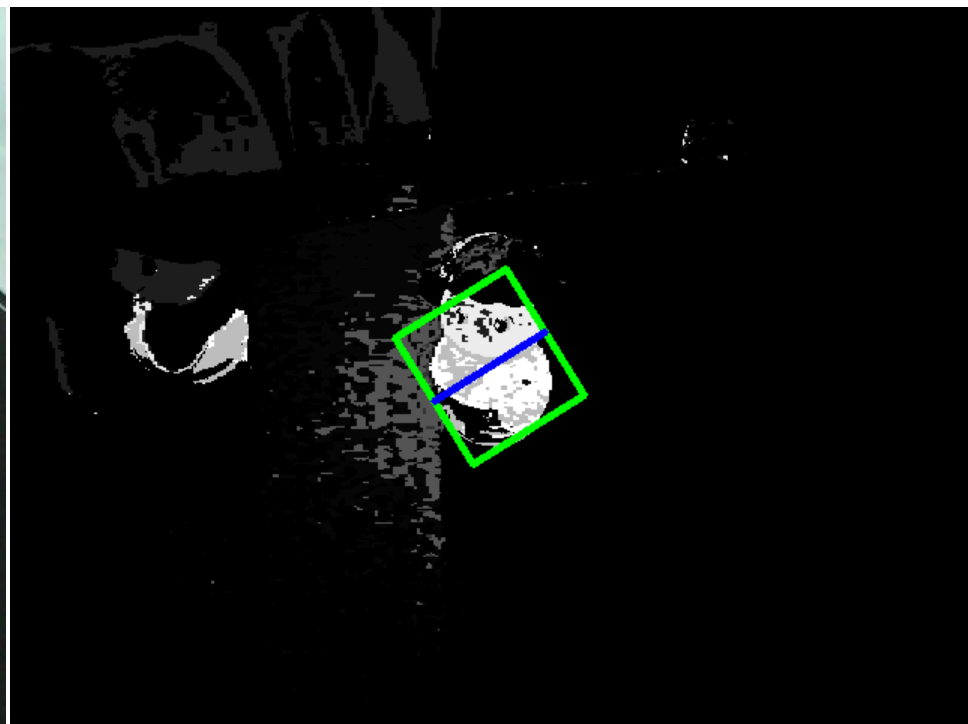
1. V rámci nejakého regiónu máme body patriace sledovanému objektu
2. Región premiestnime do ich ťažiska
3. Proces opakujeme



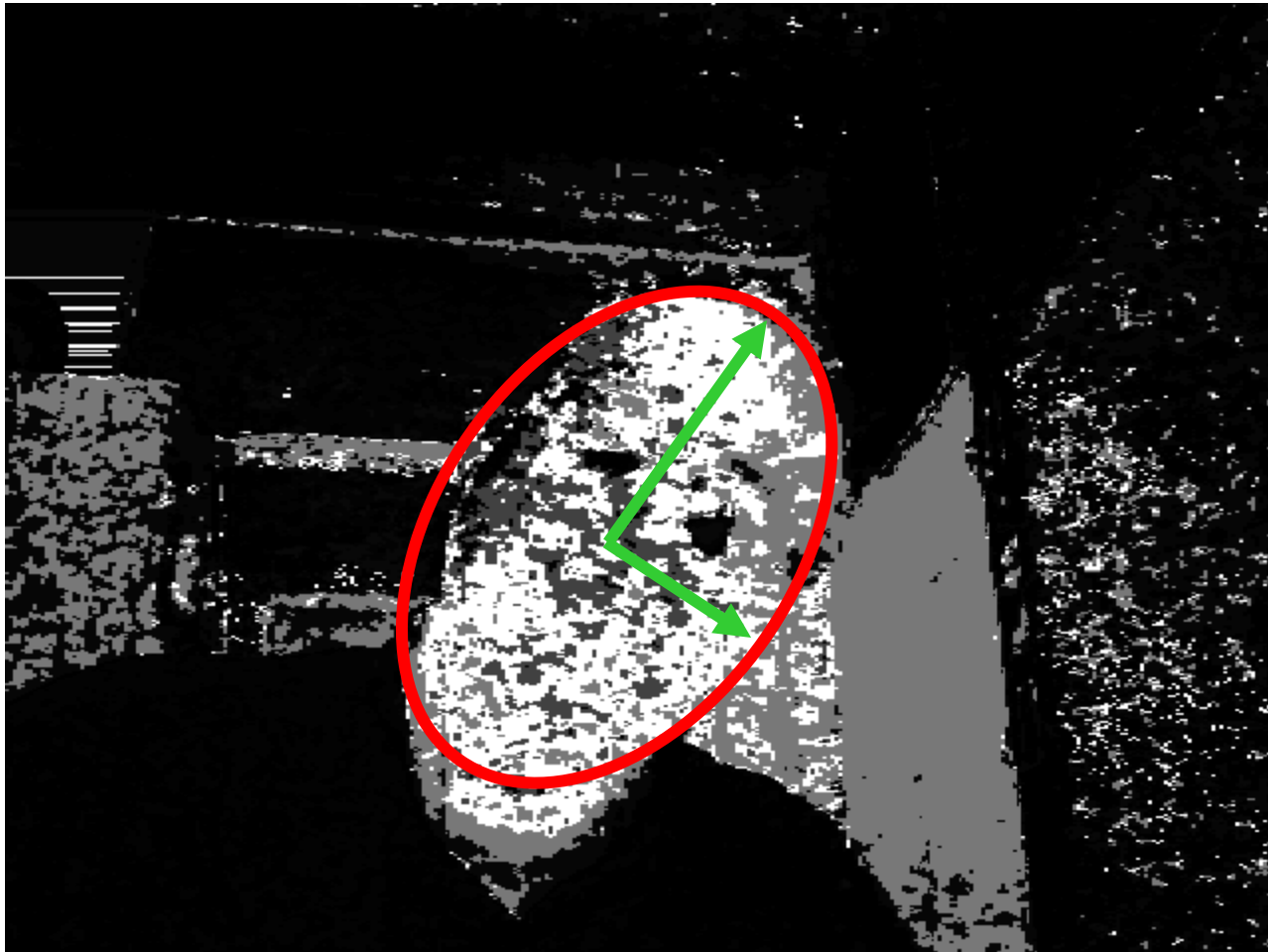
# MeanShift / CamShift tracker

1. Z vybranej oblasti sa urobí histogram farieb
2. Urobí sa šedý backProjection image v ktorom sú farby reprezentované ich výskytom vo zvolenom histograme
3. Pomocou MeanShift posúvame oblasť tak, že zodpovedá najintenzívnejším bodom. (MeanShift je vo všeobecnosti posun z bodu do priemeru blízkych bodov s podobnou farbou)
4. Pomocou CamShift hľadáme v oblasti správnu rotáciu

# MeanShift / CamShift tracker



# Back projection image





# Multiply instance learning (MIL)

- Haar features



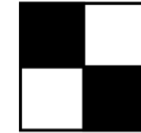
(1)



(2)



(3)



(4)

we put on ROI set of 2, 3 or 4 rectangles  
and calculate difference of white and black  
areas totals

- Initial Region of Interest (ROI)

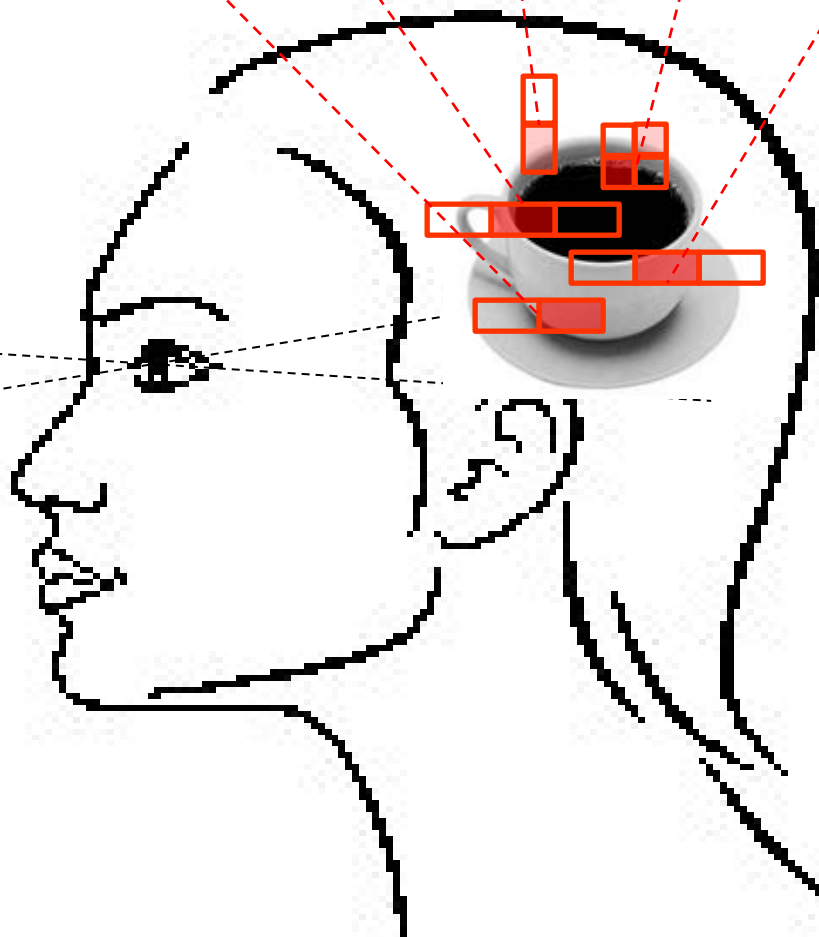
Then we teach Bayes classifier of bags of positive (ROIs close to the current ROI) and negative (far ROIs) examples

- Motion model: neighborhood ROI

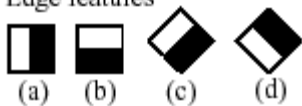
we move to neighborhood ROI which has the highest ranging provided by the classifier

# MIL tracker

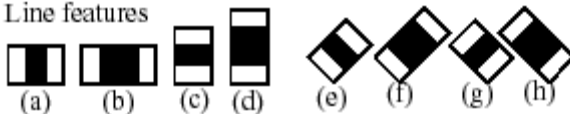
$(\Sigma_{\square}-\Sigma_{\square}, \Sigma_{\square}-\Sigma_{\square}, \Sigma_{\square}-\Sigma_{\square}, \Sigma_{\square}-\Sigma_{\square}, \Sigma_{\square}-\Sigma_{\square})$



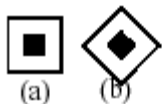
## 1. Edge features



## 2. Line features



## 3. Center-surround features



# Ďalšie tracker-y

- Boosting
- KFC
- TLD
- MedianFlow

# Inicializácia trackera

- Užívateľ označí objekt na obraze

alebo

- Detektor nájde objekt v štandardnej polohe a veľkosti

alebo

- Objekt je zdetekovaný detektorom pohybu

Alebo

- Objekt vyčnieva z obrazu (saliency)

# Detektor pohybu

- Jednoduchý detektor pohybu porovnáva predchádzajúci a nasledujúci obrázok a hľadá rozdiely. Tie zodpovedajú kontúre pohybujúceho sa objektu
- Lepší detektor je založený detekcií výčnelkov na obraze

# Saliency

- „Výčnelkovitosť“
- Výrazné hrany sa podobajú na step function
- Vieme, že jej FT obsahuje všetky frekvencie
- Ak niekde nájdeme vo Fourierovom spektre všetky frekvencie, predpokladáme, že to vyčnieva.
- Rozdiely v saliency umožňujú kvalitnejší detektor pohybu

# Saliency

