

# Eliminacija pozadine u video zapisima

Seminarski rad u okviru kursa  
Naučno izračunavanje  
Matematički fakultet

Vesna Katanić, Anja Ivanišević

14. septembar 2019.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Primenjeni algoritmi</b>	<b>2</b>
2.1	Basic Motion Detection . . . . .	2
2.2	Gaussian Mixture Model . . . . .	2
2.3	K Nearest Neighbours . . . . .	3
<b>3</b>	<b>Evaluacija algoritma</b>	<b>3</b>

# 1 Uvod

Algoritmi za eliminaciju pozadine u video zapisima su algoritmi koji imaju za cilj detekciju kretanja i odvajanje pozadine od objekata koji se kreću. Cilj ovog rada je upoznavanje različitim algoritmima za detekciju kretanja i njihovo poređenje.

## 2 Primljeni algoritmi

U ovom radu smo poredili tri različita algoritma za eliminaciju pozadine u video zapisima. Ti algoritmi su: *Basic Motion Detection (BMD)*, *Gaussian Mixture Model (GMM)* i *K Nearest Neighbours (KNN)*. Ideja svakog od ovih algoritama je da transformišu ulazni video u novi video u kojem će pozadina biti ofarbana u crno a objekti koji se kreću u belo, kako bi se oni izdvojili. U nastavku rada će ovi algoritmi biti predstavljeni.

### 2.1 Basic Motion Detection

*Basic Motion Detection* je najjednostavniji algoritam od predstavljenih algoritama. U ovom algoritmu se polazi od pretpostavke da se video  $I$  sastoji od statičke pozadine  $B$  ispred koje se nalaze objekti koji se kreću. Kako bismo detektovali objekte računa se rastojanje trenutnog modela pozadine i posmatranog frejma. Na osnovu ovog rastojanja pravi se rezultujuća crno-bela slika.

Model pozadine se ažurira na osnovu prethodnog stanja i trenutno posmatranog frejma po sledećoj formuli:  $B_{s,t+1} = (1 - \alpha) * B_{s,t} + \alpha * I_{s,t}$  gde je  $s$  posmatrani piksel,  $t$  posmatrani frejm i  $\alpha$  parametar za koju je uzeta vrednost 0.001. Za početnu vrednost modela pozadine  $B$  je uzet prvi frejm, dok je rastojanje između frejma i modela računato na osnovu Euklidskog rastojanja.

### 2.2 Gaussian Mixture Model

Ideja algoritma *Gaussian Mixture Model* jeste da u svakom trenutku  $t$  sa osnovu poslednjih  $T$  frejmova računamo verovatnoću da se određena boja javi u posmatranom pikselu. Verovatnoća se računa kombinovanjem najviše  $M$  Gausovih raspodela po formuli:  $p = \sum_{m=1}^M \pi_m N(x, \mu_m, \sigma_m^2)$ , a u kojima konfiguriraju parametri  $\pi$ ,  $\mu$  i  $\sigma^2$  koje iterativno ažuriramo po sledećim formulama:

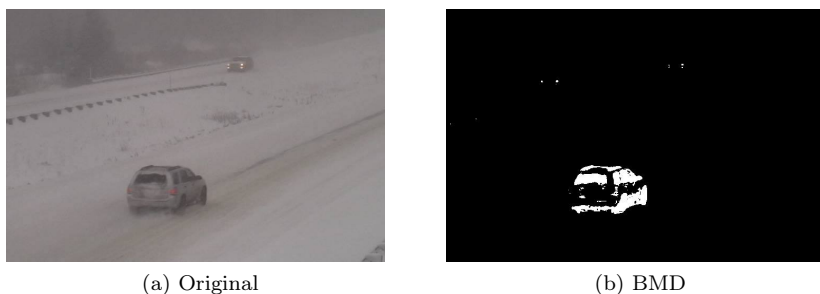
$$\begin{aligned}\pi_m &\leftarrow \pi_m + \alpha(\phi_m - \pi_m), \\ \mu_m &\leftarrow \mu_m + \phi_m(\alpha/\pi_m)\delta_m, \\ \sigma_m^2 &\leftarrow \sigma_m^2 + \phi_m(\alpha/\pi_m)(\delta_m^T \delta_m - \sigma_m^2).\end{aligned}$$

Za svaki novi piksel, parametar  $\phi_m$  će biti jednak 1 za komponentu sa najvećim  $\pi$  ako je "blizu" posmatranog piksela, u suprotnom će biti jednak 0. Kažemo da je piksel "blizu" komponente ako je Mahalanobisovo rastojanje manje od neke unapred zadate granice (na primer 3). Ako nijedna komponenta nije "blizu" posmatranog piksela, generišemo novu komponentu sa sledećim parametrima:

Ako smo dostigli maksimalan broj komponenti  $M$ , tada ćemo zameniti komponentu sa najmanjom vrednošću parametra  $\pi$  sa novokreiranom komponentom.

Model pozadine će se sastojati od  $B$  komponenti sa najvećim vrednostima parametra  $\pi$ , dok će ostale komponente predstavljati objekte koji se kreću. Broj  $B$  se računa po formuli:

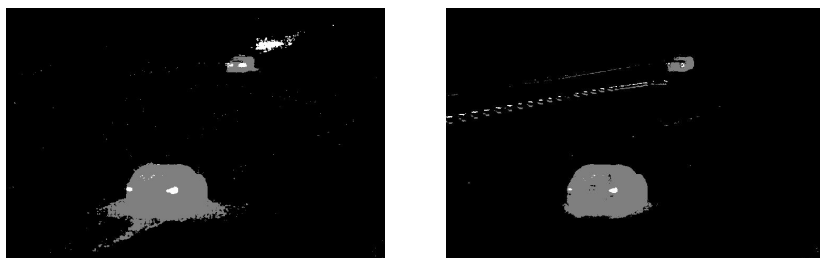
$$B = \arg \min_b (\sum_{m=1}^b \pi_m > (1 - c_f)).$$



(a) Original

(b) BMD

Slika 1: Originalna slika i rezultat BMD algoritma



(a) GMM

(b) KNN

Slika 2: Rezultati GMM i KNN algoritama

Implementacija ovog algoritma postoji u OpenCV biblioteci i nju smo koristili za potrebe ovog rada.

### 2.3 K Nearest Neighbours

Ovaj algoritam predstavlja unapređenje algoritma *Kernel density estimation* algoritma. Za razliku od njega gdje smo koristili fiksnu veličinu kernela  $D$  ovom algoritmu prilagođavamo  $D$  za svaki piksel posebno. Umesto da pokušavamo da nađemo  $D$  koje je globalno optimalno, mi povećavamo širinu  $D$  dokle god unapred zadata količina podataka nije pokrivena. Na ovaj način dobijamo velike kernele u oblastima sa malom količinom podataka i male kernele gdje je gustina podataka veća. Ovaj algoritam daje jako dobre rezultate ako ima malo objekata koji se kreću. Implementacija ovog algoritma postoji u OpenCV biblioteci i nju smo koristili za potrebe ovog rada.

## 3 Evaluacija algoritma

Na narednim slikama su redom prikazani ulazna slika i rezultati algoritama *BMD*, *GMM* i *KNN*.

Algoritmi su pokretani na tri vrste test primera - saobraćaj za vreme mećave, saobraćaj u standardnim uslovima i noćni snimak nadzorne kamere. Svaki od algoritama je posmatran kao problem klasifikacije i za evaluaciju rezultata korišćene su mere *Precision*, *Recall* i *F1 mera*.

Ostvareni su sledeći rezultati:

BMD	GMM	KNN
0.9409	0.4237	0.5218
0.4757	0.1820	0.2738
0.9822	0.8972	0.8872

BMD	GMM	KNN
0.4528	0.9414	0.9157
0.7833	0.8926	0.9025
0.9436	0.7184	0.9068

BMD	GMM	KNN
0,6114	0,5844	0,6648
0,5919	0,3023	0,4201
0,9625	0,7979	0,8969