

# Detekcija pomaka u ugradbenim računalnim sustavima niske potrošnje

Josip Grlica, Ivan Pavić, Josip Puškar, Ivan Soldo, Ivan Spasić

24. siječnja 2016.

## Sažetak

Rad daje pregled algoritama za detekciju pomaka. Detaljnije je obrađen princip  $\Sigma\Delta$  estimacije pozadine. Algoritmi su stavljeni u kontekst implementacije u ugradbenim računalnim sustavima niske potrošnje. Cilj rada je naći algoritam za što robusniju detekciju uz očuvanje učinkovitosti. Algoritam koji je implementiran jezgra je minimalnog sustava za videonadzor na *BeagleBone Black* platformi.

## 1. Uvod

Algoritam za detekciju pomaka mora precizno odvojiti objekte koji se kreću od statične pozadine. Problem detekcije svodi se na klasifikaciju piksela slike u dva razreda. Prvi razred je pozadina (engl. *background*) kojoj odgovaraju pikseli koji pripadaju statičnoj okolini. U drugi razred spadaju pomični pikseli koji čine objekt koji se kreće (engl. *foreground*). Pri klasifikaciji piksela različiti su zahtjevi na osjetljivost i specifičnost detektora. U tom smislu najvažnija je robustnost algoritma, odnosno prilagođenost pomičnim objektima različitih brzina i veličina.

Zbog što veće autonomnosti sustava u pogledu napajanja, sustav ne smije trošiti puno energije. Nadalje, prilikom obrade slike radi se s relativno velikom količinom podataka što je memorijski i procesorski zahtjevno. U radu osim pregleda metoda koje se već duže vrijeme koriste za estimaciju pozadine razrađen je i način koji bi omogućio robusniju detekciju uz manju potrošnju memorije i procesorske moći računala uz istovremeno smanjenje potrošnje. Algoritam koji je razvijen temeljen je na principu  $\Sigma\Delta$  estimacije koja se koristi u većini današnjih analogno digitalnih pretvornika.

## 2. Metode estimacije pozadine slike

Postoje razni načini estimacije pozadine slike. U ovom radu metode su podijeljene u dvije skupine. Prilikom pregleda koristit će se sljedeće oznake:  $I(t)$  - slika u trenutku  $t$ ,  $B(t)$  - pozadina u trenutku  $t$ .

### 2.1. Metode diferenciranja

Najjednostavnija metoda je oduzimanje okvira (engl. *frame differencing*). Metoda pretpostavlja da je pozadina u nekom trenutku  $t$  jednaka slici na ulazu. Metoda oduzima vrijednosti slike i pozadine na mjestu  $(x, y)$  te apsolutnu vrijednost razlike uspoređuje s pragom odlučivanja. Zapisano formulom za svaki pojedini piksel:

$$|B(x, y, t) - I(x, y, t - 1)| \quad (1)$$

$$|I(x, y, t) - B(x, y, t)| > Th \quad (2)$$

Konačno iz (1) i (2):

$$|I(x, y, t) - I(x, y, t - 1)| > Th \quad (3)$$

Nešto složenija metoda estimacije pozadine je procjena usrednjavanjem (engl. *mean filter*). Pretpostavka metode je da je pozadina u trenutku  $t$  jednaka usrednjenoj vrijednosti prethodnih slika u algoritam.

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t - i) \quad (4)$$

Za usporedbu s pragom koristimo izraz:

$$|I(x, y, t) - \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t - i)| > Th \quad (5)$$

Mogu se koristiti i nelinearne metode. Primjer takve metode je procjena medianom.

$$B(x, y, t) = \text{median}(I(x, y, t - i)) \quad (6)$$

$$|I(x, y, t) - \text{median}(I(x, y, t - i))| > Th \quad (7)$$

$$i = 1, 2..n$$

Velika prednost ovih metoda je što su jednostavne za implementaciju i korištenje. Mogu biti relativno brze ako se implementiraju na pravilan način. Srednja vrijednost pozadine nije konstantna i mijenja se s vremenom što ih čini prilagodljivima na promjenu pozadine.

Nedostatak *frame differencing* metode je to što preciznost ovisi o brzini kretanja objekta na slici i vremena uzimanja pojedine slike (engl. *frame rate*). Nedostaci median metode i usrednjavanja su memorijski i procesorski. Kod median metode ne može se odrediti median piksela ako u memoriji ne postoji  $n$  prethodnih piksela što zahtijeva veliki potrošak memorije. U slučaju usrednjavanja potrošnja memorije može se izbjeći metodom *running average* ako prilagodimo izraz (4):

$$B(x, y, t) = \frac{t-1}{t} B(x, y, t-1) + \frac{1}{t} I(x, y, t) \quad (8)$$

Međutim i ovakav pristup ima kritičan nedostatak. Naime ako je  $t$  jako velik ovo je teško numerički izvedivo na uređajima sa *fixed-point* aritmetikom. Još jedan nedostatak ovakvih metoda je fiksni globalni prag za sve piksele koji nije promjenjiv s vremenom.

## 2.2. Prilagodljivi modeli pozadine za praćenje u stvarnom vremenu

U ovom poglavlju opisana je osnovna ideja modela pozadine *Adaptive Gaussian Mixture Model for Background Subtraction* prema [3] dalje u tekstu GMM metoda (engl. *Gaussian Mixture Model*). GMM metoda je adaptivna metoda koja ima mogućnost prilagodbe multimodalnoj pozadini. Koristi različiti prag odluke za svaki od piksela i dodatno je taj prag promjenjiv u vremenu što rezultira dobrom prilagodbom na promjenu osvjetljenja. Ipak, ako su promjene osvjetljenja nagle algoritam neće dobro reagirati.

Algoritam na temelju početno izabranih parametara i početne pozadine generira  $K$  Gaussovih distribucija za pojedini piksel. Vrijednost parametra  $K$  je najčešće od 3 do 5, a svaka od distribucija osim varijance i srednje vrijednosti sadrži parametar težine. Nadalje za svaki piksel u nekom trenutku  $t$  nalazi se optimalna od  $K$  distribucija za piksel odnosno provjerava se za koju distribuciju piksel upada unutar 2.5 standardne devijacije distribucija. Ako se pronađe takva distribucija njezini parametri se podešavaju na sljedeći način:

$$\mu_{(x,y),t} = (1 - \rho)\mu_{(x,y),t-1} + \rho I(x, y, t)$$

$$\sigma_{(x,y),t}^2 = (1 - \rho)\sigma_{(x,y),t-1}^2 + \rho(I(x, y, t) - \mu_{(x,y),t-1})^2$$

, gdje je:

$$\rho = \alpha N(I(x, y, t) | \mu_{(x,y),t-1}, \sigma_{(x,y),t-1}^2)$$

a  $\alpha$  je *learning rate*. Nakon podešavanja parametara razdiobe podešavaju se i težine razdioba:

$$\omega_{(x,y),t} = (1 - \alpha)\omega_{(x,y),t-1} + \alpha M_{(x,y),t}$$

gdje je  $M_{(x,y),t} = 1$  za pronađenu distribuciju  $M_{(x,y),t} = 0$  za ostale distribucije. Ako odgovarajuća distribucija nije pronađena koriste se različite heuristike za generiranje novih distribucija. Jedan od kriterija je i  $\omega/\sigma$  kriterij. Nove generirane distribucije imaju veliku varijancu i malu težinu.

## 3. Metoda procjene pozadine $\Sigma\Delta$ algoritmom

$\Sigma\Delta$  procjena pozadine je jednostavna nelinearna metoda oduzimanja pozadine (engl. *background subtraction*) zasnovana na usporedbi i jediničnom inkrementu. U nastavku ovog poglavlja razrađeni su principi metode i dani su primjeri algoritama. Dodatno, algoritam je uspoređen s ostalim algoritmima za procjenu pozadine.

### 3.1. Rekurzivne procjene pozadine

Uz pretpostavku da je svaki piksel slike na ulazu u algoritam opisan nekom funkcijom  $f_i(t)$  najjednostavniji način procjene vrijednosti pojedinog piksela bilo bi naći srednju vrijednost  $f_i(t)$ . Takav

pristup je već razrađen i predstavljen je izrazom (8). Općenito se srednja vrijednost signala u trenutku  $t$  može prikazati izrazom (u nastavku zbog jednostavnijeg zapisa promatra se samo vremenska dimenzija signala):

$$M_t = \frac{1}{t}I_t + \frac{t-1}{t}M_{t-1} \quad (9)$$

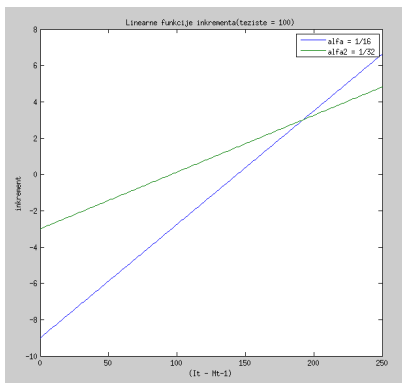
Ovakav pristup je numerički neizvediv za duže nizove stoga se uvodi konstantna težina ( $\alpha$ ) i dobiva generički zapis:

$$M_t = \alpha I_t + (1 - \alpha)M_{t-1} \quad (10)$$

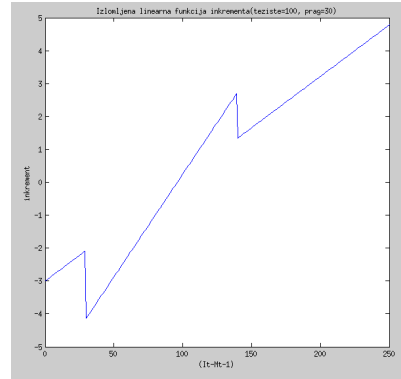
Ova jednadžba predstavlja *running average* metodu. Rješavanjem rekurzivne jednadžbe u  $Z$  domeni, očito je da se radi o visokopropusnom IIR filtru (tzv. *exponential smoothing filter*). Posljednji izraz može se prikazati u inkrementalnom obliku

$$M_t = M_{t-1} + \delta_t(I_t) \quad (11)$$

Ovisno o trenutnom uzorku signala mijenja se srednja vrijednost. U osnovnom slučaju  $\delta_t(I_t) = \alpha(I_t - M_{t-1})$  funkcija inkrementa je linearna i što je veća razlika  $\Delta$  između trenutnog uzorka signala i pozadine veći je utjecaj na procjenu pozadine što u osnovi nije dobro jer će male promjene biti u potpunosti zanemarene. Stoga se uvode različite težine  $\alpha$  za različiti raspon  $\Delta$ . Najčešće se uzima veća težina za one područja koja se smatraju pozadinskim. Za dvije različite težine stoga se uzima jedan fiksni prag. Postavljanje fiksnog praga je kritično u ovakvom načinu procjene zbog diskontinuiteta.



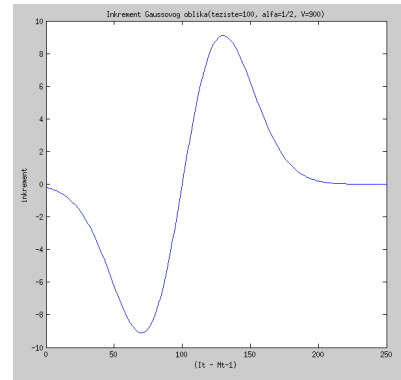
Slika 1.: Linearna funkcija inkrementa



Slika 2.: Izlomljena linearna funkcija inkrementa

Osim inkrementa oblika linearne funkcije [2] predlaže koristiti i inkrement oblikovan Gaussovom distribucijom. Ipak, u daljnjem razmatranju neće se koristiti.

$$\delta_t = \alpha_{max} e^{-\frac{(I_t - M_{t-1})^2}{2V_{t-1}}} (I_t - M_{t-1}) \quad (12)$$



Slika 3.: Funkcija inkrementa Gaussovog oblika

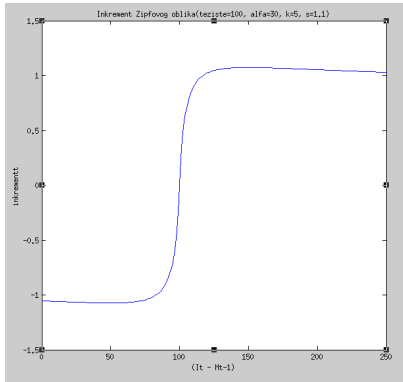
### 3.2. Zipfova razdioba i procjena pozadine

U daljnim razmatranjima koristit će se Zipfova distribucija za oblikovanje funkcije inkrementa. Zipfova distribucija odnosno Zipfov zakon izvorno je empirijski pokazan. Najpoznatiji primjer iz teorije informacije i lingvistike je pojava da je vjerojatnost pojave  $n$ -te najčešće riječi  $1/n$ . Općenito, radi se o

Zipf-Mandelbrot razdiobi čija je se funkcija razdiobe prema [2] može napisati:

$$Z_{(\mu,k,s)}(x) = \frac{(s-1)k^{s-1}}{2(|x-\mu|+k)^s} \quad (13)$$

gdje je  $\mu$  težište razdiobe,  $k$  varijanca. Ako iskoristimo ovakvu funkciju distribucije za oblikovanje inkrementa, tada inkrement postaje funkcija nalik step funkciji odnosno  $\delta_t = \kappa$  za  $x > 0$   $\delta_t = -\kappa$  za  $x < 0$  pri čemu je  $\kappa = \alpha * k^s$ . Ovo ponašanje direktno vodi na vezu između procjene srednje vrijednosti signala jediničnim inkrementom u  $\Sigma\Delta$  analognu digitalnim pretvornicima i Zipfove distribucije. Dodatno, određen je statistički model za promatranu pozadinu pa u samoj estimaciji koriste pretpostavljena statistička svojstva modela.



Slika 4.: Funkcija inkrementa Zipfovog oblika

### 3.3. Osnovni $\Sigma\Delta$ algoritam

Osnovni princip  $\Sigma\Delta$  algoritma je procijeniti parametre pozadine koristeći  $\Sigma\Delta$  modulaciju što je tipično u analogno digitalnoj pretvorbi. Za signal promjenjiv u vremenu  $f_t$  procjenjuje se diskretni signal  $d_t$  uzorkovan određenom frekvencijom. Za procjenu svakog uzorka diskretnog signala  $i$  koriste sljedeći izrazi:

$$d_t(i) = d_t(i-1) - \epsilon \quad (14)$$

za  $d_t(i-1) < f_t(i)$ , odnosno:

$$d_t(i) = d_t(i-1) + \epsilon \quad (15)$$

za  $d_t(i-1) > f_t(i)$  pri čemu je  $\epsilon$  korak diskretizacije.

U  $\Sigma\Delta$  računanju pozadine, ulazni signal je vrijednost svakog piksela u vremenu  $I_t$ , iz kojeg se proračunava težinski estimator pozadine  $M_t$ . Zatim se računa apsolutna razlika između vrijednosti piksela ulaznog signala  $I_t$  i težinskog estimatora pozadine  $M_t$ . Na temelju toga se izračunava estimator varijance  $V_t$ . Parametar za estimaciju varijance je  $N$  čija je tipična vrijednost između 1 i 4.

---

#### Algorithm 1 Osnovni $\Sigma\Delta$ algoritam

---

```

1: for svaki piksel do
2:   if  $M_{t-1}(x) < I_t(x)$  then
3:      $M_t(x) = M_{t-1}(x) + 1$ 
4:   else if  $M_{t-1}(x) > I_t(x)$  then
5:      $M_t(x) = M_{t-1}(x) - 1$ 
6:   else
7:      $M_t(x) = M_{t-1}(x)$ 
8: for svaki piksel do
9:    $O_t(x) = |M_t(x) - I_t(x)|$ 
10: for svaki piksel do
11:   if  $V_{t-1}(x) < N * O_t(x)$  then
12:      $V_t(x) = V_{t-1}(x) + 1$ 
13:   else if  $V_{t-1}(x) > N * O_t(x)$  then
14:      $V_t(x) = V_{t-1}(x) - 1$ 
15:   else
16:      $V_t(x) = V_{t-1}(x)$ 
17: for svaki piksel do
18:   if  $O_t(x) < V_t(x)$  then
19:      $E_t(x) = 0$ 
20:   else
21:      $E_t(x) = 1$ 

```

---

### 3.4. $\Sigma\Delta$ algoritam s uvjetnim inkrementom

U osnovnoj inačici algoritma nisu se koristila statistička svojstva modela. [1] predlaže da se ona mogu koristiti i da impliciraju sljedeća svojstva. Frekvencija osvježavanja pozadine proporcionalna je varijanci razdiobe, a varijanca se osvježava s nekim konstantnim periodom  $Tv$ . Prilagođeni algoritam vrlo je sličan početnom algoritmu, a [2] predlaže da se algoritam izvede ovako:

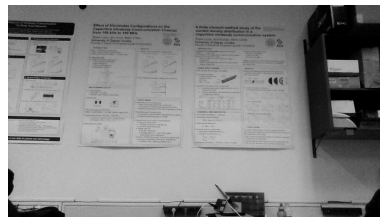
---

**Algorithm 2** Zipf  $\Sigma\Delta$  algoritam

---

```
1: for svaki trenutak  $t$  do
2:    $rank = t \% 2^m$ ;
3:   do  $\{pow2 = 2 * pow2\}$ 
4:   while  $((rank \% 2 == 0) \text{ and } (pow2 < 2^m))$ 
5:   if  $V_{t-1}(x) > 2^m / pow2$  then
6:     estimiraj pozadinu za svaki piksel
7:    $O_t(x) = |M_t(x) - I_t(x)|$ 
8:   if  $t \% Tv == 0$  then
9:     estimiraj varijancu za svaki piksel
```

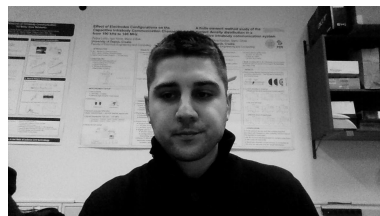
---



Slika 5.: Slika pozadine

### 3.5. Usporedba $\Sigma\Delta$ algoritma s GMM algoritmom

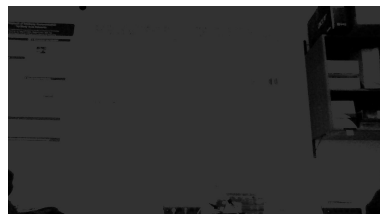
GMM algoritam zahtijeva računski puno više posla za svaki piksel (usporedbe u kontekstu  $K$  distribucija) dok se kod  $\Sigma\Delta$  algoritma radi o uvjetnom jediničnom inkrementu. Memorijski je zahtjevnije jer je za svaki piksel moraju pamtiti  $K$  srednjih vrijednosti, varijanci i težina i najčešće je potrebno raditi operacije nepogodne za *fixed-point* aritmetiku. Kod  $\Sigma\Delta$  algoritma sve operacije su jednostavnije (usporedba, inkrement, apsolutna razlika) i manja je potrošnja memorije (pamti se matrica estimatora pozadine i varijance).



Slika 6.: Slika pomaka

### 3.6. Analiza implementacije $\Sigma\Delta$ algoritma

Osnovni  $\Sigma\Delta$  algoritam neće dati dobre rezultate u svim situacijama. Osim toga ovisno o periodu osvježavanja težinskog estimatora  $M_t$  dobivaju se različiti rezultati. Uz pomno odabranu konstantu  $N$  za varijancu može se ukloniti velik dio šuma. Ipak za bolje rezultate poželjno je koristiti i neko od morfoloških filtriranja za dodatno uklanjanje šuma. U nastavku su prikazane rezultati evaluacije algoritma u pokaznoj situaciji:

Slika 7.: Estimacija srednje vrijednosti  $M_t$  - 50 koraka algoritma

## 4. Sustav za videonadzor

Uz pomoć razvijenog algoritma napravljen je jednostavan sustav za videonadzor temeljen na *BeagleBone Black* platformi. Idejna shema sustava prikazana je na slici 4..

*BeagleBone Black* platforma prikuplja podatke sa kamere. Podaci se zatim obrađuju  $\Sigma\Delta$  algoritmom, a na temelju čeka se daljnom analizom bi-

Slika 8.: Maska nakon jednog koraka algoritma  $M_t$

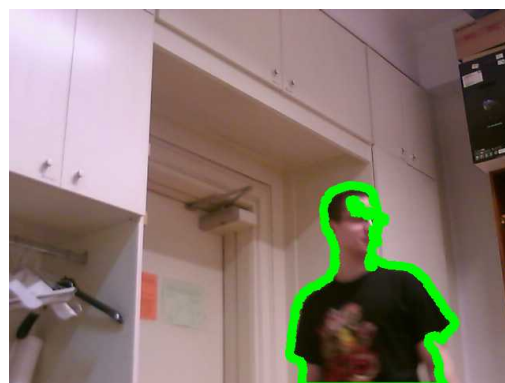
narne maske koja se dobije na izlazu  $\Sigma\Delta$  algoritma utvrđuje da li je došlo do pomaka ili ne. Korisnik je preko interaktivnog sučelja dostupnog u lokalnoj mreži obaviješten o eventualnom detektiranom pomaku. Nadalje korisniku je preko korisničkog sučelja dostupan pregled slika svih detektiranih pomaka, a svaki detektirani pomak spremljen je u bazu podataka s pripadajućim vremenom i datumom detekcije. Korisničko sučelje ima zvučni i vizualni alarm u slučaju detektiranog pomaka. Putem korisničkog sučelja korisnik može podešavati parametre algoritma za detekciju pomaka.

#### 4.1. Analiza programske podrške

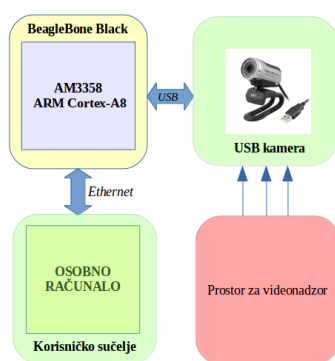
Programska podrška sastoji se od 2 procesa. Jedan od procesa zadužen je za akviziciju i obradu slike prema zadanim parametrima. Drugi proces je poslužitelj koji korisniku omogućava promjenu parametara algoritama za obradu slike. Poslužitelj generira sučelje i sadržava stanje sustava za detekciju pomaka. Sustav može biti u jednom od 3 stanja. Prvo stanje je inicijalno u koje sustav dolazi nakon pokretanja i traje određeno vrijeme koje je potrebno sustavu da procjeni pozadinu i započne s normalnim radom. Iz početnog stanja sustava sustav prelazi u drugo stanje u kojem nema pomaka. U trećem stanju sustav je detektirao pomak. Ovisno o stanju sustava, sustav sprema podatke o okolini. Konkretno, spremanje slika u memoriju moguće je samo iz stanja u kojem je pomak detektiran. Poslužitelj je stoga odgovoran i za spremanje slika u bazu podataka. Između poslužitelja i pro-

cesa za obradu slike postoji međuprocena komunikacija putem *Unix domain socket* protokola. Na taj način je prije svega omogućeno prosljeđivanje slike iz procesa za obradu slike do krajnjeg korisnika (engl. *stream*). Osim prijenosa slike prenosi se i informacija o detekciji pomaka, odnosno je li pomak detektiran ili nije. Na isti način poslužitelj ovisno o parametrima koje mijenja korisnik na sučelju šalje parametre procesu za obradu slike koji te iste parametre prilagođava.

Za razvoj programske podrške za obradu slike korištena je programska biblioteka *OpenCV 3.1*. Za razvoj programske podrške za server korišten je *Node.js framework*.



Slika 10.: Prikaz konačnog rezultata obrade dostupnog na web sučelju



Slika 9.: Razvijeni sustav za videonadzor

## 5. Zaključak

Detekcija pomaka i analiza pozadine slike jedna je od ključnih tema u računalnom vidu. Postoji mnogo metoda, a samo su neke razrađene u ovom radu. Pokazano je kako su operacije nad slikama procesorski i memorijski zahtjevne i zbog toga se traže novi načini i metode obrade. Jedan od njih je  $\Sigma\Delta$  pristup estimaciji pozadine. Najvažnije prednosti  $\Sigma\Delta$  pristupa u odnosu na ostale pristupe je jednostavnost računskih operacija. Potrebne su svega tri operacije za provedbu algoritma: apsolutna razlika, usporedba i inkrement odnosno dekrement. Jednostavnost operacija omogućava i sklopovsku realizaciju algoritma koja može dodatno ubrzati proces estimacije. Pošto je estimacija pozadine početni korak u daljnoj analizi slike kao

što je primjerice predikcija kretanja bitno je da se izvršava što brže radi zadovoljenja zahtjeva za obradu u realnom vremenu. Stoga će slični modeli i principi u budućnosti biti sve aktualniji.

## Literatura

- [1] Lacassagne, L. ; IEF/AXIS, Univ. Paris Sud, Paris, France ; Manzanera, A. ; Dupret, A. Motion detection: Fast and robust algorithms for embedded systems
- [2] A. Manzanera ENSTA - Elec. and Comp. Sc. lab, 32 Bd Victor, 75015 Paris, France  $\Sigma - \Delta$  background subtraction and the Zipf law
- [3] Z. Zivkovic, Improved Adaptive Gaussian Mixture Model for Background Subtraction
- [4] B. Tamersoy, Background Subtraction <http://www.cs.utexas.edu>