

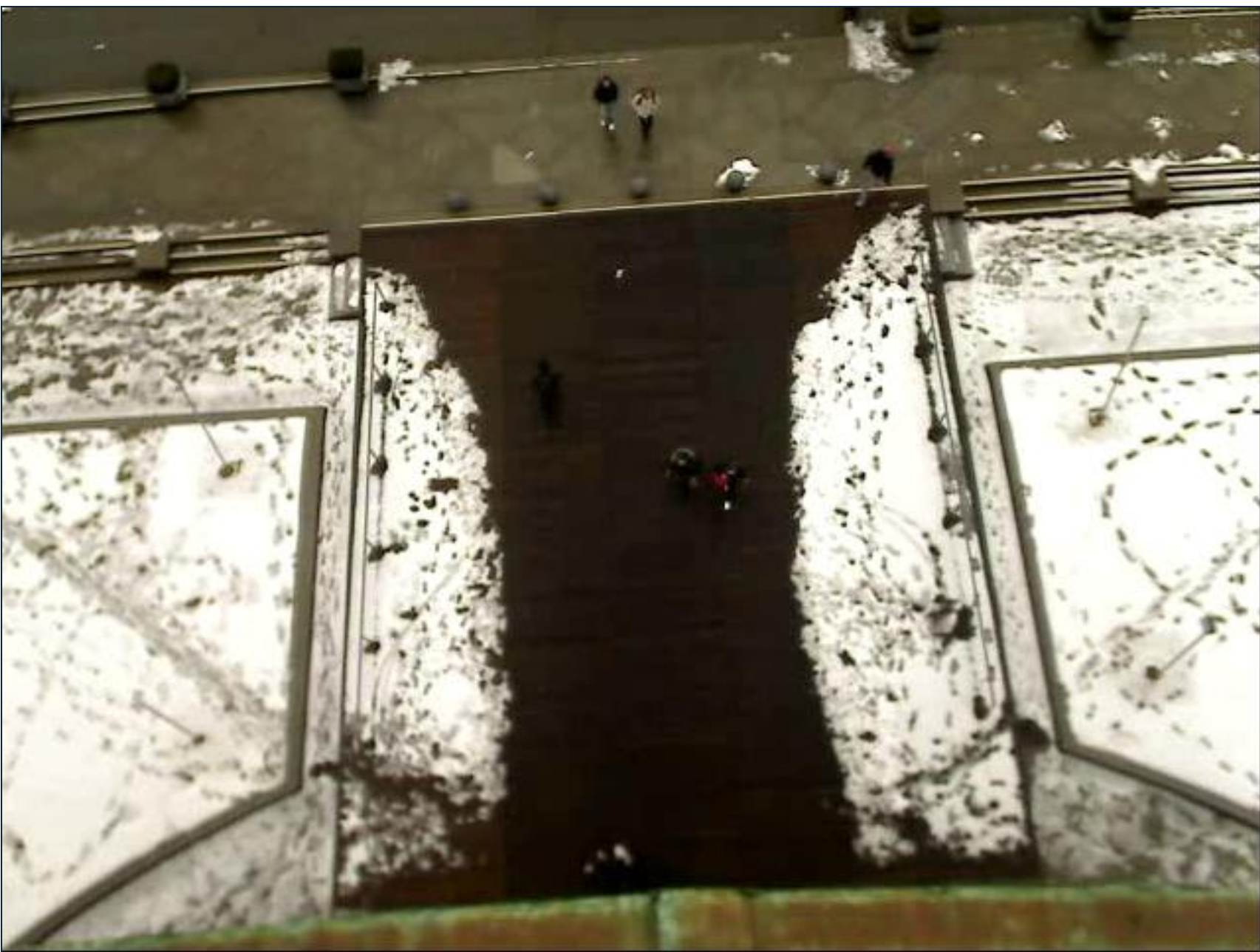


# Prebrojavanje pešaka sa snimaka nadzorne kamere

Jovan Laković – SW51/2015

## Problem

Potrebno je detektovati i prebrojati svakog pešaka koji se bar jedanput našao na platou koji je središnji deo snimka nadzorne kamere. *Dataset* koji je korišćen su snimci nadzorne kamere iznad platoa – snimci obuhvataju i deo kojim pešaci ne prolaze (Slika 1). Ukupno ima 10 različitih snimaka. Cilj je da se ostvari što veći procenat tačnosti algoritma.



Slika 1. Snimak sa fiksne kamere

## Algoritam

Algoritam možemo svesti na 2 zadatka:

1. Izdvajanje platoa i njegovih granica (Slika 2)
2. Izdvajanje segmenta platoa u kom će se vršiti detekcija pešaka (Slika 3)
3. Detekcija i prebrojavanje pešaka (Slika 4)

U implementaciji rešenja korišćena je **openCV** biblioteka za obrađivanje slike i video zapisa.

U prvom delu, potrebno je izdvojiti plato. Ovo se radi samo za prvi frejm snimka, jer je kamera fiksna. Frejm se konvertuje u *gray scale*, a zatim se prevodi u binarnu sliku upotrebom binarnog *threshold*-a. Primenjuju se morfološke operacije nad dobijenom binarnom slikom kako bi se otklonili šumovi i lakše detektovali objekti. Kada je plato detektovan, pronalaze se njegove granične tačke – uglovi.

Izdvaža se određeni segment platoa u kom će se vršiti detekcija pešaka.

Svaki  $n$ -ti frejm se obrađuje – posmatra se samo izdvojeni segment (odsečen je i deo platoa na kom je sneg – zbog agresivnog šuma). Segment se konvertuje u *gray scale*, prevodi se u binarnu sliku upotrebom adaptivnog *threshold*-a. Zatim se primenjuju morfološke operacije nad binarnom slikom radi otklanjanja šumova i lakše detekcije pešaka. Prebrojava se svaki detektovani pešak, odnosno objekti čiji je radius zanemarljivo mali – posmatraju se kao šum i zanemaruju, kao i objekti čiji je radius prevelik da bi predstavljali pešaka.

Svi parametri u metodama i operacijama korišćenim za obradu slike, kao i  $n$  u trećem koraku algoritma, utvrđeni su eksperimentalno kako bi doveli do najboljih rezultata.

Korišćene **morfološke** operacije:

*erozija* – minimizacija svih piksela u okolini posmatranog piksela (ukoliko je neki piksel u okolini nula, *output* piksel je nula).

*dilacija* – maksimizacija svih piksela u okolini posmatranog piksela (ukoliko je neki piksel u okolini 1, *output* piksel je 1).

*otvaranje* – erozija pa zatim dilacija (manje destruktivno od same erozije)

*zatvaranje* – dilacija pa zatim erozija



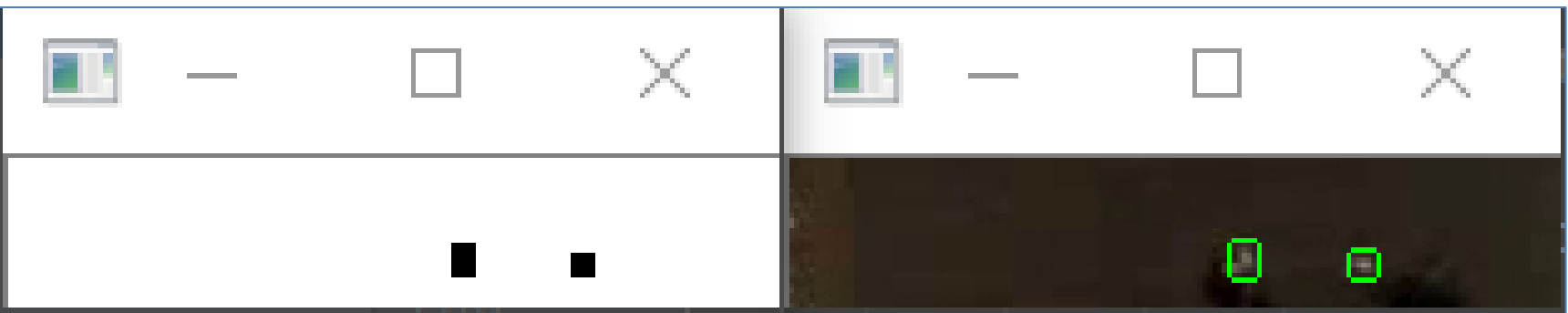
Slika 2. Izdvajanje platoa



Slika 3. Izdvajanje segmenta

## Analiza rezultata

Tačnost algoritma varira od 35% do 83.2% u zavisnosti od toga koji parametri se koriste u metodama poput morfoloških operacija nad slikama. Veliki uticaj takođe ima i parametar  $n$  – ukoliko je  $n < 5$  ili  $n > 20$ , rezultati se znatno pogoršavaju. Za malo  $n$ , isti pešak se može prebrojati 2 puta, dok za veliko  $n$ , neki pešaci mogu biti potpuno zanemareni, i time smanjiti ukupan broj detektovanih pešaka. Takođe, najbolji rezultati dobijaju se primenom Gausove metode kao algoritma za adaptivni *thresholding*, prilikom detekcije pešaka.



Slika 4. Detekcija pešaka – levo binarna slika (primenjen adaptivni thresholding); desno su detektovani pešaci

## Zaključak

Eksperimentalnim testiranjem utvrđeni su parametri metoda koji daju tačnost od preko 80%, ali ovaj problem mogao bi se rešiti na nekoliko drugačijih načina. Najveći problem za dobijanje velike tačnosti predstavlja lošiji kvalitet video zapisa – previše šuma kojeg je u nekim situacijama gotovo nemoguće potpuno ukloniti.

Bolja opcija bi bila praćenje objekata, odnosno pešaka u slučaju konkretnog problema. Ovakvom implementacijom obrađivao bi se ceo plato, a ne samo određeni segment. Ideja je u svakom trenutku obrađivati  $i$ -ti i  $i+1$ -ti frejm. Posmatra se razlika ova dva frejma, odnosno *background subtraction*. U ovakvom slučaju ne bi postojalo preskakanje frejmova, ili bi to preskakanje moralo biti minimalno (1 ili maksimalno 2 frejma)

## Kontakt

Jovan Laković  
FTN  
Email: lakovic.j@live.com