Primljen: 13.9.2017. Stručni rad
Prihvaćen: 11.10.2017. UDK: 681.52./53

Vizijski sustav za praćenje pozicije čovjeka pomoću algoritma oduzimanja pozadine

Vision System for Human Position Tracking by Means of Background Subtraction Algorithm

[1]Ante Javor, [2]Ivan Sekovanić, [3]Zoran Vrhovski
[1]student Visoke tehničke škole u Bjelovaru
[2]Visoka tehnička škola u Bjelovaru, Trg Eugena Kvaternika 4, 43000 Bjelovar
e-mail: [1]ajavor@vtsbj.hr, [2]isekovanic@vtsbj.hr, [3]zvrhovski@vtsbj.hr

Sažetak: *U ovom radu opisana je izrada vizijskog sustava za praćenje pozicije čovjeka pomoću algoritma oduzimanja pozadine. Za dohvat slike korištena je USB kamera rezolucije 640 x 480 piksela. Dohvaćena slika s kamere prolazi algoritmom oduzimanja pozadine, filtrira se i prikazuje na korisničkom sučelju. Pronađeni objekt (čovjek) ocrtava se pomoću zatvorene konture, a pozicija njegovog centra mase označena je zelenom kružnicom. Vizijski sustav opisan u ovom radu u realnom vremenu pronalazi poziciju pomičnog čovjeka sa slike. Slika se u radu obrađuje pomoću biblioteke OpenCV. Biblioteka OpenCV sadrži velik broj funkcija i korisnih klasa koje pomažu u rješavanju problema računalnog vida. Programski kôd vizijskog sustava razvijen je pomoću programskog okruženja Visual Studio i programskog jezika C++.*

Ključne riječi: *OpenCV, obrada slike, praćenje pozicije čovjeka, oduzimanje pozadine*

Abstract: *This paper describes development of the vision system for human position tracking by means of background subtraction algorithm. USB camera with 640 x 480 pixel resolution is used for capturing images. The captured image passes through the background subtraction algorithm, filters and then is displayed on the user interface. The detected object (human) is marked with closed contour while his centar of mass is marked with green circle. Vision system described in this work is able to detect a position of moving human body in*

real-time from an image feed. Images are processed by OpenCV program library. The OpenCV library contains a lot of functions and useful classes that helps to solve computer vision problems. Source code for vision system is developed using the Visual Sudio framework with C++ programming language.

Key words: *OpenCV, image processing, human body position tracking, background subtraction*

1. Uvod

Današnje kamere mogu snimati videozapis u 4K rezoluciji. Cijena kamera je sve niža, dok je njihova kvaliteta sve veća. Istovremeno, raste procesorska moć računala što omogućuje obradu velike količine podataka. Takav napredak, dostupnost kamera i računala omogućuje razvoj dijela računarstva koji se bavi računalnim vidom (engl. *computer vision*). Trend korištenja računalnog vida u svim granama znanosti u osjetnom je porastu. Sukladno tome, razvijaju se i algoritmi računalnog vida koji su tijekom povijesti, zbog slabije kvalitete kamera, računalnih resursa i kompleksnosti vizijskog sustava čekali smanjenje tehnoloških barijera. Vrlo često se računalni vid koristi u robotici za detekciju linija, krivulja i ostalih objekata (Vrhovski i Herčeki, 2011; Kos i sur., 2015). Među algoritme računalnog vida ubrajaju se i algoritmi za praćenje čovjeka.

Praćenje čovjeka i njegovih navika uvijek je bila zanimljiva tema znanstvenicima koji se bave humanističkim i društvenim znanostima. Razvojem računalnog vida praćenje čovjeka postaje sve lakše. Korištenjem računalnog vida mogu se pratiti pozicija tijela, gestikulacije, lice čovjeka pa čak i osjećaji koje čovjek prenosi svojim tijelom. Sve to otvara veliki prostor za inovacije u medicini, robotici, sportu, društvenim aktivnostima, komunikaciji s računalima i slično. Algoritmi za praćenje čovjeka zbog svoje kompleksnosti razvijaju se već dugi niz godina što ih čini jednim od najduže razvijanih algoritama u segmentu računalnog vida. Razlog tome je što su svi ljudi fizičkih različiti, različito se odijevaju i ponašaju, dok je tehnički vrlo zahtjevno stvoriti 3D sliku iz 2D izvora (kamere).

Razvoj algoritama za praćenje čovjeka (Sood i sur. 2013; KaewTraKulPong i Bowden 2001; Zivkovic, 2004) je u uzlaznoj putanji, što predstavlja put prema rješavanju problema praćenja čovjeka. Univerzalno rješenje koje rješava problem praćenja čovjeka još uvijek ne postoji. Stoga se razvijaju usko specijalizirani vizijski sustavi koji djelomično rješavaju problem praćenja pozicije čovjeka. Pri praćenju čovjeka postoje dva osnovna pristupa:

praćenja s markerima i bez markera. U ovom radu opisan je vizijski sustav čija je svrha pratiti poziciju čovjeka pomoću algoritma oduzimanja pozadine (engl. *background subtraction*). Ovaj algoritam za praćenje pozicije čovjeka ne koristi markere.

Rad je strukturiran na sljedeći način. U drugom poglavlju opisana je korištena programska podrška u razvoju vizijskog sustava za praćenje pozicije čovjeka. Obrada slike u vizijskom sustavu praćenja pozicije čovjeka pomoću algoritma oduzimanja pozadine opisana je u trećem poglavlju. U četvrtom poglavlju prikazani su rezultati rada vizijskog sustava. Kratki zaključak dan je u petom poglavlju.

2. Korištena programska podrška u razvoju vizijskog sustava za praćenje pozicije čovjeka

Kod izrade ovog rada korišteni su sljedeći programski alati: Visual Studio Community 2015, Cmake i GitHub. Za potrebe obrade slike korištena je biblioteka OpenCV (engl. *open source computer vision library*).

2.1. Visual Studio Community 2015

Visual Studio Community 2015 besplatni je proizvod tvrtke Microsoft i razvojno okruženje namijenjeno studentima, manjim grupama programera i zajednici otvorenog kôda (engl. *open source*). Pomoću razvojnog okruženja Visual Studio Community 2015 moguće je razviti aplikacije za: Windows operativne sustave, mobilne operativne sustave, web i rad u oblaku. Osim aplikacija moguće je razvijati i računalne igre, alate za paket Microsoft Office, baze podataka, ekstenzije za razvojna okruženja i slično.

2.2. Cmake

Cmake je aplikacija koja služi za upravljanje procesom stvaranja softvera. Ova aplikacija je besplatna i otvorenog je kôda. Za stvaranje softvera zahtjeva samo C++ prevoditelj. Cmake je korišten u svrhu stvaranja izvršnog kôda biblioteke OpenCV kako bi se ista mogla koristiti za stvaranje novih projekata. Više informacija o aplikaciji Cmake može se pronaći na web stranici https://cmake.org/.

2.3. OpenCV

OpenCV je računalna biblioteka koja je dostupna na web stranici http://opencv.org. Izdao ju je 1999. godine Intel čiji je cilj bio razvoj računalnoga vida i umjetne inteligencije.

Biblioteka pruža temelje računalnoga vida i dostupna je svima. Napisana je u programskim jezicima C i C++ te podržava operativne sustave Linux, Windows i Mac OS X. Trenutno su u razvoju sučelja za programske jezike: Pyton, Java i MATLAB. U razvoju je i podrška za Android i iOS mobilne operativne sustave (Kaehler i Bradski, 2016). Biblioteka OpenCV 3.2.0. podijeljena je na sljedeće module:

- *core* - modul koji sadrži sve osnove tipove podataka, njihove operatore i osnovne funkcije. Ovaj modul proteže se svim drugim modulima u biblioteci.
- *imgproc* - modul u kojem se nalaze funkcije za transformaciju slike, filtre i pretvorbe slike.
- *imgcodecs* - modul koji sadrži funkcije za čitanje i pohranjivanje slika.
- *videoio* - modul koji sadrži klase i funkcije za čitanje i pohranjivanje videa.
- *highgui* - modul koji sadrži funkcije za korisničko sučelje, prikaz slike i videa.
- *video* - modul koji se koristi za analizu i obradu videa. U njemu su sadržane klase za oduzimanje pozadine *(*engl. *background substraction*) koja se koristi u sustavu za praćenje pozicije čovjeka.
- *calib3d* - modulu u kojem se nalaze algoritmi za kalibraciju kamera i 3D rekonstrukciju.
- *features2d* - modul koji obuhvaća algoritme za detektiranje, obradu, označavanje i pretraživanje ključnih točaka sa slike.
- *objdetect* - modulu u kojem se nalaze algoritmi za detektiranje objekata kao što su ljudi, lica, registarske oznake i slično. Sadrži i algoritme koje je moguće učiti da prepoznaju neke zadane objekte.
- *ml* - modul koji sadrži velik broj algoritama za strojno učenje koji su prilagođeni da rade s tipovima podataka i objekata iz biblioteke OpenCV.
- *flann* – modulu u kojem se nalaze metode za pretragu i obradu višedimenzionalnih podataka na vrlo učinkovit način. Većina tih metoda koristi se u drugim funkcijama biblioteke OpenCV te se rijetko koristi direktno.
- *photo* - modul koji sadrži alate za računalnu fotografiju, kao što je uklanjanje šuma, obrada detalja i slično.
- *stitching* - modul koji se koristi za spajanje, kalibraciju i obradu višestrukih slika u jednu.

2.4. GitHub

GitHub je web aplikacija za kontrolu razvoja softvera koju uglavnom koriste programeri za rad na izvornom kôdu i dokumentaciji. Aplikaciji GitHub moguće je pristupiti kao privatna

osoba ili kao organizacija, a količina dopuštenih repozitorija, njihova dostupnost i ostalo ovisit će o tome koristi li se besplatni ili plaćeni pristup platformi. OpenCV se razvija na platformi GitHub gdje cijela zajednica programera i razvojnih inženjera može pristupiti izvornom kôdu biblioteke (engl. *source code*) i slati zahtjeve za izmjene i dopune. U isto vrijeme postoje programeri koji te dopune i izmjene odobravaju, odnosno odbijaju. Više informacija o web aplikaciji GitHub može se pronaći na web stranici https://github.com/.

3. Obrada slike u vizijskom sustavu praćenja pozicije čovjeka pomoću algoritma oduzimanja pozadine

Metoda oduzimanja pozadine (engl. *background subtraction*) popularna je u mnogim aplikacijama računalnog vida. Biblioteka OpenCV ima dva algoritma oduzimanja pozadine. Metoda oduzimanja pozadine temelji se na usporedbi prošle slike s trenutačnom. Svaka promjena na slici je novi objekt na sceni, dok nepromjenjivi dio slike predstavlja pozadinu. Promjena osvjetljenja tijekom dana, vjetar koji giba grane, auti koji se kreću u pozadini i slično predstavljaju probleme koji utječu na kvalitetu obrade slike metodom oduzimanja pozadine. Jedan dio navedenih problema moguće je riješiti pomoću filtara za uklanjanje šuma na slici (Kaehler i Bradski, 2016). Praćenje pozicije čovjeka pomoću algoritma oduzimanja je metoda koja ne koristi markere. Pretpostavka ove metode je da mirujuća kamera uvijek snima istu pozadinu. Kada se na sceni pojavi čovjek, algoritam će ga izdvojiti. Biblioteka OpenCV posjeduje dva algoritma za oduzimanje pozadine (BackgroundSubstractorKNN i BacgroundSubstractorMOG2) koji su detaljno objašnjeni u literaturama (KaewTraKulPong i Bowden, 2001; Zivkovic, 2004). Algoritam BackgroundSubstractorKNN (KNN) bazira se na metodi strojnog učenja i klasifikaciji najbližeg susjeda dok se algoritam BacgroundSubstractorMOG2 (MOG2) bazira na Gaussovom modelu miješanja. U narednim poglavljima bit će parcijalno prikazani ključni koraci vizijskog sustava, dok se programski kôd u cijelosti može pronaći na web aplikaciji GitHub (https://github.com/antejavor/opencv-examples).

3.1. Dohvaćanje slike s USB kamere

Slika se u ovom radu dohvaća USB web kamerom rezolucije 640 x 480 piksela. Biblioteka OpenCV posjeduje klasu VideoCapture koja nudi rješenja za čitanje, pohranjivanje i kontrolu videa. Programski kôd za dohvaćanje slike s USB web kamere prikazan je na slici 1. U

programskom kôdu stvara se objekt tipa VideoCapture. Ovaj objekt koristi se za provjeru dostupnosti kamere te za čitanje slike s kamere. Čitanje slike s kamere ostvareno je pomoću funkcije .read(). Nakon uspješne provjere čitanja slike, slijedi prikaz slike pomoću funkcije imshow() i čekanje korisničkog unosa s tipkovnice pomoću funkcije waitKey().

Slika 1. *Programski kôd za dohvaćanje slike s USB web kamere*

```
#include <opencv2\opencv2.hpp>
#include <iostream>
int main()
{
  cv::Mat slika;
  /* Otvaranje video ulaza odnosno kamere s ID-em 0. */
  cv::VideoCapture video{ 0 };
  cv::namedWindow("Video", cv::WINDOW_AUTOSIZE);

  if (!video.isOpened())
  {
        std::cerr << "Nemogucnost otvaranja video izvora." <<
std::endl;
        return -1;
  }
  /* Petlja za čitanje i prikazivanje slika. */
  while (true)
  {
        if (!video.read(slika))
        {
              std::cerr << "Problem sa očitavanjem slika" <<
std::endl;
              break;
        }
        cv::imshow("Video", slika);
        if(cv::waitKey(1) == 27)
              break;
  }
  return 0;
}
```

Izvor: autor

3.2. Algoritmi oduzimanja pozadine, uklanjanje sjene i šuma sa slike

Programski kod prikazan na slici 2 prikazuje primjenu algoritama oduzimanja pozadine na dohvaćenu sliku s kamere. Nakon deklaracije varijabli tipa cv::Mat koje se koriste u algoritmima, stvaraju se dva pokazivača na objekte tipa BackgroundSubstractor. Pokazivačima se nakon toga dodjeljuje vrijednost. Pomoću funkcija

createBackgroundSubstractor() stvara se objekt s određenim parametrima. Kod KNN i MOG2 metoda oduzimanja pozadine, prvi parametar predstavlja broj slika koji je potreban da algoritam potpuno osvježi pozadinu. Drugi parametar predstavlja granicu u kojoj piksel pripada ili ne pripada objektu. Obzirom na to da obje funkcije mogu pronaći sjene, treći parametar tipa bool određuje hoće li algoritam te sjene označiti ili ne.

Slika 2. *Programski kôd algoritma oduzimanja pozadine*

```
cv::Mat frame;
cv::Mat fgMOG2;
cv::Mat fgKNN;

cv::Ptr<cv::BackgroundSubtractor> MOG2;
cv::Ptr< cv::BackgroundSubtractor> KNN;

MOG2 = cv::createBackgroundSubtractorMOG2(2000, 16,true);
KNN = cv::createBackgroundSubtractorKNN(2000,400, true);

MOG2->apply(frame, fgMOG2);
KNN->apply(frame, fgKNN);
```

Izvor: autor

Slika koja je dobivena algoritmima oduzimanja pozadine je slika sivih nijansi. Vrijednost pojedinih piksela na sivoj slici kreće se od 0 (crna boja) do 255 (bijela boja). Nakon što je algoritmu naznačeno da označi sjene, on će piksele koji predstavljaju sjenu postaviti na vrijednost 127 (nijansa sive boje). Budući da je sjenu i ostale smetnje potrebno ukloniti, primjenjuje se funkcija za binarno ograničenje threshold() (Vrhovski i Herčeki, 2011) s graničnom vrijednosti 128 (pogledati programski kôd na slici 3). Funkcija za binarno ograničenje će sve piksele, čija je vrijednost manja od 128, postaviti u vrijednost 0 (crna boja). Rezultat binarnog ograničenja slike prikazan je na slici 4. Na slici 4 (lijevo) prikazana je slika s detektiranom sjenom (pikseli sive boje vrijednosti 127). Na slici 4 (desno) prikazana je slika nakon uklanjanja sjene binarnim ograničenjem.

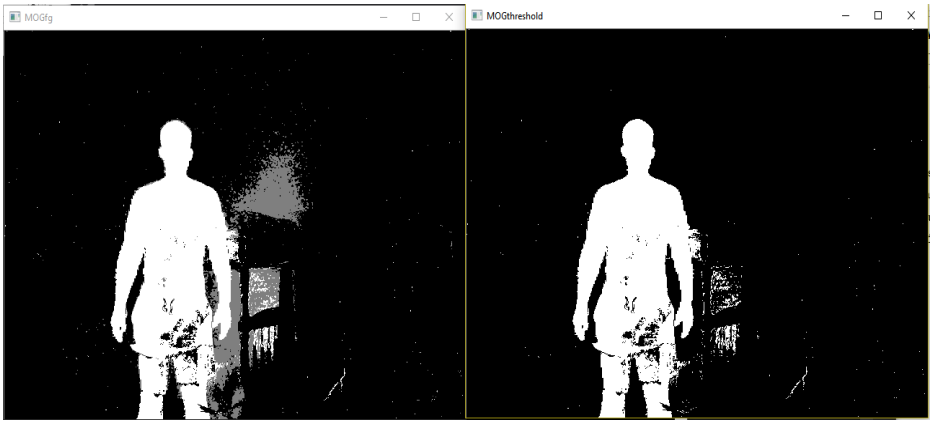
Slika 3. *Programski kôd za uklanjanje sjene i šuma sa slike*

```
/* Uklanjanje sjene sa slike */
cv::threshold(fgMOG2, thresMOG2, 128, 255, cv::THRESH_BINARY);
cv::threshold(fgKNN, thresKNN, 128, 255, cv::THRESH_BINARY);

/* filtriranje slike */
```

```
cv::Mat elementOpen =
cv::getStructuringElement(cv::MorphShapes::MORPH_RECT,
cv::Size(3,3));

cv::Mat elementDilate =
cv::getStructuringElement(cv::MorphShapes::MORPH_RECT,
cv::Size(5,5));

cv::Mat openMOG2, openKNN;
cv::morphologyEx(thresMOG2, openMOG2, cv::MORPH_OPEN, elementOpen,
cv::Point(-1,-1),1);

cv::morphologyEx(thresKNN, openKNN, cv::MORPH_OPEN, elementOpen,
cv::Point(-1, -1), 1);

cv::Mat dilateMOG2, dilateKNN;
cv::dilate(openMOG2, dilateMOG2,elementDilate);
cv::dilate(openKNN, dilateKNN,elementDilate);
```

Izvor: autor

Slika 4. *Binarno ograničenje slike – slika sa sjenom (lijevo) i slika bez sjene (desno)*



Izvor: autor

Obzirom na to da rezultati postupka oduzimanja pozadine često generiraju šum zbog promjene osvjetljenja, blagih gibanja ili nedovoljnog kontrasta novog objekta, potrebno je dodatno obraditi rezultat, odnosno ukloniti šum određenim filtrima. Prisutnost šuma je prikazana na slici 4 (desno). Programski kôd na slici 4 prikazuje postupke obrade slike koji otklanjaju nastali šum. Za potrebe filtriranja koristi se funkcija morphologyEx(). Na slici 5 prikazana je slika bez filtriranja (lijevo) i filtrirana slika (desno). Filtriranjem slike nestali su bijeli pikseli koji su predstavljali šum.

Slika 5. *Filtriranje slike – slika bez filtriranja (lijevo) i filtrirana slika (desno)*

Izvor: autor

3.3. Izdvajanje konture i izračun centra mase čovjeka

Nakon obrađenih rezultata algoritma oduzimanja pozadine, potrebno je pretražiti konture na slici kako bi se lokalizirao i detektirao objekt (čovjek). Na slici 6 prikazan je programski kôd za pretraživanje konture, izdvajanje najveće konture, izračun centra mase čovjeka, ocrtavanje konture i označavanje centra mase čovjeka zelenom kružnicom.

Slika 6. *Programski kôd za izdvajanje i ocrtavanje kontura*

```
std::vector<std::vector<cv::Point>> conturs;
cv::findContours(dilateKNN, conturs, cv::RETR_EXTERNAL,
cv::CHAIN_APPROX_NONE);

int largestArea = 0;
int conturIndex = 0;
  for (int i = 0; i < conturs.size(); i++)
  {
        double area = cv::contourArea(conturs[i]);
        if (area > largestArea)
        {
              largestArea = area;
              conturIndex = i;
        }
  }
if (conturs.size() > 0)
  {
        cv::Moments moments = cv::moments(conturs[conturIndex],
true);
        cv::Point2d centerOfMass((moments.m10 / moments.m00 ),
        (moments.m01 / moments.m00));

        cv::drawContours(frame, conturs, conturIndex,
```

```
        cv::Scalar{ 200,0,0 }, 3, 8);
        cv::circle(frame, centerOfMass, 10, cv::Scalar{ 0,200,0
}, 3);
        }
```

Izvor: autor

Programski kôd počinje funkcijom findConturs() koja u ovom slučaju izdvaja samo vanjske konture i pohranjuje ih u običan niz. Nakon toga se pomoću for-petlje izdvaja najveća kontura koja se pojavila na sceni (kontura koja predstavlja čovjeka). Nakon što je pronađena najveća kontura izračunava se centar mase čovjeka. Pronađen centar mase čovjeka na slici je označen zelenom kružnicom. Centar mase čovjeka predstavlja poziciju čovjeka na slici.

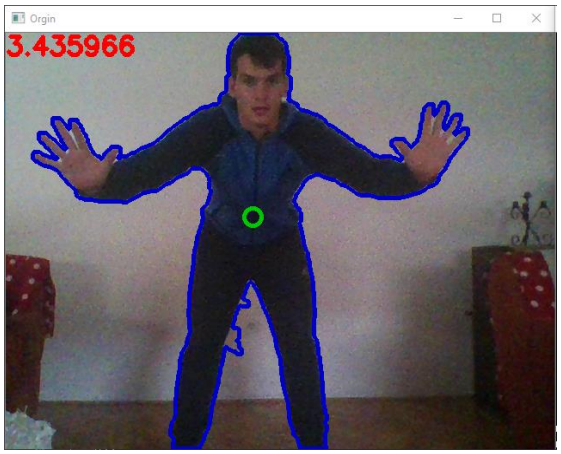
4. Rezultati rada vizijskog sustava

Vizijski sustav praćenja čovjeka, opisan u ovom radu, ima nekoliko zahtjeva da bi funkcionirao na željeni način. Osvjetljenje mora biti kvalitetno, kamera mora biti fiksna i poželjno je da postoji razlika u kontrastu između objekta koji se prati i pozadine s obzirom na to da se radi o metodi uspoređivanja piksela. Mogućnosti ovog vizijskog sustava su izdvajanje najvećeg objekta na sceni i izračuna njegovog centra mase. Rezultati rada ovog vizijskog sustava prikazani su na praćenju pozicije čovjeka na sceni. Na slikama 7, 8 i 9 prikazani su rezultati praćenja pozicije čovjeka algoritmom oduzimanja pozadine. U svim slučajevima na slici, algoritam je uspješno ocrtao konturu čovjeka i pronašao centar mase čovjeka (poziciju čovjeka). Vizijski sustav opisan u ovom radu u realnom vremenu pronalazi poziciju pomičnog čovjeka na slici.

Slika 7. *Rezultati rada algoritma oduzimanja pozadine (1)*

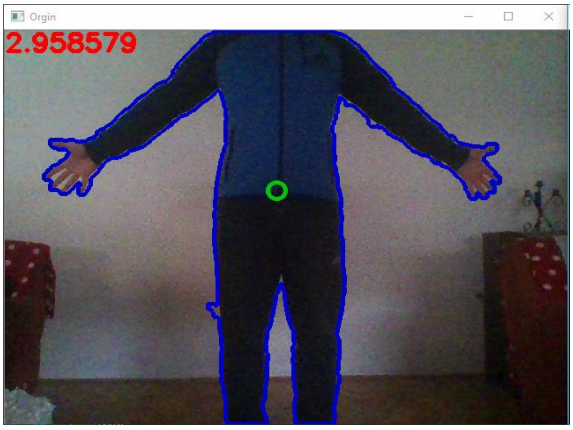
Izvor: autor

Slika 8. *Rezultati rada algoritma oduzimanja pozadine (2)*



Izvor: autor

Slika 9. *Rezultati rada algoritma oduzimanja pozadine (3)*



Izvor: autor

5. Zaključak

Računalni vid kao grana računarstva u stalnom je razvoju. Ključnu ulogu u računalnom vidu ima obrada slike u digitalnom formatu. Ozbiljna obrada slike počinje s kvalitetnom slikom, odnosno ulaznim signalom koji ima dovoljno informacija za obradu što je pokazao i ovaj rad u kojem je korištena web kamera. Zbog loše kvalitete kamere rezultati algoritama nisu savršeni, ali pokazuju potencijal. Nedovoljno kvalitetnu sliku s kamere kompenziraju kvalitetni algoritmi biblioteke OpenCV. Ova biblioteka ima nekoliko stotina algoritama koji su potpuno optimizirani i trenutačno predstavljaju jedno od najboljih rješenja većine problema računalnog vida.

Razvijen vizijski sustav praćenja pozicije čovjeka pomoću algoritma oduzimanja pozadine predstavlja specijalizirani pristup u rješavanju problema. Ovaj vizijski sustav za rad zahtjeva određene uvjete. Uvjeti pod kojima ovaj sustav daje zadovoljavajuće rezultate su dobro osvjetljenje, fiksna kamera te gibanje čovjeka. Dohvaćena slika s kamere podvrgnuta je algoritmu oduzimanja pozadine. Sa slike se uklanjaju sjene i šumovi, a vizijski sustav zatim izdvaja najveći objekt koji se pojavi na sceni. Najveći objekt u ovom radu predstavlja čovjek. Algoritam oduzimanja pozadine, opisan u ovom radu, uspješno ocrtava konturu čovjeka i pronalazi poziciju centra mase čovjeka (poziciju čovjeka). Opisani vizijski sustav u realnom vremenu pronalazi poziciju pomičnog čovjeka na slici. Budući rad na opisanom vizijskom sustavu uključivat će praćenje ključnih točaka čovjeka (zglobova, udova i slično) kao i mjerenje duljine udova.

Literatura

1. Javor, A., GitHub: Background-substraction. https://github.com/antejavor/opencv examples (24.07.2017.).
2. Kaehler, A.; Bradski, G. (2016). „Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library“. O'Reilly Media.
3. KaewTraKulPong P.; Bowden R. (2001). „An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection“, *Video-Based Surveillance Systems*, (ur. Paolo Remagnino, Graeme A. Jones, Nikos Paragios, Carlo S. Regazzoni). Boston. Springer US, str. 135-144.
4. Kos, S.; Vrhovski, Z.; Vidić, D. (2015). „Detection, localization and recognition of objects using LabVIEW“, *Tehnical journal*, vol 9(3), str. 245-250.
5. Sood, M.; Sharma, R.; Dipakkumar, C. (2013). „Motion Human Detection & Tracking Based On Background Subtraction“, *International Journal of Engineering Inventions*, vol 2(6), str. 34-37.
6. Vrhovski, Z.; Herčeki, R. (2011). „Lokalizacija ravne linije u slikovnoj sekvenci“, *Tehnički glasnik*, vol 5(2), str. 5-10.
7. Zivkovic Z. (2004). „Improved adaptive Gaussian mixture model for background subtraction“, *Proceedings of the 17th International Conference on Pattern Recognition*, vol 2(1), str. 28-31.