

Lokalizacija na osnovu markera

U ovom radu ispitana je mogućnost vizualne lokalizacije objekta u ograničenom prostoru na osnovu AruCo markera (Garrido-Jurado et al. 2014; Garrido-Jurado et al. 2016; Romero-Ramirez et al. 2018). Prvi korak lokalizacije odnosi se na detekciju i prepoznavanje markera na slikama napravljenim kamerom na objektu. Po očitavanju markera, koordinate temena markera na slici i koordinate istih u prostoru prosleđuju se solvePnP algoritmu. Kao rezultat algoritma dobijaju se položaj i rotacija kamere u trenutku nastanka slike. Sistem je testiran u prostoru dimenzija 1×1 m na čije je zidove postavljeno 8 markera. Od 288 slika, markeri se uočavaju na njih 218. Ukupno je uočeno 336 markera, od čega je algoritam za detekciju markera uspešno detektovao 298. Greška u poziciji i rotaciji definisana je kao euklidsko rastojanje dobijene pozicije i rotacije kamere od referentnih vrednosti istih. Na slikama na kojima se uočava jedan marker srednja greška pozicije iznosi 5 cm, a srednja greška rotacije 4° . Na slikama na kojima se uočavaju dva markera srednja greška pozicije iznosi 3cm, a srednja greška rotacije je 2° , dok na slikama na kojima se nalaze tri markera srednja greška pozicije iznosi 4 cm, a srednja greška rotacije 3° . Dobijeni rezultati verifikuju opisanu metodu i potvrđuju hipotezu da veći broj markera na slici dovodi do preciznije lokalizacije.

Uvod

Kompjuterska vizija predstavlja naučnu disciplinu koja se bavi metodama razumevanja slike putem računara. Zadaci kompjuterske vizije ogledaju se u stvaranju, obradi i analizi slike, odnosno u izvlačenju informacija iz realnog sveta u cilju kreiranja numeričkih ili simboličkih podataka (Beyerer et al. 2016). Jedan od čestih problema kompjuterske vizije predstavlja određivanje položaja objekta u prostoru. Lokalizacija objekta u kompjuterskoj viziji se zasniva na utvrđivanju veza između pozicija tačaka u stvarnosti i njihovih projekcija na slici.

Kompjuterska vizija se može podeliti na geometrijsku kompjutersku viziju i kompjutersku viziju koja se oslanja na prepoznavanje karakteristika

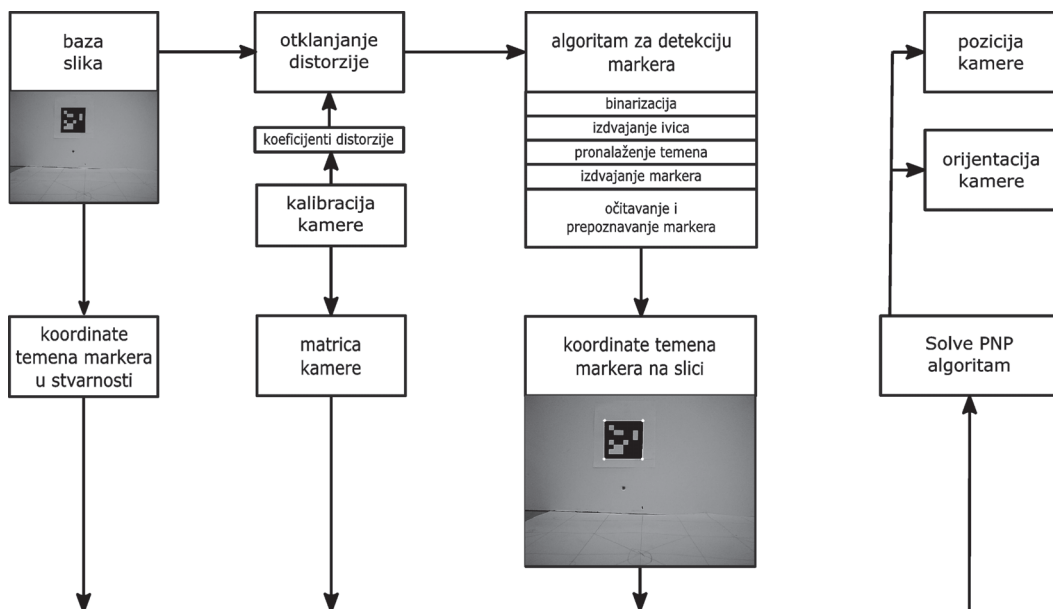
*Dragan Mičić (1998),
Arilje, Kruščica bb,
učenik 4. razreda
Gimnazije „Sveti
Sava” u Požegi*

*Danilo Tonic (1999),
Kraljevo, Dušana
Karapandžića 1k,
učenik 3. razreda
Gimnazije Kraljevo*

MENTORI:

*Andrej Lojdl, Vizlore
Labs Fondacija, Novi
Sad*

*Damjan Dakić,
Microsoft
development centar
Srbija, Beograd*

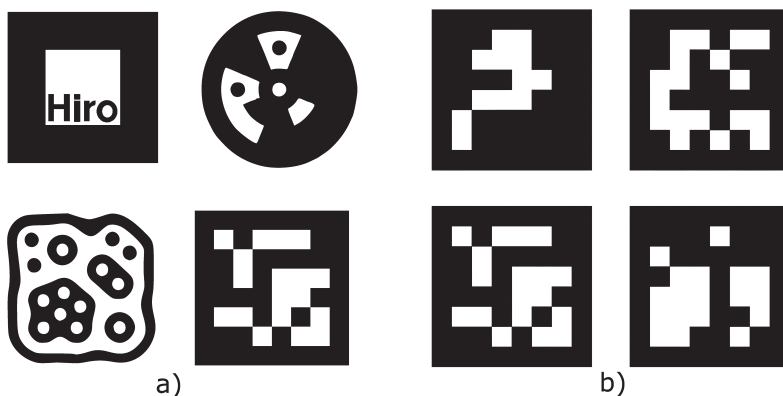


Slika 1. Blok šema sistema za lokalizaciju

Figure 1. Localization block diagram

predmeta (Szeliski 2010). Fokus ovog rada jeste na geometrijskoj kompjuterskoj viziji i korišćenju fiducijalnih markera u rešavanju njenih problema (Garrido-Jurado *et al.* 2014; 2016). Fiducijalni markeri predstavljaju objekte koji svojim izgledom određuju i otkrivaju pozicije tačaka ili njihove udaljenosti u prostoru (slika 2a). Dobra strana korišćenja markera u sistemima lokalizacije jeste mogućnost rada sistema u realnom vremenu i sa velikom preciznošću, dok se mane ogledaju u netoleranciji na okluziju i ostalim pojavama koje dovode do slabe uočljivosti markera (preveliko ili nedovoljno osvetljenje).

Cilj ovog rada jeste validacija i ispitivanje karakteristika metoda lokalizacije korišćenjem Aruco markera (slika 1) (Garrido-Jurado *et al.*



Slika 2. Primeri fiducijalnih markera (a) i Aruco markera (b)

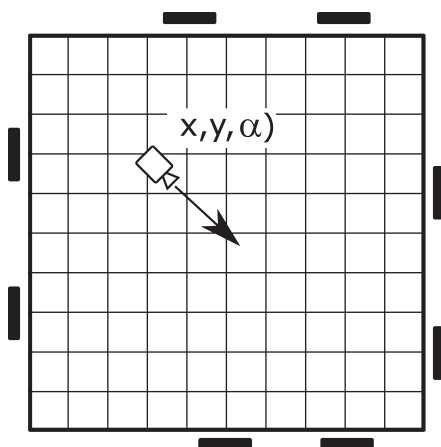
Figure 2. Examples of fiducial markers and Aruco markers (b)

2014). Aruco markeri predstavljaju vrstu fiducijalnih markera. Karakterišu ih kvadratni oblik i crne ivice (slika 2b). Zajedno sa crnim okvirom oni predstavljaju crno-belu matricu dimenzija 8×8 , gde crna odnosno bela polja određuju binarnu matricu, koja potom određuje koordinate temena markera u prostoru. Aruco markeri svojim prostim i pravilnim oblikom doprinose većoj verovatnoći i preciznosti njihove detekcije i prepoznavanja.

Pretpostavlja se da će greška lokalizacije biti veća pri većoj udaljenosti između kamere i markera, i to usled lošije detekcije temena markera, do koje dolazi zbog smanjenja dimenzija projekcije markera na slici. Takođe očekuje se da će veći broj detektovanih markera na istoj slici doprineti tačnijoj lokalizaciji.

Metod

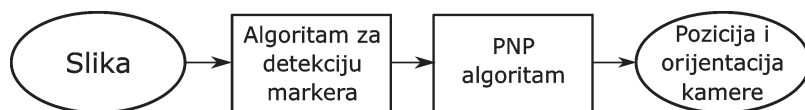
Sistem se realizuje u prostoru na čijim su zidovima postavljeni Aruco markeri (slika 3). Cilj je utvrditi poziciju i orijentaciju kamere u tom prostoru i to na osnovu markera koji se mogu uočiti na slici koja je napravljena kamerom u tom položaju.



Slika 3. Skica sistema

Figure 3. System sketch

Prvi korak u obradi napravljene slike jeste uklanjanje distorzije. Distorzija na slikama predstavlja zakrivljenost perspektive pri uglovima do koje dolazi zbog nesavršenosti sočiva kamere. Distorziju opisuje 5 koeficijenata. Koeficijenti k_1 , k_2 i k_3 opisuju radijalnu, dok p_1 i p_2 opisuju tangencijalnu distorziju. Ove koeficijente dobijamo geometrijskom kalibracijom kamere (Weng *et al.* 1992). Sve slike iz baze najpre prolaze



Slika 4.
Struktura sistema

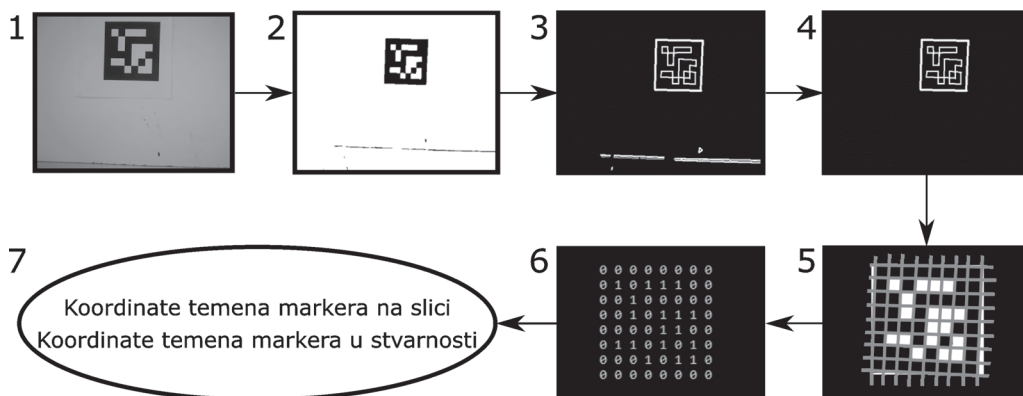
Figure 4.
Structure of the system

kroz algoritam koji na osnovu dobijenih koeficijenata otklanja distorziju koja se na slikama javlja (Docs.opencv.org 2018a).

Sistem za lokalizaciju može se predstaviti u 2 osnovna koraka (slika 4). Prvi korak predstavlja detektovanje i očitavanje markera na slikama. Drugi korak se odnosi na izračunavanje pozicije i orijentacije kamere na osnovu dobijenih podataka.

Algoritam za detekciju markera

Detekcija i očitavanje markera realizuje se nizom koraka opisanih u daljem tekstu (Romero-Ramirez *et al.* 2018; slika 5).



Slika 5. Algoritam za detekciju markera: 1. Original picture; 2. Binarizacija (Local Adaptive Threshold); 3. Izdvajanje kontura (Suzuki i Abe 1987); 4. Aproksimiranje kontura mnogouglovima (Douglas i Peucker 1973) i izdvajanje četvorouglova; 5. Izdvajanje potencijalnih markera i njihovo implicitno ispravljanje; 6. Očitavanje markera; 7. Rezultati algoritma.

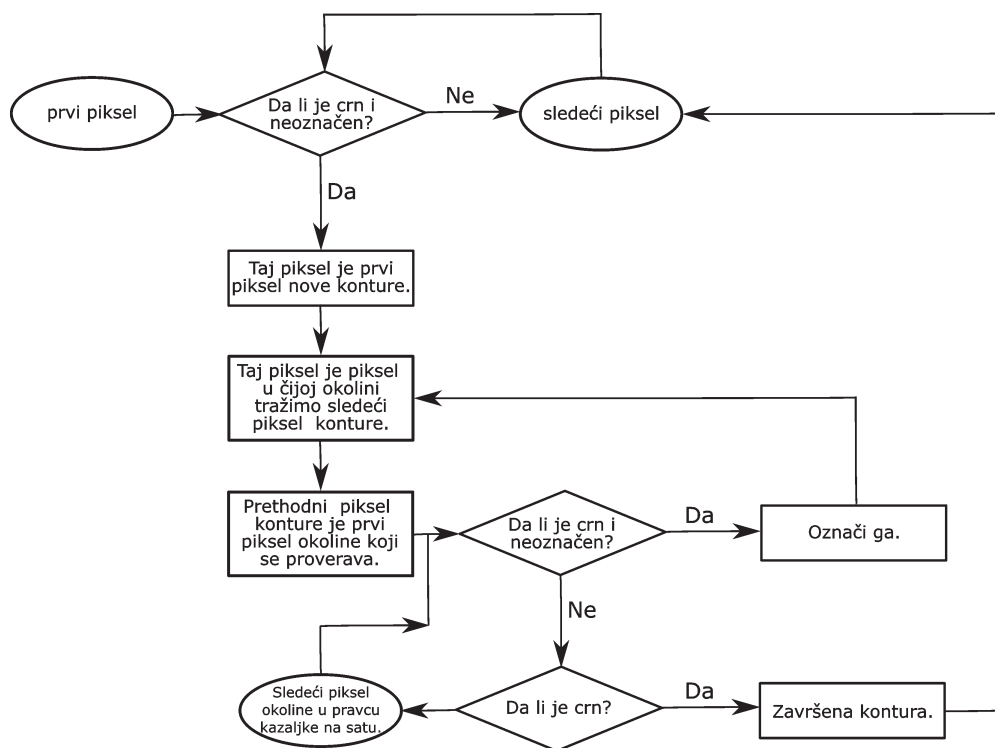
Figure 5. Algorithm for markers detection: 1. Input picture; 2. Binarization (Local Adaptive Threshold); 3. Shape selection (Suzuki and Abe 1987); 4. Approximation of contours with polygons (Douglas and Peucker 1973) and selection of foursquare polygons; 5. Exposing potential markers and their implicit correction; 6. Read markers; 7. Algorithm output.

Binarizacija

Binarizacija slike predstavlja proces kojim se pikselima slike, sa proizvoljno kanala i proizvoljnih vrednosti, dodeljuje vrednost jedinice ili nule. Kriterijumi za takvo dodeljivanje mogu biti unapred zadata vrednost, srednja vrednost svih piksela slike, srednja vrednost piksela u okolini piksela o kojem se odlučuje i drugi. Algoritam koji je korišćen za binarizaciju slike u ovom radu jeste lokalna adaptivna granična vrednost (eng. local adaptive threshold) (Burghardt *et al.* 2007). Prvi korak ovog algoritma je podela slike na pravougaonike unapred zadatih i istih dimenzija. U okviru tih pravougaonika traži se srednja vrednost svih piksela, koja je potom kriterijum binarizacije. Svi pikseli datog dela slike uzeće vrednost jedinice ukoliko je njihova vrednost veća od izračunate srednje vrednosti piksela tog dela i suprotno.

Izdvajanje kontura

Konture (ivice) čine oni pikseli na čijem položaju dolazi do značajne promene vrednosti piksela. U datom slučaju, potrebno je efikasno otkriti zatvorene konture. Za potrebe ovog rada korišćen je algoritam koji su predložili Suzuki i Abe (1987) Ideja funkcionisanja ovog algoritma jeste u obilaženju zatvorenih kontura piksela u pravcu kazaljke na satu. Kroz sliku se prvobitno prolazi u smeru dole desno, počevši od gornjeg levog piksela. Po nailasku na prvi piksel kome je binarizacijom dodeljena vrednost jedinice, ulazi se u proces obilaženja konture. Zavisno od odabranog stepena povezanosti konture koji određuje da li okolina sadrži četiri ili osam piksela, u okolini piksela na koji smo naišli traži se drugi koji ima vrednost jedinice, i to u smeru kazaljke na satu. Počevši od drugog pronađenog piksela, takođe u smeru kazaljke na satu oko prvog piksela, pronalazi se treći. Treći piksel postaje prvi, dok prvi postaje drugi i proces se za njih ponavlja. Po dolasku do prvog piksela koji je nađen, proces se prekida i nastavlja se obilazak slike i proces na onim pikselima koji nisu obuhvaćeni prethodno otkrivenim konturama. Pikselima koji su prošli obilazak dodeljuju se određene vrednosti. Te vrednosti se razlikuju za svaku konturu, pa se time konture i razlikuju (slika 6).



Slika 6. Blok dijagram Suzuki Abe algoritma

Figure 6. Block diagram of the Suzuki Abe algorithm

Aproksimiranje kontura mnogouglovima i izdvajanje četvorouglova

Izdvojene konture predstavljaju niz povezanih piksela. Delovi takvog niza sadrže piksele koji svojim položajem značajno ne određuju oblik konture. Korišćenjem Daglas-Pekerovog algoritma postiže se uprošćenje postojećih kontura, odnosno njihova poligonizacija (Douglas i Peucker 1973) Pomenuti algoritam funkcioniše tako što za dve krajnje tačke niza tačaka traži najudaljeniju tačku pravoj koju krajnje tačke određuju. Ukoliko je ta udaljenost veća od predefinisane vrednosti, onda se ta najudaljenija tačka uzima za značajnu u određenju oblika konture. Postupak se ponavlja za dva skupa tačaka, levo i desno u nizu tačaka u odnosu na prethodnim korakom odabranu tačku. Kako samo četvorouglovi definišu delove slike na kojima se može nalaziti marker, konture oblika ostalih mnogouglova bivaju odbačene.

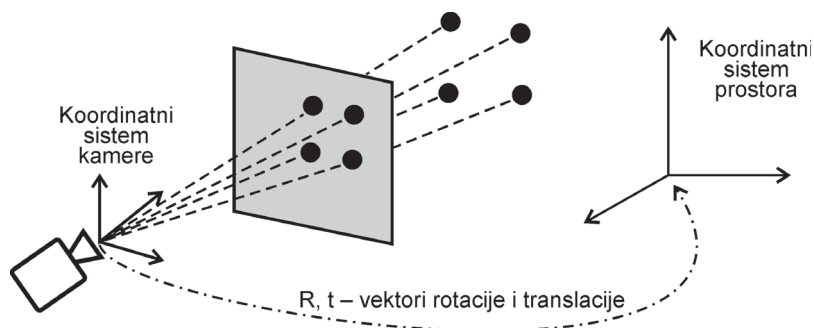
Implicitno ispravljanje, dekodifikacija i pretraga u rečniku

Po utvrđivanju kandidata za markere, potrebno je određeni deo slike transformisati u binarni kod. Kako projekcija markera može biti izrazito nepravilnog oblika, potrebno je izvršiti ispravljanje. Ispravljanje se vrši deljenjem izdvojenih četvorouglova pravama određenim odgovarajućim tačkama ravnomerno raspoređenim po naspramnim ivicama izdvojenih četvorouglova. Polja koja se izdvajaju pravama formiranom improvizovanom mrežom predstavljaju projekcije odgovarajućih polja markera. Računajući srednju vrednost piksela unutar pojedinačnih polja mreže, utvrđujemo kodifikaciju četvorougla. Određeni binarni kod pretražujemo u rečniku i ukoliko je razlika između nekog od kodova postavljenih markera i određenog binarnog koda manja od određene vrednosti, taj marker se smatra detektovanim u okviru postojećeg četvorougla.

PnP

Kako bi se na osnovu koordinata temena markera na slici i u prostoru odredila pozicija i orijentacija kamere u prostoru, koristi se PnP algoritam (Lepetit *et al.* 2008)

Koordinate temena markera u prostoru date su u odnosu na zadati referentni sistem, dok su koordinate temena markera na slici date u odnosu na sistem vezan za sliku. Kao ulaz ovog algoritma koriste se prostorne koordinate detektovanih temena markera vezane za binarni kod pronađenog markera, koordinate projekcija temena markera na slici i matrica kamere koja definiše vezu između položaja tačke na slici i u prostoru. Sistem kamere predstavlja referentni sistem čija z osa odgovara optičkoj osi kamere i čije x i y osa odgovaraju osama u referentnom sistemu vezanom za sliku. Izlaz pomenutog algoritma predstavlja vektor translacije i rotacije u sistemu kamere (R, t) kojim se sistem kamere transformiše u sistem prostora, koji



Slika 7.
PNP algoritam

Figure 7.
PNP algorithm

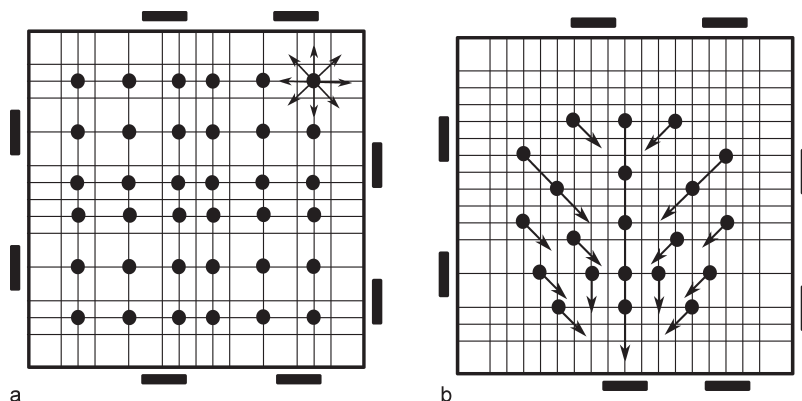
predstavlja na početku zadati referentni sistem. Inverznom transformacijom dobijenih vektora dobija se potreban rezultat o položaju kamere.

U osnovi, PnP za svake tri tačke formira sistem kosinusnih jednačina, gde nepoznate predstavljaju udaljenost svake tačke od optičkog centra kamere (slika 7). Rešavanjem ovog sistema dobijaju se udaljenosti koje se trianguliraju. Ova metoda se naziva iterativnom metodom i dolazi u okviru OpenCV implementacije PnP algoritma (Docs.opencv.org 2018b).

Rezultati i diskusija

Napravljene su 2 baze slika. U prvoj bazi iz svake od 36 obeleženih tačaka napravljeno je po 8 fotografija za 8 različitih uglova od 0° do 360° sa korakom od 45° (slika 8a). Od ukupno 288 slika, markere uočavamo na njih 218. Od toga na 113 slika uočavamo 1, na 92 slike 2, a na 13 slika 3 markera. Druga baza sastojala se od 2 seta od po 19 slika na kojima uočavamo po 1 identičan marker (slika 8b). Tačke i orijentacija kamere u drugoj bazi bili su raspoređeni oko jednog markera tako da se variraju ugao i udaljenost markera i kamere.

Uz svaku fotografiju u bazi prilaže se odgovarajući fajl sa referentnim merenjem, odnosno vrednostima pozicije i rotacije kamere pri njenom nastanku.



Slika 8.
Pozicije kamere pri fotografisanju slika prve (a) i druge (b) baze

Figure 8.
Camera position when photographing the image of the first (a) and second (b) base

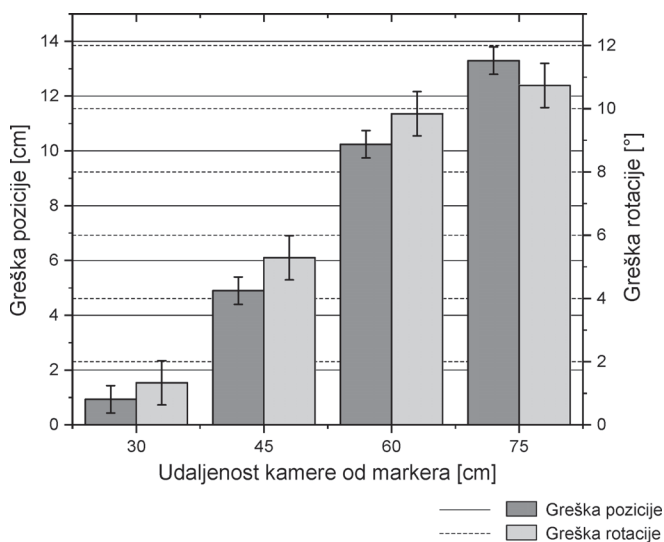
Rezultati algoritma za detekciju markera predstavljeni su u tabeli 1.

Tabela 1. Rezultati algoritma za detekciju markera

Broj markera na slici	Broj slika	Uspešnost detekcije (%) u slučaju kad algoritam:			
		ne pronalazi markere	pronalazi 1 marker	pronalazi 2 markera	pronalazi 3 markera
0	70	94.3	5.7	0	0
1	113	10.6	88.5	0.9	0
2	92	0	21.7	78.3	0
3	13	0	15.4	53.9	30.7

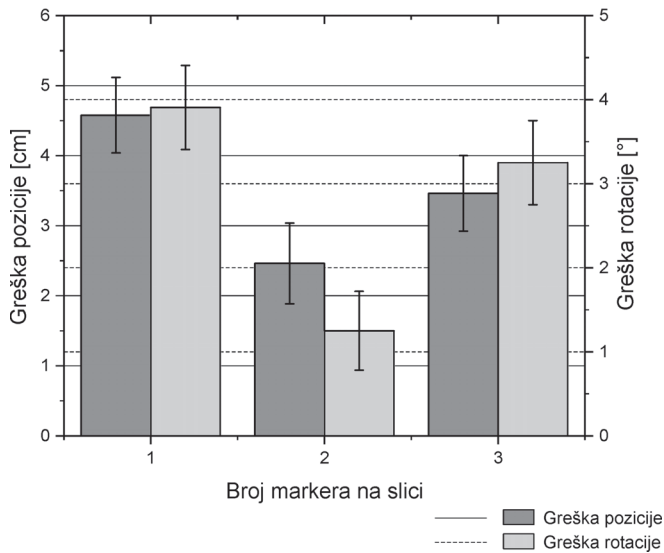
Zavisnost greške izračunavanja položaja kamere od udaljenosti između kamere i markera predstavljena je na slici 9. Primećujemo da greška raste sa porastom udaljenosti između markera i kamere. Porast udaljenosti između kamere i markera čini projekciju markera na slici manjom što kao posledicu ima neprecizniju detekciju temena markera, što je i uzrok rasta greške lokalizacije.

Zavisnost greške pozicije i rotacije kamere od broja markera koji se na slici detektuju prikazana je na slici 10. Primećujemo da tačnost lokalizacije raste sa povećavanjem broja markera na slici. Do ove pojave dolazi zbog većeg broja podataka na ulazu PnP algoritma. U slučaju gde se na slici vide 3 markera, kamera je značajno udaljena od istih. Kao što je prethodno diskutovano, greška lokalizacije raste sa udaljenošću, što objašnjava ovo odstupanje, odnosno veću tačnost lokalizacije na slikama gde se vide 2 u odnosu na slike gde se vide 3 markera.



Slika 9. Zavisnost greške izračunavanja položaja kamere od udaljenosti između kamere i markera

Figure 9. Dependence of the error of calculating the position of the camera (dark grey – position, light grey – rotation) from the distance between the camera and the marker



Slika 10. Zavisnost greške pozicije i rotacije kamere od broja markera na slici

Figure 10. Dependence of position (dark grey) and rotation (light grey) error of the camera from the number of markers on the picture

Zaključak

Sistem lokalizacije na osnovu markera je uspešno implementiran. Primećeno je da greške lokalizacije i rotacije kamere značajno zavise od udaljenosti kamere i markera i broja markera koji se na slici vide. Sa porastom udaljenosti između kamere i markera greška lokalizacije raste, dok se u slučajevima kada se na slici vidi više markera greška lokalizacije smanjuje.

Prikazani sistem lokalizacije ispunjava potrebe lokalizacije u poznatom okruženju. Prednosti sistema ogledaju se pre svega u vremenu izvršavanja, koje dozvoljava upotrebu sistema u realnom vremenu. Najveća mana sistema jeste u tome što dozvoljava postojanje položaja kamere u kojima nema rezultata lokalizacije. U potencijalnim primenama ovog sistema korisnik neće imati informaciju o svojoj lokaciji u određenim položajima. Pritom, možemo zaključiti i da je pokrivenost okruženja markerima bitan faktor u prevenciji takve pojave. Do pojave da nije detektovan nijedan marker može doći ukoliko nijedan marker nije moguće uočiti na slici, ali i ukoliko je položaj kamere u odnosu na taj marker takav da sistem ne detektuje taj marker. Pod poslednje pomenutim položajem kamere u odnosu na marker podrazuma se da je kamera previše udaljena od markera ili da se nalazi pod ostrim uglom koji neomogućuje detekciju. Ukoliko pretpostavimo da se markeri nalaze na zidovima okruženja u kojem se sistem primenjuje, dolazi se do zaključka da će sistem pokazati bolje rezultate ukoliko je to okruženje manjih dimenzija.

Rezolucija slike koju pravi kamera i brzina rada procesora predstavljaju hardverske faktore koji bi doprineli boljoj lokalizaciji i koji otvaraju prostor za dodatna istraživanja na datu temu.

Literatura

- Beyerer J., León F. P., Frese C. 2016. *Machine Vision*. Springer
- Burghardt A., Kazakia G., Majumdar S. 2007. A Local adaptive threshold strategy for high resolution peripheral quantitative computed tomography of trabecular bone. *Annals of Biomedical Engineering*, **35**: 1678.
- Docs.opencv.org. 2018a. Camera calibration With OpenCV – OpenCV 2.4.13.7 documentation.
https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html
- Docs.opencv.org. 2018b. OpenCV: Camera Calibration and 3D Reconstruction. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html
- Douglas D., Peucker T. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, **10** (2): 112.
- Garrido-Jurado S., Muñoz-Salinas R., Madrid-Cuevas F., Marín-Jiménez M. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, **47** (6): 2280.
- Garrido-Jurado S., Muñoz-Salinas R., Madrid-Cuevas F., Medina-Carnicer R. 2016. Generation of fiducial marker dictionaries using Mixed Integer Linear Programming. *Pattern Recognition*, **51**: 481.
- Lepetit V., Moreno-Noguer F., Fua P. 2008. EPnP: An accurate o(n) solution to the PnP problem. *International Journal of Computer Vision*, **81** (2): 155.
- Romero-Ramirez F., Muñoz-Salinas R., Medina-Carnicer R. 2018. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, **76**: 38.
- Suzuki S., Abe K. 1987. Binary picture thinning by an iterative parallel two-subcycle operation. *Pattern Recognition*, **20** (3): 297.
- Szeliski R. 2010. *Computer vision*. Springer
- Weng J., Cohen P., Herniou M. 1992. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14** (10): 965.

Indoor Localization Using Fiducial Markers

In this paper, the possibilities of indoor based object localization using AruCo markers were examined. AruCo markers are synthetic square markers with high contrast edges. Together with the black frame, they represent a black and white 8×8 dimension matrix, where black or white fields determine the binary matrix. AruCo's black frame allows for quick detection, while its binary codification allows for its easy identification. The system consists of 8 AruCo markers placed on the walls surrounding 1×1 m space and a camera (Figure 2). The camera was located on an object the location of which in the set space was determined.

Localization is done in two basic steps (Figure 1). The first step involves the detection, segmentation and recognition of markers in camera images. After distortion removing, the image is binarized by the local adaptive threshold method which assigns a binary value to each pixel, depending on whether its value is greater than or less than the mean pixel value in the given environment. After binarization, the contour of objects in the binary image is separated using the Suzuki-Abe algorithm (Suzuki and Abe 1987). The next algorithm conducts the approximation of singular contours composed of pixels to polygons, where only quadruple contours remain candidates for the contour of the markers (Douglas and Peucker 1973). A customized 8×8 mesh is placed on the parts of the images within the selected quadruple contours, the fields of which, depending on whether the majority of the pixels within them have a value of one or zero, determine the binary code. The obtained binary codes are compared with the codes of the set markers. If they do not differ by more than a certain value from their most similar markers, the points of the detected quadruples will be paired with the coordinates of the most similar marker in the coordinate system set in the space.

After identifying the marker, the coordinates of the marker vertices on the image and the coordinates of them in the space are passed to the solvePnP algorithm. The aforementioned algorithm, using the known dimensions of the markers in the space and the angles under which the camera's optical center looks at the marker's vertices, determines the distance of the individual vertices from the camera by geometric equations. The position of the camera is obtained by triangulation of the mentioned distances. The solvePnP algorithm results in a translational vector and a rotation vector that transforms the camera's coordinate system into a coordinate system set up in the space.

A series of images were obtained on which the method was tested. The database consists of 288 images made from 36 points correctly arranged in space. At each point, 8 images are created for 8 different angles. The position and orientation of the camera is varied by a rotating base placed on a surface on which the coordinate system is drawn (Figure 1). Of the 288 im-

ages created, the markers are spotted on 224 of them. A total of 349 markers were observable, of which the marker detection algorithm successfully detected 277.

The position and rotation error is defined as the euclidean distance of the position obtained and the rotation of the camera from the reference values of the same. In the pictures where one marker is spotted, the mean position error is 8 cm and the mean rotation error is 8° . In the pictures where two markers are spotted, the mean position error is 4 cm and the mean rotation error is 4° , while in the images with three markers, the mean position error is 4 cm and the mean rotation error is 5° .

It is noted that localization is much more precise when two or more markers appear in the image. This is due to the very nature of the work of the solvePnP algorithm, which is due to the large number of data that is brought to its input.

The obtained results verify the described method and confirm the hypothesis that a larger number of markers in the image leads to more precise localization.

