

## FRI Rehab 3D: 3-D rekonstrukcija kvadra s pomočjo kamere pri rehabilitaciji človeške roke

**Boris Simončič, Peter Peer**

*Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Tržaška 25, 1000 Ljubljana, Slovenija  
E-pošta: boris.simoncic@fri.uni-lj.si, peter.peer@fri.uni-lj.si*

**Povzetek.** V članku je obdelan problem rehabilitacije pacienta po poškodbi roke ali zapestja s pomočjo osebnega računalnika in ene spletne kamere. Pacient drži v roki preprost objekt, kvader, in ga premika. Kamera snema njegove premike, program pa na podlagi slike kamere s pomočjo barvne informacije in z uporabo modela kvadra izračuna lego predmeta v 3-D prostoru v realnem času. Program umesti ta predmet v navidezni 3-D prostor, v katerem pa je še en takšen predmet. Naloga pacienta je, da premakne realni predmet v lego navideznega. S tem trenira določene gibe zapestja in roke, ki pripomorejo k hitrejšemu okrevanju. Zato je problem zanimiv tako s tehničnega kot tudi s čisto praktičnega vidika. Pri predlagani rešitvi izhajamo iz želje omogočiti to rešitev čim večjemu krogu uporabnikov, zato smo se omejili na dve zelo osnovni zahtevi: uporabnik mora imeti osebni računalnik in spletno kamero. Na koncu podajamo tudi ocene delovanja programa, ki temeljijo na kvantitativni in kvalitativni analizi množice primerov. Na podlagi rezultatov in ob pogoju, da je uporabnik pripravljen tudi sam vložiti nekaj truda, lahko povzamemo, da smo se z opisano rešitvijo dovolj približali načrtanim ciljem. To velja predvsem tedaj, ko natančnost ni odločilna in manjša odstopanja pri rekonstrukciji ne onemogočajo postopka rehabilitacije.

**Gljučne besede:** računalniški vid, model kvadra, 3D rekonstrukcija kvadra, rehabilitacija roke, izvajanje v realnem času, spletna kamera

## FRI Rehab 3D: 3D-reconstruction of the cuboid based on a single camera used in rehabilitation of the human hand

**Extended abstract.** The paper presents a rehabilitation method using a personal computer and a single web camera for patients who suffer from an arm or wrist injury. Patient holds in his hand a simple object, cuboid, and moves it around. A camera records his movement while the software calculates in real-time the position of the object in a 3D-space on the basis of the color information and the cuboid model. The program places the object in a virtual 3D-space in which another similar object is already present.

In Chapter 2 we present the existing rehabilitation solutions, together with their accuracies and prices. As seen, they can be very expensive and need specialized hardware. Therefore, the goal of our work was to develop a much simpler and less expensive solution demanding only a personal computer and a single web camera.

In Chapter 3 we present the environment required to achieve good results (Figure 1). To segment out the object more easily we need a black background and black glove. The lighting is one of the most important factors. In Chapters 4 and 5 we discuss the methods of obtaining cuboid characteristics such as edges and corners from input image. The edges were obtained by separating the sides of the cuboid from the original image and then intersecting them. This leaves only a very narrow set of points which we use as a basis for linear regression to determine the edges (Figure 2). Determining the corners is divided into three sub-problems. When only one side is visible, a bounding rectangle is calculated touching the cuboid on four points. We

improve their location with local optimization. When two sides are visible, we use the intersection method to detect the first two corners. To detect the other four ones, we calculate and analyze perpendiculars to the intersected edge (Figure 3). In the case of three visible sides, the intersection method is used. The analysis of the resulting edges gives four corners. Approximations of the remaining three corners are calculated on the basis of the first four and refined by using optimization (Figure 4).

In the Chapter 5 we discuss how the object is projected on the image plane (Figure 5) and how the 3D-coordinates of the cuboid can be obtained on this basis (Equation 1). We describe how the cuboid model is used to construct and solve the system of equations thus giving 3D-coordinates for one side of the cuboid. The rest of the sides can be calculated with a vector product. In Chapter 7, a method of matching a virtual and real object is presented (Figure 6).

In Chapter 8 we evaluate the software. Our evaluation is based on a quantitative and qualitative analysis performed on a set of examples. The quantitative analysis gives a graph that presents the correlation between the distance of the object from the origin and the size of the error (Figure 7). In Tables 1 and 2 we present errors calculated with methods like geometric mean, median, average and standard deviation. In Figure 8 distribution of the quantitative and qualitative errors can be found. The qualitative analysis was performed on a set of three videos where all the frames (2160 frames in the first video with only one visible side, 2677 frames in the second video with two visible sides, 3477 frames in the third video with three visible sides) were assigned ranking: good, acceptable or bad. The analysis shows that 4,2 % of the frames were displayed incorrectly, but only 1

% of those were assigned the ranking bad. Such analysis however depends on many factors and results are to be taken as an approximation.

In the concluding chapter we compare our solution to those of others (Table 3) and present some ways of how to improve the accuracy and efficiency of our software. Finally we conclude that under the condition that the user himself is willing to make his own effort, the presented solution serves its purpose well enough despite some deficiencies. It is suitable mostly in cases where accuracy is not critical and smaller deviations of 3D-reconstruction do not thwart the process of the rehabilitation. Although the accuracy is quite good, under appropriate conditions, we do think that this kind of approach has a potential for significantly better results.

**Key words:** computer vision, cuboid model, 3D reconstruction of a cuboid, arm rehabilitation, real-time execution, web camera

## 1 Uvod

Cilj tega dela je razviti program, s katerim bi pacientom olajšali rehabilitacijo po poškodbi roke, zapestja ali dela možganov, ki nadzoruje ta del. Vsako leto na milijone ljudi po svetu doživi kap, samo v Združenih državah Amerike ta številka presega 600.000 [1]. Poškodovanec mora ustrezni del možganov spet naučiti oz. navaditi uporabljati roko ali zapestje. Ustrezna terapija po kapi traja zaradi motenj gibanja več tednov po poškodbi. Zaradi ekonomskih pritiskov pa imajo pacienti čedalje manj teh terapij in jih hitro pošljejo domov. Tako tudi koristi terapije niso v celoti dosežene. Zato je pomembno razviti sistem, ki bo pacientom omogočal nadaljevanje terapije brez prisotnosti zdravnika.

## 2 Sorodna dela

Ena izmed rešitev, FRI Rehab 2D, je bila razvita na Univerzi v Ljubljani [2]. Ta program ob pogojih, kot sta ustrezna osvetlitev in ne prehitro premikanje predmeta, v realnem času uspešno določa lego predmeta. Predmet uspešno določi tudi če je ta delno zakrit. Sliko zajema s črno-belo kamero. Slabost te rešitve je, da deluje v dveh dimenzijah. Naša rešitev je nadgradnja tega sistema s prehodom v tri dimenzije, uporaba barvne informacije in modela kvadra.

Sistem Optotrak Certus ima natančnost do 0,1 mm in ločljivost 0,01 mm. Sledi lahko do 512 markerjem (IR-diode) z največjo hitrostjo osveževanja 4600 Hz. Sistem Optotrak Certus vsebuje tri kamere, občutljive za infrardečo svetlobo. Sistem stane okrog 150.000 dolarjev, a ni omejen samo na rehabilitacijo, temveč se uporablja tudi v industriji, biomehaniki in na raziskovalnih inštitutih [3].

Xsensov MTi je miniaturni sistem za zaznavanje lege in smeri. Vsebuje notranji signalni procesor in posebne senzorje, s katerimi določa 3-D orientacijo [4]. Kljub širši namembnosti bi lahko sistem MTi uporabili tudi pri rehabilitaciji. Sistema stane približno 2500 evrov. Po navedah proizvajalca je napaka premikajočega se senzorja manj kot dve kotni stopinji. Za kvader, kakršnega smo

uporabljali za naš program (dolžina 14 cm, širina 8 cm, višina 6 cm), smo preračunali napako, ki znaša 5,35 mm.

Pristop za sledenje v 3-D koordinatah, ki sta ga uporabila Y. Tao in H. Hu, integrira kamero in senzor, ki meri inercijo (angl. inertial sensor). Z dodatnimi optimizacijskimi algoritmi sta dosegla natančnost  $\pm 5$  cm za vse tri koordinate [5].

## 3 Predpriprave

Pred začetkom dela s programom je treba pripraviti prostor (slika 1) in opremo, saj je pravilnost delovanja programa močno odvisna od teh dejavnikov.



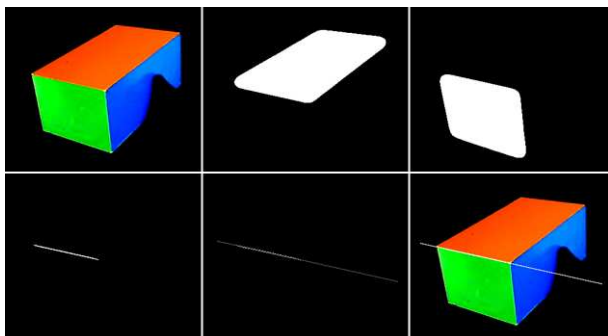
Slika 1. Ureditev delovnega okolja  
Figure 1. Work space arrangement.

V našem primeru je na zajeti sliki pomembna samo ena stvar – kvader, ki ga premikamo. Zato je pomembno, da lahko kvader ločimo od okolja. V ta namen smo uporabili podobno rešitev, kot jo pogosto uporabljajo na televiziji pri napovedovanju vremena, in sicer t. i. modri zaslon (angl. blue screen). Ideja te rešitve je, da obdržimo samo tiste dele slike, ki niso neke določene barve, v našem primeru smo se odločili za črno. Uporabimo lahko zaveso črne barve, ki jo namestimo tako, da pokriva celoten del vidnega polja kamere. Na roko, s katero držimo in premikamo kvader, si nadenemo še črno rokavico. Tako je na zajeti sliki vse, razen kvadra, črne barve.

Podatke o predmetu vnesemo v program v posebnem oknu, kjer podamo dimenzije kvadra, torej dolžino, širino in višino ter barve stranic. Lastnost, ki jo je treba upoštevati, je material, iz katerega je narejen sam kvader ali njegove stranice. Dober material je tak, ki je hkrati dovolj svetel in obenem ne preveč odbojen. Odbojnost in osvetljenost kvadra pa je odvisna tudi od virov svetlobe v prostoru. Temu je treba nameniti še posebno pozornost tudi zaradi kamer, ki jih uporabljamo, saj se osredinjamo na kamere najnižjega cenovnega razreda. Najbolje je torej uporabiti vire svetlobe, ki osvetljujejo kvader čim bolj enakomerno iz vseh strani. Dobro se obnese kombinacija dnevne svetlobe z enim ali dvema viroma razpršene umezne svetlobe.

#### 4 Iskanje robov

Za določanje oglišč potrebujemo čim bolj natančno določene notranje robove kvadra. Te določimo na podlagi presekov posameznih parov stranic (slika 2).



Slika 2. Presek stranic pri določanju roba med njima  
Figure 2. Side intersection in edge detection.

Najprej ugotovimo, katere stranice so na sliki, in nad njimi paroma naredimo preseke. Točke posameznih presekov zberemo v posebne množice, nad katerimi z linearno regresijo [6] določimo premice, ki te točke pokrivajo z najmanjšo napako.

#### 5 Iskanje oglišč

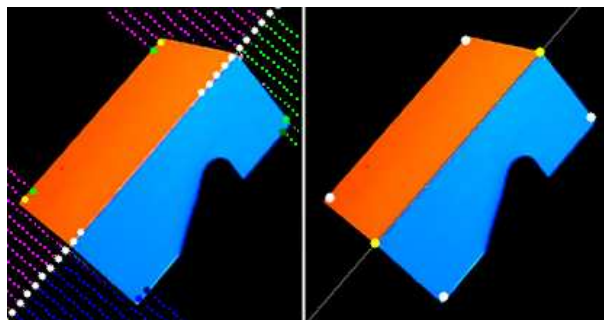
Oglišča so za računanje položaja objekta zelo uporabne značilke, zato je velik del programa namenjen njihovem iskanju. Pomemben del reševanja tega problema je metoda iskanja robov. Reševanje si olajšamo, če celoten problem določanja oglišč razdelimo na tri podprobleme.

##### 5.1 Vidna ena stranica

Ta podproblem je sestavljen iz dveh delov. V prvem delu poiščemo grob približek oglišča in ga v drugem koraku poskušamo izboljšati z lokalno optimizacijo. Približek iščemo tako, da določimo natančen najmanjši okvir. Ta okvir se na štirih točkah dotika pravokotnika, ki predstavlja objekt. Pravokotnik se dotika okvirja na štirih točkah, kar je približen položaj oglišč. Sledi optimizacija na omejenem območju. Za vsako belo točko v tem območju izračunamo število sosednjih točk bele barve. Tam, kjer je število točk minimalno, je oglišče.

##### 5.2 Vidni dve stranici

Kadar sta na sliki vidni dve stranici, moramo po prej opisanem postopku najti premico, ki določa rob med njima. Nato tej premici sledimo in ko naletimo na prvo nečrno točko, jo označimo za en konec roba. Premici sledimo še naprej, dokler ne naletimo tokrat na črno točko, kar označuje drugi konec roba.

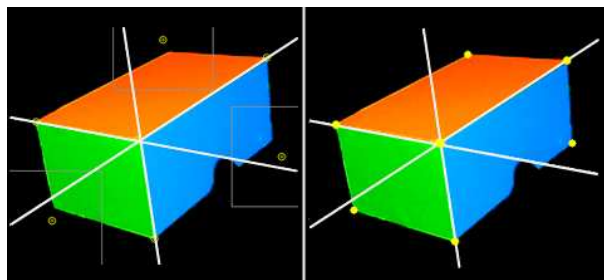


Slika 3. Iskanje ostalih oglišč s pravokotnicami.  
Figure 3. Detection of other corners with perpendiculars.

Naslednji korak je iskanje približkov preostalih štirih oglišč. V okolici prvega oglišča določimo  $n$  pravokotnic na prej dobljeno premico (slika 3). Za vsako od pravokotnic nato izračunamo razdaljo od njene izhodiščne točke, ki leži na premici, do prve ne črne točke, torej prve točke, ki ne leži več na kvadru. Vseskozi spremljamo dolžine pravokotnic. Ko pravokotnica ne teče čez kvader, je njena dolžina nič. Prva naslednja pravokotnica, ki teče čez kvader, pa ima dolžino različno od nič. Algoritem ugotovi spremembo dolžin ter tam določi prisotnost novega oglišča. Sledi še optimizacija prej dobljenih oglišč.

##### 5.3 Vidne tri stranice

Prvi korak je določitev notranjih robov z metodo preseka. S pomočjo vseh treh robov lahko zelo natančno določimo prvo notranje oglišče kvadra. Dodatna tri oglišča dobimo tako, da sledimo vsakemu od treh robov. Začnemo pri izhodišču in se premikamo po premici roba, dokler ne pridemo do prve ne črne točke.



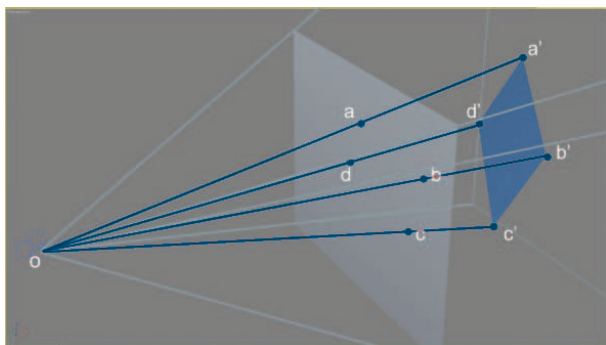
Slika 4. Na levi sliki so prikazani robovi in prva štiri oglišča ter približki preostalih treh oglišč. Desno je rezultat po optimizaciji.  
Figure 4. Left picture shows the edges and the first four corners plus three approximate corners. The picture on the right shows the result after optimization.

Preostala so nam še tri neznana oglišča. Sprejmemo predpostavko, da štiri oglišča, ki tvorijo eno stranico, določajo pravokotnik. Tri oglišča tega lika že imamo. Četrtega izračunamo z zrcaljenjem enega od oglišč prek premice, ki jo določata preostali dve oglišči. Isti postop-

pek ponovimo še za preostali stranici. Izkaže se, da so optimizirana oglišča dobro določena (slika 4).

## 6 Določanje 3-D koordinat

Na podlagi 2-D koordinat oglišč, modela kvadra, ločljivosti in goriščne razdalje kamere lahko izračunamo 3-D koordinate. Uporabljali smo kamero Logitech Quickcam 5000 Pro z ločljivostjo  $640 \times 480$  točk. Objekt se preslika na slikovno ravnino, ki tvori sliko, in na podlagi katere smo po prej opisanih postopkih določili značilki oz. njihovi koordinati  $x$  in  $y$ . Ker je slikovna ravnina od leč kamere oddaljena za goriščno razdaljo, nam ta razdalja pomeni koordinato  $z$  značilk. Za lažjo ponazoritev problema smo si pomagali z orodjem 3D Studio (3ds) Max (slika 5). Točke na sliki označimo s črkami **a**, **b**, **c** in **d**. Točke objekta oz. oglišča, ki jih iščemo, pa označimo z **a'**, **b'**, **c'** in **d'**. Poznamo tudi razdalje med njimi. Kamera je na točki z oznako **o**, ki naj bo tudi izhodišče koordinatnega sistema.



Slika 5. Podrobnejši prikaz preslikave objekta na sliko  
Figure 5. Detailed illustration of an object projected on the image plane.

Vidimo lahko, da je zveza med oglišči kvadra, ki jih iščemo, ter oglišči kvadra, preslikanimi na slikovno ravnino, naslednja:

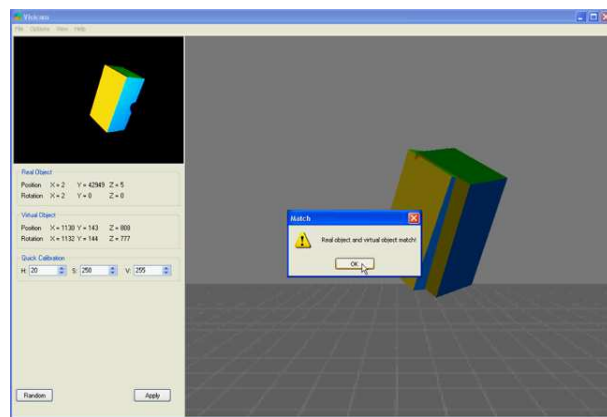
$$\begin{aligned} \vec{oa}' &= k_1 \cdot \vec{oa} \\ \vec{ob}' &= k_2 \cdot \vec{ob} \\ \vec{oc}' &= k_3 \cdot \vec{oc} \\ \vec{od}' &= k_4 \cdot \vec{od} \end{aligned} \quad (1)$$

Ugotovimo lahko, da so konstante  $k_1$ ,  $k_2$ ,  $k_3$  in  $k_4$  enolično določene, če vpeljemo dve omejitvi, ki izhajata iz modela kvadra. Prva omejitev je presečišče obeh diagonal. Druga omejitev pa je dolžina stranice, ki jo kot vhodni podatek določimo na začetku programa pri kalibraciji. Obe omejitvi skupaj tvorita štiri enačbe, ki skupaj enolično določajo sistem s štirimi neznankami. Tako smo izračunali neznanke  $k_1$ ,  $k_2$ ,  $k_3$  in  $k_4$  ter določili točke **a'**, **b'**, **c'** in **d'**.

Določanja preostalih stranic se lahko lotimo z vektorskim produktom, s katerim določimo pravokotnice na vsa štiri oglišča prej določene stranice. Rezultat moramo še normalizirati tako, da dolžino, ki jo vrne vektorski produkt, delimo z dejansko dolžino. To seveda poznamo, saj je to ena izmed dimenzij kvadra. S tem količnikom nato delimo vse tri komponente novega vektorja in tako dobimo koordinate novega oglišča kvadra. Postopek moramo ponoviti za vsa preostala oglišča in s tem je postopek določanja 3-D koordinat končan in kvader lahko postavimo v navidezni 3-D prostor.

## 7 Ujemanje z navideznim predmetom

Navidezni kvader moramo za simulacijo postaviti v naključno lego v navideznem prostoru ter ga nato čim bolj pokriti z realnim kvadrom.



Slika 6. Ujemanje navideznega in realnega kvadra izpiše sporočilo o uspehu  
Figure 6. Matching of a virtual and real cuboid is displayed by a success message.

Najprej določimo središče realnega kvadra in središče navideznega kvadra. Kadar je razdalja med obema središčema manjša od določene vrednosti, izračunamo še razdalje med oglišči navideznega in realnega kvadra. Kadar so te razdalje manjše od vnaprej definiranega parametra, lahko rečemo, da sta lega in rotacija obeh predmetov enaki (slika 6).

## 8 Rezultati

V tem poglavju smo pri kvantitativnih meritvah najprej v programu 3ds Max ustvarili prostor, v katerega smo v naključno lego postavili ustrezen kvader ter ga zasukali. Sliko smo upodobili (angl. render) ter jo uporabili kot vhodni podatek v naš program, ki je nato izračunal položaj kvadra. Izračunane koordinate oglišč kvadra smo primerjali s koordinatami kvadra v 3ds Maxu. Primerjali smo vsa vidna oglišča. Vse enote so v milimetrih. Sistem smo razvijali na osebni računalniku s procesorjem Core2 Duo 1,86 GHz z operacijskim sistemom Windows

XP v okolju VS.NET C++. Izmerili smo tudi hitrost delovanja programa in večinoma se giblje med štirimi do devetimi slikami na sekundo.

Kvantitativna napaka je definirana kot razdalja med pravilno lego oglišča, ki jo dobimo v 3ds Maxu, ter izračunano lego oglišča programa. Kvantitativna napaka v tem primeru ne da vedno pravi občutka o uporabnikovi izkušnji programa in njegovi natančnosti, zato smo vpeljali približek kvalitativne vizualne napake, ki zanemari koordinato  $z$ , saj je napaka na tej koordinati uporabniku veliko manj opazna kot pa napaka na koordinatah  $x$  in  $y$ .

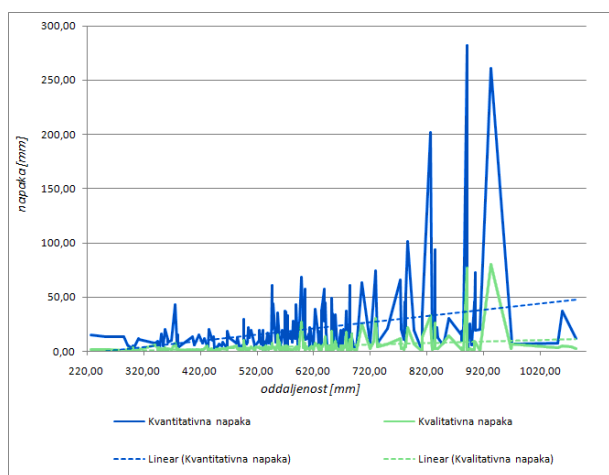
Za oba tipa napak je vključen tudi izračun nekaterih pogostejših kazalcev napak: povprečna napaka (angl. average) in standardni odklon (angl. standard deviation), srednja geometrična napaka (angl. geometric mean) in mediana (angl. median).

### 8.1 Pogosti problemi

V nekaterih situacijah program zelo slabo določi koordinate. Če je premalo svetlobe, se stranice ne vidijo v celoti. To težavo lahko odpravimo z dodatnimi viri svetlobe. Druga težava lahko nastopi, ko se zaradi direktne točkaste svetlobe stranice obarvajo drugače. V tem primeru moramo vir svetlobe zatemniti ali pa obrniti tako, da žarnica ne sveti naravnost v kvader. Preveriti moramo tudi, da ni material stranic kvadra preveč odbojen.

### 8.2 Kvantitativna ocena

Tu smo proučili skupaj 45 primerov, od tega 15 tam, kjer je vidna ena stranica, 15 kjer sta vidni dve stranici in 15 kjer so vidne vse tri stranice (skupaj 226 oglišč).



Slika 7. Graf korelacije med razdaljo od izhodišča ter velikostjo napake  
Figure 7. Graph showing correlation between the distance from the origin and the error size.

Izračun napak prikazujeta tabeli 1 in 2.

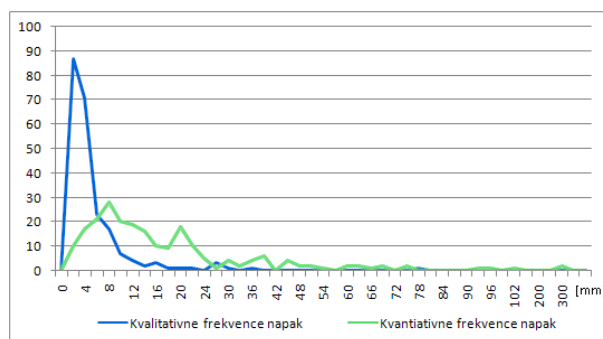
ocena	sred. geom. napaka	mediana
kvantitativna	11,73 mm	11,58 mm
kvalitativna	2,51 mm	2,34 mm

Tabela 1. Izračunane napake na podlagi 226 oglišč  
Table 1. Errors calculated on the basis of 226 corners.

ocena	povp. napaka ± std. odst.
razdalja od izh.	3,02 % ± 3,84 %

Tabela 2. Povprečna napaka, izražena v odstotkih dejanskih razdalj  
Table 2. Average error expressed in percentages of the actual distances.

Na sliki 7 je vidna korelacija med razdaljo oglišča od izhodišča in velikostjo napake za kvantitativno in kvalitativno napako. Graf prikazuje tudi linearni funkciji obeh napak. Na sliki 8 je prikazana razporeditev frekvenc kvantitativnih in kvalitativnih napak. Stabilnost je dovolj dobra, še posebno pri kvalitativnih meritvah. Tako napaka v odstotkih kot grafi stabilnosti kažejo na zadovoljivo delovanje.



Slika 8. Razporeditev frekvenc kvantitativnih in kvalitativnih napak  
Figure 8. Distribution of the quantitative and qualitative errors.

### 8.3 Kvalitativna ocena

Za vizualno (subjektivno) ocenjevanje delovanja programa smo vpeljali tri ocene: dobro (brez deformacije), sprejemljivo (majhna, srednja deformacija) in slabo (kvader je zelo deformiran oz. izgine).

Test smo izpeljali na treh videosekvencah. Pred testiranjem smo še enkrat določili barve ter preverili oz. spremenili ali dodali svetlobne vire. Tako je bilo hkrati dovolj svetlo, a brez preveč direktne svetlobe, ki bi se na kvadru preostro odsevala in povzročila slabšo detekcijo. V prvem videu (2160 slik, 72 sekund) smo premikali in obračali kvader tako, da je bila večinoma vidna le ena stranica. V drugem videu (2677 slik, 89 sekund) smo premikali in

obračali kvader tako, da sta bili večinoma vidni dve stranici. V tretjem videu (3477 slik, 116 sekund) pa smo premikali in obračali kvader tako, da so bile večinoma vidne vse tri stranice.

Rezultati so pokazali, da jih je od vseh slik le 4,2 % prikazanih napačno, vendar od teh samo 1,2 % tudi zelo slabo, preostalih 3 % pa je sprejemljivih. Ti rezultati so dosegljivi samo s pravilnim upoštevanjem osvetlitve, dobro kalibracijo barv in predhodno nastavitvijo parametrov kamere, kot je bilo opisano na začetku tega dela. Tudi sicer se te ocene lahko razlikujejo od videa do videa zaradi različnih dejavnikov, kot so oddaljenost kvadra od kamere, hitrost premikanja, kakovost izdelave kvadra ipd.

## 9 Sklep

Medtem ko druge rešitve uporabljajo napredno dodatno strojno opremo, smo se mi omejili le na navadno spletno kamero. To prinese tako prednosti kot slabosti (tabela 3). Pomembne prednosti so nizka cena in dostopnost, saj lahko program na sodobnem računalniku vsak uporablja kar doma. Glavni slabosti pa sta natančnost v primerjavi z dražjimi rešitvami in občutljivost za zunanje dejavnike.

Naša rešitev ima veliko možnosti za dopolnitev in izboljšanje delovanja. Ena možnost bi bila uporaba kamere z večjo ločljivostjo. Večjo izboljšavo bi prinesla tudi uporaba naprednejšega algoritma 3-D rekonstrukcije, ki bi bolje izrabil lastnosti modela kvadra. Program bi bilo mogoče razširiti tudi z vodenjem statistike pacienta, s katero bi imel fizioterapevt dodatno podlago za oceno napredka pacienta. Dodatna možnost bi bila tudi razširljivost z delitvijo programa na strežniški in odjemalski del.

	napaka	cena
naša rešitev	11,58 mm	cena kamere
Optotrak [3]	0,1 mm	pribl. 118.000 evrov
Xsens MTi [4]	5,35 mm	2.500 evrov
Tao, Hu [5]	50 mm	cena kamere, senzorja

Tabela 3. Sklepna primerjava različnih rešitev  
Table 3. Concluding comparison of different solutions.

Predlagano rešitev smo na koncu tudi ustrezno preizkusili. Na podlagi rezultatov lahko zaključimo, da smo se z opisano rešitvijo dovolj približali načrtanim ciljem. To seveda velja le ob pogoju, da je uporabnik upošteval podane smernice priprave prostora ter ustrezno nastavitve parametrov kamere. S tem lahko velikost napake in stabilnost detekcije zelo izboljšamo, vendar natančnosti dražjih rešitev kljub temu ne moremo doseči. Zato je opisana rešitev ustrezna pri metodah rehabilitacije, kjer izmerjene napake ne vplivajo na uspešnost terapije. Menimo pa, da

bi z vpeljavo zgoraj omenjenih izboljšav natančnost lahko še precej izboljšali.

## 10 Literatura

- [1] L. E. Sucar, G. Azcrate, R. S. Leder, D. Reinkensmeyer, J. Hernandez, I. Sanchez, P. Saucedo, "Gesture Therapy: A Vision-Based System for Arm Rehabilitation after Stroke", *Proceedings of the First International Conference on Health Informatics*, 2008, str. 107–111.
- [2] J. Ktrašnik, M. Veber, P. Peer, "Using computer vision in a rehabilitation method of a human hand", *Mediterranean Conference on Medical and Biological Engineering and Computing MEDICON 2007*, str. 947–949.
- [3] Technology in Motion, opis produkta: <http://www.ndigital.com/lifesciences/certus.php> (12.2.2009).
- [4] Xsens MTi, specifikacije: [http://www.xsens.com/Static/Documents/UserUpload/Xleaflets/leaflet\\_MTi.pdf](http://www.xsens.com/Static/Documents/UserUpload/Xleaflets/leaflet_MTi.pdf) (12.2.2009).
- [5] Y. Tao, H. Hu, "3D Arm Motion Tracking for Home-based Rehabilitation", *Proceedings of the 3rd Cambridge Workshop on universal access and assistive technology*, Cambridge, U.K., 2006, str. 105–111.
- [6] D. A. Forsyth, J. Ponce, "Computer Vision: A Modern Approach", Prentice Hall, 2003, str. 436–444.

**Boris Simončič** je diplomiral leta 2009 na univerzitetnem študiju Fakultete za računalništvo in informatiko Univerze v Ljubljani.

**Peter Peer** je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani.