

GALVOS POZICIJOS NUSTATYMAS EKRANO ATŽVILGIU NAUDOJANT
ANT GALVOS PRITVIRTINTĄ KAMERĄEvaldas Borcovas¹, Gintautas Daunys²

Šiaulių universitetas

El. paštas: ¹evaldas.borcovas@gmail.com; ²g.daunys@tf.su.lt

Santrauka. Nagrinėjamos galvos padėties pozicionavimo sistemos. Apžvelgti moksliniai straipsniai, kuriuose pateikiami įvairūs siūlomi metodai. Pasirinkta sistema, kai kamera pritvirtinama vartotojui ant galvos. Išskirti pagrindiniai galvos padėties nustatymo parametrai, nustatytas parametru suradimo eiliškumas, pateikta sistemos schema ir detali algoritmo blokinė schema. Algoritmui įgyvendinti taikyti briaunų radimo (Sobel) ir algoritmo tikslinimo (Subpixel) metodai. Sistema įgyvendinta Matlab ir C# aplinkose. Nustatyti optimalūs algoritmo vykdymo parametrai. Naudojamo kompiuterio procesorius Intel Pentium dual core T4500 – 2,3 GHz. Vidutinis algoritmo veikimo laikas Matlab aplinkoje yra 1,2 s, o C# aplinkoje – 126 ms. Ištyrus algoritmo segmentų vykdymo laiką, pastebėta, kad ilgiausiai vykdomas vaizdo filtravimas – 107 ms. Parodyta, kad algoritmo neapibrėžtis galima suskirstyti į statines ir matavimo neapibrėžtis. Maksimali statinė neapibrėžtis, matuojant galvos pozicijos padėtį, yra 1,63 mm, matuojant orientacijos parametrus – 0,16°. Maksimali matavimo neapibrėžtis yra 4,0 mm, matuojant galvos padėties parametrus, ir 0,11° – orientacijos parametrus.

Reikšminiai žodžiai: vaizdų apdorojimas, kompiuterinė rega, galvos padėties pozicija.

Įvadas

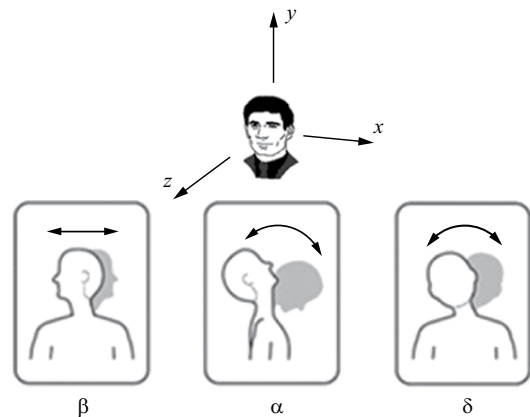
Galvos padėtis ekrane gali būti nustatoma įvairiais metodais. Nagrinėti metodai skiriasi algoritmo vykdymo sparta, tikslumu, pritaikomumu. Ne visi nagrinėti metodai pateikia visus galvos padėties nustatymo parametrus (1 pav.). Tai neleidžia metodų pritaikyti visose srityse. Metodai gali būti taikomi:

- 3D vaizduokliuose, įgyvendinant lango principą.
- Valdant kompiuterį galvos judesiais.
- Valdant kompiuterį akimis, įgyvendinant EyeWriter modifikaciją.
- Kompiuteriniuose žaidimuose.

Galvos padėties erdvėje nustatymo metodai

Taikant Matsumotoy, Zelinskyz (2000) pateiktą metodą (angl. *Stereo vision*) naudojamos dvi vaizdo kameros. Taikomas stereomatymo metodas. Pagal pateiktą informaciją šios sistemos tikslumas yra ± 1 mm, matuojant padėties parametrus, bei $\pm 1^\circ$ – orientacijos parametrus.

Taikant Foxlin, Harrington (2000) pasiūlytą metodą vaizdo kameros nenaudojamos. Matuojama naudojant jutiklius. Sistemai įgyvendinti naudojami trys mikrogroskopai, trys magnetometrai ir du akselerometrai. Metodas leidžia gauti tik galvos orientacijos parametrus. Parametru apskaičiavimo neapibrėžtis yra $\pm 0,02^\circ$.



1 pav. Galvos padėčiai nustatyti matuojami šeši parametrai: galvos poslinkio x , y , z ir galvos orientacijos α , β , δ

Fig. 1. Position of the head position is measured by 6 parameters: the head shift x , y , z and orientation α , β , δ

Pagal Gökürk, Tomasi (2004) pasiūlytą metodą (angl. *Light source*) naudojama šviesai jautri kamera. Galvos padėčiai erdvėje nustatyti yra fiksuojamas šviesos atspindys nuo stebimo objekto. Stebimas objektas yra vartotojo galva. Ši sistema randa tik galvos padėties parametrus. Gautoji neapibrėžtis yra ± 1 vaizdo taškas autorių naudotame ekrane. Sistema yra nestabili ir parametrai fiksuojami ne visą laiką. Remiantis pateiktu straipsniu, parametrai fiksuojami tik 70 % viso stebimo laiko.

Remiantis Basu *et al.* (1996) pasiūlytu metodu (angl. *3D ellipse*) naudojama viena vaizdo kamera. Ji stebi vartotoją. Iš gaunamo vaizdo išskiriami charakteringi taškai ir pagal juos pritaikomas 3D elipsės metodas. Šiuo metodu randami visi šeši ieškomi parametrai, tačiau juos nustatomi netiksliai. Gautosios neapibrėžtys: orientacijos parametrai yra $\pm 10^\circ$, apie padėties parametrus informacijos nėra pateikta.

Taikant Cootes *et al.* (2001) metodą (angl. *Appearance algorithm*) taip pat naudojama viena vaizdo kamera. Vartotojas stebimas iš šalies ir pagal charakteringus taškus sudaromas trimatis veido modelis. Metodo išskirtinumas – fiksuojami tik sklandūs ir lėti judesiai. Jei charakteringi taškai pranyksta iš interesų regiono, tuomet algoritmas vykdomas iš naujo. Apie sistemos neapibrėžtis duomenų nėra pateikta.

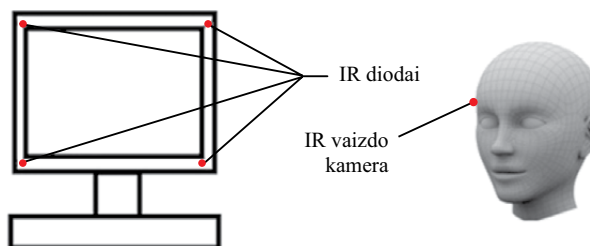
Pagal La Cascia *et al.* (2000) pasiūlytą metodą (angl. *Cylinder texture map*) naudojama viena vaizdo kamera. Stebimas vaizdas ir ieškomas vartotojo veidas. Radus veidą sukuriamas cilindras su veido tekstūra. Iš cilindro pokyčių galima nustatyti ir galvos padėties pokyčius. Metodą pasiūlę mokslininkai tikėjosi, kad jis bus spartus ir tikslus. Šios sistemos neapibrėžtys: padėties parametrai ± 75 mm, orientacijos parametrai $\pm 5^\circ$.

Remiantis Satoh *et al.* (2004) pasiūlytu metodu (angl. *Birds eye view*) naudojama viena vaizdo kamera ir objektas stebimas iš viršaus, o ne iš priekio, kaip buvo taikant minėtuose metoduose. Be to, naudojamas giroskopas orientacijos parametrui nustatyti. Ant vartotojo galvos uždedamas žinomų matmenų žymeklis. Vaizdų apdorojimo programa nustato tikslią galvos padėtį stebėdama žymeklį. Šios sistemos neapibrėžtys: padėties parametrai ± 5 vaizdo taškai, orientacijos parametrai $\pm 1^\circ$.

Keturkampio metodas

Metodui įgyvendinti reikalinga techninė įranga: kompiuteris, infraraudoniesiems spinduliams (IR) jautri kamera, IR filtras, vaizduoklis (4:3 17 colių) ar jo maketas su 4 IR diodais jo kampuose, keturių diodų maitinimo šaltinis (2 pav.). Metodui įgyvendinti reikia sukurti algoritmus keturiems taškams išskirti, keturių taškų parametrus apskaičiuoti. Algoritmas įgyvendinamas Matlab ir C# aplinkoje.

Sukonstruotas vaizduoklio maketas su keturiais IR diodais jo kampuose. Vaizduoklio parametrai apibūdina realaus vaizduoklio dydį. Bandymams pasirinktas 3:4 santykio 17 colių įstrižainės vaizduoklis. Jo matmenys: 34,5×24,9 cm. IR diodai maitinami baterija. 3 pav. pateikta IR diodų grandinės sujungimo schema.

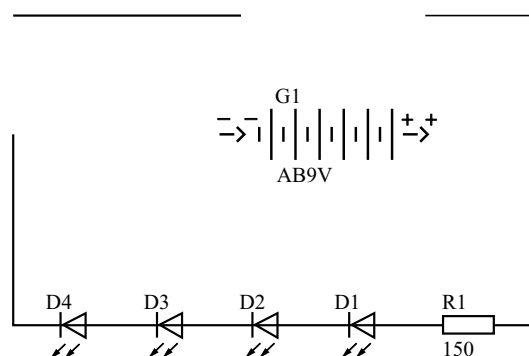


2 pav. Sistemos schema

Fig. 2. System diagram

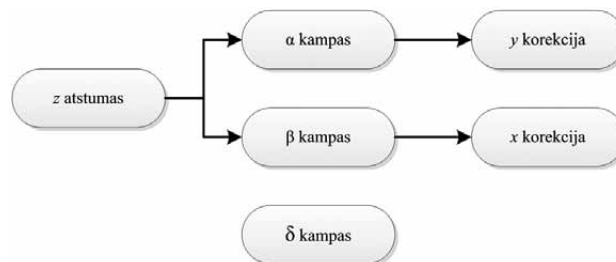
Tyrimo metu nustatyta, kaip kinta keturkampis, stebimas iš įvairių padėčių. Tyrimai vykdyti generuojant daug vaizdų. Kiekviename vaizde buvo tiriami keturkampio kraštinių, kampų, perimetro ir ploto pokyčiai. Nustatyta, kad vieni parametrai veikia kitus. Siekiant išvengti netikslumų, nustatyta parametrų radimo hierarchija.

4 pav. pateikta parametrų nustatymo tvarka. Pirmiausia apskaičiuojamas z parametras. Parametras tiesiogiai veikia α , β , x ir y parametrus. Be to, pastebėta, kad α ir β kampai atitinkamai veikia x ir y koordinates. δ kampo nėra veikiamas jokių kitų parametrų, todėl skaičiuojamas neatsižvelgiant į kitus parametrus.



3 pav. Keturių diodų elektroninė jungimo schema

Fig. 3. 4 LED connection circuit



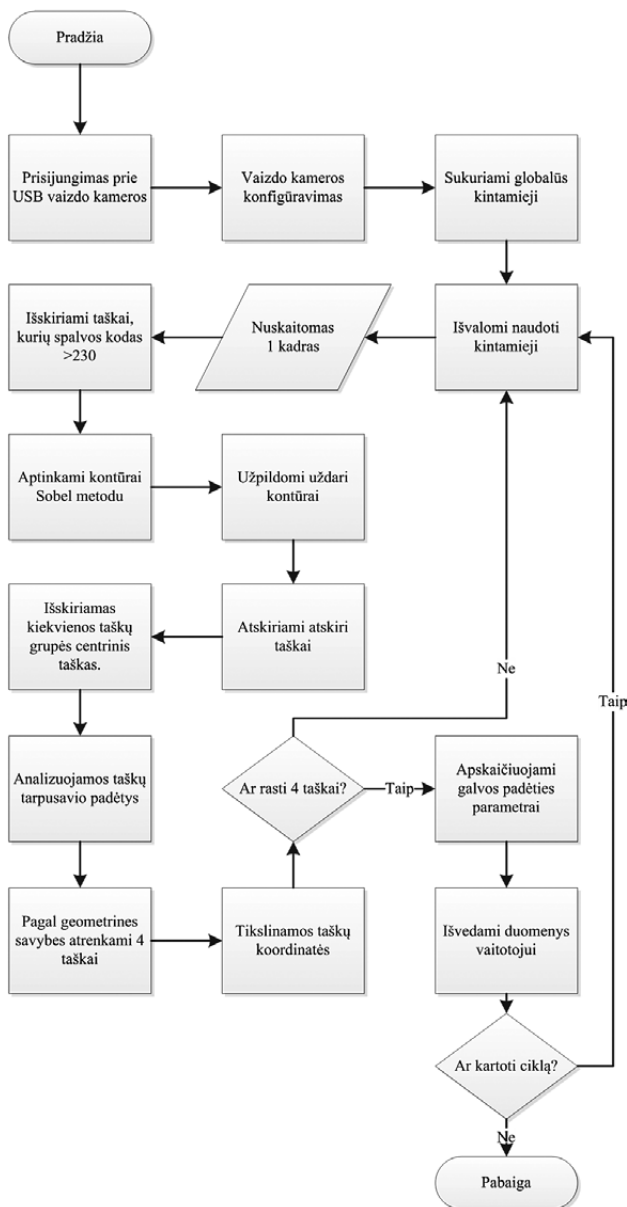
4 pav. Galvos padėties parametrų priklausomybės

Fig. 4. The head position parameter dependence

Algoritmas

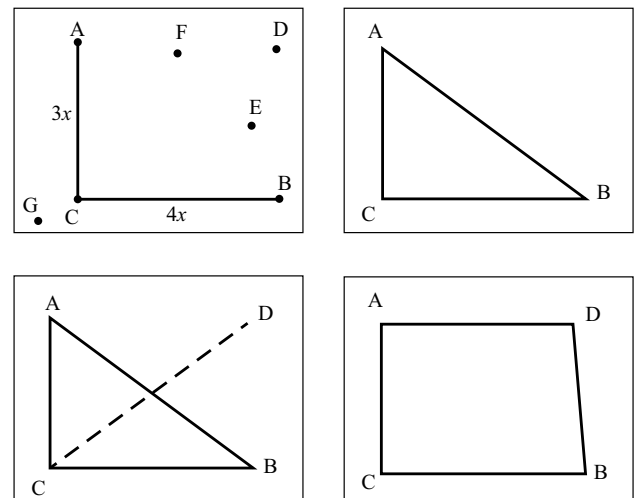
Vaizdų apdorojimo programa buvo kuriama Matlab ir C# aplinkoje (5 pav.). Programos struktūrą galima padalyti į kelias dalis: vaizdo filtravimas, taškų aptikimas ir patikrinimas, ar taškai atitinka keliamus reikalavimus, gautųjų taškų tikslinimas tarpikseliniu metodu ir stebint prieš tai buvusius taškus, parametrų apskaičiavimas pagal nustatytas priklausomybes.

Vaizdo filtravimo dalyje algoritmas panaikina nereikalingus trikdžius. Išvalius trikdžius, Sobel metodu atliekama kontūrų paieška. Randamas centrinis kiekvieno kontūro taškas. Sudaroma rastų taškų aibė.



5 pav. Algoritmo blokinė schema
Fig. 5. Algorithm flowchart

Aptikus keturis taškus tikrinama, ar trys taškai sudaro trikampį. Žinant vaizduoklio kraštinių santykį tikrinama sąlyga, ar trikampio kraštinių santykis yra toks pat – 3:4. Atmetami taškai D, E, F, G (6 pav.). Aptiktas trikampis ABC. Aptikus atkarpas, kurios atitinka pirmąją sąlygą, ieškoma ilgiausia trikampio kraštinė AB. Rastoji kraštinė yra ieškomo keturkampio įstrižainė. Ieškomas ketvirtas keturkampio taškas. Ieškomas taškas, kurio atstumas nuo taško C yra artimiausias įstrižainės AB atstumui. Aptikus ketvirtą tašką tikrinama sąlyga, ar keturkampis yra tinkamas, ir tikrinama, ar visi keturkampio kampai neviršija 100° . Jei sąlyga yra atitinkama, tuomet keturkampis yra tinkamas ir skaičiuoti galima toliau. Jei sąlyga nėra atitinkama, ieškomi kiti taškai, sudarantys trikampį. Jei tinkami taškai nerasti, pereinama prie kito kadro.



6 pav. Taškų filtravimas
Fig. 6. Point filtering

Aptikti taškai surūšiuojami pagal padėtį vaizde. Rūšiuojama tokiu principu:

- 1-o taško x ir y koordinatėjų suma yra mažiausia.
- 4-o taško x ir y koordinatėjų suma yra didžiausia.
- 2-o taško x koordinatė yra artimiausia 4-o taško x koordinatei.
- 3-io taško x koordinatė yra artimiausia 1-o taško x koordinatei.

Aptikus keturis taškus, tikslinamos jų koordinatės skaičiuojant vertes tarp vaizdo taškų. Apskaičiuotos IR telkinių centrinės vertės vaizdo taško tikslumu. Nustatoma dominanti sritis, kurio ribos yra $n \times n$. Dominančioje srityje pagal nustatytą slenkstį ieškoma aštrių ryškumo pokyčių. Tikrinant x kryptimi randamos šviesos diodo spindulio ribos iš kairės X_{i1} ir dešinės X_{i2} . Analogiškai randamos y krypti-

mi – ribos iš viršaus Y_{i1} ir iš apačios Y_{i2} . Vertės tarp vaizdo taškų apskaičiuojamos pagal šias formules:

$$X = \frac{1}{n} \sum_{i=0}^n \frac{X_{i2} - X_{i1}}{2}, \quad (1)$$

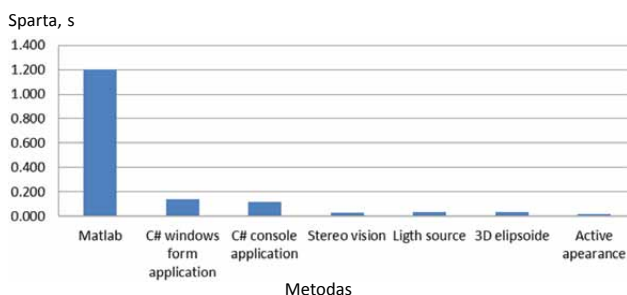
$$Y = \frac{1}{n} \sum_{i=0}^n \frac{Y_{i2} - Y_{i1}}{2}, \quad (2)$$

čia X_{i1} – 1-oji x ribinė vertė; X_{i2} – 2-oji x ribinė vertė; Y_{i1} – 1-oji y ribinė vertė; Y_{i2} – 2-oji y ribinė vertė; n – dominančios srities plotis; X – x vertė tarp vaizdo taškų; Y – y vertė tarp vaizdo taškų.

Parametrų skaičiavimo dalyje apskaičiuojami ieškomi parametrai pagal ankstesnių tyrimų nustatytas priklausomybes. Visi duomenys pateikiami vartotojui grafinėje vartotojo sąsajoje.

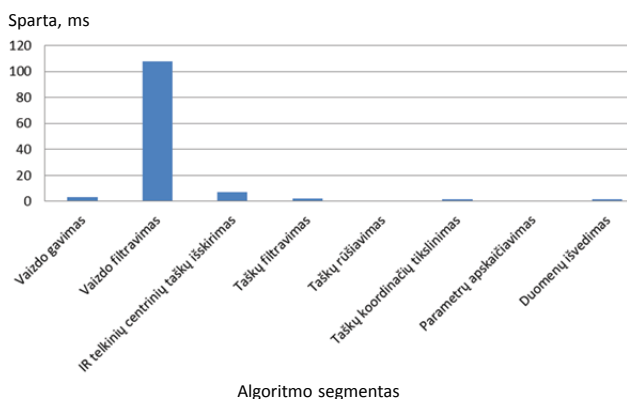
Sistemos spartos tyrimas

Tiriant algoritmo spartą algoritmas buvo vykdomas 100 kartų. Vykdomo laikas buvo fiksuojamas kiekvieno ciklo pabaigoje. Nustatyta, kad algoritmas Matlab aplinkoje vidutiniškai sugaišta 1,2 s, o C# aplinkoje – 126 ms. Gautieji rezultatai palyginti su nagrinėtų sistemų autorių pateiktais duomenimis (7 pav.).



7 pav. Algoritmų vykdymo spartos palyginimas

Fig. 7. Comparison of algorithms for execution speed



8 pav. Algoritmų segmentų vykdymo spartos tyrimas

Fig. 8. Algorithm segment execution speed investigation

Sukurti algoritmai vykdomi lėčiau nei remiantis kitų autorių siūlomais metodais. Siekiant iširti, kurioje algoritmo dalyje sugaištama daugiausia laiko, atliktas atskirų programos segmentų spartos tyrimas. Iš gautų rezultatų pastebėta, kad daugiausia laiko užgaišta filtravimo segmentas (8 pav.).

Sistemos tikslumo tyrimas

Tyrimų metu vaizdo kamera buvo pritvirtinta stabiliai ir gaunama 100 kadrų. Buvo pastebėta, kad nustatyti parametrai keičiasi nekeičiant kameros padėties. Neapibrėžtis įvertinama apskaičiuojant standartinę nuokrypį iš gautų verčių. Eksperimentas buvo kartojamas keičiant kameros padėtį. Įvertinamos atsitiktinės neapibrėžtys. Maksimali atsitiktinė neapibrėžtis, matuojant galvos pozicijos padėtį, yra 1,63 mm, o matuojant orientacijos parametrus – 0,16°.

Naudojantis sukonstruotu stendu buvo fiksuojami parametrai. Nustatomos parametrų vertės, keičiant atstumą x ir kampą β . Lyginamas realus ir programos apskaičiuotas pokytis. Nustatoma matavimo neapibrėžtis. Matavimo neapibrėžtis yra: 4 mm, matuojant galvos padėties parametrus, ir 0,11°, matuojant orientacijos parametrus.

Iš gautų duomenų galima teigti, kad sistemos tikslumas yra $\pm 0,5^\circ$ galvos orientacijos ir ± 5 mm galvos pozicijos parametrams.

Išvados

1. Parinkti geometriniai keturkampio parametrai, suteikiantys daugiausia informacijos apie kiekvieną iš galvos pozicijos ir orientacijos parametrų.
2. Išanalizavus keturkampio pokyčius keičiant stebėjimo kampą, buvo aptikta, kad z parametras tiesiogiai veikia α , β , x ir y parametrų vertes. α , β atitinkamai veikia x ir y parametrus. Nustatytas parametrų tarpusavio sąryšis ir skaičiavimo eiliškumas.
3. Sukurtas vaizdo apdorojimo algoritmas dviejose skirtingose aplinkose: Matlab ir C#.
4. Iširta algoritmų vykdymo trukmė įgyvendinimo aplinkose. Naudojamo kompiuterio procesorius – Intel Pentium dual core T4500 – 2,3 GHz. Lėčiausiai algoritmas vykdomas Matlab aplinkoje – vidutiniškai 1,2 sekundės. Vidutinis C# algoritmo vykdymo laikas – 126 ms.
5. Iširta algoritmo segmentų vykdymo sparta. Pastebėta, kad ilgiausiai vykdomas filtravimo segmentas – vidutiniškai 107 ms. Visi kiti segmentai įvykdomi greičiau nei per 5 ms. Optimizavus filtravimo procesą algoritmą galima paspartinti keletą kartų.

6. Parodyta, kad algoritmo tikslumo įvertinimą galima suskirstyti į statines ir matavimo neapibrėžtis.
7. Maksimali atsitiktinė neapibrėžtis, matuojant galvos pozicijos padėtį, 1,63 mm, o matuojant orientacijos parametrus – 0,16°.
8. Maksimali matavimo neapibrėžtis – 4 mm, matuojant galvos padėties parametrus, ir 0,11° – orientacijos parametrus.
9. Iš gautų duomenų galima teigti, kad sistemos tikslumas yra $\pm 0,5^\circ$ galvos orientacijos ir ± 5 mm galvos pozicijos parametrų.

Literatūra

- Basu, S.; Essa, I.; Pentland, A. 1996. Motion Regularization for Model-Based Head Tracking, in *Proceedings of the 13th IEEE Int'l Conf. on Pattern Recognition (ICPR '96)*: 611–616. <http://dx.doi.org/10.1109/ICPR.1996.547019>
- Cootes, T. F.; Edwards, J. G.; Taylor, J. C. 2001. Active Appearance Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6): 681–685. <http://dx.doi.org/10.1109/34.927467>
- Foxlin, E.; Harrington, M. 2000. Wear Track: A Self-Referenced Head and Hand Tracker for Wearable Computers and Portable VR, in *Proceedings of International Symposium on Wearable Computers (ISWC 2000)*: 155–162.
- Göktürk, B. S.; Tomasi, C. 2004. 3D Head Tracking Based on Recognition and Interpolation Using a Time-Of-Flight Depth Sensor, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)* 2: 211–217.
- La Cascia, M.; Sclaroff, S.; Athitsos, V. 2000. Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(4): 322–336.
- Matsumoto, Y.; Zelinsky, A. 2000. An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement, in *Proceedings of IEEE Fourth International Conference on Face and Gesture Recognition (FG '2000)*: 499–505. <http://dx.doi.org/10.1109/34.845375>
- Satoh, K.; Uchiyama, S.; Yamamoto, H. 2004. A Head Tracking Method Using Bird's-Eye View Camera and Gyroscope, in *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*: 202–211. <http://doi.ieeecomputersociety.org/10.1109/ISMAR.2004.3>

are found. The diagram of the system and detail block diagram of the algorithm were provided. Realization of the algorithm used: edge detection method (Sobel), the adjustment algorithm (Subpixel). System is realized in Matlab and C# environment. Determine the optimal parameters for the algorithm execution. Execution of the algorithm in Matlab environment is 1.2 s and C # environment – 126 ms. During the examination of the longest executing algorithm segment, it was found that image filtering is carried out in 107 ms. It is noted that the uncertainty of the algorithm can be divided into static and measurement. The maximum static uncertainty while measuring head position parameters is 1.63 mm and orientation parameters – 0.16°. The maximum measurement uncertainty while measuring head position parameters is 4 mm and orientation parameters – 0.11°.

Keywords: image processing, computer vision, positioning the head position.

TRACKING OF HEAD POSITION RELATIVE TO THE SCREEN USING HEAD MOUNTED CAMERA

E. Borcovas, G. Daunys

Abstract

In this paper head position locating systems were analyzed. There were reviewed scientific articles with different proposed methods. The chosen system is with the camera located on the user head. The main parameters of the head positioning systems were analyzed. The procedure laid down in what order parameters