

MODIFIKUOTO POŽYMIŲ VAIZDE IŠSKYRIMO SURF ALGORITMO OBJEKTUI SEKTI REALIUOJU LAIKU ĮGYVENDINIMAS LAUKU PROGRAMUOJAMOJE LOGINĖJE MATRICOJE

Tomyslav Sledevič

Vilniaus Gedimino technikos universitetas
El. paštas tomyslav.sledevic@vgtu.lt

Santrauka. Pateikiamas modifikuoto požymių vaizde išskyrimo algoritmo SURF įgyvendinimas lauku programuojamų loginių matricių (LPLM) įrenginiuose. LPLM įrenginiai pasirinkti dėl galimybės tuo pat metu įgyvendinti veikiančius procesus taikant VHDL kalbą. Tai garantuoja, kad požymiai vaizde bus išskirti realiuoju laiku. Skaičiavimams paspartinti taikomas slankusis 84×84 taškų dydžio langas, kuriame saugomas sudėtingas vaizdas. Šio slankiojo lango duomenys taikomi *Hessian* determinantui, būdingųjų taškų orientacijai ir deskriptoriams apskaičiuoti. Požymiai ieškomi aštuoniose skalėse taikant lokalių ekstremumų paiešką. Požymių orientacijos vektorius ir supaprastintas deskriptorius skaičiuojami šešiose skalėse tuo pat metu. Algoritmo veikimas tiriamas sekant keturių taškų žymeklį ir pagal jį braižant plokštumą arba erdvinį kubą. Skaičiuojama 25 MHz taktiniu dažniu. Vaizdai gauti taikoma 60 kadro per sekundę dažnio 640×480 taškų raiškos vaizdo kamera.

Reikšminiai žodžiai: lauku programuojama loginė matrica, požymių išskyrimas, vaizdo filtravimas, objekto sekimas.

Įvadas

SURF (angl. *Speeded up Robust Features*) algoritmas skirtas būdingiesiems taškams vaizde išskirti, koduoti ir sutapdinti, neatsižvelgiant į požymių skalę ir posūkio kampo pasikeitimus (Bay *et al.* 2008). Algoritmas plačiai taikomas objektams vaizde sekti, papildytos realybės programose, trimačiams vaizdams rekonstruoti. Procesoriniuose įrenginiuose įgyvendinto SURF algoritmo greitaveika priklauso nuo dvimačių filtrų skaičiaus, aptiktų požymių ir jiems taikomų posūkio kampų skaičiaus bei deskriptorių dydžio. Dėl algoritmo sudėtingumo net šiuolaikiniai procesoriai negarantuoja stabilaus didelės raiškos ir didelio dažnio vaizdo kadro apdorojimo.

Šiai problemai išspręsti siūloma SURF algoritmą įgyvendinti LPLM. Jose yra galimybė kurti vienu metu veikiančius filtrus. Filtrai gali turėti tūkstančius elementų, kuriuos įmanoma sinchroniškai su kiekvienu taktiniu signalu adresuoti. Lygiagretus procesų paskirstymas LPLM garantuoja vaizdo apdorojimą realiuoju laiku, taktavimo signalui esant kelių dešimčių megahercų eilės ir sunaudojant mažiau galios, palyginti su procesoriais.

LPLM įgyvendintas *Hessian* matricos skaičiavimas taikant kelis algoritmo supaprastinimus dėl ribotų aparatinių išteklių (Svab *et al.* 2009). Požymių paieškos etapas perkeltas į kompiuterį. Nustatyta, kad algoritmas teisingai aptinka požymius 1024×768 raiškos vaizde 10 kadro per sekundę dažniu.

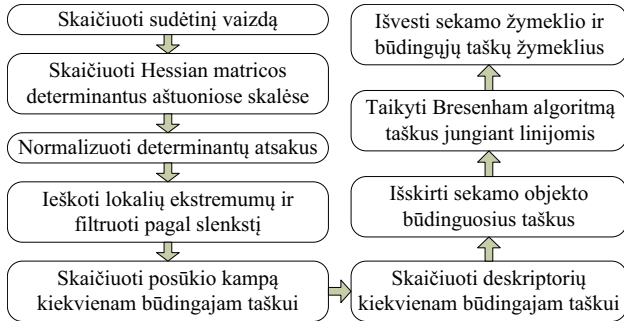
LPLM įgyvendintas SURF algoritmas iki posūkio kampų požymiams skaičiavimo etapo imtinai (Bouris *et al.* 2010). Autorių siūlomas algoritmas sunaudoja didesnę dalį BRAM bloką visiems vieno kadro sudėtiniais taškams saugoti. Pasiiektas 56 kadro per sekundę dažnio būdingųjų taškų išskyrimas 640×480 raiškos vaizde taikant 200 MHz taktinį signalą. Pasiiektas 406 kadro per sekundę dažnio deskriptorių požymiams skaičiavimas 640×480 raiškos vaizde, tačiau tik vienoje skalėje (Fisher *et al.* 2011). Vaizdai siunčiami iš kompiuterio LPLM įrenginiui, ir kiekvieno taktinio signalo metu atgal grąžinamos deskriptorių vertės. Požymiams koduoti autoriai taiko posūkiui atsparų O-DAISY deskriptorių. Siūlomas algoritmas veikia 125 MHz dažniu, sunaudojama 353 BRAM ir 403 DSP celių.

SURF algoritmas įgyvendintas dviejose LPLM sunaudojant 90 % atminties išteklių (Battezzati *et al.* 2012). Nustatyta, kad požymių kadre sutapdinimo etapas apima daugiausia skaičiavimų, sunaudoja 49 % LUT (angl. *Look-up Table*) ir užtrunka 355 ms. Algoritmas gali aptikti iki 4096 požymių 1024×1024 raiškos vaizde.

Požymių išskyrimo ir linijų braižymo algoritmai

Apibendrintas objekto sekimo pagal būdinguosius taškus algoritmas SURF pateiktas 1 pav. Skaičiuojamas sudėtinis vaizdas, *Hessian* matricos determinantai vienu metu aštuo-

niose skalėse. Kuo didesnė skalė, tuo didesnė determinanto vertė, todėl determinantų atsakus būtina normuoti. Tarp trijų gretimų skalių ieškoma lokalių ekstremumų – tai determinantų vertės, kurios didesnės už kaimyninius taškus. Stipriausiems požymiams išskirti ekstremumai filtruojami pagal slenkstį. Kiekvienam požymiui skaičiuojami posūčio kampai ir deskriptoriai. Taikant Bresenham linijų braižymo algoritmą pagal išskirtus sekamo objekto požymius braižoma plokštuma arba erdvinis kubas. Monitoriuje pateikiami sekamo žymeklio ir būdingųjų taškų žymekliai.



1 pav. Būdingųjų taškų išskyrimo ir objekto sekimo algoritmas
Fig. 1. Features extraction and object tracking algorithm

Sudėtinis vaizdas sudaromas ir taikomas pagal lygtis:

$$I_{\Sigma}(x, y) = \sum_{x=1}^X \sum_{y=1}^Y I(x, y), \quad (1)$$

$$I_{\Sigma} = I_{\Sigma}(x_a, y_a) + I_{\Sigma}(x_d, y_d) - I_{\Sigma}(x_b, y_b) - I_{\Sigma}(x_c, y_c), \quad (2)$$

čia $I(x, y)$ – taško skaisčio vertė; $I_{\Sigma}(x, y)$ – sudėtinio vaizdo taško vertė; I_{Σ} – sudėtinio vaizdo taškų verčių suma ribojamame stačiakampio $abcd$ plote. Sudėtinio vaizdo taikymo pranašumas tas, kad vienodai greitai įvertinama taškų skaisčio suma bet kokio ploto stačiakampyje. Ši savybė spartina kelių skalių ir krypčių 2D antrosios eilės Gauso filtrų ir Haar vilnelių taikymą.

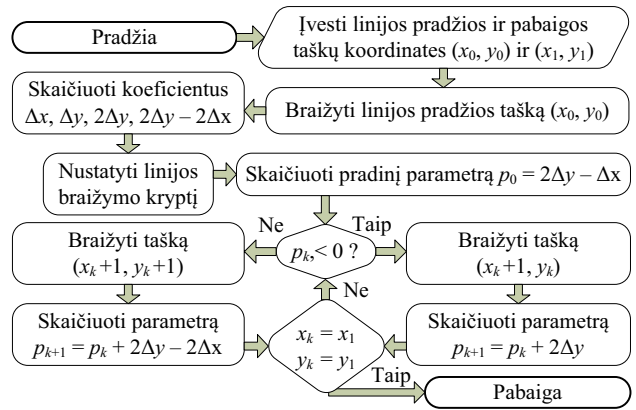
Hessiano determinantas skaičiuojamas pagal lygtis:

$$H(x, y) = \det \begin{bmatrix} A & C \\ C & B \end{bmatrix} = \det \begin{bmatrix} \partial^2 f / \partial x^2 & \partial^2 f / \partial x \partial y \\ \partial^2 f / \partial x \partial y & \partial^2 f / \partial y^2 \end{bmatrix}, \quad (3)$$

$$H(x, y) = AB - w^2 C^2, \quad (4)$$

čia w^2 – svorio koeficientas, kuris lygus 0,83 (Bay *et al.* 2008). Jis taikomas kompensuoti tolydžios funkcijos Gauso filtrų aproksimavimą stačiakampių formos filtrais. Aparatūroje kur kas greičiau vykdomos ne sandaugų iš slankiojo kabelio koeficientų, bet sumavimo ir postūmio operacijos. Todėl algoritme padarytas pakeitimas: $w^2 = 1 - 1/8 = 0,875$.

Linijoms braižyti taikomas Bresenham algoritmas (2 pav.), kuriam įgyvendinti naudojamos tik sumos, skirtumo ir palyginimo operacijos (Sunkara 2011). Algoritmo veikimas nuoseklus, nes kito linijos taško koordinatėms apskaičiuoti būtina žinoti esamo taško parametą p_k . Pagal $\Delta x, \Delta y$ vertes nustatoma linijos braižymo kryptis.

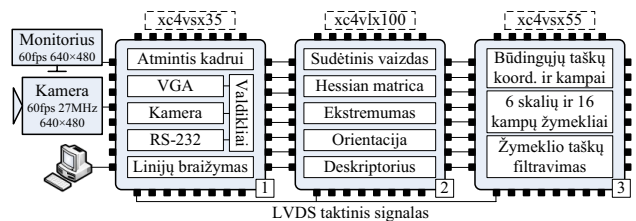


2 pav. Linijų braižymo Bresenham algoritmas
Fig. 2. Bresenham's line drawing algorithm

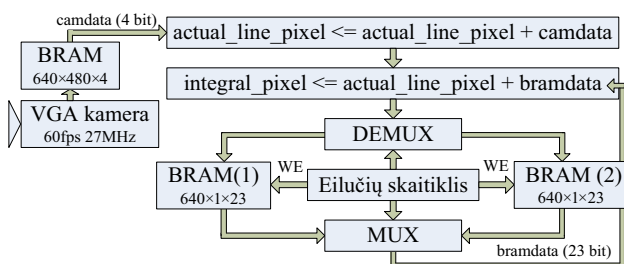
Bresenham algoritmas linijas braižo tik pirmame koordinatinių sistemos oktante, todėl būtina sukeisti koordinatinių ašis, kai linijos vektorius nepatenka į 0–45°. Parametras p_k nurodo, kur braižyti kitą tašką – $(x_k + 1, y_k)$ ar $(x_k + 1, y_k + 1)$ koordinatėse – ir koks bus kitas p_{k+1} parametras. Linija braižoma tol, kol taško (x_k, y_k) koordinatės nesutampa su pabaigos tašku (x_1, y_1) .

SURF algoritmo įgyvendinimas LPLM

SURF algoritmas įgyvendintas trijose LPLM (3 pav.). Valdymo įrenginiu laikoma xc4vsx35, pavaldžiais – xc4vlx100 ir xc4vsx55 matricos. Pirmojoje LPLM yra kadro atminties ir periferinių sąsajų valdikliai, plokštumos ir erdvinio kubo kūrimo algoritmai. Antrojoje LPLM įgyvendintas SURF algoritmas. Trečiojoje LPLM išskiriami žymeklio taškai, saugomi būdingųjų taškų koordinatės ir kampai, kurie susieti su šešių skalių ir šešiolikos kampų žymekliais. Antroji ir trečioji LPLM taktuojamos LVDS



3 pav. Algoritmo paskirstymas tarp trijų LPLM
Fig. 3. Distribution of algorithm in three FPGA



4 pav. Sudėtinio vaizdo sudarymo struktūrinė diagrama
Fig. 4. Structure diagram of integral image calculation

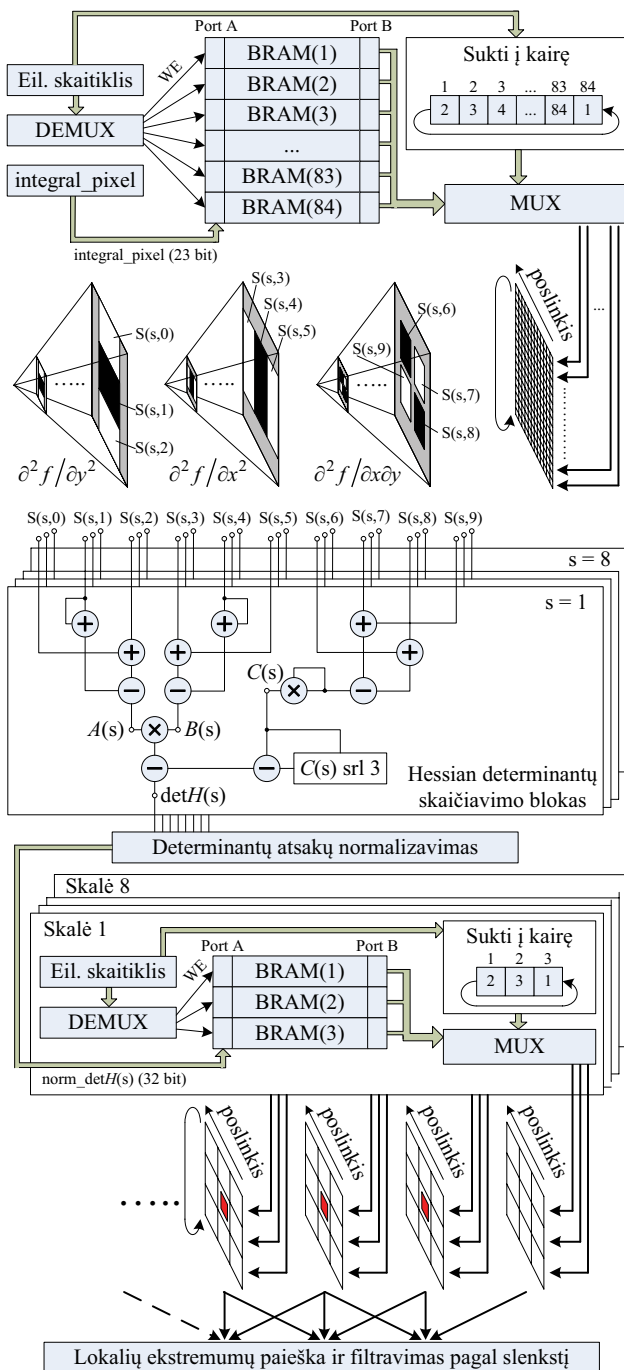
(angl. *Low Voltage Differential Signal*) signalu. Algoritmu veikimas sinchronizuojamas vienalaikiu skaitiklių, esančių trijose LPLM, numetimu.

Sudėtiniam vaizdai (angl. *Integral Image*) sudaryti kiekvieno 25 MHz taktinio signalo metu taikomos dvi sumavimo operacijos (4 pav.). Pirmosios sumos operacijos metu sudedamos vienoje eilutėje esančių taškų vertės, antrosios metu – viename stulpelyje esančių taškų vertės. Kol BRAM(1) atmintyje saugoma viena paskutinė sudėtinio vaizdo eilutė, į BRAM(2) įrašoma nauja.

Naujam 84×84 dydžio langui sukurti (5 pav.) taikomos 84 BRAM eilutės sudėtiniam vaizdai saugoti (šis eilučių skaičius būtinas septintosios skalės supaprastintam deskriptoriui skaičiuoti). BRAM atmintys kas 84 eilutes periodiškai perrašomos, todėl saugomi tik realioju laiku apdoroti būtini duomenys taikant ribotą pastovų atminties kiekį. Duomenis slankiajame lange būtina ne tik stumti į kairę, bet ir per vieną elementą kelti į viršų eilutei pasibaigus. Šiam uždaviniui išspręsti taikomas į kairę besisukantis eilučių indeksų vektorius. Slankiojo lango duomenimis dalijamasi su Hessian determinanto, požymių posūkio kampų ir deskriptorių blokais.

Žymenis $S(s, 0..9)$ nurodo suma vaizdo taškų, patenkančių į stačiakampių ar kvadratų ribojamą plotą. Kiekvieno taktinio signalo metu 80 sumos verčių (10 kiekvienai skalėi) perduodama į Hessian matricos determinanto skaičiavimo bloką. Aštuoni kiekvienos skalės determinanto rezultatai siunčiami į determinanto atsako normalizavimo bloką, kuriame kiekvienas aukštesnė už pirmąją skalę atitinkantis rezultatas dauginamas iš skalės koeficiento. Normuotieji atsakai taikomi lokalių ekstremumų paieškai ir filtravimui pagal slenksčių.

Statinį slenksčių patogu taikyti, kai žinoma, kad apšvietimas vaizde nekinta. Kitu atveju siūloma taikyti adaptyvų slenksčių, kuris priklausytų nuo vaizde aptiktų požymių ir slenksčio vertės praėjusiam kadre. Slenksčio vertė kitam kadrui gali būti išreiškiama šia lygtimi:



5 pav. Slankiojo lango kūrimas 2D Gauso filtrams taikyti, determinanto skaičiavimas ir lokalių ekstremumų paieška
Fig. 5. Creation a sliding window for 2D Gaussian filters, determinant calculation and local extreme point search

$$TH_{next} = TH_{prev} + (N_{prev} - N_d) \cdot v_{th}, \quad (5)$$

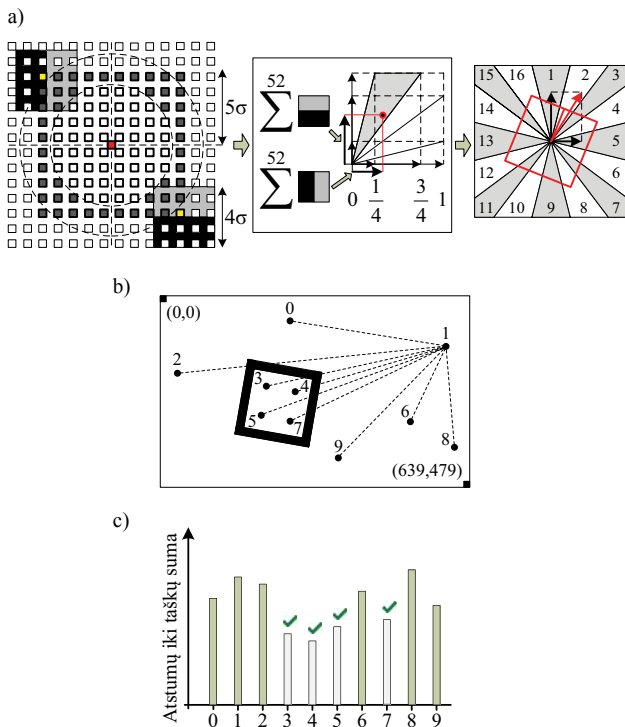
čia TH_{next} – slenksčio vertė kitame kadre; TH_{prev} – esama kadre; N_{prev} – būdingųjų taškų skaičius esamame kadre; N_d – ribinis būdingųjų taškų skaičius; v_{th} – adaptyvaus slenksčio prisitaikymo greitis. Tam, kad požymiai, kurių

determinanto vertės mažos (taškai išsidėstę išilgai įstrižos linijos ir požymius dubliuojantys atsakai), būtų atmetami, adaptyvusis slenkstis turi apatinę ribą.

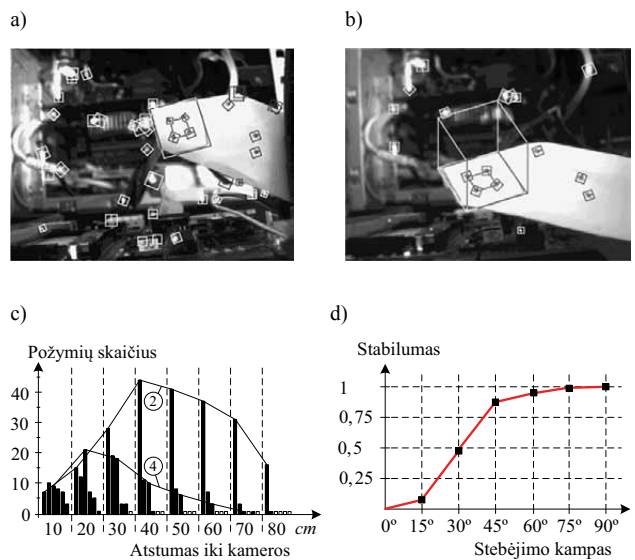
Požymių orientacijai nustatyti aplink būdingąjį tašką atstumu 4σ apskaičiuotos 52 Haar dvimatės vilnelės x ir y kryptimis sumuojamos sudarant du vienas kitam stačius vektorius (6 pav., a).

Taikant bitų postūmio ir sumos operacijas vertinami šių vektorių tarpusavio ilgiai. Požymiui gali būti priskirta viena iš šešiolikos kampų orientacija. Būdingasis taškas gali būti pažymimas kvadrato formos žymekliu, kurių yra po 16 kiekvienai iš 6 skalių. Deskriptorius sudaromas keturioms $5\sigma \times 5\sigma$ sritims skaičiuojant gradientų bei jų modulių sumas pagal x ir y koordinates. Vaizde gali būti aptinkami identiški žymekliui būdingieji taškai. Todėl jiems atmeti skaičiuojamos atstumų iki kitų požymių sumos (6 pav., c). Požymiai, kuriuos atitinka keturios mažiausios vertės sumos, priskiriami žymekliui.

Iš aptiktų būdingųjų taškų (7 pav., a, b) skaičiaus vaizde priklausomybės nuo atstumo iki kameros (7 pav., c) matyti, kad tolinant kamerą nuo objekto požymiai iš aukštesnių skalių pereina į žemesnes. Iš keturių žymeklio taškų sekimo stabilumo grafiko (7 pav., d) matyti, kad plokštumos stebėjimo kampui esant iki 45° objektas sekamas 80 % stabilumu.



6 pav. Būdingųjų taškų posūkio skaičiavimas (a); aptikti žymeklio taškai (b); atstumų iki kitų požymių histograma (c)
Fig. 6. Orientation vector estimation technique (a); marker's points detected (b); distances between features and markers (c)



7 pav. Aptikti būdingieji taškai (a, b); požymių skaičiaus priklausomybė (c); žymeklio sekimo stabilumo pokytis (d)
Fig. 7. Detected features (a, b); features dependence on camera distance (c); Marker tracking stability (d)

Išvados

1. Būdingieji taškai vaizde aptinkami ir sekami teisingai keičiant posūkio kampą ir atstumą iki jų. Taikant keturių mažiausių atstumų sumų tarp požymių paiešką, sekamo žymeklio taškai išskiriami teisingai, kai kampas tarp plokštumos ir kameros ašies kuo artimesnis 90° .
2. Modifikuotas SURF algoritmas sėkmingai įgyvendintas trijose LPLM ir veikia realiuoju laiku taikant tik 25 MHz taktinį signalą. Algoritmas gali būti greitai pritaikytas didesnės raiškos ir dažnio vaizdai apdoroti.

Literatūra

- Battezzati, N., *et al.* 2012. SURF Algorithm in FPGA: A Novel Architecture for High Demanding Industrial Applications, in *DATE Conference & Exhibition*: 161–162.
- Bay, H., *et al.* 2008. SURF: Speeded up Robust Features, *Computer Vision and Image Understanding* 110(3): 346–359. <http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- Bouris, D., *et al.* 2010. Fast and Efficient FPGA-based Feature Detection Employing the SURF Algorithm, in *18th IEEE Annual International Symposium on FCCM*: 3–10.
- Fischer, J., *et al.* 2011. A Rotation Invariant Feature Descriptor O-DAISY and its FPGA Implementation, in *IEEE International Conference on IROS*: 2365–2370.
- Sunkara, P. 2011. Line Drawing by using Bresenham's Algorithm, Wu's Algorithm, *Indiana State University*: 1–10.
- Svab, J., *et al.* 2009. FPGA-based Speeded up Robust Features, in *IEEE International Conference on TePRA*: 35–41.

MODIFIED SURF ALGORITHM IMPLEMENTATION ON FPGA FOR REAL-TIME OBJECT TRACKING

T. Sledevič

Abstract

The paper describes the FPGA-based implementation of the modified speeded-up robust features (SURF) algorithm. FPGA was selected for parallel process implementation using VHDL to ensure features extraction in real-time. A sliding 84×84 size window was used to store integral pixels and accelerate Hessian determinant calculation, orientation assignment and descriptor estimation. The local extreme searching was used to find point of interest in 8 scales. The simplified descriptor and orientation vector were calculated in parallel in 6 scales. The algorithm was investigated by tracking marker and drawing a plane or cube. All parts of algorithm worked on 25 MHz clock. The video stream was generated using 60 fps and 640×480 pixel camera.

Keywords: field programmable gate array, features extraction, video filtering, object tracking.