

Jūros dugno mozaikose aptinkamų objektų atpažinimo modeliavimas

Olegas Ramašauskas^{1,2}, Vytautas Janulevičius¹,
Aurimas Vaitkus¹, Mantas Kazlauskas¹

¹*Klaipėdos universitetas, Gamtos ir matematikos mokslų fakultetas*

H. Manto g. 84, LT-92294 Klaipėda

²*Lietuvos verslo kolegija*

Turgaus g. 21, LT-91249, Klaipėda

E. paštas: olegas.ramasauskas@ku.lt

Santrauka. Pateikiamos Klaipėdos universiteto Jūros mokslų ir technologijų centre vystomos priemonės jūros dugno skaitmeninių vaizdų tyrimui, objektų išskyrimui heterogeniškoje aplinkoje ir rezultatų įvertinimui. Darbe aprašomi povandeninių robotų pagalba gautų vaizdų gerinimo, povandeninio vaizdo mozaikų tekstūrų ir spalvų aprašymo, sričių segmentavimo ir filtravimo algoritmai bei jų programiniai komponentai.

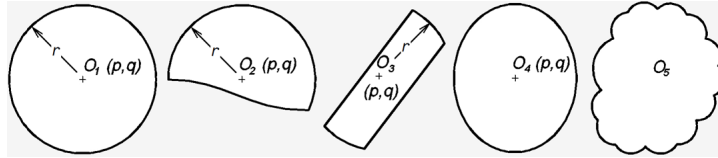
Raktiniai žodžiai: skaitmeninis vaizdas, mozaika, filtravimas, atpažinimas.

Įvadas

Įvairiarūšių (heterogeninių) jūros dugno struktūrų ir objektų tyrimuose dažniausiai naudojami mobilūs nuotoliniu būdu valdomi povandeniniai robotai (ROV, iš angl. *Remotely Operated Vehicle*) su įmontuotomis vaizdo kameromis, apšvietimo, atstumo matavimo ir pozicionavimo įranga [4]. Akivaizdu, kad įrašyta vaizdinė medžiaga yra labai svarbi priemonė jūriniuose tyrimuose [2]. Tačiau modernia įranga išgaunami didžiuliai informacijos kiekiai apsunkina šios medžiagos peržiūrą bei analizę įprastais metodais, o apdorotą – keblu panaudoti rutininiam aplinkos monitoringui [6]. Metodai, leidžiantys procesą automatizuoti ir apdoroti didelius vaizdų kiekius realaus laiko sąlygomis, registruoti kiekybinius bei kokybinius pokyčius jūros dugne – labai reikalingi ir laukiami [5]. Straipsnyje modeliuojamos jūros dugno vaizdo mozaikų apdorojimo problemos, aprašomi tyrimai ir rezultatai, parodyti daliniai sprendimai Matlab [1] sistemos priemonėmis. Tikslui pasiekti taip pat buvo sudaryti ir (arba) pritaikyti dugno nuotraukų segmentacijos algoritmai, vaizdo mozaikų apdorojimo scenarijai, Matlab funkcijos ir programos, kuriomis apdorojus pirminius ROV vizualinius duomenis, galima pagerinti jų vientisumą, objektų identifikacijos spartą ir atpažinimo tikslumą, lyginant su tradiciniais sprendimais, tuo pačiu metu atliekant kitų skaitmeninio vaizdo parametrų korekciją.

1 Heterogeninių struktūrų atpažinimo problemos

Atliekant paveikslo elemento atpažinimą, t. y. atitikimą vaizduojamo objekto su aprašymu skaitmeninių duomenų bazėje, pirmiausia reikia išspręsti panašumo problemą.



1 pav. Supaprastinti realių objektų modeliai formalių išraiškų testavimui.

Tai sąvoka, kurią gana keblu apibrėžti kiekybiniais metodais. Tarkime, kad jūros dugno mozaikos paveiksle žmogaus akis stebi II-os eilės kreivėmis aprašomus, vaizdavimo plokštumoje projektuojamus objektus kaip trimačių kūnų kirtinius ar jų dalis, 1 pav., kurių centras O yra per (p, q) nutolęs nuo gautojo skaitmeninio vaizdo koordinatžių pradžios, bendruoju pavidalu $a_{11}(x-p)^2 + 2a_{12}(x-p)(y-q) + a_{22}(y-q)^2 + 2a_{13}(x-p) + 2a_{23}(y-q) + a_{33} = 0$, čia $a_{ik} = a_{ki}$, $i, k = 1, 2, 3$. Apskaičiavę kreivių lygtyse koeficientų matricių invariantus ir, susieję su centro koordinatėmis, apibrėžtume visų įmanomų II-os eilės kreivių savybes, nepriklausomai nuo jų padėties vaizdavimo plokštumoje: centruojamų ($D \neq 0$; $D > 0$), neišsigimusių ($A \neq 0$) uždarų kreivių: apskritimų, ovalų, elipsių ir panašių, kurioms $I = a_{11} + a_{22}$, $A/I < 0$ [3]

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \quad A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix},$$

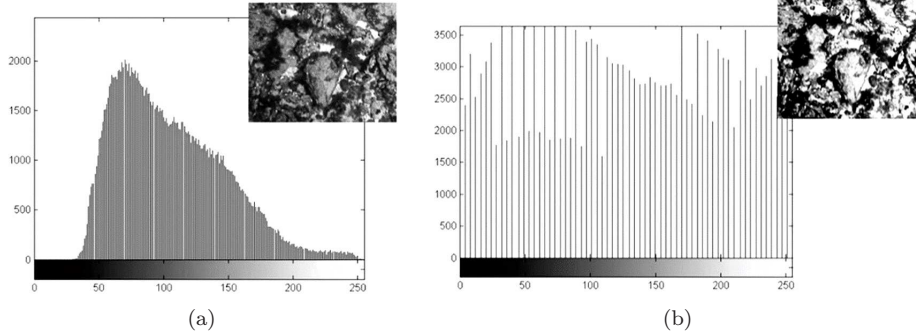
$$p = -\frac{1}{D} \begin{vmatrix} a_{13} & a_{12} \\ a_{23} & a_{22} \end{vmatrix}, \quad q = -\frac{1}{D} \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}.$$

Parinkę parametrų $p = 0$ ir $q = 0$ reikšmes, visus objektų taškus apskaičiuotume kompiuteriu, sveikaisiais išreiškę visus vaizdo pikselių koordinatžių rinkinius. Palyginę gautųjų vaizdo pikselių koordinatžių reikšmes su duomenų bazėje saugomais kitų objektų kanonizuotais aprašymais ir nustatę sutapimą, galėtume identifikuoti stebimą objektą.

Iš pirmo žvilgsnio tai trivialus uždavinys, tačiau dugno mozaikose užfiksuotų objektų panašumo sprendimo keblumus gali atskleisti pavyzdys 1-ame paveiksle. Imkime keletą realaus pasaulio objektų, kurių projekcijos vaizdavimo plokštumoje atrodytų kaip figūros O_{1-5} , o išskirti jų kontūrai nuo centrinio taško (p, q) arba bent jų dalys būtų aprašomi elementariomis II-os eilės kreivėmis: pirmosios trys – apskritimo lankais spinduliu r , ketvirtoji – elipse, penktoji – laisvos formos kontūru, kurį, galbūt, galima interpretuoti kaip įvairių neapibrėžtų lankinių segmentų aibę. Etalonu laikant O_1 , žmogus panašiausiu į jį objektu beveik visada išrinks O_4 , atmes O_2 ir O_3 , suabejos dėl O_5 . Kompiuteriu parametriškai skaičiuodami apskritimo $x^2 + y^2 = r^2$ ir elipsės $x^2/a + y^2/b = 1$, $a \neq b$, $a > b$, [3] kontūrus bei palygindami rezultatus

$$\frac{y - y_1}{x - x_1} = \frac{-x_1 y_1 \pm r \sqrt{x_1^2 + y_1^2 - r^2}}{r^2 - x_1^2}; \quad \frac{y - y_1}{x - x_1} = \frac{-x_1 y_1 \pm \sqrt{b^2 x_1^2 + a^2 y_1^2 - a^2 b^2}}{r^2 - x_1^2},$$

pastebėtume, kad daugiausia sutapimų (vadinasi, ir panašumo įverčių) aptikta O_{2-3} , ir visai mažai – O_4 . Kompiuterinė programa, lygindama skaičiavimų rezultatus, susidurtų su nemažais identifikavimo sunkumais. Programos su klasifikuojančiais, besimokančiais, miglotaisiais algoritmais, galbūt, pateiktų tokį patį atsakymą kaip



2 pav. Dugno histogramos: (a) originalaus, (b) išlyginto intensyvumo skirstinio.

žmogus, bet nesunku rasti sudėtingesnių pavyzdžių, kurių tokios programos neįveiktų. Todėl, siekiant supaprastinti uždavinį, taikomos įvairios morfologinės operacijos, spalvų modelių konvertavimo, dugno objektų histogramų procedūros, kontrasto stiprinimo, inversijos, binarizavimo ir kt. (2 pav.).

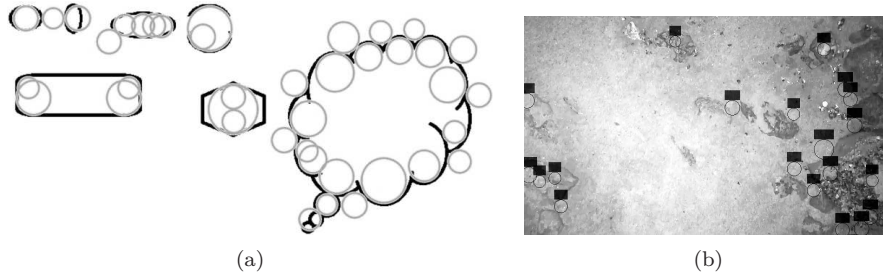
Spalvotiems vaizdams apdorojamos spalvų komponentų vektoriai ir matricos. Kuriant jūros dugno vaizdų atpažinimo metodus tikslinga naudoti objektų spalvos ir intensyvumo reikšmes visuose ar parinktuose spalvinių modelių sluoksniuose. Pradiniam sprendimo etape svarbu atskirti veiksnius, kurie turės daugiausia įtakos analizuojant jūros dugno vaizdus bei kuriant atpažinimo sistemą: charakteristikų stabilumo, matavimų (dimensijų) ir modelio dydžio pasirinkimus (sudėtingas objekto modelio artinys duomenų bazėje gali tapti kliūtimi, jei tokius pat parametrus turės kiti objektai) [4, 2, 6, 5].

2 Apskritiminės transformacijos sprendimai

Apskritimų paieškos algoritmas remiasi apskritimine *Hough* transformacija. Taikoma parametrinė išraiška: $x = p + r * \cos(\theta)$, $y = q + r * \sin(\theta)$. Apskritimo spindulys imamas pastovaus dydžio (leidžiama įvesti intervalą nuo–iki, tačiau eliminavus vieną iš kintamųjų sutrumpėja skaičiavimų laikas). Pikseliai, kurie labai skiriasi nuo aplinkos yra pažymimi ir įrašomi akumuliuojančiame masyve. Visų žymėtųjų pikselių koordinatės saugomos masyve ir vėliau pagal jas atrenkami aptikti apskritimai. Kai visi žymėtieji pikseliai ir jų vieta yra žinomi, ieškoma centro, kurį parodo didžiausios reikšmės akumuliaciniame masyve. Teisingai identifikuoto apskritimo centre susikerta daugiausiai žymėtųjų pikselių įrašus apibūdinančių apskritimų lankų. Priklausomai nuo pasirinktų jautrumo parametrų ‘Sensitivity’, ‘Edge’ algoritmas gali aptikti nupjautus arba nepilnus apskritimus ir netgi elipses (3 pav. (a), (b))

```
[cent, rad] = imfindcircles(apskritimai,[12 40],‘Sensitivity’,0.965,‘Edge’,0.03);
apskritimai = insertObjectAnnotation(apskritimai,‘circle’, [cent, rad], ‘’, ‘Color’, ‘green’);
figure, imshow(apskritimai); % uždedamos apskaičiuotos žymės.
```

Svarbu apibrėžti, kaip žymėtieji pikseliai gali jungtis, nes nuo to priklauso kaip ir kiek pikselių sancaupų bus surasta. Toliau jas vadinsime *lašeliais* (angl. *Blob*). Lašelius formuoti patogiau dviem būdais: 1) keturi pikselio $N(r, c)$ kaimynai $N4(r, c)$ priklausys lašeliui, jeigu bent vienas iš pikselių $(r - 1, c)$, $(r + 1, c)$, $(r, c - 1)$, $(r, c + 1)$



3 pav. Rezultatai, (a) teoriniame modelyje, (b) povandeninėje nuotraukoje.

turės tą pačią reikšmę; 2) aštuoni pikselio $N(r, c)$ kaimynai $N8(r, c)$ priklausys lašeliui, jeigu greta jo vienoje iš aštuonių krypčių bus tokios pačios reikšmės pikselis.

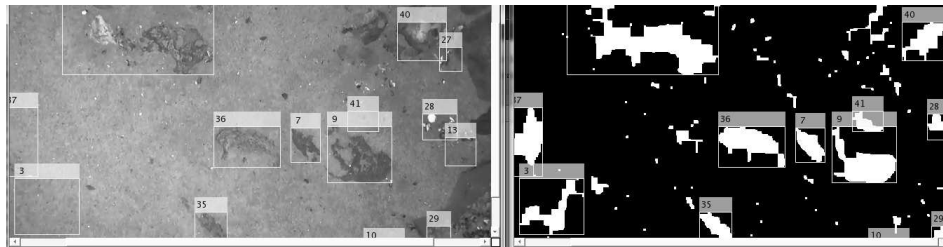
Analizuojame gautą dvejetainį paveikslą I (4 pav.) $I(r, c) = I(r', c') = v$, čia $v = 0$ arba $v = 1$. Matyti, kad pikselis (r, c) susijungia su pikseliu (r', c') turinčiu tą pačią reikšmę tada ir tik tada, jeigu egzistuoja pikselių seka $(r, c) = (r_0, c_0)$, $(r_1, c_1), \dots, (r_n, c_n) = (r', c')$, čia $I(r_i, c_i) = v$, o $i = 0, \dots, n$, ir (r_i, c_i) kaimynai $(r_i - 1, c_i - 1)$ kiekvienam $i = 1, \dots, n$. Pikselių seka $(r_0, c_0), \dots, (r_n, c_n)$ sudaro vientisą kelią nuo (r, c) iki (r', c') , čia visi pikseliai turi tą pačią reikšmę ir kiekviena pikselių pora yra susijungusi. Baigus paiešką, lašeliai surašomi į tekstinį failą (duomenų bazę).

3 Kalmano filtravimas ir lašelių analizavimas

Pašalinus foną iš tiriamojo įrašo lašelių analizatorius padeda surasti objektų ribas ir kitas tam objektui būdingas savybes, kurios skiriasi nuo esamos aplinkos savybių. Kalmano filtro ir lašelių analizatoriaus derinys padeda aptikti ir sekti vaizdo lauke objektus, kurių virtualų judesį imituoja kameros judėjimas

```
% pirmasis kodo fragmentas
FDetector = vision.ForegroundDetector('NumTrainingFrames',10, 'InitialVariance',0.05);
blobAnalyzer = vision.BlobAnalysis('AreaOutputPort', false, 'MinimumBlobArea', 2);
kalmanFilter = [];
<...>
if isObjectDetected
    kalmanFilter = configureKalmanFilter('ConstantAcceleration',...
        detectedLocation(1,:), [1 1 1]*1e5, [25, 10, 10], 25);
    <...>
    predict(kalmanFilter); trackedLocation = correct(kalmanFilter, detectedLocation(1,:))
    <...>
ColorImg = insertObjectAnnotation(ColorImg, 'circle', circle, label, 'Color', spalva);
```

Remiantis tuo, kad nedidelės jūros dugno sritys paprastai yra panašios, galima tvirtinti, kad suskaidžius dugno mozaikos vaizdą į pasirinkto dydžio juostas, gausime panašias jų šviesumo vidurkių vertes. Sekančiu žingsniu, nustatant siektiną paveikslą šviesumą, apskaičiuojamas HSI modelio apšviestumo sluoksnio matricos vidurkis, atmetant 25% labiausiai nukrypstančių reikšmių. Toliau apskaičiuojamas kiekvienos juostos matricos vidurkis ir sudaromas jų šviesumo vidurkių vektorius bei nustatomi nukrypimai nuo siektino ir esamo juostos vidurkio. Viršijus nustatytą slenkstį juostos apšvietimo reikšmės koreguojamos ir taip išlyginami tamsesni ar šviesesni kraštai.



4 pav. Lašelių analizės ir Kalmano filtro derinio veikimo iliustracija

Vaizdo paaštrinimui naudotina histogramos korekcija, pritaikius modifikuotą CLAHE (angl. *Contrast Limited Adaptive Histogram Equalization*) metodą su patobulinta $adapthisteq(O(:, :, 3), 'Range', 'original', 'Distribution', 'exponential', 'Alpha', 0.8)$ funkcija.

4 Objektų klasifikavimo rezultatai

Sukūrus mokymo duomenų masyvus ir siekiant išgauti kuo daugiau tekstūros informacijos, kiekvienas 15×15 pikselių blokas įrašomas ir pasuktas 90° , 180° , 270° kampais. Iš mokymui paruoštų blokų išskiriami savybių skaitiniai vektoriai. Spalvų aprašymui naudojami HSI, RGB ar YCbCr spalvinių modelių matricių skirtingų sluoksnių vidurkiai. Pagal kiekvieno bloko savybių vektorius priskiriama klasė iš Matlab 8 bibliotekos katalogo ir galima pradėti klasifikatoriaus apmokymą (adaptavimą), pavyzdžiui

```
load mokymo_data.mat; X = train_data(:,1:18); Y = train_data(:,19);
modelis_knn = ClassificationKNN.fit(X,Y,'NumNeighbors',20);
modelis_bayes = NaiveBayes.fit(X,Y);
```

Pradinis duomenų rinkinys sudalintas į k klosčių, viena iš jų – testavimui, o likusios – mokymui. Duomenų masyvams su 10 klosčių gauti tokie vidutiniai tikslumo įverčiai: 94,47% (paprastuoju Bajeso klasifikatoriumi), 97,17% (k -artimiausių kaimynų klasifikatoriumi). Akivaizdus skirtumas tarp skirtingomis patikros technikomis gautų rezultatų gali būti paaiškintas taip: renkant atitinkamas sritis buvo sudarytas duomenų masyvas iš skirtingų mozaikų, tačiau vykdant paeiliui, įrašomi blokai išsidėstė pagal tiriamas sritis iš eilės. Dėl to į pirmąją patikrą galėjo patekti išskirtiniai egzemplioriai, kurių savybės nepateko į mokymo imtį ir rezultatai išėjo šiek tiek prasčiau nei galėtų būti. Antruoju atveju rezultatai išėjo šiek tiek geresni nei galėtų būti, nes didžioji dalis egzempliorių buvo sukinėjami ir išsaugomi kaip atskiri pavyzdžiai. Tikėtina, kad realūs įverčiai yra išsidėstę šio spektro viduryje ir apytiksliai galėtų siekti 95 procentus.

Išvados

Aprašyti povandeninių heterogeninės aplinkos vaizdų pirminio apdorojimo reikalavimai ir priemonės. Išanalizavus prieinamus segmentavimo ir atpažinimo metodus pritaikytos alternatyvios identifikavimo technikos, filtruojant sritis pagal pikselių intensyvumą. Siūlomi jūros dugno objektų identifikavimo ir padengimo skaičiavimo

scenarijai leidžia sparčiau apdoroti didesnius jūros dugno vaizdo mozaikų masyvus, išlaikant pakankamus objektų atpažinimo įverčius.

Literatūra

- [1] K. Armalis, K. Bružas ir O. Ramašauskas. Vaizdų apdorojimas Matlab dirbtinių neuronų tinkle. *Vadyba*, **1**(12):5–10, 2008. ISSN 1648-7974.
- [2] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Comp. Vis. Imag. Underst.*, **79**(1):66–91, July 2000.
- [3] G.A. Korn and T.M. Korn. *Mathematical Handbook for Scientists and Engineers*. McGraw-Hill Co, 2nd edition, 1984, 832 pp.
- [4] D. Lee, G. Kim, D. Kim, H. Myung and H.T. Choi. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Eng.*, **48**:59–68, July 2012.
- [5] O. Ramašauskas, V. Janulevičius ir M. Kazlauskas. Jūros dugno skaitmeninių vaizdų atpažinimo aspektai, *Jūros mokslai ir technologijos 2014*. Klaipėdos universitetas, Klaipėda, 2014.
- [6] A. Šaškov ir Y. Ržanov. *Video mozaikos panaudojimas povandeniniams tyrimams, Jūros ir krantų tyrimai*. Klaipėdos universitetas, 2011, 238 pp.

SUMMARY

Seafloor mosaic objects modeling and recognition

O. Ramašauskas, V. Janulevičius, A. Vaitkus, M. Kazlauskas

Providing Klaipėda University Marine Sciences and Technology Centre instruments which are developed for seafloor digital image analysis, object recognition in the heterogeneous environment, and evaluate the results of work. The paper describes a robotic-submarine gathered image enhancement, underwater image mosaic texture and colours description, area segmentation and filtering algorithms and their software components.

Keywords: digital image, mosaic, filtering, recognition.