



(12)

BREVET DE INVENȚIE

(21) Nr. cerere: **a 2017 01176**

(22) Data de depozit: **29/12/2017**

(45) Data publicării mențiunii acordării brevetului: **28/10/2022** BOPI nr. **10/2022**

(41) Data publicării cererii:
30/08/2019 BOPI nr. **8/2019**

(73) Titular:
• **UNIVERSITATEA POLITEHNICA DIN
BUCUREȘTI, SPLAIUL INDEPENDENȚEI
NR.313, SECTOR 6, BUCUREȘTI, B, RO**

(72) Inventatori:
• **PETRESCU LUCIAN,
STR. BARBU DELAVRANCEA NR.2B,
BL.33D, SC.A, AP.2, SECTOR 1,
BUCUREȘTI, B, RO;**
• **MORAR ANCA, STR.OLĂNEȘTI NR.4,
BL.43A, SC.1, AP.5, SECTOR 6,
BUCUREȘTI, B, RO;**
• **MOLDOVEANU FLORICA,
ALEEA BAI A DE ARIEȘ NR.5, BL.1, SC.3,
AP.33, SECTOR 6, BUCUREȘTI, B, RO;**

• **MOLDOVEANU ALIN,
ALEEA BAI A DE ARIEȘ NR.5, BL.1, SC.3,
ET.4, AP.39, SECTOR 6, BUCUREȘTI, B,
RO**

(56) Documente din stadiul tehnicii:
**A. MORAR, F. MOLDOVEANU AND E.
GROLLER, "IMAGE SEGMENTATION
BASED ON ACTIVE CONTOURS WITHOUT
EDGES", IEEE 8th INTERNATIONAL
CONFERENCE ON INTELLIGENT
COMPUTER COMMUNICATION AND
PROCESSING, 2012; A. ABRAMOV, T.
KULVICIUS Ș.A., "REAL-TIME IMAGE
SEGMENTATION ON A GPU", FACING
THE MULTICOLORE-CHALLENGE,
PP. 131-142, 2010,
<http://dx.doi.org/10.1007/978-3-642-16233-614>**

(54) **SISTEM PENTRU SEGMENTAREA IMAGINILOR VIDEO
ÎN TIMP REAL, BAZAT PE TRASARE DE RAZE**



RO 133585 B1

1 Invenția se referă la un sistem și o metodă de segmentare a imaginilor video în timp
real, bazată pe extinderea regiunilor prin trasarea de raze. Metoda de segmentare rară
3 etichetează numai o fracțiune din matricea de intrare în vederea minimizării costului
computațional.

5 Invenția poate fi aplicată în domeniul informatic medical (detectare automată de
tumori din scanări CT/RMN, sisteme asistive), roboți industriali (programele de decizie ale
7 roboților autonomi industriali și a dronelor folosesc segmentare), industria automobilelor
(segmentarea este folosită de mașini cu conducere automată pentru percepția elementelor
9 de trafic), securitate (urmărirea automată pe camere de securitate folosește segmentare,
detectia automată a elementelor de interes folosește segmentare) precum și orice domeniu
11 care folosește vedere computerizată (computer vision) sau analiză de date multi-dimensio-
nale ce necesită segmentare spațială.

13 Deoarece segmentarea reprezintă un pas critic într-un număr mare de probleme,
acest subiect este intens cercetat și abordat în diverse moduri: clusterizare simplă sau dublă
15 [13] [14] [19], compresie [30], histograme [30], detectia muchiilor [30] [18], extinderea
regiunilor [26] [3] [14] [5], partiționarea grafurilor [10] [27] [29] [30], transformarea watershed
17 [15] [30], praguri adaptive [5], divizarea și unificarea [18] [30], "Random Walker" [8], contururi
ierarhice [30] [28] și active [30].

19 Cercetări recente s-au concentrat pe metode de segmentare și cosegmentare
antrenabile [30] [22] [20] [4] [16] [7] [23], folosind tehnici de învățare automată pentru a
21 produce rezultate de o acuratețe ridicată, cu supraveghere slabă. Totuși, aceste metode nu
reprezintă soluții pentru segmentări în timp real, deoarece costurile de execuție sunt mult
23 prea mari pentru puterea de calcul a sistemelor hardware nespecializate, în special pentru
laptopuri uzuale și pentru dispozitive mobile.

25 Deoarece costul computațional reprezintă un aspect critic în ce privește utilizarea
algoritmilor, o altă tendință recentă este de a exploata paralelismul hardware în vederea
27 maximizării vitezei. Această tendință a condus la dezvoltarea unor metode de segmentare/
pre-segmentare iterative rapide utilizând GPU [31] [26]. Metodele de pre-segmentare [12]
29 [19] [18] [1] [2] [17] [11] [5] creează micro-clustere locale, pe baza similarității locale, dar
necesită ulterior un proces computațional complex de clusterizare. Dacă nu ținem cont de
31 costul de clusterizare finală, acestea produc cele mai rapide rezultate locale dintre toate
metodele de segmentare. Totuși, chiar și aceste metode rapide necesită procese iterative.

33 În multe aplicații de segmentare în timp real, o calitate acceptabilă la o viteză foarte
mare este mai folositoare decât o calitate superioară la o viteză de procesare mult mai mică.
35 Segmentarea rapidă este esențială pentru urmărirea și detectia obiectelor, navigarea și
percepția pentru roboți autonomi, dispozitive asistive pentru nevăzători sau dispozitive
37 medicale controlate automat.

39 Segmentarea utilizând GPU este în general realizată prin metode iterative, unde un
număr considerabil de iterații este necesar pentru obținerea etichetării în etapa finală.
41 Metodele de pre-segmentare care utilizează GPU folosesc o strategie similară, conducând
la supra-segmentare cu etichetare locală de calitate înaltă.

43 Majoritatea metodelor inițiale de segmentare pe GPU au fost realizate în scopul
utilizării în medicină, unde accelerarea GPU a adus beneficii considerabile în manipularea
45 matricilor tridimensionale foarte mari. Schenke și alții [26] au investigat oportunitatea oferită
de hardware-ul paralel și au introdus o metodă de segmentare hibridă CPU-GPU, bazată pe
extinderea regiunilor pornind de la puncte germen, prin operații de dilatare și eroziune.
47 Hagan și alții [13] au utilizat un model LBM (Lattice Boltzmann Model) extins pentru a rezolva
ecuația „level set”, într-o abordare iterativă care generează etichetări de calitate ridicată în

detrimentul unui număr mare de iterații cu sincronizare CPU-GPU. Vineet și alții [29] au adaptat algoritmul "maxflow mincut" pentru CUDA, în care este utilizată metoda „tăierii grafului" pentru a partiționa un set de date într-o mulțime de sub-seturi disjuncte. Re-etichetarea grafului între numeroasele iterații de tăiere a grafului este realizată printr-o sincronizare intensiv computațională. Roberts și alții [25] folosesc un algoritm iterativ bazat pe "level set" cu o complexitate de $O(n \cdot \log(n))$. Korbes și alții [15] au introdus un algoritm "watershed" paralel iterativ, în care o imagine este divizată în ferestre de dimensiune 16 x 16 și fiecare fereastră realizează o transformare watershed locală, pentru fiecare iterație a algoritmului. Collins și alții [8] au mapat problema cosegmentării pe operații de algebră liniară, care au fost realizate utilizând CUDA, oferind o soluție de cosegmentare de calitate înaltă la un cost computațional scăzut. Ramirez și alții [24] au segmentat volume cu o adaptare pe GPU a GrabCut, un algoritm de flux proiectat pentru partiționarea imaginilor. Similar cu [29], algoritmul Push-Relabel este implementat în CUDA, presupunând costuri de sincronizare.

Investigații recente în ce privește segmentarea executată pe GPU se bazează pe metode antrenabile [30] [22] [20] [4] [16], în care diverși algoritmi de învățare automată supervizați slab sunt utilizați pentru învățarea și detecția informațiilor în seturile de date. Segmentarea este astfel realizată cu metode de învățare automată precum Support Vector Machines (SVM), Markov Random Fields (MRF), Condițional Random Fields (CRF) sau Fully Convolved Networks (FCN). Totuși, costul acestor metode este prea mare pentru segmentările în timp real.

Algoritmii de pre-segmentare rezolvă problema segmentării numai local, utilizând de obicei strategii de ascensiune a gradientului, unde punctele germen sunt mutate iterativ în vecinătăți locale, etichetând imaginile la nivel local, în clustere mici numite superpixeli. Algoritmii de pre-segmentare au cel mai înalt nivel de performanță în ce privește timpul de rulare. Datorită vitezei, algoritmii de pre-segmentare reprezintă candidați excelenți pentru metodele de prelucrare în timp real, deoarece pot fi combinați cu strategii ieftine de unificare a regiunilor. De aceea algoritmii de pre-segmentare sunt preferați în detrimentul algoritmilor compleți de segmentare în aplicații critice. Fulkerson și alții [11] au utilizat supra-segmentare conservativă a regiunilor mici pentru a produce super-pixeli de dimensiune variabilă. Levinshtein și alții [17] au introdus TurboPixels, o metodă în care super-pixelii sunt calculați cu fluxuri geometrice iar punctele germen sunt iterativ perturbate pentru a acoperi vecinătăți locale. Algoritmul are o complexitate de $O(n)$, unde n este dimensiunea setului de date, și se bazează pe o serie de operatori de dilatare. Fulkerson și alții [12] propun o alterare a algoritmului Quick Shift, compatibilă cu CUDA, în care un spațiu de cinci caracteristici este utilizat pentru a stabili legătura dintre pixeli și clustere. Implementarea are o complexitate de $O(d \cdot n^2)$, unde d este o constantă, dar în practică este foarte rapidă, deoarece nu este iterativă. Singurul dezavantaj al metodei este controlul slab asupra dimensiunii și gradului de compactare al super-pixelilor rezultați. Achanta și alții [1] au introdus o metodă simplă de clusterizare liniară iterativă (SLIC - simple linear iterative clustering) în super-pixeli, unind pixeli pe baza similarității într-o manieră iterativă, dar limitând spațiul de căutare la o regiune proporțională cu dimensiunea superpixelului. Complexitatea algoritmului este $O(n)$. Achanta și alții [2] au îmbunătățit metoda din [1] cu o variantă a algoritmului numită SLICO. Li și alții [18] au îmbunătățit de asemenea calitatea segmentării SLIC [1] prin utilizarea unei strategii iterative de divizare și unificare. În fiecare iterație, superpixelii sunt divizați pe baza unei hărți de muchii și sunt apoi unificați cu superpixelul adiacent cu cea mai mică distanță Bhattacharyya. Această metodă obține o calitate ridicată în detrimentul unui timp mai scăzut de rulare. Li și alții [19] folosesc clusterizare spectrală liniară pentru a îmbunătăți acuratețea segmentării SLIC, obținând un cost computațional foarte ridicat.

RO 133585 B1

1 Dintre toate metodele de pre-segmentare discutate, cele mai potrivite pentru
segmentare în timp real sunt ReallyQuickShift [12] și SLICO [2]. Ambele metode folosesc o
3 strategie de clusterizare bazată pe ascensiunea gradientului și oferă o calitate a segmentării
rezonabilă pentru un timp de rulare foarte scăzut.

5 Pentru rezolvarea problemei tehnice de minimizare a costului computațional, sistemul
pentru segmentarea imaginilor video în timp real, bazat pe trasare de raze, este capabil să
7 realizeze operații de segmentare a imaginilor și matricilor multi-dimensionale într-un mod
inedit, printr-o eșantionare rapidă a spațiului setului de date de intrare, în detrimentul
9 acoperirii tuturor punctelor dintr-o vecinătate locală.

Sistemul este compus din următoarele elemente:

- 11 a. Un dispozitiv de achiziție sau de generare a matricilor bi- sau multi-dimensionale;
- 13 b. Un modul opțional de pre-procesare a datelor de intrare (exemplu: filtrarea
zgomotelor, calculul unei hărți de normale pornind de la imagini de adâncime);
- 15 c. Un modul de segmentare rară prin trasarea de raze care obține la ieșire regiuni
continue disjuncte, prin metoda de segmentare propusă;
- 17 d. Un modul de post-procesare care utilizează regiunile disjuncte în funcție de scopul
aplicației.

19 Metoda de segmentare a imaginilor video și a matricilor multi-dimensionale are un
grad înalt de paralelizare, conducând la o segmentare rapidă, în timp real. Aceasta are
următoarea succesiune de etape:

21 - Etapa 1: Calcularea unui flux peste matricea de intrare, care descrie rata de
schimbare a valorii unui element relativ la vecinătatea acelui element (acest flux se poate
23 estima prin diferențe finite);

25 - Etapa 2: Împărțirea matricii de intrare în sub-regiuni egale (în cazul 2D, imaginile
sunt împărțite în dreptunghiuri egale de dimensiune 4x4, 16x16 pixeli sau altă dimensiune,
dată ca parametru de intrare) și alegerea unui punct germen pentru fiecare sub-regiune.
27 Aceste puncte germen acoperă cât mai omogen spațiul setului de date de intrare.

29 - Etapa 3: Trasarea de raze multiple plecând de la puncte germen. Această trasare
garantează o acoperire eficientă a întregului set de date, dar cu un număr foarte mic de
eșantioane, minimizând costurile explorării spațiului de căutare. Fiecare rază conține statistici
31 despre elementele din spațiul traversat.

33 Etapa 4: Conectarea punctelor germen (și implicit, a razelor care aparțin de acele
puncte germen) în momentul în care razele trasate se intersectează. Conectarea se face
doar în condiții de similaritate între razele care se intersectează (dacă elementele din spațiile
35 parcurse de raze au proprietăți similare). La conectarea a două puncte germen, unul dintre
ele devine părintele celuilalt punct germen. Astfel, punctele germen conectate prin raze sunt
37 unite, rezultând în final un set de puncte germen părinte care partiționează matricea de
intrare în regiuni continue care nu se suprapun unele cu celelalte.

39 Etapa 5: Extinderea opțională a regiunilor generate pentru a acoperi complet setul
de date inițial, utilizând un proces de filtrare în doi pași:

41 Pasul 1: Pentru fiecare element ne-etichetat, se trimit raze care eșantionează spațiul.
Atunci când o rază întâlnește o regiune, dacă proprietățile razei sunt asemănătoare cu cele
43 ale regiunii, elementul ne-etichetat primește id-ul regiunii respective;

45 Pasul 2: Elementele care nu au fost etichetate nici în pasul 1 și care nu se află pe
zone de flux mare, sunt prelucrate prin eșantionarea regiunilor care se află în vecinătatea
lor. Elementul ne-etichetat primește id-ul regiunii din vecinătate care are proprietățile cele
47 mai similare cu acelea ale elementului respectiv.

RO 133585 B1

Etapa 6: Unirea opțională a sub-seturilor generate, folosind un proces iterativ de calcul de statistici în interiorul regiunilor și de unificare a regiunilor vecine pe baza acestor statistici (dacă aceste două regiuni au proprietăți similare).	1
Rezultatul aplicării metodei este un set de regiuni continue disjuncte, fiecare regiune conținând elemente cu proprietăți similare. Acest rezultat poate fi folosit ulterior la segmentarea semantică a imaginilor video sau a matricilor multi-dimensionale (de exemplu, seturi de date CT).	3
Se dă în continuare un exemplu de realizare a invenției, în legătură cu fig. 1...10:	5
- fig. 1, schema de funcționare a sistemului și oferă explicații suplimentare cu privire la sub-sistemele constitutive;	7
- fig. 2, ieșirea sistemului având ca intrare o imagine 2D, cu diferite niveluri de precizie prin diverse configurări ale gradului de risipire a razelor;	9
- fig. 3, strategii de explorare a spațiului 2D, de tipul k-means (care caută în întreaga imagine), bazate pe superpixeli (care caută într-o vecinătate) și prin trasare de raze (care caută în întreaga imagine, dar cu foarte puține eșantioane);	11
- fig. 4, algoritmul de segmentare: imaginea de intrare (o imagine de adâncime), fluxul calculat pe imaginea de intrare, generarea punctelor germen (a generatorilor), explorarea spațiului prin trasarea razelor, unirea punctelor germen cu proprietăți similare, filtrarea și unificarea regiunilor;	13
- fig. 5, scenarii de trasare și terminare a razelor, în cazul întâlnirii unui pixel care a fost parcurs de o altă rază, la întâlnirea unei zone de flux ridicat (o muchie) sau la acumularea de flux;	15
- fig. 6, împărțirea razelor în funcție de puterea fluxului (stânga) și rasterizarea conservativă pentru detecția de conexiuni dacă direcția unei raze nu se aliniază perfect cu axele imaginii (dreapta);	17
- fig. 7, rezultate prin unificarea regiunilor;	19
- fig. 8, rezultate calitative ale segmentării prin trasarea de raze, unde diferite imagini sunt segmentate și comparate cu segmentări consacrate: imaginea de intrare (coloana 1), rezultatul metodei de segmentare rară, cu și fără filtrare, cu diferite configurări ale gradului de risipire a razelor (coloanele 2, 3, 4, 5), rezultatul segmentării ReallyQuickShit (coloana 6) și rezultatul segmentării SLIC (coloana 7);	21
- fig. 9, măsurători de viteză, în milisecunde, comparativ cu măsurători ale unor segmentări consacrate (SLIC și ReallyQuickShift);	23
- fig. 10, folosirea ca o metodă de pre-segmentare.	25
În fig. 1 se prezintă modul de funcționare a sistemului, care obține date de intrare de la un dispozitiv de achiziție sau de la un generator de date (de exemplu, prin calcule științifice). Un modul de pre-procesare prelucrează datele înainte de segmentare (de exemplu, pentru sistemul Sound of Vision, modulul de pre-procesare determină un nor de puncte pornind de la o imagine de adâncime și calculează o hartă de normale). Urmează segmentarea cu trasare de raze, care calculează regiuni continue disjuncte. Post-procesarea poate fi orice modul care utilizează regiunile disjuncte, pentru identificarea și urmărirea de obiecte (exemplu: în sistemul Sound of Vision modulul de post-procesare identifică obiecte de interes în mediu, cum ar fi podeaua, pereții, tavanul, obstacolele, scările și le interpretează în semnale audio și haptice).	27
Sistemul propus are la bază un algoritm original de segmentare (partiționare) a câmpurilor K-dimensionale (exemplu: imagini bidimensionale), care asigură o viteză la execuție de aproape 5 ori mai mare decât cea realizată de sistemele existente, după cum se poate observa în fig. 9. Această performanță este obținută prin redefinirea problemei de partiționare ca o problemă de trasare de raze utilizând unitatea de procesare grafică (GPU: Graphical Processing Unit) a calculatorului.	29
	31
	33
	35
	37
	39
	41
	43
	45
	47
	49

RO 133585 B1

1 Algoritmul a fost dezvoltat pentru Sound of Vision [32], un sistem asistiv pentru
nevăzători care scanează mediul înconjurător prin camere video, identifică elementele de
3 interes din imaginile obținute și emite semnale audio și haptice care permit utilizatorului să
perceapă mediul înconjurător, ajutându-l să navigheze în medii necunoscute. Cu toate că
5 în sistemul Sound of Vision algoritmul de segmentare partiționează imagini bidimensionale,
modelul său matematic poate fi folosit pentru partiționarea de matrici K-dimensionale.

7 Metoda de segmentare rară etichetează numai o fracțiune din matricea de intrare în
vederea minimizării costului computațional. Fig. 2 prezintă ieșirea sistemului având ca
9 intrare o imagine 2D și modul în care se pot obține diferite niveluri de precizie prin diverse
configurări ale gradului de risipire a razelor. Gradul de risipire este controlat parametric.
11 Chiar cu un grad de risipire ridicat, segmentarea rezultată este suficient de calitativă pentru
numeroase aplicații în timp real.

13 Metoda este proiectată astfel încât să permită procesare în paralel pe un număr mare
de procesoare (spre exemplu, paralelismul hardware al plăcilor grafice actuale) și să profite
15 de pe urma unor optimizări în trasarea razelor („packet tracing” [6], Digital Differential
Analyzer - DDA [21]).

17 Metoda este exemplificată în 2D, dar poate fi implementată în oricâte dimensiuni.
Metoda de segmentare propusă are un grad înalt de paralelizare, conducând la o
19 segmentare rapidă, în timp real.

Metoda de segmentare a imaginilor video și a matricilor multi-dimensionale în timp
21 real, bazată pe trasare de raze, are următoarea succesiune de etape:

- Etapa 1: Calcularea unui flux peste matricea de intrare, care descrie rata de
23 schimbare a valorii unui element relativ la vecinătatea acelui element (acest flux se poate
estima prin diferențe finite);

25 - exemplu de implementare: în cazul imaginilor video 2D, acest flux poate fi
gradientul; în cazul imaginilor de adâncime, se poate calcula un flux adaptiv, care ține cont
27 de eroarea senzorului de adâncime (care crește o dată cu distanța de la senzor)

- Etapa 2: Împărțirea matricii de intrare în sub-regiuni egale (în cazul 2D, imaginile
29 sunt împărțite în dreptunghiuri egale de dimensiune 4x4, 16x16 pixeli sau altă dimensiune,
dată ca parametru de intrare) și alegerea unui punct germen pentru fiecare sub-regiune.
31 Aceste puncte germen acoperă cât mai omogen spațiul setului de date de intrare.

- exemplu de implementare și optimizare: Pozițiile punctelor germen pot fi
33 alese folosind secvențe de tipul Sobol, Hammersly și Van der Corput sau distribuții Poisson.
După construcție, generatorii sunt mutați iterativ departe de zonele de flux mare, pentru a
35 maximiza șansele de generare de regiuni de segmentare din zone cu flux scăzut, adică zone
în care similaritatea dintre vecini este mare.

37 - Etapa 3: Trasarea de raze multiple plecând de la puncte germen. Această trasare
garantează o acoperire eficientă a întregului set de date, dar cu un număr foarte mic de
39 eșantioane, minimizând costurile explorării spațiului de căutare. Fiecare rază conține statistici
despre elementele din spațiul traversat.

41 - exemplu de implementare și optimizare: Pentru imagini 2D, trasarea
presupune parcurgerea pixelilor cu raze care sunt trimise în 8 direcții: pe verticală (sus, jos),
43 pe orizontală (stânga, dreapta) și pe cele patru direcții diagonale. Traversarea spațiului este
făcută cu rasterizare conservativă bazată pe algoritmul Digital Differential Analyzer modificat.

45 Etapa 4: Conectarea punctelor germen (și implicit, a razelor care aparțin de acele
puncte germen) în momentul în care razele trasate se intersectează. Conectarea se face
47 doar în condiții de similaritate între razele care se intersectează (dacă elementele din spațiile
parcurse de raze au proprietăți similare). La conectarea a două puncte germen, unul dintre

RO 133585 B1

ele devine părintele celui alt punct germen. Astfel, punctele germen conectate prin raze sunt unite, rezultând în final un set de puncte germen părinte care partiționează matricea de intrare în regiuni continue care nu se suprapun unele cu celelalte.	1 3
Etapa 5: Extinderea opțională a regiunilor generate pentru a acoperi complet setul de date inițial, utilizând un proces de filtrare în doi pași:	5
Pasul 1: Pentru fiecare element ne-etichetat, se trimit raze care eșantionează spațiul. Atunci când o rază întâlnește o regiune, dacă proprietățile razei sunt asemănătoare cu cele ale regiunii, elementul ne-etichetat primește id-ul regiunii respective;	7
Pasul 2: Elementele care nu au fost etichetate nici în pasul 1 și care nu se află pe zone de flux mare, sunt prelucrate prin eșantionarea regiunilor care se află în vecinătatea lor. Elementul ne-etichetat primește id-ul regiunii din vecinătate care are proprietățile cele mai similare cu acelea ale elementului respectiv.	9 11
Etapa 6: Unirea opțională a sub-seturilor generate, folosind un proces iterativ de calcul de statistici în interiorul regiunilor și de unificare a regiunilor vecine pe baza acestor statistici (dacă aceste două regiuni au proprietăți similare).	13 15
Rezultatul aplicării metodei este un set de regiuni continue disjuncte, fiecare regiune conținând elemente cu proprietăți similare. Acest rezultat poate fi folosit ulterior la segmentarea semantică a imaginilor video sau a matricilor multi-dimensionale (de exemplu, seturi de date CT).	17 19
Segmentarea rară prin trasarea de raze aduce următoarele contribuții:	
- extinderea de regiuni pe baza trasării de raze conduce la o segmentare cu acoperirea rapidă a spațiului setului de date de intrare (matricii K-dimensionale) în detrimentul acoperirii tuturor punctelor dintr-o vecinătate locală;	21 23
- un număr constant de treceri prin banda grafică, conducând la un timp de execuție care nu depinde de tipul imaginii, fără a afecta calitatea;	25
- lățime de bandă scăzută datorită rarității razelor, care determină un număr mic de accese la memorie;	27
- o complexitate de $O(n/tsize)$ unde n este dimensiunea setului de intrare iar $tsize$ este dimensiunea ferestrei (sub-regiunii) pentru care este generat un punct germen de pornire a razelor (exemplu: 16 x 16). Segmentările cu cele mai bune rezultate, SLIC [2] și ReallyQuickShift [12] au complexități de $O(n)$ și $O(d*n^2)$.	29 31
Segmentarea rară prin trasarea de raze este sumar descrisă în fig. 4.	
Aceasta este o componentă cheie pentru proiectul Sound of Vision [32, 33], unde este utilizată pentru segmentarea imaginilor de adâncime în timp real, pe plăci grafice uzuale.	33 35
Segmentarea K-dimensională este o problemă de partiționare în care un set de date S care conține n elemente K-dimensionale (o matrice K-dimensională) trebuie să fie împărțit în regiuni continue disjuncte, pe baza unor proprietăți de similaritate locală sau globală între elemente. Această problemă este rezolvată prin căutarea regiunilor disjuncte în S și determinarea celei mai bune soluții.	37 39
Metodele de extindere a regiunilor sau de clusterizare determină această partiționare prin încercarea iterativă de unire a elementelor din afara clusterelor, după cum este ilustrat în fig. 3. Metodele tradiționale precum k-means [14] explorează întregul set, producând un spațiu de căutare extrem de mare, care se explorează destul de greu. Metode de pre-segmentare prin extinderea regiunilor ca SLIC [2] limitează spațiul de căutare la o vecinătate foarte mică a clusterului, garantând o explorare rapidă a spațiului de căutare, dar neputând să detecteze segmente (regiuni) foarte mari.	41 43 45 47

RO 133585 B1

1 Strategia propusă de invenția noastră combină avantajele celor două abordări:
2 numărul mic de elemente explorate de abordarea SLIC și spațiul mare de căutare al metode-
3 lor tradiționale, ducând în același timp la o viteză de execuție foarte mare și la conexiuni de-a
4 lungul întregului set. Metoda de segmentare rară cu trasare de raze funcționează identic pe
5 matrici de orice dimensiuni, dar pentru simplitate este ilustrată pe imagini bidimensionale.

6 Algoritmul de segmentare complet este ilustrat sumar în fig. 4: calculul fluxului,
7 generarea punctelor germen (a generatorilor), explorarea spațiului prin trasarea razelor și
8 conectarea regiunilor parcurse.

9 Fluxul reprezintă rata de schimbare a matricii de intrare. Poate fi un simplu gradient
10 adaptiv pentru imagini color/în niveluri de gri și este extrem de ieftin de calculat pe GPU.
11 Pentru imagini de adâncime (de exemplu, cele obținute de la dispozitivul de achiziție Sound
12 of Vision) s-a utilizat un gradient adaptiv care ține cont de eroarea de estimare a senzorului
13 de adâncime. Un exemplu de flux se poate observa în fig. 4.

14 Poziționarea punctelor germen este inițial generată printr-un șablon pseudo-aleator
15 de-a lungul întregului set. De exemplu, pentru imagini 2D, oricare din secvențele Sobol, van
16 der Corput sau Hammersly pot fi utilizate. După generarea pozițiilor inițiale, se utilizează o
17 strategie bazată pe ascensiunea gradientului pentru distanțarea punctelor germen de zonele
18 cu flux ridicat, utilizând următorul algoritm, redat în pseudocod:

```
19 Pseudocode 1. Poziționarea punctelor germen  
20 1: seedPos ← getPseudoRandomPosition()  
21 2: iterations ← 0  
22 3: flux ← getFlux(seedPos)  
23 4: while flux > Threshold and iterations < MaxIterations  
24 5: flux Direction ← getFluxDirection(seedPos)  
25 6: seedPos ← seedPos + fluxDirection  
26 7: flux ← getFlux(seedPos)  
27 8: iterations ← iterations + 1  
28 9: return seedPos
```

29 Un exemplu al distribuției rezultate a punctelor germen poate fi observată în fig. 4
30 (subfig. 3), colorată cu o funcție de hash Jenkins aplicată peste indicii punctelor germen.

31 După generarea pozițiilor punctelor germen, din fiecare astfel de punct sunt trimise
32 raze prin imagine în vederea acoperirii rapide a întregului spațiu de căutare. Razele au
33 multiple cazuri de terminare, după cum se poate observa în fig. 5: întâlnirea unui pixel care
34 a fost parcurs de o altă rază, întâlnirea unei zone de flux ridicat (o muchie) sau acumularea
35 de flux.

36 Razele suportă de asemenea reflexii pe marginile imaginii sau în zone de flux ridicat
37 (muchii), reprezentând exploratori eficienți chiar și pentru cazuri complicate. Razele
38 numeroase sunt cu ușurință distribuite pe toate nucleele de procesare disponibile. Rezultatul
39 razelor trasate dar neconectate se poate observa în fig. 4 (subfig. 3).

40 O structură arborescentă este creată peste punctele germen, care leagă toate
41 punctele de plecare ale întregului set de date. Inițial toate punctele germen pornesc ca arbori
42 separați. Cu fiecare conexiune între raze, numărul de arbori separați scade. În vederea
43 minimizării înălțimii arborilor atunci când două raze se întâlnesc, numai punctele germen sunt
44 conectate. Deoarece sincronizarea datelor pe GPU poate fi implementată eficient numai prin
45 operații atomice, trebuie stabilită o ierarhie atunci când două puncte germen sunt conectate.

RO 133585 B1

```
1      23: writeLabel (pos, seed.id)
      24: writeTraceProperties(pos, rayProperties)
3      25: else
      26:   if alreadyConnectedNets (seed.id, id)
5      27:     end //the seeds are already part of the same
      28:     //forest (they have the same greatest parent)
7      29:   posProperties ← readProperties(pos)
      30:   if rayProperties not_similar posProperties
9      31:     end
      32:   updateRayProperties (rayProperties, posProperties)
11     33: //update with the properties of the other ray traced through pos
      34:   writeTraceProperties (pos, rayProperties)
13     35:   ConnectSeeds(seed.id, id)
      36:   if (ray.dir + scatterDir) aligned to image axes
15     37:     pos ← pos + ray.dir + scatterDir
      38:   else
17     39:     pos ← StaircaseTracing (ray.dir + scatterDir)
```

Traversarea conservativă este utilizată dacă direcția unei raze nu se aliniază perfect cu axele imaginii, în vederea asigurării detecției de conexiuni chiar și în cazuri dificile, după cum se poate observa în fig. 6 (dreapta). În plus, devierea slabă a direcției este utilizată dacă raza acumulează suficient flux, în vederea ghidării razei departe de regiunile cu flux mare către cele cu flux mic, urmărind aceleași principii ca transformarea watershed. Reflectanța stocastică a razelor către regiuni de flux local mic împiedică razele să realizeze conexiuni riscante în regiuni de flux ridicat, după cum este ilustrat în fig. 6 (stânga).

În final, după ce toate razele sunt trasate, rămâne o mică posibilitate ca două raze să se termine aproape una de cealaltă, dar nici una din ele să nu realizeze un caz de conexiune. Astfel, un pas în plus este necesar pentru a impune conectivitatea. Pasul final actualizează fiecare pixel la punctul germen părinte final. Punctele germen fără părinte, care acționează ca rădăcini în structura arborescentă construită peste imagine, reprezintă id-urile de segmentare unice. Pseudocodul pentru forțarea conectivității este următorul:

31 Pseudocod 4. Forțarea conectivității (în paralel pentru fiecare pixel)

```
1: pos ← getPos(pixel)
33 2: seedId ← readLabel (pos)
3: properties ← readProperties(pos)
35 4: flux ← readFlux(pos)
5: foreach posN neighbor of pos
37 6: seedNId ← readLabel(posN)
7: propertiesN ← readProperties(posN)
39 8: fluxN ← readFlux(posN)
9: if properties similar propertiesN
41 10: if maxflux, fluxN) < MaxFlux
11: ConnectSeeds (seedId, seedNId)
```

43 Numărul de operații de acces la memorie în această metodă este foarte mic. Trasarea pachetelor ("packet tracing") poate fi utilizată pe raze, iar împachetarea per punct germen conduce la rezultate bune deoarece în mod uzual fiecare generator trimite raze comparabile ca lungime și cu conexiuni similare.

RO 133585 B1

Trasarea razelor a fost îmbunătățită prin modificarea algoritmului clasic de rasterizare DDA și transformarea lui într-unul conservativ. Această îmbunătățire elimină un număr necesar de operații de acces al memoriei. Utilizarea direcțiilor predefinite poate scădea de asemenea costul computațional de traversare a spațiului. 1 3

Segmentarea rară este suficientă pentru un număr mare de aplicații practice de segmentare, dar în anumite cazuri o segmentare completă este de dorit. Această etapă opțională este prezentată în continuare. 5 7

După etichetarea rară a spațiului, id-urile unice de segmentare pot fi considerate ca o rețea rară peste imagine. Filtrarea este implementată prin conectarea pixelilor care nu au etichete, într-un proces iterativ, fiecare iterație având două etape. În prima etapă pixelii sunt conectați prin aceleași principii ca în pasul anterior, de conexiune a razelor. În a doua etapă pixelii sunt conectați pe baza unei similarități în interiorul unei vecinătăți. 9 11

Pseudocodul pentru filtrarea exactă este următorul: 13

Pseudocode 5. Filtrarea exactă (labeling complet) - prima etapă (pentru fiecare pixel)

1: pos ← getPos(pixel) 15

2: if readLabel(pos) not undefined

3: end 17

4: steps ← 0

5: for rayld ← 1, rayld < NumRays, rayld+ 19

6: alivefrayld] ← true

7: rayProperties[rayld] ← empty 21

8: for steps ← 1, steps < MaxSteps, steps ++

9: for rayld ← 1, rayld < NumRays, rayld+ 23

10: if not alivefrayld]

11: continue 25

12: ray ← getRay(rayld)

13: if ray.dir a lign ed to im age axes 27

14: rpos ← pos + steps · ray.dir

15: else 29

16: rpos ← StaircaseTracing(ray. dir)

17: flux ← readFlux(rpos) 31

18: if flux > MaxFlux

19: alive[rayld] ← false 33

20: continue

21: input ← readInput(rpos) 35

22: updateRayProperties(properties[rayld], input)

23: seedld ← readLabel(rpos) 37

24: if seedld not undefined

25: posProperties ← readProperties(rpos) 39

26: if properties[rayld] similar posProperties

27: writeLabel(pos, seedld) 41

28: end

29: else 43

30: alive[rayld] ← false

A doua etapă de filtrare, bazată pe similaritate, este identică cu etapa de filtrare exactă cu excepția constrângerilor bazate pe potrivirea proprietăților. Procesul de filtrare este aplicat iterativ, însă în general este nevoie de un număr foarte mic de iterații (2 sau 3) pentru a eticheta întregul set de date. 45 47

RO 133585 B1

1 Unificarea regiunilor este un alt proces opțional, care poate conduce la îmbunătățiri
calitative semnificative, după cum se poate observa în fig. 7. Unificarea este de asemenea
3 un proces iterativ ce conține mai multe etape. În prima etapă se pornește de la imaginea
complet etichetată și fiecare punct germen trimite raze scurte pentru a obține statistici despre
5 etichetarea locală.

Apoi, în etapa a doua se acumulează informații în părintele final al fiecărui punct
7 germen, utilizând operații atomice pentru a sincroniza scrierile. În a treia etapă se rulează
o metodă per-pixel care compară toți vecinii potențiali dintr-un nucleu cu pixelul central. Dacă
9 proprietățile se potrivesc dar etichetarea este diferită, atunci cele două etichete sunt
unificate.

11 Atât filtrarea cât și unificarea regiunilor cresc calitatea etichetării, în timp ce costurile
suplimentare conduc în continuare la timpi de procesare comparabili cu cei ai metodelor de
13 ultimă oră de (pre)segmentare, după cum este ilustrat în fig. 9.

Metodologia de evaluare a rezultatelor compară segmentarea imperfectă rară cu
15 trasare de raze atât calitativ cât și din punct de vedere al vitezei cu cele mai rapide metode
de segmentare pe GPU, cu ReallyQuickShift [12] și SLICO [2].

17 În timp ce acestea sunt metode de pre-segmentare și necesită procesare supli-
mentară pentru producerea de segmentări complete, ele reprezintă singura categorie de
19 metode care sunt comparabile în timp cu metoda prezentată. Din punctul de vedere al
vitezei, segmentarea rară cu trasare de raze are o complexitate de $O(n/tsize)$, fiind mult mai
21 rapidă decât orice algoritm complet de segmentare pe GPU, cel mai rapid având complexi-
tatea $O(n \cdot \log n)$.

23 Rezultatele arată că metoda propusă are rezultate locale bune, după cum este
prezentat în fig. 8, și rezultate de viteză excepționale, după cum este prezentat în fig. 9.
25 Metoda propusă obține segmentare completă în mai puțin timp decât au nevoie metodele
actuale să finalizeze segmentările locale.

27 Fig. 8 prezintă rezultatele calitative ale segmentării prin trasarea de raze, unde
diferite imagini sunt segmentate și comparate cu segmentări consacrate. Testarea calitativă
29 include atât varianta rară (în care se trasează numai razele) cât și filtrată (când se
etichetează tot setul de date) a algoritmului prezentat, și ferestre de dimensiuni diferite care
31 afectează direct numărul de puncte germen. Astfel, a patra coloană din fig. 8 necesită mărire,
deoarece rezultatele sunt obținute cu o valoare de raritate/risipire foarte mică.

33 Fig. 9 prezintă rezultatele de viteză, unde segmentarea bazată pe trasarea de raze
are în mod consistent timpi de rulare de aproximativ 5 ori mai mari decât SLIC și de
35 aproximativ 20 de ori mai mari decât ReallyQuickShift. Figura ilustrează faptul că metoda
propusă obține timpi de rulare aproximativ egali cu SLIC chiar și utilizând pasul opțional de
37 filtrare. Din fig. 9 se poate observa de asemenea că metoda prezentată are o degradare sub-
liniară a performanței în ce privește numărul de pixeli, fiind ideală pentru seturi de date mari
39 și foarte mari.

Segmentarea rară bazată pe trasarea de raze este o metodă flexibilă, unde compro-
41 misul între calitate și viteză este controlat printr-un singur parametru de risipire. Aceasta
poate fi folosită chiar ca o metodă de pre-segmentare, după cum este ilustrat în fig. 10. Pre-
43 segmentarea rezultată conservă o mare parte din structura locală dar descoperă complet și
unifică segmente foarte mari, diferențiindu-se față de metodele existente de pre-segmentare
45 care partiționează astfel de segmente.

O limitare a abordării noastre este dependența de calculul fluxului, care poate conține
47 mulți parametri, în special dacă se utilizează o soluție neadaptivă. Explorarea rară a spațiului
conduce la posibile erori cauzate de zgomot. O altă limitare este faptul că metoda a fost

proiectată pentru utilizare în cazuri uzuale (urmărirea video, detecția de caracteristici importante, utilizare în timp real) și nu reprezintă alegerea perfectă pentru cazuri de segmentare specializată unde calitatea este mult mai importantă decât viteza (exemplu: OCR). 1
3

Segmentarea rară bazată cu trasare de raze este deja utilizată într-o aplicație practică în cadrul sistemului Sound of Vision [32], unde segmentează cadre video de adâncime în timp real, utilizând un detector de flux adaptiv exponențial. Datorită costului computațional foarte scăzut al segmentării, aceasta poate fi folosită într-un flux de lucru complex, împreună cu alte module ca detecția caracteristicilor, estimarea normalelor, stabilizarea și filtrarea semnalelor, analiza pentru determinarea spațiului navigabil, etichetarea obiectelor și codificarea elementelor mediului în semnale audio și haptice. 5
7
9
11

Bibliografie 13

[1] R. Achanta, A. Shaji, K. Smith, A. Lucehi, P. Fua, and S. Susstrunk, SLIC Superpixels, EPFL Technical Report no. 149300, 2010. 15

[2] R. Achanta, A. Shaji, K. Smith, A. Lucehi, P. Fua, and S. Susstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, voi. 34, num. 11, p. 2274-2282, 2012. 17
19

[3] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, Proceedings of SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027-1035, 2007. 21

[4] G. Bertasius, L. Torresani, S. Yu, X. Stella and J. Shi, Convolutional Random Walk Networks for Semantic Image Segmentation, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 23
25

[5] Y. Beevi and S. Natarajan, An efficient Video Segmentation Algorithm with Real time Adaptive Threshold Technique, 2009. 27

[6] S. Boulos, D. Edwards, J. Lacewell, J. Kniss, J. Kautz, P. Shirley, I. Wald, Packet-based whitted and distribution ray tracing, Proceedings of Graphics Interface, 2007, pp. 177-184. 29

[7] R. Cinbis, J. Verbeek, and C. Schmid. Multi-fold ml training for weakly supervised object localization. Conference on Computer Vision and Pattern Recognition, 2014. 31

[8] M. Collins, J. Xu, L. Grady and V Singh, Random Walks based Multi-Image Segmentation: Quasiconvexity Results and GPU-based Solutions, Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1656-1663, 2012. 33
35

[9] D. Comaniciu, P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 24, iss 5, pp. 603-619, 2002. 37

[10] P. Felzenszwalb, D. Huttenlocher, Efficient Graph-Based Image Segmentation, Journal of Computer Vision, voi. 59, iss. 2, pp. 167-181, 2004. 39

[11] B. Fulkerson, A. Vedaldi and S. Soatto, Class Segmentation and Object Localization with Superpixel Neighborhoods, IEEE 12th International Conference on Computer Vision, 2009. 41
43

[12] B. Fulkerson and S. Soatto, Really quick shift: Image segmentation on a GPU, Proceeding ECCV'10 Proceedings of the 11th European conference on Trends and Topics in Computer Vision, pp. 350-358, 2010. 45

[13] A. Hagan and Y. Zhao, Parallel 3D Image Segmentation of Large Data Sets on a GPU Cluster, International Symposium on Visual Computing, pp. 960-969, 2009. 47

- 1 [14] T. Kanunogo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, An
Efficient k-Means Clustering Algorithm: Analysis and Implementation, IEEE Transactions on
3 Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.
- [15] A. Korbes, G. Vitor, R. Lotufo and J. Ferreira, Analysis of a step-based
5 watershed algorithm using CUDA, International Journal of Natural Computing Research,
2010.
- 7 [16] C. Lea, M. Flynn, R. Vidai, A. Reiter, G. Hager, Temporal Convolutional Networks
for Action Segmentation and Detection, The IEEE Conference on Computer Vision and
9 Pattern Recognition (CVPR), 2017.
- [17] A. Levinstein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson and K.
11 Siddiqi, TurboPixels: Fast superpixels using geometric flows, Transactions on Pattern
Analysis and Machine Intelligence, pp. 2290-2297, 2009.
- 13 [18] L. Li, J. Yao, J. Tu, X. Lu, K. Li, and Y. Liu, Edge-Based Split-and-Merge
Superpixel Segmentation, IEEE International Conference on Information and Automation,
15 2015.
- [19] Z. Li, J. Chen, Superpixel Segmentation using Linear Spectral Clustering, IEEE
17 Conference on Computer Vision and Pattern Recognition, 2015.
- [20] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic
19 Segmentation, Computer Vision and Pattern Recognition, 2015.
- [21] M. McGuire, M. Mara, Efficient GPU Screen-Space Ray Tracing, Journal of
21 Computer Graphics Techniques, Efficient GPU Screen-Space Ray Tracing, Vol. 3, No. 4,
2014.
- 23 [22] A. Mustafa, and A. Hilton, Semantically Coherent Co-Segmentation and
Reconstruction of Dynamic Scenes, The IEEE Conference on Computer Vision and Pattern
25 Recognition (CVPR), 2017.
- [23] S. Paris, Edge-preserving Smoothing and Mean-shift Segmentation of Video
27 Streams, Proceeding ECCV '08 Proceedings of the 10th European Conference on Computer
Vision: Part II, pp. 460-473, 2008.
- 29 [24] E. Ramirez, P. Temoche, R. Carmona, A volume segmentation approach based
on GrabCut, CLEI Electronic journal, vol. 16, 2013.
- 31 [25] M. Roberts, J. Packer, M. Sousa and J. Mitchell, A Work-Efficient GPU Algorithm
for Level Set Segmentation, High Performance Graphics, 2010.
- 33 [26] S. Schenke, B. Wiinsche, J. Denzler, GPU Based Volume Segmentation, In Proc.
of IVCNZ '05, 2005.
- 35 [27] J. Shi and J. Malik, Normalized Cuts, IEEE Transactions on Pattern Analysis and
Machine Intelligence, vol. 22, no. 8, 2000.
- 37 [28] P. Soille, Constrained Connectivity for Hierarchical Image Partitioning and
Simplification, IEEE transactions on pattern analysis and machine intelligence, vol. 30, no.
39 7, 2008.
- [29] V. Vineet and P. Narayanan, CUDA Cuts: Fast Graph Cuts on the GPU, 2008.
- 41 [30] H. Zhu, F. Meng, J. Cai and S. Lu, Beyond pixels: A comprehensive survey from
bottom-up to semantic image segmentation and cosegmentation, Journal of Visual
43 Communication and Image Representation, 34, pp. 12-27, 2016.
- [31] A. Abramov, T. Kulvicius, F. Worgotter, and B. Dellen, Real-time image
45 segmentation on a GPU, Facing the multicore-challenge, pp 131-142, 2010.
- [32] <https://soundofvision.net/>
- 47 [33] S. Caraiman, A. Morar, M. Owczarek, A. Burlacu, D. Rzeszotarski, N. Botezatu,
P. Herghelegiu, F. Moldoveanu, P. Strumillo, A. Moldoveanu, Computer vision for the visually
49 impaired: the Sound of Vision system, IEEE International Conference on Computer Vision
Workshops (ICCVW), pp. 1480-1489, 2017.

RO 133585 B1

Revendicări

	1
1. Metodă, implementată cu ajutorul calculatorului, de segmentare a imaginilor video și a matricilor multi-dimensionale în timp real, bazată pe trasare de raze, caracterizată prin aceea că , urmează succesiunea de etape:	3
- Etapa 1: Calcularea unui flux peste matricea de intrare, care descrie rata de schimbare a valorii unui element relativ la vecinătatea acelui element, flux estimat prin diferențe finite).	5
- Etapa 2: Împărțirea matricii de intrare în sub-regiuni egale și alegerea unui punct germen pentru fiecare sub-regiune, punct germen ce acoperă cât mai omogen spațiul setului de date de intrare.	7
- Etapa 3: Trasarea de raze multiple plecând de la puncte germen, garantând o acoperire eficientă a întregului set de date, dar cu un număr foarte mic de eșantione, minimizând costurile explorării spațiului de căutare, fiecare rază conținând statistici despre elementele din spațiul traversat.	9
- Etapa 4: Conectarea punctelor germen și implicit a razelor care aparțin de acele puncte germen, în momentul în care razele trasate se intersectează, conectarea realizându-se numai în condiții de similaritate între razele care se intersectează, dacă elementele din spațiile parcurse de raze au proprietăți similare, la conectarea a două puncte germen, unul dintre ele devine părintele celuilalt punct germen și astfel punctele germen conectate prin raze sunt unite, rezultând în final un set de puncte germen părinte care partiționează matricea de intrare în regiuni continue care nu se suprapun unele cu celelalte.	11
- Etapa 5: Extinderea regiunilor generate pentru a acoperi complet setul de date inițial, utilizând un proces de filtrare în doi pași:	13
Pasul 1: Pentru fiecare element ne-etichetat, se trimit raze care eșantionează spațiulm, iar atunci când o rază întâlnește o regiune, dacă proprietățile razei sunt asemănătoare cu cele ale regiunii, elementul ne-etichetat primește id-ul regiunii respective.	15
Pasul 2: Elementele care nu au fost etichetate nici în pasul 1 și care nu se află pe zone de flux mare, sunt prelucrate prin eșantionarea regiunilor care se află în vecinătatea lor, elementul ne-etichetat primește id-ul regiunii din vecinătate care are proprietățile cele mai similare cu acelea ale elementului respectiv.	17
- Etapa 6: Unirea a sub-seturilor generate, folosind un proces iterativ de calcul de statistici în interiorul regiunilor și de unificare a regiunilor vecine pe baza acestor statistici, dacă aceste două regiuni au proprietăți similare.	19
2. Sistem pentru segmentarea imaginilor video în timp real, bazat pe trasare de raze, capabil să realizeze operații de segmentare a imaginilor și matricilor multi-dimensionale printr-o eșantionare rapidă a spațiului setului de date de intrare, în detrimentul acoperirii tuturor punctelor dintr-o vecinătate locală, alcătuit dintr-un dispozitiv de achiziție sau de generare a matricilor bi- sau multi-dimensionale, un modul opțional de pre-procesare a datelor de intrare, un modul de post-procesare care utilizează regiunile disjuncte în funcție de scopul unei aplicații, caracterizat prin aceea că , mai conține un modul de segmentare rară prin trasarea de raze care obține la ieșire regiuni continue disjuncte, prin metoda de segmentare de la revendicarea 1.	21
	23
	25
	27
	29
	31
	33
	35
	37
	39
	41
	43

(51) Int.Cl.

G06T 7/12 (2017.01);

G06T 15/06 (2011.01)

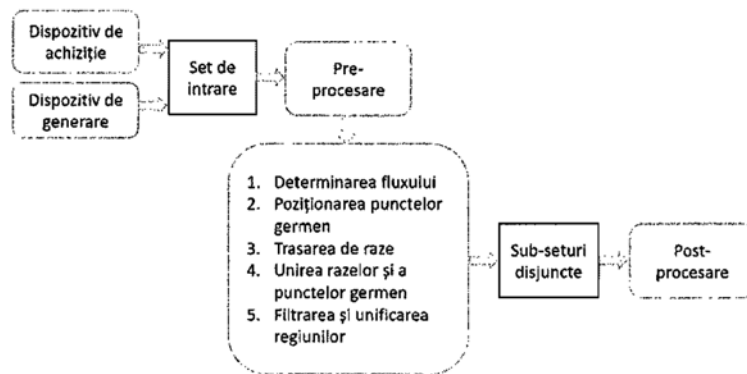


Fig. 1

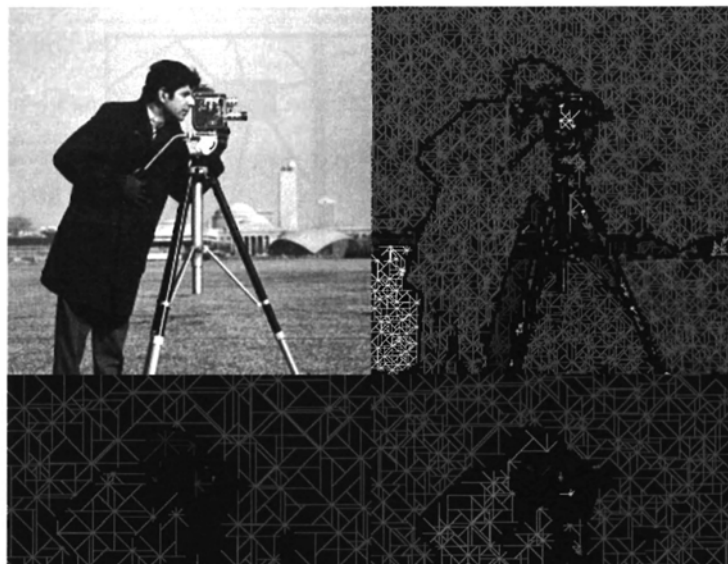


Fig. 2

(51) Int.Cl.
G06T 7/12 (2017.01);
G06T 15/06 (2011.01)

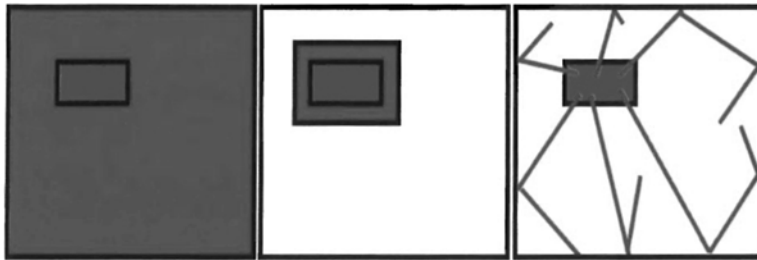


Fig. 3

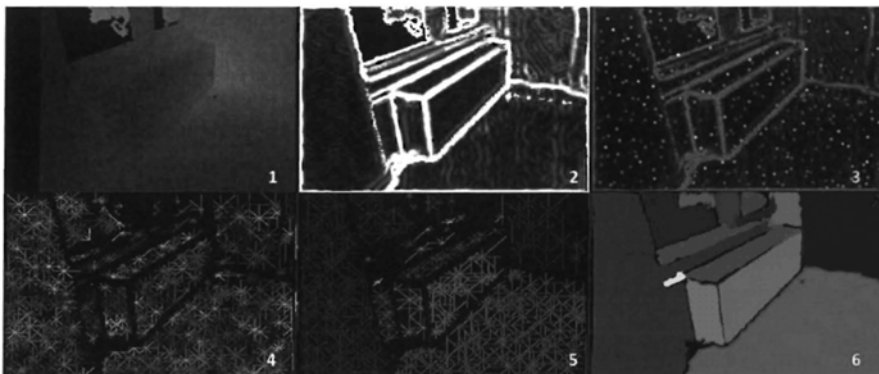


Fig. 4

(51) Int.Cl.

G06T 7/12 (2017.01);

G06T 15/06 (2011.01)

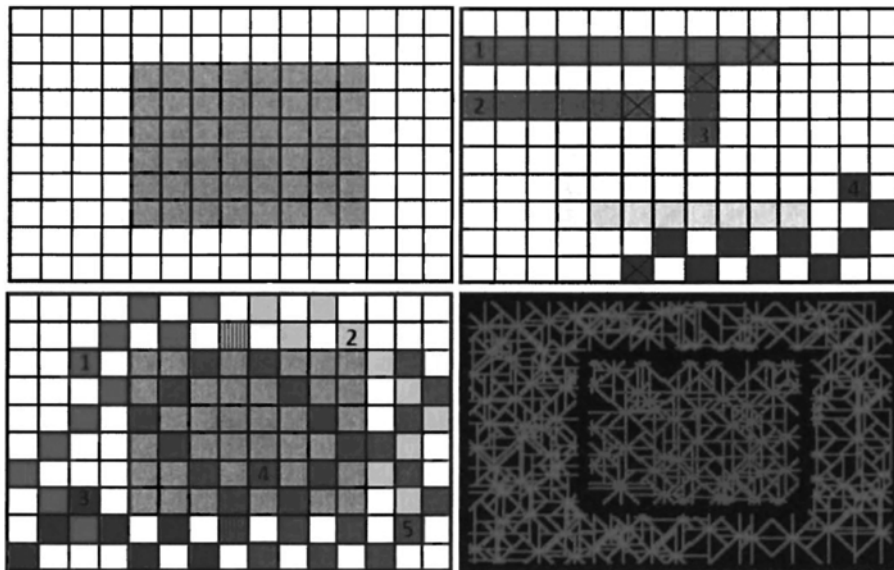


Fig. 5

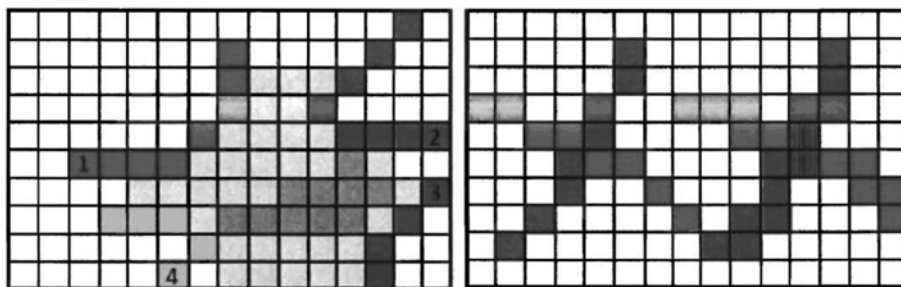


Fig. 6

(51) Int.Cl.
G06T 7/12 (2017.01),
G06T 15/06 (2011.01)

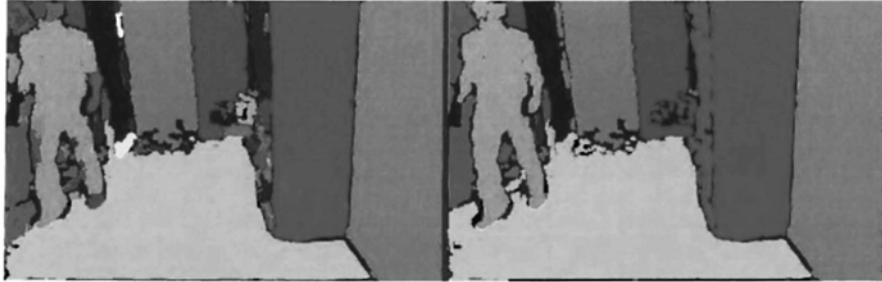


Fig. 7

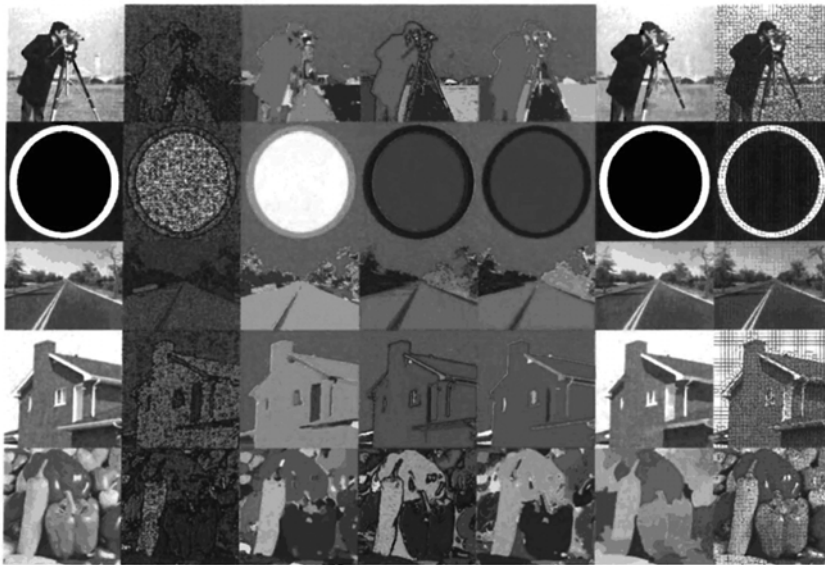


Fig. 8

(51) Int.Cl.

G06T 7/12 (2017.01);

G06T 15/06 (2011.01)

Line	file	net (i*)	opt. filter (i*)	opt. merging (i*)	FULL (i*)	SPARSE (i*)	opt (i**)	opt. filter (i**)	opt. merging (i**)	FULL (i**)	SPARSE (i**)	STAR S/C (i***)	STAR S/C (i****)	STAR RealyQuickSoft
256	0.182	0.523	0.505	0.803	2.013	0.705	0.172	4.482	2.323	7.359	0.354	2.183	5.001	5.075
512	0.699	1.955	1.852	2.415	6.881	2.814	0.988	4.772	1.785	8.200	1.647	5.185	20.375	21.788
758	0.999	3.551	5.581	3.688	13.68	4.46	1.293	10.741	2.693	15.618	2.201	13.38	35.159	62.422
1024	1.677	7.519	6.854	5.677	19.727	9.196	2.325	11.932	4.153	20.091	4.000	18.876	62.148	123.643
1536	3.731	17.102	10.752	12.522	44.107	20.833	4.887	28.031	9.627	43.642	8.618	53.188	123.268	307.103
2048	6.583	31.081	18.561	22.265	78.333	37.666	8.215	46.027	15.756	76.581	14.798	71.983	190.754	524.439
3072	15.061	73.241	41.094	45.606	178.925	88.142	20.777	120.555	35.111	191.464	35.778	203.093	481.791	1272.256
4096	26.375	134.754	68.741	87.251	317.141	161.119	37.115	215.502	62.437	342.429	63.49	292.329	770.726	2055.832
6144	58.718	307.877	164.715	191.020	725.287	367.571	85.722	456.909	140.367	772.709	145.437	362.215	1812.966	3673.915
8192	106.051	637.751	416.592	341.875	1302.18	543.802	135.667	857.222	267.256	1405.976	241.498	515.41	2990.072	6941.628

measures in milliseconds (*) very dense sparse net (one seed per 64x64) (**) normal density sparse net (one seed per 16x16 file) (***) S/C without connectivity (****) S/C with connectivity

Fig. 9



Fig. 10



Editare și tehnoredactare computerizată - OSIM
 Tipărit la Oficiul de Stat pentru Invenții și Mărci
 sub comanda nr. 475/2022