ROMÂNIA

(19) OFICIUL DE STAT
PENTRU INVENȚII ŞI MĂRCI
Bucureşti

(11) RO 133585 A2

(51) Int.Cl.
G06T 7/10 (2017.01)

(12) **CERERE DE BREVET DE INVENȚIE**

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

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

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

(71) Solicitant:
• **UNIVERSITATEA POLITEHNICA BUCUREŞTI**, *SPLAIUL INDEPENDENȚEI NR.313, SECTOR 6, BUCUREŞTI, B, RO*

(72) Inventatori:
• **PETRESCU LUCIAN**, *STR.BARBU DELAVRANCEA NR.2B, BL.33D, SC.A, AP.2, SECTOR 1, BUCUREŞTI, B, RO;*
• **MORAR ANCA**, *STR.OLĂNEŞTI NR.4, BL.43A, SC.1, AP.5, SECTOR 6, BUCUREŞTI, B, RO;*
• **MOLDOVEANU FLORICA**, *ALEEA BAIA DE ARIEŞ NR.5, BL.1, SC.3, AP.33, SECTOR 6, BUCUREŞTI, B, RO;*
• **MOLDOVEANU ALIN**, *ALEEA BAIA DE ARIEŞ NR.5, BL.1, SC.3, ET.4, AP.39, SECTOR 6, BUCUREŞTI, B, RO*

Această publicație include şi modificările descrierii, revendicărilor şi desenelor, depuse conform art. 35, alin. (20), din HG nr. 547/2008.

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

(57) **Rezumat:**

Invenția se referă la un sistem şi la o metodă de segmentare a imaginilor video în timp real. Sistemul conform invenției cuprinde: un dispozitiv de achiziție sau de generare a unor matrice bi- sau multi-dimensionale ca set de date de intrare, un modul opțional de preprocesare a datelor de intrare, de exemplu, prin filtrarea zgomotelor, un modul de segmentare rară prin trasarea de raze, ce scoate la ieşire regiuni continue disjuncte prin metoda de segmentare conform invenției, şi un modul de post-procesare ce utilizează regiunile disjuncte în funcție de scopul aplicației. Metoda de segmentare conform invenției cuprinde următoarea succesiune de etape: calcularea unui flux adaptiv peste setul de date de intrare, ce descrie rata de schimbare a valorii unui segment relativ la vecinătatea acelui element; construcția de generatori de subseturi, numiți şi puncte germen de segmentare, într-un mod pseudo-aleatoriu, folosind secvențe de tip Sobol, Hammersly şi Van der Corput, sau distribuții Poisson; trasarea de raze multiple plecând de la generatorii de subseturi, trasarea garantând o acoperire eficientă a întregului set de date, dar cu un număr foarte mic de eşantioane, minimizând costurile explorării spațiului de căutare; conectarea generatorilor de subseturi în momentul în care razele trasate anterior se intersectează; extinderea, opțională, a subseturilor generate, pentru a acoperi complet setul de date inițial, utilizând un proces iterativ de filtrare; unirea, opțională, a subseturilor generate, folosind un proces iterativ de calcul de statistici şi unificare pe baza statisticilor.

Revendicări inițiale: 1
Revendicări amendate: 2
Figuri: 10

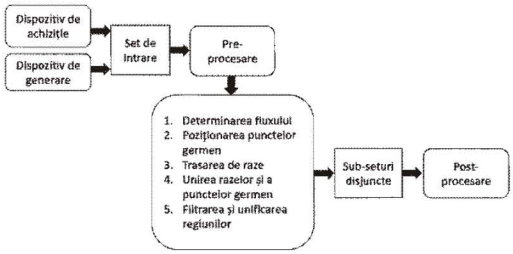

Fig. 1

RO 133585 A2

DESCRIEREA INVENȚIEI

SISTEM PENTRU SEGMENTAREA IMAGINILOR VIDEO IN TIMP REAL, BAZAT PE TRASARE DE RAZE

Sistemul propus (inventia) are la bază un algoritm original de segmentare (partitionare) a câmpurilor K-dimensionale (exemplu: imagini bidimensionale), care asigura o viteza la executie de aproape 5 ori mai mare decât cea realizata de sistemele existente, după cum se poate observa în Figura 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.

Algoritmul a fost dezvoltat pentru Sound of Vision [32], un sistem asistiv pentru nevăzători care scanează mediul înconjurător prin camere video, identifică elementele de interes din imaginile obtinute și emite semnale audio și haptice care permit utlizatorului să perceapă mediul înconjurător, ajutandu-l să navigheze în medii necunoscute. Cu toate că în sistemul Sound of Vision algoritmul de segmentare partiționează imagini bidimensionale, modelul său matematic poate fi folosit pentru partiționarea de seturi de date K-dimensionale. Figura 1 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 adândcime și calculează o hartă de normale). Urmează segmentarea cu trasare de raze, care scoate la ieșire sub-seturi disjuncte. Post-procesarea poate fi orice modul care utilizează sub-seturile 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).

Inventia poate fi aplicată in domeniul informatic medical (detectare automată de tumori din scanări CT/ RMN, sisteme asistive), roboți industriali (programele de decizie ale roboților autonomi industriali și a dronelor folosesc segmentare), industria automobilelor (segmentarea este folosită de mașini cu conducere automată pentru percepția elementelor de trafic), securitate (urmărirea automată pe camere de securitate folosește segmentare, detecția automată a elementelor de interes folosește segmentare), etc.

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 [13][14][19], compresie [30], histograme [30], detecția muchiilor [30][18], extinderea regiunilor [26][3][14][5], partiționarea grafurilor [10][27][29][30], transformarea watershed [15][30],

praguri adaptive [5], divizarea și unificarea [18][30], "Random Walker" [9], contururi ierarhice [30][28] și active [30].

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

Deoarece costul computațional reprezintă un aspect critic în ce privește utilizarea algoritmilor, o altă tendință recentă este de a exploata paralelismul hardware în vederea maximizării vitezei. Această tendință a condus la dezvoltarea unor metode de segmentare/pre-segmentare iterative rapide utilizând GPU [1][26]**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**. Metodele de pre-segmentare [12][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 costul de clusterizare finala, acestea produc cele mai rapide rezultate locale dintre toate metodele de segmentare. Totusi, chiar și aceste metode rapide necesită procese iterative.

In 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. Segmentarea rapidă este esențială pentru urmărirea și detecția obiectelor, navigarea și percepția pentru roboți autonomi, dispozitive asistive pentru nevăzători sau dispozitive medicale controlate automat.

Aceasta propunere de brevet introduce un sistem ce include o nouă metodă de segmentare, bazata pe extinderea regiunilor prin trasarea de raze. Este o metoda de segmentare imperfectă rară care etichetează numai o fracțiune din setul de intrare în vederea minimizării costului computational. Figura 2 prezintă ieșirea sistemului având ca 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. Chiar cu un grad de risipire ridicat, segmentarea rezultată este suficient de calitativă pentru numeroase aplicații în timp real

Metoda este proiectată astfel încât să utilizeze la maximum paralelismul hardware al plăcilor grafice actuale și să profite de pe urma unor optimizări in trasarea razelor („packet tracing" [6], DDA [21]). În acest brevet, metoda este exemplificată în 2D, dar poate fi implementată în oricâte dimensiuni.

Segmentarea imperfectă rară prin trasarea de raze aduce următoarele contribuții:

- extinderea de regiuni pe baza trasării de raze conduce la o segmentare imperfectă, prioritizând acoperirea rapidă a spațiului setului de date de intrare în detrimentul acoperirii tuturor punctelor dintr-o vecinătate locală
- un număr constant de treceri prin banda grafică, conducând la un timp de execuție care nu depinde de tipul imaginii, fără a afecta calitatea
- lățime de bandă scăzută datorită rarității razelor, care determină un număr mic de accese la memorie
- o complexitate de $O(n/tsize)$ unde n este dimensiunea setului de intrare iar *tsize* este dimensiuniea ferestrei pentru care este generat un punct germen de pornire a razelor (exemplu: 16x16). Segmentările cu cele mai bune rezultate, SLIC [2] și ReallyQuickShift [12] au complexități de $O(n)$ *și* $O(d*n^2)$.

Segmentarea imperfectă rară prin trasarea de raze este sumar descrisă în Figura 4. Aceasta este o componentă cheie pentru proiectul Sound of Vision [32], unde este utilizată pentru segmentarea imaginilor de adâncime în timp real, pe plăci grafice uzuale.

Stadiul actual al dezvoltării în segmentarea imaginilor:

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

Majoritatea metodelor inițiale de segmentare pe GPU au fost realizate în scopul utilizării în medicină, unde accelerarea GPU a adus beneficii considerabile în manipularea seturilor de date tridimensionale foarte mari. Shenke 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. 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 $O(n*log(n))$. Körbes și alții [15] au introdus un algoritm "watershed" paralel iterativ, în care o imagine este divizată în ferestre de dimensiune 16x16 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 computational scăzut. Ramirez și alții [23] 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 executata 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), Conditional Random Fields (CRF) sau Fully Convoluted 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 algotimilor 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 $O(d*n^2)$, 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.

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 strategie de clusterizare

bazată pe ascensiunea gradientului și oferă o calitate a segmentării rezonabilă pentru un timp de rulare foarte scăzut.

Invenția:

Segmentarea K-dimensională este o problemă de partiționare în care un set S care conține n elemente K-dimensionale trebuie să fie împărțit în seturi disjuncte, pe baza unor proprietăți de similaritate locală sau globală între elemente. Această problemă este rezolvată prin căutarea sub-seturilor disjuncte în S și determinarea celei mai bune soluții.

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 Figura 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.

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

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

Fluxul reprezintă rata de schimbare a setului de intrare. Poate fi un simplu gradient adaptiv pentru imagini color/în niveluri de gri și este extrem de ieftin de calculat pe GPU. Pentru imagini de adâncime (de exemplu, cele obținute de la dispozitivul de achiziție Sound of Vision) s-a utilizat un gradient adaptiv care ține cont de eroarea de estimare a senzorului de adâncime. Un exemplu de flux se poate observa în Figura 4.2.

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

```
seedPos ← getPseudoRandomPosition()
iterations ← 0
flux ← getFlux(seedPos)
while flux > Threshold and iterations <MaxIterations
        seedPos ← seedPos + fluxDirection
```

```
6:          iterations ← iterations + 1
7: return seedPos
```

Un exemplu al distribuției rezultate a punctelor germen poate fi observată în Figura 4.3, colorată cu o funcție de hash Jenkins aplicată peste indicii punctelor germen.

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

Razele suportă de asemenea reflexii pe marginile imaginii sau în zone de flux ridicat (muchii), reprezentând exploratori eficienți chiar și pentru cazuri complicate. Razele numeroase sunt cu ușurință distribuite pe toate nucleele de procesare disponibile. Rezultatul razelor trasate dar neconectate se poate observa în Figura 4.4.

O structură arborescentă este creată peste punctele germen, care leagă toate punctele de plecare ale întregului set de date. Inițial toate punctele germen pornesc ca arbori separați. Cu fiecare conexiune între raze, numărul de arbori separați scade. În vederea minimizării înălțimii arborilor atunci când două raze se întâlnesc, numai punctele germen sunt conectate. Deoarece sincronizarea datelor pe GPU poate fi implementată eficient numai prin operații atomice, trebuie stabilită o ierarhie atunci când două puncte germen sunt conectate. Noi am utilizat indicele minim al punctelor care trebuie conectate. Următorul algoritm este utilizat pentru conectarea a două pucte germen:

```
GetGreatestParent(seed)
1: seed ← initial seed
2: do seed ← getParent(seed)
3:        while seed not null
4: return seed

ConnectSeeds(seed1, seed2)
1: parent1 ← getGreatestParent(seed1)
2: parent2 ← getGreatestParent(seed2)
3: if parent1 < parent2
4:          atomicExchange(parent2.parent, parent1)
5:else
6:          atomicExchange(parent1.parent, parent2)
```

Fiecare rază menține un set de proprietăți legate de spațiul traversat, incluzând valoarea medie a punctelor, varianța și valoarea ultimelor T puncte traversate. Proprietățile sunt salvate într-un buffer auxiliar. O conexiune potențială între puncte germen are loc atunci când o rază întâlnește puncte traversate de altă rază. Conexiunea este realizată numai dacă există potrivire locală și globală între cele două puncte germen. Conexiunea are loc numai dacă proprietățile celor două raze sunt similare. Următorul pseudocod descrie acest proces:

```
Initialization Pass:
1:      initialize label matrix to empty
2: initialize trace matrix to empty

Ray Tracing Pass (1 thread per each ray per seed):
1: ray ← the initial ray id
2: seed ← getSeed (ray)
3: properties ← initializeRayProperties()
4: iterations ← 0
5: pos ← seed.pos
6: while iterations < MaxIterations:
7:              flux ← readFlux (pos)
8:              if flux > MaxFlux
9:                      end
10:     input ← readInput (pos)
11:     updateProperties(properties, input)
12:     if rayPropertiesDeviateFromCluster(properties)
13:             end
14:     scatterDir ← getScatteringBasedOnAccumFlux()
15:     id ← readLabel (pos)
16:     if id is empty
17:             writeLabel (pos, seed)
18:     else
19:             parentid ← getGreatestParent(seed)
20:             parentid2 ← getGreatestParent(id)
21:             if degenerateParents(id, seed, parentid, parentid2)
22:                     end
23:             properties2 ← readProperties(pos)
24:             if properties1 similar properties2
25:                     updateProperties(properties1)
26:             writeTraceProperties(pos, properties1)
27:             ConnectSeeds(seed, id)
28:     if ray.dir aligned to image axes
29:             pos ← pos + ray.dir + scatterDir
30:     else
31:             pos ← staircase tracing on (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 Figura 6. 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 Figura 6.

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

structura arborescentă construită peste imaginie, reprezintă id-urile de segmentare unice. Pseudocodul pentru forțarea conectivității este urmatorul:

Enforce Connectivity Pass (per each pixel):
1: pos ← pixel
2: seed ← readLabel (pos)
3: properties ← readProperties(pos)
4: flux ← readFlux(pos)
5: ***for*** *pos2 neighbor of pos*
6: *seed2 ← readLabel(pos2)*
7: *properties2 ← readProperties(pos2)*
8: *flux2 ← readFlux(pos)*
9: ***if*** *properties* ***similar*** *properties2*
10: ***if*** *max(flux, flux2) < MaxFlux*
11: ***ConnectSeeds****(seed, seed2)*

Numărul de operații de acces al memoriei 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.

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

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.

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.

Pseudocodul pentru filtrarea exacta este următorul:

Exact Filtering (per each pixel):
1: pos ← pixel
2: ***if*** *readLabel(pos)* ***not*** *empty*
3: ***end***
4: steps ← 0
5: ***for*** *ray = 1, ray < NumRays, ray++*
6: *alive[ray] ← true*
7: *properties[ray] ← empty*
8: ***for*** *steps = 1, steps < MaxSteps, steps++*
9: ***for*** *ray = 1, ray < NumRays, ray++*
10: ***if not*** *alive[ray]*

```
11:                 continue
12:         if ray.dir aligned to image axes
13:                 rpos ← pos + steps * ray.dir
14:         else
15:                 rpos ← staircase tracing on ray.dir
16:         flux ← readFlux(rpos)
17:         if flux > MaxFlux
18:                 alive[ray] ← false
19:                 continue
20:         input ← readInput(rpos)
21:         updateProperties(properties[ray], input)
22:         seed ← readLabel(rpos)
23:         if seed not empty
24:                 properties2 ← readProperties(rpos)
24:                 if properties[ray] similar properties2
25:                         writeLabel(pos, seed)
26:                         end
27:                 else
28:                         alive[ray] ← 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.

Unificarea regiunilor este un alt proces opțional, care poate conduce la îmbunătățiri calitative semnificative, după cum se poate observa în Figura 7. Unificarea este de asemenea un process 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 etichetarea locală. Apoi, în etapa a doua se acumulează informații în părintele final al fiecărui punct 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 proprietățile se potrivesc dar etichetarea este diferită, atunci cele două etichete sunt unificate.

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 ultimă oră de (pre)segmentare, după cum este ilustrat în Figura 9.

Evaluarea invenției în comparație cu metodele deja existente de segmentare:

Metodologia de evaluare a rezultatelor compară segmentarea imperfectă rară cu trasare de raze atât calitativ cât și din punct de vedere al vitezei cu cele mai rapide metode de segmentare pe GPU, cu ReallyQuickShift **Error! Reference source not found.** și SLICO **Error! Reference source not found.**. În timp ce acestea sunt metode de pre-segmentare și necesită procesare suplimentară pentru producerea de segmentări complete, ele reprezintă singura categorie de 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 rapidă decât orice algoritm complet de segmentare pe GPU, cel mai rapid având complexitatea *O(n*log n)*.

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

Figura 8 prezintă rezultatele calitative ale segmentării imperfecte prin trasarea de raze, unde diferite imagini sunt segmentate și comparate cu segmentări consacrate. Testarea calitativă 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 afectează direct numărul de puncte germen. Astfel, a patra coloană din Figura 8 necesită mărire, deoarece rezultatele sunt obținute cu o valoare de raritate/risipire foarte mică.

Figura 9 prezintă rezultatele de viteză, unde segmentarea imperfectă bazată pe trasarea de raze are în mod consistent timpi de rulare de aproximativ 5 ori mai mari decât SLIC și de 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 filtrare. Din Figura 9 se poate observa de asemenea că metoda prezentată are o degradare sub-liniară a performației în ce privește numărul de pixeli, fiind ideală pentru seturi de date mari și foarte mari.

Segmentarea imperfectă bazată pe trasarea de raze este o metodă flexibilă, unde compromisul 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 Figura 10. Pre-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 care partiționează astfel de segmente.

O limitare a abordării noastre este dependența de calculul fluxului, care poate conține 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).

Segmentarea imperfectă 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 rimp 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.

Referinte:

[1] A. Abramov, T. Kulvicius, F. Wörgötter, and B. Dellen, Real-time image segmentation on a GPU, Facing the multicore-challenge, pp 131-142, 2010.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, SLIC Superpixels, EPFL Technical Report no. 149300, 2010.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, 2012.

[4] 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.

[5] 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.

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

[7] 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

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

[9] 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

[10] 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

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

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

[13] 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.

[14] 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.

[15] T. Kanunogo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, An Efficient k-Means Clustering Algorithm : Analysis and Implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.

[16] A. Körbes, G. Vitor, R. Lotufo and J. Ferreira, Analysis of a step-based watershed algorithm using CUDA, International Journal of Natural Computing Research, 2010.

[17] C. Lea, M. Flynn, R. Vidal, A. Reiter, G. Hager, Temporal Convolutional Networks for Action Segmentation and Detection, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[18] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson and K. Siddiqi, TurboPixels: Fast superpixels using geometric flows, Transactions on Pattern Analysis and Machine Intelligence, pp. 2290–2297, 2009.

[19] 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, 2015

[20] Z. Li, J. Chen, Superpixel Segmentation using Linear Spectral Clustering, IEEE Conference on Computer Vision and Pattern Recognition, 2015

[21] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, Computer Vision and Pattern Recognition, 2015.

[22] M. McGuire, M. Mara, Efficient GPU Screen-Space Ray Tracing, Journal of Computer Graphics Techniques, Efficient GPU Screen-Space Ray Tracing, Vol. 3, No. 4, 2014

[23] A. Mustafa, and A. Hilton, Semantically Coherent Co-Segmentation and Reconstruction of Dynamic Scenes, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[24] S. Paris, Edge-preserving Smoothing and Mean-shift Segmentation of Video Streams, Proceeding ECCV '08 Proceedings of the 10th European Conference on Computer Vision: Part II, pp 460-473, 2008.

[25] E. Ramírez, P. Temoche, R. Carmona, A volume segmentation approach based on GrabCut, CLEI Electronic journal, vol. 16, 2013

[26] M. Roberts, J. Packer, M. Sousa and J. Mitchell, A Work-Efficient GPU Algorithm for Level Set Segmentation, High Performance Graphics, 2010.

[27] S. Schenke, B. Wünsche, J. Denzler, GPU Based Volume Segmentation, In Proc. of IVCNZ '05, 2005.

[28] J. Shi and J. Malik, Normalized Cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000.

[29] P. Soille, Constrained Connectivity for Hierarchical Image Partitioning and Simplification, IEEE transactions on pattern analysis and machine intelligence, vol. 30, no. 7, 2008.

[30] V. Vineet and P. Narayanan, CUDA Cuts: Fast Graph Cuts on the GPU, 2008.

[31] 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 Communication and Image Representation, 34, pp 12-27, 2016

[32] https://soundofvision.net/

REVENDICARI

Se revendică sistemul ce conține metoda de segmentare cu trasare de raze pe placa grafica, în sensul creşterii performanței operației de segmentare a imaginilor (şi a seturilor de date multi-dimensionale), ce constă din următoarele operații executate în mod repetat:

- Calcularea unui flux adaptiv peste setul de date de intrare, care descrie rata de schimbare a unui element relativ la vecinătatea acelui element în setul inițial.

- Construcția de generatori de sub-seturi, numiți şi puncte germen de segmentare, într-un mod pseudo-aleator folosind secvențe de tipul Sobol, Hammersly si Van der Corput sau distribuții Poisson. După construcție, generatorii sunt mutați iterativ departe de zonele de flux mare, pentru a maximiza şansele de generare de sub-seturi din zone cu flux scăzut, adică zone în care similaritatea dintre vecini este mare, maximizând şansa lor de a fi în acelaşi sub-set de segmentare.

- Trasarea de raze multiple plecând de la generatorii de sub-seturi. Această trasare garantează o acoperire eficientă a întregului set de date, dar cu un numar foarte mic de eşantione, minimizând costurile explorării spațiului de căutare. Trasarea este făcută cu rasterizare conservativă bazată pe algoritmul DDA modificat.

- Conectarea generatorilor de sub-seturi în momentul în care razele trasate anterior se intersectează. Conectarea se face doar în condiții de similaritate între razele ce se intersectează. Astfel puncteşe germen conectate prin raze sunt unite, rezultând în final doar un set de generatori unici ce partiționează setul de date inițial în sub-seturi care nu se suprapun unele cu celelalte.

- Extinderea opțională a sub-seturilor generate pentru a acoperi complet setul de date inițial, utilizând un proces iterativ de filtrare.

- Unirea opțională a sub-seturilor generate, folosind un proces iterativ de calcul de statistici şi unificare pe baza statisticilor.

DESENE EXPLICATIVE

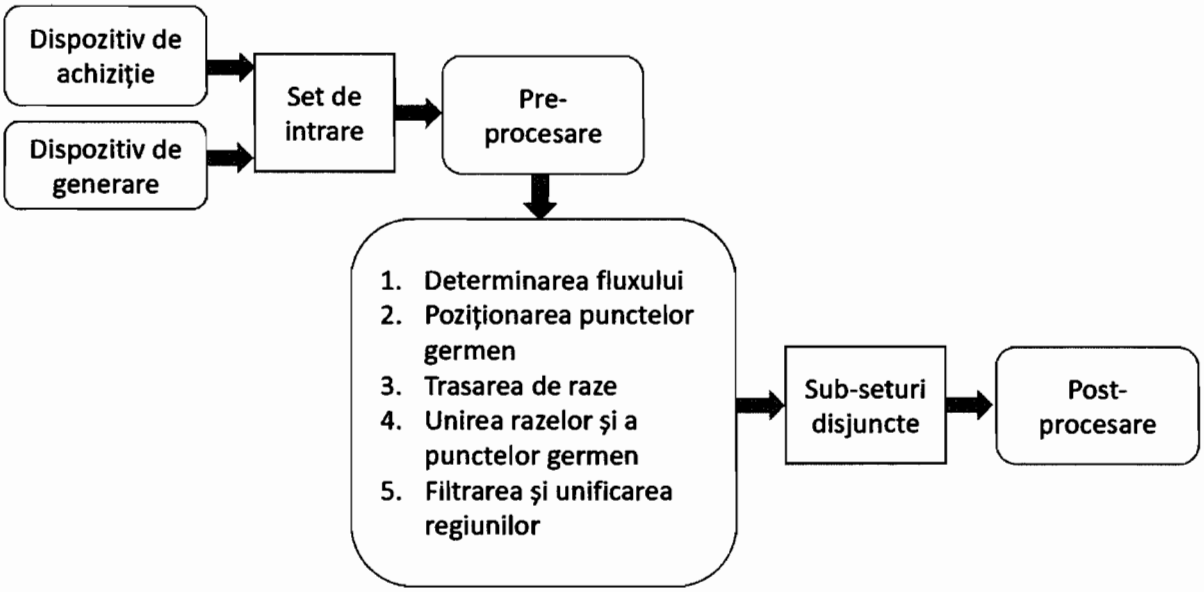

Figura 1. Schema modului de funcţionare a sistemului. Setul de intrare este produs fie de un dispozitiv de achiziţie (exemplu: camera prototipului Sound of Vision, sau un dispozitiv CT/ RMN) fie de o metodă de generare de date (exemplu: prin calcule ştiinţifice). Un modul de pre-procesare prelucrează 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 adândcime şi calculează o hartă de normale). Urmează segmentarea cu trasare de raze, care scoate la ieşire sub-seturi disjuncte. Post-procesarea poate fi orice modul care utilizează sub-seturile disjuncte, pentru identificarea şi urmărirea de obiecte (exemplu: în sistemul Sound of Vision modulul de post-procesare identifică obiecte de interes în mediu, cum ar fi podeaua, pereţii, tavanul, obstacolele, scările şi le interpretează în semnale audio şi haptice)

Figura 2. Segmentarea imperfectă rară cu trasarea de raze, cu diferite niveluri de precizie prin diverse configurări ale gradului de risipire a razelor

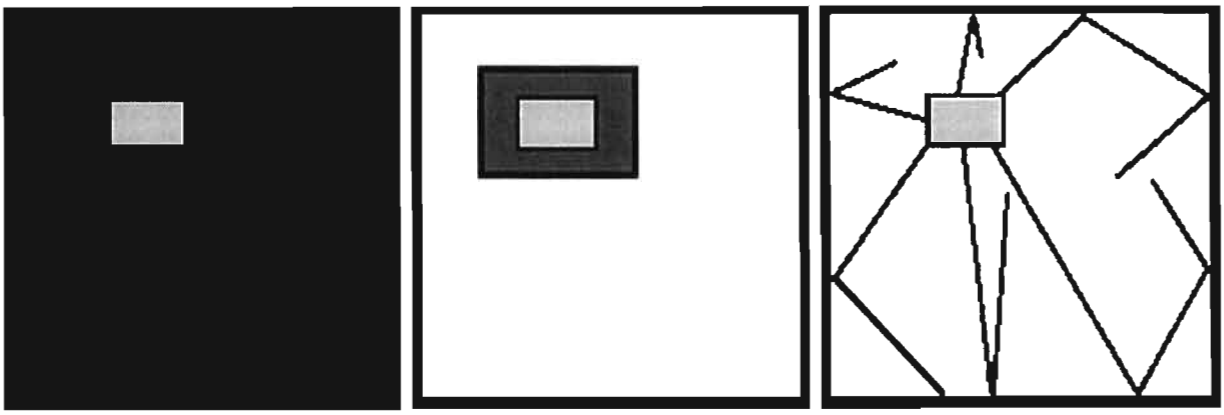

Figura 3. Strategii de explorare a spațiului în 2D. Pixelii colorați în gri deschis reprezintă clusterul curent și pixelii colorați în gri închis reprezintă spațiul de căutare pentru potențiale elemente noi ale clusterului. Strategiile tradiționale (k-means, în stânga) caută întreaga imagine. Metodele bazate pe superpixeli (în centru) caută numai într-o vecinătate mica. Segmentarea cu trasare de raze (dreapta) caută întreaga imagine, dar cu foarte puține eșantioane, concentrându-se pe similaritate globală în detrimentul similarității locale

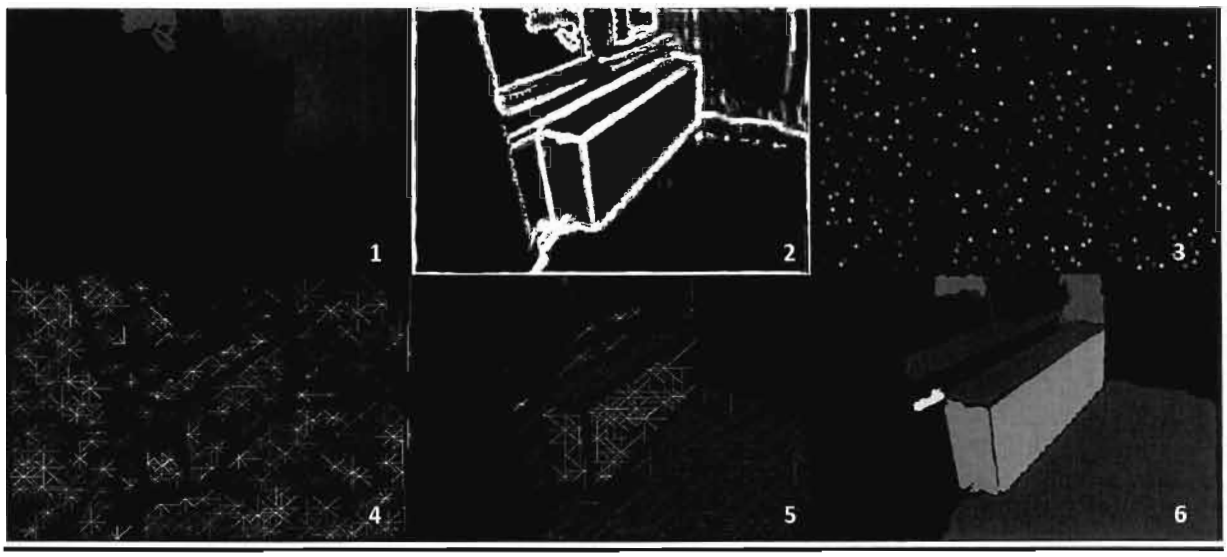

Figura 4. Etapele segmentării cu trasare de raze sunt ilustrate aici pe o hartă de adâncime. Imaginea de intrare este prezentată în 1. Fluxul (rata de schimbare a intrării) calculată pe imaginea de adâncime este ilustrată în 2. Punctele germen sunt prezentate (exagerat) în 3. Rețeaua creată pentru fiecare generator este ilustrată în 4. În 5 rețelele pentru fiecare punct germen sunt unite în rețele conectate. În 6 elementele neetichetate sunt determinate prin filtrare opțională și prin unificarea regiunilor.

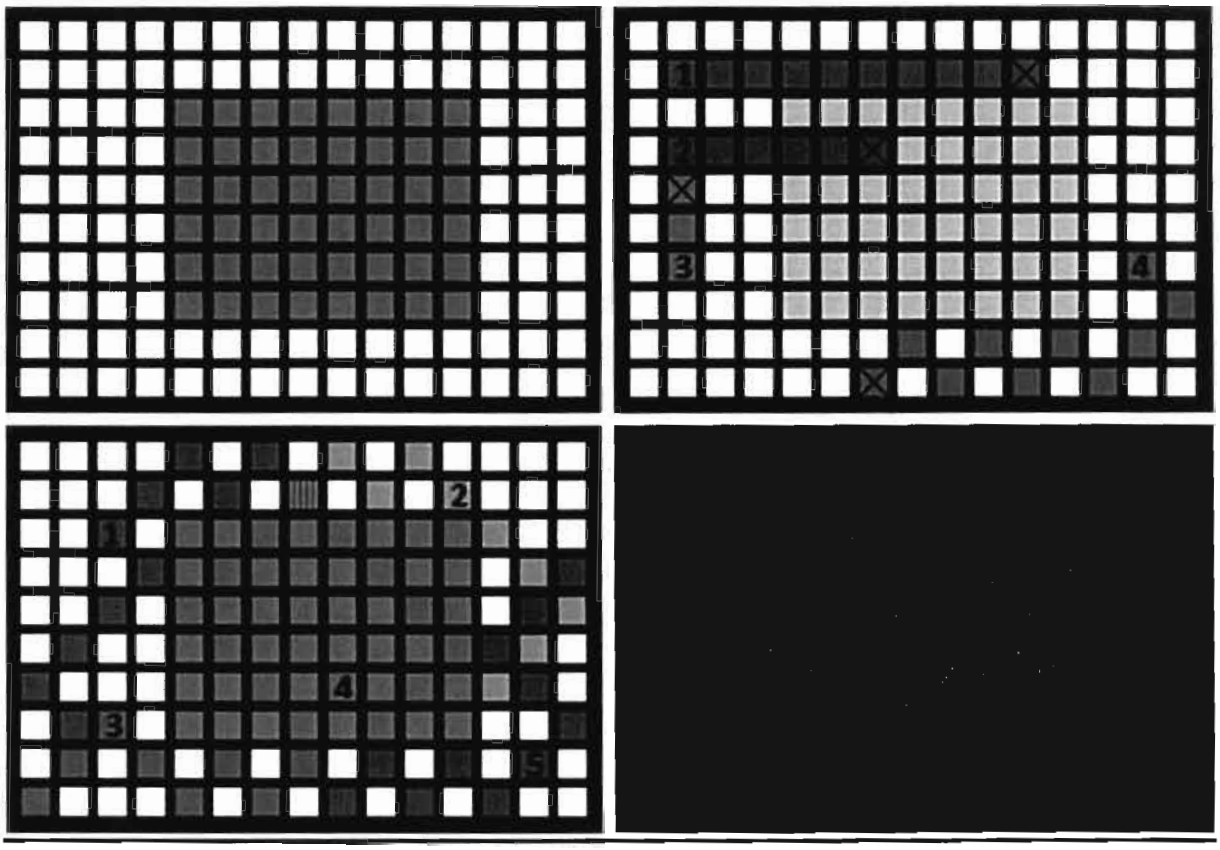

Figura 5. Imaginea din stânga sus prezintă imaginea de intrare şi pixelii, cu un obiect în mijloc. Imaginea din stânga jos ilustrează rezultatul aplicării algoritmului pe imaginea de intrare (2 raze pentru fiecare punct germen, pentru simplitate). Imainea din dreapta jos arată rezultatul segmentării rare pe o imagine de rezoluţie mare. Imaginea din dreapta sus prezintă diferite scenarii de trasare a razelor pe o imagine cu flux scăzut: raza 1 şi 4 se termină după un număr maxim de paşi (10 în acest caz), raza 2 se termină din cauza fluxului acumulat iar raza 3 se termină la întâlnirea unei alte raze.

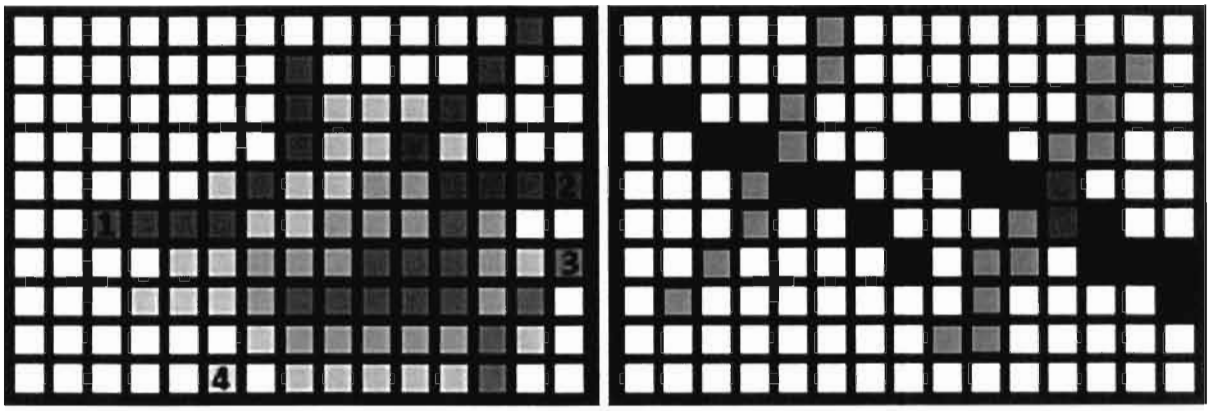

Figura 6. Imaginea din stânga ilustrează un flux de densitate variabilă, unde diverse raze suferă un proces de împrăștiere, în funcție de puterea fluxului. Acest process este utilizat pentru ghidarea razelor către regiuni de flux mare. Imaginea din dreapta ilustrează diferența dintre utilizarea unei rasterizări neoconservative sau conservative pentru trasarea razelor. Rasterizarea neconservativă a razelor poate pierde unele cazuri de contact

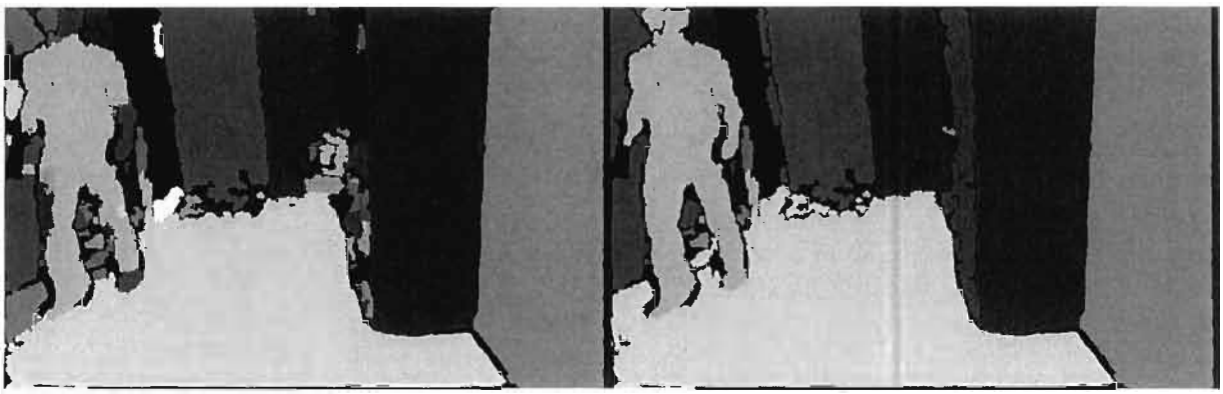

Figura 7. Unificarea regiunilor este utilizată pentru combinarea regiunilor similare care nu au putut fi unite din cauza variațiilor mare de flux. Acest pas este optional, dar îmbunătățcștc calitatea segmentării

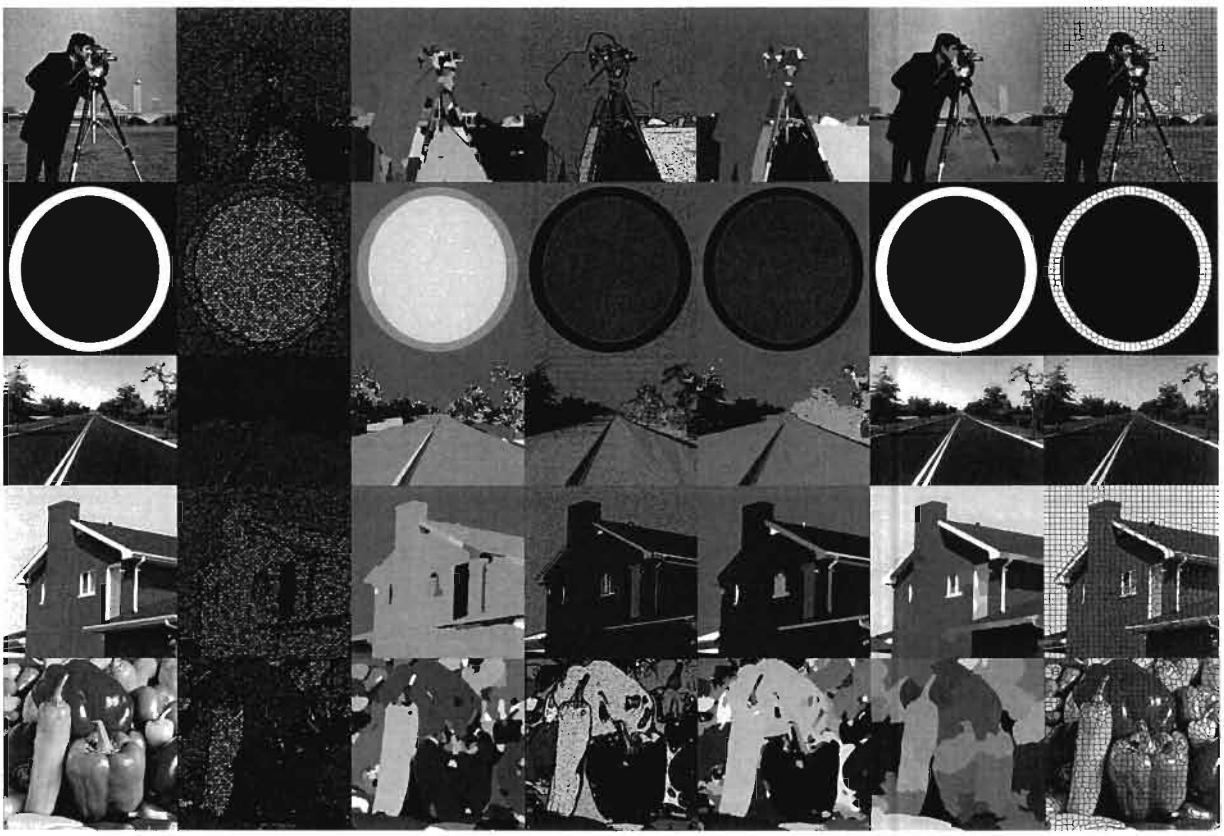

Figura 8. Rezultate calitative. Prima coloană prezintă imaginea de test. A doua și a treia coloană prezintă segmentarea cu trasare de raze cu cate un punct germen pentru fiecare fereastră de 16x16 pixeli (rară și filtrată). A patra și a cincea coloană prezintă segmentarea cu trasare de raze cu câte un punct germen pentru fiecare fereastră de 4x4 pixeli. A șasea coloană prezintă segmentarea obținută cu ReallyQuickShift [12]. A șaptea coloană prezintă pre-segmentarea în superpixeli obținută cu SLIC [2]. Segmentarea propusă prezintă calitate bazată pe percepție similară cu metodele de ultimă oră. În plus, algoritmul propus scoate la ieșire segmentări complete, în timp ce atât SLIC cât și RealyQuickShift necesită unificări adiționale din cauza strategiei de supra-segmentare.

size	flux	net (*)	opt. filter (*)	opt. merging (*)	FULL (*)	SPARSE (*)	net (**)	opt. filter (**)	opt. merging (**)	FULL (**)	SPARSE (**)	STAR SLIC (***)	STAR SLIC (****)	STAR ReallyQuickShift
256	0.182	0.523	0.505	0.803	2.013	0.705	0.172	4.682	2.323	7.359	0.354	2.183	5.001	5.075
512	0.659	1.955	1.853	2.415	6.882	2.614	0.988	4.772	1.785	8.204	1.647	5.135	20.175	21.788
758	0.909	3.551	5.531	3.689	13.68	4.46	1.293	10.741	2.693	15.636	2.202	13.38	36.159	62.422
1024	1.677	7.519	4.854	5.677	19.727	9.196	2.329	11.932	4.153	20.091	4.006	18.876	62.148	123.643
1536	3.731	17.102	10.752	12.522	44.107	20.833	4.887	26.001	9.023	43.642	8.618	53.188	123.768	307.163
2048	6.583	31.083	18.501	22.166	78.333	37.666	8.215	46.027	15.756	76.581	14.798	71.998	196.794	524.439
3072	15.001	73.141	41.004	49.809	178.955	88.142	20.777	120.555	35.111	191.444	35.778	203.093	481.791	1272.256
4096	26.375	134.754	68.761	87.251	317.141	161.129	37.115	215.502	62.437	341.479	63.49	792.329	770.729	2055.837
6144	59.714	307.857	164.714	193.002	725.287	367.571	85.723	486.909	140.363	772.709	145.437	762.215	1842.969	3673.915
8192	106.051	437.751	416.503	341.875	1302.18	543.802	135.447	897.222	267.256	1405.976	241.498	1151.41	2990.072	6941.628

measurements in milliseconds (*) very dense sparse net (one seed per 4x4 tile) (**) normal density sparse net (one seed per 16x16 tile) (***) SLIC without connectivity (****) SLIC with connectivity

Figura 9. Rezultate de viteză. Această imagine prezintă comparația în ce privește viteza (în milisecunde) pe imagini de dimensiune *size* x *size*. Coloanele evidențiate prezintă timpul de execuție total pentru segmentarea imperfectă rară cu trasare de raze cu un punct germen pentru fiecare 4x4 pixeli (*) și cu un punct germen pentru fiecare 16x16 pixeli (**). Ambii timpi sunt net superiori rezultatelor celei mai rapide metode (STAR). Este de menționat faptul că paralelismul crescut al algoritmului prezentat conduce la o viteză de execuție care se degradează sub-liniar cu numărul de puncte germen. Toate măsurătorile au fost realizate pe o placă video NVIDIA GTX 960M.

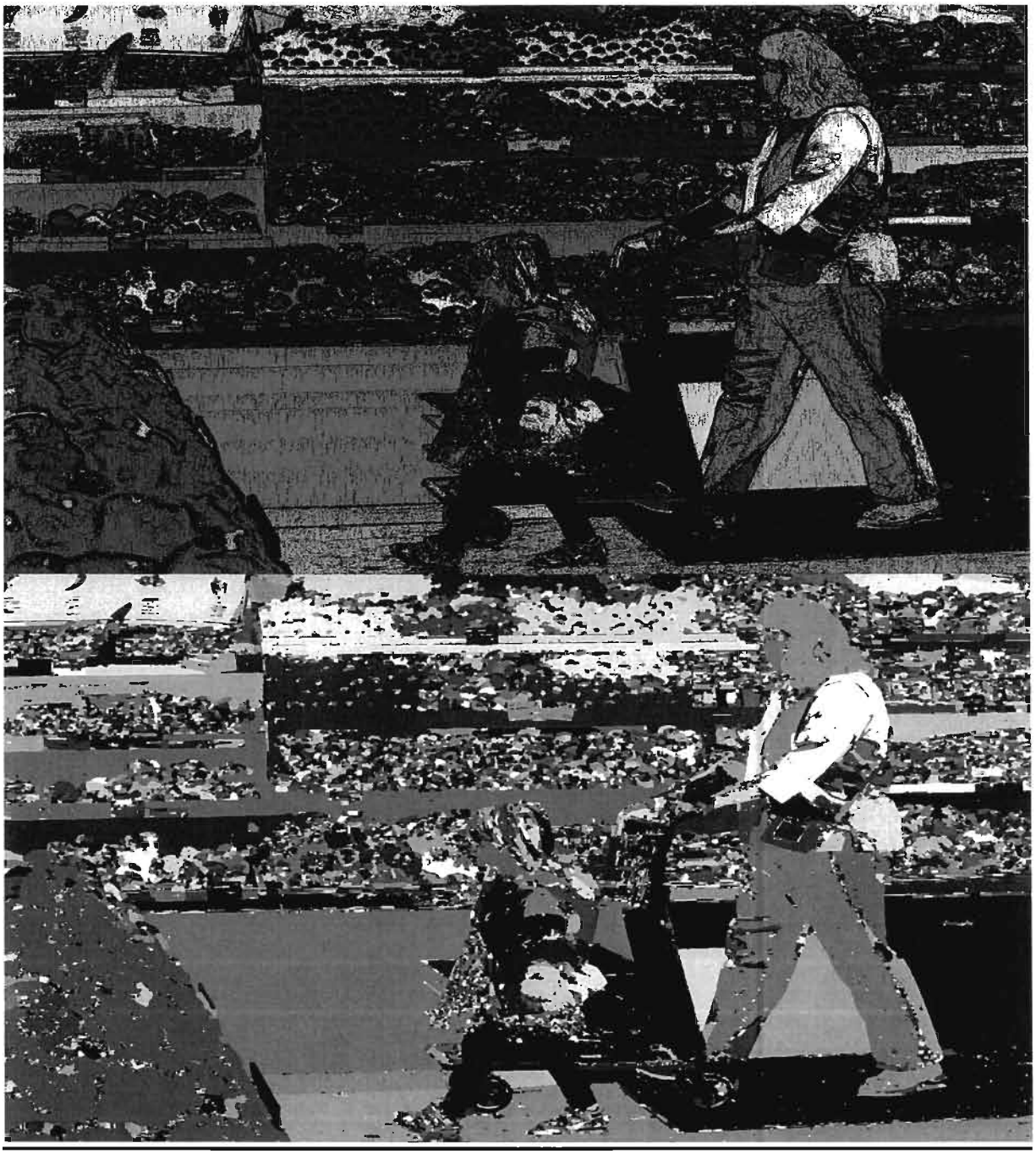

Figura 10. Segmentarea rară cu trasare de raze poate fi utilizată ca un pre-segmentator pur. Imaginea de sus prezintă o segmentare rară cu multe puncte germen, în timp ce imaginea de jos prezintă rezultatele filtrate. Imaginea de jos combină în același timp segmente mari și multe segmente de dimensiunea unui super-pixel. O altă metodă de pre-segmentare ar fi segmentat și regiunile mari în superpixeli

DESCRIEREA INVENȚIEI

SISTEM PENTRU SEGMENTAREA IMAGINILOR VIDEO IN TIMP REAL, BAZAT PE TRASARE DE RAZE

Sistemul propus (inventia) are la bază un algoritm original de segmentare (partitionare) a câmpurilor K-dimensionale (exemplu: imagini bidimensionale), care asigura o viteza la executie de aproape 5 ori mai mare decât cea realizata de sistemele existente, după cum se poate observa în Figura 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.

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

Inventia poate fi aplicată in domeniul informatic medical (detectare automată de tumori din scanări CT/ RMN, sisteme asistive), roboți industriali (programele de decizie ale roboților autonomi industriali și a dronelor folosesc segmentare), industria automobilelor (segmentarea este folosită de mașini cu conducere automată pentru percepția elementelor de trafic), securitate (urmărirea automată pe camere de securitate folosește segmentare, detecția automată a elementelor de interes folosește segmentare), etc.

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 [13][14][19], compresie [30], histograme [30], detecția muchiilor [30][18], extinderea regiunilor [26][3][14][5], partiționarea grafurilor [10][27][29][30], transformarea watershed [15][30],

praguri adaptive [5], divizarea şi unificarea [18][30], "Random Walker" [8], contururi ierarhice [30][28] şi active [30].

Cercetări recente s-au concentrat pe metode de segmentare şi cosegmentare antrenabile [30][22][20][4][16][7][23], folosind tehnici de învăţare automata pentru a produce rezultate de o acurateţe ridicată, cu supraveghere slabă. Totuşi, aceste metode nu reprezintă soluţii pentru segmentări în timp real, deoarece costurile de execuţie sunt mult prea mari pentru puterea de calcul a sistemelor hardware nespecializate, în special pentru laptopuri uzuale şi pentru dispozitive mobile.

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

In multe aplicaţii 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. Segmentarea rapidă este esenţială pentru urmărirea şi detecţia obiectelor, navigarea şi percepţia pentru roboţi autonomi, dispozitive asistive pentru nevăzători sau dispozitive medicale controlate automat.

Aceasta propunere de brevet introduce un sistem ce include o nouă metodă de segmentare, bazata pe extinderea regiunilor prin trasarea de raze. Este o metoda de segmentare rară care etichetează numai o fracţiune din matricea de intrare în vederea minimizării costului computational. Figura 2 prezintă ieşirea sistemului având ca intrare o imagine 2D, şi modul în care se pot obţine diferite niveluri de precizie prin diverse configurări ale gradului de risipire a razelor. Gradul de risipire este controlat parametric. Chiar cu un grad de risipire ridicat, segmentarea rezultată este suficient de calitativă pentru numeroase aplicaţii în timp real

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

În acest brevet, metoda este exemplificată în 2D, dar poate fi implementată în oricâte dimensiuni.

Segmentarea rară prin trasarea de raze aduce următoarele contribuţii:

- 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
- un număr constant de treceri prin banda grafică, conducând la un timp de execuție care nu depinde de tipul imaginii, fără a afecta calitatea
- lățime de bandă scăzută datorită rarității razelor, care determină un număr mic de accese la memorie
- o complexitate de *O(n/tsize)* unde *n* este dimensiunea setului de intrare iar *tsize* este dimensiuniea ferestrei (sub-regiunii) pentru care este generat un punct germen de pornire a razelor (exemplu: 16x16). Segmentările cu cele mai bune rezultate, SLIC [2] și ReallyQuickShift [12] au complexități de $O(n)$ și $O(d*n^2)$.

Segmentarea rară prin trasarea de raze este sumar descrisă în Figura 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.

Stadiul actual al dezvoltării în segmentarea imaginilor:

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

Majoritatea metodelor inițiale de segmentare pe GPU au fost realizate în scopul utilizării în medicină, unde accelerarea GPU a adus beneficii considerabile în manipularea 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. 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 $O(n*log(n))$. Körbes și alții [15] au introdus un algoritm "watershed" paralel iterativ, în care o imagine este divizată în ferestre de dimensiune 16x16 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 computational 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 executata 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), Conditional Random Fields (CRF) sau Fully Convoluted 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 algotimilor 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 $O(d*n^2)$, 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.

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 strategie de clusterizare

bazată pe ascensiunea gradientului şi oferă o calitate a segmentării rezonabilă pentru un timp de rulare foarte scăzut.

Invenţia:

Segmentarea K-dimensională este o problemă de partiţionare în care un set de date *S* care conţine *n* elemente K-dimensionale (o matrice K-dimensională) trebuie să fie împărţit în regiuni continue disjuncte, pe baza unor proprietăţi de similaritate locală sau globală între elemente. Această problemă este rezolvată prin căutarea regiunilor disjuncte în S şi determinarea celei mai bune soluţii.

Metodele de extindere a regiunilor sau de clusterizare determină această partiţionare prin încercarea iterativă de unire a elementelor din afara clusterelor, după cum este ilustrat în Figura 3. Metodele tradiţionale precum k-means [14] explorează întregul set, producând un spaţiu de căutare extrem de mare, care se explorează destul de greu. Metode de pre-segmentare prin extinderea regiunilor ca SLIC [2] limitează spaţiul de căutare la o vecinătate foarte mică a clusterului, garantând o explorare rapidă a spaţiului de căutare, dar neputând să detecteze segmente (regiuni) foarte mari.

Strategia propusă de invenţia noastră combină avantajele celor două abordări: numărul mic de elemente explorate de abordarea SLIC şi spaţiul mare de căutare al metodelor tradiţionale, ducând în acelaşi timp la o viteză de execuţie foarte mare şi la conexiuni de-a lungul întregului set. Metoda de segmentare rară cu trasare de raze funcţionează identic pe matrici de orice dimensiuni, dar pentru simplitate este ilustrată pe imagini bidimensionale.

Algoritmul de segmentare complet este ilustrat sumar în Figura 4: calculul fluxului, generarea punctelor germen (a generatorilor), explorarea spaţiului prin trasarea razelor şi conectarea regiunilor parcurse.

Fluxul reprezintă rata de schimbare a matricii de intrare. Poate fi un simplu gradient adaptiv pentru imagini color/în niveluri de gri şi este extrem de ieftin de calculat pe GPU. Pentru imagini de adâncime (de exemplu, cele obţinute de la dispozitivul de achiziţie Sound of Vision) s-a utilizat un gradient adaptiv care ţine cont de eroarea de estimare a senzorului de adâncime. Un exemplu de flux se poate observa în Figura 4.2.

Poziţionarea punctelor germen este iniţial generată printr-un şablon pseudo-aleator de-a lungul întregului set. De exemplu, pentru imagini 2D, oricare din secvenţele Sobol, van der Corput sau Hammersly pot fi utilizate. După generarea poziţiilor iniţiale, se utilizează o strategie bazată pe ascensiunea gradientului pentru distanţarea punctelor germen de zonele cu flux ridicat, utilizând următorul algoritm, redat în pseudocod:

Pseudocode 1. Poziţionarea punctelor germen

1: seedPos ← getPseudoRandomPosition()
2: iterations ← 0

```
3: flux ← getFlux(seedPos)
4: while flux > Threshold and iterations <MaxIterations
5:     fluxDirection ← getFluxDirection(seedPos)
6:     seedPos ← seedPos + fluxDirection
7:     flux ← getFlux(seedPos)
8:     iterations ← iterations + 1
9: return seedPos
```

Un exemplu al distribuției rezultate a punctelor germen poate fi observată în Figura 4.3, colorată cu o funcție de hash Jenkins aplicată peste indicii punctelor germen.

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

Razele suportă de asemenea reflexii pe marginile imaginii sau în zone de flux ridicat (muchii), reprezentând exploratori eficienți chiar și pentru cazuri complicate. Razele numeroase sunt cu ușurință distribuite pe toate nucleele de procesare disponibile. Rezultatul razelor trasate dar neconectate se poate observa în Figura 4.4.

O structură arborescentă este creată peste punctele germen, care leagă toate punctele de plecare ale întregului set de date. Inițial toate punctele germen pornesc ca arbori separați. Cu fiecare conexiune între raze, numărul de arbori separați scade. În vederea minimizării înălțimii arborilor atunci când două raze se întâlnesc, numai punctele germen sunt conectate. Deoarece sincronizarea datelor pe GPU poate fi implementată eficient numai prin operații atomice, trebuie stabilită o ierarhie atunci când două puncte germen sunt conectate. Noi am utilizat indicele minim al punctelor care trebuie conectate. Următorul algoritm este utilizat pentru conectarea a două pucte germen:

Pseudocod 2. Conectarea a două puncte germen

```
getGreatestParent(initial_seedId)
1: seedId ← initial_seedId
2: while seedId ≠ getParentId(seedId)
3:         seedId ← getParentId(seedId)
4: return seedId

ConnectSeeds(seedId1, seedId2)
1: parent1.id ← getGreatestParent(seedId1)
2: parent2.id ← getGreatestParent(seedId2)
3: if parent1.id < parent2.id
4:         atomicExchange(parent2.parentId, parent1.id)
5: else
6:         atomicExchange(parent1.parentId, parent2.id)
```

Fiecare rază menține un set de proprietăți legate de spațiul traversat, incluzând valoarea medie a punctelor, varianța și valoarea ultimelor *T* puncte traversate. Proprietățile sunt salvate într-un buffer auxiliar (matricea *TProps* din pseudocodul 3). O conexiune potențială între puncte germen are loc atunci când o rază întâlnește puncte traversate de altă rază. Conexiunea este realizată numai dacă există potrivire locală și globală între cele două puncte germen. Conexiunea are loc numai dacă proprietățile celor două raze sunt similare. Următorul pseudocod descrie acest proces:

Pseudocod 3. Trasarea de raze

```
Initialization Pass:
initialize label matrix Labels to empty
initialize traced properties matrix TProps to empty

Ray Tracing Pass (1 thread per each ray per seed):
seed ← getSeed (ray.id)
rayProperties ← initializeRayProperties()
iterations ← 0
pos ← seed.pos
while iterations < MaxIterations
    flux ← readFlux (pos)                    //from flux matrix
    if flux > MaxFlux
        end                    //high flux (edge) termination case
    input ← readInput (pos)                  //from input image
    updateRayPropertiesInputFlux (rayProperties, input,
                                 flux)
    if rayPropertiesDeviateFromCluster(rayProperties)
        end            //average value of the last T traced pixels
                       //deviates from average value of the ray
    if (rayProperties.accumulatedFlux > AccMaxFlux)
        end        //accumulating enough flux termination case
    scatterDir ← getScatteringBasedOnAccumFlux()
                                             //see Figure 6
    id ← readLabel (pos)
    if id is undefined
                   // the pixel has not been traversed by other rays
        writeLabel (pos, seed.id)
        writeTraceProperties(pos, rayProperties)
    else
        if alreadyConnectedNets(seed.id,id)
            end          //the seeds are already part of the same
                         //forest (they have the same greatest parent)
        posProperties ← readProperties(pos)
        if rayProperties not_similar posProperties
            end
        updateRayProperties(rayProperties,posProperties)
//update with the properties of the other ray traced through pos
        writeTraceProperties(pos, rayProperties)
        ConnectSeeds(seed.id, id)
    if (ray.dir + scatterDir) aligned to image axes
            pos ← pos + ray.dir + scatterDir
```

```
38:     else
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 Figura 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 Figura 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 imaginie, reprezintă id-urile de segmentare unice. Pseudocodul pentru forțarea conectivității este urmatorul:

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

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

Numărul de operații de acces al memoriei 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.

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

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.

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.

Pseudocodul pentru filtrarea exacta este următorul:

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

```
pos ←getPos(pixel)
if readLabel(pos) not undefined
    end
steps ← 0
for rayId ← 1, rayId < NumRays, rayId++
    alive[rayId] ← true
    rayProperties[rayId] ← empty
for steps ← 1, steps < MaxSteps, steps++
    for rayId ← 1, rayId < NumRays, rayId++
        if not alive[rayId]
            continue
        ray ← getRay(rayId)
        if ray.dir aligned to image axes
            rpos ← pos + steps · ray.dir
        else
            rpos ← StaircaseTracing(ray.dir)
        flux ← readFlux(rpos)
        if flux > MaxFlux
            alive[rayId] ← false
            continue
        input ← readInput(rpos)
        updateRayProperties(properties[rayId], input)
        seedId ← readLabel(rpos)
        if seedId not undefined
            posProperties ← readProperties(rpos)
            if properties[rayId] similar posProperties
                writeLabel(pos, seedId)
                end
            else
                alive[rayId] ← 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.

Unificarea regiunilor este un alt proces opțional, care poate conduce la îmbunătățiri calitative semnificative, după cum se poate observa în Figura 7. Unificarea este de asemenea un process 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 etichetarea locală.

Apoi, în etapa a doua se acumulează informații în părintele final al fiecărui punct 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 proprietățile se potrivesc dar etichetarea este diferită, atunci cele două etichete sunt unificate.

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 ultimă oră de (pre)segmentare, după cum este ilustrat în Figura 9.

Evaluarea invenției în comparație cu metodele deja existente de segmentare:

Metodologia de evaluare a rezultatelor compară segmentarea rară cu trasare de raze atât calitativ cât și din punct de vedere al vitezei cu cele mai rapide metode de segmentare pe GPU, cu ReallyQuickShift **Error! Reference source not found.** și SLICO **Error! Reference source not found.**. În timp ce acestea sunt metode de pre-segmentare și necesită procesare suplimentară pentru producerea de segmentări complete, ele reprezintă singura categorie de 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 rapidă decât orice algoritm complet de segmentare pe GPU, cel mai rapid având complexitatea *O(n*log n)*.

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

Figura 8 prezintă rezultatele calitative ale segmentării prin trasarea de raze, unde diferite imagini sunt segmentate și comparate cu segmentări consacrate. Testarea calitativă 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 afectează direct numărul de puncte germen. Astfel, a patra coloană din Figura 8 necesită mărire, deoarece rezultatele sunt obținute cu o valoare de raritate/risipire foarte mică.

Figura 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 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 filtrare. Din Figura 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 și foarte mari.

Segmentarea rară bazată pe trasarea de raze este o metodă flexibilă, unde compromisul 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 Figura 10. Pre-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 care partiționează astfel de segmente.

O limitare a abordării noastre este dependența de calculul fluxului, care poate conține 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).

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 rimp 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.

Referinte:

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

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

[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.

[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.

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

[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

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

[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

[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

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

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

[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.

[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.

[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 Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.

[15] A. Körbes, G. Vitor, R. Lotufo and J. Ferreira, Analysis of a step-based watershed algorithm using CUDA, International Journal of Natural Computing Research, 2010.

[16] C. Lea, M. Flynn, R. Vidal, A. Reiter, G. Hager, Temporal Convolutional Networks for Action Segmentation and Detection, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[17] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson and K. Siddiqi, TurboPixels: Fast superpixels using geometric flows, Transactions on Pattern Analysis and Machine Intelligence, pp. 2290–2297, 2009.

[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, 2015

[19] Z. Li, J. Chen, Superpixel Segmentation using Linear Spectral Clustering, IEEE Conference on Computer Vision and Pattern Recognition, 2015

[20] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, Computer Vision and Pattern Recognition, 2015.

[21] M. McGuire, M. Mara, Efficient GPU Screen-Space Ray Tracing, Journal of Computer Graphics Techniques, Efficient GPU Screen-Space Ray Tracing, Vol. 3, No. 4, 2014

[22] A. Mustafa, and A. Hilton, Semantically Coherent Co-Segmentation and Reconstruction of Dynamic Scenes, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[23] S. Paris, Edge-preserving Smoothing and Mean-shift Segmentation of Video Streams, Proceeding ECCV '08 Proceedings of the 10th European Conference on Computer Vision: Part II, pp 460-473, 2008.

[24] E. Ramírez, P. Temoche, R. Carmona, A volume segmentation approach based on GrabCut, CLEI Electronic journal, vol. 16, 2013

[25] M. Roberts, J. Packer, M. Sousa and J. Mitchell, A Work-Efficient GPU Algorithm for Level Set Segmentation, High Performance Graphics, 2010.

[26] S. Schenke, B. Wünsche, J. Denzler, GPU Based Volume Segmentation, In Proc. of IVCNZ '05, 2005.

[27] J. Shi and J. Malik, Normalized Cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000.

[28] P. Soille, Constrained Connectivity for Hierarchical Image Partitioning and Simplification, IEEE transactions on pattern analysis and machine intelligence, vol. 30, no. 7, 2008.

[29] V. Vineet and P. Narayanan, CUDA Cuts: Fast Graph Cuts on the GPU, 2008.

[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 Communication and Image Representation, 34, pp 12-27, 2016

[31] A. Abramov, T. Kulvicius, F. Wörgötter, and B. Dellen, Real-time image segmentation on a GPU, Facing the multicore-challenge, pp 131-142, 2010.

[32] https://soundofvision.net/

[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 impaired: the Sound of Vision system, IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1480-1489, 2017.

REVENDICARI

- **Revendicarea 1**: Se revendică **sistemul pentru segmentarea imaginilor video în timp real, bazat pe trasare de raze**, caracterizat în primul rând prin faptul că este capabil să 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 acoperirii tuturor punctelor dintr-o vecinătate locală. Metoda de segmentare propusă are un grad înalt de paralelizare, conducând la o segmentare rapidă, în timp real. Sistemul este compus din următoarele elemente:
    a. Un dispozitiv de achiziție sau de generare a matricilor bi- sau multi-dimensionale
    b. Un modul opțional de pre-procesare a datelor de intrare (ex: filtrarea zgomotelor, calculul unei hărți de normale pornind de la imagini de adâncime)
    c. Un modul de segmentare rară prin trasarea de raze care obține la ieșire regiuni continue disjuncte, prin metoda de segmentare descrisă în Revendicarea 2
    d. Un modul de post-procesare care utilizează regiunile disjuncte în funcție de scopul aplicației

  Figura 1 prezintă schema de funcționare a sistemului și oferă explicații suplimentare cu privire la sub-sistemele constitutive

- **Revendicarea 2:** Se revendică **metoda de segmentare a imaginilor video și a matricilor multi-dimensionale în timp real, bazată pe trasare de raze**, caracterizată prin următoarea succesiune de etape:
  - **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 estima prin diferențe finite)
    - 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 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 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. Aceste puncte germen acoperă cât mai omogen spațiul setului de date de intrare.
    - Exemplu de implementare și optimizare: Pozițiile punctelor germen pot fi alese folosind secvențe de tipul Sobol, Hammersly si Van der Corput sau distribuții Poisson. După construcție, generatorii sunt mutați iterativ departe de zonele de flux mare, pentru a maximiza șansele de generare de regiuni de segmentare din zone cu flux scăzut, adică zone în care similaritatea dintre vecini este mare.
  - **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 numar foarte mic de

eşantione, minimizând costurile explorării spaţiului de căutare. Fiecare rază conţine statistici despre elementele din spaţiul traversat.

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

- **Etapa 4:** Conectarea punctelor germen (şi implicit, a razelor care aparţin de acele puncte germen) în momentul în care razele trasate se intersectează. Conectarea se face doar în condiţii de similaritate între razele care se intersectează (dacă elementele din spaţiile parcurse de raze au proprietăţi similare). La conectarea a două puncte germen, unul dintre ele devine părintele celuilalt punct germen. Astfel, punctele germen conectate prin raze sunt unite, rezultând în final un set de puncte germen părinte care partiţionează matricea de intrare în regiuni continue care nu se suprapun unele cu celelalte.
- **Etapa 5:** Extinderea opţională a regiunilor generate pentru a acoperi complet setul de date iniţial, utilizând un proces de filtrare în doi paşi:
  - **Pasul 1:** Pentru fiecare element ne-etichetat, se trimit raze care eşantionează spaţiul. Atunci când o rază întâlneşte o regiune, dacă proprietăţile razei sunt asemănătoare cu cele ale regiunii, elementul ne-etichetat primeşte id-ul regiunii respective.
  - **Pasul 2:** Elementele care nu au fost etichetate nici în pasul 1 şi care nu se află pe zone de flux mare, sunt prelucrate prin eşantionarea regiunilor care se află în vecinătatea lor. Elementul ne-etichetat primeşte id-ul regiunii din vecinătate care are proprietăţile cele mai similare cu acelea ale elementului respectiv.
- **Etapa 6:** Unirea opţională a sub-seturilor generate, folosind un proces iterativ de calcul de statistici în interiorul regiunilor şi de unificare a regiunilor vecine pe baza acestor statistici (dacă aceste două regiuni au proprietăţi similare).

Rezultatul aplicării metodei este un set de regiuni continue disjuncte, fiecare regiune conţinând elemente cu proprietăţi similare. Acest rezultat poate fi folosit ulterior la segmentarea semantică a imaginilor video sau a matricilor multi-dimensionale (de exemplu, seturi de date CT).