

Implementarea unei aplicații pentru recunoașterea gesturilor mâinii, cu aplicabilitate în interacțiunea om-calculator

Bradi Rareș, Dorian Gorgan

Universitatea Tehnică Cluj-Napoca

Str. Memorandumului nr. 28, CP 400114, Cluj-Napoca

E-mail: bradi.rares@gmail.com, dorian.gorgan@cs.utcluj.ro

Rezumat. Sarcina de recunoaștere a gesturilor mâinii este una activ cercetată și provocatoare datorită existenței fundalurilor complexe având iluminări diferite precum și varietatea de gesturi posibile ale mâinii. Lucrarea de față prezintă analiza și implementarea unei aplicații capabilă să recunoască centrul mâinii, degetele, precum și încheietura acesteia. Analiza cuprinde evaluarea cerințelor împreună cu proiectarea arhitecturii aplicației. Implementarea tratează recunoașterea gesturilor dinamice făcute de mână. Recunoașterea centrului mâinii se face prin cercul de rază maximă înscris în conturul ei, iar vârful degetelor utilizând curbura k (k -curvature). Traectoria mâinii este urmărită și diferențele poziției intermediare sunt analizate, pentru a identifica o direcție de deplasare. Pentru a dobândi stabilitate și robustețe trăsăturile extrase sunt urmărite prin filtre Kalman, iar gesturile sunt agregate în timp printr-un algoritm de monitorizare. Gesturile sunt compuse dintr-o direcție de deplasare și o anumită poziționare a vârfului degetelor. Analiza aplicației se face în trei etape și urmărește robustețea la fundal și iluminare, viteza de procesare și rata de eroare.

Cuvinte cheie: recunoașterea gesturilor, gesturi ale mâinii, sisteme de timp real, segmentare imagine, camere, viziune pe calculator, extragere trăsături, urmărire, vârful degetelor.

1. Introducere

Odată cu extinderea utilizării calculatoarelor în societate, interacțiunea om - calculator a devenit o parte componentă din viața de zi cu zi. Una dintre barierele existente, conform lui Chaudhary et al. (2011), în interacțiunea om-calculator este modalitatea prin care interacționăm cu fluxul mare de informație. Soluția este dezvoltarea unor tehnici de interacțiune “naturală” cu calculatorul, asemănătoare celor prin care oamenii interacționează între ei. Folosirea gesturilor mâinii reprezintă o alternativă naturală la tehnicile de

interacțiune din interfețele utilizator clasice. Prin formele non-verbale de comunicare, realizate prin mișcarea mâinilor, gesturile devin componente ale unui limbaj de dialog cu calculatorul, utilizat pentru schimbul de informație.

Dezvoltarea tehnicilor de recunoaștere vizuală pe calculator face posibilă abordarea recunoașterii gesturilor din perspectiva comunicării dintre om și calculator. În ultimii ani tehnicile de recunoaștere vizuală a gesturilor mâinii au cunoscut o dezvoltare accentuată, atât datorită creșterii puterii de calcul a sistemelor, cât și a evoluției tehnologiei software, permițând tehnici de interacțiune în timp real.

Conform lui Fakhreddine et al. (2008) cele două caracteristici importante în proiectarea unei aplicații interactive sunt utilizabilitatea și funcționalitatea. Pentru a realiza o interacțiune om calculator performantă și de bună calitate, ambele caracteristici trebuie tratate în egală măsură.

Dificultățile care apar în implementarea unor tehnici de recunoaștere sunt:

- Detecția prezenței și a poziției mâinii în scenă;
- Identificarea clasei de gesturi căreia îi aparține forma mâinii din scenă;
- Determinarea deplasării în scenă a obiectului identificat;
- Latența sistemului;
- Stabilitatea urmăririi: variația poziției măsurate să fie cât mai mică, dacă obiectul măsurat nu se mișcă.

Detecția prezenței mâinii în scenă se face prin determinarea cercului de rază maximă care poate fi potrivit în interiorul conturului mâinii. Identificarea gesturilor se face prin intermediul vârfurilor degetelor extrase folosind curbura-k ale conturului mâinii care îndeplinesc condițiile necesare. Pe baza acestor determinări se identifică degetul mare și cel mic, ca fiind acele vârfuri care satisfac condițiile de poziționare relativă în regiunea mâinii. Urmărirea se face într-o fereastră de analiză de dimensiune fixă, care acumulează date precedente și ia decizii pe baza lor. Pentru a avea stabilitate în datele captate se folosesc filtre Kalman. Astfel se elimină variațiile bruște ale datelor măsurate și permite o estimare bună a poziției, chiar în momentele în care măsurătorile lipsesc. Traectoria urmărită este clasificată pe baza punctelor intermediare captate, utilizând predicția liniară.

Contribuția acestei lucrări este integrarea numeroaselor aspecte pentru recunoașterea unui gest făcut de mână într-o aplicație completă.

În secțiunea Realizări Asemănătoare sunt prezentate sisteme construite, folosind tehnici și algoritmi diferiți, dar cu același rezultat final pentru utilizator. În Arhitectura aplicației se prezintă principalele componente funcționale. În Extragerea Trăsăturilor se analizează în detaliu algoritmi folosiți în detecția principalelor caracteristici prin care se face identificarea gestului. Secțiunea Recunoașterea Gesturilor descrie soluțiile problemelor de urmărire și stabilitate. În secțiunea Evaluare și Rezultate sunt descrise scenariile de test relevante și indicatorii de performanță folosiți. Ultima parte, Concluzii evidențiază pe scurt ideile și observațiile importante expuse în lucrare.

2. Realizări asemănătoare

Ca urmare a activităților de cercetare în domeniul recunoașterii gesturilor care pot opera în timp real, s-au conturat clase de abordări distincte. O primă clasă sunt cele care folosesc modelul matematic/computațional de rețele neuronale artificiale. Acestea își propun simularea structurii biologice a sistemului neuronal al omului. Rețelele neuronale acceptă ca intrare trăsături și produc la ieșire decizii. Maung (2009) aplică acest model pentru recunoașterea limbajului de semne realizate cu mâinile. Se folosește conceptul de perceptron ca algoritm de clasificare. Trăsăturile alese pentru antrenarea lui sunt histogramele orientării gradientilor, în care operatorul Sorbel, din pașii de calcul a trăsăturilor a fost înlocuit printr-un alt operator, însă având coeficienți similari. Această abordare presupune antrenarea modelului, într-o fază de preprocesare, înaintea rulării.

Față de metoda programată folosită în abordarea noastră, soluția rețelei neuronale bazate pe antrenare oferă avantajul adaptării la condiții care se schimbă într-un ritm constant. Metoda programată poate eșua pentru situații care diferă de cele luate în calcul în etapa proiectării. De asemenea în metoda programată determinarea diferitelor constante pentru algoritmi se face adesea empiric. Totuși formarea bazei de date și antrenarea modelului sunt consumatoare de timp și pot fi repetate doar în anumite condiții. Însă metoda Față de abordarea curentă sistemul descris în lucrare recunoaște s-a rezumat la identificarea doar a gesturilor statice.

O altă clasă de abordări sunt cele bazate pe Support Vector Machines (SVM) care învață un model statistic, care aplicat pe date multi-dimensionale oferă rezultate bune. Scopul algoritmului este găsirea hiperplanului optim separă punctele din clase diferite, maximizând marginea dintre ele. Ke et al. (2013) folosesc ca date de intrare pentru acest algoritm rezultatul aplicării transformatei undă (wavelet) Gabor pe conturul extras al mâinii. Traectoria gesturilor este compusă din simboluri obținute din metoda cuantizării vectoriale care, sunt apoi date de intrare pentru un Model Markov Ascuns (HMM) în vederea clasificării mișcării. Modelul HMM este folosit adesea pentru modelarea dimensiunii temporale, în mod special în sisteme de recunoaștere vocală. Pentru a adăuga un nou gest în sistem trebuie extinsă baza de date cu mostre noi, iar procesul de antrenare trebuie rulat din nou. Acest lucru este laborios. În lucrare extinderea numărului de gesturi se face prin adăugarea condițiilor care modelează gesturile noi. Procesul este rapid și ușor dacă gestul poate fi descris de trăsăturile aplicației.

Kulshreshth et al. (2013) captează datele vizuale despre poziția mâinii folosind o cameră Kinect. Segmentarea mâinii, folosind informațiile de adâncime, determină conturul mâinii, rezultând mai puține goluri în interiorul conturului față de alte metode de segmentare. Din conturul extras, se alege puncte echidistante pentru care se calculează distanța față de centroidul regiunii mâinii. Pe aceste puncte se aplică transformata Fourier, iar rezultatul obținut poartă denumirea de descriptori Fourier. Un gest este dat de o grupare de astfel de descriptori. La recunoaștere se aplică distanța euclidiană dintre modelul antrenat și cel captat, gestul recunoscut fiind cel pentru care se obține scorul cel mai mic. Principalul dezavantaj vine din presupunerea că mâna este cel mai apropiat obiect de interes din scenă, lucru care nu este valabil în toate cazurile. Abordarea prezentă nu se bazează pe această presupunere. O altă distincție față de lucrarea prezentată reprezintă trăsăturile folosite pentru a modela gesturile. Deoarece se folosesc vârfulurile degetelor, modelul folosit în această lucrare are o dimensiune mică, iar recunoașterea are un cost computațional mai redus decât cel folosit de Kulshreshth et al. (2013).

3. Arhitectura aplicației

Figura 1 pune în evidență arhitectura. La bază, aplicația este formată din patru componente:

1. Modulul de preluare al imaginilor de la camera video. Pe lângă rolul de captare el mai are sarcina de a scala în jos imaginea, pentru o procesare rapidă.
2. Modulul de detecție alcătuiește backend-ul aplicației. Pe lângă rolurile din cerințele de detecție, el păstrează starea internă. Pentru a putea opera în condiții complexe, precum schimbări de iluminare, umbrii sau ocluziuni date de elemente din scenă, el comunică intern cu logica de urmărire și corecție pentru a putea indentifica corect trăsăturile.

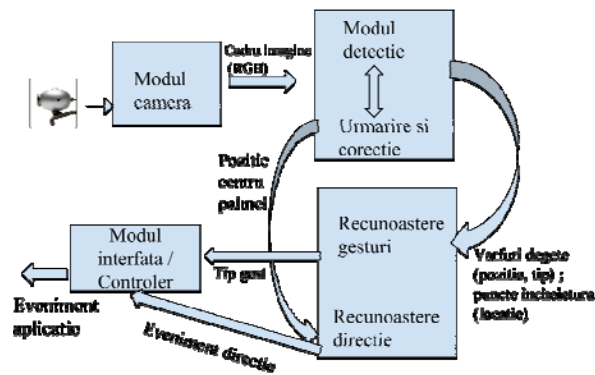


Figura 1. Arhitectura aplicației de recunoaștere a gesturilor mâinii

3. Modulul de recunoaștere. Odată identificate, trăsăturile mâinii sunt transmise componentei în care sunt modelate tipurile de posturi recunoscute și componentei de recunoaștere a direcțiilor de deplasare. Aplicația, în acest punct, va opera cu gesturi și direcție generală de deplasare.

4. Agregarea acestor informații în timp și efectuarea unei decizii finale ținând cont de rezultate în momente de timp anterioare, este sarcina ultimei componente de controler. Utilizând această informație ea va transmite evenimentul de producere a gestului aplicației utilizator.

Dezavantajele general valabile pentru astfel de tehnici sunt timpul de convergență până când fundalul este corect estimat. Un alt dezavantaj este captarea doar a obiectelor în mișcare. Segmentarea pe baza unui interval de culoare a fost a doua tehnică analizată. Fiecare pixel al cadrului este verificat, dacă valoarea lui se află într-un interval specific culorii pielii. Avantajul major față de prima tehnică este lipsa aceluși interval de conversie până când se estimează corespunzător fundalul. Dezavantajele constau în schimb în faptul că diferențele de iluminare pot afecta culoarea obiectului dorit de analizat. Pentru a diminua influența pe care o are iluminatul asupra segmentării adesea se face o transformare într-un alt spațiu de culoare invariant la aceasta precum HSV sau YC_bC_r . Lucrări precum Chai & Ngann (1999), Saxe & Foulds (1996) au studiat intervalul corespunzător pielii în astfel de spații, însă Khanal & Sidibé (2011) propun noi spații de culoare mai robuste. O evaluare a acestora a fost făcută pe o bază de date de imagini, BD (2014), statice împreună cu masca de adevăr. Pe baza rezultatelor s-a ales spațiul propus de Khanal & Sidibé (2011) cu intervalul de culoare sugerat de autori $\log(R/G) = [0.15, 1.1]$; $\log(B/G) = [-4, 0.3]$. Rezultatul aplicării segmentării asupra mâinii se poate vedea în Figura 3 imaginea din dreapta.

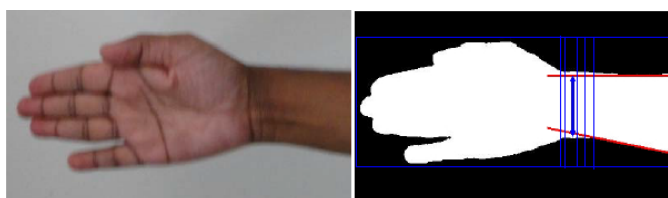


Figura 3. Imaginea originală în stânga iar în dreapta imaginea binară cu regiunea mâinii segmentată. Dreptele albastre reprezintă liniile de baleiaj, iar cea groasă candidatul pentru încheietură.

4.1.1 Detecție contur

În această etapă se determină un contur al regiunii segmentate format de punctele care-l compun. În unele cazuri regiunea mâinii poate prezenta găuri negre în interiorul conturului regiunii segmentate. Acesta este umplut pentru stagiul următor. Determinarea și umplerea se fac prin funcții ale bibliotecii OpenCV: *cvFindContours*.

4.2 Detecția centrului palmei.

În lucrările referite detecția și poziția mâinii este dată de existența unui centru al palmei. O definiție des utilizată pentru centrul palmei este punctul dat de cercul de rază maximă care poate fi înscris în regiunea mâinii. Algoritmul de determinare al acestui centru diferă de alte lucrări. Pentru determinarea razei maxime se aplică transformata distanță pe imaginea binară. Aceasta în implementarea curentă din OpenCV folosește algoritmul descris de Borgefors (1986) care calculează cea mai scurtă cale către un pixel al fundalului, folosind deplasări pe orizontală, verticală, diagonală sau deplasarea calului din șah. La ieșire vom avea o imagine în care valoarea fiecărui pixel reprezintă distanța anterior amintită. Centrul cercului va corespunde pixelului cu cea mai mare distanță, iar cea din urmă reprezintă raza. Din această cauză este necesar umplerea conturului mâinii.

Există însă excepții în care cercul de rază maximă nu reprezintă cercul mâinii precum imaginea din Figura 4.

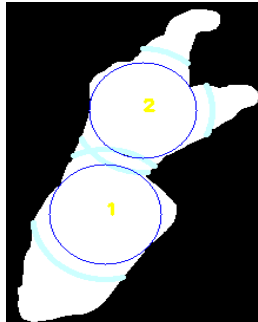


Figura 4. Exemplu caz critic în care metoda directă ar alege primul cerc. Aplicând măsura putem diferenția astfel de situații. Segmentele de cerc figurează cu cian

Soluția care se propune este folosirea unei măsuri secundare pentru a discerne astfel de excepții. Măsura este lungimea totală a segmentelor de piele aflate pe un cerc suprapus peste regiunea mâinii. Determinarea lungimii segmentelor de cerc se face prin algoritmul lui Bresenham (2014) pentru desenare a cercului. Deși beneficiem de o implementare eficientă, surplusul de calcul este de nedorit astfel încât folosirea acestei reguli de decizie se face doar când raportul dintre a doua distanță și prima este mai mare decât 0.8.

4.3 Detecția încheieturii

Grzejszczak et al.(2013) studiază această problemă din perspectiva unei implementări în timp real și urmărind o rată de eroare scăzută. Observația exprimată în lucrarea lor este regăsită și în alte abordări și spune că lățimea încheieturii formează un minim local în silueta mâinii. Acest lucru se poate observa în Figura 3. La fel ca în secțiunea precedentă aplicația se folosește de această observație, dar folosește propria implementare pentru a-l determina. Algoritmul este descris informal următorii pași aplicați unei imagini binarizate a mâinii precum cea din Figura 3:

- Se încadrează silueta mâinii cu un dreptunghi care minimizează suprafața acoperită.
- Se parcurge suprafața dreptunghiului cu o linie de baleiaj de-a lungul laturii mari a dreptunghiului. Linia are direcția dată de latura mică și punctul de început și sfârșit se află pe latura mare (liniile subțiri albastre).
- Pentru fiecare astfel de linie de baleiaj se calculează lungimea segmentului de piele (linia albastră groasă) și se rețin punctul de început și sfârșit al acestuia.
- Se rețin acele segmente de minim local dintr-o fereastră de analiză de dimensiune fixă (în implementare 10).

Încheietura va reprezenta segmentul minim cu lungimea mai mare decât un prag minim și sub un prag maxim (raza centrului palmei). Dacă pentru o linie de baleiaj se găsesc mai multe astfel de segmente, atunci se alege segmentul de lungime maximă. Pentru eficiență din analiza făcută prin linia de baleiaj se pot elimina extremitățile dreptunghiului, deoarece încheietura nu se află în acea zonă a mâinii.

4.4 Determinarea vârfulor degetelor

Vârful unui deget poate fi privit ca o curbă sau cotitură de-a lungul conturului mâinii. Detecția vârfulor degetelor se va face pe baza acestei observații, algoritmul fiind cunoscut în literatură drept k-curvature Du & Li (2000). Procedura de determinare a punctelor candidat pentru vârful este ilustrată pe Figura 5.

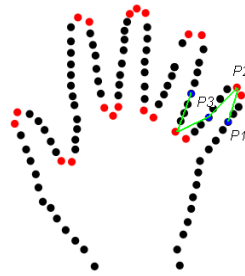


Figura 5. Ilustrarea calcului curburilor- k pentru $k = 4$. Punctele roșii sunt puncte candidat alese. Cele albastre sunt la distanță k față de cel analizat. Unghiul format de cele trei puncte este $P_1P_2P_3$

Fiecare punct de pe conturul mâinii este analizat. Pentru el se găsesc vecinii aflați la distanță k , după și înaintea lui în vector. Curbura k este definită ca unghiul format de aceste trei puncte în ordinea din Figura 5. Dacă unghiul este sub un anumit prag atunci acel punct este candidat a fi punct de vârf al degetului. Din figură se observă că există mulți candidați falși. În cele ce urmează se impun condiții care să-i elimine. Datorită anatomiei mâinii vârfurile degetelor se vor regăsi în pozițiile de pe contur aflate pe aceeași parte cu planul delimitat de încheietură, în care se află și centrul palmei. Astfel se elimină regiunea antebrațului din analiză. O altă problemă este filtrarea acelor candidați care se află într-un punct de vale de pe contur. Decizia se face asemănător cu observația de dinainte. Vârfurile se vor găsi în aceeași parte a planului delimitat de dreapta formată de vecinii săi distanțați k . Pentru vârfuri adiacente, cum regăsim în Figura 5, vom face o medie a candidaților. Un alt exemplu de candidați sunt încheieturile degetelor care respectă condiția de curbura. Aceștia sunt eliminați pe baza distanței euclidiene dintre poziția lor și poziția centrului palmei. Acele puncte rămase în urma acestor trieri, vor fi recunoscute ca vârfuri de degete în aplicație. În acest stadiu se observă cum fiecare trăsătură calculată până acum își aduce contribuția în filtrarea vârfurilor false.

5. 5. Recunoașterea gesturilor

Recunoașterea gesturilor în aplicație are trei componente. Prima componentă face recunoașterea unei posturi a mâinii definită de pozițiile degetelor, tipul degetelor și poziția centrului palmei într-un cadru de imagine. A doua componentă urmărește traiectoria în timp a mâinii. Ea

discerne cele patru direcții generale de deplasare: stânga, dreapta, sus, jos. Ultima componentă va combina aceste informații în timp pentru a putea decide starea unui gest.

5.1 Recunoașterea gesturilor statice

Deoarece se dorește diversitate în modelarea diferitelor posturi ale mâinii, aplicația va trebui să decidă și tipul fiecărui deget. În starea curentă, ea poate eticheta degetul mare și cel mic dacă le consideră prezente. Etichetarea se face pe baza a trei observații: 1. Dacă degetul mare este vizibil, atunci el va fi, fie cel mai din dreapta, fie cel mai din stânga. La fel este valabil și pentru degetul mic în funcție de mână și orientarea față de cameră; 2. Unghiul format de degetul mic sau degetul mare cu centrul palmei și centrul încheieturii sunt mai ascuțite decât cele formate de restul degetelor cu aceleași puncte; 3. Degetul mare este mai aproape de centrul palmei decât cel mic.

Pentru prima observație este necesară găsirea unei metode de ordonare a vârfulor, odată ce poziția lor este cunoscută. Algoritmul implementat este cel folosit de Mokhtar & Proamod (2012). În urma aplicării sortării, vom avea o secvență ordonată de poziții ale vârfulor degetelor de la stânga la dreapta. Doar primul și ultimul vârf sunt analizate. Dacă unghiul format de acestea cu centrul palmei și mijlocul încheieturii este mai mare decât 120° acesta este eliminat din analiză. Dacă în final rămânem cu două vârfuri, cel pentru care distanța dintre proiecția vârfului pe direcția mâinii și centrul palmei este mai mică va fi etichetat ca deget mare. Dacă însă rămânem doar cu un singur candidat, atunci distanța anterior menționată este comparată cu două praguri. Dacă este mai mic decât primul atunci este clasificat ca deget mare, dacă este mai mare decât al doilea este clasificat ca deget mic. Prin direcția mâinii înțelegem vectorul format de mijlocul încheieturii și centrul palmei.



Figura 6. Exemple de gesturi ce pot fi modelate. Începând de la stânga ele sunt numerotate cu 1 până la 9.

Cu aceste două informații avem la dispoziție o gamă mai variată de posturi 2D care pot fi modelate în sistem. Exemplificăm condițiile ce trebuie satisfăcute pe siluetele din Figura 6.

Pentru postura 1 trebuie să găsim un singur vârf și acela să fie etichetat ca deget mare. Postura 2 necesită existența a două vârfuri dintre care unul să fie degetul mare. Următoarea postură este caracterizată de prezența celor 5 degete. Vecina ei este definită prin lipsa vârfurilor dar prezența centrului palmei. Postura 5 este dată de aceleași condiții ca 2, dar mai adăugăm condiția ca unghiul format de cele două vârfuri să fie ascuțit și sub 70° . Pentru următoarea postură avem nevoie de prezența unui vârf care să nu fie etichetat nici ca deget mare, nici ca deget mic. Postura 7 necesită existența a două vârfuri neetichetate, iar postura 8 are nevoie de trei. Ultima este condiționată de prezența a două vârfuri, dintre care unul să fie degetul mic, iar celălalt neetichetat. Limitări ale modelării programatice exemplificate sunt: posturile trebuie să poată fi distincte în termenii cunoscuți de aplicație; mâna poate lua alte forme diferite, dar care satisfac cerințele modelate de clasificare.

5.2 Urmărirea trăsăturilor

Extragerea trăsăturilor nu este lipsită de erori. Elemente precum tremuraturile mâinilor, condițiile de iluminare, ocluziunile, mișcări bruște ale mâinilor și limitări ale algoritmilor, determină apariția abaterilor bruște ale pozițiilor trăsăturilor în anumite cadre. Pentru ca sistemul să fie robust împotriva acestor abateri se folosește filtrul Kalman, Welch & Bishop (1995). Acesta este capabil să estimeze stări curente pe baza unui model de actualizare a stărilor, împreună cu informații acumulate din stări trecute și un model al erorii în procesul de observare. Deși OpenCV oferă cadrul prin care se poate folosi acest filtru, el trebuie configurat cu un anumit model, iar apoi parametrii lui trebuie inițializați. Figura 7 pune în prim plan principalii parametri.

Starea reținută de filtru sunt coordonatele 2D ale trăsăturii împreună cu viteza de deplasare pe cele două axe. Elementele de măsurat reprezintă poziția 2D. Modelul de actualizare al mișcării îl reprezintă legile mișcării, în care accelerația dintre două momente de timp se presupune a fi constantă. Ecuația prin care se actualizează starea filtrului ținând cont de starea precedentă, modelul zgomotului și eroarea în măsurare se găsește în (2). F

este matricea de tranziție a stării, B este matricea modelului de control aplicat vectorului de control u_t , iar E_x modelează zgomotul prezent în proces. În (3) avem actualizarea observației pe baza stării estimate. H este matricea de tranziție a observației iar E_z reprezintă matricea de covarianță a unui model de zgomot gaussian alb de medie zero asociat observației.

$$\begin{aligned} stare = x_t &= \begin{pmatrix} pos_x \\ pos_y \\ v_x \\ v_y \end{pmatrix} & observația = z_t &= \begin{pmatrix} pos_x \\ pos_y \end{pmatrix} \\ x_{t+1} &= F \cdot x_t + B \cdot u_t + E_x \\ z_t &= H \cdot x_t + E_z \end{aligned}$$

Figura 7. Parametrii filtrului Kalman împreună cu starea și observația aplicației.

Valorile parametrilor sunt dați în Figura 8.

$$\begin{aligned} F &= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & B &= \begin{bmatrix} \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} \\ \Delta t \\ \Delta t \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} & E_x &= \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \cdot \sigma_x \end{aligned}$$

Figura 8. Parametri filtrului Kalman folosiți de aplicație.

Matricea de covarianță aferentă zgomotului observației E_z are toate elementele 0.3. Această aproximare nu reflectă realitatea, iar în cazul ideal ar trebui estimate din fiecare cadru imagine. La fel este valabil și pentru deviația standard a zgomotului de proces. Se dorește însă studierea acestei probleme în viitor. Cu această configurație sunt folosite 3 filtre Kalman pentru estimarea poziției centrului palmei și poziției celor două puncte care definesc încheietura.

Determinarea direcției de mișcare a mâinii se face folosind panta dreptei formate de locația centrului palmei din cadrul precedent și cel curent.

Valoarea ei ne spune înclinația, iar semnul, dacă este crescătoare sau descrescătoare. Locațiile relative a celor două centre din cadre succesive ne spun dacă direcția este de la dreapta la stânga sau invers. Deoarece distingem numai între cele 4 direcții, este necesară definirea unor limite privind ceea ce aplicația va clasifica, ca fiind vertical față de oblic. Figura 9 arată diferitele valori posibile.

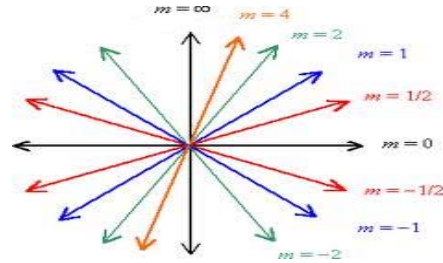


Figura 9. Limite posibile ale pantei pentru a decide între direcția de sus sau jos și direcția de stânga sau dreapta

În aplicație, orice valoare aparținând $[1; \infty)$ sau $[-1; -\infty)$ se consideră a fi direcție verticală. O decizie finală se face prin vot majoritar asupra direcțiilor deduse din punctele consecutive ale traiectoriei mâinii. Determinarea momentului de început și sfârșit a traiectoriei mâinii, se face calculând viteza instantanee a mâinii între cadre succesive. Valoarea rezultată se compară cu un prag corespunzător vitezei minime, peste care mâna se află în mișcare. Mișcarea se sfârșește când această valoare coboară sub pragul aminitit.

5.3 Recunoașterea gesturilor dinamice

Un gest dinamic este alcătuit din direcția de deplasare a unui gest static, de-a lungul unui număr minim de cadre. Direcția, după felul în care se calculează, poate declanșa un eveniment către o interfață a aplicației client, însă componenta de recunoaștere statică detectează doar postura prezentă în cadrul curent. Se folosește un algoritm de monitorizare în timp a gesturilor pentru a da o decizie finală. El este aplicat pentru un gest static determinat și acceptă la intrare prezența ,1, sau lipsa lui, 0 în cadrul curent. Algoritmul folosește o coadă de lungime N care reține prezența sau absența lui în ultimele N cadre trecute. La fiecare cadru procesat, o nouă valoare de adevăr se adaugă la începutul listei și o valoare veche se elimină de la capătul ei.

Când coada este plină se numără numărul maxim de valori de 1 consecutive din ea, n . Reținem, de asemenea, o variabilă *Contor* inițializată la 0. Dacă $n > T_{\text{posibil}}$ atunci $\text{Contor} = \text{Timer}_{\text{posibil}}$. Altfel dacă $n > T_{\text{probabil}}$ atunci $\text{Contor} = \max(\text{Timer}_{\text{probabil}}, \text{Contor})$. În caz contrar se decrementează *Contor*. Dacă *Contor* este mai mare decât 0 atunci acel gest este considerat prezent în cadru. $T_{\text{posibil}} < T_{\text{probabil}} \leq N$. Figura 10 ilustrează algoritmul.

```

Intrare:  $\text{gestPrezent} \leftarrow 1, 0$ 
Initializare:  $\text{buffer} = \{0, \dots, 0\}$ ,  $n \leftarrow 0$ ,  $\text{contor} \leftarrow 0$ 
1: if  $|\text{buffer}| < N$  then
2:    $\text{buffer}[n] \leftarrow \text{gestPrezent}$ 
3:    $n \leftarrow n + 1$ 
4: else
5:    $\text{buffer} \leftarrow \text{buffer} \setminus \text{buffer}[0]$ 
6:    $\text{buffer}[N] \leftarrow \text{gestPrezent}$ 
7: end if
8: if  $|\text{buffer}| = N$  then
9:    $\text{max}_c \leftarrow \text{count}(\text{buffer})$  {numără lungimea maximă a aparițiilor consecutive de 1}
10:  if  $\text{max}_c > T_{\text{posibil}}$  then
11:     $\text{contor} \leftarrow \text{Timer}_{\text{posibil}}$ 
12:  else
13:    if  $\text{max}_c > T_{\text{probabil}}$  then
14:       $\text{contor} \leftarrow \max(\text{Timer}_{\text{probabil}}, \text{contor})$ 
15:    else
16:       $\text{contor} \leftarrow \text{contor} - 1$ 
17:    end if
18:  end if
19: end if
20: return  $\text{contor} > 0$  {True  $\rightarrow$  gestul este prezent în cadru}

```

Figura 10. Algoritmul de monitorizare a gestului în timp

Logica expusă se bazează pe continuitatea gesturilor în timp și pe o perioadă minimă de persistență. Astfel se resping cazuri de clasificări incorecte a unor gesturi și se tolerează situații în care un gest nu este prezent, dar avem informații trecute care indică prezența lui. Valorile alese pentru variabile sunt $N = 7$, $T_{\text{posibil}} = 3$, $T_{\text{probabil}} = 5$, $\text{Timer}_{\text{posibil}} = 5$, $\text{Timer}_{\text{probabil}} = 15$. În momentul detecției sfârșitului mișcării, se interoghează starea contorului și se returnează prezența sau absența gestului. Algoritmul este rulat individual pentru fiecare gest static modelat. Dacă sunt returnate mai multe gesturi atunci se alege acela cu variabila *contor* cea mai mare.

6. Evaluare și rezultate

Evaluarea se face la nivelul: segmentării, determinării încheieturii, recunoașterii statice și dinamice a gesturilor.

6.1 Evaluarea segmentării

Etapa de segmentare este importantă din punctul de vedere al timpului de procesare și robusteții. O regiune a mâinii perturbată de alți pixeli care nu fac parte din silueta ei, pot produce erori în determinarea centrului palmei. Prezența pixelilor din fundal în imaginea segmentată crește procesarea pentru găsirea de contururi. De aceea este nevoie de o evaluare a metodelor de segmentare.

Testarea se face pe baza de imagini, BD (2014). Indicatorii măsurați sunt: procentajul pixelilor pielii corect segmentați (CDR), procentajul pixelilor diferiți de culoarea pielii incorecți detectați (FDR), procentajul pixelilor corecți clasificați (CR). Rezultatul efectuării testelor este sumarizat în Tabelul 1.

Tabel 1 Rezultatele evaluării implementării algoritmilor pe baza de imagini

Algoritm	CDR (%)	FDR (%)	CR (%)
YCbCr,Chai & Ngann (1999)	83.70	29.85	72.94
HSV, Saxe & Foulds (1996)	77.50	25.72	68.13
LCCS, Bishesh& Desire (2011)	89.34	21.67	76.89

Segmentarea în fiecare spațiu de culoare s-a realizat utilizând intervalele de culoare menționate în fiecare lucrare în parte. Deoarece condițiile de iluminat din imagini au fost necontrolate, rezultatele obținute sunt interpretate ca o expectanță pentru diversele situații reale.

6.2 Evaluarea algoritmului de determinare a încheieturii

Încheietura este cea care ne filtrează cea mai mare parte a vârfurilor degetelor false. Este ușor de văzut cum o localizare greșită determină creșterea erorii. În cel mai rău caz se poate exclude chiar regiunea degetelor mâinii. O evaluare a algoritmului este necesară pentru a cuantifica acest lucru. De

asemenea punctele de încheietură în momentul de față sunt folosite doar ca metodă de filtrare. Ele nu au nici un rol în modelarea gesturilor. Avem astfel un schimb între robustețe la detecția vârfurilor și creșterea timpului de procesare. Din acest motiv vom măsura în această parte și timpul. Întrucât s-au folosit aceleași imagini ca Grzejszczak et al, (2013) s-a preluat și metodologia. Conform ei pentru încheietura din tabela de adevăr $|UV|$ și încheietura determinată de algoritmul $|U'V'|$ se calculează eroarea $e = |W'W| / |UV|$. W', W reprezintă mijloacele segmentelor. Se definesc cercurile $E=0.5$ și $E=1.0$, în care cifrele indică lungimea razei ca procentaj din segmentul $|UV|$. Se consideră că o încheietură este corect găsită dacă eroarea e este mai mică decât constanta E aleasă. Procesul este ilustrat în Figura 11.

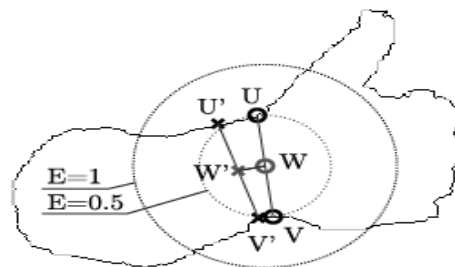


Figura 11. Măsurarea erorii. Se număra punctele din exteriorul primului cerc și cele din exteriorul celui de-al doilea Grzejszczak et al. (2013)

Tabelul 2 prezintă rezultatele algoritmului nostru împreună cu cele publicate de Grzejszczak et al. (2013) pentru toată baza de date.

Tabel 2 Rezultatele evaluării detecției încheieturii

Denumire	LC (Grzejszczak et al. (2013))	Aplicație
Timp (ms)	32-81	22-52
$e > E, E=1.0$ (%)	14.7	12.11
$e > E, E=0.5$ (%)	36.0	23.4

Avantajul metodei noastre constă în determinarea implicită a orientării mâinii dată de dreptunghiul care înglobează silueta. Acest lucru face căutarea după încheietură mai ușoară. Timpul de procesare a fost măsurat pe un laptop având un procesor Intel Core 2 Duo T7300 și 2GB RAM și diferă de platforma utilizată de Grzejszczak et al. (2013).

6.3 Evaluarea gesturilor statice

Modelul de gest static văzut ca o postură a mâinii și descris prin constrângeri ale pozițiilor vârfurilor degetelor față de centrul palmei, este evaluat din punct de vedere al fiabilității.

Se pornește de la aceeași bază de date Bd (2014) din care s-au ales acele imagini în care se regăsesc posturile din Figura 6. Elementul măsurat reprezintă eroarea de clasificare definită ca procentajul nerecunoașterii unui gest când acesta este prezent. Rezultatele se găsesc în Figura 12.

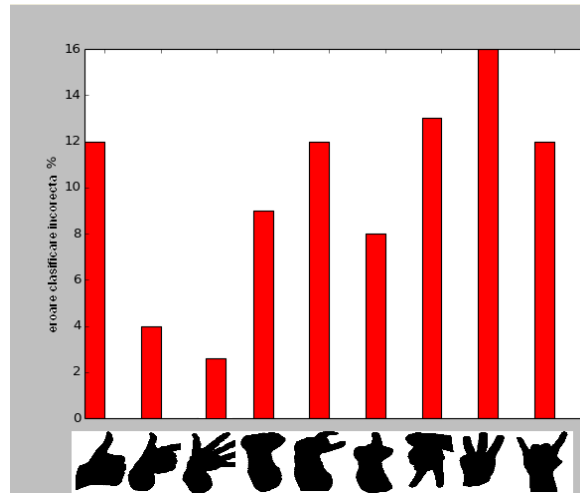


Figura 12. Rezultatele recunoașterii statice de gesturi.

Rezultatele arată că trăsăturile și modelele definite pe baza lor sunt potrivite pentru folosirea în recunoașterea de gesturi.

6.4 Evaluarea gesturilor dinamice

În acest caz se utilizează baza de secvențe video creată de Liu & Shao (2013). Un gest dinamic este alcătuit din direcția de deplasare a unui gest static, de-a lungul unui număr minim de cadre. Pentru a folosi baza de date, mișcările din secvențele video se definesc ca o secvență, de cele 4 direcții de mișcare manuală, de către un operator uman. Pentru fiecare secvență video se verifică gesturile rezultate, cu cele definite pentru acea secvență.

Tabel 3 Rezultate recunoaștere gesturi dinamice

Postură	Timp procesare (ms)	Rata eroare (%)
Pumn	96-134	26
Palmă	89-127	13
Deget indicator	110-143	15

Eroarea de clasificare incorectă, reprezintă procentajul gesturilor dinamice pentru care condiția de dinainte nu este adevărată. Pentru fiecare secvență se măsoară timpul de procesare. Acești indicatori se calculează la nivelul întregii baze de date. Rezultatele sunt grupate după postura mâinii.

7. Limitări.Îmbunătățiri.Concluzii

Limitările actuale ale aplicației se datorează algoritmilor folosiți. Segmentarea pe baza intervalului de culoare a pielii limitează fundalul în care se poate afla mâna. Lipsa clasificării între față și mână este un aspect negativ. Cazuri în care încheietura mâinii este puțin vizibilă și algoritmul prezentat dă greș, contribuie la rata de eroare. Logica prin care se etichetează degetul mic și cel mare, nu ține cont de perspectiva camerei, astfel că sunt poziții în care degetul mare va fi în imagine mai lung decât cel mic. Mișcările reduse recunoscute sunt o altă limitare a aplicației.

În vederea adresării acestor limitări se enumeră locurile în care aplicația poate fi îmbunătățită. Se poate adăuga un detector de fețe care în intervale regulate să identifice regiunile feței și să le excludă din analiză. Pentru segmentare se poate folosi o metodă hibridă, care combină tehnica de diferențiere a imaginilor succesive, cu cea a segmentării pe baza culorii. Ultima îmbunătățire amintită este extinderea dinamicii gesturilor pentru detecția mișcărilor, precum triunghi sau cerc.

În concluzie, lucrarea de față a prezentat o serie de soluții pentru a construi o aplicație capabilă să recunoască gesturi dinamice simple. Prin rezultatele obținute se arată că trăsăturile pot modela o varietate de gesturi, iar tehnicile de determinare a lor, permit o implementare eficientă, făcând posibilă înglobarea acestuia în sisteme mai mari de timp real.

Referințe

- Borgefors, G. (1986) Distance transformations in digital images, *Computer Vision, Graphics and Image Processing*, Vol. 34, 344-371.
- Bresenham, J. E (1977) A linear algorithm for incremental digital displaz of circular arcs, *Communications of the ACM*, 100-106.
- Chai, D., Ngan, K.N., 1999 Face segmentation using skin-color map in videophone applications, *IEEE Transactions on Circuits and System for Video Technology*, Vol. 9, 551-564.
- Chaudhary, A., Raheja, J.L, Das, K., 2011 Intelligent approaches to interact with machines using hand gestrure recognition in natural way: a survey, *International Journal of Computer Science and Engineering Survey*, Vol. 2, 122-133.
- Du, W., Li, H. 2000 Vision based gesture recognition system with single camera, *5th International Conference on Signal Processing Proceedings WCCC-ICSP*, Vol. 2, 1351-1357.
- Fakhreddine, K., Milad, A., Jamil, A.S., Mo, N., 2008 Human-Computer Interaction: Overview on State of the Art, *Canada International Journal on Smart Sensing and Intelligent Systems*, 137-159.
- Grzejszczak, T., Nalepa, J., Kawulok, M., 2013: Real-time wrist localization in hand silhouettes, *Conference on Computer Recognition System*, 439-449.
- Ke, W., Li, W., Ruifeng, L., Lijun, Z., 2010 Real-Time Hand Gesture Recognition for Service Robot, *Intelligent Computation Technology and Automation*, Vol. 2, 976-979.
- Khanal, B., Sidibé, D. (2011) Efficient kin detection under severe illumination changes and shadow, *Intelligence Robotics and Applications*, Springer, Vol. 7102, 609-618.
- Kulshreshth, A., Zorn C., LaViola, J.J., 2013 Poster: Real-time markerless kinect based finger tracking and hand gesture recognition for HCI, *3D IEEE Symposium on User Interfaces*, 187-188.
- Liu, L., Shao, L. (2013): Learning Discriminative Representations from RBG-D Video Data, *Proccedings International Joint Conference on Artificial intelligence*, 1493-1500.
- Maung, T.H.H. (2009) Real-Time Hand Tracking and Gesture Recognition System using Neural Networks, *Proceedings of World Academy of Science Engineering and Technology*, Vol. 3, 393-397.
- Mokhtar, M.H., Proamod, K.M. (2012) Real time fingers and palm locating using dynamic circle templates, *International Journal of Computer Applications*.
- Parker, J.R., Baumbach, M. (2009) Finger recognition for hand pose determination, *IEEE International Conference on Systems, Man and Cybernetics*, 2492-2497.
- Saxe, D., Foulds R. (1996) Towards robust skin identification in video images, *Proceedings 2nd conference of Automatic Face and Geture Recognition*, 379-384.
- Welch, G., Bishop, G. (1995): An introduction to the Kalman filter, *Computer Science Technical Report*.
- BD 2014, Sit către baza de imagini statice cu posturi ale mâinii
<http://sun.aei.polsl.pl/~mkawulok/gestures/>.