

Interacțiunea cu un browser web prin intermediul gesturilor

Mihaela Coneac, Adrian Iftene

Universitatea "Al. I. Cuza" din Iași, Facultatea de Informatică
General Berthelot, Nr. 16, 700483, Iași
E-mail: mihaela.coneac@infoiasi.ro, aiftene@infoiasi.ro

Rezumat. Articolul de față are ca scop prezentarea unei noi modalități de a interacționa cu un calculator personal. Astfel, un utilizator poate folosi camera web și gestica mâinii pentru a da comenzi uzuale atunci când folosește un browser web ca Internet Explorer. După un proces de antrenare a aplicației, în care aceasta este învățată ce înseamnă posturile palmă, două degete sau pumn, folosim algoritmul AdaBoost pentru a clasifica o nouă postură recepționată în timp real de la utilizator. Combinând etapa de identificare a posturii cu etapa de identificare a traiectoriei acesteia obținem comanda pe care utilizatorul o trimite browser-ului. Operațiile implementate ne permit să facem scroll vertical, zoom, salt între ferestre, salt între opțiuni și selectarea unei opțiuni. Rezultatele obținute până în prezent sunt încurajatoare și ne fac să sperăm că acest mod de interacțiune va fi din ce în ce mai folosit pe viitor.

Cuvinte cheie: Gestica mâinii, interacțiune om-calculator, algoritm de clasificare.

1. Introducere

În ultimii ani s-a acordat o atenție deosebită modurilor neconvenționale de interacțiune cu dispozitivele mobile și chiar cu calculatorul. Sunt binecunoscute ecranele LCD de tip *touchscreen* existente pe dispozitivele de tip *smartphone* ca iPhone, pe Tablet PC-uri sau chiar pe ecranele calculatoarelor. Aceste tipuri de ecrane ne permit să interacționăm cu dispozitivele, prin folosirea directă a degetelor fără utilizarea unui mouse sau a unei tastaturi. De asemenea este binecunoscută folosirea vorbirii în comunicarea inteligentă om-calculator (Candance et al., 1997). Camerele video conectate la un calculator pot prelua imagini de la un utilizator, care poate accesa meniuri speciale ale aplicațiilor soft. Acum cu ajutorul acestor camere folosind mișcări ale mâinii puteți avea acces direct la obiectele dintr-o aplicație de tip CAD și puteți selecta, roti, mări sau micșora obiectele cu care lucrați (Gorgan et al., 2008). Împătimitii de jocuri 3D pe

consolă sau pe calculator pot folosi telecomenzi speciale care au incorporate senzori de mișcare. De exemplu, cu telecomanda pentru Wii (<http://ro.wikipedia.org/wiki/Wii>) jucătorul poate controla personajul unui joc prin intermediul controlerului. Pornirea calculatorului sau semnalarea faptului că ai ajuns sau plecat de la serviciu se face folosind recunoașterea mâinii cu metode biometrice, deoarece oferă o foarte mare securitate (Öden et al., 2001). Gestica mâinii poate fi folosită în aplicații de desenare, de explorare pe Internet, de navigare între paginile unei prezentări de tip PowerPoint (von Hardenberg și Bérard, 2001) sau în aplicații ce construiesc o realitate augmentată (Reifinger et al. 2007).

Lucrarea de față reprezintă o extindere a lucrării (Coneac și Iftene, 2011) și se încadrează în domeniul folosirii gesticii mâinii pentru a interacționa cu un calculator. Dintre tehnicile cele mai folosite pentru identificarea gesturilor mâinii amintim folosirea recunoașterii de șabloane (*pattern recognition*) (Freeman și Roth, 1994), (Sánchez-Nielsen et al., 2003), (Chakraborty et al., 2008), (Roomi et al., 2010), folosirea histogramelor pentru identificarea orientării și folosirea metodelor care nu depind de schimbările de iluminare (Freeman și Roth, 1994), folosirea mașinilor cu stări finite pentru a modela gesturile (Davis și Shah, 1994). Alte abordări au la bază identificarea degetelor (von Hardenberg și Bérard, 2001) sau a caracteristicilor geometrice (Öden et al., 2001). În (Kölsch și Turk, 2004), autorii prezintă o abordare interesantă, care folosește Viola-Jones pentru a identifica diferite posturi ale mâinii și analizează limitele rotației acestora pentru care se obține o performanță bună de detecție. HandVu (Kölsch et al., 2004) reprezintă o colecție de librării ce permit recunoașterea și urmărirea în timp real a posturilor mâinii, acestea putând fi folosite în aplicații ce redau fișiere video. Cele mai recente abordări construiesc modelele mai complexe pe baza imaginii recepționate și au rezultate mai bune decât abordările anterioare. Amintim aici folosirea modelului Gaussian (Roomi et al., 2010), folosirea rețelelor neuronale (Ahmeda et al., 2008), și recunoașterea de caracteristici multi-culoare (Bretzner et al., 2002).

În continuare lucrarea e organizată după cum urmează. În capitolul 2 sunt prezentate principalele elemente utilizate în lucrare, precum mâna umană, detecția în timp real, trăsătură Haar, imagine integrală și algoritmul AdaBoost. În capitolul 3 sunt prezentate experimentele pe care le-am realizat în analiza mișcării mâinii și în detecția posturilor mâinii. Arhitectura aplicației și evaluarea ei sunt prezentate în capitolele 4 și 5. La final avem

concluziile și direcțiile viitoare de cercetare.

2. Elemente utilizate în lucrare

2.1 Mâna umană

Mâna umană este formată din oase conectate prin articulații și are o structură foarte complexă. Din cauza celor 27 de grade de libertate ale mâinii (Napier, 1980), recunoașterea gesturilor acesteia reprezintă o problemă foarte complexă. În recunoașterea gesticii umane am ținut cont de două concepte de bază:

- ✓ *Postura mâinii*, care se referă la poziționarea statică a mâinii;
- ✓ *Gestica mâinii*, care este o secvență succesivă de posturi ale mâinii conectate prin mișcarea mâinii sau a degetelor într-o perioadă mică de timp.

De exemplu, ținerea pumnului într-o poziție este o postură, iar mișcarea acestuia de sus în jos reprezintă un gest.

2.2 Detecția, recunoașterea și urmărirea obiectelor în timp real

Domeniul *detecția și recunoașterea obiectelor* este legat în primul rând de domeniile securității și supravegherii. După 11 Septembrie 2001, toate țările au acordat o atenție deosebită domeniilor securitate și supraveghere (Fuchs, 2009) cu scopul de a preveni atentatele care pot avea loc în primul rând în aeroporturi, dar și în gări și în stații de metrou. În prezent o atenție deosebită s-a acordat algoritmilor necesari în recunoașterea fețelor. Chiar dacă algoritmi existenți astăzi necesită o putere computațională foarte mare și chiar dacă aceștia nu reușesc să ne ofere în timp real rezultatele de care avem nevoie, procesările realizate de aceștia ne pot oferi informații prețioase și ne pot avertiza de eventuale pericole. În prezent, supravegherea în marile aeroporturi sau gări ale lumii se face cu ajutorul operatorilor umani special pregătiți în identificarea de comportamente anormale și a persoanelor periculoase.

În procesul de detecție și de recunoaștere a obiectelor am realizat următoarele operații:

- ✓ *Urmărirea obiectelor*: localizarea dinamică a obiectelor prin determinarea poziției acestora în fiecare imagine dintr-o succesiune

de imagini;

- ✓ *Detectarea de obiecte*: localizarea claselor generice de obiecte din imagine (mâna în cazul nostru);
- ✓ *Recunoașterea de obiecte*: clasificarea obiectelor specifice unei categorii dintr-o imagine (cum ar fi identificarea posturii unei mâini în cazul nostru).

Pentru a măsura *acuratețea* în detecția și recunoașterea obiectelor am folosit caracteristicile: *rată de succes* (procentul obiectelor corect identificate) și *rată de eșec* (procentul imaginilor clasificate incorect).

2.3 Trăsătură Haar

În (Viola și Jones, 2001), autorii prezintă o abordare statistică folosită în detecția fețelor umane. În studiul lor, au descris o *trăsătură Haar* ca fiind un șablon care include dreptunghiuri albe și negre interconectate, dimensiunea acestora, precum și poziția relativă față de originea ferestrei de căutare. În Figura 1 avem trăsăturile Haar propuse în (Lienhart și Maydt, 2002).

Valoarea unei trăsături Haar este diferența dintre suma pixelilor din dreptunghiul negru și suma pixelilor din dreptunghiul alb. Aceste valori trebuie să îndeplinească în plus condiția compensației descrisă în (Whitehill și Omlin, 2006).

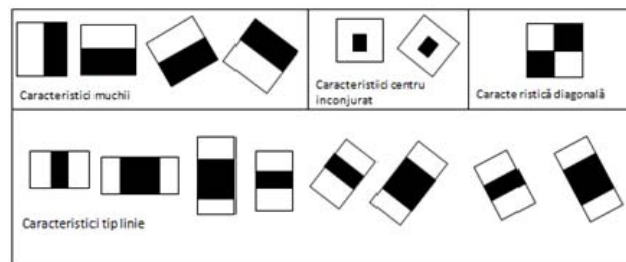


Figura 1. Setul extins de caracteristici Haar

2.4 Imagine integrală (Summed Area Table-SAT)

Suma pixelilor dintr-o regiune I se calculează folosind w (lățimea regiunii I), și h (înălțimea regiunii I). Această operație se va realiza în timpul $O(n)$, unde $n = w * h$ (operație, care în funcție de dimensiunile regiunii I , poate dura foarte mult).

Viola și Jones prin introducerea termenului de „*image integrală*” au eliminat acest inconvenient al duratei mari de execuție și au putut realiza această operație în timp constant $O(1)$. Astfel „*imagea integrală*” de la locația (x, y) , $ii(x, y)$, reprezintă suma pixelilor din dreptunghiul mărginit stânga sus de punctul $(0, 0)$ și dreapta jos de punctul (x, y) (Viola și Jones, 2001).

2.5 Extindere SAT – RSAT (Rotated Summed Area Table)

În (Lienhart și Maydt, 2002), autorii au introdus conceptul de RSAT („Rotated Summed Area Table”) pentru trăsăturile Haar ce conțin dreptunghiuri rotite cu 45° . RSAT reprezintă suma pixelilor din dreptunghiul rotit având colțul din dreapta jos în punctul $ii(x, y)$ și extins până la marginile imaginii. Figura 2 conține un astfel de dreptunghi rotit cu 45° .

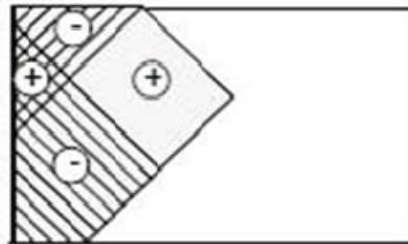


Figura 2. Exemplu de dreptunghi rotit cu 45° (Lienhart și Maydt, 2002)

În Figura 3 am exemplificat modul în care detectăm o mână folosind sub-ferește ce conțin trăsături Haar.



Figura 3. Detectarea mâinii cu o sub-fereastră ce conține o trăsătură Haar

2.6 Algoritmul AdaBoost

Pentru a decide dacă o imagine conține sau nu o mână am folosit un clasificator, care a fost îmbunătățit cu algoritmul de învățare AdaBoost. Algoritmul folosește sub-ferestre cu ajutorul cărora parcurge imaginea curentă folosind o trăsătură Haar. Pentru detectarea obiectului de interes (mâna în cazul nostru), imaginea este scanată cu ajutorul unei sub-ferestre cu o anumită caracteristică Haar.

Un clasificator, f , este definit ca o funcție $f: X \rightarrow Y$, unde X este mulțimea de imagini, iar, iar $Y = \{\pm 1\}$ (+1 dacă imaginea conține mâini și -1 dacă nu conține). Pentru fiecare element f_i , corespunzător unei imagini X_i , se definește un clasificator corespunzător $h_i(x)$ astfel:

$$h_i(x) = \begin{cases} 1, & \text{dacă } p_i f_i(x) < p_i \theta_i \\ -1, & \text{alfel} \end{cases}$$

unde x este sub-ferestra din X_i , θ_i este pragul, iar p_i indică direcția semnului de inegalitate. Algoritm AdaBoost folosește tehnica de „Boosting” (un algoritm de învățare bazat pe o serie de clasificatori slabi) (Freund și Schapire, 1999).

3. Experimente

3.1 Analiza mișcării mâinii

Studiind traiectoria pe care o realizează mâna umană când se dorește deplasarea pe o dreaptă am observat că e dificil să se obțină o traiectorie perfectă. Din acest motiv am decis să încercăm încadrarea traiectoriei într-un interval prestabilit (apropiat unei drepte în cazul de mai sus), și dacă această operație are loc cu succes să considerăm că am identificat traiectoria mâinii umane.

Înainte de a le folosi efectiv în aplicație, am studiat două tipuri de traiectorii: *verticală* și *orizontală*. Motivul alegerii acestor două tipuri e legat de simplitatea realizării lor de către utilizatori, și totodată de ușoara analiză a lor.

Astfel, pentru a determina tipul traiectoriei realizate de mâna utilizatorului, orizontală sau verticală, aceasta va fi încadrată într-un

dreptunghi de o lățime sau lungime predefinită. Cu cât dimensiunile dreptunghiului sunt mai mici, cu atât traiectoria este mai apropiată de cea a unei drepte, și evident precizia este mai mare.

În urma experimentelor efectuate, pentru a oferi utilizatorilor o ușurință cât mai mare în utilizarea aplicației am decis ca aceste dimensiuni să fie de 50 pixeli. În Figura 4 avem o traiectorie a mâinii utilizatorului care a fost încadrată la traiectorie orizontală.



Figura 4. Specificarea traiectoriei orizontale într-un dreptunghi cu lățimea de 50 pixeli

3.2 Detectia posturilor mâinii

În experimentele pe care le-am realizat am testat trei posturi ale mâinii: *palma*, *două degete* și *pumn* (Figura 5).



Figura 5. Posturile palmă, două degete și pumn

Algoritmul AdaBoost a primit ca intrare două tipuri de imagini: *imagini pozitive* (care conțin posturi ale mâinii) și *imagini negative* (care nu conțin posturi ale mâinii).

Obținerea imaginilor pozitive

Imaginile pozitive au fost colectate folosind diverse fundaluri, cu diverse variații de iluminare și considerând diferite dimensiuni. Inițial într-o prima etapă, pentru fiecare postură am considerat 750 de imagini. Mai apoi, în cea de a doua etapă, pornind de la aceste imagini, am folosit funcțiile oferite de

biblioteca OPENCV (<http://opencv.willowgarage.com/wiki/>), care ne permit să realizăm rotații în spațiul tridimensional, și am obținut 7.000 de imagini pozitive. Pentru ca scara imaginilor ce conțin posturile să nu afecteze procesul de antrenare, în cadrul creării clasificatoarelor, acestea au fost ajustate la o dimensiune de 15 x 28 pixeli (pentru postura palmă), 15 x 30 pixeli (pentru postura două degete), respectiv 20 x 20 pixeli (pentru postura pumn).

Obținerea imaginilor negative

Cu (Tutorial Haar Negatives, 2011) am obținut 3000 de imagini ce nu conțin niciuna dintre posturile mâinii. *Imaginile negative* pe lângă faptul că nu conțin posturi ale mâinii, trebuie să conțină o variație a nivelurilor de gri, a formelor, a muchiilor, și a texturilor astfel încât clasificatorul slab să poată obține o mai bună acuratețe prin eliminarea unor mari varietăți de mostre negative. Figura 6 ilustrează câteva imagini negative folosite în procesul de antrenare.



Figura 6. Imagini negative folosite pentru antrenarea celor trei clasificatoare

Antrenarea

Pentru antrenarea clasificatoarelor am folosit o succesiune de 20 de runde. Pentru fiecare rundă în execuția cascadei de clasificatori, am stabilit rata minimă de succes la 96.9% și rata maximă de eșec la 50%. La final, pentru postura palmă am obținut o rată de succes a clasificatorului final de 90%,

pentru postura două degete am obținut o rată de succes de 92%, iar pentru postură pumn rata de succes a fost de 97%. Figura 7 prezintă trăsăturile Haar asociate celor 3 posturi.







Postura mâinii	Trăsătura Haar
	
	
	

Figura 7. Trăsăturile Haar asociate posturilor

Evaluarea

Pentru a evalua performanța clasificatorilor obținuți am considerat 100 de imagini de test pentru fiecare postură. Tabelul 1 ilustrează atât performanțele cascadelor de clasificatori, cât și timpul necesar procesării celor 100 de imagini.

Tabelul 1. Performanța clasificatorilor antrenați

Postura	Potriviri	Nepotriviri	False	Durată detecție (s)
Palma	90	10	50	3.00400
Două degete	92	8	30	2.98700
Pumn	97	3	20	1.81900

Coloana *Potriviri* din Tabelul 1 ilustrează numărul de identificări

corecte, coloana *Nepotriviri* prezintă numărul de posturi omise de clasificator, coloana *False* reține numărul de detecții false ale acestuia, iar ultima coloană *Durata detecției* precizează care este timpul necesar pre-procesării tuturor imaginilor folosite mai apoi în etapa de testare.

De exemplu, pentru postura *Palmă* putem observa că au fost identificate corect 90 de posturi, 10 posturi au fost omise, astfel înregistrându-se 50 de detecții false (datorită fundalului aglomerat ce conține arii mici a cărui nivel de gri este asemănător cu cel al posturii palmă).

Figura 8 prezintă câteva imagini în care este corect detectată postura palmă.

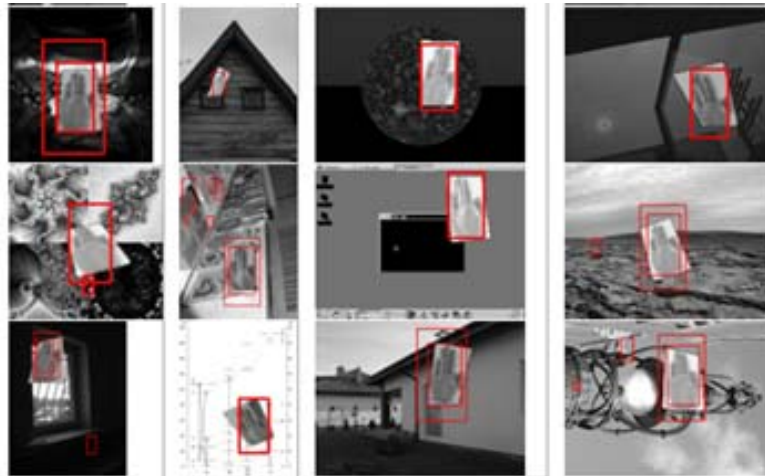


Figura 8. Rezultatele obținute în urma testării cascadei de clasificatori „palmă”

Pentru testare am folosit o camera-web setată să capteze imagini la o rezoluție de 320 x 240 pixeli la un interval de 15 cadre/secundă și nu s-a înregistrat nici o latență în detecția sau urmărirea posturilor. Rezultatele evaluării au demonstrat robustețea clasificatorilor antrenați la variația de lumină sau de rotație cu $\pm 15^\circ$ în plan.

Prin analiza rezultatelor de detecție am remarcat faptul că rotațiile excesive provoacă omiterea unor imagini pozitive. De asemenea, majoritatea detecțiilor false sunt identificate în arii foarte mici, ce au o probabilitate foarte mare de a conține același nivel de gri ca al obiectul antrenat. Pentru a elimina această inconveniență am ajustat pragul pentru

sub-fereastra de scanare.

4. Arhitectura aplicației

Pentru a demonstra eficiența sistemului nostru de identificare a posturilor, am realizat o aplicație pentru browser-ul Internet Explorer. Cu ajutorul acestei aplicații, utilizatorul poate interacționa cu acest browser prin intermediul camerei web, folosind următoarele operații de navigare:

- ALT + TAB (saltul între ferestre) – prin mișcarea orizontală a palmei;
- SCROLL vertical (defilarea verticală a ferestrei) – prin mișcarea verticală a pumnului;
- ZOOM (mărirea sau micșorarea conținutului ferestrei active) – prin mișcarea pumnilor din interior/exterior spre exterior/interior;
- CLICK (selectarea unei opțiuni) – prin mișcarea verticală a posturii două degete;
- TAB (salt de la o opțiune la alta) – mișcarea orizontală a posturii două degete.

Iată reprezentarea grafică a două comenzi: comanda ALT + TAB în Figura 9 (cu cele două variante: prima, *de a selecta fereastra următoare*, caz în care traiectoria mâinii este orizontală de la stânga la dreapta și cea de a doua, *de a selecta fereastra anterioară*, caz în care traiectoria mâinii este orizontală de la dreapta la stânga) și comanda ZOOM-OUT, în Figura 10 (care folosește posturile pumn a ambelor mâini, care realizează simultan o mișcare orizontală dinspre afară spre interior).



Figura 9. Comanda ALT + TAB

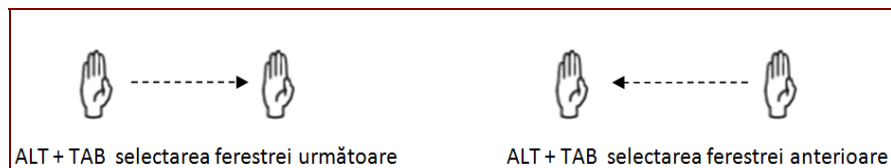


Figura 10. Comanda ZOOM-OUT

Aplicația este alcătuită din 5 module mai importante, pe care le prezentăm în continuare:

- Modulul *Win32* ne oferă informații legate de instanțele active din sistem ale browser-ului web Internet Explorer;
- Modulul *Computer Vision Library* oferă funcțiile necesare procesării imaginilor pentru detecția posturilor și a traiectoriei mâinii (cele mai importante funcții sunt pentru antrenare, și pentru clasificare);
- Modulul *Periferal Service* ne permite accesul la capturile realizate de camera web;
- Modulul *User Interface* afișează comenzile realizate de gesturile utilizatorului;
- Modulul *Interaction Engine* este modulul central al aplicației.

Sucesiunea pașilor executați de algoritm sunt dați în continuare. Astfel, după pornirea aplicației:

Pas 1. Modulul *Interaction Engine* comunică cu modulul *Periferal Service* pentru a captura imaginea curentă oferită de cameră.

Pas 2. Imaginea primită este trimisă către modulul *Computer Vision Library*, care o procesează și identifică postura curentă a mâinii. De asemenea, pe baza procesării unei succesiuni de imagini se identifică traiectoria mâinii. Combinând etapa de identificare a posturii cu etapa de identificare a traiectoriei se obține gestul utilizatorului uman.

Pas 3. În urma procesării acesta îi transmite modulului *Interaction Engine* că s-a realizat un gest și care este acesta (și se merge la Pas 4) sau îi transmite că nu s-a realizat un gest (și se reia de la Pas 1).

Pas 4. Modulul *Interaction Engine* produce comenzile pentru browser Internet Explorer cu ajutorului modulului *Win32* și notifică modulul *User Interface* cu privire la noile schimbări. Se merge la Pas 1.

5. Evaluare preliminară

Pentru a putea evalua într-o primă etapă aplicația pe care am realizat-o, am apelat la 10 persoane nespecialiste în domeniul informaticii, dar care posedă cunoștințe despre navigarea pe Internet. Pornind de la o pagină web aparținând unui ziar online, participanții trebuiau (1) să citească un articol ce depășea o pagină (în acest fel ei erau obligați să facă scroll și să facă operații de mărire sau micșorare a textului), (2) să selecteze una din secțiunile preferate din meniul paginii web pentru a găsi un articol de actualitate și (3) să facă salt între mai multe ferestre deschise simultan. Scopul experimentelor a fost de a observa atât gradul de comoditate în a folosi comunicarea bazată pe gesturi cu un calculator, cât și robustețea algoritmilor implementați atunci când avem utilizatori noi.

La sfârșitul experimentelor am identificat faptul că aplicația este foarte utilă în special în cazul persoanelor care sunt greu de inițiat în domeniul informaticii, oferindu-le un grad sporit de comoditate atunci când folosesc calculatorul. Acesta este cazul persoanelor cu dizabilități sau al persoanelor care folosesc prima dată calculatorul sau al persoanelor în vârstă, care se adaptează mai greu la tehnologiile mai noi.

Un alt mare avantaj vine din faptul că prin utilizarea trăsăturilor Haar aplicația noastră reușește să realizeze identificarea corectă a posturilor în timp real, acest lucru permițându-ne să putem folosi cu succes gestica mâinii în cazul în care dorim să o aplicăm la prezentări sau la proiecții.

Dezavantaje identificate pe parcursul experimentelor se datorează faptului că aplicația e destul de sensibilă la tranziții între poziții succesive ale mâinii atunci când se dorește urmărirea ei. O altă problemă semnalată de cei care au participat la experiment este datorată faptului că funcționalitățile implementate sunt încă minimale, neeliminându-se total folosirea mouse-ului și a tastaturii.

Pe viitor, intenționăm să realizăm experimentele de mai sus pe un grup mult mai mare de utilizatori cărora să le cerem să răspundă și la câteva întrebări legate de modul în care au folosit aplicația noastră. De asemenea, intenționăm ca într-o a doua etapă să facem o evaluare din punct de vedere al utilizabilității aplicației.

6. Concluzii

Lucrarea de față prezintă o metodă neconvențională de a comunica cu un calculator. Folosind o cameră web care este conectată la calculator utilizatorul poate interacționa cu browser-ul web Internet Explorer folosind gestica mâinii.

Aplicația a folosit ca punct de pornire una din cele mai bune soluții existente în prezent în detectarea de obiecte prezentată în (Viola și Jones, 2001). Originalitatea sistemului nostru constă în faptul că prin modul de construire, inițial avem o etapă de antrenare, urmată de o etapă în care folosim algoritmul AdaBoost pentru clasificare, obținem o robustețe ridicată (operațiile de antrenare sunt independente de utilizatorul care va folosi aplicația).

Comenzile pe care le-am implementat în aplicație ne permit să facem salturi între ferestre (prin mișcarea orizontală a palmei), să facem scroll vertical (prin mișcarea verticală a pumnului), să facem operații de zoom (prin mișcarea pumnilor), să facem click (prin mișcarea verticală a posturii două degete), să facem salt între opțiuni (prin mișcarea orizontală a posturii două degete).

Pe viitor se dorește îmbunătățirea clasificatorilor creați, prin colectarea imaginilor de la un număr ridicat de persoane diferite. Astfel acest sistem va deveni mult mai independent și robust. O altă îmbunătățire ar fi extragerea fundalului din imaginea ce necesită a fi procesată pentru o îmbunătățire substanțială a performanței sistemului.

Alte direcții viitoare de dezvoltare au ca scop oferirea mai multor opțiuni de comunicare cu browser-ul și totodată permiterea simultană a recunoașterii și a altor părți ale corpului uman pentru a crea noi comenzi. Ca aplicațiile să fie mai interactive dorim să integrăm și alte tehnici de comunicare cum ar fi recunoașterea vocii.

Mulțumiri

Autorii doresc să aducă mulțumiri participanților la experimente pentru bunăvoința și pentru răbdarea acordată. Cercetarea prezentată în această lucrare a fost finanțată de către Programul Operațional Sectorial Dezvoltarea Resurselor Umane prin proiectul „Dezvoltarea capacității de inovare și creșterea impactului cercetării prin programe post-doctorale POSDRU/89/1.5/S/49944”.

Referințe

- Ahmeda, S. M. H., Alexander, T. C., Georgios, C., Anagnostopoulos, B. *Real-time, Static and Dynamic Hand Gesture Recognition for Human-Computer Interaction*. TR, 2008.
- Bretzner, L., Laptev, I., Lindeberg, T. *Hand Gesture Recognition using Multi-Scale Colour*

- Features, Hierarchical Models and Particle Filtering*. Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02) 0-7695-1602-5/02 IEEE, 2002.
- Candace, A., Kamm, M., Walker, A., Rabiner, L. R. *The role of speech processing in human-computer intelligent communication*. Speech Communication, 1997.
- Chakraborty, P., Sarawgi, P., Mehrotra, A., Agarwal, G., Pradhan, R. *Hand Gesture Recognition: A Comparative Study*. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, Vol. I, IMECS 2008, 19-21 March, Hong Kong, 2008.
- Conecac, M., Iftene, A. *Folosirea gesticii mâinii în interacțiunea om-calculator*. Revista Română de Interacțiune Om-Calculator 4 (Număr special - RoCHI 2011), Pp. 107-110, București, 8-9 Septembrie, 2011.
- Davis, J., Shah, M. Recognizing Hand Gestures. Computer Vision - ECCV '94 Proceedings of Third European Conference on Computer Vision, Stockholm, Sweden, Volume I, May 2-6, 1994.
- Freeman, W. T., Roth, M. *Orientation Histograms for Hand Gesture Recognition*. IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995.
- Freund, Y., Schapire, R. E. *A short introduction to boosting*. Journal of Japanese Society for Artificial Intelligence, vol. 14, nr. 5, pp. 771 – 780, 1999.
- Fuchs, C. *Social Networking Sites and the Surveillance Society. A Critical Case Study of the Usage of studiVZ, Facebook, and MySpace by Students in Salzburg in the Context of Electronic Surveillance*. Salzburg/Vienna, Austria, 2009; Forschungsgruppe "Unified Theory of Information" - Verein zur Förderung der Integration der Informationswissenschaften, ISBN 978-3-200-01428-2, 2009.
- Gorgan, D., Stefanut, T., Veres, M., Gabos, I. *Tehnici de adnotare grafica în 3D în aplicațiile de e-learning interactive*. Revista Română de Interacțiune Om-Calculator Vol.1, Nr.1, ISSN 1843-4460, 2008.
- Kölsch, M., Höllner, T., DiVerdi, S. *HandVu: A New Machine Vision Library for Hand Tracking and Gesture Recognition*. Demo at ISWC/ISMAR, <http://www.movesinstitute.org/~kolsch/HandVu/HandVu.html>, 2004.
- Kölsch, M., Turkm, M. *Analysis of Rotational Robustness of Hand Detection with a Viola-Jones Detector*. International Conference on Pattern Recognition (ICPR), Cambridge, U.K., August 2004.
- Lienhart, R., Maydt, J. An extended set of Haar-like features for rapid object detection. IEEE ICIP 2002, Pp. 900-903, 2002.
- Napier, J. *Hands*. New York: Panthon Books, 1980.
- Öden, C., Erçil, A., Yildiz, V.T., Kirmizita, H., Büke, B. *Hand Recognition Using Implicit Polynomials and Geometric Features*. AVBPA'01 Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication. Springer-Verlag London, UK, ISBN:3-540-42216-1, 2001.
- Reifinger, S., Wallhoff, F., Ablassmeier, M., Poitschke, T., Rigoll, G. *Static and dynamic*

- hand-gesture recognition for augmented reality applications*. Proceedings of the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments, HCI'07, Beijing, China, Springer-Verlag, Pp. 728-737, 2007.
- Roomi, R., Priya, S. M. M., Jayalakshmi, H. *Hand Gesture Recognition for Human-Computer Interaction*. Journal of Computer Science 6 (9): 994-999, ISSN 1549-3636, 2010.
- Sánchez-Nielsen, E., Antón-Canalis, L., Hernández-Tejera, M. *Hand Gesture Recognition for Human-Machine Interaction*. Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972 WSCG'2004, February 2-6, Plzen, Czech Republic, 2003.
- Tutorial Haar Negatives: tutorialhaartraining.googlecode.com/svn/trunk/data/negatives, 2011
- Viola, P., Jones. M. *Rapid Object Detection using a Boosted Cascade of Simple Features*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01), Volume 1, Pp. 511-518, 2001.
- Von Hardenberg, C., Bérard, F. *Bare-hand human-computer interaction*. Proceedings of the 2001 workshop on Perceptive user interfaces PUI 01, ACM Press, 2001.
- Whitehill, J., Omlin, C. W. *Haar Features for FACS AU Recognition*. Proceeding of 7th International Conference on Automatic Face and Gesture Recognition, Pp. 97-101, 2006.
- Sánchez-Nielsen, E., Antón-Canalis, L., Hernández-Tejera, M. *Hand Gesture Recognition for Human-Machine Interaction*. Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972 WSCG'2004, February 2-6, Plzen, Czech Republic, 2003.
- Tutorial Haar Negatives: tutorialhaartraining.googlecode.com/svn/trunk/data/negatives, 2011
- Viola, P., Jones. M. *Rapid Object Detection using a Boosted Cascade of Simple Features*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01), Volume 1, Pp. 511-518, 2001.
- Von Hardenberg, C., Bérard, F. *Bare-hand human-computer interaction*. Proceedings of the 2001 workshop on Perceptive user interfaces PUI 01, ACM Press, 2001.
- Whitehill, J., Omlin, C. W. *Haar Features for FACS AU Recognition*. Proceeding of 7th International Conference on Automatic Face and Gesture Recognition, Pp. 97-101, 2006.