Folosirea gesticii mâinii în interacțiunea om-calculator

Mihaela Coneac

Facultatea de Informatică, Universitatea "Al. I. Cuza", Iași

General Berthelot, 16, 700483, România

mihaela.coneac@infoiasi.ro

Adrian Iftene

Facultatea de Informatică, Universitatea "Al. I. Cuza", Iași

General Berthelot, 16, 700483, România

adiftene@infoiasi.ro

REZUMAT

În acest articol se prezintă cum poate fi folosită gestica mâinii în interacțiunea om-calculator. Acest tip de interacțiune poate fi foarte util în cazul persoanelor, care întâmpină de obicei dificultăți, atunci când trebuie să folosească tastatura sau mouse-ul, pentru a putea utiliza calculatorul. Evaluarea pe care am realizat-o ne confirmă acest lucru, subiecții umani care au participat la experiment arătându-se încântați de faptul că pot cu ajutorul mâinii să dea mult mai ușor comenzi uzuale.

Cuvinte cheie

Gestica mâinii, interacțiune om-calculator.

Clasificare ACM

H5.2. Information interfaces, I5. Pattern recognition.

INTRODUCERE

Lucrarea de față se încadrează în încercările de a adapta comunicarea cu PC- ul la modalitățile noastre naturale de comunicare: vorbirea și limbajul corpului. Deja au apărut noi tipuri de dispozitive cu două camere video atașate, care ne permit preluarea de imagini 3D. Acestea ne permit accesul la meniuri speciale ale unei aplicații soft, care la rândul ei este conectată wireless la un televizor sau la o consolă de jocuri. În CAD sau în jocurile de acțiune 3D, controlarea calculatorului prin gestică ne permite să realizăm operații de selectare, de rotire sau de adăugare de adnotări 2D sau 3D [6]. Recunoașterea mâinii cu metode biometrice este utilizată în identificare și verificare, deoarece pe de o parte este foarte ușor de folosit, iar pe de altă parte oferă o foarte mare securitate [10].

Tehnicile folosite în identificarea gesturilor mâinii au la bază recunoașterea de șabloane (*pattern recognition*) [3, 4, 11, 12]. Acestea sunt combinate cu folosirea histogramelor pentru identificarea orientării, sau cu metode robuste care nu depind de schimbările de iluminare [4]. Alte abordări se bazează pe identificarea degetelor [7] sau a caracteristicilor geometrice [10]. Modelele mai complexe care au rezultate mai bune folosesc fie modele Gaussian [11], fie rețele neuronale [1], fie recunoaștere de caracteristici multi-culoare [2].

În capitolele următoare se prezintă modul în care cu ajutorul unei camere web am folosit mâna în procesul de comunicare om-calculator. Spre final se prezintă experimentele realizate cu ajutorul unor subiecți umani și concluziile la care am ajuns.

MÂNA UMANĂ

Mâna umană are o structură complexă formată din multe oase conectate prin articulații. Considerând faptul că încheietura mâinii are 4 grade de libertate, vom avea în total 27 grade de libertate pentru mâna. Din cauza gradului mare de libertate al mâinii, recunoașterea gesturilor acesteia reprezintă o problemă foarte complexă.

Două concepte foarte importante, de care trebuie să ținem cont atunci când încercăm să identificăm gestica umană, sunt legate de:

- Postura mâinii (în engleză *hand posture*): care se referă la poziționarea statică a mâinii în care locația și mișcarea nu sunt luate în considerare;
- Gestica mâinii (în engleză *hand gesture*): care este o secvență 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.

DETECȚIA, RECUNOAȘTEREA ȘI URMĂRIREA OBIECTELOR ÎN TIMP REAL

Domeniul *detecția și recunoașterea obiectelor* a făcut progrese semnificative în ultimii ani. Mulți algoritmi dezvoltați în această arie au fost referiți la recunoașterea și detecția feței datorită multitudinilor de aplicații în domenii precum securitate și supraveghere. Cu toate acestea, o detecție rapidă a feței umane a fost imposibil de realizat în timp real, până de curând.

În procesul de detecție și de recunoaștere a obiectelor trebuie să realizăm următoarele operații:

- *Urmărirea obiectelor*: localizarea dinamică a obiectelor prin determinarea poziției acestora în fiecare imagine cadru 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 feței unei anumite persoane).

Acuratețea în detecția și recunoașterea obiectelor este măsurată folosind caracteristicile: *rată de succes* (procentul obiectelor corect identificate) și *rată de eșec* (procentul imaginilor care nu aparțin clasei obiectului respectiv).

Caracteristici Haar

Viola şi Jones au realizat o abordare statistică pentru problema detectării feţelor umane, utilizând o bază de cunoştinţe variată (feţe ce aveau: rotaţii şi culori diferite, obiecte opţionale cum ar fi: barbă, ochelari etc.) [13]. În studiul lor, au creat un nou concept intitulat „imagine integrală" ce reprezintă extragerea foarte rapidă a caracteristicilor Haar dintr-o imagine cu o dimensiune variată. Algoritmul de învăţare propus de autori primeşte la intrare un set de imagini „pozitive", ce includ obiectul de interes, şi un set de imagini „negative", ce nu conţin obiectul respectiv. În timpul procesului de învăţare, sunt selectate caracteristici Haar distincte la fiecare etapă în scopul identificării imaginilor care conţin obiectul de interes. În cazul în care clasificatorul antrenat omite un obiect sau detectează un obiect fals, problema se poate rezolva prin adăugarea unor noi caracteristici Haar, astfel se permite corectarea mostrelor clasificate greşit. Algoritmul propus de Viola şi Jones este aproximativ de 15 ori mai rapid decât abordările anterioare.

O trăsătură Haar este descrisă de un şablon care include dreptunghiuri albe şi negre interconectate, dimensiunea acesteia, precum şi poziţia relativă faţă de originea ferestrei de căutare. Figura 1 introduce setul extins de caracteristici Haar propus de către Lienhart [8], set ce include: 4 caracteristici pentru muchii, 8 liniare, una diagonala şi 2 caracteristici de tip „centru înconjurat" (în engleză center-surround feature).

Valoarea unei caracteristici Haar este diferenţa dintre suma pixelilor din dreptunghiul negru şi suma pixelilor din dreptunghiul alb.

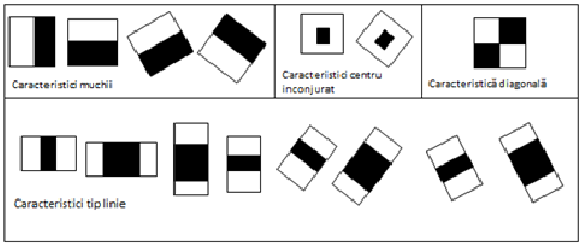


Figura 1. Setul extins de caracteristici Haar

$$f(x) = \mathbb{W}_{black} * \sum_{black-region} pixelValue - \mathbb{W}_{white} * \sum_{white-region} pixelValue$$

$\mathbb{W}_{black}$ şi $\mathbb{W}_{white}$ sunt valori care îndeplinesc condiţia compensaţiei [14]:

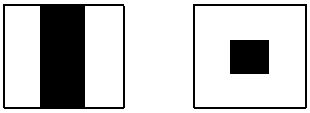
$$\mathbb{W}_{black} * black - region = \mathbb{W}_{white} * white - region$$

Conform condiţiei de compensaţie, pentru prima trăsătura Haar din Figura 2 (a), $\mathbb{W}_{black} = 2 * \mathbb{W}_{white}$ (deoarece exista o regiune neagră şi 2 albe), iar pentru cea de-a doua caracteristica din Figura 3 (b), $\mathbb{W}_{black} = 8 * \mathbb{W}_{white}$.

Dacă realizăm o convenţie pentru setarea $\mathbb{W}_{white} = 1$, atunci:

$$f(a) = 2 * \sum_{black-region} pixelValue - 1 * \sum_{white-region} pixelValue$$

$$f(b) = 8 * \sum_{black-region} pixelValue - 1 * \sum_{white-region} pixelValue$$



(a) *(b)*

Figura 2. Condiţia compensaţiei pentru diferite caracteristici Haar

Imagine Integrală (Summed Area Table-SAT)

Suma pixelilor dintr-o regiune I se calculează astfel:

$$Sum = \sum_{x=1}^{w} \sum_{y=1}^{h} I(x, y)$$

unde *w* = lăţimea, iar *h* = înălţimea regiunii *I*. Această operaţie se va realiza în timpul *O*(*n*), unde *n* = *w* * *h* (operaţie, care în funcţie de dimensiunile regiunii *I*, poate dura foarte mult).

Viola şi Jones prin introducerea termenului de „imagine integrală" au eliminat acest inconvenient al duratei mari de execuţie şi au putut realiza această operaţie în timp constant *O*(*1*).

Astfel „imaginea integrală" la locaţia *ii*(*x*, *y*) reprezintă suma pixelilor din dreptunghiul mărginit stânga sus de punctul (0, 0) şi dreapta jos de *ii*(*x*, *y*) [13].

$$ii(x, y) = \sum_{x' \le x, y' \le y} ii(x', y')$$

Conform definiţiei imaginii integrale, suma nivelului de gri din zona „D" (din Figura 3) se poate calcula astfel: $R_1 + R_4 - R_2 - R_3$, deoarece

$R_1 + R_4 - R_2 - R_3$ = A + (A + B + C + D) - (A + B) - (A + C) = D.

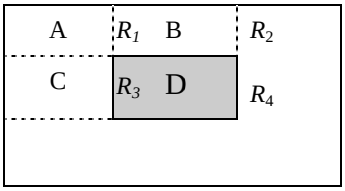


Figura 3. Conceptul „imagine integrală"

Extindere SAT – RSAT (Rotated Summed Area Table)

Lienhart în [8] a introdus conceptul de RSAT („Rotated Summed Area Table") pentru trăsăturile Haar ce conţin dreptunghiuri rotite cu 45^0. RSAT reprezintă suma pixelilor din dreptunghiul rotit având colţul din dreapta jos în punctul *ii*(*x*, *y*) şi extins până la marginile imaginii. Figura 4 ilustrează acest concept.

$$ii(x, y) = \sum_{y' \le y, y' \le y - |x - x'|} ii(x', y')$$

Figura 4. Conceptul „Rotated Summed Area Table (RSAT)" [8]

În figura 5 avem prezentat modalitatea în care detectăm o mână folosind sub-ferestre ce conțin caracteristici Haar.

Figura 5. Detectarea mâinii cu o sub-fereastră ce conține o caracteristică Haar

PROBLEMA CLASIFICĂRII. BOOSTING

În acest capitol se prezintă modul în care problema clasificării ne poate ajuta să decidem dacă într-o imagine avem sau nu o mână.

Problema clasificării: Fie X o mulțime de imagini, iar Y o mulțime cu valori $\{\pm 1\}$. Trebuie să găsim un clasificator: f: $X \rightarrow Y$ în care dându-se un element $X_i \in \boldsymbol{X}$ să prezică eticheta acestuia Y_i (dacă imaginea conține sau nu mâini în cazul nostru). O mulțime de clasificatori care prezice etichetele Y_i mai bine decât o alegere oarecare (mai bine de 50%) se numește o mulțime de clasificatori slabi.

Pentru detectarea obiectului de interes (mâna în cazul nostru), imaginea este scanată cu ajutorul unei sub-ferestre cu o anumită caracteristică Haar. Pe baza fiecărei caracteristici f_i, se definește un clasificator corespunzător $h_j(x)$ astfel:

$$h_j(x) = \begin{cases} 1, & dacă\ p_j f_j(x) < p_j \theta_j \\ -1, & alfel \end{cases}$$

unde x este sub-fereastra, θ este pragul, iar p_j indică direcția semnului de inegalitate.

Pentru a îmbunătăți acuratețea algoritmului nostru de clasificare de la o rulare la alta am folosit algoritmul de învățare AdaBoost. Acest algoritm folosește tehnica de „Boosting" (un algoritm de învățare bazat pe o serie de clasificatori slabi [5]).

AdaBoost folosește un set de antrenare (X_1,Y_1), …, (X_n,Y_n) cu ponderile uniforme w_i, unde X_i reprezintă imaginea, iar $Y_i \in \{1,-1\}$ indică tipul acesteia (pozitiv sau negativ). Acesta iterează peste datele de intrare un număr de T runde. În fiecare rundă, pentru fiecărei caracteristică f_i i se definește un clasificator $h_j(X)$.

În procesul de îmbunătățire iterativă se alege clasificatorul $h_j(X)$ cu cea mai mică eroare de clasificare a ponderii din mulțimea de clasificatori slabi ai caracteristicii f_i. La sfârșitul fiecărei runde ponderile instanțelor de antrenare clasate greșit sunt incrementate, astfel ca în următoarea rundă procesul de îmbunătățire să se focalizeze asupra acestora. Clasificatorul final $H(x)$ este compus dintr-o combinație liniară a clasificatorilor selectați în cadrul fiecărei runde. În practică, implementarea acestei cascade de clasificatori are rolul de a crește performanța.

Sub-fereastra pozitivă, pentru a putea fi detectată de cascada de antrenare, trebuie să treacă de fiecare etapă a acestui proces iterativ. Evident, un rezultat negativ în orice punct al cascadei conduce în final la respingerea acelei sub-ferestre. În prima rundă a procesului de antrenare, am stabilit pragul clasificatorului slab foarte jos astfel încât obiectele țintă să poată fi detectate în procent de 100%. Aplicarea acestei strategii se bazează pe faptul că majoritatea sub-ferestrelor sunt negative (reprezentând o parte din fundal) și este puțin probabil ca o instanță pozitivă să parcurgă toate stagiile. Având aceasta strategie, cascada poate crește considerabil timpul de procesare întrucât inițial clasificatorul slab va încerca să respingă cât mai multe sub-ferestre negative, iar o parte mare din timp va fi folosit pentru clasificarea acelor imagini.

EXPERIMENTE

În experimentele pe care le-am realizat am testat trei posturi ale mâinii: postura palmă, două degete și postura pumn (Figura 6). Pentru aceste experimente am folosit o cameră web Microsoft HD6000, configurată să capteze 15 cadre pe secundă la rezoluția 320x240.

Figura 6. Posturile palmă, două degete și pumn

Am colectat imagini pozitive (ce conțin o postură) de la două persoane. Pentru fiecare postură au fost realizate 750 de imagini, cu diferite dimensiuni și condiții de iluminare. Prin folosirea unor funcții oferite de biblioteca OPENCV (http://opencv.willowgarage.com/wiki/) s-au realizat 7000 de variații de imagini pozitive cu diferite rotații în spațiul tridimensional.Folosind(tutorialhaartraining.googlecode.com/svn/trunk/data/negatives) am obținut 3.000 de imagini care nu conțin nici o postură.

Pentru obținerea rezultatelor în timp real am optat folosirea a trei fire de execuție menite să ruleze simultan câte un clasificator, astfel detecția posturilor va fi concurentă.

S-a realizat o aplicației pentru browser-ul *Internet Explorer* în scopul demonstrării eficacității sistemului nostru de recunoaștere a posturilor. 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 (defilarea ferestrei) – prin mișcarea verticală a pumnului;
- ZOOM (mărirea sau micșorarea) – prin mișcarea pumnilor din interior/exterior spre exterior/interior;
- CLICK – postura „două degete" urmată de „pumn";
- TAB (selectarea unui link) – mișcarea două degete.

În antrenarea clasificatorilor au fost setate opțiunile: rata minimă de succes 96.9%, rata maximă de esec 50% și 20 de etape.

EVALUARE

Pentru a evalua performanța clasificatorilor obținuți s-au realizat câte 100 de imagini de test. Tabelul 1 prezintă performanțele cascadelor de clasificatori: pentru postura palmă rata de succes a fost de 90%, postura două degete 92% ,iar pentru pumn de 97%.

Tabelul 1. Performanța clasificatorilor antrenați

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

Prin analiza rezultatelor de detecție am remarcat faptul că rotațiile excesive provoacă omiterea unor imagini pozitive. 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. Această inconvenientă poate fi rezolvată prin ajustarea pragului pentru sub-fereastra de scanare. Am testat performanța în timp real cu intrări de la camera-web și nu s-a înregistrat nici o latență în detecția sau urmărirea posturilor. Clasificatorii antrenați au un grad de robustețe ridicat în ceea ce privește variația de lumină sau rotația ± 15° în plan.

Experimentele pe care le-am realizat cu ajutorul a 5 persoane, ne-au demonstrat 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 dezabilități sau al persoanelor care folosesc prima data 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.

CONCLUZII

Lucrarea de față ne prezintă modul în care gestica mâinii poate fi folosită cu rezultate promițătoare în interacțiunea om-calculator. Chiar dacă suntem la început evaluarea pe care am realizat-o ne-a demonstrat interesul sporit al celor care au participat la experiment în a folosi un alt mod de comunicare.

Problemele identificate ne-au dus la concluzia că pe viitor ar trebui realizate aplicații specifice acestui mod de interacțiune, și care să fie capabile să ofere accesul la mult mai multe opțiuni ale aplicațiilor doar prin folosirea câtorva gesturi ale mâinii.

MULȚUMIRI

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

1. Ahmeda, S. M. H., Alexanderb, T. C., Georgios C. Anagnostopoulos, B. (2008). Real-time, Static and Dynamic Hand Gesture Recognition for Human-Computer Interaction. TR. 2008.
2. Bretzner, L., Laptev, I., Lindeberg, T. (2002). 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.
3. Chakraborty, P., Sarawgi, P., Mehrotra, A., Agarwal, G., Pradhan, R. (2008). Hand Gesture Recognition: A Comparative Study. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008, 19-21 March, 2008, Hong Kong.
4. Freeman, W. T., Roth, M. (1994). Orientation Histograms for Hand Gesture Recognition. IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition, Zurich, June, 1995.
5. Freund, Y., Schapire, R. E. (1999). A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, (14): 5, pp. 771--780.
6. Gorgan, D., Stefanut, T., Veres, M., Gabos, I. (2008). Tehnici de adnotare grafica în 3D în aplicatiile de e-learning interactive. Revista Româna de Interactiune Om-Calculator (1): 1, 2008 ISSN 1843-4460.
7. Hardenberg, C., Bérard, F. (2001). Bare-Hand Human-Computer Interaction. Proceedings of the ACM Workshop on Perceptive User Interfaces, Orlando, Florida, USA, Nov. 15-16 2001.
8. Lienhart, J. Maydt, (2002). An extended set of Haar-like features for rapid object detection. 2002.
9. Napier, J. (1980). Hands. New York: Panthon Books.
10. Öden, C., Erçil, A., Yildiz, V.T., Kirmizita, H., Büke, B. (2001). 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 2001. ISBN: 3-540-42216-1.
11. Roomi, R., Priya, S. M. M., Jayalakshmi, H. (2010). Hand Gesture Recognition for Human-Computer Interaction. Journal of Computer Science 6 (9): pp. 994--999, 2010 ISSN 1549-3636.
12. Sánchez-Nielsen, E., Antón-Canalís, L., Hernández-Tejera, M. (2003). Hand Gesture Recognition for Human-Machine Interaction. Journal of WSCG (12), pp. 1-3, ISSN 1213-6972 WSCG’2004, February 2-6, 2003, Plzen, Czech Republic.
13. Viola, P., Jones. M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. 2001.
14. Whitehill, J., Omlin, C.W. (2006). Haar Features for FACS AU Recognition. Procceding of 7th Internation Conference on Automatic Face and Gesture Recognition. pp. 97--101, 2006.