

Comportament Adaptiv pentru Interfețe Utilizator Avansate. Comunicarea prin Intermediul Gesturilor

Eng. Drd. Delia Mitrea*, Prof. Dr. Eng Dorian Gorgan**

Universitatea Tehnică Cluj-Napoca

E-mail: * delia.mitrea@cs.utcluj.ro, ** dorian.gorgan@cs.utcluj.ro

Rezumat. Lucrarea cuprinde un studiu referitor la interfețele utilizator adaptive, sensibile la necesitățile și preferințele utilizatorului, insistându-se asupra posibilității de comunicare prin mijloace naturale, anume prin gesturi. Sunt prezentate atât metode specifice de interpretare a gesturilor, din domeniul recunoașterii de forme și al procesării de imagini, cât și o posibilitate de implementare prin intermediul modelului obiectelor active (AOM). Sunt discutate și aspecte legate de performanța, urmărindu-se ca viteza de execuție să fie minimă.

Cuvinte cheie: gesturi, interacțiune naturală, interfețe adaptive.

1. INTRODUCERE

Lucrarea de față are ca obiectiv studiul posibilității de realizare a unor interfețe inteligente, adaptabile la necesitățile și preferințele utilizatorului, bazate pe o comunicare rapidă cu acesta, prin mijloace naturale - și anume prin gesturi. Se va realiza, în acest sens, studiul unor metode destinate învățării și recunoașterii gesturilor umane, general valabile, dintre care se va alege o metodă adecvată cazului concret tratat. Se va face o mapare a celor studiate în domeniul Obiectelor Active (Active Object Models -AOM), căutându-se o modelare adecvată în vederea implementării. Se va aborda, de asemenea, problema performanței, urmărindu-se realizarea unui sistem care să funcționeze în timp real.

Problema concretă care se dorește a fi abordată este următoarea: dată fiind o aplicație - cum ar fi cea de redare a înregistrărilor multimedia (media player) - se dorește comandarea acesteia de către utilizator prin intermediul gesturilor. Presupunându-se că aplicația beneficiază și de o interfață clasică, învățarea semnificației gesturilor, specifice fiecărui utilizator în parte, se va face prin asocierea formei vizuale (senzoriale) a gestului cu acțiunea care ar fi declanșată prin acționarea elementului din interfața clasică. (ex: utilizatorul face un gest, apoi apasă pe butonul dorit).

Sistemul construit va avea următoarele faze semnificative :

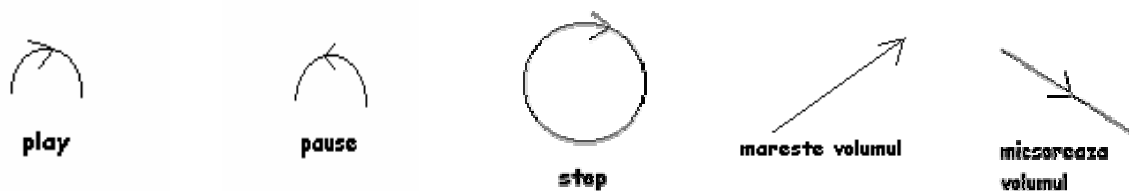


Figura 1. Trajectoriile unor gesturi făcute cu mâna

- **faza de identificare a utilizatorului:** utilizatorul va fi identificat, conform modului obișnuit, după nume și parolă
- **faza de învățare**, care va consta din memorarea într-o bază de date, a asociațiilor (identificator utilizator, traiectorie gest, denumire acțiune corespunzătoare). Această fază se va declanșa prin acționarea de către utilizator a unui buton "**Invață**". Utilizatorul apasă pe butonul "**Invață**", face gestul dorit și, concomitent, apasă pe butonul asociat acelei acțiuni care se dorește a fi reprezentată prin gestul respectiv.
- **faza de identificare gest**, care va consta din identificarea gestului făcut de utilizator în fața camerei și declanșarea acțiunii corespunzătoare acestui gest; această fază se va declanșa în absența acționării butonului "**Invață**"

Atât *faza de învățare*, cât și cea de *identificare gest*, vor fi precedate de o etapă de *preluare a datelor de intrare* sub forma unei secvențe de imagini și *prelucrare a acestora*, în vederea obținerii unor trăsături ce vor fi folosite în continuare, în procesul de învățare/recunoaștere. **Rezultatul** va fi fie învățarea unui nou gest - memorarea în baza de date a coordonatelor traiectoriei și a denumirii asociate- și afișarea pe ecran a traiectoriei descrise (pentru ca utilizatorul să poată repeta gestul în mod identic), fie declanșarea unei acțiuni (play, stop, mărește/micșorează volumul audio), ca urmare a gestului făcut de utilizator în fața camerei.

2. CONSIDERAȚII TEORETICE

2.1 Definire concepte

Gest

Se va înțelege prin **gest** orice mișcare a utilizatorului (abatere a poziției corpului, în particular a mâinii, de la poziția normală) căreia i se atribuie o anumită semnificație (comanda care se dorește a fi dată sistemului). În exemplul tratat, atenția se va focaliza asupra mișcării mâinii, considerată a fi plană.

Traietorie asociată unui gest

Prin *traietorie asociată unui gest* vom înțelege o succesiune de perechi de coordonate (x,y) reprezentând pozițiile succesive ale centrului mâinii utilizatorului, poziții identificate în imaginile consecutive ale secvenței de imagini preluate în timpul efectuării gestului de către utilizator.

Interfețe adaptive

Interfețele adaptive ("user-tailored environments") sunt acele interfețe ale aplicațiilor ce sunt capabile să-și modifice modul de prezentare, respectiv funcționalitățile în conformitate cu preferințele exprimate de utilizator, deci dotate cu capacitate de învățare.

Interacțiune naturală ("Natural interaction")

Conceptul de "*natural interaction*" (*interacțiune naturală*) se referă la acele interfețe adaptive care facilitează interacțiunea cu utilizatorul prin mijloace naturale, precum gesturile sau vorbirea.

2.2 Conceptul de interacțiune naturală. Privire generală

Noile tehnologii de realizare a interfețelor utilizator tind să elimine convențiile tradiționale ale comunicării om-calculator și să faciliteze o interacțiune naturală, intuitivă, specific umană. Scopul este acela de a atrage un număr cât mai mare de utilizatori, aceștia neavând nevoie de cunoștințe suplimentare pentru utilizarea calculatorului, precum și de a integra calculatorul în viața curentă, de a construi dispozitive integrate în medii inteligente (ex: cameră inteligentă - "intelligent room"). Aceste interfețe inteligente vor trebui să recunoască utilizatorul și să se adapteze cerințelor acestuia. Mijloacele de comunicare trebuie să fie cele dorite, preferându-se cele non- WIMP (Windows, Icons, Menus, Pointers), și anume cele bazate pe vorbire, gesturi, expresie facială, scriere manuală [5].

Conceptul de *interacțiune multimodală*, foarte vehiculat în contextul interacțiunii naturale (natural interaction) se referă la folosirea simultană a mai multor modalități de comunicare cu utilizatorul - atât dintre cele tradiționale, cât și dintre cele naturale (de ex: atât vorbire, cât și gesturi, cât și expresie facială) [5], [6].

Dincolo de modul clasic de interacțiune om-calculator, există situații în care calculatorul joacă rolul unui intermediar în comunicarea interumană [5], cum ar fi ,

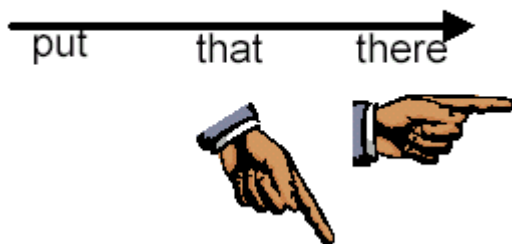


Figura 2. Exemplu de interacțiune multimodală: "Pune asta acolo" - Gesturi și Vorbire

de exemplu, comunicarea la distanță dintre doi subiecți umani. Se dorește, în acest context, realizarea unui *calculator "transparent", "invizibil"* (*"The Transparent, Invisible Computer"*). Acest lucru se face tot prin intermediul interfețelor adaptive, bazate pe mijloace naturale de comunicare. Noul calculator "invizibil", pe lângă calitatea de a fi conștient de context și adaptabil la acesta, va trebui să fie mai degrabă proactiv decât condus de comenzile utilizatorului; calculatorul va trebui să se adapteze la om, nu invers. Cele mai reprezentative aplicații care implementează elemente ale conceptului "interacțiune naturală" sunt cele de tipul *cameră inteligentă* ("smart room"), camere care se adaptează nevoilor utilizatorului dispunând de facilități precum reglarea automată a temperaturii sau luminii, sau de posibilitatea de a comanda prin gesturi funcționarea dispozitivelor din cameră.[6].

2.3 Metode adecvate învățării și interpretării gesturilor utilizator

Există două modalități de a capta gesturile utilizator:

- *Prin senzori*
- *Prin intermediul camerei video*

Camera video limitează zona de mișcare a utilizatorului, motiv din care, uneori, se preferă folosirea tehnicilor bazate pe senzori.

Cele mai răspândite metode destinate învățării și recunoașterii gesturilor se referă atât la situațiile de preluare a gesturilor prin intermediul senzorilor sub formă de semnale, cât și la cele în care gesturile se preiau prin intermediul camerei video.

Metodele cele mai des utilizate sunt cele bazate pe logica Fuzzy [2], cele bazate pe lanțuri Markov ascunse [7], respectiv cele bazate pe potrivire de contur [4].

2.4 Metoda propusă

În [4] autorii prezintă o metodă de recunoaștere a gesturilor complexe, bazate pe analiza mișcărilor întregului corp (a siluetelor). Recunoașterea se face prin construirea a două semnături unidimensionale - pentru valorile celor două coordonate polare ρ și θ , pentru fiecare din imaginile dintr-o secvență de poziții ce caracterizează un anumit gest. Din aceste semnături se construiește o nouă imagine caracteristică mișcării respective. Pentru a reduce numărul imaginilor de acest tip obținute din setul de antrenament, se folosește metoda Karhunen-Loeve (PCA).

O metodă mai simplă, rapidă, adecvată multor situații practice, este cea de detecție a centrului mâinii, construirea traiectoriei descrise de acest punct pentru un anumit gest sub forma unui contur, apoi recunoașterea gesturilor prin potrivire de contururi. În cazul particular luat în considerare, bazat pe recunoașterea gesturilor manuale, se va opta pentru o metodă bazată pe potrivire de contur care să țină cont și de modul de parcurgere a traiectoriei asociate gestului, nu doar de forma conturului descris. Pentru exemplificare, se va alege metoda bazată pe calculul distanței euclidiene între semnături, semnăturile fiind formate din coordonatele x_i , y_i ale traiectoriei. Pentru reducerea spațiului trăsăturilor de la doi vectori, X și Y, la

un singur vector, se poate utiliza analiza componentelor principale (Principal Component Analysis-PCA). Pentru că discriminarea doar în funcție de distanță poate fi, uneori, insuficientă, se va lua în considerare și **frecvența** cu care utilizatorul efectuează un anumit gest.

3. PARTICULARIZARE LA MODELELE CU OBIECTE ACTIVE

Se dorește construirea unui model care să permită învățarea, pentru fiecare utilizator în parte, a gesturilor specifice diferitelor acțiuni posibile pentru aplicația media-player. Sistemul este prevăzut *cu două opțiuni*: una de *învățare*, atunci când se dorește învățarea gesturilor utilizator - fapt care poate fi specificat prin intermediul unui buton, "learn" (învață), respectiv una de *recunoaștere*, atunci când se dorește identificarea semnificației gestului și execuția acțiunii corespunzătoare.

Este nevoie de comunicarea cu o bază de date, pentru stocarea modelului fiecărui utilizator. Sunt necesare trei tipuri de acțiuni: **preluarea datelor de intrare**-preluarea secvenței de imagini și extragerea trăsăturilor care ne interesează, în cazul nostru coordonatele (x,y) ale traiectoriei descrise manual; **construirea modelului fiecărui utilizator** - identificarea utilizatorului; învățarea gesturilor pentru fiecare utilizator în parte, însemnând asocierea vectorului de trăsături cu numele acțiunii respective; de asemenea, recunoașterea (identificarea) gestului în funcție de vectorul de trăsături; **controlul aplicației** - declanșarea acțiunii corespunzătoare gestului identificat .

3.1 Definirea obiectelor active și a relațiilor dintre acestea

Modelarea interfeței adaptive care învață gesturile utilizatorului, se poate realiza prin intermediul a trei asemenea **obiecte active cu rol de agent**, care să comunice cu dispozitivul de intrare pentru imagini (cameră), prin intermediul unei interfețe, sau să acționeze asupra unor alte obiecte, cum sunt baza de date, respectiv aplicația media-player :

- un obiect **InputProcessor**, care să preia datele de intrare și să interpreteze aceste date, furnizând un vector de trăsături (coordonatele (X,Y) ale traiectoriei)
- un obiect care să controleze utilizatorul, să-l identifice, să construiască un model adecvat acestuia în faza de învățare, respectiv să returneze modelul adecvat, deja existent (**UserSupervisor**)
- un obiect **ApplicationSupervisor**, care să declanșeze acțiunea corespunzătoare gestului identificat, predând controlul aplicației

Obiectul InputProcessor

Rolul obiectului **InputProcessor** este acela de preluare a datelor de input (o secvență de imagini), prelucrarea acestora și extragerea trăsăturilor - a traiectoriei descrise de centrul mâinii, sub forma unei secvențe de coordonate. Comportamentul obiectului, dat de traiectoria asociată, este descris de următorii pași (**metoda**

GestureFeatures): (1) *preluarea secvenței de imagini*; (2) *deteția mâinii* în fiecare imagine și *localizarea centrului* acesteia; (3) *memorarea secvenței de coordonate* (X,Y) corespunzătoare traiectoriei descrise de mână.

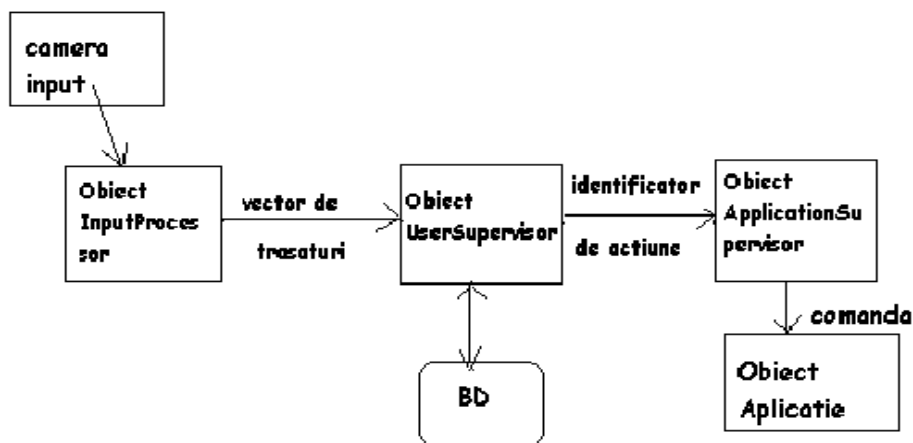


Figura 3. Diagrama obiectelor active

Obiectul UserSupervisor

Identificarea utilizatorului se va face, în cazul de fata, după numele de utilizator și parolă.

Învățarea se va realiza în felul următor: se va înscrie în baza de date asociația formată din identificatorul de utilizator, coordonatele (X,Y) ale traiectoriei corespunzătoare gestului, respectiv numele asociat gestului, dedus din acțiunea utilizatorului asupra elementelor interfeței clasice.

Recunoașterea se va realiza în felul următor: se extrag din baza de date vectorii de coordonate corespunzatori gesturilor învățate, împreună cu denumirea asociată, în structura `vect_trăsături`; pentru fiecare gest din structura `vect_trăsături`: se calculează distanța dintre vectorul de coordonate ale gestului deja învățat și vectorul de coordonate preluat de la intrare; dacă distanța este minimă, se instanțiază atributul `denumire_gest` cu numele corespunzător din structura `vect_trăsături`; - dacă există mai multe gesturi pentru care distanța este minimă, se diferențiază în funcție de frecvența de realizare a unui gest.

Obiectul ApplicationSupervisor

Obiectul *ApplicationSupervisor* conține, la rândul său, un *atribut id_ acțiune*. În funcție de valoarea acestui atribut, *obiectul ApplicationSupervisor* declanșează execuția acțiunii corespunzătoare, predând controlul aplicației.

Comunicarea între obiecte

Obiectul *InputProcessor* trebuie să comunice obiectului *ApplicationSupervisor* trăsăturile extrase (vectorii de coordonate X și Y). Într-un model obișnuit, orientat pe obiecte, acest lucru se realizează prin *delegare*.

Structura bazei de date

Sistemul trebuie să comunice cu o *bază de date* relațională sau cu un fișier aflat pe disc care să conțină următoarele corespondențe: 1. *Identificator utilizator, Nume utilizator, Parolă (date de identificare)*; 2. *Identificator utilizator, Denumire acțiune, Coordonate vizuale ale gestului -trăsături, Frecvența gestului*

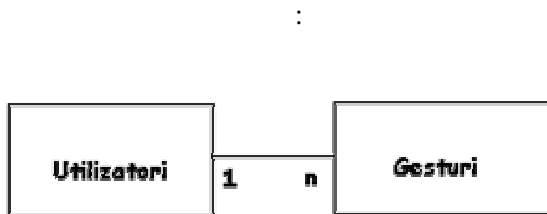


Figura 4. Structura bazei de date

4. PARTICULARIZARE LA MODELUL AOM (ACTIVE OBJECT MODEL, [1])

Interfața adaptivă, capabilă de a învăța și identifica gesturi pentru fiecare dintre utilizatori, se va implementa în AOM [1] prin intermediul celor trei obiecte active, descrise în paragraful precedent. Comunicarea între obiecte nu se va face prin delegare, ci prin mecanismul de *bounding* specific AOM, ceea ce duce la eliminarea, din cadrul obiectului *UserSupervisor* a referinței la obiectul *InputProcessor*. Atributul "trăsături", al obiectului *ApplicationSupervisor*, va fi legat de atributul "trăsături" al obiectului *UserSupervisor* printr-un mecanism de tip *bound-from* [1]. De asemenea, atributul *id_ acțiune* al obiectului *UserSupervisor* va fi legat de atributul *id_ acțiune* al obiectului *ApplicationSupervisor* printr-un mecanism de tip *bound to* [1]. Gestul va fi reprezentat prin traiectoria corespunzătoare acestuia - care se va implementa prin intermediul unui vector conținând perechi de coordonate (X,Y), respectiv prin numele acțiunii asociate gestului, care se va implementa sub forma unui string. Modelarea vectorilor (a vectorilor de trăsături și a imaginii) se va face folosind primul model propus în

acest sens în [1]. Lucrându-se cu date de dimensiuni necunoscute a priori, metoda optimă este aceea de a folosi o alocare dinamică a acestor vectori. Imaginea se va reprezenta ca o matrice bidimensională de valori de intensitate (culoare), iar vectorii de trăsături se vor reprezenta prin tablouri unidimensionale, având ca elemente perechi de coordonate (X,Y). Pentru că vectorii sunt mari consumatori de resurse în modelul AOM, se va expune, în paragraful "*Considerații tehnice și tehnologice*", o modalitate de a lucra doar cu date scalare. Referința la baza de date ar corespunde unui obiect activ distinct, capabil de a se conecta la baza de date existentă pe disc și de a extrage datele din aceasta, instanțiind propriile sale attribute. Totuși, pentru că un obiect activ este foarte costisitor din punctul de vedere al memoriei, se preferă reprezentarea bazei de date în AOM prin intermediul unui comportament.

```

Behaviour BuserS{
Parameter username{string;};
Parameter passw{string;};
Parameter gesture_features{vector;};
Trajectory TuserS{
    Trjposition P1{
        Type PRECOND
        Expression{username!=NULL && passw!=NULL}
        Rule R1{
            Condition C1 { bhv_identificare(username,
                passw)== FALSE }
            Action A1 { user_id=0;
                Jump trjposition P5 };
        };
        . . . . .
    };};
    
```

Figura 5. Descrierea traiectoriei obiectului *UserSupervisor* - faza de identificare a utilizatorului (AGML)

Comportamentul general al sistemului va fi descris prin intermediul comportamentului individual al fiecărui obiect în parte și de comunicarea dintre aceste obiecte [1].

Metodele mai complexe ale obiectelor vor avea un comportament, deci o traiectorie proprie. Implementarea se va face în limbajul asociat modelului AOM, AGML (Agent Modelling Language).

4.1 Considerații tehnologice și tehnice de realizare. Performanță

Pentru a evita lucrul cu vectori, care este costisitor în AOM, se va încerca eliminarea vectorilor de trăsături și lucrul cu valori scalare.

În loc de atributele vector_trăsături, ale obiectelor *InputProcessor* și *UserSupervisor*, se vor folosi două valori scalare (primitive) de tip întreg, x și y , reprezentând coordonatele unui punct de pe traiectoria care reprezintă gestul. Acest lucru implică funcționarea alternativă a celor două obiecte – pe două thread-uri separate, cu necesitatea sincronizării între acestea: obiectul *UserSupervisor* prelucrează perechea (x,y) furnizată de către obiectul *InputProcessor*, așteptând apoi furnizarea unei noi perechi. Funcționarea s-ar putea face în *manieră pipeline* (extragerea coordonatelor centrului pentru o poziție ulterioară a mâinii – de către obiectul *InputProcessor*, să coincidă, în timp, cu prelucrarea coordonatelor pentru poziția precedentă, de către obiectul *ApplicationSupervisor*). Pentru că prima operație este mult mai costisitoare ca timp de prelucrare decât a doua, este necesară implementarea unei modalități de sincronizare între obiecte.

Pentru a reduce spațiul de memorie externă (disc) ocupat de baza de date și pentru a mări viteza de lucru, soluția ar fi evitarea comunicării cu discul, deci informațiile legate de utilizatori să se păstreze în cod - fiind deci necesară implementarea de cod dinamic.

5. CONSIDERAȚII CRITICE

Sistemul realizează urmărirea și recunoașterea gesturilor efectuate cu mouse-ul de către utilizator în fereastra *Internet Explorer*: exemplu - dragging (tragere) în jos (spre partea inferioară a ecranului) pe un link (referință) înseamnă cerere de deschidere a paginii referite într-o fereastră nouă. Aplicația este dezvoltată în mediul Visual Studio, în limbajul Visual C++. Aplicația Spy, din pachetul Visual Studio s-a dovedit a fi deosebit de utilă pentru identificarea ferestrei HTML a browser-ului unde se efectuează urmărirea mișcării mouse-ului. Remarcăm diferențe față de aplicația prezentată în această lucrare (urmărirea și recunoașterea gesturilor manuale) - atât în ceea ce privește mijloacele de interacțiune cu utilizatorul, cât și în ceea ce privește implementarea obiectuală. Astfel, comunicarea gesturilor prin intermediul mouse-ului poate duce la o creștere considerabilă a vitezei sistemului, fiind eliminată partea de preluare a imaginilor de la interfața cu camera digitală, respectiv partea de prelucrare a acestora în vederea detecției centrului mâinii, scăzând, deci, în acest mod, complexitatea computațională. Observăm, de asemenea, că modul de funcționare al aplicației bazate pe mouse nu este în spiritul interacțiunii naturale ("natural interaction").

Sistemul MouseGestures for InternetExplorer presupune participarea dispozitivului mouse la realizarea gesturilor, fapt care are următoarea implicație: calitatea fizică a dispozitivului limitează capacitatea de mișcare a cursorului pe ecran și, astfel, puterea de expresie a utilizatorului, forma descrisă a gestului cu ajutorul mouse-ului putând diferi de forma dorită de utilizator.

6. CONCLUZII

Am descris în lucrarea de față modul de realizare a unei interfețe inteligente, bazată pe comunicarea cu utilizatorul prin intermediul gesturilor identificate din secvențe de imagini preluate cu ajutorul unei camere video, atașate sistemului. O modalitate, aproape imediată, de îmbunătățire, este introducerea, în faza de învățare, a posibilității de a intui noi gesturi similare prin predicție (inversarea ordinii perechilor de coordonate ale traiectoriei pentru gestul cu semnificație antonimică). Posibilitățile de dezvoltare se referă atât la îmbunătățirea tehnologiei de prelucrare în sensul creșterii vitezei, cât și la îmbunătățirea, în continuare, a mijloacelor de comunicare cu utilizatorul: identificarea după figură (față), comunicarea prin mimică, respectiv analiza expresiei faciale și modificarea funcționalităților sau a modului de prezentare al aplicației în funcție de aceasta.

Referințe

1. Gorgan, D. *Note de curs - Sisteme Interactive*, Universitatea Tehnică Cluj-Napoca, 2004, <http://bavaria.utcluj.ro/~gorgan>
2. Holden, E., Roy, G., Owens, R. Department of Computer Science, The University of Western Australia, *Adaptive Classification of Hand Movement*, 2000
<http://www.cs.uwa.edu.au/~robyn/Papers/ejlcnn.ps.gz>
3. Hare, R. *Mouse gestures for Internet Explorer*,
<http://www.codeproject.com/atl/MouseGestures.asp>
4. Peixoto, P., Goncalves, J., Araujo, H., *Real Time Gesture Recognition System based on Contour Signatures*, <http://labvis.isr.uc.pt/pub-2002/icpr2002-px.pdf>
5. Plomp, J., Ailisto, H. *Natural Interaction*,
http://www.vtt.fi/ict/publications/plomp_et_ailisto_030821.pdf
6. Plomp, J. *Smart Rooms, Intelligent Environments*, Seminar SS ,2003
<http://isl.ira.uka.de/~stiefel/Seminar2003/7.5.IntroductionIntelligentRooms.Stiefelhagen.ppt>
7. Wilson, A. The Media Laboratory, Massachusetts Institute of Technology, *Learning Visual Behaviour for Gesture Analysis*, 1995
<http://citeseer.ist.psu.edu/cache/papers/cs/405/ftp://zszszwhitechapel.media.mit.edu/zpubzsztech-reportszszTR-337.pdf/wilson95learning.pdf>