Algoritmi de recunoaştere a gesturilor

Ionuț-Alexandru ZAIȚI
Universitatea "Ştefan cel Mare" Suceava
str.Universității nr.13, RO-720229 Suceava, iozaiti@stud.usv.ro

***Abstract*—This paper presents algorithms for static and dynamic hand gestures recognition following at the same time their recording and creating a library of gestures. The equipment used for developing the hand gesture recognition techniques consists in a 5DT Ultra Glove with 14 sensors.**

***Index terms*—dynamic time warping, hand gesture, nearest neighbor, pattern recognition, sensor glove**

I. INTRODUCERE

În Dicționarul explicativ al limbii române gestul este definit ca fiind o mișcare a mâinii, a capului etc. care exprimă o idee, un sentiment, o intenție, înlocuind uneori vorbele sau dând mai multă expresivitate vorbirii, o faptă sau purtare dictată de un anumit scop, de anumite interese, având o anumită semnificație etc[1]. În lucrarea de față definiția gestului este extinsă la orice mișcare de interacțiune cu mediul înconjurător, de manipulare a acestuia, orice mișcare destinată preluării de date din mediul înconjurător. De asemenea se va mai lua în considerare o clasificare abstractă a gesturilor sau posturilor mâinii ce nu ține cont de proprietățile obiectului țintă, de scopul acțiunii sau constrângerile acesteia:

- gesturi(posturi) simple – presupun menținerea mâinii într-o poziție statică pentru o anumită perioadă de timp
- gesturi(posturi) complexe – presupun realizarea unei mișcări a mâinii și a degetelor, fiind de fapt o succesiune de gesturi simple

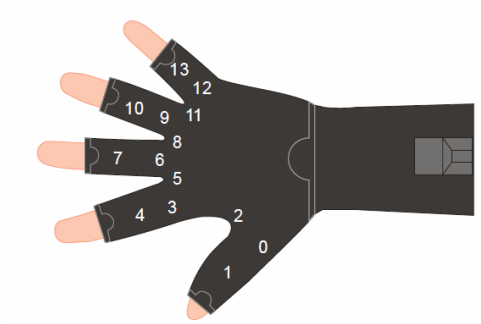

Fig. 1. Mănușa 5DT Ultra – Maparea senzorilor (Sursă: 5DT Data Glove Ultra Series User's Manual[7]).

Lucrând cu o mănușă 5DT Ultra cu 14 senzori poziționați pe monturi și încheieturile degetelor aceste posturi sunt reprezentate ca forme având 14 caracteristici cu valori cuprinse între 0 și 1, 0 însemnând că încheietura de pe care s-au cules datele a fost flexată la maxim iar 1 că încheietura respectivă a fost întinsă la maxim:

$f = (a_1,a_2,a_3,...,a_{14})$, $a_i \in [0,1]$

II. GESTURI SIMPLE

Problema recunoașterii gesturilor simple presupune preluarea succesivă a posturilor ce vin de la mănușa 5DT și clasificarea acestora în funcție de o bibliotecă de gesturi creată anterior. Ca măsură a diferenței dintre două gesturi simple am ales distanța euclidiană definită ca:

$f_1 = (a_1,a_2,a_3,...,a_{14})$, $a_i \in [0,1]$
$f_2 = (b_1,b_2,b_3,...,b_{14})$, $b_i \in [0,1]$

$$d(f1,f2) = \sqrt{\sum_{i=1}^{14}(a_i - b_i)^2}$$

Având la dispoziție 14 senzori care ne transmit valori variind între 0 și 1, valoarea maxima teoretică a distanței dintre două gesturi este de aproximativ 3.74($\sqrt{14}$) și valoarea minima teoretică este 0.

III. ÎNREGISTRAREA GESTURILOR SIMPLE

Deși în definiția considerată pentru un gest simplu se presupune că mâna rămâne într-o poziție statică, în practică acest lucru nu este posibil. Nu putem repeta aceleași gesturi în acelasi mod, la fiecare execuție vor apare variații oricât de minore ale tuturor celor 14 senzori, fapt de care trebuie să ținem cont atât în momentele în care construim biblioteca de gesturi cât și în momentul în care încercăm să le recunoaștem.

O soluție pentru înregistrarea unui gest simplu este să efectuăm acel gest simplu o perioadă de timp, gestul ce va fi memorat reprezentând de fapt media tuturor posturilor care au fost înregistrate. Motivul pentru care nu putem să alegem la întâmplare una din posturile prin care am trecut în timpul efectuării gestului respectiv este existența variațiilor menționate mai sus și posibilitatea ca postura aleasă să reprezinte o extremă a clasei din care face parte.

IV. RECUNOAȘTEREA GESTURILOR SIMPLE

Una dintre cele mai accesibile metode pe care o putem utiliza în recunoașterea gesturilor simple este căutarea celui mai apropiat vecin(„nearest neighbor search")[2]. Acesta clasifică o postură căutând în biblioteca disponibilă acel gest față de care este cea mai apropiată(distanța euclidiană dintre

cele două este cea mai mică). Există mai multe metode de căutare cum ar fi căutarea liniara, LSH(„locality sensitive hashing")[3], „branch and bound"[4], fişiere de aproximare vectorială [5] etc. Dintre acestea cea mai simplă este căutarea liniara, algoritmul având o complexitate O(N*p) unde N este numărul de gesturi din bibliotecă şi p este numărul de caracteristici(în cazul nostru 14, dat de numărul de senzori ai manuşii). Căutarea liniară compară în mod succesiv postura curentă cu toate gesturile din biblioteca noastră căutând soluția, soluție care poate fi prezentată de asemenea în mai multe variante, fie ca un singur gest, fie ca o mulțime de gesturi. Trecând prin toate gesturile disponibile, este cea mai costisitoare din punctul de vedere al timpului însă ne garantează în contextul unei logici corecte a aplicației noastre găsirea soluției optime.

În primul caz în care soluția va fi reprezentată de un singur gest acesta va fi gestul față de care distanța euclidiană este cea mai mica. În cel de-al doilea caz avem o altă variantă a căutării celui mai apropiat vecin, căutarea celor mai apropiați k vecini, în care soluția va fi compusă din k gesturi din bibliotecă față de care distanțele sunt cele mai mici, complexitatea algoritmului crescând în funcție de metoda aleasă pentru găsirea celor k vecini(algoritmul de sortare)[2].

În ambele cazuri putem avea de asemenea două variante. Prima variantă presupune să găsim de fiecare dată o soluție nevidă, cei mai apropiați vecini sunt luați întotdeauna indiferent de cât de mare este distanța minimă. Avantajul este că fiecare postură este clasificată şi nu trebuie tratată problema pragului de separare. Dezavantajul este că putem trage concluzia că două gesturi aparent diferite fac parte din aceeaşi clasă.

Cea de-a doua variantă este cea în care acceptăm să nu avem o soluție sau soluția să fie o mulțime vidă, ceea ce înseamnă că nu luăm în considerare acele gesturi față de care distanța nu este mai mică decât un prag stabilit anterior. Implicația este că se impune un pas important şi anume alegerea unui prag. Dacă alegem un prag prea mic rata de recunoaştere va scade dar va creşte varietatea gesturilor pe care le putem înregistra, în schimb dacă alegem un prag prea mare din nou două gesturi aparent diferite vor putea face parte din aceeaşi categorie dar rata de recunoaştere va creşte. Alegerea pragului va fi determinată în cele din urmă de scopul aplicației şi de nivelul de precizie dorit.

V. GESTURI COMPUSE

Cazul gesturilor statice este unul relativ simplu atât în partea de înregistrare cât şi la recunoaştere, la gesturile compuse însă situația se schimbă. Biblioteca noastră nu va conține gesturi simple ci colecții de gesturi simple ale căror ordine ne dau gestul compus ce trebuie recunoscut.

VI. ÎNREGISTRAREA ŞI RECUNOAŞTEREA GESTURILOR COMPUSE

Înregistrarea gesturilor compuse presupune execuția gestului țintă şi reținerea tuturor posturilor prin care trecem, în final prima postura din colecția obținută reprezentând gestul de start al gestului şi ultima postura gestul de stop.

După înregistrare, pentru etapa de recunoaştere, putem utiliza toate posturile din gestul respectiv, oferind astfel informații complete însă procesul va fi foarte costisitor din punctul de vedere al timpului. În plus nu toate posturile oferă o informație utilă. De exemplu mănuşa 5DT Ultra oferă 60 de pachete de date pe secundă ceea ce înseamnă că pentru un gest compus de complexitate redusă (cum ar fi deschiderea pumnului) efectuat cu o viteză mică posturile succesive vor fi foarte apropiate puține dintre ele fiind relevante. De aceea recomand ca înaintea includerii unui gest compus în biblioteca noastră să realizăm procesarea acestuia.

Pentru algoritmul ce va fi prezentat în această secțiune procesarea unui gest compus constă din filtrarea sa, adică selecția dintre toate posturile a celor mai relevante în modul următor:

1. se va stabili un prag de separare ce va fi folosit în filtrare (se va explica în continuare cum anume)
2. prima postură din gestul compus total este aleasă automat
3. următoarea postură aleasă va fi cea pentru care distanța euclidiană față de postura anterior aleasă
4. se repetă pasul 3 până când nu mai avem posturi în gestul compus total

În final vom avea un gest compus alcătuit din doar câteva posturi simple reprezentative pentru gestul compus total. Numărul de posturi ale gestului compus final va depinde de pragul ales la pasul 1. Cu cât pragul ales este mai mic cu atât vom avea mai multe posturi(pentru un prag egal cu 0 gestul compus final va coincide cu gestul compus total).

Odată ce avem o bibliotecă de astfel de gesturi procesate putem trece la faza de recunoaştere care constă la modul cel mai simplist din preluarea unei succesiuni din pachetele de date primite de la mănuşă şi căutarea unui gest echivalent în bibliotecă. Având în vedere cantitatea de date achiziționată de la mănuşă această abordare este mult prea costisitoare din punctul de vedere al timpului. De aceea fiecare succesiune de posturi primite va trece printr-un proces inițial de selecție. Astfel fiecare postură primită va fi analizată mai întâi individual în modul următor:

1. în primul rând vom avea o listă de gesturi compuse candidat în care vom reține indicii acelor gesturi compuse din bibliotecă, reprezentând un posibil echivalent pentru o anumită succesiune de posturi venite de la mănuşă a cărei selecții va fi descrisă în continuare
2. postura curentă este comparată cu posturile de start ale gesturilor din bibliotecă căutându-se cei mai apropiați vecini față de care distanța euclidiană este mai mică decât un prag stabilit anterior(ca în cazul recunoaşterii gesturilor simple). Toți aceşti vecini vor fi memorați în lista gesturilor candidat. În acelaşi timp se reține şi momentul în care a fost găsit gestul de start respectiv. Din acest moment vom reține toate posturile găsite.
3. postura curentă este comparată cu posturile de stop

ale gesturilor candidat, găsindu-se cei mai apropiați vecini față de care distanța euclidiană este mai mică decât un prag stabilit anterior. În cazul în care avem asemenea gesturi compuse trecem la compararea succesiunilor de posturi înregistrate între posturile de start şi de stop primite, corespunzătoare gesturilor compuse din lista candidat.

4. dacă o succesiune de posturi este identificată ca fiind unul din gesturile compuse din bibliotecă se iau deciziile specifice aplicației şi se golesc toate listele curente. Din acest motiv se recomandă ca un gest compus să nu aibă în componența sa un alt gest compus care e de asemenea în bibliotecă pentru că e puțin probabil să fie vreodată recunoscut fiind astfel inutil. De asemenea este de preferat ca postura de stop să apară doar o singură dată (la sfârşit) în cadrul unui gest compus.
5. dacă nu avem nici o succesiune de posturi care să fie identificată ca fiind un gest compus din biblioteca noastră facem „curățenie" în listele noastre în sensul că eliminăm din lista de gesturi candidat acele gesturi pentru care s-a depăşit o limită de timp(pentru a evita o aglomerare a listelor şi identificări eronate ale unor succesiuni candidate).

Presupunând că am găsit o succesiune de posturi care poate reprezenta un gest compus (au acelaşi gest de start şi stop) trebuie să introducem o măsură a distanței dintre ele(în cazul gesturilor simple aceasta era distanța euclidiană) şi să vedem dacă distanța dintre ele este destul de mică pentru a fi considerate „egale". În cele din urma decizia va fi făcută tot în funcție de un prag stabilit anterior.

În primul rand succesiunea înregistrată de posturi trebuie să treacă prin acelaşi proces de selecție ca toate gesturile din biblioteca noastră, obținând de fapt în final un nou gest compus. În continuare va fi descris algoritmul ce permite calcularea distanței dintre două gesturi compuse.

O soluție pentru calculul distanței ar fi să calculăm suma distanțelor dintre posturile ce compun gesturile comparate. Însă nu este garantat ca în urma procesului de selecție cele două gesturi compuse finale vor avea acelaşi număr de posturi componente. De aceea cele două gesturi compuse trebuie aduse la o formă în care vor avea acelaşi număr de posturi şi in acelaşi timp să nu existe o altă organizare pentru care distanța dintre ele să fie mai mică. Cu alte cuvinte gesturile trebuie reorganizate astfel încât să aibă aceeaşi structură şi distanța între ele să fie minimă. Această cerință încadreaza algoritmul utilizat în categoria algoritmilor de tip „dynamic time warping"[6].

Metoda „dynamic time warping" poate fi aplicată pentru a compara două gesturi astfel:

1. având două gesturi compuse alcătuite din mai multe gesturi simple, scopul algoritmului este de a grupa posturile fiecărui gest într-un număr egal de blocuri formate din una sau mai multe posturi, fiecare bloc urmând să fie înlocuit la final cu media posturilor din care este compus;
2. vom nota gesturile compuse ce sunt comparate cu gc1 si gc2(n1 – numărul de posturi ale lui gc1, n2 – numărul de posturi ale lui gc2) şi vom construi inițial câte o matrice pentru fiecare cu următoarea semnificație – elementul de pe poziția i şi j, cu i<j, reprezintă media blocului de posturi dintre i şi j (matrice necesară pentru ca ulterior să avem aceste informații fără a face calcule repetate inutil); i reprezintă indicele unei posturi din gc1 iar j indicele unei posturi din gc2
3. soluția va fi construită în 3 matrici care ne vor indica pentru posturile fiecăruia dintre cele două gesturi compuse blocurile din care fac parte şi poziția de început a blocului respectiv
4. inițial vom grupa primul element din gc1(elementul cu indicele 0) cu blocurile (0,j) din gc2, $0 \le j \le n2$, şi primul element din gc2 pe rând cu blocurile (0,i) din gc1, $0 \le i \le n1$

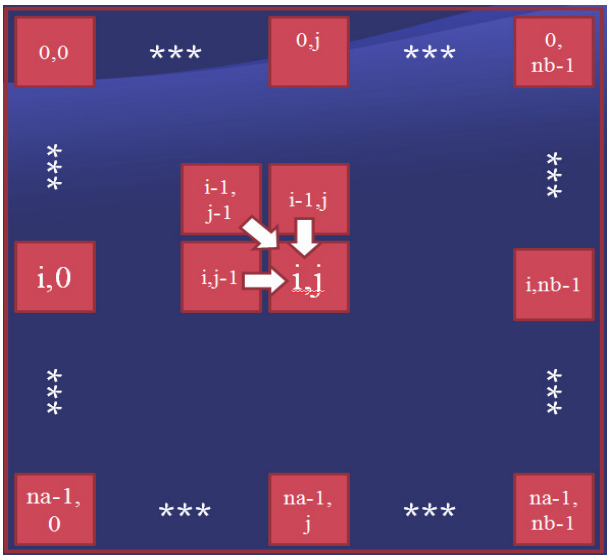

Fig. 2. Construirea matricii solutie.

apoi pentru fiecare pereche de posturi (i,j) cu $1 \le i \le n1$ si $1 \le j \le n2$ vom avea patru opțiuni pentru ca în final să avem acelaşi număr de blocuri pentru ambele gesturi compuse şi distanța dintre ele să fie minimă, bazându-ne pe faptul că orice pereche (i1,j1) procesată anterior ne va da o soluție validă şi optimă:

- perechea (i-1,j) – integrăm postura i din gc1 în ultimul bloc stabilit pentru gc1 pentru (i-1,j) şi calculăm distanța dintre blocuri pentru (i,j) – d1
- perechea (i,j-1) – integrăm postura j din gc2 în ultimul bloc stabilit pentru gc2 pentru (i,j-1) şi calculăm distanța dintre blocuri pentru (i,j) – d2
- perechea (i-1,j-1) – 2 cazuri
- integrăm postura i din gc1 şi j din gc1 în ultimele blocuri stabilite pentru gc1 şi gc2 pentru (i-1,j-1) şi calculăm distanța dintre blocuri pentru (i,j) – d3

- creăm câte un bloc nou pentru gc1 (ce va conține postura i din gc1) şi pentru gc2(ce va conține postura j din gc2) şi calculăm distanța dintre blocuri pentru (i,j) – d4
- alegem dintre cele 4 distanțe calculate pe cea minimă şi facem modificările necesare în cele trei matrici în care construim soluția. De menționat e că pentru a putea fi comparate cele patru distanțe vor fi împărțite fiecare la numărul de blocuri pentru care au fost calculate.

În final vom obține cele două gesturi compuse organizate pe blocuri în mod optim astfel încât distanța dintre ele să fie minimă şi pe baza acesteia vom decide dacă putem considera cele doua gesturi ca fiind egale.

Recunoştinte

Aş vrea să mulțumesc domnului profesor Radu Vatavu pentru ajutorul şi sprijinul acordat pe întreaga perioadă a studiului şi dezvoltării acestor algoritmi de recunoaştere a gesturilor.

REFERINȚE

[1] Dicționarul explicativ al limbii române, Academia Română, Institutul de Lingvistică „Iorgu Iordan", Editura Univers Enciclopedic, 1998
[2] R.O. Duda, P.E.Hart & D.G. Stork, „Pattern Classification", 2nd Edition 2001,pp. 4.21-30
[3] R. Motwani, A. Naor & R. Panigraphi , „Lower bounds on Locality Sensitive Hashing", pp.1-3
[4] J. Clausen, „Branch and Bound Algorithms – Principles and Examples", 1999, pp. 2-10
[5] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, A.E. Abbadi, „Vector Approximation based Indexing for Non-uniform high Dimensional Data Sets", pp. 2-7
[6] P. Senin, „Dynamic Time Warping Algorithm Review", 2008 pp.3-9
[7] "5DT Data Glove Ultra Series User's Manual", Fifth Dimension Technologies,2004,www.5DT.com