ISIM_GestureCapture: Java gesture recognition for natural interaction

M. Corona*, C. Giovannella+
S. Panzieri*, P. Selva+
P. Vargiu*+
*Dipartimento di Informatica e Automazione
Università degli Studi di Roma III
+ScuolaIAD & ISIM_lab
University of Rome Tor Vergata
I-00133 Rome
info@mifav.uniroma2.it

ABSTRACT

We present the design and development of the ISIM_MotionCapture, a Java application for 2D gesture recognition, that can be used, also in daily light condition, with an efficiency of about 83%.

Categories and Subject description

H.5 INFORMATION INTERFACES AND PRESENTATION; H.5.2 **[User Interfaces]**; I.4 IMAGE PROCESSING AND COMPUTER VISION; I.4.7 **[Feature Measurement]**; I.5 PATTERN RECOGNITION; I.5.5 **[Implementation]**; H.1.2 **[User/Machine Systems]**.

General Terms

Human Factors, Algorithms, Languages, Measurements, Design

Keywords

Gesture tracking, gesture recognition, multimodal interaction, computer vision, data, fusion, human factors, interaction design.

1. INTRODUZIONE

Stiamo vivendo in un frangente storico, che potremmo definire era organica dell'interazione, in cui la smaterializzazione dell'informazione, la miniaturizzazione di tutti i dispositivi elettronici con conseguente pervasività della macchina, il potenziamento delle infrastrutture di rete base necessaria per lo sviluppo di internet delle cose, stanno rendendo i 'place' e più in generale la società tutta, liquidi.

In un cotal contesto - in cui sempre più centrale sarà la progettazione per l'esperienza - l'interazione con la macchina, sempre più nascosta e discreta, non potrà che svolgersi secondo modalità quanto più possibile naturali che, molto probabilmente, ci porteranno a interagire più che con la 'macchina' con la computabilità diffusa e le sue reificazioni immateriali.

Gesti, voce e percezione dello stato emotivo diverrano i canali modali privilegiati dello scambio comunicativo mediato. Le realizzazioni di questi ultimi tempi di grandi superfici multitouch [1], di schermi flessibili [2], di "superfici" di varia natura sensibili all'interazione gestuale [3] non sono altro che gli apripista di quanto probabilmente vedremo nei prossimi anni.

La gestualità è una modalità di interazione molto ricca all'interno della quale può essere veicolata tra il 70% e l'80% del contenuto di un discorso. E per questo che il visual gesture tracking e il gesture recognition sono delle branche della ricerca che a partire dalla metà degli anni '90 si sono costantemente sviluppate [4] sino a consentire le realizzazioni sopra elencate. Ancor prima, dalla fine degli anni '70, si sono sviluppati anche sistemi di tracking basati su l'uso di guanti che vanno dai costosissimi Cyberglove II agli economicissimi guanti sperimentali in Lycra come quelli sviluppati all'inizio di questo decennio [5], agli ultimissimi esiti rappresentati dai sistemi g-speak [6]. Indipendentemente dal costo del dispositivo di tracciamento, l'uso di guanti o similaria limita l'uso della comunicazione gestuale a contesti molto specifici (games, ambienti virtuali di simulazioni, particolari applicazioni mediche, ecc...) e non ne consente la portabilità su applicazioni di massa. Non a caso, con lo sviluppo di webcam sempre più performanti ed economiche il "main stream" della ricerca si è focalizzato sempre di più sui sistemi ottici.

E' in questo filone che si inserisce il nostro lavoro di design e sviluppo di un applicativo compatto, operante in real-time e cross-plat (dunque potenzialmente utilizzabile in applicazioni di massa) per il tracciamento (tracking) e il riconoscimento in real-time di gesti in 2D in vari situazioni di illuminamento: dall'infrarosso alla luce diurna.

Nel seguito descriveremo dapprima l'architettura e le caratteristiche dell'applicativo (par. 2), per poi passare alla discussione di alcuni test effettuati sul suo funzionamento (par. 3) e concludere, infine, con la descrizione dell'interfaccia di controllo del sistema e l'accenno ad alcune dimostratori.

2. ARCHITETTURA E PECULIARITA': TRACKING E RICONOSCIMENTO

I sistemi di riconoscimento dei gesti si possono dividere tra: a) sistemi che operano a posteriori (ad acquisizione del gesto avvenuta) attraverso tecniche di "template matching", ovvero di confronto tra le features caratteristiche dei tracciati

registrati e quelle dei template presenti nel database del sistema [4]; usualmente il riconoscimento è basato sul calcolo della distanza euclidea e non prevede una fase di rappresentazione; b) sistemi dinamici che si servono di varie tecniche, spesso tra di loro ibridizzate: dal template matching alle reti neuronali, alla logica fuzzy, agli ANFIS e, per quello che a noi interessa maggiormente qui, alle Hidden Markov Chain (HMM) [7].

L'ISIM_MotionCapture è stato progettato per essere un riconoscitore dinamico real-time e comprende due diversi livelli di operatività: i) la rilevazione e il tracking delle aree di interesse (siano esse in movimento o meno), a sua volta, composto da due sottolivelli rappresentati dal rilevamento dei blob di colore e da quello delle aree in movimento; ii) l'identificazione delle features e il riconoscimento delle tracce in 2D.

Si è deciso di sviluppare l'ISIM_MotionCapture in Java (benché alcune delle prove iniziali siano state condotte in Matlab) per rendere l'applicativo compatibile con tutti i sistemi operativi; questo nonostante le controindicazioni che circolano e che indicano Java come un linguaggio troppo lento per affrontare task di computer vision in real time, anche solo per il blob-tracking!

2.1 Il tracking dei blob di colore

Una volta lanciato, l'ISIM_MotionCapture avvia il processo di cattura delle immagini da webcam servendosi delle API FMJ (Feedom for Media in Java) - che di recente hanno rimpiazzato, nel nostro sistema le API JFM (Java Media Framework). Immediatamente dopo, all'utilizzatore del pacchetto viene data la possibilità di scegliere lo spazio di lavoro (infrarosso o colore) e la modalità (single blob o multi blob). Nel caso di immagini infrarosse il sistema procede all'identificazione delle aree di interesse, blob, cercando tutti i pixel che presentano un livello di luminosità compresa all'interno di un range preassegnato per default, i cui limiti possono essere in ogni momento modificati dall'utilizzatore. Viene effettuato un controllo di prossimità tra pixel e con una tecnica di flood-filling vengono individuati e classificati i blob presenti nell'immagine (utilizzabili anche per lavorare in configurazione multitouch). Nel caso delle immagini a colori si ha la possibilità di scegliere tra l'utilizzo di uno spazio colore RGB o quello di uno spazio ridotto R*G* più adatto al riconoscimento, ad esempio, del colore della pelle, e comunque più stabile nei confronti di possibili variazioni di intensità luminosa. E' necessario, poi, identificare il colore dei blob che si intende tracciare; è sufficiente fare click su di un pixel dell'immagine video che abbia la colorazione desiderata o utilizzare le barre R, G e B per la scelta del colore messe a disposizione dal pannello di controllo. Queste ultime possono essere impiegate anche per rifinire la precedente selezione del colore effettuata direttamente sull'immagine video tramite un dispositivo-puntatore. Il sistema, nel procedere all'identificazione dei blob, considera quali pixel di identico colore tutti quelli che ricadono all'interno degli intervalli di livello predefiniti per default o modificati dall'utilizzatore tramite le barre R, G e B. Il programma consente, inoltre, di ridurre al minimo il tempo di calcolo, grazie alla selezione di un'area ridotta del fotogramma entro la quale effettuare il tracking dei blob,. Può essere interessante notare che l'applicativo permette di effettuare anche il tracking di blob figli, presenti all'interno del bounding-box di un blob genitore di colore diverso.

Per ciascuno dei blob classificati vengono mostrati, a richiesta, il proprio bounding box e la posizione del baricentro. Se desiderato, i dati sui blob e sulle loro corrispettive traiettorie possono essere registrati su file e utilizzati per analisi successive.

2.2 Il tracking del movimento

La funzionalità di tracking del movimento può essere utilizzata sia autonomamente che in combinazione OR o AND con quello di tracking del blob di colore. A differenza di quest'ultimo il motion tracker non richiede l'immissione di alcun parametro da parte dell'utilizzatore ed entra in attività non appena si avvia il grabbing da webcam.

Una volta catturato, il fotogramma viene convertito in scala di grigio, tramite un filtro GrayscaleBT709, per poi essere messo a confronto con un'immagine di background di cui viene effettuato l'aggiornamento dinamico ogni tot frame. A seguito di tale confronto, e in base a un valore di soglia prestabilito, l'immagine viene binarizzata (1 per i pixel che appaiono differenti rispetto al background) e passata, in caso di uso combinato, al blob tracker che, per ottimizzare le risorse, opererà la ricerca sui soli pixel di valore 1.

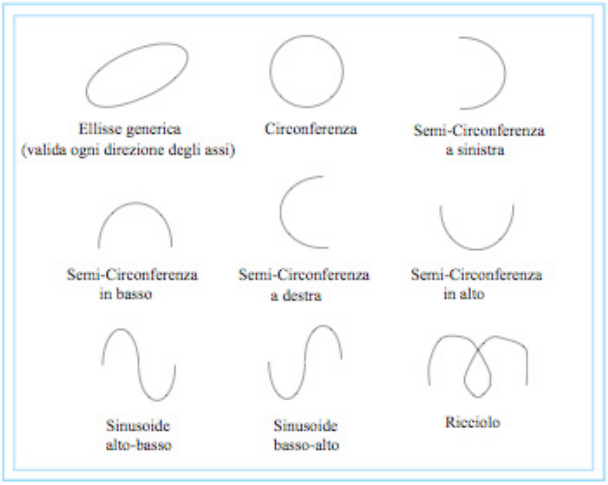
2.3 Il riconoscitore di gesti bidimensionali

Nel caso si voglia effettuare anche il riconoscimento delle figure geometriche disegnate dalle traiettorie dei blob, i dati registrati vengono passati al riconoscitore di gesti; al momento è possibile operare su di una sola traccia alla volta.
Per la costruzione del riconoscitore si è partiti dalla definizione di un vocabolario di gesti, scelti tra quelli che possono essere eseguiti e riconosciuti in maniera abbastanza semplice dalla maggior parte delle persone (inclusi anziani e disabili). Lo vediamo riportato in fig. 1. La separazione in due sottovocabolari relativi a figure curve e a figure rettilinee rispecchia, come vedremo meglio, la struttura dell'estimatore che, sin dai primi segmenti della traiettoria, permette di effettuare una distinzione tra tali categorie di gesti. Il riconoscimento del gesto, infatti, procede punto dopo punto, nell'ipotesi che tutti i gesti possano essere suddivisi in segmenti orientati lungo una delle 8 direzioni individuate da due rette perpendicolari e dalle due bisettrici degli angoli rettangoli formati da tali rette. In questo modo è possibile lavorare prima sulla ricostruzione della sequenza di segmenti di cui è composto il tracciato del gesto e poi sul riconoscimento della figura geometrica associata alla sequenza identificata.

Il primo di questi step si realizza in tempo reale grazie all'utilizzo di uno stimatore probabilistico, basato sulle Catene di Markov Nascoste (HMM) a 9 stati - le 8 direzioni privilegiate più lo stato di fermo. Lo stimatore valuta la probabilità che la sequenza di tratti appartenga a una delle direzioni privilegiate. Non appena riconosciuta con "certezza" la direzione questa viene inserita nella sequenza di primitive che definiranno il gesto da identificare per confronto con il vocabolario pre-definito.

In contemporanea alla ricostruzione della sequenza di segmenti che compone il gesto vengono fatti operare anche:

a) gli stimatori dei parametri geometrici; nel caso di figure curve vengono utilizzati due stimatori ricorsivi ai minimi quadrati, uno per le ellissi e uno per le sinusoidi, mentre nel caso di figure costituite da segmenti rettilinei si usa uno stimatore ai minimi quadrati per le rette e si calcolano i parametri geometrici di ciascun segmento che compone il gesto;



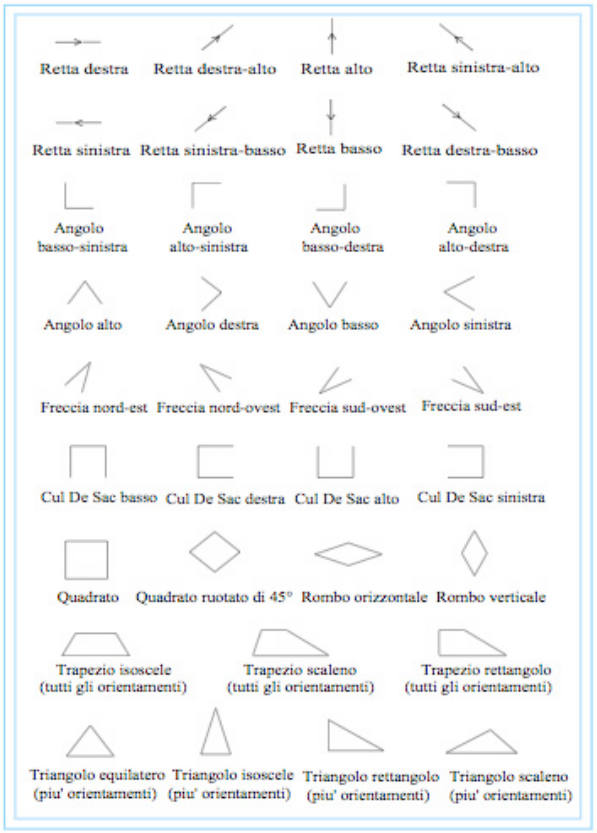


Fig. 1: vocabolario di gesti utilizzati come base per il riconoscimento suddiviso in figure curvilinee e figure rettilinee

b) il calcolo di alcune variabili di controllo - curvatura media della funzione approssimante, scarto dalla curva di fitting ed errore quadratico medio rispetto alla stima effettuata - che, non appena la traccia del gesto raggiunge una lunghezza di soglia predefinita, consente di prendere una decisione sulla sottocategoria di appartenenza del gesto (fig. 1) e, quindi, di procedere allo spegnimento di tutti gli stimatori non più utili.

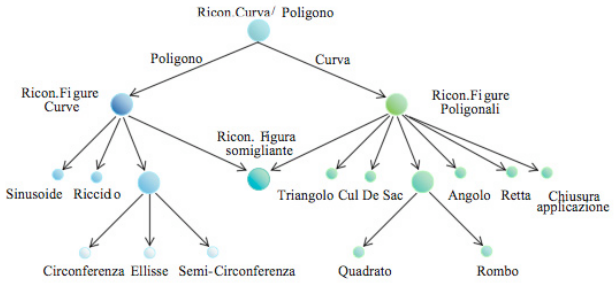


Fig. 2: rappresentazione dell'albero decisionale utilizzato per identificare la forma geometrica associata al gesto

A gesto concluso, e sequenza di primitive definita, si passa alla classificazione finale del gesto, alla conseguente elaborazione di tutte le sue caratteristiche geometriche e al calcolo della velocità di esecuzione. Il riconoscimento avviene sulla base dei dati registrati nella fase precedente - tra questi, ad esempio, la sequenza dei tratti, l'angolo totale percorso (figure curve), la matrice delle diagonali (solo per discriminazione tra rombi e quadrati), le coordinate del primo e dell'ultimo punto, ecc... - in base all'albero decisionale rappresentato in fig. 2.

3. I TEST DI EFFICIENZA

Per valutare il potenziale utilizzo dell'applicativo in applicazioni di massa, sul riconoscitore 2D sono stati eseguiti anche alcuni test di efficienza mirati a calcolare: a) il fattore di riconoscimento definito come E = numero gesti riconosciuti/numero gesti effettuati; b) il tempo di reazione (CT), ovvero il tempo che intercorre tra l'avvio dell'esecuzione e il riconoscimento dello stesso; il numero di features necessarie al riconoscimento del gesto.

I test sono stati eseguiti utilizzando 15 soggetti per un totale di 246 acquisizioni. Ai soggetti è stato chiesto di posizionarsi di fronte all'interfaccia e di eseguire il gesto loro richiesto; nel caso di gesto eseguito non conforme alla richiesta è stata proposta la sua ripetizione. Per questo test il tracciamento del gesto è stato effettuato utilizzando il solo tracker di colore; test basati sul tracciamento combinato di colore e movimento sono in corso di realizzazione e non ancora disponibili.

L'efficienza totale del riconoscitore è risultata essere dell'82,9%. In tabella 1 sono riportati i risultati segmentati per tipologia di gesto prodotto. Come si può notare le figure curve vengono riconosciute con una più alta percentuale 86,7% contro l'80,8% riscontrato per i gesti composti da segmenti lineari. Una possibile spiegazione di questo risultato potrebbe essere individuata nel fatto che nel primo caso il riconoscitore lavora su di un'unica funzione geometrica (curva continua), mentre nel secondo caso utilizza una sequenze di segmenti la cui direzione può variare repentinamente a causa

dell'imprecisione del gesto eseguito. Si consideri infatti che la decisione di lavorare con una HMM a 9 stati comporta un possibile errore nel riconoscimento della direzione del segmento rettilineo quando lo scarto superiori i 22°.

Tabella 1 - Efficienza di riconoscimento segmentata per tipologia di figura geometrica

Gesto	N. gesti registrati	E (eff. riconoscim.)
Angolo	26	89,7
Retta	12	80,0
Cul de Sac	16	88,9
Triangoli	16	76,2
Quadrati/Rombi	33	78,6
Trapezi	17	70,8
Chiusure apllicazione	6	85,7
Circonferenze/Ellissi	33	84,6
Semi-Circonferenze	15	93,7
Sinusoidi	20	90,9
Riccioli	10	76,9
COMPLESSIVO	246	82,9

Una percentuale di errore relativamente alta viene riscontrata anche per tutte le figure poligonali chiuse. La ragione di ciò è da rintracciarsi, probabilmente, nella difficoltà per l'utente di far coincidere il punto di inizio con il punto finale, ovvero di effettuare una buona chiusura della figura poligonale. In questi casi, infatti, il riconoscitore potrebbe interpretare la traccia come figura aperta. Non essendo, però, quest'ultima inclusa nel vocabolario pre-definito il riconoscitore termina in una condizione di errore. Situazioni di forte emotività dell'utente, inoltre, possono produrre gesti ondeggianti e/o incerti della mano che rendono arduo il compito del riconoscitore, anche nel caso più semplice di una retta. Il problema della chiusura è stato osservato anche nel caso di figure curve quali l'ellissi/cerchio. Per queste figure geometriche, inoltre, un prolungamento oltre il punto di chiusura aspettato può generare possibili ambiguità di riconoscimento tra le suddette figure geometriche e il ricciolo.

4. L'INTERFACCIA DI CONTROLLO

In fig. 3 è mostrato un prototipo dell'interfaccia di controllo dell'ISIM_GestureCapture. Il pannello di controllo è diviso in tre aree. Su quella di sinistra è possibile visionare le immagini provenienti dalle telecamere. Il programma si presta a lavorare con un numero N a piacere di telecamere, essendo limitato dalla sola potenza di calcolo della CPU. La figura mostra un esempio di utilizzo di due telecamere per la gestione di una bacheca interattiva in luce diurna. Le check box posizionate sotto l'immagine consentono di associare le varie viste a differenti assi e di scegliere, eventualmente, configurazioni di assi ruotati di 45°. Il colore selezionato per il tracking dei blob (azzurro) è mostrato nella colonna centrale, mentre nella colonna di sinistra, sovraimposti alle immagini provenienti della webcam sono mostrati i rettangoli di lavoro, i bounding box associati ai blob tracciati, con evidenziazione degli estremi e del baricentro. Le immagini della webcam possono essere nascoste lasciando visibili i soli i blob di colore.

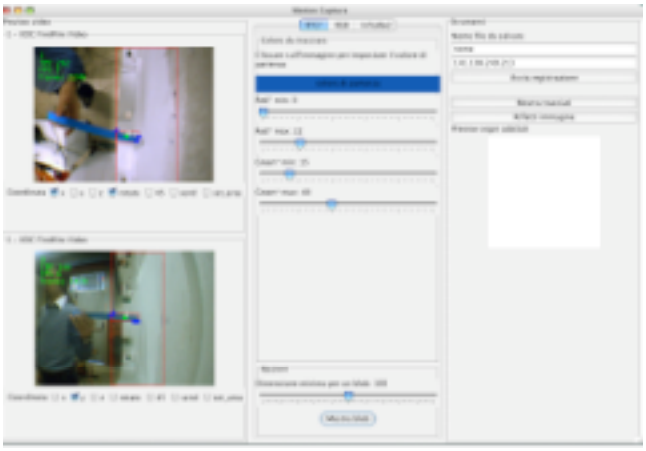


Fig. 3: Interfaccia grafica della plancia di controllo dell'ISIM_MotionCapture

Nella parte centrale è possibile scegliere lo spazio di colore di lavoro - nel caso in figura è stato scelto lo spazio ridotto R*G* -, agire sugli slider che definiscono gli intervalli di colore all'interno dei quali i pixel dovranno essere considerati di colore equivalente, selezionare la dimensione lineare minima perché i cluster di pixel individuati possano essere considerati come blob significativi.

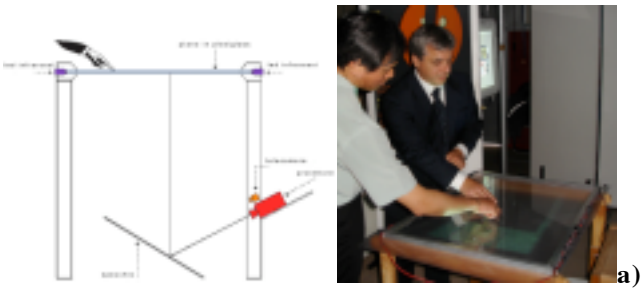
Sulla colonna di destra ci sono i controlli che consentono di registrare il file dati e/o di avviarne la trasmissione ad altro computer tramite protocollo OSC. Nella stessa colonna trova posto anche il proiettore per la previsualizzare dei tracciati che verranno dati in pasto al riconoscitore (i tracciati possono essere mostrati anche riflessi, onde evitare di confondere chi dovesse usare, nel corso del tracciamento del gesto, il proiettore come schermo di controllo). Vale la pena notare che il sistema può acquisire dati anche da sistemi diversi dalle webcam, come mouse o tablet.

In fig.4 sono mostrati a mo' di esempio, quantunque non siano a livello delle realizzazioni citate nell'introduzione, alcuni prototipi che fanno uso dell'ISIM_GestureCapture: un prototipo di tavolino multitouch (che lavora con una webcam infrarossa), un prototipo di bacheca che consente di interagire con i contenuti immateriali tramite "bacchetta magica" e un prototipo di sintetizzatore che utilizza dei marker attivi all'infrarosso.

5. CONCLUSIONI

In questo articolo abbiamo dimostrato - risultato niente affatto scontato - la realizzabilità in Java di un riconoscitore in tempo reale di gesti bidimensionali in grado di gestire n telecamere, utilizzabile sia con sorgenti infrarosse che in luce diurna. Mentre il tracking è operante anche in configurazione multiblob, al momento, il sistema di riconoscimento è utilizzabile su di una sola traccia alla volta e presenta un'efficienza di riconoscimento globale di quasi l'83%. Un

livello che si pensa di poter facilmente migliorare sia aumentando il numero di stati sui quali far lavorare le HMM che utilizzando stimatori perfezionati adatti a riconoscere con più precisione la chiusura di curve poligonali ed ellissi e le direzioni degli angoli. Si pensa inoltre che una più alta efficienza potrebbe ottenersi introducendo dei riconoscitori basati reti ANFIS [8].


a)


b)


c)

Fig. 4: Prototipi utilizzanti l' ISIM_GestureCapture: un tavolino multitouch (a), una bacheca che consente di interagire tramite "bacchetta magica" (b) e un sintetizzatore che utilizza dei marker attivi a led infrarossi (c).

Il lavoro qui presentato apre, inoltre, interessanti prospettive per lo sviluppo di riconoscitori di gesti tridimensionali basati sulla combinazione di immagini provenienti da due webcam. Le prime esplorazioni a riguardo sembrano molto promettenti e verranno descritte in contributi futuri.

6. RINGRAZIAMENTI

Gli autori di questo articolo sono grati agli studenti di Scienza dei Media e della Comunicazione che con i loro lavori di tesi hanno dimostrato l'utilità dell' ISIM_MotionCapture: Stefania Castiglioni, Giulio Pernice, Valeria Righi.

7. RIFERIMENTI BIBLIO-WEBGRAFICI

[1] Han, J. Y., 2005, *Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection,* In Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology
http://www.youtube.com/watch?v=zwGAKUForhM
http://www.youtube.com/watch?v=J5SQmuJYSck

Microsoft surface
http://www.youtube.com/watch?v=Zxk_WywMTzc

http://naturalinteraction.org/

[2] Sony flexible display
http://www.youtube.com/watch?v=1SJN93E8kd4
Rollable displays
http://www.readius.com/
http://www.youtube.com/watch?v=8VoNkd4OGqU

[3] see for exasmple the proceedings of the Gesture Recognition (GW) series of conferences by Springer: Lecture Notes in Computer Science, N.: 5085, 3881, 2915, 2298, 1739, 1371, and references therein

[4] tessuti schermo:
http://www.lumalive.com/business/
http://www.youtube.com/watch?v=Yd99gyE4jCk
muri sensibili
http://www.we-make-money-not-art.com/archives/2004/07/notsowhite-walls-interactive-w.php

[5] see for example: Giovannella C., Selva P. E., 2003, *Smart-Gloves as Internet-Nodes to Interact with Virtual Environments on the Web* - extended abstract - 5th International Gesture Workshop: "Gesture-Based Communication in Human-Computer Interaction",Genova

[6] g-speak
http://oblong.com/

[7] Min B.W., Yoon H.S., Soh J., Ohashi T., Ejima T., 1999, *Visual recognition of static/dynamic gesture: Gesture-driven editing system*, J. of Visual Languages and Computing, 291-309; and 1997, *Hand gesture recognition using Hidden Markov models*, Technical report, Image Processing Div. SERI

Ramamoorthy A. Vasawani N. Chaudhury S. Banerjee, 2002, *Recognition of dynamic hand gesture*, J. of Pattern Recognition Society, 20069-2081

[8] Al-Jarrah O. Halawani A., 2001, *Recognition of arabic sign language using neuro-fuzzy systems*, Artificial Intelligence, 117-138