

Implementazione in Java di un sistema di tracking basato su Optical Flow

RELATORE: Ch.mo Prof. Enoch Peserico Stecchini Negri De Salvi

CORRELATORE: Dott. Alberto Silletti

LAUREANDO: Gambarotto Filippo

Corso di laurea in Ingegneria dell'Informazione



UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

TESI DI LAUREA

Implementazione in Java di un sistema di tracking basato su Optical Flow

RELATORE: Prof. Enoch Peserico Stecchini Negri De Salvi

CORRELATORE: Dott. Alberto Silletti

LAUREANDO: Gambarotto Filippo

A.A. 2011-2012

Indice

| | | |
|----------|---|-----------|
| 1 | Optical Flow | 7 |
| 1.1 | Definizioni generali | 7 |
| 1.2 | Conservazione dell'Intensitá | 9 |
| 2 | Tracking | 11 |
| 2.1 | Definizioni generali | 11 |
| 2.2 | Background | 12 |
| 3 | Applicazioni | 13 |
| 3.1 | KINECT | 13 |
| 3.2 | Descrizione e funzionamento | 14 |
| 3.3 | Articoli | 15 |
| 3.3.1 | Microsoft per lautismo infantile: Kinect aiuta la dia- gnosi precoce | 15 |
| 3.3.2 | Interfacce evolute per visualizzare le immagini medicali | 16 |
| 3.4 | Conclusioni | 21 |
| 4 | Il mio approccio | 23 |
| 4.1 | Introduzione | 23 |
| 4.2 | Algoritmo | 24 |
| 4.2.1 | Cattura delle immagini | 24 |
| 4.2.2 | Riduzione della dimensione delle immagini e conversio- ne a scala di grigi | 25 |
| 4.2.3 | Calcolo del flusso ottico | 26 |
| 4.2.4 | Eeguire l'operazione richiesta | 27 |
| 4.3 | Euristiche | 28 |
| 4.3.1 | Ridurre la grandezza della matrice \mathcal{A} | 29 |
| 4.3.2 | Limitare l'area di ricerca | 29 |

| | | |
|----------|--|-----------|
| 4.3.3 | Intervalli fra i punti da analizzare | 30 |
| 4.3.4 | Filtro gaussiano sulla matrice \mathcal{C} | 30 |
| 5 | Risultati | 33 |

Sommario

Questo elaborato illustrerá un algoritmo per rilevare il movimento di oggetti grazie all'uso di una semplice webcam, allo scopo di aiutare i medici a consultare ed elaborare immagini durante le visite dermatologiche in modo touchless. Sará riportato quindi il metodo utilizzato e altre tecnologie che mirano allo stesso scopo, con le varie applicazioni che queste hanno nel mondo medico.

Introduzione

Da sempre la tecnologia, come anche l'informatica, hanno svolto un ruolo fondamentale nell'ambito medico; il loro utilizzo ha portato a grandi innovazioni, ha permesso di automatizzare e rendere più precisi e sicuri moltissimi processi della medicina. Questo progetto si inserisce per aiutare ulteriormente il medico nel suo ambito, che nello specifico è la dermatologia e il controllo preventivo dei nevi. Naeviinsilico è infatti un grande progetto volto a mappare ogni nevo di una persona, registrandone e studiandone l'evoluzione. L'apparecchiatura consiste in due schermi da 40 pollici installati nell'ambulatorio del medico che, attraverso gli appositi programmi, visualizzano i nevi del paziente fornendo le informazioni necessarie. Durante la visita il dottore potrà, quindi, consultare la storia dei nevi del paziente. Sorge però un problema: come poter sfogliare le immagini che il medico vuole osservare o i grafici che il programma fornisce? In generale i medici dovranno interrompere momentaneamente la visita e svolgere questo lavoro con mouse o tastiera, il che comporta una notevole perdita di tempo e la necessità di ricominciare le operazioni di sterilizzazione. La soluzione che il progetto offre, che verrà presentata in questo elaborato, permette tramite un semplice gesto o movimento della mano di eseguire i comandi scelti, come visualizzare negli schermi la storia del nevo in questione, scorrerne l'evoluzione nel tempo, visualizzare i grafici, oppure passare al nevo successivo. Verranno presentati gli algoritmi seguiti per rendere questo processo tanto preciso quanto rapido nell'esecuzione, i problemi e le soluzioni adottate e i risultati ottenuti; si parlerà inoltre delle applicazioni di questa tecnologia nel mondo medico e delle apparecchiature che vengono utilizzate al momento per questo scopo.

Capitolo 1

Optical Flow

1.1 Definizioni generali

L'Optical Flow é un concetto che considera il moto di un oggetto all'interno di una rappresentazione visuale digitale. Tipicamente il moto é rappresentato come un vettore che si origina da un pixel, ad esempio in una sequenza di frame. Lo scopo dell'Optical Flow é quello di assegnare ad ogni pixel appartenente al frame corrente un motion vector che punta verso la posizione dello stesso pixel in un frame di riferimento successivo. Un punto di partenza comune per la stima dell'optical flow é di assumere che le intensitá dei pixel siano trasmesse da un fotogramma al successivo:

$$I(\vec{x}, t) = I(\vec{x} + \vec{u}, t + 1) \quad (1.1)$$

Dove $I(\vec{x}, t)$ é l'intensitá dell'immagine in funzione dello spazio $\vec{x} = (x, y)^T$ e del tempo t , ed $\vec{u} = (u_1, u_2)^T$ é il vettore di spostamento. Naturalmente, la luminositá rimane raramente costante. L'ipotesi é che la luminositá della superficie rimanga fissa da un fotogramma al successivo. Anche se poco realistico, é interessante notare che la nostra ipotesi sulla luminositá costante funziona bene in pratica. Per ricavare una stima del vettore spostamento \vec{u} , abbiamo preso in considerazione il caso 1D. Sia $f_1(x)$ e $f_2(x)$ segnali (immagini) 1D in due istanti di tempo.

Come illustrato in Fig.1, supponiamo inoltre che $f_2(x)$ sia una versione traslata di $f_1(x)$, ovvero sia $f_2(x) = f_1(x - d)$ dove d indica lo spostamento.

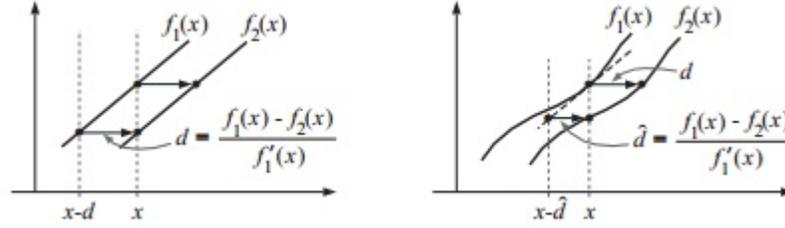


Figura 1.1: Il gradiente vincolato riguarda lo spostamento del segnale per la differenza temporale e spaziale derivati (pendenza). Per lo spostamento di un segnale lineare (a sinistra), la differenza di valori di segnale in un punto diviso per la pendenza consente lo spostamento. Per segnali non lineari (a destra), la differenza divisa per la pendenza da un ravvicinamento per lo spostamento.

Un'espansione in serie di Taylor di $f_1(x - d)$ di x é data da:

$$f_1(x - d) = f_1(x) - df'_1(x) + O(d^2 f''_1), \quad (1.2)$$

dove $f' \equiv df(x)/dx$. Con questa espansione possiamo riscrivere la differenza fra i due segnali nella posizione x come:

$$f_1(x) - f_2(x) = df'_1(x) + O(d^2 f''_1),$$

Ignorando i termini oltre il primo grado, otteniamo questa approssimazione di d :

$$\hat{d} = \frac{f_1(x) - f_2(x)}{f'_1(x)}, \quad (1.3)$$

Il caso 1D generalizza direttamente a 2D. Come sopra, assumere che l'immagine spostata é ben approssimata da una serie di Taylor di ordine uno:

$$I(\vec{x} + \vec{u}, t + 1) \approx I(\vec{x}, t) + \vec{u} \cdot \nabla I(\vec{x}, t) + I_t(\vec{x}, t), \quad (1.4)$$

dove $\nabla I \equiv (I_x, I_y)$ e I_t indicano le derivate parziali spazio e tempo dell'immagine I , $\vec{u} = (u_1, u_2)^T$ indica il vettore spostamento in 2D. Ignorando i termini di grado piu alto della serie di Taylor e sostituendo quindi l'approssimazione lineare (1.1), otteniamo:

$$\nabla I(\vec{x}, t) \cdot \vec{u} + I_t(\vec{x}, t) = 0, \quad (1.5)$$

L'equazione (1.5) riguarda la velocità spazio-tempo dell'immagine derivata da una posizione dell'immagine, ed é spesso chiamato il equazione a gradiente vincolato.

1.2 Conservazione dell'Intensità

Il tracciamento di punti di luminosità costante pu essere anche visto come la stima di percorsi 2D $\vec{x}(t)$ lungo la quale l'intensità é costante:

$$I(\vec{x}(t), t) = c, \quad (1.6)$$

la cui derivata in funzione del tempo é data da:

$$\frac{d}{dt}I(\vec{x}(t), t) = 0. \quad (1.7)$$

Espandere il lato sinistro di (1.7) con la regola della catena ci offre:

$$\frac{d}{dt}I(\vec{x}(t), t) = \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} \frac{dt}{dt} = \nabla I \cdot \vec{u} + I_t, \quad (1.8)$$

dove la derivata del percorso é proprio l'Optical Flow $\vec{u} \equiv (dx/dt, dy/dt)^T$.
Se uniamo (1.7) e (1.8) si ottiene l'equazione a gradiente vincolato (1.5).

Capitolo 2

Tracking

2.1 Definizioni generali

Il tracciamento di oggetti in tempo reale ha una grande varietà di applicazioni in molti campi. Il tracciamento basato su flusso ottico è un meccanismo di monitoraggio in grado di tenere traccia degli oggetti in movimento anche in contesti complessi e in varie condizioni di luce. Nella ricerca presentata in questo documento vengono descritte le possibilità di utilizzare flusso ottico per seguire gli oggetti acquisiti in una telecamera, per estrarre le informazioni di base relative agli oggetti.

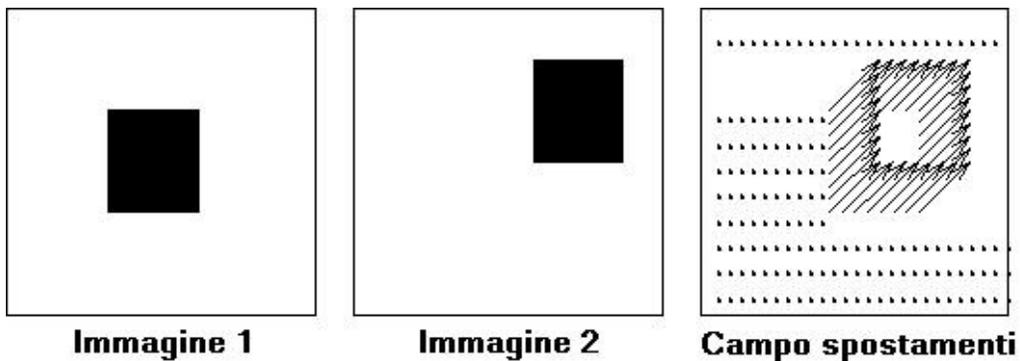


Figura 2.1: Esempio di tracking

2.2 Background

Nella computer vision, il tracciamento di oggetti é un settore importante. Sebbene vi siano molti algoritmi e metodi di monitoraggio, di contare o di identificare oggetti, i metodi piú comunemente usati utilizzano la minimizzazione della differenza di fondo e il tracciamento del colore di base. Il flusso ottico é la quantitá di movimento di un'immagine all'interno di un determinato periodo di tempo. Questo metodo di rilevamento pu essere facilmente convertita in elaborazione in parallelo ed é molto piú veloce rispetto ai tradizionali metodi di monitoraggio. Utilizzando il metodo della differenza di fondo, l'immagine di sfondo viene catturato all'inizio. Quindi, ogni fotogramma del video viene confrontata con la trama iniziale. In questo modo, gli oggetti in movimento possono essere identificati a causa della differenza. Questo metodo pu identificare e tenere traccia degli oggetti anche in ambienti complessi. Il grande svantaggio di questo metodo é l'esigenza di uno sfondo statico. Un altro metodo é quello di individuare alcune proprietá dell'oggetto tracciato e a seguire. Questo metodo pu tenere traccia degli oggetti anche in ambienti complessi e possono tollerare modifiche dello sfondo o i movimenti della telecamera. Ma, se gli oggetti simili sullo sfondo, il tracking pu fallire. Il monitoraggio basato sul flusso ottico che viene utilizzato in questo lavoro é limitato a rilevare lo spostamento degli oggetti. Tuttavia, questo pu tenere traccia degli oggetti in ambienti altamente complessi. Sebbene una telecamera rotante pu produrre valori ingannevole durante la rotazione, non appena la rotazione si ferma, la ricerca pu avere inizio. Non é necessario per descrivere gli oggetti in dettaglio. Inoltre, la velocitá e il flusso delle informazioni in ogni punto possono essere acquistati direttamente se si usa il controllo del flusso ottico.

Capitolo 3

Applicazioni

3.1 KINECT

Tutti conosciamo Microsoft Kinect, la periferica che attraverso un sistema di videocamere e sensori permette di gestire i contenuti sullo schermo con il solo movimento del corpo o anche attraverso comandi vocali. Disponibile per XBox 360 in Europa dallo scorso 10 novembre, il device era inizialmente pensato come semplice periferica di input per comandare i giochi da remoto e senza fili, un po' come accade già ora con altri tipi di controller affini, come ad esempio quello della Nintendo Wii, ma in maniera ancora più avanzata, visto che non richiede l'utilizzo di alcun oggetto o altro dispositivo che faccia da tramite, ma soltanto l'uso del nostro stesso corpo come controller; ed proprio grazie a questa sua proprietà che risultato molto importante nell'ambito che tratteremo.

Tuttavia il prodotto ha riscosso un enorme successo mondiale (con un totale di 8 milioni di unità nei suoi primi 60 giorni, il Kinect detiene il Guinness World Record per essere l'apparecchio nell'ambito dell'elettronica di consumo venduto più velocemente) e le implicazioni delle sue funzionalità hanno aperto inesplorati scenari per quanto riguarda l'interazione uomo-macchina, tanto che molti si erano già messi a lavoro per sviluppare soluzioni che permettessero di impiegare Kinect anche sul computer, per interfacciarsi al sistema operativo e ad altri applicativi. Consapevole del successo e dell'interesse Microsoft promise di rilasciare un SDK (Software Development Kit) che permettesse in maniera ufficiale di portare questo rivoluzionario controller su PC, cosa che è successa nel giugno del 2011. Il Software Development Kit

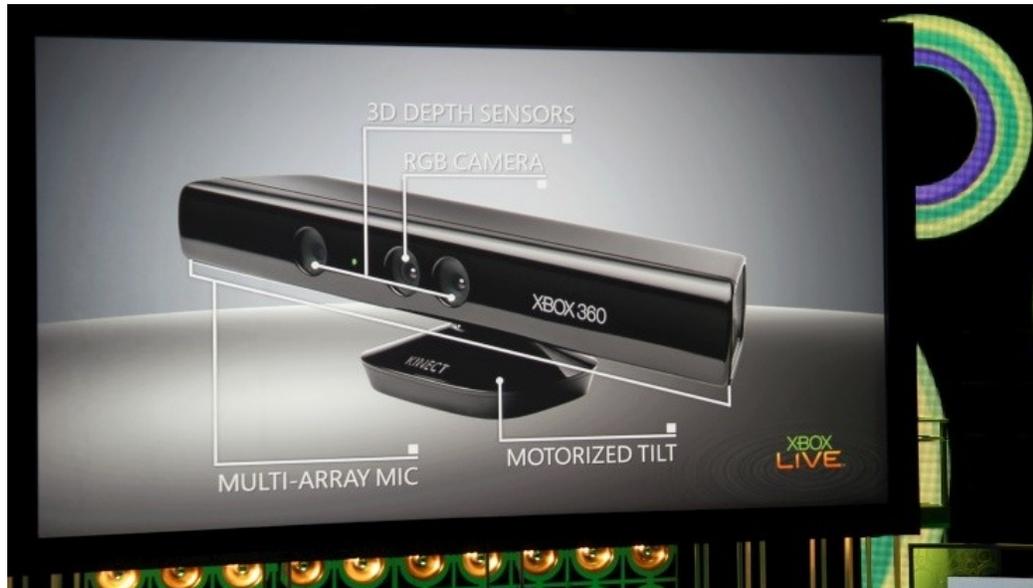


Figura 3.1: Modello di Kinect

contiene numerosi driver, librerie e altre risorse ancora, che consentono agli sviluppatori di creare applicazioni in Visual Basic, C++ e C#, tramite l'uso di Microsoft Visual Studio 2010. Attualmente la beta è destinata principalmente alle comunità scientifiche e accademiche, ma anche gli appassionati e i singoli sviluppatori possono scaricarla per saggiarne le enormi potenzialità. Presenterò quindi una piccola descrizione dell'apparecchio ed alcune applicazioni pratiche molto interessanti che esulano dall'ambiente ludico.

3.2 Descrizione e funzionamento

Kinect è dotato di telecamera RGB, doppio sensore di profondità a raggi infrarossi composto da un proiettore a infrarossi e da una telecamera sensibile alla stessa banda. La telecamera RGB ha una risoluzione di 640 480 pixel, mentre quella a infrarossi usa una matrice di 320 240 pixel. Kinect dispone anche di un array di microfoni utilizzato dal sistema per la calibrazione dell'ambiente in cui ci si trova, mediante l'analisi della riflessione del suono sulle pareti e sull'arredamento. In tal modo il rumore di fondo e i suoni del gioco vengono eliminati ed è possibile riconoscere correttamente i comandi

vocali (in Italia, il controllo vocale sarà attivo dal 6 dicembre 2011, data che vedrà anche un importante aggiornamento della dashboard di Xbox 360, con una nuova interfaccia sullo stile quella di Windows 8 e l'introduzione della Xbox Live TV). La barra del Kinect é motorizzata lungo l'asse verticale e segue i movimenti dei giocatori, orientandosi nella posizione migliore per il riconoscimento dei movimenti. Di fatto, la periferica permette all'utente di interagire con la console senza l'uso di alcun controller da impugnare, ma solo attraverso i movimenti del corpo, i comandi vocali o attraverso gli oggetti presenti nell'ambiente. Microsoft dichiara che Kinect pu seguire i movimenti di al piú 4 giocatori, sia in piedi che seduti.

3.3 Articoli

Qui di seguito ho proposto alcuni articoli molto interessanti dell'utilizzo del kinect per scopi scientifici:

3.3.1 Microsoft per lautismo infantile: Kinect aiuta la diagnosi precoce

Non solo un gioco. Kinect si rivela ancora una volta l'amico dei piú piccoli, prima aiutandoli a superare alcuni problemi legati alla sfera del comportamento, ora consentendo la diagnosi precoce dell'autismo. La periferica, sviluppata per Xbox 360 per permettere all'utente di interagire con la console senza l'uso di alcun controller da impugnare, ma soltanto attraverso i movimenti del corpo, i comandi vocali o attraverso gli oggetti presenti nell'ambiente, é stata utilizzata nel Minnesota per diagnosticare l'autismo in maniera precoce.

In una classe di bimbi dai 3 ai 5 anni sono state disposti cinque Kinect collegati a relativi PC per monitorare e memorizzare gli spostamenti e le azioni di questi bambini. Ma non é la prima volta che la periferica dell'Xbox viene utilizzata a scopi scientifici e non puramente ludici. Guardando al passato ancora il Minnesota é protagonista di un altro esperimento di tale tipo. Kinect venne impiegato per rilevare alcuni disturbi infantili. E con successo. L'Universitá di medicina, in collaborazione con il College of Science and Engineering, ha annunciato di aver realizzato un software open-source per il sensore di movimento di Microsoft, capace di monitorare i movimenti e i comportamenti dei piccoli pazienti. Il programma permette di osservare at-



Figura 3.2: Il personale del Lakeside Center ha integrato la tecnologia di riconoscimento di Kinect nelle sessioni di terapia per aiutare i bambini affetti da autismo

tentamente i movimenti eseguiti dai bambini durante i momenti di gioco, ed evidenziare alcune malattie, come i disturbi ossessivi-compulsivi e alcuni deficit dell'attenzione. Il professore Nikolaos Papanikolopoulos ha confermato che l'uso di Kinect in questo ambito medico risulta meno costoso e molto più efficace rispetto a metodi alternativi che utilizzano strumenti professionali.

Il prossimo articolo in particolare é utilissimo per capire la vera funzione del tracking e del lavoro che ho svolto; questo progetto é, infatti, molto simile al mio come intenti e svolgimento, anche se é su un livello completamente diverso:

3.3.2 Interfacce evolute per visualizzare le immagini medicali

Il crescente utilizzo di procedure guidate dalle immagini in ambienti sterili é alla base della domanda di una nuova tipologia di interfacce uomo-macchina



Figura 3.3: I medici addetti alla riabilitazione del Royal Berkshire Hospital hanno fatto di Kinect una parte importante del processo di recupero nei pazienti colpiti da ictus o altre malattie al cervello.

compatibili con l'asepsi degli ambienti. Recenti studi scientifici, infatti, hanno dimostrato che l'utilizzo da parte di medici e infermiere di dispositivi di input tradizionali, quali mouse e tastiere, è ad oggi la principale causa del propagarsi di infezioni in ambienti sterili. Questo articolo descrive un sistema innovativo che consente di visualizzare e manipolare interattivamente immagini medicali in ambienti sterili senza dover toccare alcun tipo di dispositivo. L'interazione con il sistema, che avviene tramite riconoscimento di postura e movimenti di mani e braccia, è resa possibile dall'utilizzo del controller Kinect della console Microsoft Xbox, che funge da unico dispositivo di interazione. [...]

INTERAZIONE TOUCHLESS IN SALA OPERATORIA STATO DELL'ARTE Sin dai primi anni 90, numerose iniziative di ricerca sono state attivate per realizzare sistemi basati su videocamere che consentissero al personale medico di esplorare immagini digitali in ambienti sterili. I primi prototipi sono per stati perfezionati solo in tempi recenti. Nel 2008 è stato realizzato Gestix, un sistema basato su singola videocamera, che è in grado di tracciare i movimenti della mano e di utilizzarli per consentire la navigazione di im-



Figura 3.4: Tedesys utilizza la tecnologia di Kinect in alcune sale operatorie in Spagna per aiutare i medici a gestire l'Imaging a risonanza magnetica e le scansioni per la tomografia computerizzata con il solo movimento delle mani.

magini di risonanza magnetica. Tuttavia, tale sistema non pu funzionare in ambienti scarsamente illuminati, richiede che l'operatore resti a una distanza prefissata dal sistema, e ha un basso tasso di riconoscimento se l'utente esegue movimenti rapidi o se vi sono numerosi oggetti sullo sfondo. Per superare tali limitazioni, sono stati proposti sistemi basati sull'utilizzo di due videocamere poste in configurazione stereo, allo scopo di ricostruire la profondità della scena e quindi di ovviare alla perdita di informazione derivante dall'utilizzo di proiezioni bidimensionali di ambienti tridimensionali. È del 2009 il primo prototipo di questo tipo, WagO. Tale sistema, per, presenta delle difficoltà legate al costo computazionale richiesto per risolvere il problema della corrispondenza, allo scopo di consentire una corretta ricostruzione della scena tridimensionale, e a calibrazione e corretto posizionamento delle videocamere, il che lo rende poco pratico per un utilizzo in ambienti altamente dinamici quali le sale operatorie. Più recentemente, il primo sistema basato su Kinect è stato realizzato ed installato in una sala operatoria dello spedale Sunnybrook di Toronto. Tale sistema consente al chirurgo di visualizzare immagini tomografiche senza dover lasciare il campo sterile intorno al paziente. In ba-

se ai test riportati, per il solo fatto di consentire al chirurgo di non eseguire ripetutamente le operazioni di sterilizzazione, un suo utilizzo pu ridurre la durata dell'operazione anche di due ore. Il sistema, per, consente solo scorrere immagini tomografiche in una serie, selezionarne una e mostrarla a video. Operazioni quali zoom, rotazione, modifica dei parametri di visualizzazione delle immagini, estrazione di regioni di interesse, possono essere quindi effettuate solo prima di inserire le immagini nel sistema.



Figura 3.5: Kinect usato per manipolare immagini virtuali in sala operatoria

UN SISTEMA PER LA VISUALIZZAZIONE INTERATTIVA É datato 2011 il primo sistema altamente interattivo per la visualizzazione touchless di immagini medicali in sala operatoria tramite Kinect. Tale sistema consente di eseguire molte delle operazioni comunemente effettuate dai medici nell'esplorazione di immagini digitali, quali il controllo della posizione di un cursore sullo schermo, la selezione di una regione di interesse di un'immagine e la sua estrazione (segmentazione manuale), selezione, zoom, traslazione e rotazione delle singole immagini nonché modifica dei parametri utilizzati per visualizzarle (ampiezza e livello). Un video esplicativo del sistema é visionabile sul sito iHealth lab, www.ihealthlab.icar.cnr.it. L'unico dispositivo di input utilizzato nel sistema é il Kinect. Tale dispositivo fornisce una mappa di profonditá della scena a una frequenza di 30 fotogrammi al secondo, con

una risoluzione, ad una distanza di 2 m, di 3 mm sul piano parallelo al sensore e di 10 mm sull'asse ortogonale a tale piano. La procedura di calibrazione del sistema, che richiede non più di 30 secondi, è necessaria per associare all'utente un modello scheletrico, in cui egli viene rappresentato come un insieme di snodi corrispondenti a testa, spalle, gomiti e da segmenti che li congiungono. Inoltre, nella fase di calibrazione il sistema esegue il calcolo della lunghezza del braccio dell'utente, l'identificazione della mano dominante, e calcola i parametri necessari alla messa a punto dei filtri utilizzati per smussare i movimenti dell'utente e attenuare l'effetto dei tremori delle mani sul sistema. L'interfaccia utente realizzata segue il paradigma evento-azione, ovvero le azioni interattive sono dipendenti non solo dall'evento in input, ma anche dallo stato del sistema. Ci vuol dire che uno stesso movimento della mano, può dar luogo a comandi differenti in base allo stato attuale dell'interfaccia. Nel sistema realizzato, tutte le azioni interattive sono associate a gesti dell'utente. Tali gesti hanno componenti sia statiche (l'utente assume una certa posa o configurazione di mani e braccia), sia dinamiche. In particolare, la postura delle mani è principalmente utilizzata per discriminare tra i possibili stati del sistema, mentre i movimenti spazio-temporali delle braccia sono utilizzati per eseguire azioni. Per consentire una transizione di stato deterministica tra gli stati dell'interfaccia e la specifica non ambigua di inizio e fine gesto, nello spazio e nel tempo, l'interfaccia si basa sul concetto di area di attivazione: il sistema controlla continuamente il modello scheletrico dell'utente; ogni qual volta egli stende un braccio per più del 55°.

Inoltre, il sistema consente di filtrare i movimenti di mani e braccia in modo da gestire dinamicamente il compromesso tra velocità di esecuzione e precisione nel controllo, introducendo un anisomorfismo tra i movimenti dell'utente e la risposta a video del sistema. Per farlo, viene considerata la velocità con cui i movimenti dell'utente vengono eseguiti. Difatti, quando la velocità di esecuzione è bassa, l'utente è probabilmente intenzionato ad avere maggiore precisione (specie nel puntamento), quindi il sistema automaticamente scalerà i movimenti eseguiti in modo da innalzare il livello di accuratezza. Oltre alla velocità, viene considerato anche il livello di tremore delle mani, stimato durante la fase di calibrazione, in modo da adattare ulteriormente il sistema alle caratteristiche individuali dell'utilizzatore.

SFIDE FUTURE Il sistema di visualizzazione interattiva touchless di immagini digitali medicali basato su Kinect è solo il primo passo verso l'obiettivo più ampio della progettazione di sistemi efficaci, efficienti e con un elevato livello di usabilità per le sale operatorie. Attività di ricerca future saranno finalizzate alla definizione di paradigmi di intera-

zione collaborativa con il sistema, sia con il personale presente in loco, sia con medici collegati in remoto. Verrá considerato il passaggio al 3D, per consentire la manipolazione diretta e non invasiva di ricostruzioni tridimensionali di parti anatomiche del paziente. Inoltre, saranno oggetto di studio strumenti che consentano una stima accurata della posizione spaziale, dell'orientamento e dell'articolazione delle mani in assenza di guanti e marcatori, in modo da consentire un'interazione quanto piú possibile precisa e intuitiva dell'utente con il sistema. In primis, é per fondamentale capire quali funzionalità, quali gesti, movimenti e quindi quali tecniche e metafore di interazione siano maggiormente compatibili con le diverse tipologie di interventi eseguiti in ambienti sterili. Solo la sperimentazione ospedaliera del sistema, appena iniziata, potrà dire quanto e come questa tecnologia potrà realmente entrare a far parte della pratica operatoria

Articolo a cura di Luigi Gallo Ricercatore ICAR-CNR, Alessio Pierluigi Placitelli Collaboratore di ricerca ICAR-CNR, Giuseppe De Pietro Responsabile sede di Napoli ICAR-CNR

3.4 Conclusioni

L'articolo sopra riportato ripropone molto bene lo scopo del progetto; la principale differenza sta nelle apparecchiature utilizzate, da una parte infatti viene adoperata una periferica di altissimo livello come il Kinect, mentre nel mio caso si tratta di una semplice webcam. Indubbiamente il primo approccio si rivelerá molto piú preciso e dotato di una gran quantità di gesture da poter attuare, svolgendo cos svariati comandi; questa superioritá per si ripercuote sul costo dell'apparecchio che é piuttosto elevato, impedendo cos che possa essere fruibile da una vasta gamma di utenza. Altre idee preliminari avevano considerato la possibilitá di usare una coppia di schermi touch-screen per consentire al dottore di navigare all'interno del programma e visualizzare le foto; purtroppo anche in questo caso il problema sono i costi, ma non solo; utilizzando il touch-screen il dottore andrebbe a sporcare gli schermi e dovrebbe in ogni caso effettuare nuovamente le operazioni di sterilizzazione. Dopo queste analisi siamo giunti alla conclusione che effettivamente l'utilizzo di una semplice webcam potesse essere la scelta ottimale; da un lato si avrá un tracking un po meno preciso e variegato per quel che riguarda le gesture, ma se ne guadagnerá in portabilitá e costi. Il fatto che non sia preciso quanto il

Kinect non compromette il funzionamento del programma dato che i comandi non sono molti e, in ogni caso, i gesti vengono interpretati quasi sempre nel modo corretto.

Capitolo 4

Il mio approccio

4.1 Introduzione

L'obiettivo del mio lavoro é dare la possibilitá di svolgere delle azioni, sullo strumento con cui si sta lavorando, con il solo spostamento, per esempio, di una mano davanti alla webcam. Nello specifico questo progetto é stato ideato per essere utilizzato da un dottore durante una visita dermatologica, durante la quale non ha la possibilitá di toccare mouse o tastiera, per scattare foto ai nevi, scorrere una galleria di immagini o fare altre operazioni. In realtá, questo progetto pu essere utilizzato in qualsiasi campo ne abbia bisogno dato che lascia libertá di associare qualsiasi comando alle gesture. Durante lo svolgimento del progetto, é stato testato collegandosi al cursore del mouse; ovvero, se il programma legge uno spostamento verso l'alto dei un qualsiasi oggetto nel flusso di immagini catturate, il cursore si sposterá anch'esso verso l'alto proporzionalmente all'intensitá dello spostamento dell'oggetto. Il metodo che ho utilizzato per implementare questo sistema di tracking é basato sulla minimizzazione della differenza assoluta; in questo metodo per ogni pixel in posizione (x,y) della prima immagine viene creato un quadrato $L \times L$ attorno al punto di cui dobbiamo determinare lo spostamento, ottenendo quindi una matrice $A_{x,y}$ $L \times L$ che avrá per elementi i rispettivi valori RGB dei pixel; per ricavare il vettore di spostamento del pixel (x,y) , sará necessario minimizzare la differenza dei rispettivi elementi della matrice $A_{x,y}$ con le varie matrici $B_{x+u,y+v}$ ottenute centrando la matrice nel pixel $(x+u,y+v)$ della seconda immagine; in questo modo si otterrá un vettore di spostamento $s=(u,v)$ che parte dal pixel (x,y) . Questo procedimento deve essere ripetuto per tutti

i pixel dell'immagine, ottenendo quindi una matrice che ha per elementi i vettori di spostamento delle due immagini. Si tratta quindi di un real-time tracking, dato che l'utente ha la necessità di avere una risposta immediata al suo gesto; questo tipo di tracking, che non si lega a nessun oggetto in particolare dell'immagine catturata, richiede una completa mappatura dei pixel da un'immagine alla successiva per creare una griglia di vettori di spostamento, operazione molto dispendiosa e lunga per un calcolatore; la priorità del progetto, per, rimane il real-time, per questo motivo verranno usate delle semplificazioni al calcolo dei vettori per rendere più veloce il calcolo del flusso degli oggetti.

4.2 Algoritmo

Dopo questa introduzione generale, verrà ora spiegato l'algoritmo di ricerca della matrice di spostamento; si possono definire 4 passi principali di questo metodo:

4.2.1 Cattura delle immagini

Il processo di confronto viene effettuato fra 2 immagini successive, che quindi vengono catturate ed analizzate, per poi passare al confronto successivo. Si crea per subito un problema, questo programma non deve essere fine a sé stesso, per cui dovrà lavorare in background ed essere funzionale ad un altro programma che dovrà avere la precedenza, per questo motivo è stato creato un thread parallelo per contenere il tracking; l'idea di partenza era catturare le immagini con una frequenza di campionamento prefissata e quindi svolgere il confronto fra I_1 e I_2 , poi fra I_2 e I_3 ; purtroppo questa ipotesi è da scartare, perché se l'esecuzione del programma principale impiega troppe risorse, il confronto verrà eseguito con troppo ritardo, risultando inutile e controproducente, l'obiettivo rimane il real-time, non è accettabile che il comando venga eseguito anche solo 2-3 secondi in ritardo. Se invece proviamo a catturare un'immagine ogni volta che il confronto è finito, il problema si sposta sulla precisione: se il confronto impiega troppo tempo per essere completato, le due immagini successive potrebbero essere molto diverse fra di loro e rendere il tracking impossibile.

Ho quindi creato un ulteriore thread che cattura immagini con una frequenza preimpostata, ma ne tiene in memoria solo le ultime 2 catturate; non

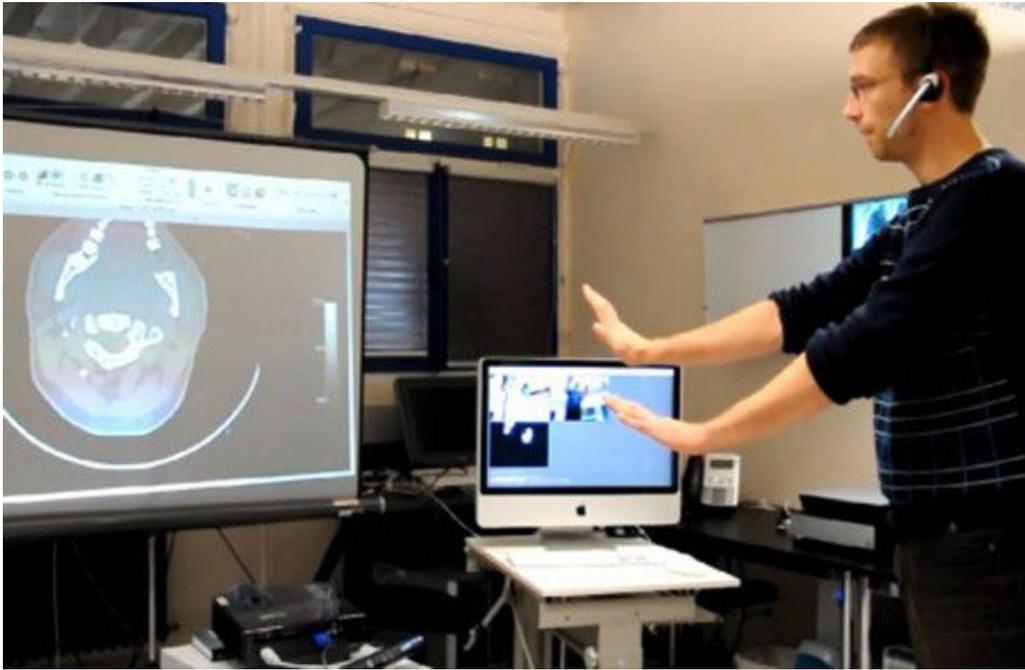


Figura 4.1: Le immagini vengono catturate dalla webcam

appena il confronto precedente sarà completato, il programma andrà a prelevare queste 2 immagini e le confronterà. Questo è a mio avviso il metodo migliore per diversi motivi: le immagini catturate avranno una distanza temporale piccola a piacere, risultando molto simili fra di loro, fatto che renderà il tracking il più preciso possibile; il fatto che le due immagini siano molto simili permette di attuare una semplificazione atta al risparmio del tempo di ricerca del vettore di spostamento, questo infatti non potrà avere una norma molto grande, dato che le due immagini sono molto simili, permettendo di fare una ricerca di questo vettore più mirata e veloce.

4.2.2 Riduzione della dimensione delle immagini e conversione a scala di grigi

Le immagini catturate da webcam sono nel formato 640x480, in genere. Questa dimensione è eccessiva e non necessaria per il semplice e veloce calcolo del flusso dell'immagine che serve a noi; è stata quindi ridotta a 320x240, inoltre è stata convertita a scala di grigi, permettendo un notevole risparmio di

memoria e avendone un guadagno prestazionale, trattandosi di soli 8 bit per pixel invece che 32bit; nonostante questa semplificazione il tracking rimane piuttosto preciso ma ne guadagna in velocità.

4.2.3 Calcolo del flusso ottico

In questo momento abbiamo a disposizione le due immagini, di grandezza 320x240 pixel, quest'ultimi di 8 bit; preso un pixel P nella posizione (x,y), creiamo la matrice A di grandezza LxL, con L dispari, attorno a P:

$$\mathcal{A}_{x,y} = \begin{pmatrix} a_{x-l,y-l} & \dots & \dots & \dots & a_{x+l,y-l} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & a_{x,y} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{x-l,y+l} & \dots & \dots & \dots & a_{x+l,y+l} \end{pmatrix}$$

dove $l=(L-1)/2$ e ogni $a_{i,j}$ é un numero compreso tra -127 e 128 che rappresenta il colore del pixel nella scala di grigi. Nella seconda immagine andremo a creare matrici di dimensione identica centrate nei pixel attorno al punto (x,y) che chiameremo $\mathcal{A}'_{x+u_i,y+v_j}$, con (u_i, v_j) che varia; l'obiettivo é trovare la matrice che meno si differenzia da quella iniziale, ovvero quella centrata nella posizione in cui si é spostato il pixel dalla prima alla seconda immagine. Con due semplici cicli *for* possiamo eseguire l'operazione di ricerca della matrice che minimizza la differenza:

$$\mathcal{A}_{x,y} - \mathcal{A}'_{x+u'_i,y+v'_j} = \begin{pmatrix} a_{u'_i-l,v'_j-l} & \dots & \dots & \dots & a_{u'_i+l,v'_j-l} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & a_{u'_i,v'_j} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{u'_i-l,v'_j+l} & \dots & \dots & \dots & a_{u'_i+l,v'_j+l} \end{pmatrix} = \mathcal{B}_{u'_i,v'_j}$$

dove (u'_i, v'_j) sará il vettore di spostamento trovato con la minimizzazione della somma degli elementi della matrice \mathcal{B}_{u_i,v_j} . Questo procedimento di ricerca del vettore di spostamento va quindi ripetuto per ogni pixel dell'immagine; é evidente ke non si potrà fare questa ricerca per tutti i pixel, ma degli accorgimenti ne parleremo presto. Come operazione conclusiva si deve ricavare il vero e proprio vettore di spostamento:

$$u = \frac{1}{N} \sum_{\forall pixel} u_n$$

$$v = \frac{1}{N} \sum_{\forall pixel} v_m$$

dove $(u, v) \equiv \vec{s}$, ovvero il vettore spostamento, mentre N é il numero di vettori calcolati nell'immagine. Alla fine di questo procedimento abbiamo il vettore di spostamento \vec{s} che indica la direzione e l'intensità del movimento rilevato.

Ho qui di seguito riportato alcune immagini scelte per mostrare il funzionamento del programma; le prime due sono le immagini di cui dobbiamo calcolare il flusso ottico: Con una piccola aggiunta al codice, ho salvato la



Figura 4.2: La coppia di immagini da analizzare

matrice dei vettori che risulta dall'elaborazione di queste immagini; con il comando Quiver di Matlab ho quindi stampato a video il campo di vettori che ne risulterebbe: Intrecciando l'immagine di partenza e il campo dei vettori si può comprendere in modo più intuitivo cosa si é spostato nell'immagine: Come si può subito notare il programma ha rilevato il movimento dei pixel della mano, in particolare dei bordi, dato che risultano essere maggiormente riconoscibili, grazie al contrasto con il fondo dell'immagine. La media dei vettori generati dará perci il vettore da noi voluto.

4.2.4 Eseguire l'operazione richiesta

In questo momento il programma principale riceve i dati necessari per eseguire i comandi richiesti; dato che il tracking lavora in background, non ne compromette le prestazioni. Nel progetto svolto l'operazione richiesta é di

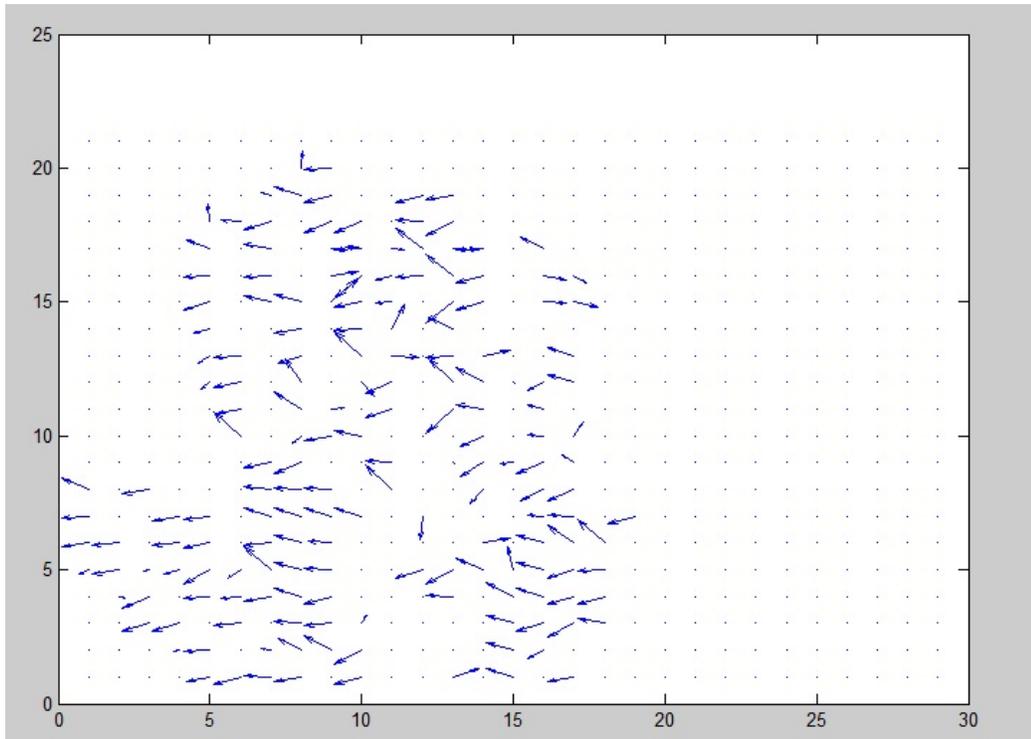


Figura 4.3: Campo dei vettori stampato con l'uso di Matlab

muovere il cursore del mouse, ma in sala operatoria sar quella di scorrere la galleria dei foto del nevo selezionato, ingrandire l'immagine e analizzare il nevo con gli strumenti che il programma fornisce, oppure tornare alla pagina di selezione del nevo. Come abbiamo letto negli articoli proposti, piú movimenti puó leggere la periferica utilizzata, maggiori sono i possibili comandi che il medico potrà svolgere.

4.3 Euristiche

É fin troppo evidente che il tracking non potrà essere eseguito su tutti i pixel dell'immagine perché, seppur ridimensionata, rimane troppo grande, contando $320 \times 240 = 76800$ pixel; questo comprometterebbe la natura real-time del progetto. Sono quindi stati inseriti alcuni accorgimenti per semplificare e rendere piú veloce il programma di tracking:



Figura 4.4: Sovrapposizione dell'immagine 2 con il campo dei vettori

4.3.1 Ridurre la grandezza della matrice \mathcal{A}

\mathcal{A} , che ricordiamo essere la matrice contenente i valori dei pixel nella scala di grigi attorno al punto di cui dobbiamo studiare il movimento, ha dimensione $L \times L$; per rendere il procedimento preciso si dovrebbe scegliere L abbastanza grande, ma per semplificare il calcolo non sono andato oltre $L=7$; questo in realtà pu essere un vantaggio anche dal punto di vista della precisione, dato che, se il punto trattato si muove, non è detto che la matrice rimanga inalterata nella seconda immagine, soprattutto per punti distanti fra loro; scegliere quindi una matrice più piccola è risultato un vantaggio.

4.3.2 Limitare l'area di ricerca

Il fatto di aver catturato due immagini in un lasso di tempo molto breve le rende molto simili fra loro, questo implica che, se un oggetto si è mosso in questo breve tempo, la sua posizione sarà simile fra la prima e la seconda

immagine. Facendo questa ipotesi, possiamo impostare un raggio massimo di ricerca della matrice che minimizza la differenza; il vettore (u,v) non potrà quindi essere maggiore di R , con R il suddetto raggio massimo di ricerca. In questo modo, si risparmi moltissimo dato che la maggior parte dei punti rimarrá fissa, dato che a muoversi di solito é un solo oggetto, ovvero una minima parte dell'immagine.

4.3.3 Intervalli fra i punti da analizzare

In questo caso, andremo ad analizzare e valutare lo spostamento solo di un punto ogni T . A primo impatto pu sembrare una semplificazione troppo drastica, perché tralascia moltissimi punti; in realtà anche questa approssimazione funziona piuttosto bene, dato che non influenza molto il calcolo complessivo del flusso; prestando attenzione a questo calcolo:

$$u = \frac{1}{N} \sum_{\forall pixel} u_n$$

$$v = \frac{1}{N} \sum_{\forall pixel} v_m$$

si capisce che, se diminuisce il valore della sommatoria, diminuisce allo stesso modo anche il valore di N . Questo implica che il risultato finale non é influenzato dalla diminuzione controllata dei pixel analizzati.

4.3.4 Filtro gaussiano sulla matrice \mathcal{C}

Consideriamo ora la matrice cos creata:

$$\mathcal{C}_{x,y} = \begin{pmatrix} b_{1,1} & \dots & b_{R,1} \\ \dots & \dots & \dots \\ b_{1,R} & \dots & b_{R,R} \end{pmatrix}$$

dove $b_{u,v} = \sum_{i=1}^{i=L} \sum_{j=1}^{j=L} \mathcal{B}_{u_i,v_j}$; ricordiamo che la matrice $\mathcal{B}_{u,v}$ é ricavata facendo la differenza fra $\mathcal{A}_{x,y}$ e $\mathcal{A}'_{x+u,y+v}$. Sostanzialmente gli elementi della matrice $\mathcal{C}_{x,y}$ simboleggiano quanto l'intorno del punto (u,v) é simile a quello di (x,y) ; l'algoritmo cerca quindi il valore piú basso fra i $b_{u,v}$ e restituisce il vettore di movimento. Pu per capitare che ci sia un b molto piccoli circondato da

elementi invece molto alti, il che significa che probabilmente non era ci che cercavamo. Ho quindi inserito un Filtro Gaussiano alla matrice \mathcal{C} per evitare l'insorgere di questi errori; in questo modo questi picchi vengono compensati dai valori che stanno attorno permettendo di avere una matrice piú omogenea e verosimile.

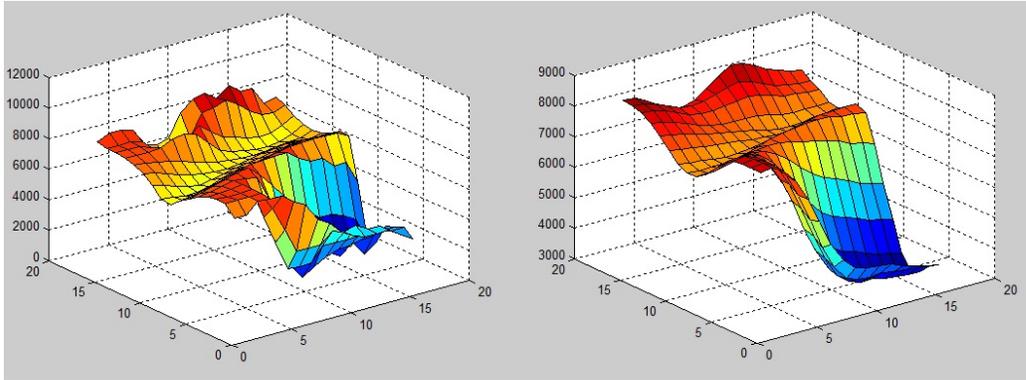


Figura 4.5: Le due matrici messe a confronto prima e dopo il filtro, con l'uso di MatLab

Capitolo 5

Risultati

Al fine di valutare la funzionalità del progetto, non potendo effettuare dei test matematici, ho deciso di farlo utilizzare direttamente a delle persone che si sono offerte di testarlo. La prova consiste nello spostare il cursore del mouse all'intero di un piccolo riquadro con l'utilizzo dell'algoritmo, quindi solo con lo spostamento della mano di fronte alla webcam, e valutare il tempo impiegato, chiedendo poi un feedback, ovvero se l'utente ha trovato l'operazione facile, intuitiva, comoda, ecc.. Qui di seguito é riportata una tabella che associa ad ognuno dei 4 utenti che hanno testato la funzionalità del programma, se é riuscito a svolgere l'operazione di puntamento e il tempo che é stato necessario:

| | Operazione riuscita | Tempo impiegato |
|-----------------|---------------------|-----------------|
| Utente 1 | Si | 4.5 sec |
| Utente 2 | Si | 4 sec |
| Utente 3 | Si | 6 sec |
| Utente 4 | Si | 4 sec |

Come si pu notare, nessuno degli utenti ha riscontrato difficoltà nel portare a termine l'operazione richiesta, mentre i tempi vanno da 4 a 6 secondi; questo test é piuttosto esigente nei confronti del programma, dato che non dovrà utilizzare il cursore del mouse nelle normali mansioni richieste dal dottore, ma semplicemente rilevare se vi é stato uno spostamento, in quale direzione e con quale intensità. Il fatto che riesca ad essere cos preciso da poter muovere

anche il cursore in un piccolo riquadro si ripercuote sui tempi; l'utilizzo di un mouse avrebbe permesso delle prestazioni migliori, ma non é lo scopo del progetto. Dal punto di vista del feedback, gli utenti hanno espresso giudizi positivi soprattutto sulla buona risposta del cursore al loro movimento, piuttosto veloce e pronta; non é per semplice filtrare perfettamente il movimento, perché la mano si muove verso un lato ben preciso, ma l'avambraccio farà spesso un naturale movimento in diagonale e questo pu impedire una perfetta lettura del movimento voluto. Questo fatto per non ha influito molto sull'esperimento che é riuscito infatti con tutti gli utenti sottoposti al test.

Bibliografia

- [1] H. Haussecker and D. J. Fleet. *Estimating optical flow with physical models of brightness variation. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):661673, 2001 .
- [2] http://www.cmb.ac.lk/?page_id=3580
- [3] <http://it.ibtimes.com/articles/29432/20120511/kinect-xbox-360-microsoft-autismo-infanzia-minnesota-periferica-diagnosticare.htm>
- [4] <http://www.mobimed.it/una-xbox-kinect-per-visualizzare-immagini-radiologiche-in-sala-operatoria/>
- [5] http://it.wikipedia.org/wiki/Microsoft_Kinect