Relazione Progetto Sistemi Intelligenti
anno 2012/2013

Riconoscimento e proiezione di oggetti su un ambiente virtuale

Miguel Rosales, Laura Musica

---

1. Introduzione

Lo sviluppo della tecnologia Microsoft kinect ha permesso l'introduzione di interfacce NUI(natural user interface) dove l'utente interagisce con programmi/giochi soltanto attraverso gesti naturali. Le librerie kinect SDK allo stato del arte permettono infatti di tracciare i movimenti del corpo umano e utilizzarli per animare avatar in 2D o 3D.

Nel nostro progetto abbiamo voluto studiare l'interazione con un ambiente virtuale dove l'utente fa uso non solo di gesti del corpo ma anche di oggetti per interagire con l'ambiente. In particolare abbiamo implementato un algoritmo che permette il riconoscimento di un particolare oggetto nel mondo reale e l'analisi del suo movimento ai fini di proiettarlo in un oggetto virtuale.

Nel corso di questo documento descriveremo che tecnica abbiamo usato per riconoscere e tracciare il movimento dell'oggetto per poi trasferirlo e animarlo sull'ambiente virtuale.

2. Riconoscimento di un oggetto colorato su immagine RGB

Esistono diverse tecniche per riconoscere oggetti, basate sull'analisi del colore, sulla forma e sulla posizione dell'oggetto all'interno della scena. Le librerie SDK di kinect, in particolare, mettono a disposizione degli sviluppatori tre differenti flussi di dati. Il flusso *colorImageStream* fornisce in tempo reale le immagini riprese dalla camera RGB. Il flusso *depthStream* (profondità) definisce la distanza dalla camera

di un qualsiasi punto presente nella scena, permettendo così di avere un sistema di riferimento tridimensionale. Infine il flusso *skeletonStream* riporta le informazioni riguardanti la posizione dello scheletro del soggetto preso in considerazione.

Nel nostro lavoro abbiamo deciso di utilizzare lo stream RGB da cui estrarre le informazioni riguardanti il colore e l'orientamento dell'oggetto (fig1) che si trova in mano all'utente. Tramite la camera RGB si ha accesso ad uno stream di 30 frame per secondo con una risoluzione di 640 x 480 pixels. Abbiamo quindi creato un algoritmo che, per ogni frame ricevuto, analizza in tempo reale i dati in esso contenuti per rintracciare la posizione dell'oggetto e la sua inclinazione.

Fig 1. Oggetto colorato

Nella fase di preprocessing l'immagine ottenuta dallo stream della kinect (fig.2a) viene passata attraverso un filtro, per eliminare l'eventuale rumore dovuto all' imprecisione del sensore. L'operazione di filtraggio è stata effettuata tramite filtro mediano, con una finestra di grandezza 5x5, che sostituisce ad ogni pixel il valore mediano(di colore) dei suoi vicini (fig.2b).

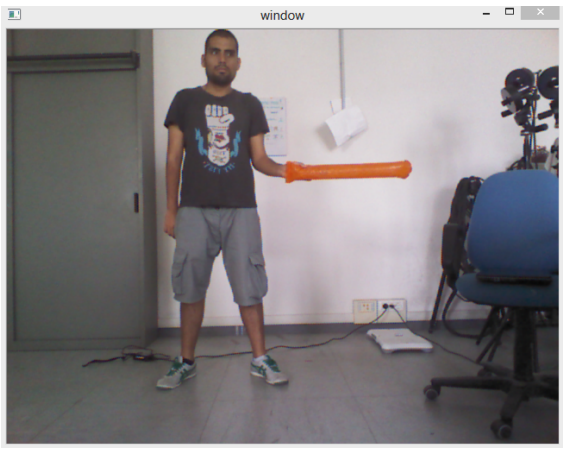

Fig 2a) immagine rgb originale

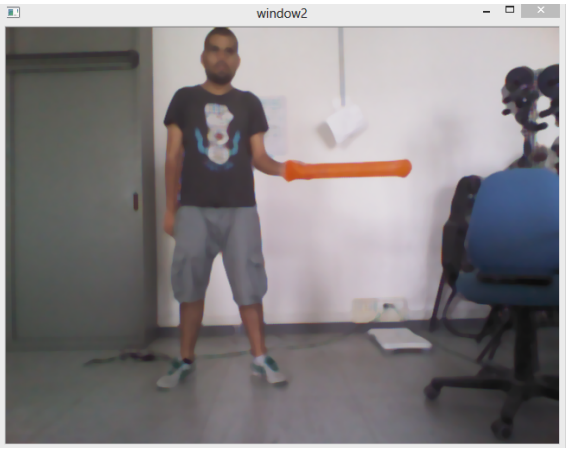

Fig 2b) applicazione filtro mediano

Per distinguere l'oggetto dallo sfondo sulla base del colore, lo spazio Rgb non è il più indicato, perché sensibile alle variazioni di luminosità. Esistono altri spazi

colore, come YCrCb e Hsv, che permettono di separare l'informazione strettamente legata al colore dalla componente legata all'intensità dell'immagine e alla sua luminosità. Il primo dei due separa la componente di luminanza Y dalle componenti di crominanza Cb (blu) e Cr (rosso). Il secondo, invece, definisce i colori attraverso i tre parametri tinta(hue) H,saturazione S e valore V.

Per il progetto abbiamo scelto di usare YCbCr e abbiamo definito un range di valori entro cui si trova il colore dell'oggetto da ricercare (min=(0,60, 170), max=(250,130,255)). È stata generata quindi una maschera, dove solo i pixel il cui colore è compreso nel range sono bianchi, mentre tutti gli altri sono impostati a nero (fig.3b).

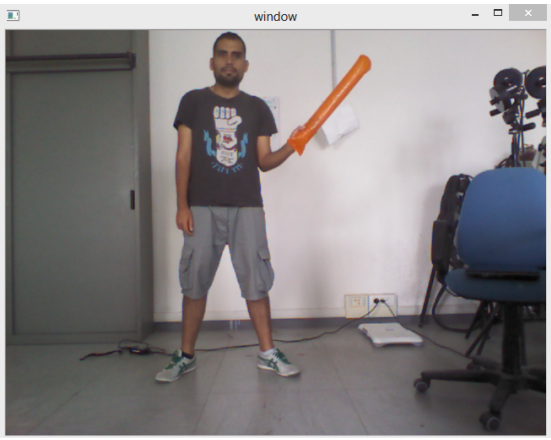

Fig 3a. immagine rgb originale

Fig 3b. maschera bianco-nero

3. Tracking dell'oggetto in movimento

A causa della limitazione del sensore rgb, abbiamo deciso di considerare il movimento dell'oggetto(Fig 1) solo sul piano frontale, in due dimensioni. Per poter proiettare questo movimento sulla scena tridimensionale è necessario quindi determinare in ogni frame un angolo di rotazione, formato dall' asse principale dell'oggetto con l'asse orizzontale.

Una volta ottenuta dallo stream rgb l'immagine in bianco e nero, in cui l'oggetto è separato dal resto della scena, vengono calcolati i contorni di tutti le regioni chiare che commpaiono nell'immagine. Per eliminare il rumore residuo, viene poi considerato solo il contorno di lunghezza massima, che corrisponde con alta probabilità a quello dell'oggetto. A questo punto è possibile approssimare l'angolo cercato tramite l'inclinazione del rettangolo di area minima che racchiude il contorno.

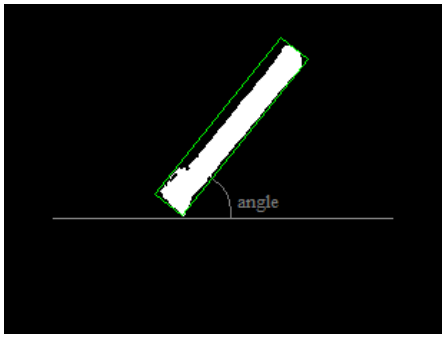

Fig 4. angolo di rotazione

Per permettere un movimento a 360 gradi, questo angolo deve però essere calcolato rispetto ad un sistema di riferimento che ha come origine la mano dell'utente, che corrisponde al centro della rotazione. Per fare ciò sono stati pertanto utilizzati i dati contenuti nello stream dello scheletro per ottenere le coordinate del punto corrispondente alla mano.

Ai fini del calcolo, possiamo considerare al posto dell' intero rettangolo il solo segmento che corrisponde ad uno dei suoi lati più lunghi. L'angolo rispetto all'asse orizzontale rimane così invariato, ma in più è possibile identificare i due punti estremi della figura. Il punto tra questi che si trova a distanza minima dalla mano sarà l'estremo dell'oggetto che è in mano all'utente e di conseguenza il centro di rotazione.

nota: durante i test abbiamo notato una certa difficoltà da parte dell'algoritmo di tracking nel riconoscere la mano. è capitato infatti che essa venisse rilevata erroneamente sull'estremità superiore dell'oggetto come se l'oggetto stesso fosse un estensione del braccio. Per risolvere questo problema si è deciso di utilizzare il gomito anzichè la mano come punto di riferimento per calcolare le distanze e definire il centro di rotazione. Questo ha escluso le rilevazioni errate da parte del kinect.

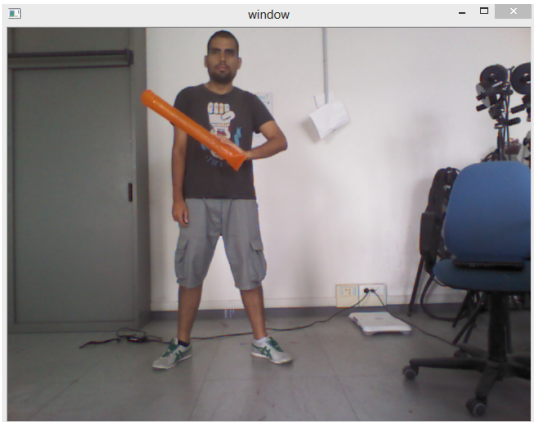

Fig 5a. immagine originale

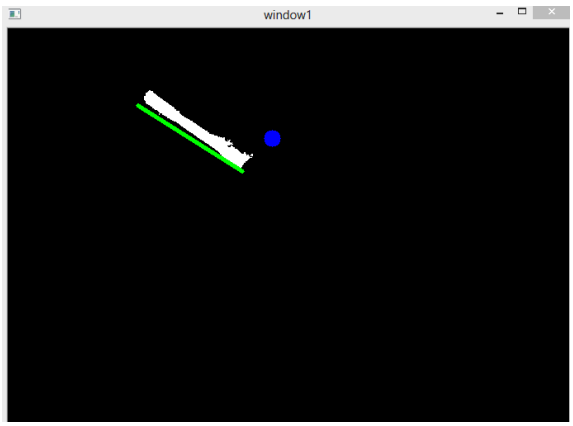

Fig 5b. lato del rettangolo(verde) e gomito(blu)

4. Ambiente di sviluppo

Il modulo da noi sviluppato si integra con l'engine grafico del laboratorio AISLab, per la creazione dei giochi di riabilitazione basati sull'interazione con un ambiente virtuale. L'engine si interfaccia al motore grafico opensource panda3D per la creazione dell'avatar e della scena virtuale e alla sdk di kinect per il tracking dello scheletro dell'utente. Noi abbiamo scelto come linguaggio di sviluppo python perche più semplice da integrare con i moduli già esistenti, mentre per la parte di image processing ci siamo appoggiati alle librerie di opencv, per avere prestazioni più ottimizzate. Il risultato finale del lavoro è visibile nelle immagini sottostanti. L'avatar si muove seguendo i movimenti dell'utente grazie al tracking dello scheletro mentre la spada parentata alla mano si muove secondo la rotazione da noi calcolata, che rispecchia quella dell'oggetto.

5. Possibili sviluppi e miglioramenti futuri

In questo progetto è stato studiato l'utilizzo del colore per il riconoscimento di oggetti in una scena. I risultati dati da questo tipo di analisi non sono però sempre robusti a causa delle variazioni di luminosità nella stanza e alla difficoltà nel individuare un colore che si distingua nettamente dagli altri colori della scena. Il nostro metodo si è rilevato particolarmente efficace su una scena dove i toni prevalenti sono dati da colori freddi. Se nella scena compaiono elementi di colore simile a quello dell'oggetto utilizzato (fig.1), l'algoritmo può dare risultati non accurati.

Un possibile miglioramento potrebbe essere quello di tenere traccia della proffondità a cui si trova l'oggetto in modo da distinguerlo meglio dallo sfondo oppure utilizzare un'oggetto luminoso. Un'altro approccio potrebbe essere quello di considerare oltre al colore la forma dell'oggetto .