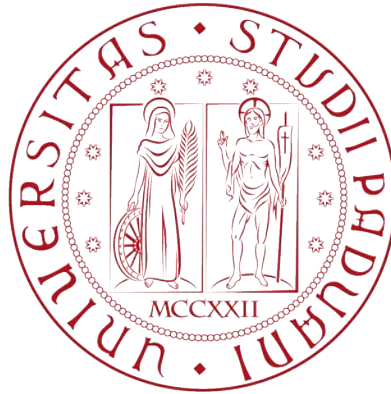


UNIVERSITÀ DEGLI STUDI DI PADOVA



Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Bioingegneria

Anno Accademico 2013 - 2014

Algoritmi per l'elaborazione dei dati video e tracking di features in analisi del movimento

Relatore : Prof.ssa Chiara DALLA MAN

Correlatore : Ing. Zimi SAWACHA Ph.D.

Laureando : Alberto DEMO

Matricola 1035980

Padova, 9 Dicembre 2013

Chi non ha mai sbagliato non ha mai fatto nulla.

[Robert Baden – Powell]

Abstract

L'analisi del movimento è uno strumento fondamentale per lo studio e l'analisi di gesti motori, e per essere applicata necessita di tecnologie di *motion capture*. La grande maggioranza dei sistemi di *motion capture* è basata sull'utilizzo di *marker*, tuttavia sia in ambito clinico che sportivo si sente la necessità di svincolarsi dall'ambiente del laboratorio per poter lavorare direttamente sul campo. In quest'ottica stanno prendendo piede tecnologie basate su telecamere standard o ad alta velocità, nelle quali la ricostruzione del movimento avviene tramite una funzione di *tracking*, manuale o automatico, di *feature* definite direttamente sulle immagini video. I software commerciali di *tracking*, ora presenti sul mercato, non permettono però all'operatore di conoscere gli algoritmi che stanno alla base del processo di inseguimento.

Obiettivo del presente lavoro di tesi è stato lo sviluppo di un *toolbox open source*, utilizzabile nella suite di calcolo MatLab, che permetta la gestione e il processing di dati video e la creazione finale di file in formato C3D (formato standard per l'archiviazione e la gestione di dati di analisi del movimento); la ricostruzione del gesto motorio è stata effettuata con un *feature tracker* semi - automatico, sviluppato nel laboratorio di Analisi del Movimento del Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova.

Al fine di valutare l'accuratezza della ricostruzione delle traiettorie dei *marker* relativamente al *tool* sviluppato, si sono confrontate le traiettorie dei *marker*, ottenute utilizzando il sopracitato *toolbox*, e le medesime calcolate con un sistema stereofotogrammetrico. Per ottimizzare le funzionalità del *toolbox*, si è quindi confrontata la stima delle coordinate di un set di *marker* (nonché degli angoli articolari stimati a partire dalle traiettorie dei *marker* stessi) ricostruite a partire da video distorti e non distorti; nel primo caso la distorsione veniva rimossa nel processo di triangolazione, nel secondo la distorsione veniva rimossa prima del *tracking*.

Indice

Introduzione	1
1 Tecnologie di Motion Capture	3
1.1 Sistemi Commerciali di Motion Capture	3
1.1.1 Sistemi Optoelettronici	4
1.1.1.1 Sistemi Stereofotogrammetrici	5
1.1.1.2 Sistemi Video e Feature Tracking	10
1.2 Formato File C3D	10
1.2.1 Introduzione e Storia	11
1.2.2 Descrizione del file	12
1.2.3 Limitazioni	14
2 Algoritmi per il Tracking di Features	15
2.1 Video Tracking in Analisi del Movimento	17
2.1.1 Calibrazione delle Telecamere	18
2.1.2 Feature Tracking	24
2.1.2.1 Optical Flow	25
2.1.2.2 Metodo di Lucas – Kanade e Algoritmo KLT	26
2.1.2.3 Software Commerciali: Simi Motion	31
2.1.3 Triangolazione	32
2.1.4 Errori Strumentali e Compensazione	36
2.1.4.1 MAL Test	37
3 Materiali e Metodi	41
3.1 Set up Sperimentale	43
3.1.1 MAL Test	43
3.1.2 Acquisizioni Subacquee	46
3.1.3 Acquisizioni a Secco	47

3.1.4	Video Processing	49
3.1.5	Calibrazione Telecamere	50
3.2	Rimozione della Distorsione e Feature Tracking	52
3.2.1	Correzione della Distorsione	52
3.2.2	Implementazione del Feature Tracker	52
3.2.3	Triangolazione	53
3.3	Algoritmo per la Scrittura dei Dati Triangolati in Formato File C3D	54
3.4	Elaborazione dati MAL Test	55
3.5	Elaborazione dati da Video Subacquei	55
3.5.1	Calcolo degli Angoli Articolari	58
3.6	Elaborazione dati da Video a Secco	59
4	Risultati e Discussione	61
4.1	Elaborazione dati MAL Test	61
4.2	Elaborazione dati da Video Subacquei	63
4.3	Elaborazione dati da Video a Secco	72
4.4	Discussione dei Risultati	73
4.5	Conclusioni	74
	Bibliografia	77

Introduzione

Alla fine del '800, l'uomo d'affari e governatore della California Leland Stanford, chiese all'appassionato di fotografia, e libraio, Edward James Muybridge, di verificare una sua ipotesi, ovvero che durante il galoppo di un cavallo, ci fosse un momento in cui tutte le quattro zampe dell'animale fossero sollevate da terra. Nel 1878 Muybridge esaudì la richiesta e confutò l'ipotesi di Stanford, fotografando un cavallo in corsa con l'ausilio di 50 fotocamere, azionate da fili colpiti dalle zampe del cavallo, disposte parallelamente al tracciato percorso. Alcuni considerano questo episodio come la nascita dell'analisi del movimento. Al giorno d'oggi l'analisi del movimento è un ambito di ricerca multidisciplinare che ha lo scopo di misurare e quantificare un gesto motorio, sia esso semplice o complesso, per poterne estrapolare variabili cinematiche e dinamiche; le sue applicazioni spaziano dall'attività clinica a quella sportiva, dall'attività cinematografica a quella ludica. In ambito clinico ad esempio la *gait analysis* (analisi del cammino) non può prescindere dall'informazione temporale di posizione e orientamento delle articolazioni del paziente, per la determinazione di eventuali stati patologici. Mentre In ambito sportivo, l'analisi cinematica di specifici gesti motori fornisce informazioni importanti ad atleti e allenatori nella valutazione delle performance tecniche durante le competizioni, e nell'ottimizzazione delle sessioni d'allenamento.

Per raggiungere gli scopi preposti, l'analisi del movimento si avvale delle tecniche di *motion capture*, ovvero l'insieme di processi impiegati per l'acquisizione e la registrazione di un movimento, utili poi alla creazione di un modello approssimato del soggetto in esame.

La maggior parte dei metodi per l'analisi cinematica è *visual - based* e basata su marcatori (*marker*) attivi o passivi ancorati al corpo del soggetto in esame, tramite i quali è possibile l'inseguimento (*tracking*) di specifici punti di riferimento anatomici.

In tal senso i sistemi più frequentemente usati sono quelli stereofotogrammetrici, basati sull'utilizzo di *marker* passivi e telecamere a infrarosso. Tali sistemi sono però poco utilizzabili in ambienti diversi dal laboratorio, motivo per il quale c'è grande attenzione all'evoluzione dei sistemi puramente video, che saranno l'oggetto di discussione del presente lavoro di tesi.

In particolare, il presente capitolo si prefigge di dare un'introduzione generale delle tecnologie utilizzate in questa tesi (*Motion Capture*, Sistemi Stereofotogrammetrici, *Feature Tracking*, File C3D).

Nel Capitolo 2 vengono trattati più nello specifico la procedura di calibrazione delle telecamere, gli algoritmi di *Feature Tracking* e il processo di triangolazione, parzialmente anticipati nel presente capitolo. Vengono successivamente presentati gli errori che si possono riscontrare nell'analisi e i metodi per la compensazione utilizzati.

Nel Capitolo 3 si espongono materiali e metodi utilizzati per lo svolgimento della parte sperimentale del presente lavoro di tesi.

Infine il Capitolo 4 raccoglie i risultati dei test effettuati e le corrispondenti discussioni.

Obiettivo del presente lavoro di tesi è stato lo sviluppo di un toolbox open source, utilizzabile nella suite di calcolo MatLab, che permetta la gestione e il processing di dati video e la creazione finale di file in formato C3D (formato standard per l'archiviazione e la gestione di dati di analisi del movimento); la ricostruzione del gesto motorio è stata effettuata con un *feature tracker* semi-automatico, sviluppato nel laboratorio di Analisi del Movimento del Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova.

Per lo sviluppo di questa tesi si sono utilizzate le versioni di MatLab R2013a per UNIX e MatLab R2012b per Windows.

Capitolo 1

Tecnologie di Motion Capture

1.1 – Sistemi Commerciali di Motion Capture

Nell'ambito dell'analisi del movimento sono stati sviluppati molteplici strumenti per l'acquisizione del moto, tutti estremamente eterogenei per tipologia di funzionamento, dati raccolti e grado di invasività nel gesto motorio da studiare.

Tra i vari si distinguono:

- *Sistemi elettromeccanici (elettrogoniometri, accelerometri, sensori potenziometrici):* forniscono una misura diretta delle variabili di interesse impiegando dispositivi elettrici in diretto contatto con la pelle. Tale configurazione fa sì che i dati raccolti siano accurati e che il volume di acquisizione sia virtualmente illimitato; tuttavia, dato l'elevato ingombro della strumentazione, tali sistemi tendono a ridurre la naturalità del movimento [13];
- *Sistemi elettromagnetici:* funzionano tramite sensori elettromagnetici, posizionati in corrispondenza delle articolazioni, e un generatore di campo magnetico esterno, in grado di rendere rilevabili posizione e orientamento dei sensori; sono estremamente sensibili alla presenza di oggetti ferromagnetici, che possono modificare pesantemente la misura;
- *Sistemi basati su sensori acustici:* tramite rilevatori fissi, solidali con il laboratorio, e sorgenti, poste direttamente sul soggetto, forniscono una stima indiretta delle variabili cinematiche di interesse. Sono caratterizzati da una buona accuratezza nella localizzazione spaziale, ma risentono di

eco, problemi di interferenza e incostanza della velocità del suono nell'aria.

- *Sistemi optoelettronici*: monitorano il movimento tramite una serie di telecamere sincronizzate, posizionate in punti strategici. Forniscono solo una stima delle variabili cinematiche, non una misura diretta, a partire dalle immagini acquisite e dai marcatori posizionati sul soggetto [14]. Ad oggi sono i sistemi maggiormente utilizzati in letteratura [15].
- *Sistemi markerless*: sono sistemi basati su modelli, nati da ricerche nell'ambito della *computer vision*. Il grande vantaggio che hanno è che non necessitano di indumenti o strumenti particolari da indossare, quindi assicurano assoluta naturalezza del movimento, oltre che tempi molto ridotti di preparazione del soggetto. Tuttavia, pur non risentendo degli artefatti presenti con l'applicazione dei *marker*, non sono ancora giunti ad un livello di precisione tale da essere comparabile con i sistemi *marker-based*.

Ai fini della tesi verranno approfonditi gli aspetti riguardanti i sistemi optoelettronici stereofotogrammetrici con *marker* passivi e, ancor più nello specifico, i sistemi video basati su *feature tracking*.

1.1.1 - Sistemi Optoelettronici

Come già anticipato, questi sistemi si basano sull'utilizzo di *marker* posizionati sulla cute del soggetto, in zone che rappresentino al meglio le azioni articolari. Nonostante costituiscano la maggioranza dei sistemi utilizzati, non sono esenti da restrizioni, che bisogna considerare.

Innanzitutto, per ricostruire la posizione di un *marker* nello spazio, è necessario che questo venga rilevato contemporaneamente da almeno due telecamere. Nell'applicazione di un gesto motorio complesso questo non è sempre possibile: per ovviare in parte a questo problema è possibile però utilizzare un numero maggiore di telecamere, così da aumentare il numero di angolazioni visibili.

Anche il posizionamento dei *marker* sulla cute introduce degli artefatti: infatti, oltre alla non completa ripetibilità dell'azione e alla forte influenza legata all'esperienza dell'operatore, si aggiungono anche i cosiddetti *artefatti da tessuto molle*, ovvero i movimenti relativi presenti tra cute e osso, che introducono errori nella misurazione [16][17].

Le configurazioni possibili per questi sistemi sono molteplici, infatti è possibile utilizzare sia telecamere che operano nel campo del visibile, che dell'infrarosso, e i *marker* possono essere attivi o passivi.

I *marker* attivi sono composti da LED che generano un segnale luminoso sincronizzato con il sistema centrale, in modo da rendere più semplice il tracciamento. Non necessitano di sistemi di illuminazione esterna, né di preelaborazione per identificare e classificare i *marker*. Tuttavia hanno limitazioni riguardanti l'alimentazione dei dispositivi e la sincronizzazione via cavo, che limitano la naturalezza del movimento, e inoltre hanno angoli di emissione limitati, per cui per movimenti complessi risulta critico il set-up delle telecamere.

I *marker* passivi, invece, possono essere in materiale catarifrangente o semplicemente disegnati sulla cute: i primi necessitano di telecamere a infrarossi ad illuminazione stroboscopica e di software di preelaborazione per essere identificati.

I *marker* disegnati sulla cute, al contrario, vengono utilizzati con telecamere commerciali. Necessitano di un software semi - automatico per l'inseguimento dei *marker* sulle immagini del gesto motorio, e hanno in genere misurazioni meno precise rispetto a quelle con *marker* catarifrangenti, tuttavia permettono di fare acquisizioni anche al di fuori di un laboratorio attrezzato.

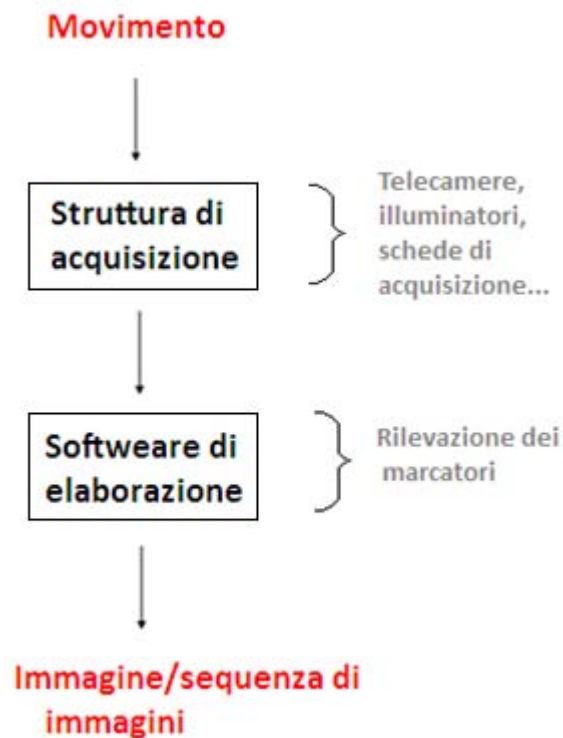
1.1.1.1 - Sistemi Stereofotogrammetrici

I sistemi stereofotogrammetrici, come detto, sono i più utilizzati nell'ambito di un laboratorio, siano essi con *marker* attivi o passivi.

Nel caso più comune in cui il sistema lavori con *marker* passivi, le telecamere sono costruite in modo da emettere esse stesse fasci di luce infrarossa proiettata

nell'area di lavoro; la luce quindi viene riflessa dai *marker* passivi ricoperti da materiale riflettente e acquisita dalle camere sensibili alla medesima lunghezza d'onda. Il segnale luminoso di ritorno proveniente dai *marker* sarà molto più intenso, rispetto al background, permettendo così una facile localizzazione.

La struttura di acquisizione nel caso di un sistema optoelettronico *stereofotogrammetrico* può essere quindi riassunta secondo lo schema seguente:



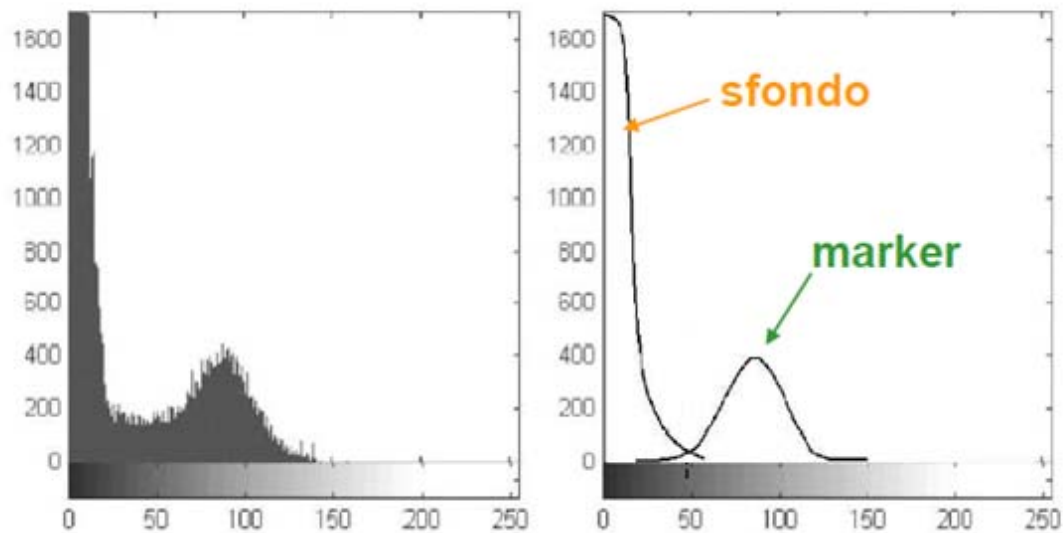
1.1.1 - Schematizzazione della struttura di acquisizione di un sistema stereo fotogrammetrico.

Dallo schema si nota come oltre alla struttura di acquisizione (composta da telecamere e illuminatori IR) il sistema di analisi prevede la presenza di un software atto a elaborare le informazioni rilevate dalle telecamere stesse al fine di fornire all'operatore la posizione istante per istante di ciascun *marker* presente nella scena.

In particolare ogni frame acquisito viene sottoposto ad una prima fase di segmentazione con lo scopo di estrarre dall'immagine la posizione 2D dei *marker*, che servirà poi per effettuare il processo di triangolazione.

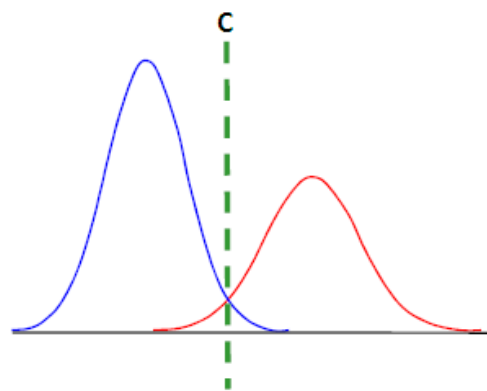
L'immagine viene inizialmente filtrata, al fine di ridurre i gradienti di illuminazione presenti, per poi procedere con la costruzione dell'istogramma

delle frequenze nelle diverse classi di tonalità di grigio. L'istogramma assume un tipico andamento a doppia gaussiana dove il primo picco, molto vicino allo zero, indica tutta l'informazione proveniente dallo sfondo mentre il secondo rappresenta l'oggetto della nostra elaborazione ossia l'informazione proveniente dai *marker*.



1.1.2 - Istogramma dell'immagine e sua approssimazione con due gaussiane [22]

Utilizzando una tecnica a mistura di gaussiane è possibile approssimare l'istogramma con due curve e calcolare, quindi, il valore della soglia ottima come valore in corrispondenza del quale le curve si intersecano; tale punto permette la minimizzazione degli errori di tipo I (falso positivo) e di tipo II (falso negativo).



1.1.3 - Calcolo del valore ottimo della soglia come intersezione tra le due curve [22]

Applicando una sogliatura all'immagine di partenza, è possibile ottenere i cosiddetti *blobs*, ovvero la proiezione bidimensionale sul piano immagine del set di *marker*. Di tali *blob* si dovranno stimare le coordinate del centro e per farlo è possibile procedere secondo diversi approcci.

In primis il centro del marcatore può essere stimato come baricentro dei pixel sopra la soglia (indipendentemente dai livelli di grigio). In particolare si procede al calcolo delle grandezze b_x e b_y , rispettivamente coordinate nel piano immagine lungo x e lungo y del centro del *blob*, secondo:

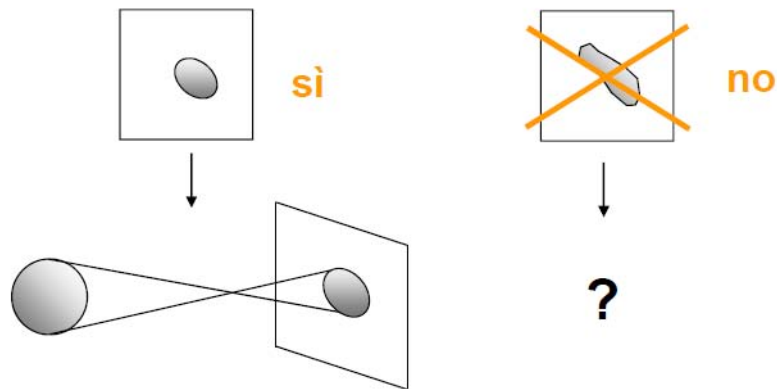
$$b_k = \frac{1}{N} \sum_{i=1}^N p_{ki} \quad k = \{x, y\} \quad (1.1)$$

Dove N è il numero di pixel sopra la soglia e p_{ki} è la coordinata lungo gli assi x e y del pixel i-esimo.

Si può procedere alla stime del centro del *marker* anche mediante una tecnica denominata “*circle fitting*” ovvero evidenziando il bordo dell'area illuminata (sopra soglia) e approssimandolo con una circonferenza.

In genere, prima di procedere al calcolo del centro dei diversi *marker*, è buona norma effettuare un processo cosiddetto di *blob - analysis*. Questa tecnica ha come obiettivo quello di estrarre dall'immagine solo le zone di effettivo interesse, eliminando false misure dovute principalmente ai riflessi.

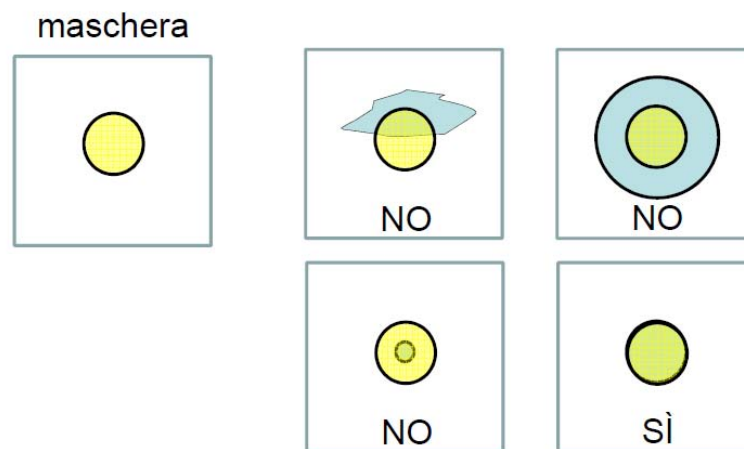
Si procede quindi con una prima verifica sulle dimensione del *blob*, andando ad eliminare tutte quelle proiezioni che risultano incompatibili con la reale dimensione del *blob* (troppo grandi o troppo piccole). Successivamente viene effettuato un controllo sulla forma assunta dal *blob*. In tal caso, l'ellissoide che approssima il marcatore può essere stimato a partire dalla matrice di covarianza sperimentale delle posizioni dei *pixel* rispetto al baricentro $B = [b_x, b_y]^T$: dall'analisi delle componenti principali della matrice così ottenuta è possibile risalire al fattore di forma del *blob*, che deve essere compatibile con la proiezione di una sfera.



1.1.4 - Analisi del blob in base alla forma dello stesso. A sinistra immagine compatibile con la proiezione di una sfera. A destra proiezione non compatibile con quella di una sfera. [22]

In questo modo è possibile eliminare tutta l'informazione che attraverso una semplice sogliatura verrebbe invece mantenuta e processata. Ne è un esempio il fenomeno dell'occlusione parziale: l'immagine del *marker* non è compatibile con la proiezione di una sfera e il dato viene, quindi, correttamente scartato in quanto condurrebbe ad un calcolo errato del baricentro.

Un metodo alternativo all'identificazione del *blob* prevede l'utilizzo di un processo di correlazione. In tal caso viene creata una maschera (*kernel*), di forma e dimensioni pari a quelle attese dell'immagine del *marker*, con la quale si analizza l'intera immagine alla ricerca di aree che presentino un fattore di correlazione superiore ad una soglia prestabilita.



1.1.5 - Analisi del blob in base alla forma dello stesso. A sinistra immagine compatibile con la proiezione di una sfera. A destra proiezione non compatibile con quella di una sfera. [22]

Infine, note per ogni *marker* le coordinate 2D in ciascun frame acquisito dalle diverse telecamere, è possibile risalire alla posizione spaziale tramite il processo di triangolazione, previa calibrazione del sistema adottato. Tali argomenti verranno trattati nello specifico nel Capitolo 2.

1.1.1.2 – Sistemi Video e Feature Tracking

Nel caso in cui non ci sia la possibilità di utilizzare telecamere dotate di dispositivi per la rilevazione e la trasduzione del segnale luminoso in segnale elettrico, si ricorre all'utilizzo di particolari software che permettono l'inseguimento digitale dei punti che caratterizzano il gesto motorio. Tale procedura è nota con il nome di *video tracking* o *feature tracking*. La maggiore difficoltà legata al *feature tracking* è quella di riuscire a identificare la corretta posizione di un *marker* nella sequenza di fotogrammi (*frames*), in particolare quando la velocità del gesto motorio è superiore al frame rate della telecamera. Dal momento che i punti da seguire non sono casuali, è necessario specificare le loro posizioni iniziali e il frame di partenza. Inoltre, a causa della complessità dei movimenti e delle variazioni di luminosità, è possibile che dei punti vengano persi nel tracciamento: in tal caso sarà compito dell'operatore fornire nuovamente la posizione del *marker*. Approfondimenti sugli algoritmi di *video tracking* saranno oggetto, anch'essi, del Capitolo 2.

1.2 – Formato File C3D

Il formato C3D (*Coordinate 3D*) è un formato di file di pubblico dominio per la memorizzazione accurata, in un unico contenitore, di traiettorie 3D e dati analogici per ogni processo di misura, insieme a tutti i relativi parametri che descrivono i dati. I dati salvati in un file C3D sono accessibili a chiunque, senza il bisogno di una licenza, e possono, quindi, essere facilmente scambiati e/o confrontati in laboratori differenti. Molte aziende che lavorano nell'ambito della *motion capture*, tra i quali BTS, CODA, C – Motion, Motion Analysis Corporation,

Motion Lab Systems, Noraxon, Qualisys, Vicon Motion Systems e altri, supportano questo tipo di file all'interno dei loro software, sia per il salvataggio diretto dei dati raccolti, sia per la conversione, in ambo i sensi, con i formati proprietari.

1.2.1 - Introduzione e storia

Fino agli anni '80 era di uso comune, fra i laboratori di analisi del movimento, utilizzare differenti sistemi commerciali, o addirittura sistemi creati all'interno dei laboratori stessi, per la gestione e l'elaborazione dei dati. Dal momento che ogni laboratorio utilizzava un formato di file diverso dagli altri, era praticamente impossibile, per i ricercatori che avevano diversi sistemi di analisi, scambiarsi vicendevolmente i dati. Di conseguenza, i dati raccolti in un esperimento tenuto in un laboratorio, con un certo sistema, non potevano essere verificati e analizzati in un altro laboratorio che non avesse il medesimo sistema di gestione, e misure identiche fra diversi sistemi non potevano essere comparate a causa dei diversi dati e parametri memorizzati.

Nel 1987, il Ph.D. Andrew Dainis sviluppò il formato di file C3D, come prodotto commerciale per AMASS (ADTech Motion Analysis Software System), un sistema software di fotogrammetria che rimpiazzasse i poco accurati ed inefficienti software di fotogrammetria disponibili all'epoca. La prima installazione di AMASS fu nel Laboratorio di Biomeccanica del National Institutes of Health. Alla fine degli anni '80 Oxford Metrics Ltd. acquisì i diritti di distribuzione di AMASS, e nel 1992 sviluppò l'applicazione Vicon Clinical Manager, che utilizzava il formato ADTech C3D come formato standard.

Il design di base del formato di file C3D era fondato sul desiderio di avere un unico formato di file che comunicasse i parametri e i dati tra le varie componenti di AMASS (calibrazione, *tracking*, identificazione dei *marker*...) e che inoltre fornisse in uscita le traiettorie 3D e i dati analogici raccolti. Di conseguenza le priorità di costituzione del file erano la flessibilità di memorizzazione di dati numerici e parametri testuali, e una memorizzazione comunque compatta ed

efficiente di tutta l'informazione utile in un unico file. Con queste prerogative è nato il C3D, un formato di file che rende semplice il confronto delle informazioni a ricercatori e clinici che operano in laboratori diversi e con diversi software, e grazie al quale gli utenti possono accedere a tutti i dati di cui hanno bisogno da un unico contenitore. Inoltre, grazie a queste peculiarità, non risente degli aggiornamenti rilasciati per i software, né dei progressi dell'hardware, quindi non diventa obsoleto.

Non essendo affiliato ad alcun produttore, le specifiche del file e la documentazione completa sono liberamente disponibili sul sito, che viene mantenuto come risorsa per gli utenti dalla Motion Lab Systems Inc., essendo autrice di diversi software che utilizzano il formato C3D.

1.2.2 - Descrizione del file

I file C3D sono in formato binario; a ciò è dovuta l'efficienza in termini di memorizzazione e accesso, e la loro flessibilità.

Il formato C3D tratta le informazioni come se queste appartenessero a due classi fondamentali: *Misure Fisiche* e *Parametri* relativi alle misure fisiche.

Misure Fisiche

Le misure fisiche sono a loro volta suddivise in due tipi: informazione di posizione (coordinate 3D) e dati numerici (informazioni analogico - digitali).

Ogni coordinata 3D è memorizzata in maniera grezza come X, Y e Z del singolo campione, con appresso le informazioni riguardo l'accuratezza (errore medio o residui) e i contributi delle telecamere (quale telecamera o osservatore ha prodotto i dati). Ogni campione di dati numerici, invece, può contenere informazioni analogiche digitalizzate, provenienti da fonti come EMG, celle di carico, piattaforme di pressione e di forza etc. Questi campioni vengono sincronizzati con i campioni delle coordinate 3D, in modo che sia facilmente determinabile a quale campione 3D corrispondano determinati dati numerici.

Parametri

Oltre alle misure fisiche un file C3D può contenere anche le informazioni relative a queste misure, come unità di misura, nomi etc. In più si possono memorizzare anche informazioni relative ai soggetti (nome, diagnosi), elementi relativi ad un protocollo di acquisizione o a situazioni particolari. Il formato permette di fornire una descrizione per ogni set di parametri, che diventa una sorta di documentazione interna riguardo quei parametri, tramite la quale ogni utente può condividere le informazioni che ritiene opportune e necessarie alla comprensione dei dati inseriti.

Dal punto di vista tecnico tutti i file C3D contengono un minimo di tre sezioni, ciascuna costituita da almeno un blocco di 512 byte. Ogni C3D file ha una sezione di intestazione (*header section*): questa contiene un puntatore alla sezione dati 3D e alla sezione parametri, informazioni utili a leggere e interpretare la sezione dei dati 3D, copie di alcuni parametri presenti nella sezione parametri e informazioni riguardo all'evento salvato. A questa sezione è associato il primo blocco.

Dal secondo blocco generalmente comincia la sezione parametri (*parameter section*), che può contenere uno più blocchi di 512 byte. Generalmente varia dagli otto ai dieci blocchi di lunghezza.

Dopo la sezione parametri c'è la sezione dati 3D (*3D data section*), indicata con il puntatore POINT:DATA_START, che contiene le coordinate e i dati analogici. Ovviamente la sua lunghezza varia dalla quantità dei dati memorizzati. Il formato più comunemente usato per salvare coordinate e dati analogici è il *signed integer* (ogni numero è memorizzato come un integer con segno a 16 bit nell'intervallo [-32768, +32767]). È comunque disponibile la possibilità di salvare i dati in formato virgola mobile (*floating point*), in particolare per i dati analogici che richiedono una precisione maggiore.

1.2.3 - Limitazioni

Le limitazioni legate all'utilizzo di file C3D in genere non causano problemi all'utente medio, ma si è preferito darne un accenno per completezza di trattazione.

Come detto nella sezione precedente, il formato numerico usato in prevalenza per il C3D è il *signed integer* a 16 bit, questo impone un limite massimo di frame da poter immagazzinare (32767) e un limite anche nel numero di caratteri in un parametro (127). Tale limite è facilmente superabile utilizzando *unsigned integer*, con il quale il numero massimo di frame salvabili raddoppia ed è esattamente equivalente da usare. Le uniche problematiche legate all'utilizzo di *unsigned integer* sono dovute ai vecchi programmi che possono interpretare elevati numeri *unsigned* come numeri negativi. A parte questo, relativamente piccolo, inconveniente, i file C3D *signed* e *unsigned* sono da considerarsi identici, con la differenza che possano contenere più o meno di 32767 frame 3D e parametri con più o meno di 127 variabili. Per il resto, a livello binario, non ci sono strutturali differenze tra i due tipi di file.

Esiste un'altra piccola limitazione legata alla frequenza di campionamento dei dati analogici e delle coordinate puntuali. È infatti richiesto che la frequenza di campionamento dei dati analogici sia la stessa, o al più un multiplo, della frequenza di campionamento delle coordinate 3D, in modo da assicurare la sincronizzazione dei dati analogico - digitali. Questa restrizione permette di calcolare facilmente l'esatta posizione di ogni frame e di ogni dato all'interno del file C3D [12].

Capitolo 2

Algoritmi per il Tracking di Features

Si è fatto accenno alla presenza di differenti tipologie di *marker* nell'ambito dell'analisi cinematica. Per completezza seguirà una breve trattazione di sistemi agenti con entrambe le tipologie.

I sistemi basati su *marker* attivi, come SELSPOT, OPTOTRACK, COSTEL etc, utilizzano *marker* ad emissione luminosa, collegati tramite fili ad una centralina che viene trasportata direttamente dal soggetto in questione. La luce viene emessa stroboscopicamente, in sincronia con il frame rate delle telecamere, così da facilitare l'estrazione e il *tracking* dei *marker*. Tuttavia l'applicabilità di tali sistemi è fortemente limitata dalle costrizioni meccaniche introdotte dalla massiccia presenza di fili, dalla necessità di un ambiente controllato e dalla limitazione nel numero di *marker* utilizzabili, dovuta al timing stroboscopico.

L'utilizzo di *marker* passivi, ricoperti da una pellicola retroriflettente, permette di superare facilmente alcune di queste limitazioni. Sistemi a infrarossi, come ELITE, MaxReflex, VICON etc, emettono luce nella banda dell'infrarosso, che viene riflessa dai *marker* e quindi registrata dalle telecamere. Le immagini ottenute con tali sistemi evidenziano un grande contrasto fra *marker* e sottofondo, permettendo anche in questo caso un processo di estrazione e *tracking* dei *marker* relativamente semplice, senza l'impedimento di fili e centralina. Tuttavia, la presenza simultanea di molti *marker* nella stessa immagine, rende necessario l'utilizzo di algoritmi di classificazione così da fornire un'identificazione univoca dei *marker* nella sequenza di immagini. Tale processo risulta però fallace in alcune situazioni, ad esempio nel caso in cui i *marker* siano disposti in punti anatomici molto vicini tra loro [1].

Anche con *marker* passivi retroriflettenti vi è la necessità di un ambiente controllato, e a questo si aggiunge anche l'ingombro dei *marker* stessi nel gesto atletico. Nella pratica natatoria, ad esempio, l'utilizzo di *marker* sferici procura un aumento inevitabile della resistenza passiva, che può influenzare negativamente la prestazione dell'atleta. Allo stesso modo, in condizioni di asciutto, la meccanica di un gesto complesso può essere modificata dalla presenza ingombrante dei *marker*, inficiando quindi l'analisi cinematica.

Un possibile approccio, per evitare l'ingombro dei *marker*, consiste nella sostituzione dei *marker* sferici con *marker* non riflettenti biadesivi o disegnati direttamente sulla pelle dell'atleta. Con questo setup, i movimenti vengono filmati con telecamere convenzionali, e i video risultanti vengono analizzati usando un software specifico per il *feature tracking*.

L'inseguimento manuale rappresenta di certo la soluzione più rozza in questo senso e richiede un grande dispendio in termini di risorse umane e tempo [2]. Pertanto la disponibilità di un software appropriato per il *tracking* automatico rappresenterebbe un significativo avanzamento nella pratica dell'analisi cinematica.

A livello commerciale sono disponibili diversi software per l'inseguimento dei *marker* e il calcolo delle variabili cinematiche, che restituiscono dati utilizzabili da allenatori e atleti [3]. Sebbene molti di questi software siano in grado di fare un *tracking* completamente automatico dei *marker*, nulla viene rivelato sui loro algoritmi e sulle tecniche di analisi utilizzate. Tale lacuna è determinante a livello di ricerca, in quanto la conoscenza di queste informazioni permetterebbe di ottimizzare la procedura di *tracking* automatico in ogni ambiente e condizione.

Partendo dal presupposto che il processo di *tracking* è già stato accuratamente descritto, come si vedrà nel proseguo del capitolo, rimane comunque la difficoltà di selezionare le *features* che possano essere opportunamente seguite [4]; sulla base di questo è stata sviluppata una grande varietà di algoritmi allo scopo di avere la maggiore robustezza con il minor tempo computazionale possibile.

Tra i più comuni metodi di *tracking* si ricorda il "*point tracking*" che è basato sul rilevamento di punti che rappresentano oggetti in frame consecutivi; tra le

tecniche di implementazione di tale metodo di *tracking*, quelle basate sul flusso ottico sono tra le più complesse, ma permettono di riconoscere il movimento in maniera molto accurata e senza la conoscenza a priori dello sfondo [5]. Il flusso ottico è una rappresentazione flessibile del movimento “visivo”, ed è particolarmente appropriato per l’analisi di immagini digitali.

Tra le varie tecniche basate sul flusso ottico, è stato verificato a livello sperimentale come l’algoritmo di Kanade - Lucas - Tomasi (KLT) [6][7] sia il più efficiente ed accurato, in termini di deviazioni angolari medie rispetto alla corretta orientazione spazio - temporale [8]. In particolare è stato appurato che, su sequenze di immagini sia reali che create ad hoc, i risultati forniti dall’algoritmo KLT sono i più affidabili e precisi di quelli ottenibili con altre tecniche basate sul flusso ottico [8]. Tali risultati sono stati confermati in un ulteriore esperimento [9] in cui, oltre ad essere comprovata nuovamente la precisione, è stato riscontrato che l’algoritmo aveva anche prestazioni migliori a livello computazionale. Infine, in un terzo esperimento [10], sono state verificate nuovamente le ottime prestazioni in precisione e costo computazionale, ed è stato notato che l’algoritmo mostrava anche una buona tolleranza alla presenza del rumore.

2.1 - Video Tracking in Analisi del Movimento

Il *video tracking* prevede una precisa procedura per essere attuato correttamente e, di conseguenza, dare dei risultati attendibili. Di seguito ne verrà data una trattazione specifica, seguita anche durante i lavori legati a questa tesi: calibrazione delle telecamere, *feature tracking*, triangolazione, stima e compensazione degli errori strumentali.

2.1.1 – Calibrazione delle telecamere

Il setup per l'analisi del movimento prevede la calibrazione delle telecamere, ossia la determinazione dei loro parametri geometrici, operazione necessaria per la ricostruzione tridimensionale. Vengono definiti quindi i seguenti parametri:

- *Parametri intrinseci (o interni)*: specificano le caratteristiche ottiche, geometriche e digitali della telecamera, come sensore, in modo indipendente dalla sua posizione spaziale. Essi sono:

- *Lunghezza focale* $\mathbf{f} = \begin{bmatrix} f_u \\ f_v \end{bmatrix}$: esprime la distanza focale orizzontale e verticale; il rapporto $\frac{f_u}{f_v}$ (detto rapporto di aspetto), esprime la forma del pixel, se il rapporto di aspetto è unitario il pixel è quadrato;

- *Punto principale* $\mathbf{cc} = \begin{bmatrix} cc_1 \\ cc_2 \end{bmatrix}$: raccoglie le coordinate del punto principale del piano immagine;

- *Coefficiente di asimmetria* α_c : scalare che definisce l'angolo tra gli assi x e y del pixel;

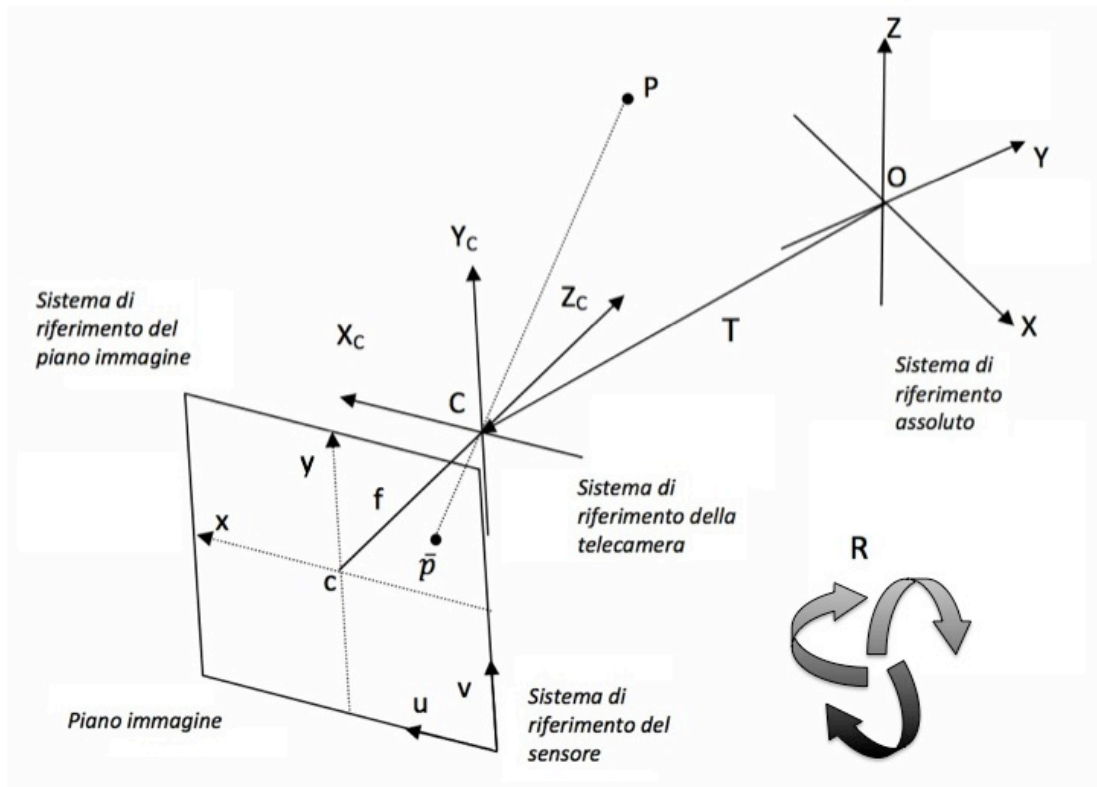
- *Coefficienti di distorsione* $\mathbf{kc} = \begin{bmatrix} kc_1 \\ kc_2 \\ kc_3 \\ kc_4 \\ kc_5 \end{bmatrix}$: raccoglie i coefficienti di

distorsione radiale e tangenziale dell'immagine.

- *Parametri estrinseci (o esterni)*: definiscono la posizione del sistema di riferimento della telecamera rispetto al sistema di riferimento assoluto.

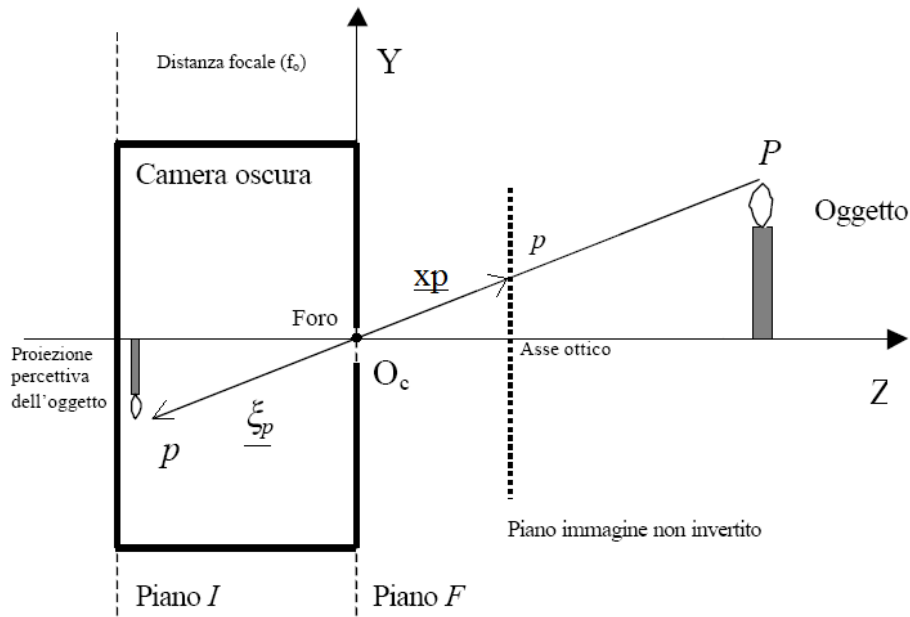
- *Matrice di rotazione* \mathbf{R} : matrice 3x3 che definisce l'orientamento della telecamera nel sistema di riferimento assoluto;

- *Vettore di traslazione* \mathbf{T} : vettore 3x1 che definisce le coordinate spaziali della telecamera nel sistema di riferimento assoluto.



2.1.1 - Proiezione sul piano immagine di un punto P nello spazio 3D [22]

Per poter calibrare il sistema, tuttavia, è necessario introdurre un modello matematico dello stesso. Il più utilizzato in letteratura è il modello *pin - hole* (o a foro stenopeico o modello della camera oscura). Le ipotesi di tale modello trascurano la presenza dell'ottica, che viene modellata come un foro di diametro infinitesimo posto al centro della prospettiva, sul piano focale F. I raggi, provenienti da un oggetto nello spazio, attraversano il foro e formano sul piano immagine una rappresentazione invertita dell'oggetto stesso.



2.1.2 - Modello della camera oscura o modello pin hole [23]

A livello puntuale, il punto P nello spazio viene proiettato sul piano immagine sul punto p .

Formalmente si può scrivere:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_u & f_v \end{bmatrix} \begin{pmatrix} \frac{1}{Z_P} & 0 \\ 0 & \frac{1}{Z_P} \end{pmatrix} \begin{bmatrix} X_P \\ Y_P \end{bmatrix} \quad (2.1)$$

Passando in coordinate omogenee e chiamando $\lambda = \frac{1}{Z_P}$ si può riscrivere:

$$\lambda \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{pmatrix} -f_u & 0 & 0 & 0 \\ 0 & -f_v & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} = \mathbf{M} \mathbf{P} \quad (2.2)$$

Per convenzione, il sensore della telecamera ha un proprio sistema di riferimento centrato nell'angolo inferiore destro del piano immagine. Le coordinate nel piano immagine relative a tale sistema di riferimento sono (u, v) ; la trasformazione che

collega le coordinate di p alle coordinate rispetto al sistema di riferimento del sensore è:

$$\mathbf{p}_s = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} \frac{1}{k_u} & 0 & cc_1 \\ 0 & \frac{1}{k_v} & cc_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{H} \mathbf{p} \quad (2.3)$$

Ove k_u e k_v sono le dimensioni orizzontale e verticale del pixel.

Inoltre, dato che il punto P è espresso rispetto al sistema di riferimento della telecamera:

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{pmatrix} \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} = \mathbf{D} \mathbf{P}_a \quad (2.4)$$

Ove \mathbf{P}_a è il punto P espresso rispetto al sistema di riferimento globale.

In conclusione, il modello *pin - hole* può essere rappresentato da una matrice 3x4, risultato delle trasformazioni fin qui viste:

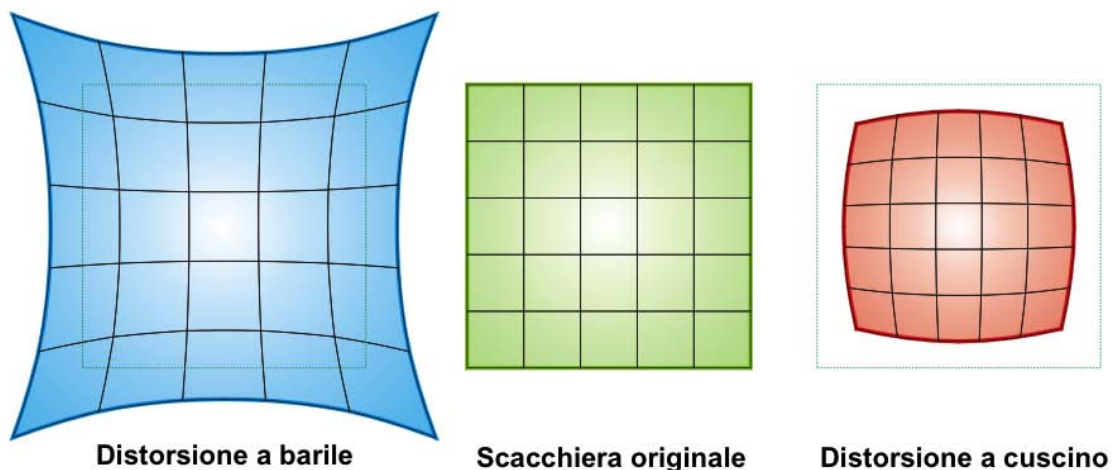
$$\mathbf{A} = \mathbf{H} \mathbf{M} \mathbf{P} = \begin{pmatrix} \frac{1}{k_u} & 0 & cc_1 \\ 0 & \frac{1}{k_v} & cc_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -f_u & 0 & 0 & 0 \\ 0 & -f_v & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{pmatrix} \quad (2.5)$$

A partire da questa matrice si possono definire le *equazioni di collinearità* che permettono di trasformare un punto dal sistema di riferimento del sensore della telecamera, al sistema di riferimento assoluto:

$$\lambda \mathbf{p}_s = \mathbf{A} \mathbf{P}_a \quad (2.6)$$

Tuttavia il modello *pin - hole* non può che essere un'approssimazione della telecamera, che fornisce delle semplici equazioni nel calcolo delle coordinate

spaziali di un punto. Nella realtà le telecamere non utilizzano un foro di dimensione infinitesima, in quanto l'intensità del fascio luminoso risulterebbe troppo bassa per essere rilevata dagli elementi fotosensibili del piano immagine. Di conseguenza è necessario allargare il foro e far uso di un'ottica per far convergere i raggi luminosi sul piano immagine (ovvero per *mettere a fuoco* l'immagine). L'utilizzo di lenti introduce inevitabilmente della distorsione, legata all'aberrazione. Il risultato è che i raggi vengono fatti convergere ad una distanza diversa dalla distanza focale. Se la lunghezza focale diminuisce all'aumentare della distanza dall'asse ottico si ha la cosiddetta distorsione a botte, se invece aumenta si parla di distorsione a cuscino.



2.1.3 - Tipologie di distorsione

Fatte queste considerazioni, in genere si estende il modello *pin - hole* con delle correzioni per le coordinate dell'immagine sistematicamente distorte.

Nella *Camera Calibration Toolbox for Matlab* proposta da J. Y. Bouget [24] viene proposto il modello di distorsione di Brown - Conrady, o a filo a piombo [25].

Tale modello prevede l'aggiunta di distorsione tangenziale e radiale; formalmente, detto \mathbf{x}_n un punto dell'immagine normalizzata:

$$\mathbf{x}_n = \begin{bmatrix} \frac{X_p}{Z_p} \\ \frac{Y_p}{Z_p} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.7)$$

Sia $r^2 = x^2 + y^2$, aggiungendo il contributo della distorsione della lente, le coordinate normalizzate del punto diventano:

$$\mathbf{x}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + kc_1 r^2 + kc_2 r^4 + kc_5 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{dx} \quad (2.8)$$

$$\mathbf{dx} = \begin{bmatrix} 2 kc_3 xy + kc_4(r^2 + x^2) \\ kc_3(r^2 + y^2) + 2 kc_4 xy \end{bmatrix} \quad (2.9)$$

Le coordinate finali del pixel, tenuto conto dei contributi della distorsione, diventano quindi:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_u (x_d + \alpha_c y_d) + cc_1 \\ f_v y_d + cc_2 \end{bmatrix} \quad (2.10)$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{KK} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{pmatrix} f_u & \alpha_c & cc_1 \\ 0 & f_v & cc_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (2.11)$$

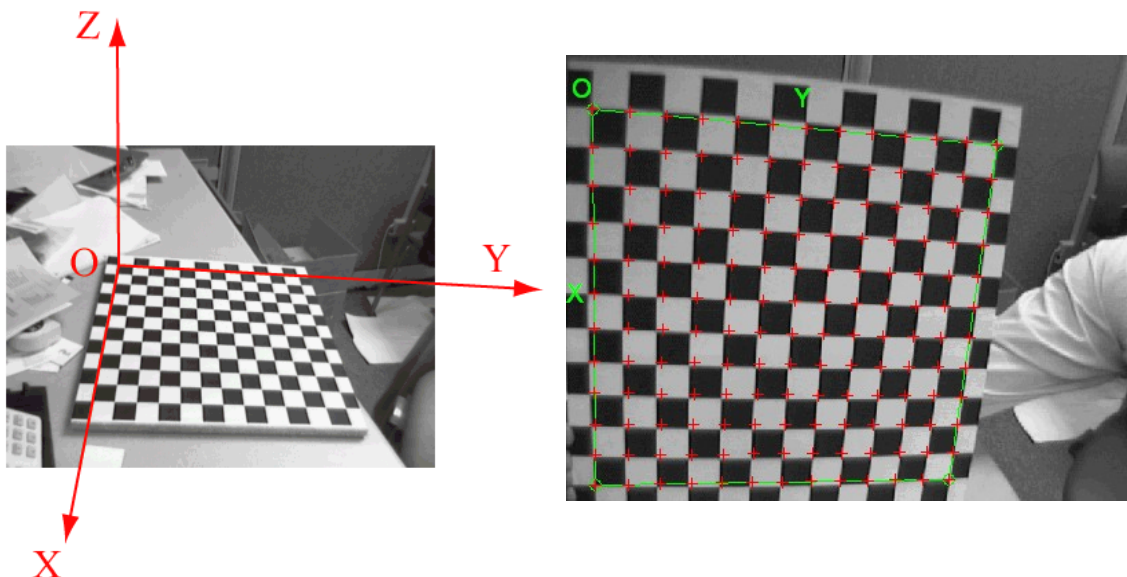
La procedura per ricavare i parametri fin qui utilizzati è nota come *calibrazione intrinseca* del sistema. Operativamente viene utilizzata una scacchiera, di cui si conoscono il numero dei quadrati e la lunghezza del loro lato; vengono quindi digitalizzati i bordi e mappate le distanze fra i vertici di ogni quadrato. Poiché le distanze sono note, è possibile ricavare i parametri attraverso la risoluzione ai minimi quadrati dell'equazione (2.11).

La *calibrazione estrinseca* è invece necessaria per poter allocare nello spazio la telecamera, e per farlo vengono utilizzati i parametri visti precedentemente e nuovamente una scacchiera. Infatti, noti i parametri geometrici interni della telecamera, si può collocare la scacchiera al centro del volume di acquisizione, in

modo tale che sia visibile da ogni telecamera. Preso un vertice all'interno della scacchiera, è possibile collegare le coordinate di tale punto rispetto al sistema di riferimento identificato dalla scacchiera, con le sue coordinate rispetto al sistema di riferimento globale:

$$\mathbf{P} = \mathbf{R}_i \mathbf{P}_a + \mathbf{T}_i \quad i \in \{1, \dots, \#Telecamere\} \quad (2.12)$$

Una volta fatta la calibrazione estrinseca, è possibile esprimere, secondo il sistema di riferimento globale, le coordinate di un punto nel sistema di riferimento della telecamera.



2.1.4 - Fase di calibrazione estrinseca (a sinistra) ed intrinseca (a destra) [24]

2.1.2 - Feature Tracking

Come anticipato nel Capitolo 1, il *feature tracking* non è altro che l'inseguimento a video di specifici punti, detti per l'appunto *features*. Il concetto cardine alla base degli algoritmi di *feature tracking* è il cosiddetto *optical flow*, che considera il moto all'interno di una rappresentazione digitale.

2.1.2.1 - Optical Flow

Con il termine flusso ottico (*optical flow*), secondo la teoria di Gibson [18], si intende il modello di moto apparente di oggetti, superfici e bordi in una rappresentazione visuale digitale (scena visiva), causato dal moto relativo tra l'osservatore e la scena stessa. Tale concetto è quindi alla base della stima del movimento. Recentemente infatti il termine è stato utilizzato anche dagli ingegneri, ad indicare tecniche correlate di elaborazione delle immagini, rilevazione del movimento, segmentazione di oggetti, *pattern recognition* etc. [19][8]

Dal punto di vista matematico, il metodo del flusso ottico tenta di stimare il movimento fra due frames presi ai tempi t e $t + \Delta t$ in ogni posizione.

Prendiamo in considerazione il caso più comune con due coordinate spaziali, più la coordinata temporale ($2D + t$), sarà poi possibile generalizzare anche a casi n -dimensionali. Un pixel è caratterizzato dalle coordinate $\mathbf{x} = (u, v, t)$ e da un'intensità luminosa $I(x, y, t)$. Preso il medesimo pixel nel frame successivo, esso si sarà genericamente mosso di $\Delta \mathbf{x} = (\Delta u, \Delta v, \Delta t)$ e la sua funzione di intensità risulterà essere $I(u + \Delta u, v + \Delta v, t + \Delta t)$. Per il vincolo di costanza della luminosità tra due frame successivi deve valere la relazione:

$$I(u, v, t) = I(u + \Delta u, v + \Delta v, t + \Delta t) \quad (2.13)$$

Assumendo che il movimento compiuto dal pixel sia sufficientemente piccolo, è possibile sviluppare in serie di Taylor la funzione intensità al frame successivo:

$$I(u + \Delta u, v + \Delta v, t + \Delta t) = I(u, v, t) + \frac{\partial I}{\partial u} \Delta u + \frac{\partial I}{\partial v} \Delta v + \frac{\partial I}{\partial t} \Delta t + \sigma \quad (2.14)$$

Sostituendo nella (2.13) e dividendo per Δt si ottiene:

$$\frac{\partial I}{\partial u} \Delta u + \frac{\partial I}{\partial v} \Delta v + \frac{\partial I}{\partial t} \Delta t = 0 \quad (2.15)$$

$$\frac{\partial I}{\partial u} \frac{\Delta u}{\Delta t} + \frac{\partial I}{\partial v} \frac{\Delta v}{\Delta t} + \frac{\partial I}{\partial t} = 0 \quad (2.16)$$

$$\frac{\partial I}{\partial u} V_u + \frac{\partial I}{\partial v} V_v + \frac{\partial I}{\partial t} = 0 \quad (2.17)$$

Dove V_u e V_v sono le componenti lungo gli assi u, v del flusso ottico di $I(u, v, t)$. Chiamando I_u, I_v e I_t le derivate parziali di $I(u, v, t)$ lungo le tre coordinate spazio - temporali, si ottiene:

$$I_u V_u + I_v V_v = -I_t \quad (2.18)$$

$$\nabla I^T \cdot \mathbf{V} = -I_t \quad (2.19)$$

L' equazione (2.18) è scalare e in due incognite V_u e V_v , quindi non risolvibile. Questo problema degli algoritmi di *optical flow* è noto come problema dell'apertura¹. Per trovare il flusso ottico, quindi, si necessita di ulteriori vincoli addizionali, che tutti i metodi per la stima del flusso introducono².

2.1.2.2 - Metodo di Lucas - Kanade e Algoritmo KLT [6] [7]

Il metodo proposto da Lucas - Kanade si basa sull'ipotesi che il flusso sia essenzialmente costante in un intorno locale del pixel considerato, e risolve il problema del flusso ottico su tutti i pixels presenti nella *neighbourhood* del pixel principale, con il metodo dei minimi quadrati.

¹ Problema dell'apertura: Riguarda l'impossibilità di determinare in modo certo la componente traslazionale di un oggetto in movimento, osservandolo da un foro. Supponendo infatti di osservare da un foro una bacchetta che si sposta, non siamo in grado di determinarne la lunghezza (a meno che questa non sia inferiore al diametro del foro), e conseguentemente la velocità traslazionale, ma solo la componente di velocità ortogonale alla bacchetta stessa.

² Metodi per la determinazione del flusso ottico:

- Correlazione di fase
- Metodi block - based
- Metodi differenziali
 - Metodo di Lucas - Kanade [6]
 - Metodo di Horn - Schunck [20]
 - Metodo di Buxton - Buxton [21]
 - Metodo di Black - Jepson [8]
- Metodi di discretizzazione ottimizzata

Formalmente: siano q_1, q_2, \dots, q_n gli n pixel presenti nell'intorno del pixel p , le componenti del flusso ottico, V_u e V_v , devono soddisfare alle equazioni:

$$I_u(q_i)V_u + I_v(q_i)V_v = -I_t(q_i) \quad i = \{1, \dots, n\} \quad (2.20)$$

Scrivendole sotto forma matriciale, $\forall q_i, i \in \{1, \dots, n\}$

$$\mathbf{A} = \begin{bmatrix} I_u(q_1) & I_v(q_1) \\ \vdots & \vdots \\ I_u(q_n) & I_v(q_n) \end{bmatrix}, \mathbf{v} = \begin{bmatrix} V_u \\ V_v \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -I_t(q_1) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \quad (2.21)$$

Il sistema risulta evidentemente sovradeterminato, in quanto $n \gg 2$, quindi si giunge ad una soluzione di compromesso ai minimi quadrati:

$$\mathbf{v} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (2.22)$$

In queste ipotesi, è ragionevole affermare che due immagini riferite istanti temporali vicini siano fortemente correlate, e si può ammettere che tali immagini siano in realtà la medesima immagine, presa però da differenti punti di vista. Di conseguenza la funzione $I(u, v, t)$ soddisfa alla seguente proprietà:

$$I(u, v, t + \tau) = I(u - \xi, v - \eta, t) \quad (2.23)$$

Due immagini successive si possono, quindi, ottenere spostando ogni singolo punto (u, v) dell'immagine di una quantità $\mathbf{d} = (\xi, \eta)$, detta *displacement*, risultato del flusso ottico dell'immagine. In termini pratici però la determinazione di \mathbf{d} , pixel per pixel, è complicata dalla difficile distinzione dell'intensità luminosa di due pixel adiacenti, e dalle variazioni di intensità luminosa dovute al rumore. Risulta quindi preferibile considerare una finestra di pixel. In tal caso però si pongono altre problematiche, legate alle differenze di comportamento di diversi punti in condizioni limite. Se ad esempio la superficie tridimensionale è molto inclinata, il pattern può risultare deformato da un frame all'altro; oppure, se la

finestra scelta è in punto vicino ad un confine di occlusione, i pixel possono muoversi a velocità differenti e sparire o riapparire in frame successivi.

In definitiva, quindi, i problemi riguardano la capacità di inseguire sempre la stessa finestra, anche se il contenuto cambia, e come ottenere il *displacement* combinando diverse velocità.

Il primo problema è stato superato considerando solamente la generica finestra il cui aspetto non variava di molto, scartando quelle con grandi variazioni. Il secondo problema è stato invece affrontato introducendo una mappa affine, ovvero un modello che permettesse di considerare trasformazioni più complesse delle semplici traslazioni, e fosse in grado di associare diverse velocità a diversi punti dell'immagine. Tale approccio è stato però accantonato, in quanto i vantaggi di un modello più complesso non pareggiavano le difficoltà di un'eccessiva parametrizzazione del sistema, che comportava di dover lavorare su finestre più grandi. Considerare finestre più piccole permette di stimare certamente meno parametri, ma riduce di molto il lavoro per ottenerli. In queste ipotesi, e ponendo di avere una frequenza di acquisizione sufficientemente alta, risulta accettabile trascurare il contributo della mappa affine, approssimando così il movimento alla sola traslazione \mathbf{d} ; il vettore peraltro consta di soli due parametri, di conseguenza perfettamente compatibile con le ipotesi di lavoro con piccole finestre. Ogni discrepanza fra finestre successive che non possa essere spiegata da una traslazione viene considerata un errore, e il vettore di *displacement* viene scelto in modo da minimizzare l'errore residuo.

Formalmente:

si ridefiniscono le funzioni

$$J(\mathbf{x}) = I(u, v, t + \tau) ; I(\mathbf{x} - \mathbf{d}) = I(u - \xi, v - \eta, t) \quad (2.24)$$

Il modello dell'immagine comprendente il rumore diventa (dove $n(\mathbf{x})$ è il rumore):

$$J(\mathbf{x}) = I(\mathbf{x} - \mathbf{d}) + n(\mathbf{x}) \quad (2.25)$$

Il vettore \mathbf{d} è scelto in modo da minimizzare l'errore residuo ε :

$$\varepsilon = \int_{\mathcal{W}} [I(\mathbf{x} - \mathbf{d}) - J(\mathbf{x})]^2 \omega(\mathbf{x}) d\mathbf{x} \quad (2.26)$$

ove \mathcal{W} è la finestra considerata e $\omega(\mathbf{x})$ è una funzione peso, che nel caso più semplice può essere imposta a 1.

Ponendo ora che il vettore \mathbf{d} sia sufficientemente piccolo, è possibile sviluppare in serie di Taylor la funzione $I(\mathbf{x} - \mathbf{d})$, fermandola al primo ordine:

$$I(\mathbf{x} - \mathbf{d}) \approx I(\mathbf{x}) - \begin{bmatrix} I_u(\mathbf{x} - \mathbf{d}) \\ I_v(\mathbf{x} - \mathbf{d}) \end{bmatrix} \cdot \mathbf{d} \quad (2.27)$$

Chiamando $\mathbf{g} = \begin{bmatrix} I_u(\mathbf{x} - \mathbf{d}) \\ I_v(\mathbf{x} - \mathbf{d}) \end{bmatrix}$ e $h = I(\mathbf{x}) - J(\mathbf{x})$, si può ridefinire l'errore residuo:

$$\varepsilon = \int_{\mathcal{W}} [I(\mathbf{x}) - \mathbf{g} \cdot \mathbf{d} - J(\mathbf{x})]^2 \omega(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{W}} [h - \mathbf{g} \cdot \mathbf{d}]^2 \omega(\mathbf{x}) d\mathbf{x} \quad (2.28)$$

Come si può notare, il residuo ε è una funzione quadratica di \mathbf{d} , di conseguenza la minimizzazione può essere fatta in forma chiusa. Derivando ε per \mathbf{d} , e uguagliando a zero il risultato, si ottiene:

$$\int_{\mathcal{W}} (h - \mathbf{g} \cdot \mathbf{d}) \mathbf{g} \omega(\mathbf{x}) dA = 0 \quad (2.29)$$

Dal momento che $(\mathbf{g} \cdot \mathbf{d}) \mathbf{g} = (\mathbf{g} \mathbf{g}^T) \mathbf{d}$ e assumendo \mathbf{d} costante nella finestra \mathcal{W} , possiamo riscrivere la (2.29) come:

$$\left(\int_{\mathcal{W}} \mathbf{g} \mathbf{g}^T \omega(\mathbf{x}) dA \right) \mathbf{d} = \int_{\mathcal{W}} h \mathbf{g} \omega(\mathbf{x}) dA \quad (2.30)$$

Questo è un sistema di due equazioni scalari in due incognite, quindi risolvibile:

$$\mathbf{G}\mathbf{d} = \mathbf{e} \quad (2.31)$$

in cui la matrice dei coefficienti è la matrice simmetrica 2x2

$$\mathbf{G} = \int_{\mathcal{W}} \mathbf{g}\mathbf{g}^T \omega(\mathbf{x}) dA \quad (2.32)$$

E i termini noti sono nel vettore 2x1

$$\mathbf{e} = \int_{\mathcal{W}} (I - J)\mathbf{g} \omega(\mathbf{x}) dA \quad (2.33)$$

L'equazione (2.31) è lo step fondamentale per il *tracking*: per ogni paio di frame consecutivi la matrice \mathbf{G} può essere calcolata su un frame, stimando i gradienti e ricavando i loro momenti del secondo ordine, mentre il vettore \mathbf{e} si trova come differenza dei due frames, insieme al gradiente di cui sopra.

Nella pratica tuttavia, le soluzioni del sistema possono ritenersi affidabili se e solo se la matrice \mathbf{G} è al di sopra del livello di rumore dell'immagine e se non è malcondizionata³. Questo si traduce nella necessità che gli autovalori della matrice \mathbf{G} siano grandi, per risolvere il problema del rumore, e che non differiscano di molti ordini di grandezza, per il mal - condizionamento.

In conclusione: dal momento che le variazioni di intensità all'interno di una finestra sono limitate dal massimo valore ammissibile per un pixel, l'autovalore maggiore non può assumere valori arbitrariamente grandi; quindi si può concludere che quando l'autovalore più piccolo di \mathbf{G} è sufficientemente grande da soddisfare al criterio sul rumore, allora la matrice \mathbf{G} è anche bencondizionata.

³ **Mal - condizionamento**: un problema si definisce mal - condizionato se a piccole perturbazioni dei dati iniziali, corrispondono grandi variazioni nelle soluzioni finali.

La finestra quindi non viene scartata se:

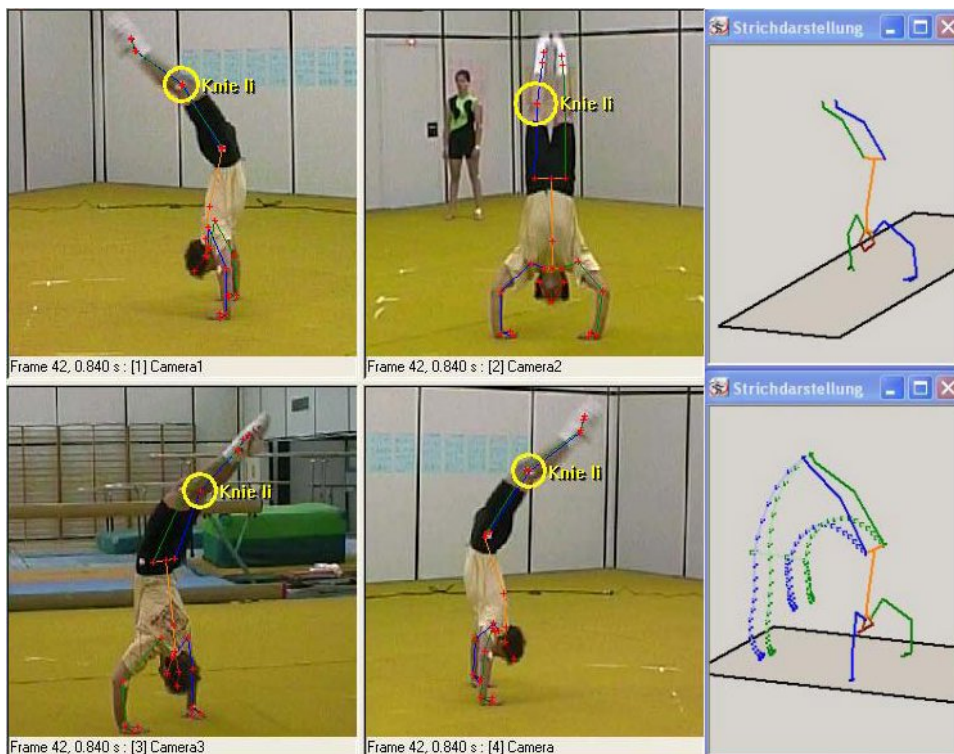
$$\min(\lambda_1, \lambda_2) > \lambda \quad (2.34)$$

Con λ_1, λ_2 autovalori della matrice \mathbf{G} , e λ soglia prefissata per il rumore. [7]

2.1.2.3 – Software Commerciali: Simi Motion

A livello commerciale esistono diversi software in grado di fare *feature tracking*. Tra questi probabilmente il più noto è Simi Motion.

A differenza di molti software sul mercato, Simi ha abbandonato la tecnologia con *marker* passivi catarifrangenti e telecamere a LED che emettono luce infrarossa, e si basa esclusivamente su tecniche basate su immagini digitali.



2.1.5 - Esempio di utilizzo di Simi Motion

Oltre alla possibilità di eseguire *tracking marker - based*, Simi permette anche *pattern matching* o *silhouette tracking markerless*. Il sistema si basa sull'utilizzo di telecamere industriali sincronizzate, ad alta risoluzione ed elevati *frame rate*. I dati

ottenuti in uscita, comprendenti dati di cinematica, cinetica, EMG e pressione, possono poi essere ulteriormente elaborati per ottenere parametri di analisi supplementari, come altri angoli, dati EMG filtrati etc. Tutti i risultati possono essere poi visualizzati in report standard o memorizzati in ASCII, C3D e altri formati.

Certamente questo software, e i suoi concorrenti, offrono una grande possibilità di calcolo e di precisione, ma essendo licenziati non permettono di accedere agli algoritmi interni del programma di *tracking*, di conseguenza non permettono all'operatore finale di poter decidere se l'algoritmo usato possa essere o meno funzionale alle proprie necessità. Tale limite è superato da eseguibili *open source*, che offrono prestazioni di certo leggermente inferiori a quelle dei software commerciali, ma permettono all'utente finale di poter verificare e controllare l'algoritmo alla base del *tracking* e poter quindi fare valutazioni in base alle proprie esigenze.

2.1.3 - Triangolazione

Una volta note le caratteristiche geometriche e spaziali delle telecamere, è possibile ricostruire le coordinate nello spazio dei punti, a partire da almeno una coppia di telecamere. Per raggiungere tale scopo è possibile seguire due strade differenti, in ordine di presentazione: il metodo lineare non omogeneo e il metodo lineare omogeneo.

Metodo Lineare Non Omogeneo

Nel metodo lineare non omogeneo il punto di partenza prevede la stesura delle equazioni di collinearità (2.28) per ciascuna telecamera:

$$\begin{cases} A_1 P_a = \lambda_1 p_{s1} \\ A_2 P_a = \lambda_2 p_{s2} \\ \vdots \\ A_k P_a = \lambda_k p_{sk} \end{cases} \quad k = \#Telecamere \quad (2.35)$$

Per ogni telecamera si ottengono tre equazioni:

$$\begin{cases} A_{11}^i X_a + A_{12}^i Y_a + A_{13}^i Z_a + A_{14}^i = u_i (R_{31}^i X_a + R_{32}^i Y_a + R_{33}^i Z_a + T_z^i) \\ A_{21}^i X_a + A_{22}^i Y_a + A_{23}^i Z_a + A_{24}^i = v_i (R_{31}^i X_a + R_{32}^i Y_a + R_{33}^i Z_a + T_z^i) \\ A_{31}^i X_a + A_{32}^i Y_a + A_{33}^i Z_a + A_{34}^i = R_{31}^i X_a + R_{32}^i Y_a + R_{33}^i Z_a + T_z^i \end{cases} \quad (2.36)$$

Per costruzione della matrice \mathbf{A} , si può verificare che la terza riga di equazioni è in realtà un'identità, di conseguenza per ogni telecamera si avranno due equazioni linearmente indipendenti:

$$\begin{cases} (A_{11}^1 - u_1 R_{31}^1) X_a + (A_{12}^1 - u_1 R_{32}^1) Y_a + (A_{13}^1 - u_1 R_{33}^1) Z_a = u_1 T_z^1 - A_{14}^1 \\ (A_{21}^1 - v_1 R_{31}^1) X_a + (A_{22}^1 - v_1 R_{32}^1) Y_a + (A_{23}^1 - v_1 R_{33}^1) Z_a = v_1 T_z^1 - A_{24}^1 \\ (A_{11}^2 - u_2 R_{31}^2) X_a + (A_{12}^2 - u_2 R_{32}^2) Y_a + (A_{13}^2 - u_2 R_{33}^2) Z_a = u_2 T_z^2 - A_{14}^2 \\ (A_{21}^2 - v_2 R_{31}^2) X_a + (A_{22}^2 - v_2 R_{32}^2) Y_a + (A_{23}^2 - v_2 R_{33}^2) Z_a = v_2 T_z^2 - A_{24}^2 \\ \vdots \\ (A_{11}^k - u_k R_{31}^k) X_a + (A_{12}^k - u_k R_{32}^k) Y_a + (A_{13}^k - u_k R_{33}^k) Z_a = u_k T_z^k - A_{14}^k \\ (A_{21}^k - v_k R_{31}^k) X_a + (A_{22}^k - v_k R_{32}^k) Y_a + (A_{23}^k - v_k R_{33}^k) Z_a = v_k T_z^k - A_{24}^k \end{cases} \quad (2.37)$$

$$k = \#Telecamere$$

In forma compatta si può scrivere:

$$\begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ \vdots & \vdots & \vdots \\ B_{2k1} & B_{2k2} & B_{2k3} \end{pmatrix} \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_{2k} \end{pmatrix} \quad (2.38)$$

$$\mathbf{B} \mathbf{P}_a = \mathbf{c} \quad (2.39)$$

Il sistema è evidentemente sovradeterminato con $2k$ equazioni in tre incognite, a causa del rumore non esiste soluzione esatta, per cui si riformula il problema tenendo conto del rumore di misura:

$$\mathbf{c} = \mathbf{B} \mathbf{P}_a + \mathbf{v} \quad (2.40)$$

Tale problema può essere facilmente risolto con una soluzione ai minimi quadrati:

$$\widehat{\mathbf{P}}_a = \operatorname{argmin}(\|\mathbf{v}\|) = \operatorname{argmin}([\mathbf{c} - \mathbf{B}\mathbf{P}_a]^T [\mathbf{c} - \mathbf{B}\mathbf{P}_a]) \quad (2.41)$$

$$\widehat{\mathbf{P}}_a = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{c} \quad (2.42)$$

Metodo Lineare Omogeneo

Nel metodo lineare omogeneo invece il cardine è lavorare direttamente sul piano immagine.

Nella fattispecie, se $\mathbf{p}_s = \begin{bmatrix} u \\ v \end{bmatrix}$ è un punto misurato sul piano immagine e

$\mathbf{P}_a = \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix}$ è il corrispettivo punto (ancora incognito) ricostruito nello spazio, è

immediato poter scrivere la proiezione sul piano immagine, $\overline{\mathbf{p}}_s = \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix}$, di \mathbf{P}_a :

(in coordinate omogenee)

$$\overline{\mathbf{p}}_s = \begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = \frac{1}{\lambda} \mathbf{A} \mathbf{P}_a = \frac{1}{\lambda} \mathbf{A} \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} \quad (2.43)$$

Affinché la retroproiezione di \mathbf{P}_a ($\overline{\mathbf{p}}_s$) coincida con il punto \mathbf{p}_s è sufficiente imporre che i due vettori siano paralleli:

$$\mathbf{p}_s \times \overline{\mathbf{p}}_s = 0 \quad (2.44)$$

$$\mathbf{p}_s \times \frac{1}{\lambda} \mathbf{A} \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} = 0 \quad (2.45)$$

A questo punto è possibile trascurare il fattore incognito λ e scrivere il prodotto vettoriale come prodotto di una matrice antisimmetrica, contenente le coordinate di \mathbf{p}_s , con il vettore $\overline{\mathbf{p}}_s$:

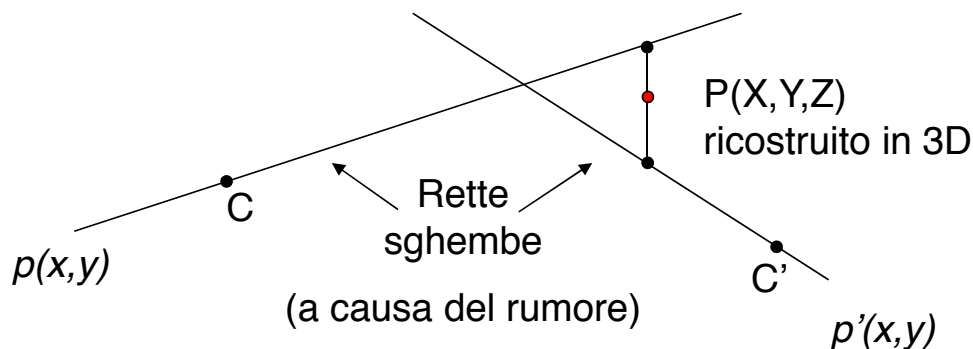
$$\begin{pmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{pmatrix} \mathbf{A} \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} = 0 \quad (2.46)$$

Il punto \mathbf{P}_a può essere retroproiettato su tutte le telecamere che “lo vedono”, per cui reiterando e chiamando il risultato del prodotto fra le due matrici \mathbf{S} , si ha:

$$\begin{pmatrix} 0 & -1 & v_i \\ 1 & 0 & -u_i \\ -v_i & u_i & 0 \end{pmatrix} \mathbf{A} = \mathbf{S}_i \quad i \in \{1, \dots, k = \#Telecamere\} \quad (2.47)$$

$$\begin{cases} \mathbf{S}_1 \mathbf{P} = 0 \\ \mathbf{S}_2 \mathbf{P} = 0 \\ \vdots \\ \mathbf{S}_k \mathbf{P} = 0 \end{cases} \Rightarrow \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \\ \mathbf{S}_k \end{bmatrix} \mathbf{P} = 0 \Rightarrow \mathbf{Q} \mathbf{P} = 0 \quad (2.48)$$

Il problema è sovradeterminato, a causa del rumore non esiste una soluzione esatta: le rette che congiungono il punto \mathbf{P}_a al centro del sistema di riferimento di ogni telecamera, sono sghembe.



2.1.6 - Ricostruzione 3D tramite triangolazione

Si riformula quindi il problema come:

$$\begin{cases} \operatorname{argmin}_P \|\mathbf{Q} \mathbf{P}\| \\ \|\mathbf{P}\| = 1 \end{cases} \quad (2.49)$$

La sua soluzione avviene tramite *Singular Value Decomposition*.

Ovvero:

1. si trova la decomposizione ai valori singolari di \mathbf{Q} :

$$\mathbf{Q} = \mathbf{U} \text{diag}(\sigma_1, \dots, \sigma_p, 0 \dots, 0) \mathbf{V}^T \quad (2.50)$$

2. si trova il minimo valore singolare di \mathbf{Q} :

$$\sigma_i = \text{argmin}(\sigma_1, \dots, \sigma_p) \quad (2.51)$$

3. si determina il punto \mathbf{P}_a come *i-esima* colonna della matrice \mathbf{V} ; la matrice \mathbf{V} è una matrice $4 \times k$, quindi restituisce \mathbf{P}_a in coordinate omogenee. Di conseguenza, per garantire che il quarto elemento della *i-esima* colonna di \mathbf{V} sia 1, si divide tutto il vettore per quella quantità:

$$\mathbf{P}_a = \frac{\mathbf{V}(:, i)}{\mathbf{V}(4, i)} \quad (2.52)$$

2.1.4 – Errori Strumentali e Compensazione

Le traiettorie dei punti ricostruiti sono inevitabilmente affette da errori associati con lo specifico protocollo sperimentale utilizzato.

Gli errori riferiti al sistema fotogrammetrico includono: l'errore con cui vengono ricostruite le traiettorie istantanee dei *marker* nella calibrazione stereofotogrammetrica, la propagazione di questo errore alla posizione e all'orientamento stimati del segmento corporeo e, infine, alle coordinate stimate del generico punto nello spazio tridimensionale.

Le componenti d'errore che affliggono le posizioni ricostruite dei *marker* sono gli errori strumentali casuali (IRE) e sistematici (ISE).

Tra gli errori casuali vi sono l'errore elettronico, il *flickering*, dovuto all'imprecisione con cui vengono ricostruiti i punti immagine della posizione dei

marker, e la quantizzazione nel processo di digitalizzazione che trasforma le coordinate dei *marker* nei loro valori numerici; tali errori sono in genere assunti essere a media nulla e vengono superati con tecniche di filtraggio.

Gli errori di natura sistematica, invece, sono causati dalla validità limitata del modello di sistema, dall'imprecisione della calibrazione e dalle non-linearità non considerate nella calibrazione fotogrammetrica (come ad esempio la distorsione dovuta all'ottica delle lenti). Tali errori variano con la posizione istantanea del *marker* nel volume di acquisizione e per essere compensati necessitano dell'esecuzione di *spot - check* ad hoc, per ottenere una misura precisa di accuratezza e precisione della misura della posizione dei *marker*.

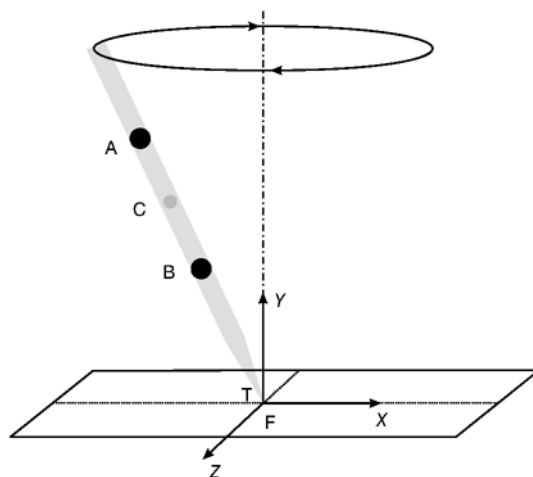
L'esecuzione di tali *spot - check* è una pratica comune nei laboratori di analisi del movimento, viene effettuata prima di ogni sessione sperimentale così da poter stimare gli errori e valutarne correttamente la propagazione nei dati raccolti per la stima delle variabili cinematiche.

Un metodo classico per la stima dell'errore strumentale in un sistema stereofotogrammetrico è il MAL test, introdotto da Cappozzo e Della Croce nel 2000 [11].

2.1.4.1 - MAL test

Il MAL test, fornisce informazioni quantitative di accuratezza e precisione con cui viene ricostruita la posizione istantanea di un punto nello spazio, e permette l'identificazione di diverse componenti di errore e le loro proprietà.

Il MAL test è basato sulla registrazione del movimento di una bacchetta rigida su cui sono montati due *marker* sferici (A e B). Un punto target (T) viene assunto in posizione nota rispetto agli altri due *marker*. Questo viene posizionato in un punto fisso del sistema (F).



2.1.7 - Bacchetta utilizzata nel test e movimento da effettuare durante l'esperimento

L'insieme cluster - punto simula un segmento corporeo. Il test deve essere fatto nelle medesime condizioni nelle quali vengono svolte le acquisizioni sperimentali, in particolare per quel che riguarda il *frame rate* e la velocità media di spostamento dei *marker*.

Per l'esperimento vengono fatte delle acquisizioni in statica, con la bacchetta ferma, e in dinamica, con la bacchetta che viene fatta ruotare intorno a T, muovendo l'altra estremità lungo una traiettoria pseudo circolare o su due archi ortogonali. La manovra dinamica viene eseguita manualmente, ad una velocità comparabile con quella dell'esercizio fisico in esame, come specificato precedentemente.

Per ogni acquisizione vengono determinate le seguenti quantità:

- La geometria dell'oggetto, ovvero le distanze relative fra i punti A,B e T, tramite misurazione diretta;
- Le coordinate del punto F rispetto al laboratorio, tramite misurazione diretta;
- Le coordinate 3D dei punti A e B, ricostruite tramite il sistema stereofotogrammetrico, sia in statica che in dinamica.

Per ogni istante temporale vengono calcolate le deviazioni standard per gli ISE e

gli IRE e viene ottenuta una stima del *bias* degli ISE; è, a questo punto, possibile quantificare la precisione e l'accuratezza con cui viene ricostruito il punto T nello spazio, tramite delle semplici equazioni lineari. [11]

Capitolo 3

Materiali e Metodi

Nel presente capitolo si descrive come è stato realizzato un *toolbox* in grado di gestire e processare dati video, fare *tracking* di *feature* e creare infine un file in formato C3D (formato standard per la ricostruzione di dati 3D in analisi del movimento (1.2)), partendo da un precedente *toolbox*, sviluppato presso il Laboratorio di Bioingegneria del Movimento, ideato inizialmente per il solo *Feature Tracking*. I passi per il raggiungimento di tale scopo sono stati:

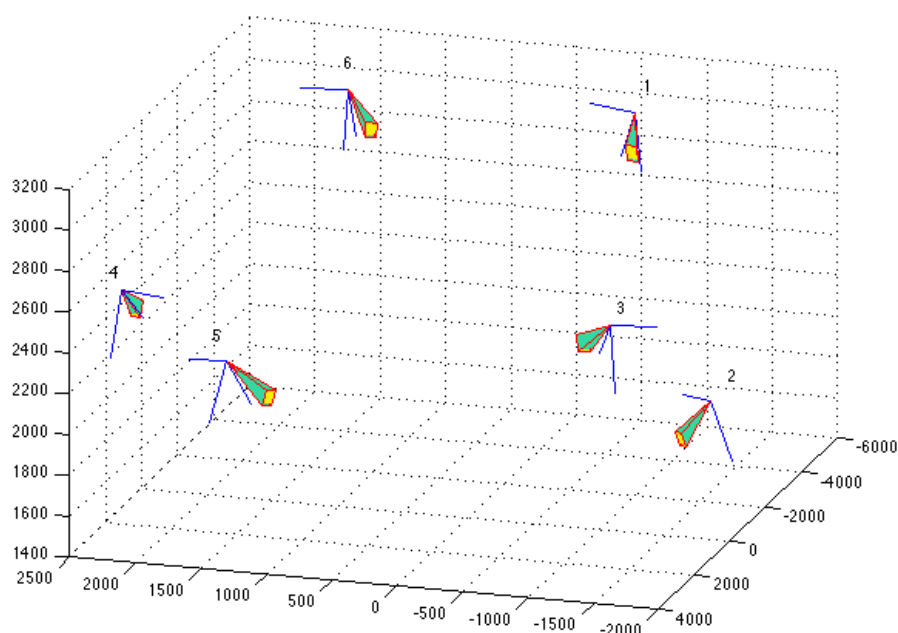
1. Selezione di due tipologie di video disponibili presso il Laboratorio di Bioingegneria del Movimento; in particolare: video acquisiti con telecamere subacquee e video acquisiti con telecamere commerciali a secco. Per la prima tipologia di video si è scelto il gesto del cammino in acqua, per la seconda tipologia di video si è scelto il gesto della battuta nella pallavolo;
2. Esecuzione di una fase di preelaborazione dei dati video, comune per entrambe le tipologie di cui sopra, che prevede due fasi di elaborazione: la correzione della distorsione dei video e la calibrazione delle telecamere. A causa della differente frequenza di acquisizione delle telecamere commerciali utilizzate per le acquisizioni a secco, si è resa necessaria, per quest'ultime, un'ulteriore operazione di sincronizzazione dei dati ottenuti dalle varie telecamere. Operazione non necessaria per i video subacquei, in quanto il set up di acquisizione prevede l'utilizzo della stessa tipologia di telecamere;

3. Sviluppo di un algoritmo, in ambiente MatLab, che scriva in formato C3D le traiettorie triangolate dei *marker*, nonché i parametri di calibrazione, intrinseci ed estrinseci, delle telecamere;
4. Sviluppo di un algoritmo, in ambiente MatLab, per la correzione della distorsione ai dati video, prima della procedura di *Feature Tracking*, e per la verifica dell'efficacia dello stesso tramite confronto tra i dati 3D ricostruiti applicando la correzione della distorsione prima e dopo l'operazione di *tracking*;
5. Validazione del metodo sviluppato, tramite confronto con un *gold standard* stereofotogrammetrico. A tal fine si è utilizzato il Mal Test (2.1.4.1), e sono state ricostruite le traiettorie dei marcatori sia tramite stereofotogrammetria che tramite *tracking di feature* individuate a video con il *toolbox*; per il confronto tra le traiettorie dei marcatori acquisiti durante il Mal Test, si è utilizzato il calcolo della *Root Mean Square Distance* (RMSD);
6. Verifica della validità del metodo sviluppato tramite elaborazione dei dati video, a secco e subacquei, precedentemente descritti. Gli stessi dati sono stati utilizzati anche per la verifica di cui al punto 4. Si è quindi confrontata la cinematica intersegmentale ottenuta applicando la correzione della distorsione prima e dopo l'operazione di *tracking* in termini di traiettorie spaziali dei *marker* e angoli articolari ricostruiti. Per il confronto tra le variabili si è utilizzato il calcolo della *Root Mean Square Distance* (RMSD).

3.1 - Set up Sperimentale

3.1.1 - MAL Test

Il test è stato effettuato all'interno del Laboratorio di Analisi del Movimento del Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova. Il laboratorio contiene un sistema stereofotogrammetrico con sei telecamere a infrarossi che, per l'esperimento in oggetto, sono state disposte come in Figura 3.1.1.



3.1.1 - Disposizione delle telecamere nel laboratorio stereofotogrammetrico

L'apparecchio stereofotogrammetrico presente nel laboratorio è lo SMART Motion Capture System, sviluppato dalla BTS, Padova. È costituito dalle seguenti parti [33]:

- Videocamere: sei videocamere digitali che utilizzano sensori CCD (1/2" F 1.4/4.5 mm), dotate di filtri IR (figura 3.2). Possono acquisire frame a due frequenze diverse (60/120 Hz), la cui risoluzione video è pari a 640x480 pixel.

- Illuminatori: sei illuminatori, ciascuno dei quali è montato su una videocamera; sono a luce infrarossa ad alta potenza e controllo digitale. La durata di ciascun impulso luminoso è di $250\mu\text{s}$ ed è ottimizzata per evitare inconvenienti con la luce solare e l'illuminazione dell'ambiente.
- Lenti: un set completo di sei lenti da 3.5mm e due zoom da 6-12mm; sono interscambiabili, permettendo una maggiore flessibilità. La messa a fuoco ed il diaframma sono regolabili manualmente.
- HUB: due M-HUB digitali, che forniscono alimentazione e sincronismo alle videocamere e agli illuminatori. Trasmettono il segnale digitale per il controllo della durata e della potenza dell'impulso degli illuminatori ed il segnale per il controllo digitale dell'otturatore in modo sincronizzato con la gestione degli impulsi degli illuminatori. Ciascuno di questi HUB è in grado di gestire un numero massimo di tre videocamere.
- Cavi: cavi di connessione per sincronizzare tra loro i vari HUB e per collegare a questi le videocamere, in modo tale da consentire la trasmissione del segnale di sincronismo, di controllo dell'otturatore e di controllo degli illuminatori e la ricezione di segnali video e dati.
- Workstation: si tratta di un personal computer nel quale sono inserite una scheda PCI di acquisizione e conversione analogico-digitale (National Instruments NI 6023E) e due frame *grabber* PCI (Matrox Meteor II). La prima è collegata all'uscita dell'hub analogico, mentre le seconde ricevono le uscite dei due M-HUB digitali.



3.1.2 - Telecamera con illuminatori ad infrarosso presente nel sistema stereofotogrammetrico [34]

Il MAL test (2.1.4.1) è stato eseguito per mezzo di una bacchetta con 3 *marker* a distanza nota rappresentata in Figura 3.1.3. I *marker* erano ricoperti da una pellicola retroriflettente.



3.1.3 - Bacchetta utilizzata per il MAL test

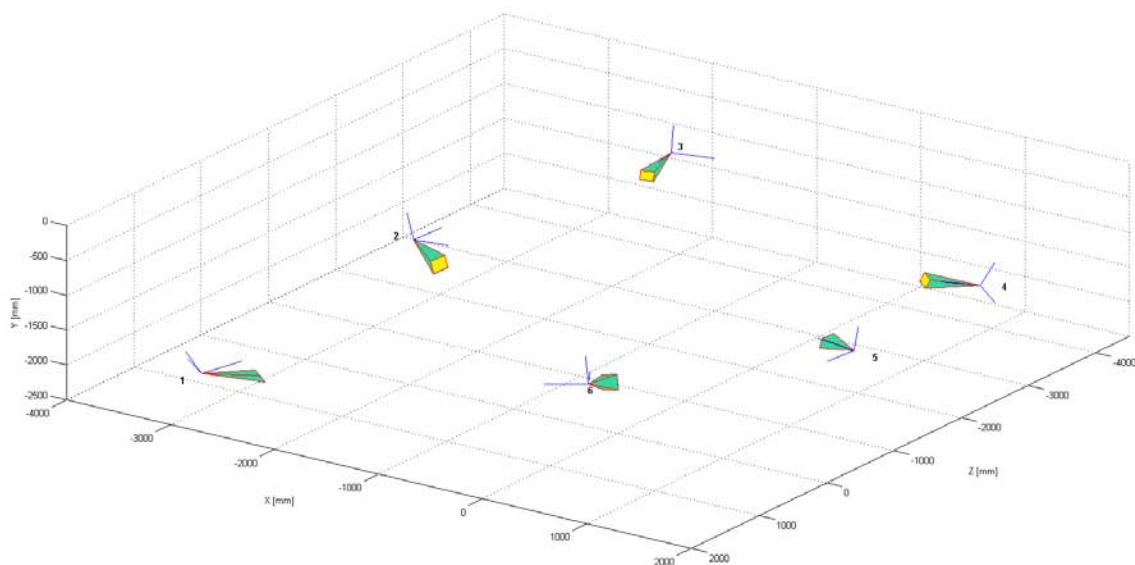
Sono state eseguite varie dinamiche contenenti dei movimenti elementari della bacchetta come descritto in 2.1.4.1. I dati sono stati acquisiti simultaneamente effettuando l'acquisizione dei dati avvalendosi sia del metodo stereofotogrammetrico, quindi utilizzando la ricostruzione 3D delle traiettorie dei *marker* rifrangenti all'infrarosso, sia con il metodo *markerless*, quindi acquisendo le immagini in scale di grigio.

3.1.2 – Acquisizioni Subacquee

I dati video subacquei sono relativi ad una camminata eseguita presso la piscina Record di Bologna, in collaborazione con il laboratorio del Movimento del Dipartimento di Scienze Motorie dell'Università di Bologna, utilizzando sei telecamere subacquee uguali (TS-6021PSC, Tracer Technology Co. Ltd) (Tab. 3.1.1 e fig 3.1.4). Per consentire l'acquisizione simultanea e sincrona dei dati, ogni telecamera è stata collegata ad un convertitore analogico - digitale (Canopus ADVC55; uscita video DV, PAL interlacciato, 25 frame/s), collegato a sua volta ad un notebook con un collegamento FireWire (standard IEEE 1394). L'intero sistema è stato sincronizzato in modo automatico impiegando un'applicazione software, DVApp (sviluppata presso il laboratorio di Analisi del Movimento dell'Università di Padova [29], installata su ogni computer, che invia attraverso una rete locale dei segnali d'inizio e di arresto acquisizione.

<i>Telecamera</i>	<i>Marca e modello</i>
Cam 1	Matr. AA138652 (<i>Asus</i>) PC1
Cam 2	Matr. AA120692 (<i>Toshiba - Server</i>) 0
Cam 3	Matr. AA120688 (<i>Toshiba</i>) PC2
Cam 4	Matr. AA120695 (<i>Asus</i>) PC3
Cam 5	Matr. AA120679 (<i>Asus</i>) PC4
Cam 6	Matr. AA138653 (<i>Asus</i>) PC0

Tabella 3.1.1 - Marca e modello delle telecamere utilizzate



3.1.4 - Disposizione delle telecamere nel setup Piscina

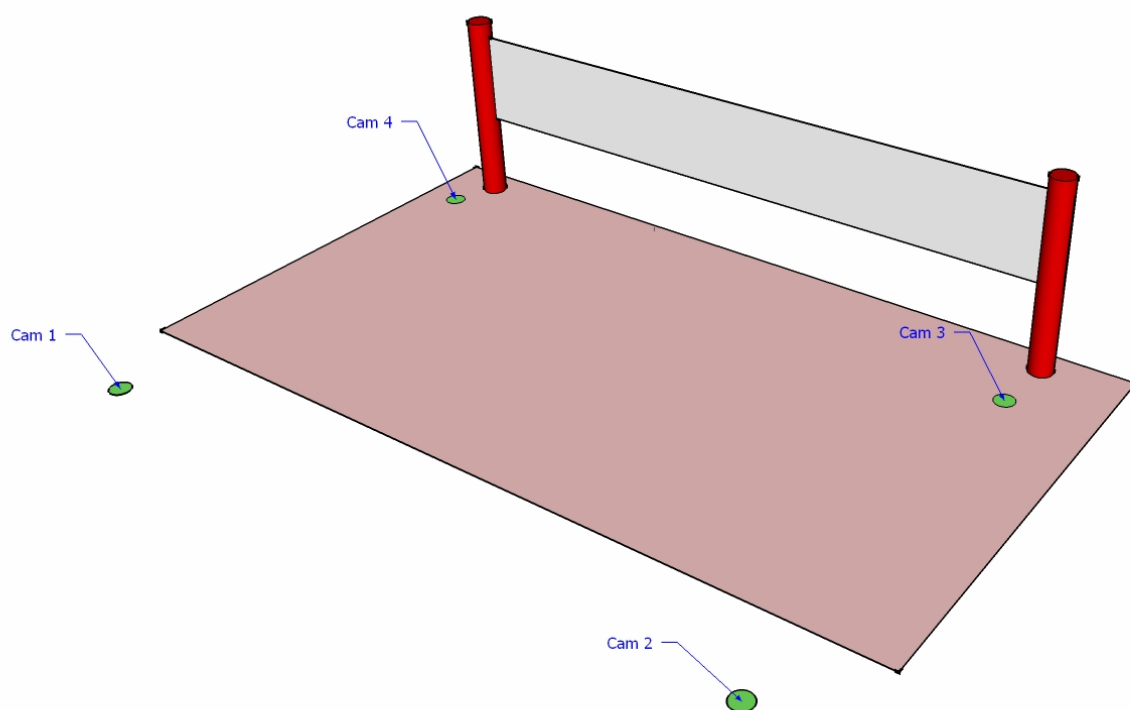
3.1.3 Acquisizioni a Secco

Nella tesi in oggetto sono stati elaborati anche dei dati video relativi alla battuta nella pallavolo, di un atleta *d'elite*, acquisiti all'interno di un progetto di collaborazione tra il Laboratorio di Analisi del Movimento del Dipartimento di Ingegneria dell'Informazione dell'Università di Padova (gruppo di ricerca di Bioingegneria), e la squadra maschile della società S.S.C.D. Pallavolo Padova S.r.l., che al tempo militava in Serie A2, presso il palazzetto dello sport PalaFabris di Padova, con 4 telecamere commerciali, connesse a quattro personal computer. La descrizione dei dati utilizzati è riportata in Tabella 3.1.2 e Figura 3.1.5.

Anche in questo caso, per l'acquisizione dei dati, ogni telecamera è stata collegata ad un convertitore analogico - digitale (Canopus ADVC55; uscita video DV, PAL interlacciato, 25 frame/s), collegato ad un notebook tramite collegamento FireWire (standard IEEE 1394), per il supporto del trasferimento di dati sincrono e simultaneo. Per la sincronizzazione del sistema, si è fatto uso nuovamente dell'applicazione software DVApp, di cui sopra.

<i>Telecamera</i>	<i>Marca e Modello</i>
Cam 1	Canon Legria fs 200
Cam 2	Sony dcr sx 21 e
Cam 3	Sony dcr sr 35
Cam 4	Sony dcr sr 90

Tabella 3.1.2 - Marca e modello delle Telecamere utilizzate



3.1.5 - Disposizione delle telecamere nel setup Pallavolo

3.1.4 - Video Processing

I video in uscita dai convertitori A/D si presentano interlacciati (vedi specifiche sopra); nonostante il ridotto ingombro in memoria, questa tecnica di scansione introduce degli artefatti dovuti alla maggiore velocità del movimento registrato rispetto al *frame rate* della telecamera. L'interlacciamento dei dati video prevede la divisione delle linee di scansione in due parti, dette campi o semiquadri, suddivisi in linee pari e dispari.

In particolare lo standard PAL visualizza 50 semiquadri al secondo (25 pari e 25 dispari). Un quadro completo, quindi, viene tracciato 25 volte al secondo utilizzando entrambi i semiquadri (pari e dispari) che però sono stati registrati in istanti leggermente differenti (un cinquantesimo di secondo nella fattispecie). Se il movimento registrato ha una velocità superiore, i due semiquadri rappresentano due immagini consecutive e quindi non identiche, da qui l'origine degli artefatti (fig. 3.1.6).

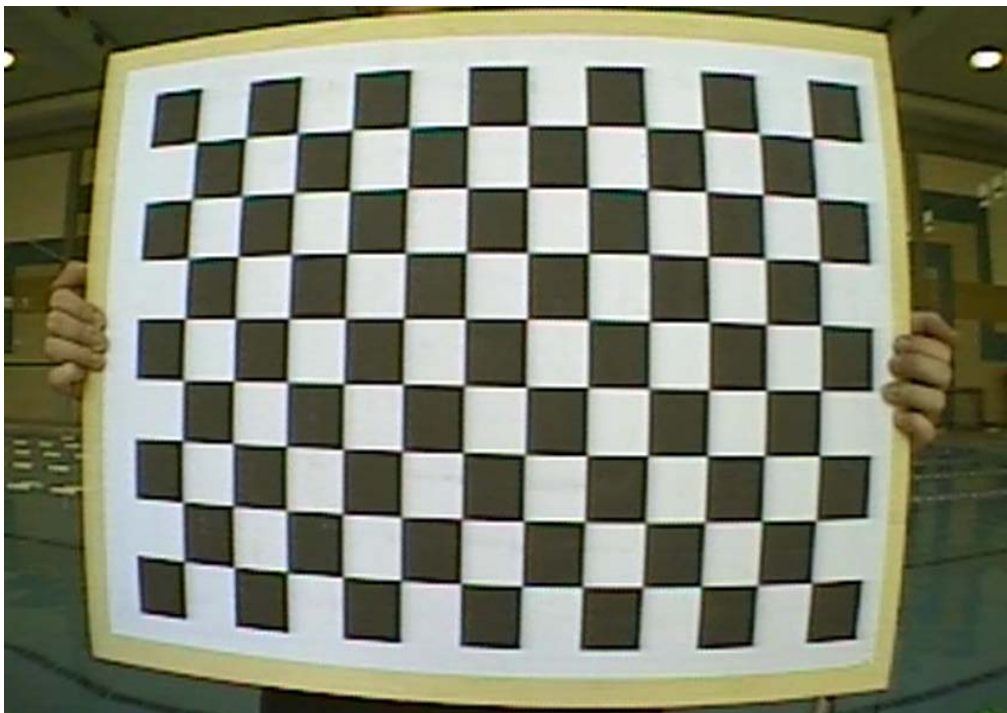


3.1.6 - Esempio di frame interlacciato (sinistra) e deinterlacciato (destra)

Per minimizzare tali artefatti, ogni video è stato "deinterlacciato", tramite il filtro *smart deinterlacer* del software VirtualDub sviluppato da Donald Graft (<http://neuron2.net/bob.html>).

3.1.5 - Calibrazione Telecamere

Le telecamere sono state calibrate per ricavarne i parametri fondamentali, come descritto in (2.1.1). L'operazione è stata resa possibile grazie al software MMC - Suite, sviluppato presso il Laboratorio di Analisi del Movimento dell'Università degli Studi di Padova [30], in cui è implementata la *Calibration Toolbox* di J.Y. Bouget [24]. Le calibrazioni intrinseche sono state effettuate per ogni telecamera,



3.1.7 - Esempio di griglia utilizzata per l'estrazione dei parametri di calibrazione intrinseca

a secco, riprendendo in diverse posizioni dello spazio una scacchiera di cui erano note le dimensioni (scacchiera 13x9 cm; lato quadrati 42 mm), come indicato da Bouget [24]. Per ogni frame il software riconosce i vertici dei quadrati in modo automatico e attraverso il calcolo delle distanze tra i vertici, note le distanze reali, fornisce i parametri intrinseci e l'errore di riproiezione. Per le riprese in acqua si è applicato un fattore di correzione dovuto al differente indice di rifrazione tra aria e acqua, in particolare è stata moltiplicata la lunghezza focale per un opportuno parametro (1,33), come indicato da Lavest nel lavoro dedicato [26].

Per la calibrazione estrinseca invece si sono seguite due procedure differenti. Per le acquisizioni subacquee è stata ripresa una griglia (fig. 3.1.7) di dimensioni note

(2,07 x 1,07 x 1,40 m), di cui sono stati mappati sia i vertici che dei punti prestabiliti all'interno di ogni lato del parallelepipedo, in ogni frame. Mentre per le acquisizioni di pallavolo si sono segnate a terra le dimensioni della scacchiera, di cui sopra, e ne sono stati mappati anche in questo caso i vertici frame per frame, essendo note le dimensioni reali. Una volta completata l'operazione, sia in laboratorio che in piscina, è stato definito il sistema di riferimento globale.



3.1.8 - Esempio di griglia e scacchiera utilizzate per la calibrazione estrinseca delle telecamere

Infine è stata valutata la precisione della calibrazione grazie a due parametri ad hoc: l'errore di calibrazione e l'errore di ricostruzione. Il primo è stato determinato per ciascuna telecamera come la distanza quadratica media tra la posizione dei punti di controllo misurati e le proiezioni 3D corrispondenti dei punti di controllo, mentre l'errore di ricostruzione è stato calcolato come la distanza quadratica media tra le coordinate note dei punti di controllo 3D e i corrispondenti punti ricostruiti dalle misurazioni delle proiezioni 2D.

3.2 - Rimozione della Distorsione e Feature Tracking

3.2.1 - Correzione della distorsione

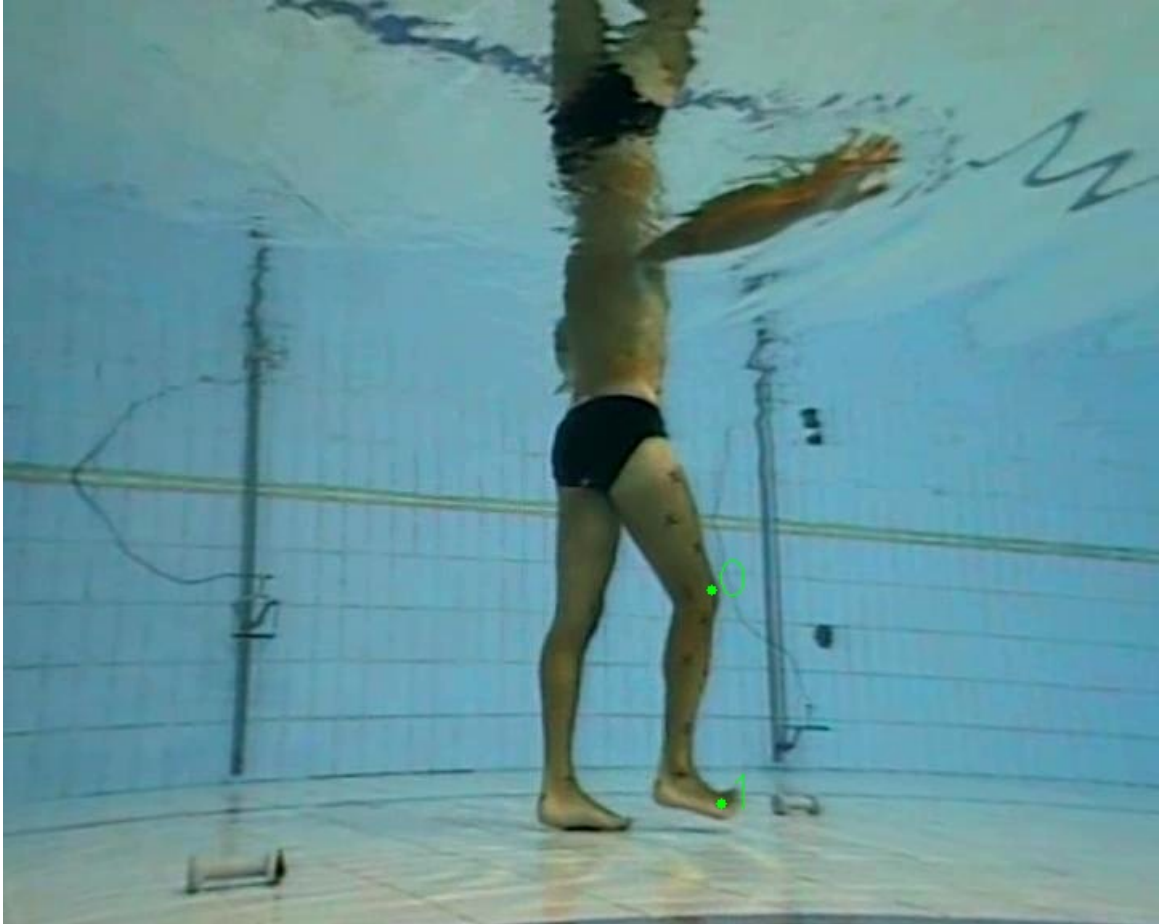
La procedura di rimozione della distorsione è stata eseguita mediante uno script creato ad hoc all'interno di questo lavoro di tesi, eseguibile nella suite MatLab, utilizzando i metodi implementati nella *Camera Calibration Toolbox for MatLab* proposta da J. Y. Bouget [24].

3.2.2 - Implementazione del Feature Tracker

L'algoritmo utilizzato [7] è stato sviluppato per eseguire il *tracking* automatico dei *marker* passivi, fornendo una semplice interfaccia utente per l'implementazione del *tracker* KLT proposta da Birchfield [27][28].

Il programma richiede che venga fornita la posizione iniziale di una *feature*; da questa il *tracker* automatico utilizza il gradiente di intensità spaziale per cercare il vettore ottimo che minimizza la differenza di intensità luminosa fra gli intorni della *feature* in fotogrammi adiacenti [6][7].

In breve, l'algoritmo punta a trovare lo spostamento di punti specifici tra due immagini consecutive. Nell'ipotesi di luminosità costante, le variazioni di intensità dell'immagine sono dovute esclusivamente al movimento; pertanto, l'intensità luminosa di un punto deve corrispondere all'intensità del medesimo punto nell'immagine seguente. Tale corrispondenza può essere espressa in funzione dello spostamento avvenuto sul piano immagine. Il valore dello spostamento viene calcolato minimizzando l'errore residuo, dopo aver approssimato la funzione intensità con la sua espansione in serie di Taylor troncata ai termini del primo ordine. Il passo di minimizzazione viene poi ripetuto iterativamente in stile Newton-Raphson. L'algoritmo ivi presentato, e utilizzato, ha il vantaggio di essere facilmente implementabile con codici open source, disponibili presso la Intel Open Source Computer Vision Library (<http://software.intel.com/en-us/articles/intel-integrated-performance-primitives-intel-ipp-open-source-computer-vision-library-opencv-faq>).



3.2.1- Digitalizzazione manuale dei marker con Feature Tracker

3.2.3 - Triangolazione

Le coordinate bidimensionali di tutti i punti, estratte con la procedura di *Feature Tracking* per ogni telecamera, sono state triangolate così da ottenere le coordinate spaziali.

Per la procedura di triangolazione, si è seguito il metodo lineare non omogeneo (2.1.3). Per le caratteristiche dell'algoritmo di triangolazione è necessario fornire in input i parametri intrinseci anche ai video dove sia già stata effettuata la correzione della distorsione.

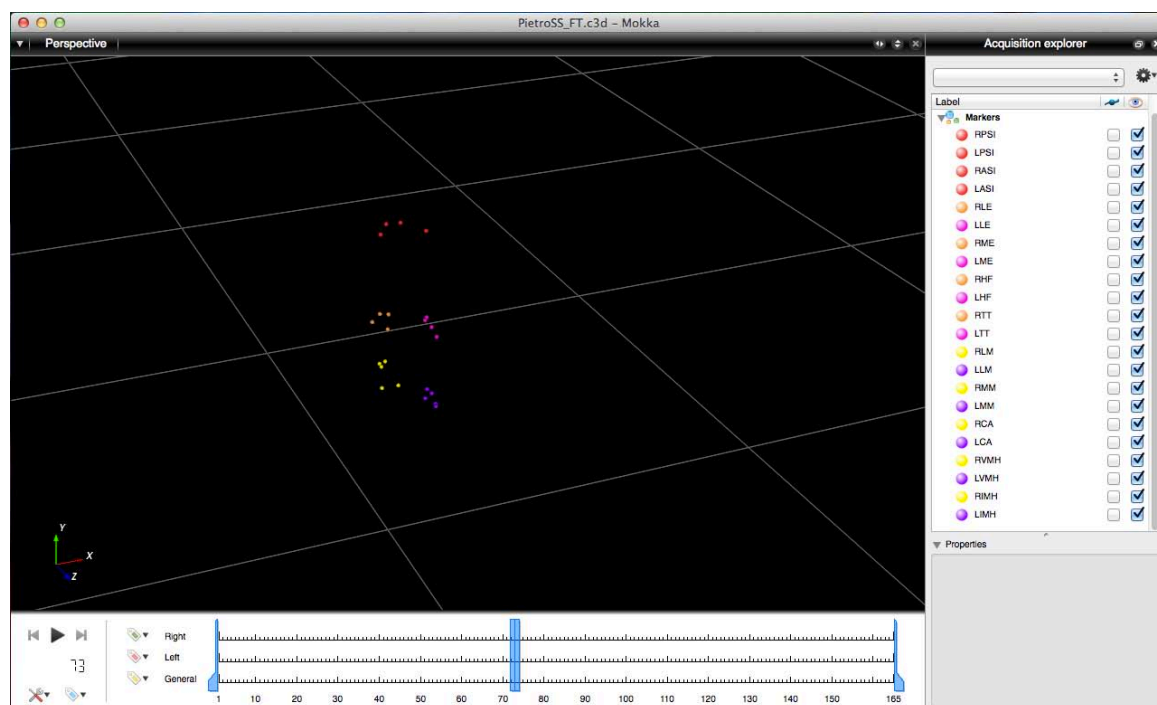
Per la visualizzazione dei risultati, le coordinate sono state esportate nel formato C3D.

3.3 - Algoritmo per la Scrittura dei Dati Triangolati in Formato File C3D

Per l'esportazione delle coordinate in formato C3D, è stata utilizzata la *toolbox* per MatLab presente nell'eseguibile *c3dserver*, scaricabile dal sito (<http://www.c3dserver.com/>).

Come segnalato nel sito, alla sezione *MATLAB access*, è stato necessario utilizzare una versione di MatLab a 32 bit, installata in un sistema operativo a 64 bit. È stato quindi creato un script ad hoc che permettesse la creazione di un file.c3d contenente le coordinate tridimensionali di ogni singolo *marker*, denominato con un'apposita *label*.

Una volta creati i file è stato possibile visualizzarli attraverso il software Mokka (MOtion Kinematic & Kinetic Analyzer).



3.3.1 - Screenshot del software Mokka con i dati utilizzati

Mokka è un software open - source che permette di leggere, scrivere, modificare e visualizzare dati ottenuti da *motion capture* (traiettorie, modelli, eventi, dati analogici, piattaforme di forza). È stato progettato appositamente per applicazioni delle scienze del movimento, come *gait analysis*, ricerca biomeccanica e analisi sportiva. È un software compatibile con molti tipi di formato presenti in biomeccanica, anche di software proprietari (BTS, Vicon, Motion Analysis, Ascension Technology Corporation etc).

3.4 – Elaborazione dati MAL Test

Il sistema descritto precedentemente (3.1.1) consente di effettuare l'acquisizione dei dati avvalendosi sia del metodo stereofotogrammetrico, quindi utilizzando la ricostruzione 3D delle traiettorie dei *marker* rifrangenti all'infrarosso tramite *blob analysis* e sogliatura (1.1.1.1), sia con il metodo *markerless*, quindi ricostruendo le traiettorie spaziali a partire dalle immagini salvate in scale di grigio. Nel primo caso la ricostruzione delle traiettorie dei *marker* è avvenuta in modo automatico, tramite il software Smart Tracker per *tracking* stereofotogrammetrico; nel secondo caso è stato effettuato *tracking* a video tramite *Feature Tracker* sia su dati video distorti che non distorti. L'esperimento è stato svolto valutando le distanze relative fra i *marker* ricostruiti nelle due acquisizioni e con le due modalità di correzione della distorsione (prima e dopo triangolazione).

Di ogni distanza relativa sono state calcolate media, deviazione standard e RMSD

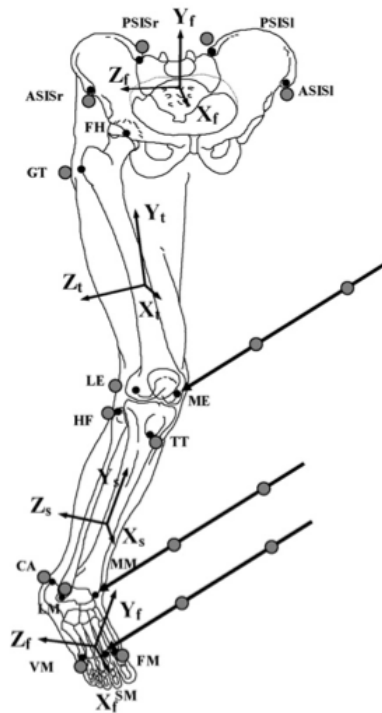
3.5 – Elaborazione dati da Video Subacquei

Al soggetto in esame è stato applicato come *marker - set* il protocollo definito IOR - Gait [31]. Tale protocollo prevede l'individuazione dei punti di reperi attraverso il posizionamento dei *marker* e la calibrazione anatomica con i cluster. I punti di reperi identificati direttamente con il posizionamento sono riportati in Tabella 3.5.1

Struttura Ossea	Descrizione	Acronimo
<i>Bacino</i>	Spina iliaca superiore anteriore destra	RASIS
	Spina iliaca superiore anteriore sinistra	LASIS
	Spina iliaca superiore posteriore destra	RPSIS
	Spina iliaca superiore posteriore sinistra	LPSIS
<i>Femore</i>	Gran Trocantere	GT
	Epicondilo Femorale Laterale	LE
<i>Gamba</i>	Apice Esterno della Testa della Fibula	HF
	Prominenza della Tuberosità Tibiale	TT
	Malleolo Laterale	LM
<i>Piede</i>	Calcagno	CA
	Quinta Testa Metatarsale	VM
	Prima Testa Metatarsale	IM

Tabella 3.5.1 - Marker-set del protocollo IOR Gait [31]

A partire dai punti citati pocanzi è possibile poi calibrare anatomicamente il centro dell'anca, l'epicondilo femorale mediale e il malleolo mediale.

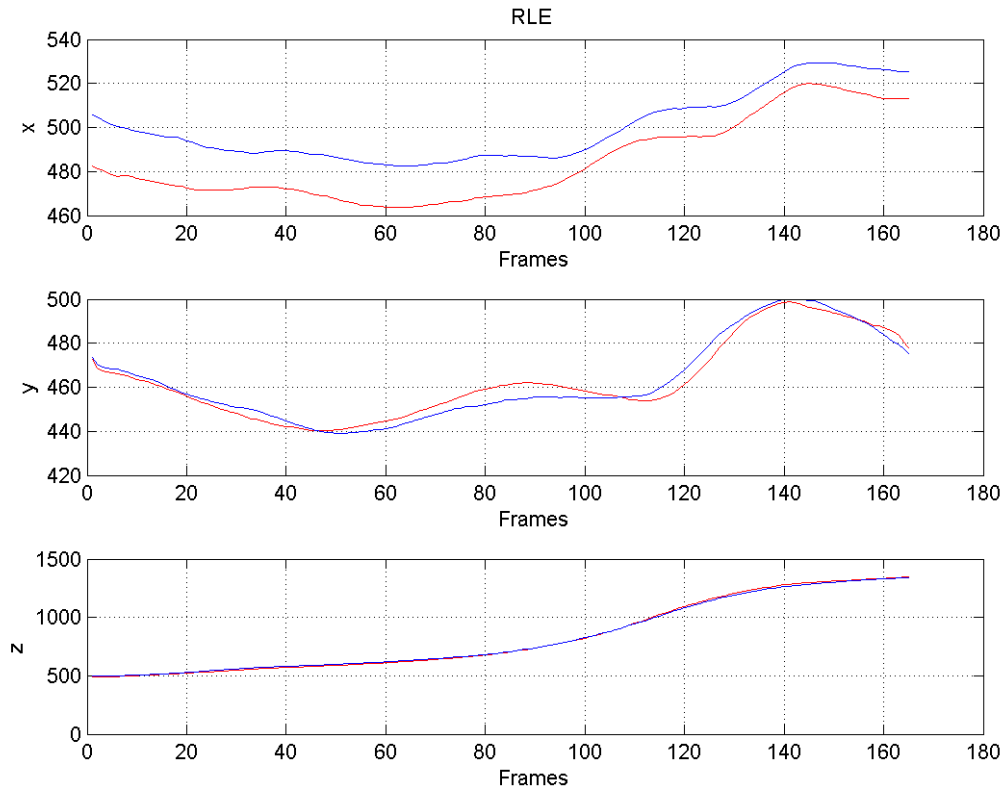


3.5.1 - Rappresentazione della posizione dei punti di reperi anatomico utilizzati nel protocollo IOR-gait per la gamba destra (per la gamba sinistra la disposizione è analoga) [31]

I punti di reperi sono stati digitalizzati a video con il *Feature Tracker* e successivamente sono state calcolate le traiettorie spaziali dei punti.



3.5.2 - Esempio di inseguimento a video dei punti di reperi in acquisizioni subacquee



3.5.3 – Esempio di traiettorie ricostruite per l’epicondilo laterale destro: in rosso la traiettoria ricostruita da video non distorto, in blu la traiettoria ricostruita da video distorto

Le traiettorie finali sono state confrontate tramite la loro *Root Mean Square Distance (RMSD)*.

3.5.1 – Calcolo degli Angoli Articolari

Gli angoli sono stati calcolati come in [31], nel quale viene applicata la convenzione di Grood & Suntay [35]. Tale convenzione è la più utilizzata per rappresentare gli angoli articolari e prevede di associare gli assi di rotazione riportati in figura 3.5.1, ottenendo una descrizione del movimento fisiologica.

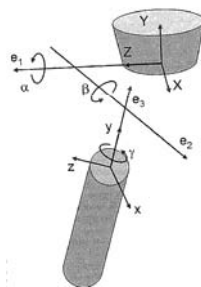
In particolare definendo un sistema di riferimento prossimale e uno distale, si possono definire gli angoli:

- Flesso/Estensione, come rotazione relativa del sistema distale attorno all’asse medio – laterale del sistema prossimale;

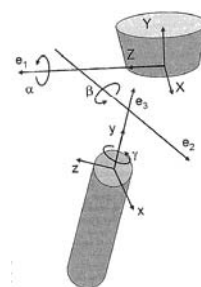
- Ab/Adduzione, come rotazione del sistema relativa distale attorno all'asse flottante (asse antero - posteriore) del sistema prossimale;
- Intra/Extrarotazione, come rotazione relativa del sistema distale attorno all'asse sagittale del sistema prossimale.

Nel caso della caviglia i nomi prendono l'accezione di Flessione dorsale e plantare, Ab/Adduzione, Inversione/Eversione. In figura 3.5.4 si riporta una proposta di standardizzazione per le articolazioni di anca e ginocchio come da [35]

- Asse e_1 solidale con la pelvi e coincidente col suo asse z (z_p) (α angolo di flessione-estensione)
- Asse e_3 solidale con il femore e coincidente col suo asse y (y_d) (γ angolo di rotazione interna-esterna)
- Asse e_2 asse flottante (ortogonale a e_1 ed e_3) (β angolo di ab-adduzione)



- Asse e_1 solidale con il femore e coincidente col suo asse z (z_p) (α angolo di flessione-estensione)
- Asse e_3 solidale con la tibia e coincidente col suo asse y (y_d) (γ angolo di rotazione interna-esterna)
- Asse e_2 asse flottante (ortogonale a e_1 ed e_3) (β angolo di ab-adduzione)



3.5.4 - Definizione del riferimento articolare secondo la convenzione di Grood & Suntay per l'articolazione dell'anca (a sinistra) e del ginocchio (a destra)

Gli angoli calcolati sono stati confrontati con quelli ottenuti da video distorti per mezzo della *Root Mean Square Distance (RMSD)*.

3.6 – Elaborazione dati da Video a Secco

Per l'atleta esaminato nei video a secco è stato utilizzata una combinazione del protocollo IOR - Gait e quello Ceccon et al 2012 [29] per la ricostruzione della cinematica relativa all'arto superiore.

In questo lavoro di tesi si è deciso di analizzare le traiettorie di alcuni specifici punti, per cui i *marker* considerati sono stati (Tabella 3.6.1):

Struttura ossea	Descrizione	Acronimo
Braccio	Acromion Sinistro	LA
	Epicondilo Laterale dell'Omero	LEL

Tabella 3.6.1 - Marker-set utilizzato per il calcolo delle traiettorie a secco

Come descritto precedentemente per i video subacquei, anche in questo caso tali punti sono stati digitalizzati a video tramite *Feature Tracker* e ne sono state calcolate le traiettorie spaziali.



3.6.1 - Esempio di Feature Tracking dell'Epicondilo laterale sinistro

Il confronto tra video distorti e non distorti è stato fatto nuovamente tramite la *Root Mean Square Distance* delle traiettorie in ambo i casi.

Capitolo 4

Risultati e Discussione

Il seguente capitolo riporta i risultati degli esperimenti descritti nel capitolo precedente.

L'obiettivo è evidenziare le differenze fra i risultati ottenuti con la procedura proposta in questa tesi, ovvero rimozione della distorsione a priori sui video e conseguente procedura di *tracking* e triangolazione, rispetto alla procedura che caratteristica dei sistemi commerciali di video *tracking* (es: *Simi Motion*, *Dartfish*, *Kinovea*, *APAS*), ovvero esecuzione del *tracking* sui video dove non è stata rimossa la distorsione e successiva rimozione della distorsione in sede di triangolazione.

4.1 – Elaborazione dati MAL Test

Per il MAL test è stata utilizzata una bacchetta in cui erano fissati tre *marker* a distanza relativa di 30 cm e 15 cm. Vengono quindi presentati i risultati ottenuti con un sistema stereofotogrammetrico (Tabella 4.1.2), utilizzato come *gold standard*, e i risultati ottenuti su video distorti (Tabella 4.1.3) e non distorti (Tabella 4.1.4), in termini di distanza massima, minima, media, deviazione standard e *RMSD*.

<i>Distanze Reali</i>	
<i>d01 (cm)</i>	30,000000
<i>d12 (cm)</i>	15,000000
<i>d02 (cm)</i>	45,000000

Tabella 4.1.1 – Distanze relative reali fra i marker

<i>TDF</i>					
	<i>MAX</i>	<i>MIN</i>	<i>MEDIA</i>	<i>SD</i>	<i>RMSD</i>
<i>d01 (cm)</i>	30,189733	29,861681	30,019968	0,064380	30,020036
<i>d12 (cm)</i>	15,090063	14,895973	14,989964	0,048595	14,990042
<i>d02 (cm)</i>	45,128926	44,876831	45,009611	0,057387	45,009647

Tabella 4.1.2 - Risultati delle distanze relative nell'acquisizione effettuata con stereofotogrammetria

<i>Video Con Distorsione</i>					
	<i>MAX</i>	<i>MIN</i>	<i>MEDIA</i>	<i>SD</i>	<i>RMSD</i>
<i>d01 (cm)</i>	30,475957	28,320805	29,591908	0,646630	29,598852
<i>d12 (cm)</i>	15,521816	14,386565	14,883548	0,215579	14,885083
<i>d02 (cm)</i>	45,427925	43,007393	44,450630	0,721483	44,456386

Tabella 4.1.3 - Risultati delle distanze relative ottenuti con video distorti

<i>Video Senza Distorsione</i>					
	<i>MAX</i>	<i>MIN</i>	<i>MEDIA</i>	<i>SD</i>	<i>RMSD</i>
<i>d01 (cm)</i>	31,232797	21,129615	29,831043	1,415345	29,864032
<i>d12 (cm)</i>	16,933786	10,980755	15,476429	1,017095	15,509249
<i>d02 (cm)</i>	46,492568	32,056317	45,268685	1,777399	45,302974

Tabella 4.1.4 - Risultati delle distanze relative ottenuti con video non distorti

<i>ΔRMSD</i>			
	<i>Reali - TDF</i>	<i>Reali - Distorti</i>	<i>Reali - Non Distorti</i>
<i>d01 (cm)</i>	-0,020036	0,401148	0,135968
<i>d12 (cm)</i>	0,009958	0,114917	-0,509249
<i>d02 (cm)</i>	-0,009647	0,543614	-0,302974

Tabella 4.1.5 - Errore dell' RMSD rispetto ai valori reali

I test sono stati eseguiti nelle migliori condizioni possibili, in quanto ci si trovava all'interno di un laboratorio e il *tracking* a video è risultato particolarmente semplice grazie alle camere ad infrarossi; di conseguenza si può presupporre che questi siano dei valori da considerarsi non raggiungibili, ne tanto meno superabili nella normale pratica di *video tracking*.

Come si può notare dalla Tabella 4.1.5, le differenze tra i risultati ottenuti con video distorti e non distorti, sono favorevoli a questi ultimi per quanto riguarda in particolare le "lunghe" distanze, mentre nelle "brevi" si constata un maggiore avvicinamento del dato distorto. Questo comunque pone delle buone basi per l'analisi effettuata su video reali, in quanto le grandezze in gioco sono per la maggior parte dell'ordine di d01 o d02 o maggiori.

4.2 – Elaborazione dati da Video Subacquei

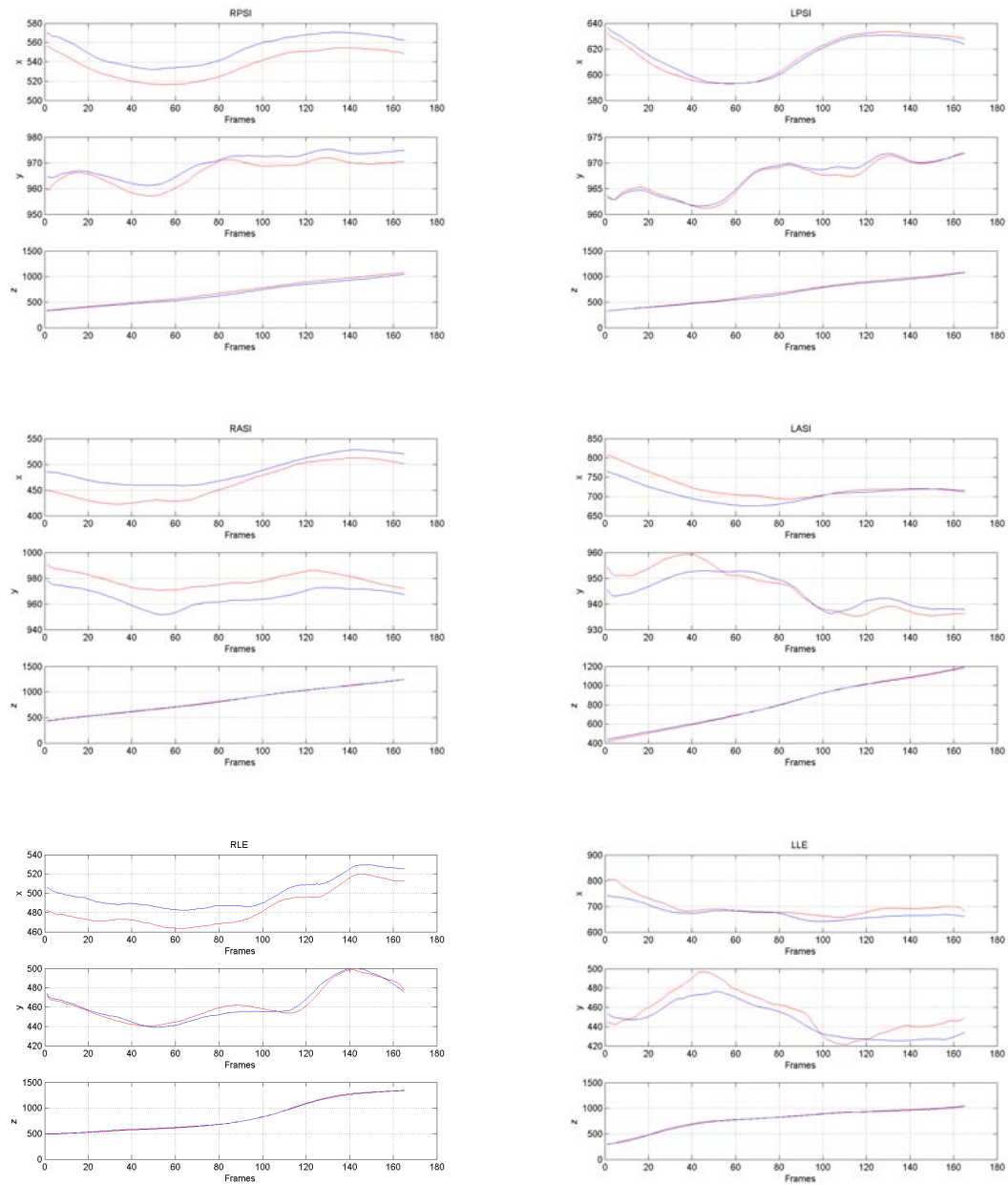
In questa sezione sono stati riportati i confronti fra le diverse traiettorie dei *marker* e tra gli angoli articolari calcolati sia a partire da video distorti che non distorti. Per gli angoli si è riportato anche il confronto con le fasce di normalità calcolate con i sistemi stereofotogrammetrici *marker – based* al fine di fornire un indicazione relativamente all'attendibilità dei risultati in termini di stima dei parametri cinematici. Non si è proceduti a valutare l'RMSD rispetto alle fasce perché questo necessiterebbe la creazione di fasce di normalità a partire da dati video non stereofotogrammetrici, con almeno 10 soggetti come nel caso delle fasce utilizzate, in questa sede. Di seguito in tutti i casi viene riportata in calce la *Root Mean Square Distance* fra i risultati ottenuti con video distorti e con video non distorti.

Sia per gli angoli articolari che per le traiettorie, vengono riportati in rosso i risultati ottenuti senza distorsione, in blu quelli ottenuti con distorsione.

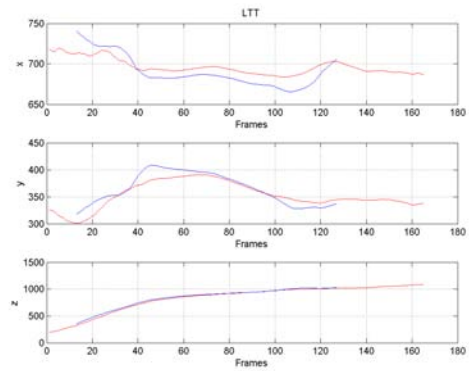
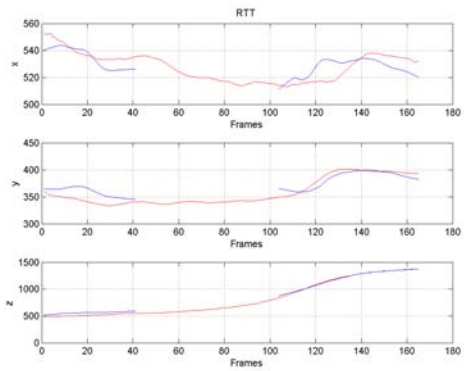
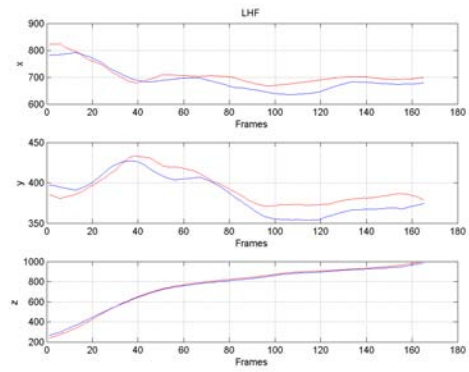
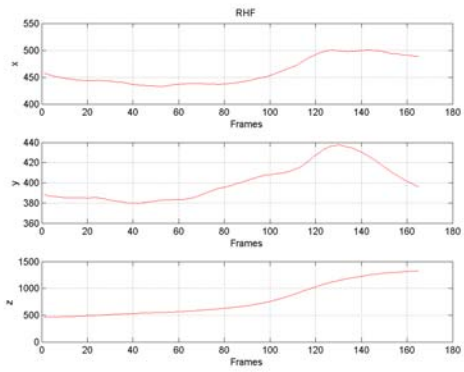
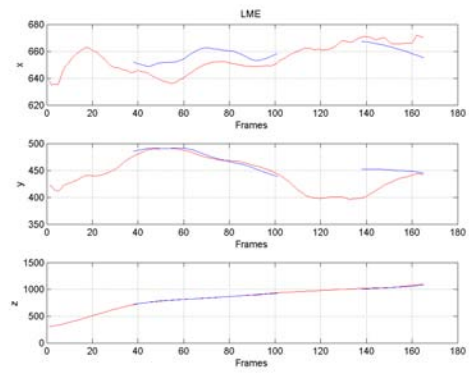
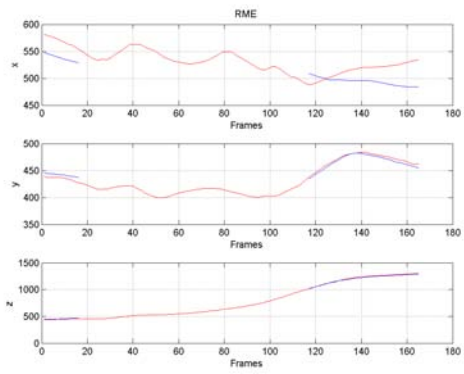


4.2.1 – Legenda per i grafici degli angoli articolari e delle traiettorie dei marker

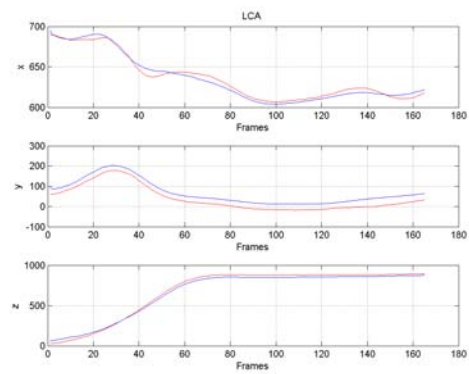
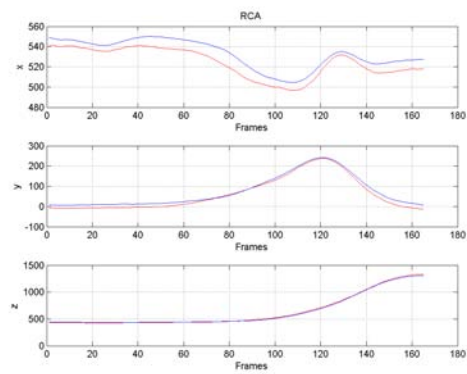
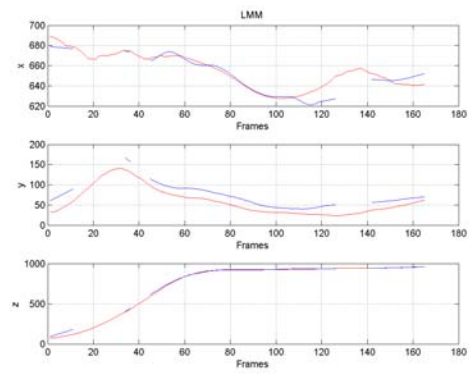
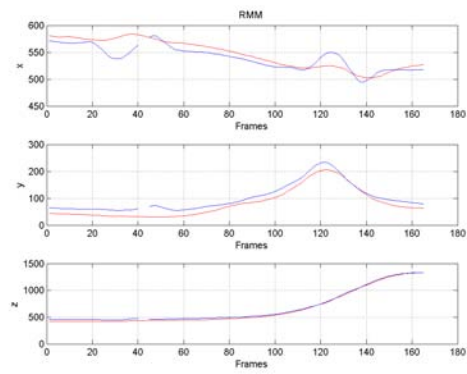
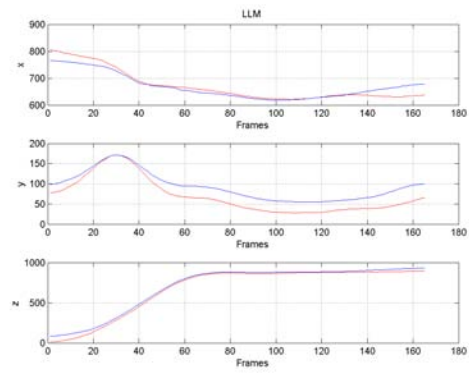
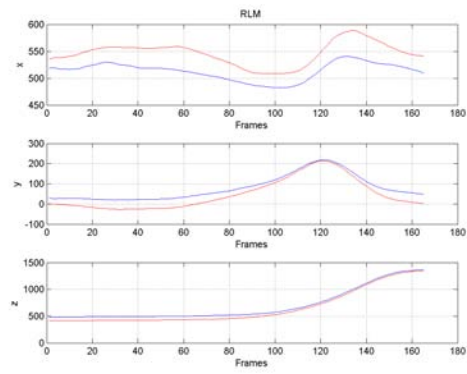
Traiettorie dei Marker



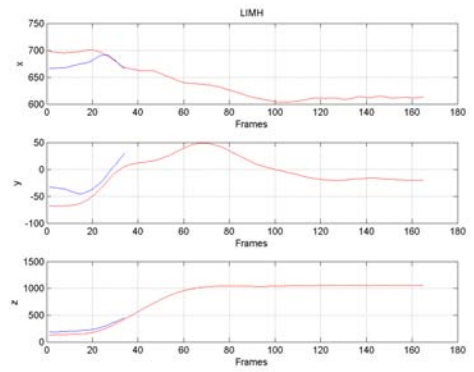
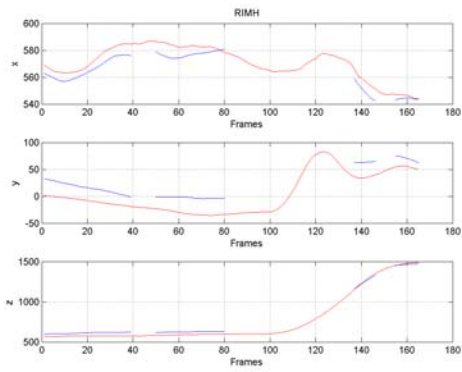
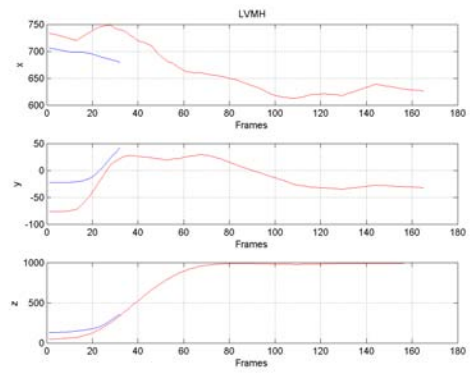
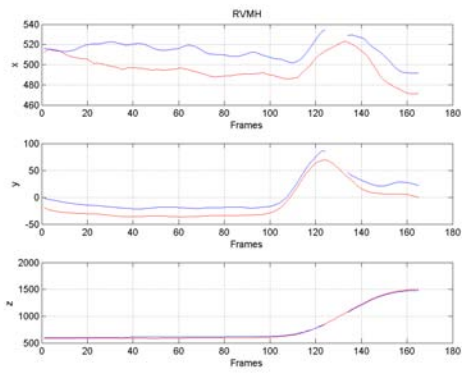
4.2.2 – Traiettorie delle spine iliache e degli epicondili laterali



4.2.3 - Traiettorie degli epicondili mediali, delle teste della fibula e delle tuberosità tibiali



4.2.4 - Traiettorie dei malleoli laterali e mediali e dei calcagni

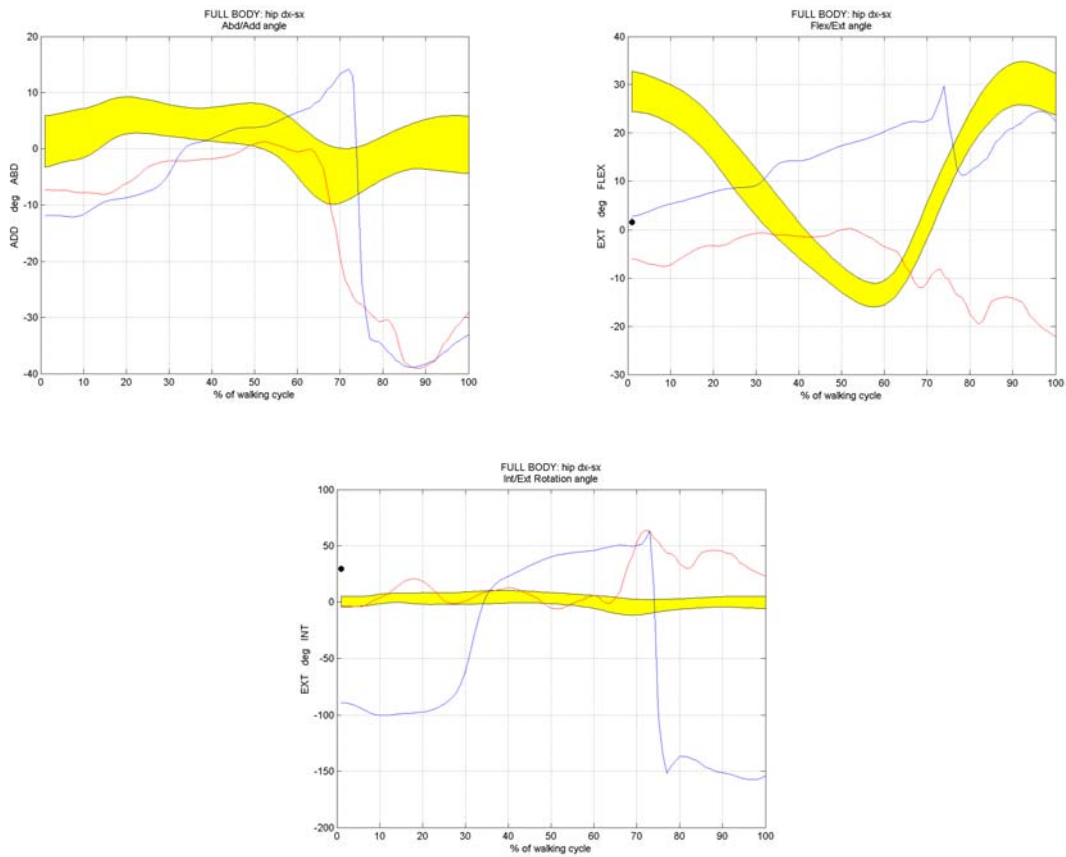


4.2.5 - Traiettorie della quinte e prime teste metatarsali

<i>RMSD Traiettorie</i>			
<i>Marker</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>RPSIS</i>	16,05478696	3,282271863	35,73114398
<i>LPSIS</i>	2,904785343	0,632266829	21,41299372
<i>RASIS</i>	25,30551586	13,21642548	5,40486275
<i>LASIS</i>	22,44721594	4,29790857	8,537703484
<i>RLE</i>	15,95670573	3,751578662	8,618916874
<i>LLE</i>	24,77967834	11,29814197	9,757732404
<i>RME</i>	29,28579386	4,75022231	8,684772188
<i>LME</i>	9,364350605	16,44304609	5,996281326
<i>RHF</i>	NaN	NaN	NaN
<i>LHF</i>	24,55705282	12,37679831	13,28909056
<i>RTT</i>	7,764347894	12,35437726	30,31206468
<i>LTT</i>	12,54550353	13,08732376	18,29875598
<i>RLM</i>	33,79003697	34,41109376	54,20311104
<i>LLM</i>	18,49160191	25,41631872	28,6291968
<i>RMM</i>	14,28047505	20,89204688	22,31921791
<i>LMM</i>	6,570042871	20,03843353	15,03002348
<i>RCA</i>	8,929078085	14,0855407	10,62175186
<i>LCA</i>	4,240652895	29,19097238	26,278434
<i>RVMH</i>	18,89324921	15,51145405	18,05834092
<i>LVMH</i>	42,54758569	40,07291072	61,7867873
<i>RIMH</i>	6,387584127	25,57440752	33,10724495
<i>LIMH</i>	20,34370581	22,42765078	52,30825709

Tabella 4.2.1 - Root Mean Square Distance fra le traiettorie calcolate da video con e senza distorsione

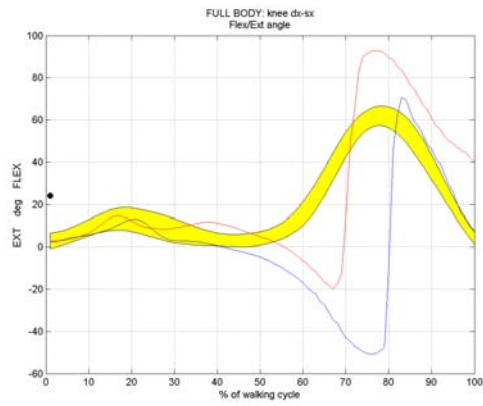
Angoli Articolari



4.2.6 - Angoli articolari dell'anca destra

<i>RMSD</i>	<i>AB/ADDUZIONE</i>	<i>FLESSO/ ESTENSIONE</i>	<i>INTRA/ EXTRAROTAZIONE</i>
<i>ANCA</i>	9,584721405	24,45540125	111,1159888

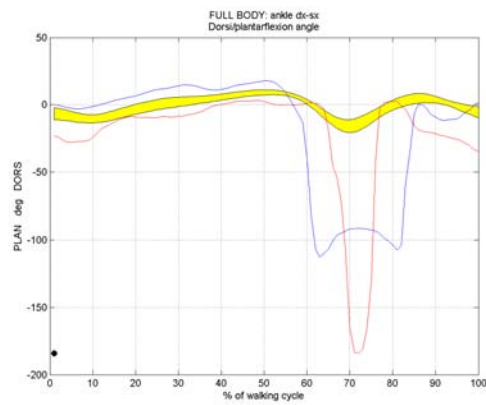
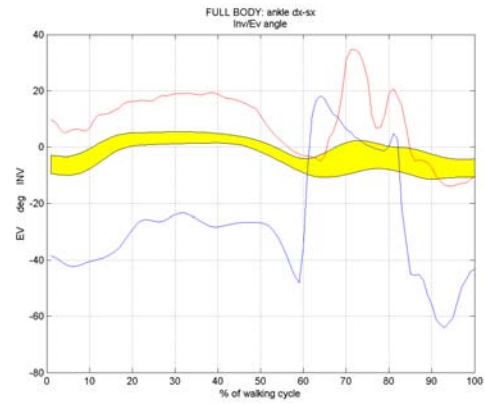
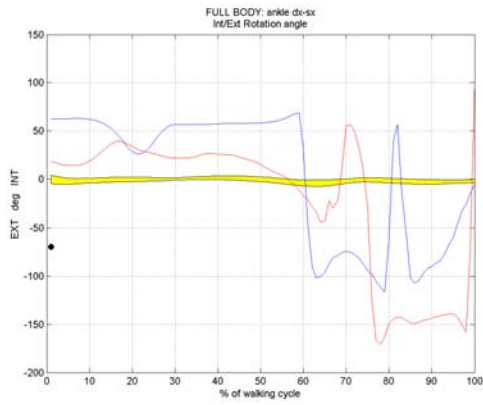
Tabella 4.2.2 - Root Mean Square Distance fra gli angoli calcolati con video distorti e non distorti



4.2.7 - Angolo di Flesso/Estensione del ginocchio destro

<i>RMSD</i>	<i>FLESSO/ ESTENSIONE</i>
<i>GINOCCHIO</i>	43,30318548

Tabella 4.2.3 - Root Mean Square Distance fra gli angoli calcolati con video distorti e non distorti



4.2.8 - Angoli articolari della caviglia destra

<i>RMSD</i>	<i>INVERSO/ EVERSIONE</i>	<i>DORSI/ PLANTARFLESSIONE</i>	<i>INTRA/ EXTRAROTAZIONE</i>
<i>CAVIGLIA</i>	39,907465	43,55680454	63,82987365

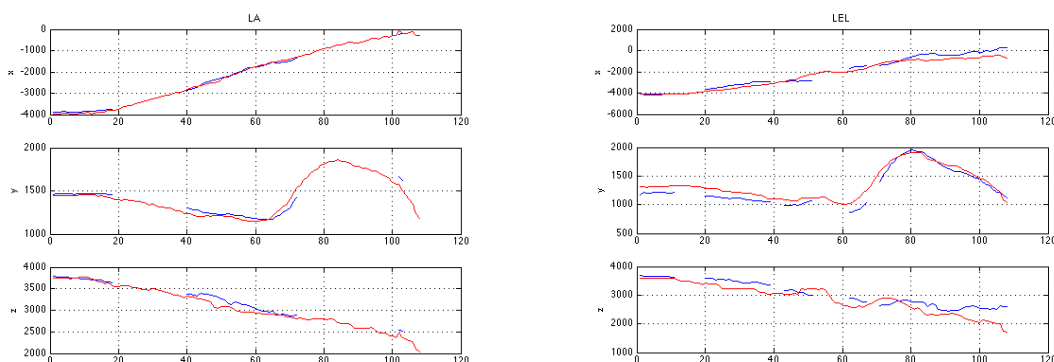
Tabella 4.2.4 - Root Mean Square Distance fra gli angoli calcolati con video distorti e non distorti

4.3 – Elaborazione dati da Video a Secco

In questa sezione sono stati riportati i confronti fra le diverse traiettorie dei *marker* che identificano due specifici punti di repere anatomico, l'acromion sinistro e l'epicondilo laterale dell'omero sinistro. Come per i video acquisiti in acqua, anche in questo caso in calce viene riportata la *Root Mean Square Distance* fra i risultati ottenuti con video distorti e con video non distorti.

La legenda è quella utilizzata per i video subacquei.

Traiettorie dei Marker



4.3.1 – Traiettorie dell'acromion sinistro e dell'epicondilo laterale sinistro

<i>RMSD Traiettorie</i>			
<i>Marker</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>LA</i>	67,537619	47,891963	102,983530
<i>LEL</i>	372,112478	97,819761	294,067232

Tabella 4.2.1 – Root MeanSquare Distance fra le traiettorie dei marker calcolate da video con e senza distorsione

4.4 – Discussione dei Risultati

Si può facilmente evincere dalle *Root Mean Square Distance* che i risultati ottenuti con la rimozione a priori della distorsione sono non solo consistenti con i risultati ottenuti rimuovendo la distorsione a posteriori, ma anche lievemente migliori, come ci si aspettava fin dall'inizio. Una prima conferma di ciò si ha con i risultati del MAL test, in cui si può notare come le RMSD siano inferiori e quindi le misure stimate siano leggermente più fedeli alle misure reali, in particolare sulle distanze più lunghe (d_{01} e d_{02}). Tale risultato è confermato anche nel calcolo degli angoli articolari e delle traiettorie dei *marker*, come si può constatare anche dai grafici (figure da 4.2.2 a 4.3.1). I valori di RMSD molto elevati trovati per i video a secco, sono una probabile conseguenza della rimozione della distorsione dal video; si può infatti anche notare visivamente in figura 3.6.1, come questa fosse accentuata.

Certamente si noterà che le curve degli angoli articolari non permangono all'interno delle fasce di normalità: questo è dovuto alla minore precisione dei sistemi di video *tracking* rispetto ai sistemi *marker - based* stereofotogrammetrici, che sono, come è noto (1.1.1), più precisi. Si può comunque notare come l'andamento generale delle curve ottenute ricalchi quantomeno l'andamento generale delle fasce di normalità, nella maggior parte degli angoli articolari.

Non deve stupire poi che alcune traiettorie, ottenute da video distorti, siano parzialmente o totalmente mancanti. Tale problema può essere imputabile alla difficoltà della procedura di *tracking* su video distorti.

4.5 – Conclusioni

Il presente lavoro di tesi ha portato allo sviluppo di un *toolbox open source* che permettesse la gestione e il processing di dati video, il *tracking* a video e la creazione di un file in formato C3D che contenesse tutte le traiettorie e i corrispettivi *label*. I test effettuati riguardo all'accuratezza della ricostruzione delle traiettorie dei *marker*, hanno confermato in buona parte le aspettative iniziali, ovvero che rimuovere la distorsione dai video prima di effettuare il *tracking* a video da risultati migliori, in controtendenza con la normale prassi seguita nei più comuni software commerciali di *video tracking*. Tale risultato è molto importante in quanto permette di avere con un unico *toolbox* la possibilità di processare da zero un video, a partire dal "deinterlacciamento", passando per la rimozione della distorsione e il *tracking*, fino ad arrivare alla creazione di un file C3D, che può essere letto in qualsiasi piattaforma software di biomeccanica e utilizzato per la stima di parametri cinematici (come gli angoli), cinetici (momenti) o in software per la simulazione muscoloscheletrica (OpenSim). Il tutto con il notevole vantaggio di conoscere quale algoritmo sta alla base del *tracker*, altra grossa limitazione dei software commerciali, e di poter intervenire direttamente sul codice per apportare eventuali modifiche, qualora ce ne fosse la necessità.

Bibliografia

- [1]. Figueroa, P. J., Leite, N. J. and Barros, R. M. L. (2003) A flexible software for tracking of markers used in human motion analysis. *Computer Methods and Programs in Biomedicine* 72(2), 155-165.
- [2]. Psycharakis, S. G. and Sanders, R. H. (2008) Shoulder and hip roll changes during 200- m front crawl swimming. *Medicine & Science in Sports & Exercise* 40(12), 2129-2136.
- [3]. Barris, S. and Button, C. (2008) A review of vision-based motion analysis in sport. *Sports Medicine* 38(12), 1025-1043.
- [4]. Shi, J. and Tomasi, C. (1994) Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*. 593-600.
- [5]. Lu, N., Wang, J., Yang, L. and Wu, Q. H. (2007) Motion Detection Based On Accumulative Optical Flow and Double Background Filtering. *World Congress on Engineering*. 602-607
- [6]. Lucas, B. D. and Kanade, T. (1981) An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*. 674 - 679.
- [7]. Tomasi, C. and Kanade, T. (1991) Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University.
- [8]. Barron, J. L., Fleet, D. J. and Beauchemin, S. S. (1994) Performance of optical flow techniques. *International Journal of Computer Vision* 12(1), 43-77.
- [9]. Liu, H., Hong, T.-H., Herman, M. and Chellappa, R. (1996) Accuracy vs. efficiency trade-offs in optical flow algorithms. In: *Computer Vision – ECCV'96*. Springer. 174-183.
- [10]. Galvin, B., McCane, B., Novins, K., Mason, D. and Mills, S. (1998) Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. *British Machine Vision Conference*. 195-204.

- [11]. Della Croce, U. and Cappozzo, A. (2000) A spot check for estimating stereophotogrammetric errors. *Medical & Biological Engineering & Computing* 2000, Vol. 38
- [12]. The C3D File Format User Guide, Motion Lab Systems, February 26, 2008
- [13]. J. Bray, Markerless Based Human Motion Capture: A Survey, Master's Thesis - Department Systems Engineering Brunel University, 2001
- [14]. L. Chiari, U. Della Croce, A. Leardini, A. Capozzo, Human Movement Analysis Using Stereophotogrammetry - Part 2: Instrumental Errors, *Gait & Posture*, 2004
- [15]. M. Donati, 3D Reconstruction of The Human Skeleton During Motion, Ph.D. Thesis - University of Bologna, 2006
- [16]. Mantoan, Underwater Gait Analysis: A Markerless Approach, Thesis - University of Padua, 2011
- [17]. L. Chiari, U. Della Croce, A. Leardini, A. Capozzo, Human Movement Analysis Using Stereophotogrammetry - Part 3: Soft Tissue Artifact Assessment and Compensation, *Gait & Posture*, 2004
- [18]. Gibson, J.J. (1950). *The Perception of the Visual World*. Houghton Mifflin.
- [19]. Kelson R. T. Aires, Andre M. Santana, Adelardo A. D. Medeiros (2008). *Optical Flow Using Color Information*. ACM New York, NY, USA. ISBN 978-1-59593-753-7.
- [20]. B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185-203, 1981
- [21]. Glyn W. Humphreys and Vicki Bruce (1989). *Visual Cognition*. Psychology Press.
- [22]. A. Cappello, A. Cappozzo, P.E. di Prampero, *Bioingegneria della Postura e del Movimento*, Pàtron Editore, 2003
- [23]. Silvio Savarese. Calibrazione in "Scansione Tridimensionale con Metodi a Luce Debolmente Strutturata". Università degli studi di Napoli Federico II and University of Michigan, 1998

- [24]. Jean-Yves Bouguet, Camera Calibration Toolbox for Matlab® (http://www.vision.caltech.edu/bouguetj/calib_doc/), 2010
- [25]. D.C. Brown, Close Range Camera Calibration, DBA Systems, 1971
- [26]. J.M. Lavest, G. Rives, J.T. Lapresté, Dry Camera Calibration for Underwater Applications, Machine Vision and Applications, 2003
- [27]. S. Bircheld. Derivation of Kanade-Lucas-Tomasi Tracking Equation. <http://www.ces.clemson.edu/~stb/klt/>, Aug 2007.
- [28]. Magalhães FA., Sawacha Z., Di Michele R., Cortesi M., Gatta G., Fantozzi S., Performance of an automatic tracking software for underwater motion analysis. *Journal of Sports Science and Medicine* (2013) 12, 660 - 667.
- [29]. S Ceccon, E Ceseracciu, Z Sawacha, G Gatta, M Cortesi, C Cobelli, S Fantozzi (2012). Application of CAST technique to 3D motion analysis of front crawl swimming. *Journal of Sports Sciences*. 2012 Oct 11)
- [30]. Pretto F, Sawacha Z, Gambaretto E, Corazza S, Menegatti M, Cobelli C. A platform for multipurpose markerless motion capture. Proceedings Siamoc, 26th-28th September 2013, Pisa, Italy
- [31]. A Leardini, Z Sawacha, G Paolini, R Nativo, S Ingrassio, MG Benedetti. (2007). A New Anatomical Based Protocol for Gait Analysis in Children. *Gait & Posture*. Vol. 26, Pp.560-571
- [32]. Sviluppo di un metodo per confrontare l'analisi del cammino markerless e l'analisi del cammino marker-based in ambiente acquatico, Minelle F., Università degli Studi di Padova
- [33]. Copyright eMotion S.r.l. SMART Motion Capture System - Getting started, 2000-2004 edition. Manuale di utilizzo SMART eMotion
- [34]. www.bts.it

Ringraziamenti

In primis concedetemi di ringraziare la mia relattrice, professoressa Chiara Dalla Man, per la gentilezza e disponibilità che non sono mai mancate. Ringrazio di cuore anche la mia correlatrice, ing. Zimi Sawacha Ph.D., per essermi stata sempre accanto in questi mesi di lavoro e avermi aiutato ogni qualvolta ne avessi bisogno. Ringrazio tutte le ragazze del Laboratorio di Bioingegneria del Movimento, Alessandra, Alice, Annamaria, Fabiola e Martina, per tutti i consigli e gli aiuti dispensatimi in questi mesi passati assieme. Ricorderò sempre con gioia i pranzi in compagnia e gli immancabili caffè al bar.

Ringrazio i ragazzi del laboratorio di robotica per la compagnia e il sostegno morale che mi hanno dato, oltre che per la rete Wi - Fi ovviamente!

Una menzione speciale la merita Filippo, per tutto l'aiuto, i consigli e l'amicizia dimostrati in questo tempo passato fianco a fianco, davanti ai nostri immancabili Mac.

Voglio ringraziare tutti gli amici della compagnia, non vi nomino uno ad uno sennò non finisco più! Grazie per essermi stati accanto, per ogni battuta, scherzo, risata fatta assieme, per avermi sopportato e aiutato anche nei momenti più complicati.

Un grosso grazie ai miei compagni di corso, sia della triennale che della magistrale, per essere stati ottimi colleghi, compagni di gioco, collaboratori, per avermi spronato a dare sempre di più, e più di tutto per essere stati dei grandi amici.

Grazie ai miei condiscipoli della scuola Long Son di Viet Vo Dao di Campodarsego, per tutte le botte prese, per le proiezioni (riuscite e mancate), per le tavolette rotte etc...: per ogni singolo momento di ogni allenamento passato assieme. In particolare grazie al mio maestro Marco Bao Dahn Salmaso, per essere una guida e un punto di riferimento.

Grazie ai coristi, in particolare al mio direttore Alessandro e al pianista Paolino, per riuscire ogni martedì a sopportare la mia presenza ingombrante.

Grazie a tutti gli scout che conosco, in particolare alla Comunità Capi di Campodarsego per la strada trascorsa assieme.

Un ringraziamento particolare lo dedico a Marsida, per essermi stata accanto nei due anni di magistrale e per avermi dimostrato affetto, nonostante tutto, anche in quest'ultimo periodo.

Un grazie speciale alla funzione $-e^x$ semplicemente per la sua monotonia decrescente, e grazie a chi me l'ha fatto ricordare un sabato sera. Anche la sola presenza è un buon motivo per ringraziare. Grazie, dal latino gratia, benevolenza, concordia, fascino.

Infine ringrazio la mia famiglia, in particolare i miei genitori, senza i quali sicuramente non sarei qui. Grazie a loro per il sostegno morale, che non è mai mancato, ma anche per il sostegno economico, decisamente non indifferente. Grazie per tutto quello che mi avete concesso e che avete sopportato di me. Grazie degli insegnamenti ricevuti, grazie ai quali ora mi sento un ragazzo migliore. Crescendomi con il vostro amore, mi avete reso un ragazzo felice, spero di poter ricambiare al meglio l'affetto che mi avete donato in questi anni e in un qualche modo spero possiate esser fieri di me. Vi voglio bene.

Alberto