

UNIVERSITÀ DEGLI STUDI DI ROMA
TOR VERGATA



FACOLTÀ DI SCIENZE MATEMATICHE,
FISICHE E NATURALI

**Corso di Laurea Triennale in
Scienze e Tecnologie per i Media**

**Integrazione all'Ambiente di Google Earth Studio
per Matchmoving non basato su Tracking**

Relatore:
Prof.ssa
Francesca Pelosi

Laureando:
Federico March
Matricola **0213327**

Correlatore:
Prof.
Fernando Tornisiello

Anno Accademico 2018/2019

*Dedicato alla mia famiglia,
a Marco Cascio, Tate ed Ariel*

Ringraziamenti

Vorrei porgere un ringraziamento speciale al prof. Fernando Tornisiello ed alla prof.ssa Francesca Pelosi per avermi concesso l'opportunità di poter lavorare ad un progetto molto interessante come quello proposto in questa tesi, per la cordialità e la disponibilità mostrata.

Vorrei ringraziare la mia famiglia per aver appoggiato sempre le mie scelte, per aver condiviso con me ogni momento che ha caratterizzato questi anni, per avermi potuto permettere di poter raggiungere questo traguardo, per avermi sostenuto nel momento del bisogno, per aver continuato a credere in me e nelle mie capacità anche nei momenti di difficoltà. Vorrei ringraziare nonna Giovanna, zia Mariolina, Oxana e Antoneta, per essersi sempre interessate alla mia carriera universitaria.

Vorrei ringraziare Marco Cascio per essermi sempre stato vicino, per essere stato sempre pronto ad aiutarmi nel momento del bisogno, per i consigli, per la pazienza, per avermi permesso di realizzare tanti desideri che avevo nel cassetto sin da piccolo.

Vorrei ringraziare Paola Lazzarini e Raffaella di Gregorio, con cui ho condiviso i traguardi raggiunti in questi anni grazie ai loro insegnamenti impartiti.

Vorrei ringraziare i miei colleghi Ilaria Teodori detta “Annie”, Benedetta Bugianesi, Elena Bearzotti, Isabel Toro Sima, Simone Fiorellino, Eleonora Macchi, Emanuele Rosmarino per i bei momenti passati insieme, per le risate, gli scherzi, i consigli che hanno caratterizzato questo percorso universitario che insieme abbiamo condiviso.

Vorrei ringraziare soprattutto la mia amica Noemi Gravante, per essermi stata accanto in ogni momento, per essere sempre in grado di capirmi ed ascoltarmi, per non avermi mai lasciato solo nei momenti di difficoltà. Vorrei ringraziare tutti i miei amici, in particolare Giordano Rodilossi, Beatrice di Blasi, Claudia Calò, Flavia Palmeri, Giulia Cinotti, Nisha Tak, Sabrina Romanò, Christian Camuto, Alessandro Garzia, Federica Solimena, Erica Vannucci, Serena Tedeschi, Gemma Sgrò, Alessandro Sgrò, Elena Khoury, Gina Koutsandria, Valerio Di Valerio, Alice Musarra, Paola Abrescia, Nicolino D'Alessandro, Ilaria Scalas, Ilaria Banchetti ed Ilaria Chiaroni, per avermi dato il loro supporto in questi anni, per aver condiviso tanti bei momenti insieme.

Indice

1. Introduzione	1
1.1 Matchmoving negli Effetti Visivi e altre applicazioni	2
1.2 Structure from Motion per Matchmoving	3
1.2.1 Tracking 2D	4
1.2.2 Calibrazione della camera	6
1.3 Tecniche di raffinamento per il matchmoving automatico	8
1.4 Contributi ed Outline	9
2. Sistema di Coordinate Geografiche	10
2.1 Latitudine, longitudine e altitudine	11
2.1.1 Latitudine	11
2.1.2 Longitudine	14
2.1.3 Altitudine	16
2.2 Coordinate Cartesiane	18
2.2.1 Coordinate ECEF	19
2.2.2 Piano Tangente Locale	22
2.2.2.1 Coordinate NED	23
2.2.2.2 Coordinate ENU	24
2.3 Modelli di rappresentazione per la forma della Terra	25
2.3.1 Modello Sferico	26
2.3.1.1 Conversione delle coordinate da LLH a ECEF	28
2.3.2 Modello Ellittico	30
2.3.2.1 Modello WGS84	32
2.3.2.2 Conversione delle coordinate da LLH a ECEF	34
3. Geometria delle Rotazioni	36
3.1 Rotazione intorno agli assi Cartesiani	39
3.2 Rotazioni intorno ad un asse arbitrario	42
3.2.1 Matrici Ortogonali	42
3.2.2 Costruzione di una base ortonormale associata all' asse di rotazione	43
3.2.3 Cambiamento di base	45
3.3 Angoli di Eulero	48
3.3.1 Angoli di Roll, Pitch e Yaw	51

4. Google Earth Studio	53
4.1 Creare un'animazione con Google Earth Studio	53
4.2 Analisi dei dati	57
4.2.1 Camera Position	57
4.2.2 Camera Rotation	59
5. GES per matchmoving non basato su tracking	63
5.1 Esportare i dati da Google Earth Studio	65
5.2 Calibrare la camera	67
5.2.1 Gestione dei dati in input	67
5.2.2 Calcolo delle coordinate ENU per la posizione della camera	69
5.2.3 Orientare la camera nel sistema ENU	72
5.2.4 Stima della lunghezza focale	83
5.3 Animare la camera virtuale	87
5.3.1 Gestione dei dati in output	87
5.3.2 Realizzare un'animazione in Maya	88
5.3.3 Python per l'automatizzazione del processo di animazione della camera..	89
5.4 Ricostruire la scena di GES in Maya	92
5.5 Matchamation	94
6. Test e Valutazioni	97
6.1 Via dei Fori Imperiali	98
6.2 Piazza Venezia	101
6.3 Molgolfiere allo Stadio Olimpico	104
6.3.1 Clip Test 1	104
6.3.2 Clip Test 2	107
6.4 Astronavi a Fiumicino	109
6.4.1 Clip Test 1	109
6.4.2 Clip Test 2	111
7. Conclusioni	114
7.1 Sviluppi futuri	114
Bibliografia	116
Software usati:.....	118

Capitolo 1

Introduzione

Da sempre, il cinema ha cercato di stupire i suoi spettatori con effetti speciali: si pensi alle opere di George Méliès, in cui i rudimentali effetti visivi permettevano di realizzare apparizioni, sparizioni, trasformazioni, spostamento di oggetti inanimati; dunque, tutti effetti che miravano a meravigliare il pubblico, ottenuti attraverso semplici tecniche di montaggio o la sovrapposizione di due pellicole.

Col passare degli anni, le tecniche per realizzare effetti speciali vennero raffinate e perfezionate, tanto da poter introdurre l'uso di retroproiezioni, animazioni, miniature, animatronics, stampanti ottiche, fino ad arrivare a oggi, dove queste tecniche sono state digitalizzate, grazie al progresso della computer grafica.

Oggigiorno, dunque, attraverso tecniche di CGI (Computer-Generated Imagery) è possibile realizzare effetti computerizzati elaborati e molto realistici, tanto da determinare una distinzione tra effetti speciali ed effetti visivi.

Gli effetti visivi permettono di creare creature fantastiche, reali o estinte, come, ad esempio, draghi, leoni o dinosauri, estendere ambienti esistenti (Fig.1.1), costruire ambienti irreali o che non possono essere ripresi dal vivo, simulare esplosioni, inondazioni e altro, garantendo un risultato sufficientemente realistico [17].

Una tecnica a cui si ricorre per generare questo tipo di effetti in presenza di movimento di macchina è quella del matchmoving.

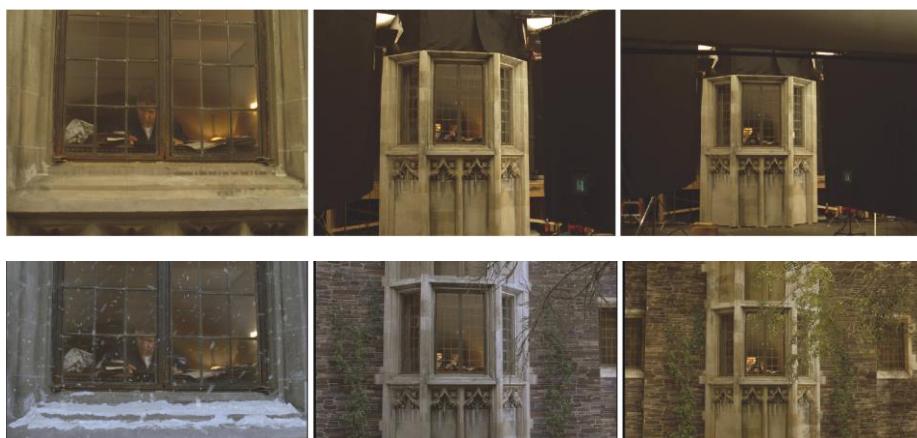


Figura 1.1. Un esempio di applicazione del matchmoving per estendere ambienti esistenti, in [17]. La prima riga mostra alcuni scatti del set originale. La seconda riga mostra il risultato finale ottenuto con la tecnica del matchmoving.

1.1 Matchmoving negli Effetti Visivi e altre applicazioni

Nell'ambito degli effetti visivi, il matchmoving è una tecnica che permette di inserire elementi in CGI in un girato live-footage.

Supponiamo di voler realizzare, ad esempio, una clip video in cui un'astronave plana su una strada di una città, magari distruggendo qualche oggetto e scheggiando qualche edificio. Per far questo, abbiamo bisogno di due elementi: una clip video che riprenda la città su cui si vuol far volare l'astronave e il modello 3D dell'astronave animato. Quando si gira il video della città, le riprese devono essere effettuate immaginando di inquadrare e seguire l'astronave, anche se l'astronave non esiste. Spetterà, poi, al reparto effetti visivi realizzare ed animare il modello 3D dell'astronave nel modo più realistico possibile; prima di inserire questo modello nel girato, però, è necessario ricostruire il movimento della camera reale nella clip, ovvero la sua posizione, la sua orientazione, lunghezza focale, istante per istante, al fine di conferire alla scena finale un aspetto che sia il più corretto possibile. Questo è il compito che spetta a chi si occupa di matchmoving. Dunque, affinchè l'astronave appaia ben inserita nella scena, è necessario che la camera virtuale con cui si realizza l'animazione esegua lo stesso movimento della camera che è stata usata per filmare la clip.

Il matchmoving è una tecnica disponibile per ricostruire le informazioni relative alla camera di una ripresa dal vivo e quindi deve garantire che l'oggetto realizzato in CGI abbia posizione, orientazione, proporzioni, movimento corretti e coerenti agli oggetti che appaiono nella scena in cui questo va inserito (Fig.1.2)

Oltre agli effetti visivi, la tecnica del matchmoving trova il suo impiego anche in altri ambiti, quali:

- **Archeologia:** in questo ambito, il matchmoving consente di poter studiare la storia di un sito archeologico, inserendo in CGI quelli elementi che permettono di ricostruire l'aspetto originale degli scavi [3].
- **Videogame e applicazioni:** in questo ambito, è possibile sfruttare il matchmoving per inserire gli asset 3D destinati ai videogame e/o applicazioni. Si pensi ad esempio, alle applicazioni che fanno uso di realtà aumentata.



Figura 1.2. Un esempio di matchmoving per effetti visivi, in [17]. La prima riga mostra dei frame della clip originale. Nella seconda riga mostra il risultato finale, inserendo degli oggetti in CGI ricorrendo alla tecnica di matchmoving.

- **Ingegneria e architettura:** in questo ambito, il matchmoving viene utilizzato per realizzare visualizzazioni architettoniche molto dettagliate in contesti reali; ad esempio, è possibile inserire il modello 3D di un edificio in una ripresa video del luogo in cui questo dovrà essere costruito, al fine di avere un'idea più accurata di quello che sarà il risultato finale e di analizzare come questo edificio si integri con l'ambiente circostante [6].

In questo paragrafo, è stato descritto quale sia il fine del matchmoving e alcuni ambiti in cui questo viene applicato; delineiamo, ora, come funziona questa tecnica.

1.2 Structure from Motion per Matchmoving

Come detto nel paragrafo precedente, per inserire in modo corretto un elemento in CGI all'interno di una sequenza video, si ricorre alla tecnica del matchmoving. L'algoritmo su cui si basa il matchmoving è lo Structure for Motion (SfM). In computer vision, lo Structure for Motion è una tecnica che permette di stimare la geometria 3D di una scena, da cui deriva il termine “Structure”, e che, allo stesso tempo, permette di stimare le posizioni della camera che la riprende, da cui deriva il termine “Motion” [20]. Questa tecnica, dunque, consente a chi si occupa di matchmoving di ricavare le informazioni 3D della scena che viene analizzata. Le informazioni spaziali della camera e degli oggetti di scena vengono ricavate attraverso la selezione di feature in ogni frame che compone una sequenza video.

Le feature sono dei punti caratteristici dell'immagine che possono essere presi come punti di riferimento per analizzare il modo in cui si muove la camera frame per frame.

In base al modo in cui vengono individuate queste feature, è possibile distinguere il matchmoving manuale dal matchmoving automatico.

Nel matchmoving manuale, è l'operatore stesso, detto matchmover, che si occupa di individuare, all'interno di un frame, quelle feature che possono essere importanti al fine di ricavare le informazioni spaziali della camera. Nel caso di matchmoving automatico, invece, la scelta delle feature viene effettuata attraverso software, secondo particolari criteri. A prescindere dal fatto che sia svolta in modalità automatica o manuale, la tecnica di matchmoving si suddivide in due parti [5]:

- Tracking 2D:** il movimento delle feature viene seguito e tracciato in ogni frame che costituisce la sequenza video.
- Calibrazione della camera:** analizzando le tracce ottenute nella fase precedente, è possibile ricostruire il movimento della camera, la sua posizione, la sua orientazione e altre caratteristiche intrinseche.

La Figura 1.3 riassume le fasi di cui è composta la tecnica di matchmoving.

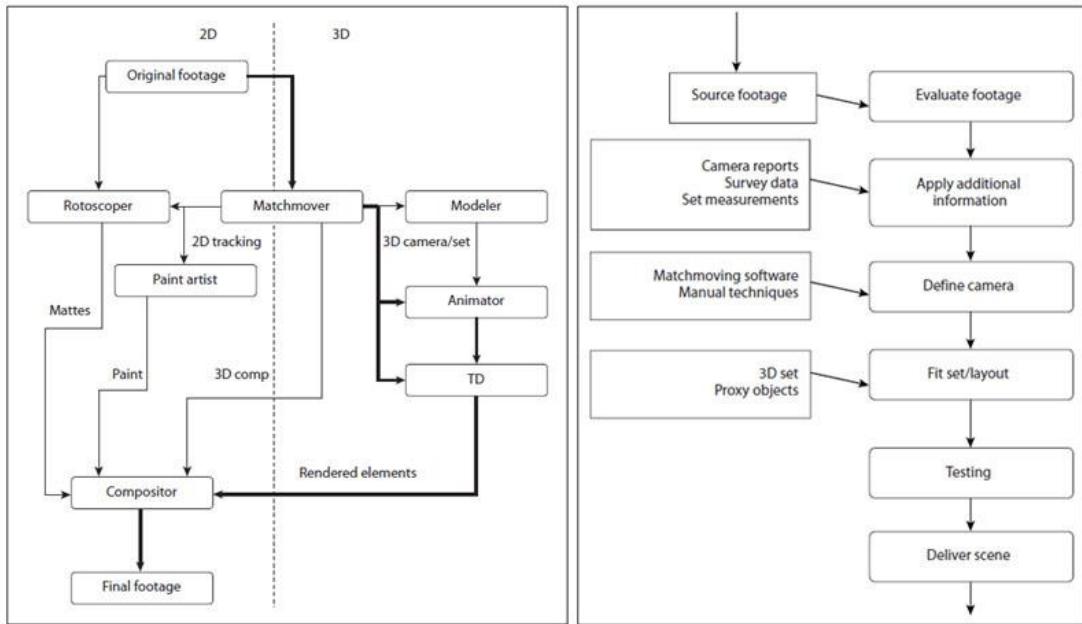


Figura 1.3. Il workflow del matchmoving [5].

1.2.1 Tracking 2D

La prima fase del matchmoving consiste nel tracking 2D, ovvero nel determinare la posizione di ogni feature in ogni frame della sequenza video.

Di solito, nel matchmoving manuale, vengono selezionati come features punti abbastanza significativi, come, ad esempio, quei punti che si sanno essere fissi nella realtà, come gli spigoli degli edifici o dei cartelli stradali (Fig.1.4a). Se si prendono in considerazione solo punti che si trovano in corrispondenza del centro dell'immagine, in fase di calibrazione, sarà possibile ricostruire la scena solo parzialmente, in quanto non si hanno informazioni spaziali a sufficienza. Una buona politica, dunque, sta nello scegliere feature che possano dare, il più possibile, informazioni spaziali della scena da analizzare (Fig.1.4b). Ognuna di queste feature viene individuata attraverso dei marcatori. Prima di scegliere le feature nei frame, può essere utile sapere in quali parti dell'immagine l'oggetto 3D verrà posizionato e quali sono le sue caratteristiche oppure effettuare un ritocco dell'immagine, per poter individuare meglio quei punti che si potrebbero rivelare importanti ai fini del tracking. Per ottenere una buona calibrazione della camera, di solito, il numero di feature minimo richiesto per ogni frame dai software di tracking va da 7 a 12 [5]. Ovviamente, il numero effettivo di feature selezionate dipende dalla complessità della scena che si deve analizzare.

Una volta scelte le feature in un frame, il software di tracking effettua delle scansioni nel frame successivo per determinare il punto in cui ognuna di esse si è spostata.



Figura 1.4. (a) Una feature selezionata su uno spigolo di un edificio, in [5]. Come si può vedere dall’immagine, ogni feature è composta da tre parti: un punto centrale, una pattern area (il quadrato più interno) ed una search area (il quadrato più esterno). (b) Un esempio di selezione di feature proposto in [5], in cui le feature scelte cercano di fornire informazioni spaziali della scena.

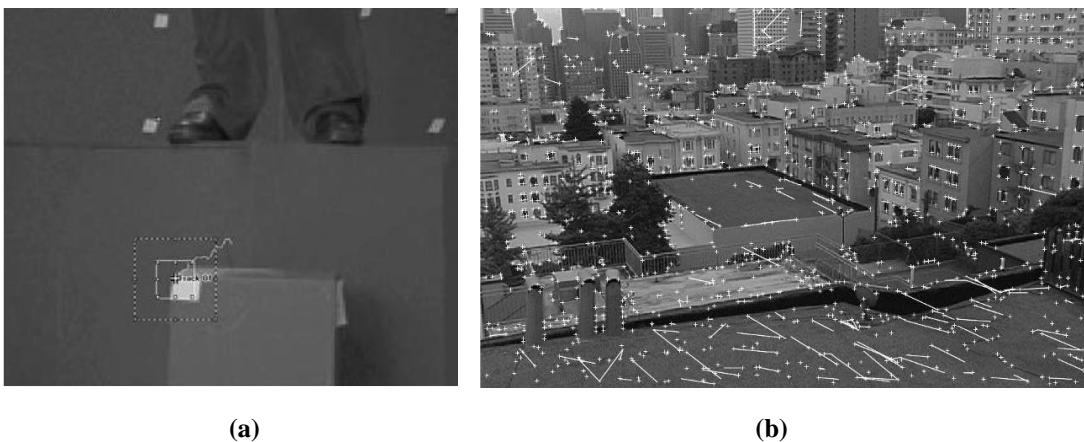


Figura 1.5. (a) Lo spostamento di una feature tracciato dopo la fase di tracking 2D [5]. (b) Un esempio di matchmoving automatico [5].

Una volta identificata la feature, il programma prosegue analizzando il frame seguente, ripetendo così il procedimento, fin quando tutte le feature non sono state analizzate. Per ogni feature analizzata, dunque, viene tracciato il suo movimento in ogni frame.

Se la feature cambia improvvisamente la sua luminosità, viene occlusa da un altro oggetto della scena, o l’oggetto corrispondente alla feature esce dal frame, il software di tracking smetterà di tracciare la feature corrente e passerà ad analizzarne un’altra.

Nel matchmoving automatico, invece, è un apposito software ad effettuare la scelta delle feature. Di solito, il programma prende in considerazione quei punti dell’immagine che appaiono più chiari, ben contrastati e che abbiano un’elevata diversità cromatica, selezionando quelle feature che potrebbero essere usate in più di un frame. Una volta determinate, il software esamina il modo in cui queste si allineano nei frame successivi, tentando di creare, per ogni feature, un’unica traccia che si possa

estendere per tutti i frame che compongono la sequenza video (Fig.1.5a). Durante il tracking, se individuate, il programma provvede ad eliminare quelle tracce che si muovono in modo particolarmente diverso dalle altre; in questo modo possono essere scartate quelle feature che sono state selezionate in fase iniziale prendendo in considerazione oggetti in movimento, riflessioni, bagliori di luce.

Grazie al matchmoving automatico è possibile definire un elevato numero di tracce (Fig.1.5b). Allo stesso tempo, però, è preferibile ricorrere al matchmoving manuale, in quanto in quello automatico, proprio a causa dell'alto numero di feature considerate, viene prodotta un'elevata quantità di tracce errate; questo comporta soluzioni non propriamente accurate per la calibrazione della camera [1].

1.2.2 Calibrazione della camera

La seconda fase in cui si articola la tecnica del matchmoving è quella relativa alla calibrazione della camera; calibrare la camera significa determinare la sua posizione nello spazio, la sua orientazione, oltre ad alcune proprietà intrinseche, quali,

- Lunghezza focale,
- Dimensioni del sensore,
- Eventuale distorsione dell'immagine dovuta al tipo di lenti usate,

al fine di poter riprodurre in maniera esatta il movimento della camera che è stata usata per girare la clip.

Per la calibrazione della camera, lo Structure from Motion, affiancato da alcune tecniche di fotogrammetria, gioca un ruolo importante. Infatti, l'algoritmo di SfM consente di ricavare le informazioni sulla posizione della camera a partire dai frame, calcolando la corrispondenza tra i punti delle immagini, individuati nella fase di tracking.

Prendiamo in considerazione due frame consecutivi della stessa clip video.

Supponiamo che in fase di tracking siano stati individuati, nel primo frame, tre punti che possiamo considerare come rilevanti ai fini della calibrazione della camera.

Nel secondo frame, si ha che la stessa scena viene ripresa da un punto di vista differente, in quanto si suppone che la camera si sia spostata oppure abbia cambiato il suo orientamento. Anche nel secondo frame vengono marcati gli stessi tre punti che sono stati considerati nel primo. Abbiamo supposto che il secondo frame è stato ottenuto da un movimento della camera nello spazio. Dunque, possiamo definire, in modo equivalente, due camere che riprendono la stessa scena da due punti di vista diversi; ad ognuna di queste camere associamo uno dei due frame considerati. Per trovare le corrispondenze tra i frame, da ognuna di queste camere tracciamo dei raggi di visuale, dando importanza soltanto ai quei tre raggi che passano per i tre punti considerati in ognuno dei due frame; indichiamo con a' , b' , c' i raggi uscenti dalla prima camera, mentre con a'' , b'' , c'' quelli uscenti dalla seconda. Calcolando i punti di intersezione tra i raggi a' e a'' , b' e b'' , c' e c'' , è possibile ricavare la posizione dei punti considerati nei frame e le posizioni delle due camere (Fig.1.6a).

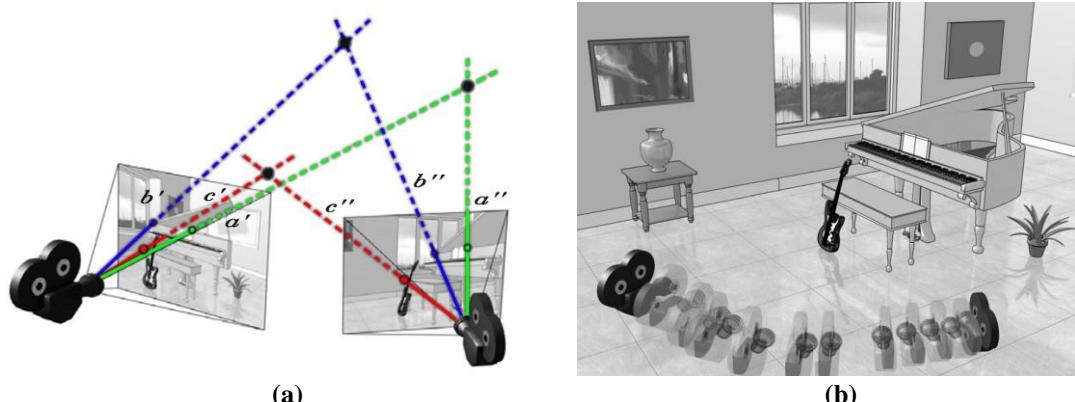


Figura 1.6. (a) Stima della posizione 3D della camera determinando i punti di corrispondenza tra i frame [5]. (b) Ricavate le posizioni consecutive della camera è possibile delinearne il movimento [5].

In questo modo, dunque, abbiamo ottenuto le due posizioni consecutive della camera, quindi, possiamo cominciare a delinearne il movimento. Prendendo in considerazione sempre più frame, e aggiungendo sempre più punti su ognuno di essi, si aggiungono sempre più vincoli che permettono di ottenere una calibrazione sempre più precisa della camera (Fig.1.6b).

Talvolta, poter attingere ad informazioni aggiuntive, permette di velocizzare questi calcoli, ottenendo una calibrazione della camera migliore. Tali informazioni riguardano:

- dati relativi alle impostazioni di alcuni parametri della camera usata per realizzare la ripresa, come, ad esempio, lunghezza focale, apertura diaframma, dimensioni sensore, eventuale distorsione dovuta alle lenti usate;
- dati provenienti e raccolti sul set, come, ad esempio, l'altezza a cui si trovava la camera, misure dei vari oggetti che compaiono nella scena, la loro distanza dalla camera, sapere le dimensioni e le proporzioni del set;
- presenza di parallasse nei frame, dove con il termine parallasse si indica quel fenomeno per cui un oggetto sembra spostarsi rispetto allo sfondo quando si cambia il punto da cui lo si osserva.

Una volta calibrata la camera, occorre effettuare un test che verifichi la correttezza e l'accuratezza del matchmoving ottenuto. Per testare un matchmoving, si inseriscono degli oggetti 3D nella scena, per vedere se la loro prospettiva coincide con quella della clip (Fig.1.7a). Un matchmoving scorretto mostrerà una clip video in cui gli elementi in CGI sembrano slittare. In un matchmoving corretto, invece, la scena e gli oggetti 3D appaiono perfettamente allineati nelle posizioni, orientamento e proporzioni.

Se il risultato finale che si ottiene è corretto, allora si può passare ad inserire la nostra astronave animata nella nostra clip.

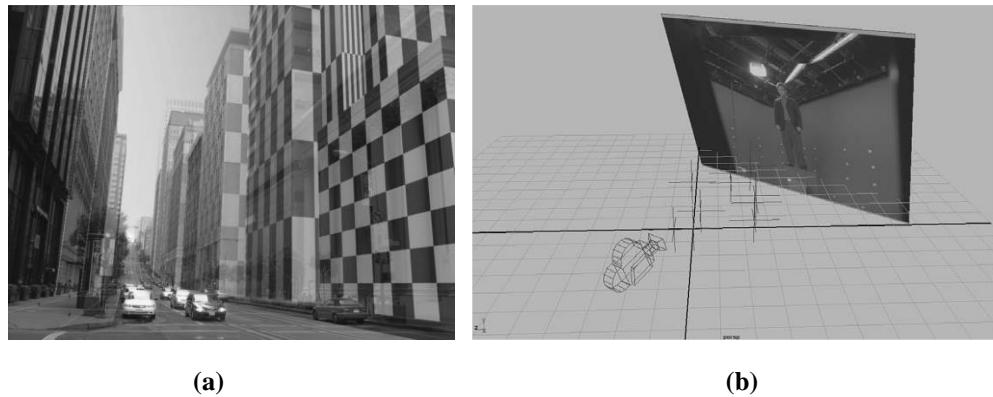


Figura 1.7. (a) Test per verificare l'accuratezza e la correttezza del matchmoving [5]. (b) La camera virtuale riproduce lo stesso movimento della camera che è stata usata per girare la clip video [5].

1.3 Tecniche di raffinamento per il matchmoving automatico

Come specificato nel paragrafo precedente, esistono due tipologie di matchmoving: manuale e automatico. Nello stato dell'arte sono state proposte alcune soluzioni finalizzate al raffinamento del matchmoving automatico. Di seguito, verranno menzionate alcune tra queste tecniche.

Determinare delle feature è una procedura molto importante e difficile allo stesso tempo, in quanto le feature scelte devono risultare adeguate ai fini della calibrazione. Spesso, occlusioni, presenza di motion blur, scarsa illuminazione possono complicare questo procedimento. E' sorta, dunque, la necessità di sviluppare nuovi metodi che tentano di risolvere i problemi sopra elencati.

Una tra le tecniche più usate nel tracking è SIFT (Scale Invariant Feature Transform), che permette di stimare e rilevare automaticamente feature abbastanza adeguate e stabili da usare come input nello SfM. In [19], gli autori propongono un metodo che permette di rilevare feature sia negli oggetti occlusi, sia in quegli oggetti che prima erano occlusi e che poi sono tornati ad essere visibili. Rimanendo ancora in ambito di occlusioni, gli autori in [9] propongono una tecnica che, dividendo l'immagine in regioni, consente di tracciare feature a livello di regioni e feature a livello di oggetti che compongono la scena, considerando il modo in cui questi appaiono, la loro distribuzione spaziale ed il modo in cui vengono occlusi. In [21], viene presentato un approccio che consente di rilevare ed eliminare occlusioni e riflessioni, separandoli dal resto dell'immagine.

Gli autori in [10] propongono un metodo che permette di arginare il problema della presenza di motion blur nella clip video originale; la presenza di motion blur, solitamente, complica il processo di calibrazione, in quanto gli algoritmi standard di tracking fallirebbero nel loro intento sotto queste condizioni. Invece di provare a rendere nitide quelle parti della clip che appaiono sfocate, gli autori del metodo propongono di stimare la sfocatura e la distorsione dell'immagine, sfruttando tali stime per il tracking di feature nelle regioni sfocate. In [12], invece, gli autori tentano di risolvere il problema legato al motion blur proponendo come soluzione quella di rilevare le feature attraverso una stima del movimento della camera, usando sensori inerziali montati su di essa.

In [22], viene presentata una tecnica per la calibrazione della camera, riprendendo una superficie piana da almeno due angolazioni diverse. Infine, in [1] gli autori suggeriscono di sfruttare scanner laser con tecnologia LiDAR per ottenere informazioni aggiuntive finalizzate a velocizzare il processo di calibrazione. Sfruttando questa tecnologia, infatti, è possibile generare una nuvola di punti 3D molto accurata della scena da poter allineare con il tracking delle feature, permettendo di misurare e riprodurre le proporzioni del set.

1.4 Contributi ed Outline

Nei paragrafi precedenti si è descritto in cosa consiste la tecnica del matchmoving per creare effetti visivi e di come questa procedura ricorra a tecniche di tracking per calcolare il movimento della camera virtuale.

In questa tesi viene proposto un metodo per effettuare matchmoving in cui dei modelli creati in CGI vengono inseriti nel girato senza l'utilizzo di tecniche di tracking. In questo caso, la calibrazione della camera avviene attraverso coordinate geografiche, fornite dal software Google Earth Studio, evitando, così, di ricorrere al tracking. Attraverso una conversione di coordinate, è possibile ricostruire ed importare posizioni, orientamento e movimento della camera in Maya, software di modellazione 3D con cui verranno realizzati i modelli in CGI da inserire nella clip originale. Le caratteristiche intrinseche della camera sono state ricavate in maniera empirica. In questa tesi, non verranno prese in considerazione riprese dal vivo ma clip video realizzate con Google Earth Studio.

Il lavoro proposto si articolerà nei seguenti capitoli:

- **nel capitolo 2** verrà introdotto e descritto il Sistema delle Coordinate Geografiche, le varie tipologie di coordinate attraverso cui è possibile esprimere posizioni sulla Terra ed alcuni modelli di rappresentazione della superficie terrestre, utili per il calcolo delle coordinate;
- **nel capitolo 3** verrà presentata la geometria delle rotazioni, che permette di descrivere il passaggio da coordinate ECEF a ENU, oltre ad orientare la camera attraverso gli angoli di Eulero;
- **nel capitolo 4** verrà presentato il software Google Earth Studio e sarà redatta un'analisi del file JSON contenente i dati della camera;
- **nel capitolo 5** verranno descritte le procedure che permettono di calibrare la camera senza ricorrere a tecniche di tracking, prendendo come sistema di riferimento il sistema delle coordinate ENU;
- **nel capitolo 6** verranno presentati alcuni risultati di matchmoving applicando la tecnica descritta in questo lavoro;
- **nel capitolo 7** saranno tratte le conclusioni ed eventuali sviluppi per il futuro.

Capitolo 2

Sistema di Coordinate Geografiche

In questo capitolo verrà introdotto il Sistema di Coordinate Geografiche, indicando gli aspetti che lo caratterizzano, descrivendo le varie tipologie di coordinate in uso e delineando due modelli di rappresentazione matematica della Terra.

Il Sistema di Coordinate Geografiche è un sistema di coordinate che permette di individuare univocamente la posizione di un oggetto o altre entità, come, ad esempio, luoghi, città, sulla superficie terrestre.

Al giorno d'oggi, alle entità sopra elencate, possono essere aggiunte anche le persone, grazie alla sempre più vasta diffusione di smartphone. Su tali dispositivi, infatti, è montata un'antenna di ricezione GPS che, con l'ausilio di applicazioni mobile, (Google Maps Mobile, WhatsApp) è possibile identificare il luogo geografico in cui si trova la persona che ne fa uso.

Ad esempio, Google Maps Mobile (Fig.2.1) e WhatsApp permettono al cliente di poter, rispettivamente, sfruttare il GPS integrato nel dispositivo per la navigazione satellitare e di condividere la propria posizione con altri utenti.

Di solito, nel Sistema di Coordinate Geografiche, ogni posizione viene specificata con numeri, simboli e/o lettere. Ogni posizione, dunque, è definita da delle coordinate, di cui due o tre si riferiscono ad una posizione orizzontale, mentre una si riferisce ad una posizione verticale.

Al fine di evitare eventuali ambiguità su quella che possa essere considerata la “direzione orizzontale” o “verticale” della superficie che si vuole misurare, si è scelto di usare un sistema di riferimento sulla Terra, con origine e direzione fissata, detto datum. Questi datum possono essere sia di tipo globale, in quanto essi rappresentano l'intero pianeta Terra, sia di tipo locale, in quanto questi rappresenterebbero soltanto una porzione della superficie terrestre; ad ogni modo, qualunque sia il datum che si sceglie di adottare, il suo fine è sempre quello di georeferenziare un luogo sulla Terra.

Pertanto, una volta scelto un datum opportuno, attraverso un sistema di coordinate ad esso riferito, è possibile esprimere la posizione dell'entità di interesse.



Figura 2.1. Esempio di applicazione mobile per la geolocalizzazione.

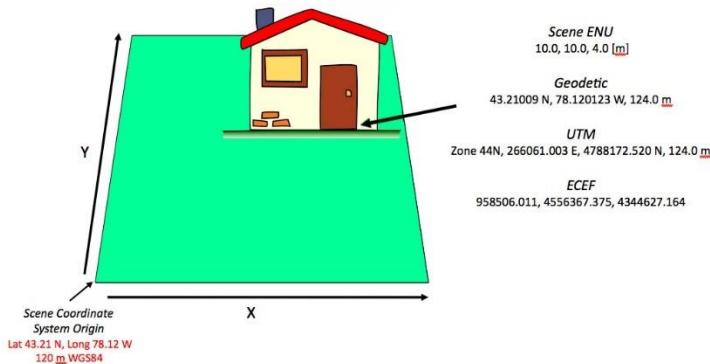


Figura 2.2. Una stessa posizione geografica può essere espressa mediante diversi sistemi di coordinate.

Esistono diverse tipologie di coordinate geografiche, tra loro tutte equivalenti, ed è possibile passare da un sistema di coordinate all'altro grazie ad opportune equazioni matematiche. La Figura 2.2 riassume come una stessa posizione geografica può essere espressa attraverso coordinate diverse.

In questa parte del capitolo verranno descritte alcune tra le tipologie di coordinate geografiche più usate.

2.1 Latitudine, longitudine e altitudine

Le coordinate geografiche più comuni e conosciute sono latitudine, longitudine ed altitudine.

Quando si vuole georeferenziare un'entità sulla superficie terrestre, spesso, una soluzione molto conveniente è quella di esprimere la sua posizione mediante una posizione verticale (altezza o profondità) ed una posizione orizzontale.

La posizione orizzontale è caratterizzata da due gradi di libertà; dunque, sono necessari almeno due parametri per poter esprimere univocamente una posizione.

La posizione verticale, invece, descrive una posizione lungo una direzione verticale, che sia al di sopra o al di sotto di uno specifico datum, ovvero una superficie di riferimento.

Esistono diverse opzioni per la rappresentazione di posizioni orizzontali, tra cui, le più conosciute sono latitudine e longitudine, mentre, per la posizione verticale la scelta più comune è data dall'altitudine.

2.1.1 Latitudine

La latitudine di un punto sulla superficie terrestre viene definita come l'angolo formato tra il piano equatoriale e la linea retta che congiunge il punto in questione con il centro della Terra, od un punto ad esso molto vicino (Fig 2.3a).

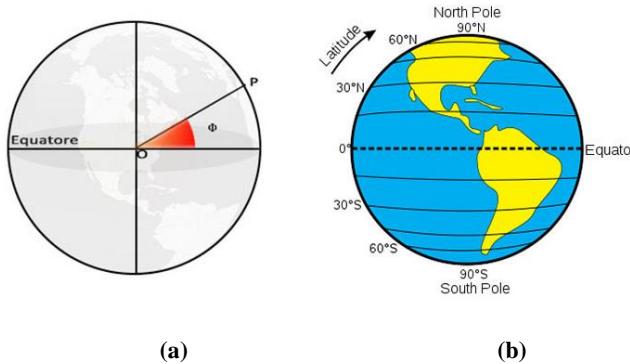


Figura 2.3 (a) Un esempio di latitudine intesa come angolo formato tra l'Equatore e la retta che congiunge il centro della Terra con un punto P sulla superficie terrestre. (b) La superficie terrestre divisa in paralleli.

Unendo tutti i punti che sono situati alla stessa latitudine, si disegnano delle circonferenze ideali sulla superficie terrestre, dette paralleli. I paralleli vengono così chiamati in quanto le circonferenze che li definiscono sono tutte circonferenze parallele sia tra loro, sia all'Equatore (Fig.2.3b).

Questi circoli hanno tutti centro lungo l'asse di rotazione terrestre, ad essi perpendicolare, ed il loro raggio diminuisce con l'aumentare della distanza dal piano equoriale, fino a degenerare ai Poli; pertanto, ai Poli, la circonferenza che descrive il parallelo degenera in un punto ed il suo raggio, quindi, è nullo.

La circonferenza di raggio massimo, al contrario, si ottiene in corrispondenza dell'Equatore.

Per uno sferoide rotante (come il pianeta Terra), viene detto Equatore il parallelo che viene definito per un angolo di latitudine pari a 0° .

Il piano equoriale, piano su cui giace l'Equatore, divide la superficie terrestre in due emisferi, quali, l'Emisfero Boreale (a Nord dell'Equatore), e l'Emisfero Austral (a Sud dell'Equatore).

Si definisce Polo geografico uno dei due punti di intersezione tra l'asse di rotazione della Terra e la superficie terrestre. Tale punto corrisponde ad un angolo di latitudine pari a 90° .

Sul pianeta Terra, dunque, si riconoscono, un Polo Nord Geografico, che definisce un angolo di latitudine di 90°N (Nord) ed un Polo Sud Geografico, che, in modo analogo, definisce un angolo di latitudine di 90°S (Sud).

A fronte di quanto detto, si può, dunque, definire la latitudine come la distanza angolare ϕ tra un parallelo e l'Equatore, dove ϕ varia nel range $[0^\circ, 90^\circ]$.

Al valore di tale angolo, va aggiunto il riferimento relativo all'emisfero in cui è situata l'entità che si vuole georeferenziare.

Se ad, esempio, si vuole specificare la posizione del Polo Nord, situato nell'Emisfero Boreale, la latitudine di tale punto sul globo terrestre sarà 90° N , ovvero 90° a Nord dell'Equatore.

Analogamente, per specificare la posizione del Polo Sud, situato nell'Emisfero Australi, la latitudine di tale punto sarà 90° S, ovvero 90° a Sud rispetto al piano equatoriale.

Un altro modo di misurare la latitudine è quello di pensarla come un angolo che assume valore positivo nell'Emisfero Boreale, ed un valore negativo nell'Emisfero Australi.

Pertanto, l'angolo ϕ che esprime la latitudine di un punto sulla superficie terrestre assumerà valori nel range $[-90^\circ, 90^\circ]$.

In questo caso, il Polo Nord, situato nell'Emisfero Boreale, verrà espresso mediante un angolo di latitudine di 90° , mentre il Polo Sud, situato nell'Emisfero Australi, verrà identificato da un angolo di latitudine di -90° .

L'equatore, invece, corrisponde sempre ad una latitudine di 0° .

Infine, esistono diverse tipologie di latitudine; dunque, il valore di tale parametro non è unico, ma varia a seconda del modello matematico che viene utilizzato per rappresentare la forma della Terra. Pertanto, in base all'interpretazione che si dà ad essa, si usa un particolare tipo di latitudine. Si distinguono, dunque:

- **Latitudine geodetica:** angolo formato tra il piano equatoriale e la normale ad un punto sulla superficie scelta come modello di rappresentazione della forma della Terra (Fig.2.4a).
- **Latitudine geocentrica:** angolo formato tra il piano equatoriale ed il raggio terrestre, congiungente il centro della Terra con un punto sulla sua superficie (Fig.2.4b).
- **Latitudine sferica:** angolo formato tra la normale ad un punto su una superficie sferica di riferimento ed il piano equatoriale di tale sfera (Fig.2.4c).
- **Latitudine astronomica:** angolo formato tra la direzione della forza di gravità agente su un punto della superficie terrestre ed il piano equatoriale.

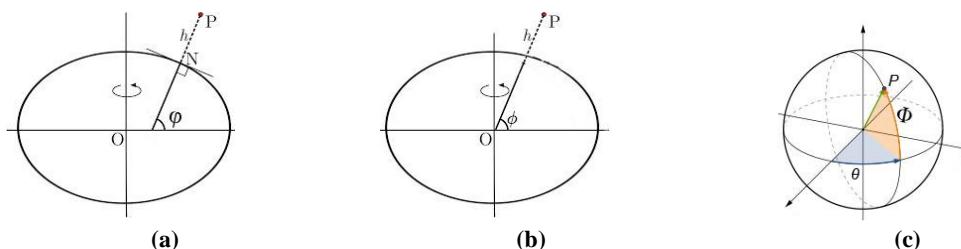


Figura 2.4. Le diverse tipologie di latitudine. (a) Latitudine geodetica. (b) Latitudine geocentrica. (c) Latitudine sferica.

2.1.2 Longitudine

La longitudine di un punto sulla superficie terrestre viene definita come l'angolo formato tra il piano passante per il Meridiano Fondamentale ed il piano passante per il punto in questione che congiunge i Poli.

Partendo dal Polo Nord e terminando al Polo Sud, unendo tutti i punti che sono situati alla stessa longitudine, si disegnano delle metà di circonferenze immaginarie sulla superficie terrestre, tutte aventi stessa e massima lunghezza, dette meridiani (Fig.2.5b).

La distanza tra un meridiano e l'altro diminuisce con l'aumento della distanza dal piano equatoriale, fino a diventare nulla ai Poli, dove ogni meridiano converge in un punto.

Inoltre, ogni meridiano è perpendicolare a tutti i circoli di latitudine che tagliano orizzontalmente la superficie terrestre.

Tra tutti i meridiani che possono essere tracciati sulla superficie terrestre, viene distinto il Meridiano Fondamentale, ovvero il Meridiano di Greenwich, che definisce una longitudine pari a 0° .

Dunque, a fronte di questa considerazione, la longitudine di un punto sulla superficie terrestre può essere definita come l'angolo θ tra il Meridiano di Greenwich e il meridiano su cui giace il punto in questione (Fig. 2.5a).

Tale angolo θ varia nel range $[0^\circ, 180^\circ]$.

Ogni meridiano, essendo metà di una circonferenza immaginaria, possiede anche un suo anti-meridiano, ovvero l'altra metà della circonferenza immaginaria, posta a 180° di distanza dal meridiano considerato. Pertanto, il Meridiano Fondamentale ed il suo anti-meridiano formano una circonferenza immaginaria che divide la superficie terrestre in due emisferi, l'Emisfero Orientale (a Est del Meridiano di Greenwich) e l'Emisfero Occidentale (a Ovest del Meridiano di Greenwich).

Pertanto, al valore dell'angolo di longitudine va aggiunto il riferimento relativo all'emisfero in cui si trova l'entità che si vuole georeferenziare.

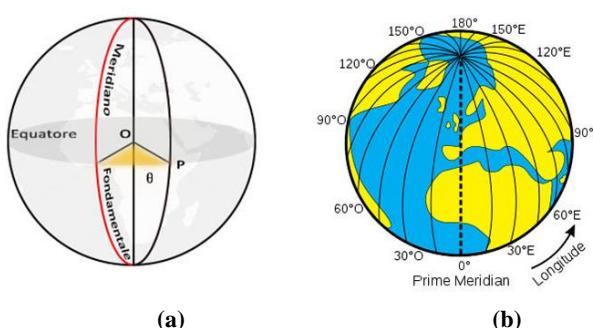


Figura 2.5. (a) Un esempio di longitudine intesa come angolo tra il Meridiano Fondamentale ed il meridiano su cui giace il punto P. (b) La superficie terrestre divisa in meridiani.

Ad esempio, il Meridiano di Monte Mario, a Roma, è situato nell'Emisfero Orientale, dunque, la sua longitudine sarà di $12^{\circ} 27'$ E (Est), ovvero $12^{\circ} 27'$ ad Est del Meridiano di Greenwich.

Analogamente, il Meridiano di Ferro, situato nell'isola di El Hierro (Canarie), è situato nell'Emisfero Occidentale, dunque, la sua longitudine sarà di $17^{\circ} 39'$ W (West, Ovest), ovvero $17^{\circ} 39'$ ad Ovest del Meridiano di Greenwich.

Un altro modo di misurare la longitudine è quello di pensarla come un angolo che assume un valore positivo nell'Emisfero Orientale, ed un valore negativo nell'Emisfero Occidentale.

Pertanto, l'angolo θ che esprime la longitudine di un punto sulla superficie terrestre assumerà valori nel range $[-180^{\circ}, 180^{\circ}]$.

In questo caso, il Meridiano di Monte Mario (Roma), situato nell'Emisfero Orientale, verrà espresso mediante un angolo di longitudine di $12^{\circ} 27'$, mentre il Meridiano di Ferro (Canarie), situato nell'Emisfero Occidentale, verrà identificato da un angolo di longitudine di $-17^{\circ}39'$.

Il Meridiano di Greenwich, invece, corrisponde sempre ad una longitudine di 0° .

Il calcolo della longitudine, però, non può essere molto accurato in prossimità dei Poli, essendo questa ad essi singolare. Vi è, inoltre, una discontinuità al meridiano che corrisponde ai $\pm 180^{\circ}$ di longitudine, che deve essere maneggiata con cura nei calcoli.

Riepilogando quanto detto, si può constatare, dunque, che latitudine e longitudine sono due grandezze angolari, in quanto esse individuano una posizione attraverso la misura di due angoli. Pertanto, volendo georeferenziare lo Stadio Olimpico (Roma) attraverso un sistema di coordinate angolari, ad esempio, si avrà che la sua posizione sarà data da una latitudine di $41^{\circ}56'N$ e da una longitudine di $12^{\circ} 27'E$.

2.1.3 Altitudine

Nei paragrafi precedenti (**2.1.1** e **2.1.2**) sono stati descritti due parametri che individuano una posizione orizzontale sulla superficie terrestre. In questo paragrafo, invece, verrà presentato un parametro che permette di localizzare una posizione verticale.

Una posizione verticale è individuata dall'elevazione, dall'altitudine o dalla profondità. E' bene specificare, però, che questi vocaboli non sono sinonimi tra loro.

Con il termine elevazione di un luogo geografico si intende l'altezza di tale luogo al di sopra o al di sotto di un punto fisso di riferimento, scelto un datum opportuno (Fig.2.6). Solitamente, l'elevazione viene calcolata in direzione della forza di gravità che agisce sul luogo considerato.

L'altitudine, invece, viene definita come l'altezza di un'entità dal livello del mare, preso come livello 0.

Tale altezza dal livello del mare viene definita altezza assoluta, mentre la differenza tra due altitudini diverse viene detta altezza relativa (altezza di una torre, ad esempio, o l'altezza a cui vola un aereo rispetto alla cima di una montagna, Fig. 2.7).

L'altitudine, quindi, è una distanza verticale che viene misurata soltanto verso l'alto.

La profondità, al contrario, è una distanza verticale che viene misurata soltanto verso il basso, dunque, in direzione perpendicolare al di sotto di una superficie presa come riferimento.

Tra quelli elencati, il parametro più comune usato per determinare una posizione verticale è l'altitudine.

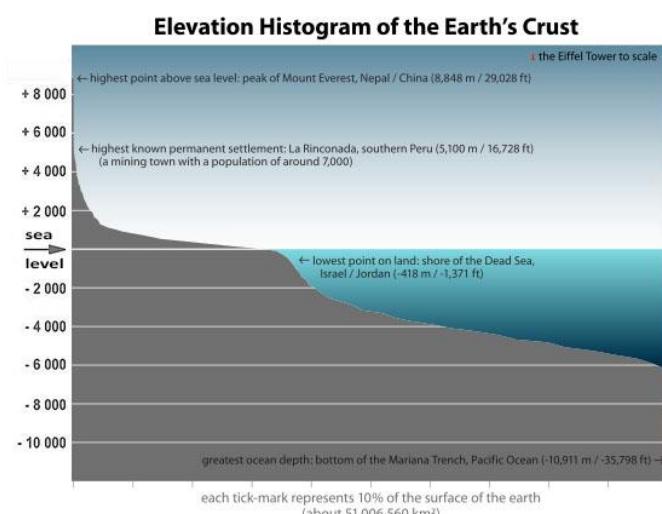


Figura 2.6. Istogramma dell'elevazione della Crosta terrestre.

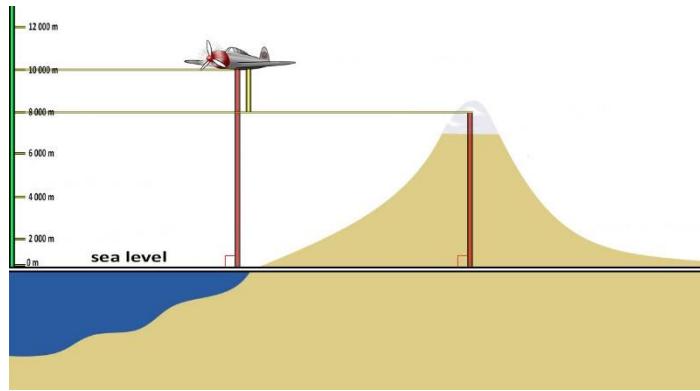


Figura 2.7. In rosso, un esempio di altitudine, definita come altezza assoluta a cui vola un aereo rispetto al livello del mare. In giallo, un esempio di altezza relativa, definita come la differenza tra l'altitudine a cui vola l'aereo e l'altitudine della cima di una montagna. In entrambi i casi, si considera come altitudine la distanza verticale perpendicolare al livello del mare.

Dato un luogo geografico o un'entità, come calcolare la sua altitudine?

Solitamente, l'altitudine di un'entità viene determinata considerando la sua distanza verticale in direzione perpendicolare al livello del mare, o livello 0 (Fig.2.7). Negli Stati Uniti e nel Regno Unito, l'unità di misura con cui viene misurata l'altitudine è il piede, mentre in Europa e nel resto del mondo l'unità di misura utilizzata è il metro. Se l'entità risulta essere un comune od una città, o più in generale, un centro abitato, la sua altitudine sarà data dal valore dell'altitudine a cui è situato il piano terra del municipio di tale comune o città.

Ad esempio, la città di Roma, intesa come centro abitato, è situata ad un'altitudine di 20 m dal livello abitato, mentre l'aeroporto di Roma Fiumicino è situato ad un'altitudine di 5 m.

2.2 Coordinate Cartesiane

Un sistema di coordinate cartesiane è dato da terne ordinate di numeri che individuano univocamente una posizione nello spazio.

In tale sistema, ogni terna fa riferimento a tre rette tra loro ortogonali, dette assi cartesiani, che si intersecano in un punto, denominato origine degli assi. Gli assi vengono indicati con le lettere “ x ”, “ y ”, “ z ”, mentre l’origine viene indicata con la lettera “ O ”.

Dunque, dato un sistema di assi cartesiani, ogni punto \mathbf{P} dello spazio è individuato dalla terna di numeri

$$\mathbf{P} = (x_p, y_p, z_p) \quad (2.2.1)$$

Questo significa che ogni punto, e, quindi, anche ogni posizione è definita da tre coordinate cartesiane (Fig. 2.8):

- La coordinata x_p viene detta ascissa del punto \mathbf{P} e corrisponde alla proiezione del punto \mathbf{P} sull’asse delle x .
- La coordinata y_p viene detta ordinata del punto \mathbf{P} e corrisponde alla proiezione del punto \mathbf{P} sull’asse delle y .
- La coordinata z_p viene detta quota del punto \mathbf{P} e corrisponde alla proiezione del punto \mathbf{P} sull’asse delle z .

Poiché gli assi cartesiani sono assi orientati, per specificare la direzione ed il verso in cui puntano, a tali assi, di solito, vengono associati dei versori, ognuno dei quali è parallelo ad ogni asse. Ad esempio,

- All’asse x è associato il versore \mathbf{i} ,
- All’asse y è associato il versore \mathbf{j} ,
- All’asse z è associato il versore \mathbf{k} , con

$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.2.2)$$

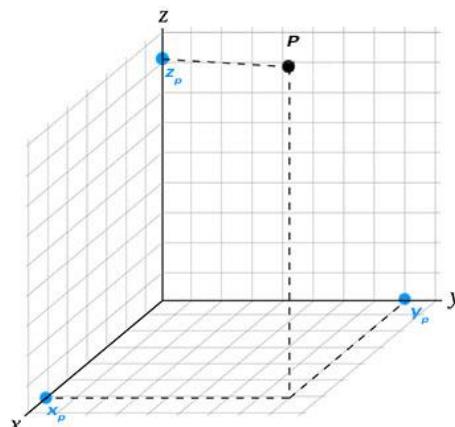


Figura 2.8. Un punto \mathbf{P} nello spazio rappresentato in un sistema di riferimento cartesiano definito dalle coordinate x_p, y_p, z_p (in blu).

Quindi, il versore **i** indicherà la direzione dell'asse *x*, il versore **j** la direzione dell'asse *y* e il versore **k** la direzione dell'asse *z*.

Oltre agli assi cartesiani, anche il punto **P** può essere pensato in termini vettoriali. Infatti, si può pensare **P** come segmento orientato **p** (vettore) che ha punto di applicazione in **O** = (0,0,0) e punto finale in **P**.

In queste condizioni, dato un sistema di riferimento di assi cartesiani con origine in **O** e data una posizione, rappresentata da un vettore **p** nello spazio, si ha che

- l'ascissa x_p del punto **P** corrisponde al prodotto scalare tra i vettori **p** e **i**,
- l'ordinata y_p del punto **P** corrisponde al prodotto scalare tra i vettori **p** e **j**,
- la quota z_p del punto **P** corrisponde al prodotto scalare tra i vettori **p** e **k**.

In formule,

$$x_p = \langle \mathbf{p}, \mathbf{i} \rangle \quad y_p = \langle \mathbf{p}, \mathbf{j} \rangle \quad z_p = \langle \mathbf{p}, \mathbf{k} \rangle \quad (2.2.3)$$

Un sistema di coordinate cartesiane può essere sfruttato per descrivere posizioni geografiche, in quanto tale sistema consente di semplificare alcuni calcoli matematici. In tal caso, dunque, si può parlare di Sistemi di Coordinate Geografiche e Cartesiane.

Tra i Sistemi di Coordinate Geografiche e Cartesiane rientrano il sistema di coordinate ECEF ed il sistema di coordinate definite dal piano tangente locale. Tali sistemi verranno delineati nei prossimi paragrafi (2.2.1 e 2.2.2).

2.2.1 Coordinate ECEF

Il sistema di coordinate ECEF è un Sistema di Coordinate Geografiche e Cartesiane destrorso.

L'acronimo ECEF indica un sistema Earth-Centered Earth-Fixed.

In tale sistema, l'origine **O** degli assi cartesiani *x*, *y*, *z* coincide con il centro della Terra; più in particolare, l'origine **O** viene fatta coincidere con il centro di massa del pianeta (Fig.2.9).

Per tale scelta di **O**, dunque, questo sistema viene definito Earth-Centered.

Essendo un Sistema Geografico-Cartesiano, nel sistema ECEF, una posizione geografica viene determinata mediante le coordinate cartesiane *x*, *y*, *z*.

In tal caso, però, ogni terna di valori farà riferimento ad un punto specifico sul pianeta Terra.

Ad esempio, le coordinate ECEF che individuano la posizione del Colosseo a Roma sono

$$\begin{cases} x = 4\ 630\ 471.322551522\ m \\ y = 1\ 025\ 892.063267668\ m \\ z = 4\ 253\ 967.601680439\ m \end{cases} \quad (2.2.1.1)$$

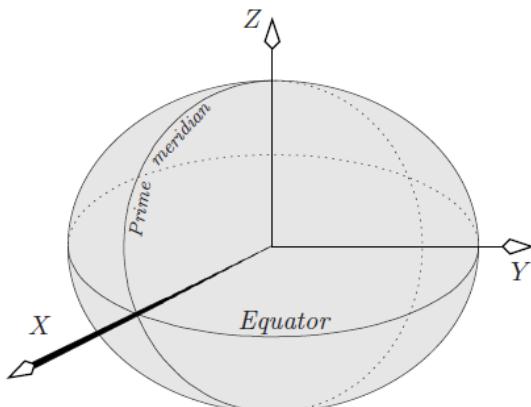


Figura 2.9. Il sistema di coordinate ECEF (Earth Fixed - Earth Centered) in [13].

Come si può constatare dall'esempio, queste coordinate sono date da valori dell'ordine di 10^6 metri, in quanto esprimono la posizione in metri del Colosseo rispetto al centro della Terra, coincidente con l'origine del sistema.

Nel sistema ECEF (Fig.2.9), l'asse x corrisponde all'asse che interseca la superficie terrestre nel punto $0^\circ\text{N } 0^\circ\text{E}$ (0° di latitudine, 0° di longitudine), ovvero il punto di intersezione tra l'Equatore ed il Meridiano di Greenwich.

L'asse y corrisponde all'asse che interseca la superficie terrestre nel punto $0^\circ\text{N } 90^\circ\text{E}$, ovvero 0° N di latitudine e 90° E di longitudine (punto situato nell'Oceano Indiano tra lo Sri Lanka e l'Indonesia).

L'asse z corrisponde all'asse passante per i poli terrestri, diretto dal Polo Sud al Polo Nord.

E' bene precisare che l'asse z del sistema ECEF non coincide esattamente, istante per istante, con l'asse di rotazione terrestre, in quanto quest'ultimo è soggetto ad una leggera oscillazione (Polodìa, movimento dei poli).

Inoltre, in base a questo orientamento degli assi, il piano equatoriale della Terra giace sempre nel piano xy . Ciò significa che tale sistema di assi è un sistema di assi cartesiani fisso, che ruota in modo solidale alla rotazione della Terra; da questo risultato, ne consegue che anche le coordinate cartesiane di un punto sulla superficie terrestre rimarranno invariate.

Per tale motivo, dunque, questo sistema viene denominato sistema Earth-Fixed.

Per esprimere una posizione geografica, il sistema delle coordinate ECEF richiede un modello relativo alla rappresentazione matematica della forma della Terra.

Come verrà specificato nel paragrafo 2.3, a seconda del modello che viene utilizzato, il valore delle coordinate ECEF cambia; una stessa posizione geografica, dunque, può avere valori diversi di coordinate ECEF, a seconda della forma che assume la superficie terrestre.

Scelto il modello, si possono distinguere due tipologie di coordinate ECEF:

- **coordinate ECEF Globali:** l'origine degli assi cartesiani è posta nel centro della Terra, preso come origine globale; dunque, ogni posizione geografica viene ad esso riferita (Fig.2.9).
- **coordinate ECEF Locali:** è possibile scegliere come origine delle coordinate un luogo sulla superficie terrestre. In questo caso, basterà traslare il sistema di assi cartesiani dal centro della Terra al punto scelto come nuova origine locale (Fig.2.10).

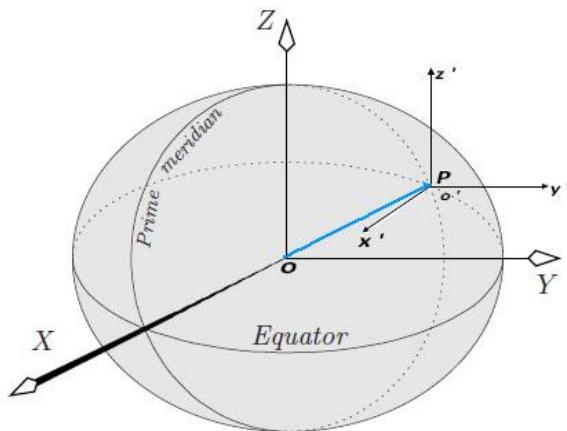


Figura 2.10. Il sistema $x'y'z'$ di coordinate ECEF locali. Se \mathbf{P} è un punto sulla superficie terrestre, tale sistema si ottiene traslando l'origine \mathbf{O} del sistema XYZ nel punto \mathbf{P} . Le coordinate che descrivono una posizione geografica saranno calcolate rispetto al punto \mathbf{P} [13].

2.2.2 Piano Tangente Locale

Il Piano Tangente Locale, come il sistema ECEF, è un Sistema di Coordinate Geografiche – Cartesiane destrorso.

Dato un luogo geografico sulla Terra, indicato con P, si definisce Piano Tangente Locale il piano tangente alla superficie terrestre nel punto P (Fig.2.11).

Essendo questo un sistema di coordinate cartesiane, all'interno dell'area ricoperta da questo piano si assume che la Terra sia piatta; dunque, tale piano non può coprire grandi distanze, ma avrà un'estensione di circa 4- 10 km intorno al punto P. Pertanto, questo sistema di coordinate farà riferimento solo al luogo in cui è situato il punto P, ovvero il luogo geografico preso in considerazione. Per questo motivo tale piano tangente viene detto locale.

Ogni Piano Tangente Locale definisce un sistema di tre coordinate cartesiane locali, di cui una rappresenta una direzione verticale, mentre le altre due rappresentano una direzione orizzontale. Il punto P viene preso come origine di questo sistema.

A seconda del modo in cui sono orientati gli assi cartesiani che definiscono questo nuovo sistema locale, si distinguono:

- **le coordinate NED**
- **le coordinate ENU**

In aeronautica, il Piano Tangente Locale gioca un ruolo molto importante, in quanto permette di individuare la posizione di un velivolo ed il luogo geografico che questo sta sorvolando, rispetto alla località in cui è posizionato un radar di riferimento.

Questo sarà il principio guida che verrà applicato, in questa tesi, alla camera virtuale di Google Earth Studio, al fine di esprimere la sua posizione mediante coordinate locali.

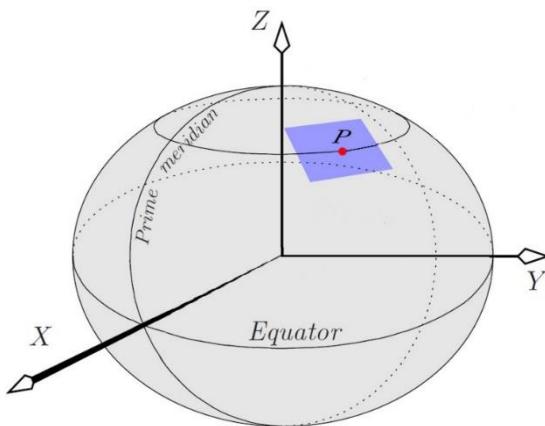


Figura 2.11. Piano tangente locale alla superficie terrestre definito nel punto **P**, in [13]. Il punto **P** diventa l'origine del nuovo sistema di coordinate locali definito da questo piano.

2.2.2.1 Coordinate NED

Le coordinate NED (North – East - Down) sono un Sistema di Coordinate Cartesiane Locali.

Dato un punto P sulla superficie terrestre, preso come origine, il sistema NED è costituito da tre assi cartesiani, “*North*” (Nord), “*East*” (Est), “*Down*” (Sotto).

Per il modo in cui sono orientati gli assi, questo sistema definisce un Piano Tangente Locale alla superficie terrestre nel punto P (Fig.2.12). Infatti,

- l’asse “*North*” è tangente ai meridiani,
- l’asse “*East*” è tangente ai paralleli,
- l’asse “*Down*” è diretto lungo la normale alla superficie terrestre nel punto P, verso il basso.

In generale, a seconda del modello che viene utilizzato per rappresentare matematicamente la forma della Terra, la retta che definisce questo asse può non passare per il centro della Terra.

Il sistema di coordinate NED viene utilizzato in ambito aeronautico, in quanto gli oggetti di interesse, di solito, si trovano nell’area sottostante sorvolata dal velivolo. Per indicare la posizione di questi oggetti con valori non negativi, è, dunque, necessario usare un asse rivolto verso il basso, ovvero l’asse “*Down*”.

Per convenzione, di solito si associa l’asse “*North*” all’asse x, l’asse “*East*” all’asse y e l’asse “*Down*” all’asse z, essendo il sistema NED un sistema di coordinate cartesiane.

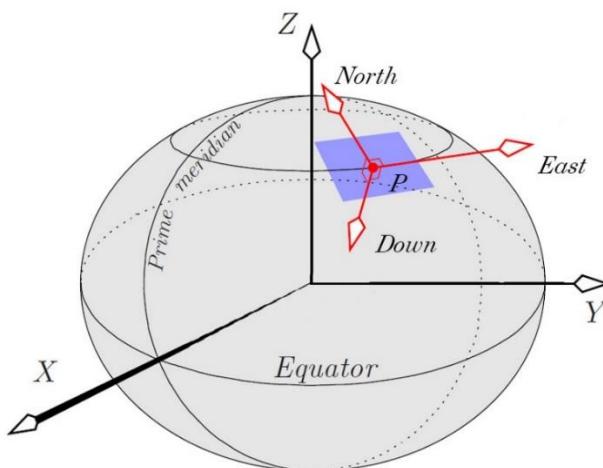


Figura 2.12. Il sistema di coordinate NED definito dal piano tangente locale alla superficie terrestre nel punto P, descritto in [13]. In tale sistema, gli assi *North* ed *East* individuano una posizione orizzontale, mentre l’asse *Down* descrive una posizione verticale.

2.2.2.2 Coordinate ENU

Le coordinate ENU (East – North - Up) sono un Sistema di Coordinate Cartesiane Locali.

Dato un punto P sulla superficie terrestre, preso come origine, il sistema ENU è costituito da tre assi cartesiani, “*East*” (Est), “*North*” (Nord), “*Up*” (Sopra).

Anche in questo caso, per il modo in cui sono orientati gli assi, questo sistema definisce un Piano Tangente Locale alla superficie terrestre nel punto P (Fig.2.13). Infatti,

- l'asse “*North*” è tangente ai meridiani,
- l'asse “*East*” è tangente ai paralleli,
- l'asse “*Up*” è diretto lungo la normale alla superficie terrestre nel punto P, verso l'alto.

Il sistema di coordinate ENU risulta essere una soluzione più intuitiva per la geolocalizzazione di entità sulla Terra rispetto agli altri sistemi di coordinate presentati in questo capitolo. Pertanto, tale sistema trova la sua applicazione nelle operazioni di tracciamento (tracking), ovvero nel tracciare la posizione geografica di oggetti che si muovono sulla porzione di superficie terrestre che rientra all'interno del Piano Tangente Locale.

Per convenzione, di solito si associa l'asse “*East*” all'asse x, l'asse “*North*” all'asse y e l'asse “*Up*” all'asse z, essendo il sistema ENU un sistema di coordinate cartesiane.

In questa tesi, come specificato nel capitolo 5, per esprimere le coordinate locali della camera virtuale di Google Earth Studio verranno utilizzate le coordinate ENU.

La Figura 2.14 mostra un esempio di coordinate ENU, i cui valori sono calcolati rispetto al punto P preso come punto di origine della scena.

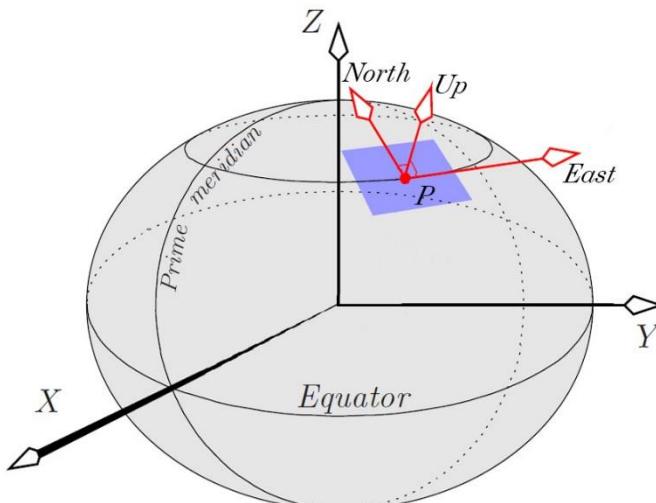


Figura 2.13. Il sistema di coordinate ENU definito dal piano tangente locale alla superficie terrestre nel punto P, descritto in [13]. In tale sistema, gli assi *East* e *North* individuano una posizione orizzontale, mentre l'asse *Up* descrive una posizione verticale.

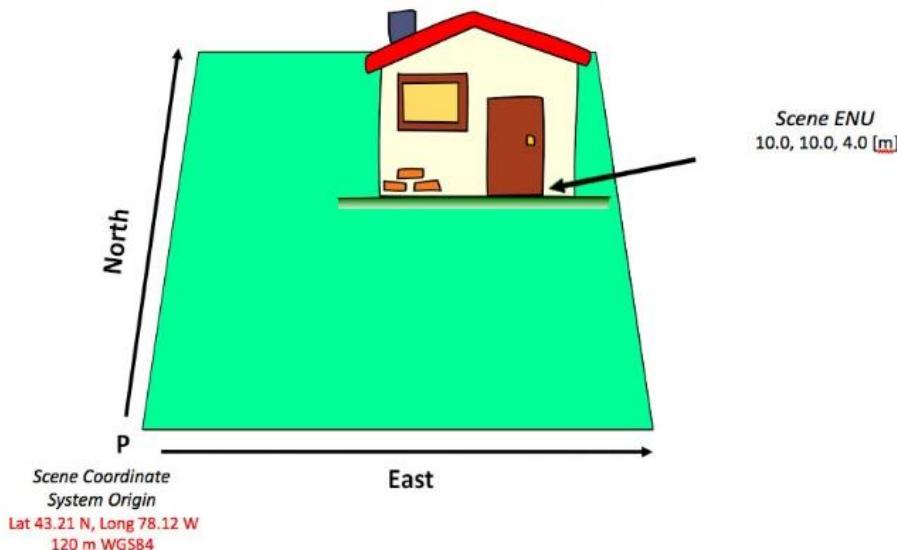


Figura 2.14. Esempio di coordinate ENU calcolate rispetto al punto **P**, preso come punto di origine della scena. Il piano su cui giace la casa è una porzione del piano tangente locale alla superficie nel punto **P**.

2.3 Modelli di rappresentazione per la forma della Terra

Come anticipato nei paragrafi precedenti, per geolocalizzare un punto sulla Terra attraverso un sistema di coordinate geografiche è necessario associare al sistema scelto un modello di rappresentazione matematica della forma della Terra.

Si cerca, dunque, di rappresentare la superficie terrestre attraverso una superficie che può essere descritta in termini matematici; tale superficie, detta superficie di riferimento, è la forma che la Terra assume secondo il modello considerato.

Nel corso della storia, sono stati sviluppati diversi modelli; ogni modello varia a seconda della superficie che viene presa come riferimento, degli scopi e degli ambiti in cui viene applicato, in base alla loro complessità e all'accuratezza con cui si riescono ad approssimare le dimensioni e la forma della superficie terrestre.

Nei prossimi paragrafi verranno delineati alcuni dei modelli più usati, quali,

- **Modello Sferico**
- **Modello Ellittico**

2.3.1 Modello Sferico

Il modello più semplice per descrivere matematicamente la forma della Terra è il modello sferico.

Secondo questo modello, la superficie terrestre viene approssimata con una sfera, detta sfera di riferimento, il cui raggio medio R misura 6 371,005076123 km (Fig.2.15). In questo modo, è possibile descrivere analiticamente tale superficie attraverso la sua equazione cartesiana.

Se si fa coincidere il centro della Terra con il punto $C_0 = (x_0, y_0, z_0)$, allora l'equazione cartesiana della superficie sferica di raggio R con centro in C_0 che approssima la superficie terrestre è data da

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \quad (2.3.1.1)$$

dove il raggio R indica la distanza tra il centro della Terra C_0 ed un punto $P = (x, y, z)$ sulla superficie terrestre.

Sostituendo il punto $C_0 = (x_0, y_0, z_0)$ con $C_0 = (0, 0, 0)$, la (2.3.1.1) diventa

$$x^2 + y^2 + z^2 = R^2 \quad (2.3.1.2)$$

che rappresenta l'equazione cartesiana della sfera con centro nell'origine.

Indipendentemente dal modello considerato, le distanze tra i punti sulla superficie terrestre ed il centro della Terra variano tra i 6 357 e i 6 378 km. Questi valori corrispondono, rispettivamente, alla distanza tra il centro della Terra ed i Poli, detta raggio polare, e alla misura del raggio equatoriale. Da questi dati, risulta, quindi, che il raggio polare è più piccolo di 21 km rispetto al raggio equatoriale e tale differenza corrisponde allo 0,3% della lunghezza totale di quest'ultimo.

Nel modello sferico, invece, questi due valori, coincidono con il raggio medio terrestre R , in quanto tutti i punti di una sfera sono situati alla stessa distanza dal centro.

Questo risultato ci permette di concludere che una superficie sferica può ben approssimare la superficie terrestre, in quanto la Terra differisce solo dello 0,3% dalla sfera che l'approssima.

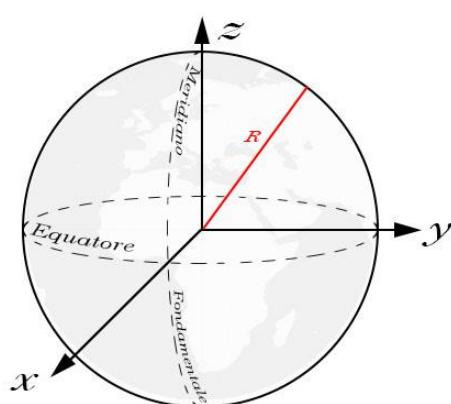


Figura 2.15. La superficie terrestre secondo il modello sferico viene rappresentata da una sfera di raggio medio R , con centro coincidente con il centro di massa della Terra.

Questa differenza dello 0,3% risulta essere un valore molto piccolo, tale da considerarsi trascurabile. Dunque, può essere una buona politica adottare il modello sferico per rappresentare matematicamente la forma della Terra. In alcuni casi, però, a seconda degli scopi, tale valore, se pur piccolo, può rappresentare una differenza significativa, tanto da rendere il modello sferico un modello non più adatto alle esigenze richieste. In tal caso, dunque, è necessario ricorrere ad un altro tipo di modello che rappresenti al meglio la superficie terrestre.

Analizziamo, ora, alcuni aspetti relativi al modello sferico. Nel paragrafo 2.1 abbiamo descritto come un punto sulla Terra può essere individuato mediante una coppia di angoli (ϕ, θ), rispettivamente, di latitudine e longitudine. Se si assume che la forma della Terra sia approssimata da una sfera, ciò implica che tutti i punti sulla superficie terrestre sono situati alla stessa distanza dal centro. Nel modello sferico, questo aspetto gioca un ruolo molto importante nel calcolo della latitudine di un luogo geografico specifico. Nel paragrafo 2.1.1, sono state introdotte diverse tipologie di latitudine; in particolare, la latitudine geocentrica e la latitudine geodetica sono quelle più usate.

Abbiamo definito la latitudine geocentrica come l'angolo formato tra il piano equatoriale e il raggio che congiunge il centro della Terra con un punto sulla sua superficie (Fig.2.16a).

La latitudine geodetica, invece, è stata definita come l'angolo formato tra il piano equatoriale e la retta normale ad un punto sulla superficie terrestre. In generale, tale retta non passa per il centro della Terra (Fig.2.16b).

Sia dato un punto P sulla superficie terrestre, la cui forma è descritta dal modello sferico.

Poiché la Terra viene approssimata da una sfera, la normale al punto P è diretta radialmente, dunque, essa congiunge il punto P con il centro della Terra.

Pertanto, si può affermare che, nel modello sferico, la latitudine geodetica e quella geocentrica coincidono, come mostrato in Fig.2.16.

In particolare, quest'ultime corrispondono alla latitudine sferica, descritta nel paragrafo 2.1.1.

Da questo risultato ne consegue che, quando si adotta questo modello, la latitudine a cui si fa riferimento per la geolocalizzazione è quella sferica. In particolare, qualsiasi sia il tipo di latitudine considerato, questo non crea variazioni dei risultati nel passare da coordinate LLH (Latitudine, Longitudine e Altitudine) a coordinate ECEF, e viceversa.

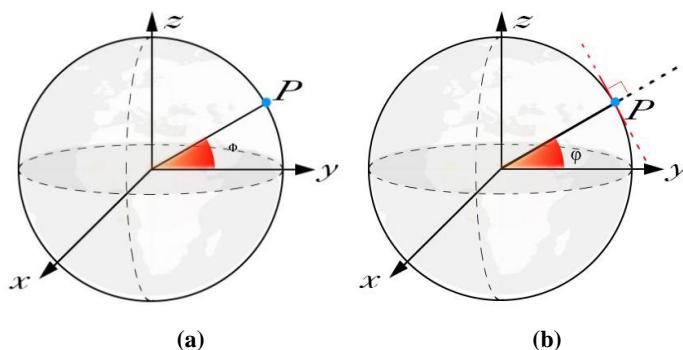


Figura 2.16. (a) Latitudine geocentrica secondo il modello sferico. (b) Latitudine geodetica secondo il modello sferico. Nel modello sferico latitudine geocentrica e geodetica coincidono.

2.3.1.1 Conversione delle coordinate da LLH a ECEF

In questo capitolo, abbiamo presentato la possibilità di esprimere la posizione di un'entità sulla superficie terrestre attraverso diverse tipologie di coordinate geografiche. Abbiamo anche evidenziato come queste siano tra loro equivalenti, permettendo di passare da un sistema all'altro attraverso formule ed equazioni matematiche.

Descriveremo, in questo paragrafo, le formule che permettono di convertire le coordinate di latitudine, longitudine e altitudine (coordinate LLH) in coordinate ECEF.

La prima differenza che bisogna evidenziare tra queste due tipologie di coordinate è quella relativa al fatto che le prime costituiscono un sistema di coordinate sferiche, dove le posizioni vengono espresse mediante angoli (paragrafo 2.1), mentre le seconde corrispondono a coordinate cartesiane (paragrafo 2.2). Dunque, per mettere in relazione questi due sistemi di coordinate è necessario far ricorso alle funzioni trigonometriche, le quali permetteranno il passaggio da un sistema all'altro.

Sia dato un punto P sulla superficie terrestre.

Sia (x_p, y_p, z_p) la terna di coordinate cartesiane che descrive la posizione del punto P in coordinate ECEF.

Siano, invece, ϕ, θ, h , rispettivamente la latitudine, longitudine e altitudine del punto P sulla Terra.

Allora, la conversione da coordinate LLH a coordinate ECEF è data dalla formula

$$\begin{cases} x_p = (R + h) \cos(\phi) \cos(\theta) \\ y_p = (R + h) \cos(\phi) \sin(\theta) \\ z_p = (R + h) \sin(\phi) \end{cases} \quad (2.3.1.1.1)$$

dove R è il raggio medio della Terra dato dal modello sferico.

L'equazione (2.3.1.1.1) permette, quindi, di scrivere le coordinate sferiche in coordinate cartesiane, prendendo come riferimento la sfera di raggio

$$R_{p,h} = R + h \quad (2.3.1.1.2)$$

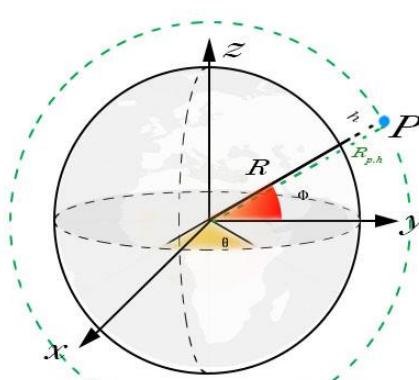


Figura 2.17. Sfera di raggio R (in nero), su cui il punto P è dato dall'angolo di latitudine ϕ , dall'angolo di longitudine θ e dall'altitudine h. La sfera di raggio $R_{p,h}$ (in verde) che viene utilizzata per la conversione delle coordinate da LLH ad ECEF.

Pertanto, possiamo riscrivere la (2.3.1.1.1) come

$$\begin{cases} x_p = R_{p,h} \cos(\phi) \cos(\theta) \\ y_p = R_{p,h} \cos(\phi) \sin(\theta) \\ z_p = R_{p,h} \sin(\phi) \end{cases} \quad (2.3.1.1.3)$$

Determiniamo, ora, la formula che permette il passaggio dalle coordinate cartesiane a coordinate sferiche. Dall'equazione (2.3.1.1.3), si ricavano le seguenti catene di eguaglianze

$$\cos(\theta) = \frac{x_p}{R_{p,h} \cos(\phi)} = \frac{x_p}{R_{p,h} \sqrt{1 - \sin^2(\phi)}} = \frac{x_p}{\sqrt{R_{p,h}^2 - z_p^2}} = \frac{x_p}{\sqrt{x_p^2 + y_p^2 + z_p^2 - z_p^2}} \quad (2.3.1.1.4)$$

da cui segue che

$$\theta = \arccos\left(\frac{x_p}{\sqrt{x_p^2 + y_p^2}}\right) \quad (2.3.1.1.5)$$

e

$$\sin(\phi) = \frac{z_p}{R_{p,h}} = \frac{z_p}{\sqrt{x_p^2 + y_p^2 + z_p^2}} \quad (2.3.1.1.6)$$

da cui segue che

$$\phi = \arcsin\left(\frac{z_p}{\sqrt{x_p^2 + y_p^2 + z_p^2}}\right) \quad (2.3.1.1.7)$$

dove

$$R_{p,h} = R + h = \sqrt{x_p^2 + y_p^2 + z_p^2} \quad (2.3.1.1.8)$$

Pertanto, la conversione da coordinate ECEF a coordinate LLH è data dalla formula

$$\begin{cases} \phi = \arcsin\left(\frac{z_p}{\sqrt{x_p^2 + y_p^2 + z_p^2}}\right) \\ \theta = \arccos\left(\frac{x_p}{\sqrt{x_p^2 + y_p^2}}\right) \\ h = \sqrt{x_p^2 + y_p^2 + z_p^2} - R \end{cases} \quad (2.3.1.1.9)$$

E' bene evidenziare che nella (2.3.1.1.9) l'angolo di longitudine θ non è definito ai Poli, ovvero è singolare in quanto si annulla il denominatore. Inoltre, l'angolo di longitudine θ assume valori nel range $[-180^\circ, 180^\circ]$, ed in tal caso si perderebbe il range $[-180^\circ, 0^\circ]$. Pertanto, per il calcolo di θ è necessario fare le seguenti considerazioni:

$$\begin{cases} \theta = \arctg\left(\frac{y_p}{x_p}\right) \text{ se } x > 0 \text{ e } y \geq 0 \\ \theta = \arctg\left(\frac{y_p}{x_p}\right) + 2\pi \text{ se } x > 0 \text{ e } y < 0 \text{ oppure } x < 0 \text{ e } y > 0 \\ \theta = \arctg\left(\frac{y_p}{x_p}\right) + 2\pi \text{ se } x < 0 \text{ e } y \leq 0 \end{cases} \quad (2.3.1.1.10)$$

2.3.2 Modello Ellittico

Per il semplice fatto che è schiacciata ai Poli, la Terra non può essere perfettamente ricondotta ad una sfera; in alcuni casi, questo aspetto rende il modello sferico inadatto. Occorre, dunque, trovare un modello alternativo, che tenga conto di questo appiattimento.

Un buon candidato risulta essere il modello ellittico, il quale considera un ellissoide di rotazione come superficie di riferimento.

Un ellissoide di rotazione è una superficie ottenuta dalla rotazione di un'ellisse intorno ad uno dei suoi assi principali, caratterizzata da

- Un semiasse maggiore a , detto **raggio equatoriale**.
- Un semiasse minore c , detto **raggio polare**.
- **Eccentricità** e , un parametro che misura quanto è schiacciata l'ellisse rispetto ai propri assi, definita dalla formula

$$e = \sqrt{1 - \frac{c^2}{a^2}} \quad (2.4.3.1)$$

- **Flattening o ellitticità** f , un parametro che indica lo schiacciamento dei poli dello sferoide rispetto al suo equatore, dunque, quanto l'ellissoide si avvicina alla forma sferica, dato dal rapporto

$$f = \frac{a - c}{a} \quad (2.4.3.2)$$

A seconda dell'asse intorno a cui avviene la rotazione, un ellissoide viene definito:

- **sferoide prolato**: se la rotazione avviene intorno all'asse maggiore dell'ellisse. La sua forma ricorda quella di un pallone da rugby (Fig.2.18a).
- **sferoide oblatto**: se la rotazione avviene intorno all'asse minore dell'ellisse. La sua forma ricorda quella di un pianeta (Fig.2.18b).

Per le sue caratteristiche, l'ellissoide di rotazione che meglio approssima la forma della Terra è lo sferoide oblato. Dunque, il modello ellittico permette di descrivere matematicamente la forma della Terra, approssimando la sua superficie con uno sferoide oblato (Fig.2.19). Pertanto, anche in questo caso, possiamo descrivere analiticamente questa superficie attraverso la sua equazione cartesiana.

Facendo coincidere l'origine dell'ellissoide con il centro della Terra, l'equazione dello sferoide oblato è data da

$$\frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{c^2} = 1 \quad (2.3.2.3)$$

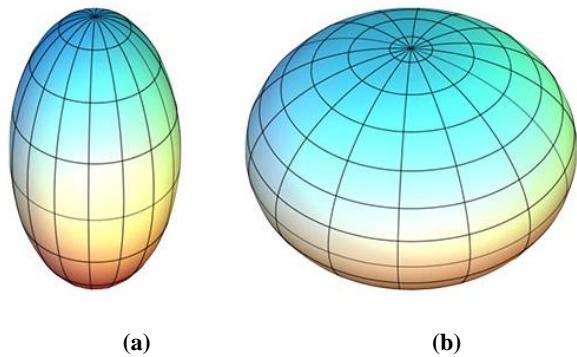


Figura 2.18. (a) Sferoide prolato ottenuto dalla rotazione di un'ellisse intorno al suo asse maggiore. (b) Sferoide oblato ottenuto dalla rotazione di un'ellisse intorno al suo asse minore.

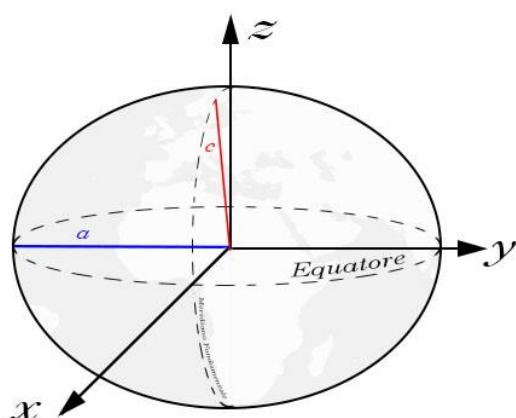


Figura 2.19. La superficie terrestre secondo il modello ellittico viene rappresentata da uno sferoide oblato con raggio equatoriale a e raggio polare c , avente centro coincidente con il centro di massa della Terra.

L’ellissoide che viene preso come riferimento da questo modello, viene utilizzato sia per approssimare il pianeta Terra nella sua totalità, sia per approssimare una porzione di essa, e, dunque, definire un datum con orientamento locale. A seconda di come viene usato questo ellissoide, sia la sua forma, che il suo valore di flattening possono variare in base ai valori che assumono i parametri di tale ellissoide, scelto come superficie di riferimento.

Al giorno d’oggi, l’ellissoide di riferimento standard che viene usato per rappresentare matematicamente la forma della Terra è quello descritto nel World Geodetic System (WGS).

2.3.2.1 Modello WGS84

Il World Geodetic System è il modello matematico standard che rappresenta la forma ellittica della Terra. Tale modello viene applicato in tanti ambiti, tra i quali, spiccano la cartografia, il sistema di navigazione satellitare GPS e la navigazione aerea. In particolare, al giorno d’oggi, di questo modello si usa la sua versione più recente, ovvero il WGS84 (World Geodetic System elaborato nel 1984).

L’ellissoide di riferimento associato a questo modello è uno sferoide oblato avente

- semiasse maggiore $a = 6\ 378\ 137$ metri
- semiasse minore $c = 6\ 356\ 752, 314245$ metri
- eccentricità $e = 0.081819$
- flattening $f = \frac{1}{298,257}$

con

- origine coincidente con il centro di massa del pianeta;
- asse z passante per il Polo Nord;
- asse x passante per il punto di intersezione tra l’equatore ed il Meridiano di Greenwich;
- asse y orientato in modo da formare un sistema di assi cartesiani destrorso;

in quanto il modello WGS84 associa anche un sistema di assi cartesiani al suo ellissoide di riferimento.

Come il modello sferico, anche il modello WGS84 permette di localizzare una posizione sulla superficie terrestre mediante angoli di latitudine e longitudine (ϕ, θ) .

Se si sceglie una sfera come superficie di riferimento, la latitudine di un luogo sarà data dalla latitudine sferica, la quale coincide sia con la latitudine geodetica sia con quella geocentrica (paragrafo 2.3.1). Queste due tipologie di latitudine, però, non saranno più coincidenti nel caso dell’ellissoide.

Infatti, nel modello WGS84 si avrà che

- la latitudine geocentrica sarà sempre definita come l'angolo formato tra il piano equatoriale e la retta che congiunge il centro della Terra con un punto sulla sua superficie (Fig.2.20a).
- La latitudine geodetica, invece, sarà definita come l'angolo formato tra il piano equatoriale e la retta normale ad un punto sulla superficie ellittica (Fig.2.20b). Di conseguenza, la retta normale non sarà una retta passante per l'origine, ad eccezione dei punti dati dall'intersezione tra l'ellissoide e i raggi equatoriale e polare.

Dunque, la latitudine geodetica e latitudine geocentrica coincidono solo ai Poli e per quei punti che giacciono all'Equatore.

E' importante distinguere queste due tipologie di latitudine quando si ricorre al modello ellittico, in quanto, usare un tipo di latitudine anzichè un altro comporta risultati diversi nei calcoli. Tuttavia, a seconda delle necessità, è possibile passare da una latitudine geodetica ad una geocentrica secondo la relazione

$$\phi = \arctan((1-f)\tan(\varphi)) \quad (2.3.2.1.1)$$

dove ϕ indica la latitudine geocentrica, φ la latitudine geodetica e f il parametro di flattening dell'ellissoide.

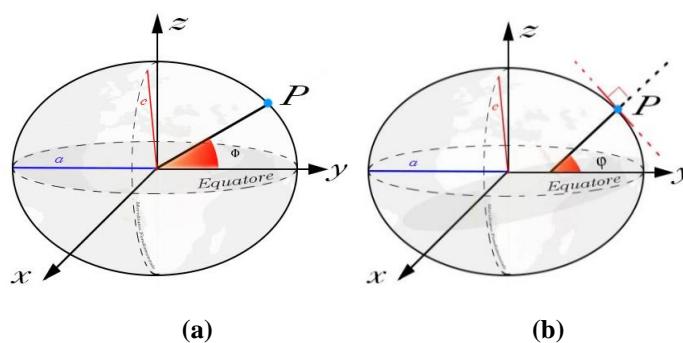


Figura 2.20. (a) Latitudine geocentrica secondo il modello ellittico. (b) Latitudine geodetica secondo il modello ellittico. Nel modello ellittico latitudine geodetica e latitudine geocentrica non coincidono, ad eccezione di quei punti situati ai Poli e all'Equatore.

2.3.2.2 Conversione delle coordinate da LLH a ECEF

Come il modello sferico, anche il modello ellittico prevede la possibilità di convertire le coordinate LLH in coordinate cartesiane ECEF. In particolare, questa conversione di coordinate avviene prendendo come superficie di riferimento l'ellissoide del modello WGS84, descritta nel paragrafo 2.3.2.1 di questo capitolo.

Sia dato, dunque, un punto P sulla superficie terrestre.

Sia (x_p, y_p, z_p) la terna di valori reali che permette di localizzare P in un sistema di coordinate cartesiane.

Siano φ, θ, h rispettivamente, la latitudine geodetica, longitudine e altitudine del punto P sulla Terra.

Allora, il passaggio da coordinate LLH a coordinate ECEF è dato dalla [13]

$$\begin{cases} x_p = \left(\frac{a}{k} + h\right) \cos(\varphi) \cos(\theta) \\ y_p = \left(\frac{a}{k} + h\right) \cos(\varphi) \sin(\theta) \\ z_p = \left(\frac{c}{s} + h\right) \sin(\varphi) \end{cases} \quad (2.3.2.2.1)$$

con

$$k = \sqrt{\cos^2(\varphi) + \frac{c^2}{a^2} \sin^2(\varphi)} \quad (2.3.2.2.2)$$

$$s = \sqrt{\frac{a^2}{c^2} \cos^2(\varphi) + \sin^2(\varphi)} \quad (2.3.2.2.3)$$

Inoltre, se necessario, è possibile esprimere l'equazione (2.3.2.2.1) in termini della latitudine geocentrica ϕ , sostituendo il termine φ con il valore di ϕ ricavato dalla (2.3.2.1.1).

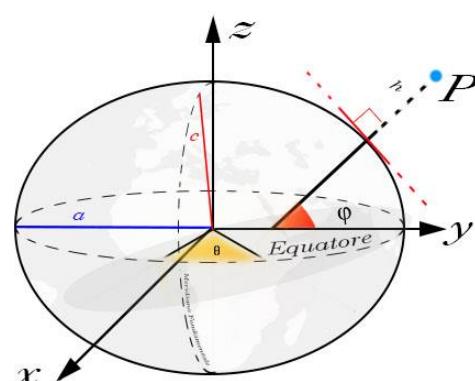


Figura 2.1. Ellissoide del modello WGS84 sul quale il punto P è dato dall'angolo di latitudine geodetica φ , dall'angolo di longitudine θ e dall'altitudine h .

Il passaggio inverso di coordinate da ECEF a LLH richiede calcoli complicati; per questo motivo, in questa tesi, ci limitiamo a riportare le equazioni descritte in [13]:

$$h = \frac{\sqrt{x_p^2 + y_p^2}}{\cos(\varphi)} - \frac{a^2}{\sqrt{a^2 \cos^2(\varphi) + c^2 \sin^2(\varphi)}} \quad (2.3.2.2.4)$$

$$\theta = \arccos\left(\frac{x_p}{\sqrt{x_p^2 + y_p^2}}\right) \quad (2.3.2.2.5)$$

$$\varphi = \operatorname{arctg}\left(\frac{a^2}{b^2} \frac{z_p}{\sqrt{x_p^2 + y_p^2}}\right) \quad (2.3.2.2.6)$$

Capitolo 3

Geometria delle Rotazioni

In questo capitolo verranno presentati gli aspetti matematici e geometrici che definiscono le rotazioni come trasformazioni geometriche dello spazio, con riferimento ad alcuni concetti di algebra lineare. Nell'ultimo paragrafo del capitolo, verranno descritte le rotazioni che sfruttano gli angoli di Eulero per orientare un corpo nello spazio.

Prima di trattare le rotazioni, è bene delineare qualche concetto di algebra lineare.

Sia dato un insieme V . L'insieme V si dice spazio vettoriale se in esso sono definite le operazioni di somma e prodotto per uno scalare, per cui,

- dati $\mathbf{u}, \mathbf{v} \in V$,
- $$(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{u} + \mathbf{v} \tag{3.1}$$

- dati $\mathbf{v} \in V, \lambda \in \mathbb{R}$,
- $$(\mathbf{v}, \lambda) \mapsto \lambda \cdot \mathbf{v} \tag{3.2}$$

Ogni elemento $\mathbf{v} \in V$ viene detto vettore [14].

Siano, ora, V e W due spazi vettoriali.

Una funzione $f : V \rightarrow W$ si dice trasformazione lineare se soddisfa le seguenti due proprietà

- dati $\mathbf{u}, \mathbf{v} \in V$,
- $$f(\mathbf{u} + \mathbf{v}) = f(\mathbf{u}) + f(\mathbf{v}) \tag{3.3}$$

- dati $\mathbf{v} \in V, \lambda \in \mathbb{R}$,
- $$f(\lambda \cdot \mathbf{v}) = \lambda \cdot f(\mathbf{v}) \tag{3.4}$$

Siano \mathbb{R}^n e \mathbb{R}^m due spazi vettoriali, dove \mathbb{R} è l'insieme dei numeri reali.

Sia $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ una trasformazione lineare. Allora, è possibile associare a T una matrice A di dimensioni $m \times n$, per cui le seguenti affermazioni sono equivalenti

- $\mathbf{y} = T(\mathbf{x})$, dove $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$
 - $\mathbf{y} = A \cdot \mathbf{x}$, dove $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$
- $$(3.5)$$

dove \cdot indica il prodotto matrice-vettore.

In questo caso, la matrice A viene detta matrice associata alla trasformazione T .

Chiariti i concetti di spazio vettoriale e di trasformazione lineare, possiamo definire una rotazione come una trasformazione lineare dello spazio che ruota un vettore o un oggetto di un dato angolo θ intorno ad una retta, detta asse di rotazione, lasciando immutati i punti che giacciono su di essa (Fig.3.1).

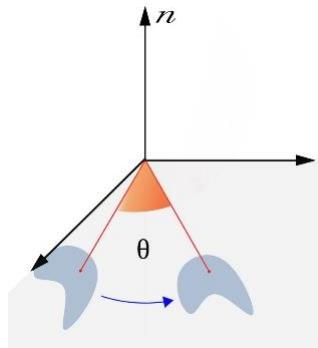


Figura 3.1. Rotazione di un oggetto di angolo θ intorno all'asse di rotazione dato dalla retta n .

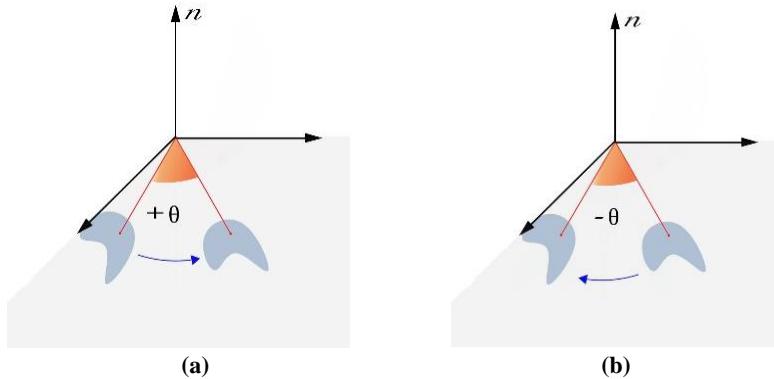


Figura 3.2. (a) Una rotazione che avviene in senso anti-orario ha segno positivo. (b) Una rotazione che avviene in senso orario ha un segno negativo.

Dunque, per definire una rotazione, sono necessari:

- **un verso di rotazione**, che indica se la rotazione avviene in senso orario od in senso antiorario;
- **un angolo di rotazione**, che indica di quanto deve essere ruotato un vettore intorno all'asse di rotazione. In generale, l'angolo di rotazione viene considerato positivo se la rotazione avviene in senso anti-orario (Fig.3.2a), negativo se la rotazione avviene in senso orario (Fig.3.2b). Tale angolo può essere misurato in gradi od in radianti;
- **asse di rotazione**, indicato con il versore che specifica la direzione della retta intorno a cui avviene la rotazione.

Se la trasformazione lineare che descrive la rotazione è una trasformazione del tipo $T: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, allora, alla rotazione di angolo θ intorno ad un dato asse di rotazione si può associare la matrice $R_n(\theta)$, dove R è una matrice 3×3 ; pertanto, possiamo riscrivere la rotazione come

$$\mathbf{x}' = R_n(\theta) \mathbf{x} \quad , \text{ con } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^3 \quad (3.6)$$

dove, \mathbf{n} indica il versore dell'asse di rotazione, θ indica l'ampiezza dell'angolo di rotazione, \mathbf{x} è il vettore da ruotare e \mathbf{x}' è il vettore trasformato dato dalla rotazione di

\mathbf{x} intorno ad \mathbf{n} di angolo θ . Da questo momento in poi, onde evitare ambiguità con il prodotto scalare, il prodotto matrice-vettore verrà indicato omettendo il simbolo \cdot .

La rotazione di un oggetto, detto anche corpo rigido, è una trasformazione che mantiene invariate le relazioni angolari, le distanze tra i punti trasformati e l'orientamento della figura che viene ruotata. Pertanto, la rotazione rientra nella classe delle isometrie. Di conseguenza, è possibile comporre più rotazioni ottenendo ancora una rotazione. Ad esempio, possiamo ruotare un vettore prima intorno ad un asse n di angolo θ e poi intorno ad un asse m di angolo ϕ .

Componendo queste due rotazioni, l'effetto che si otterrà sul vettore dato è ancora quello di subire una rotazione nello spazio. Se a queste due rotazioni associamo, rispettivamente, la matrice $\mathbf{R}_n(\theta)$ alla prima rotazione e la matrice $\mathbf{R}_m(\phi)$ alla seconda, possiamo scrivere questa trasformazione [7] come

$$\mathbf{x}' = \mathbf{R} \mathbf{x} \quad \text{con } \mathbf{x}', \mathbf{x} \in \mathbb{R}^3 \quad (3.7)$$

dove \mathbf{R} è data da

$$\mathbf{R} = \mathbf{R}_m(\phi) \mathbf{R}_n(\theta) \quad (3.8)$$

se le rotazioni $\mathbf{R}_n(\theta)$ e $\mathbf{R}_m(\phi)$ avvengono rispetto ad assi fissi (la configurazione iniziale degli assi rimane invariata al termine delle rotazioni, come illustrato in Fig.3.3), altrimenti,

$$\mathbf{R} = \mathbf{R}_n(\theta) \mathbf{R}_m(\phi) \quad (3.9)$$

se le rotazioni $\mathbf{R}_n(\theta)$ e $\mathbf{R}_m(\phi)$ avvengono rispetto ad assi mobili, ovvero che possiamo scrivere m come [7] [2]

$$\mathbf{m} = \mathbf{R}_n(\theta) \mathbf{m}_0 \quad (3.10)$$

dove \mathbf{m}_0 indica la configurazione iniziale dell'asse \mathbf{m} prima di essere stato ruotato di θ intorno ad \mathbf{n} (Fig.3.4).

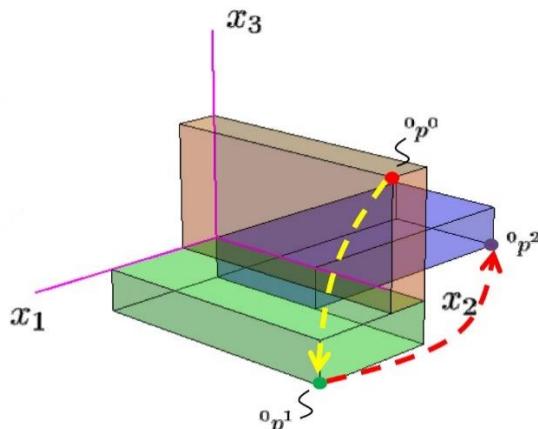


Figura 3.3. Rotazioni intorno ad assi fissi. Gli assi x_1 , x_2 , x_3 rimangono invariati al termine delle rotazioni [7].

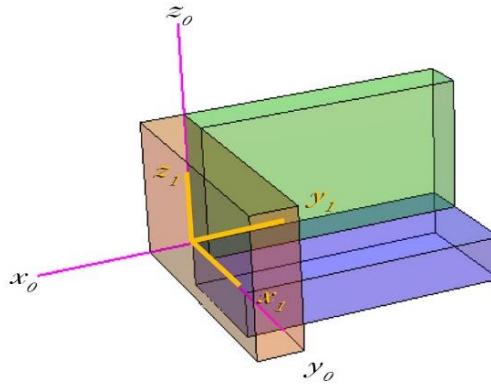


Figura 3.4. Rotazioni intorno ad assi mobili. La configurazione iniziale è data dagli assi x_0, y_0, z_0 . La prima rotazione intorno all'asse z_0 porta gli assi in una nuova configurazione x_1, y_1, z_1 , dove x_1 coincide con y_0 , y_1 coincide con $-x_0$ e z_1 coincide con z_0 [7].

Le rotazioni più comuni sono le rotazioni intorno agli assi cartesiani.

3.1 Rotazione intorno agli assi Cartesiani

Come detto in precedenza, una rotazione è una trasformazione lineare a cui è possibile associare una matrice. La matrice di rotazione, dunque, indica una rotazione all'interno di un sistema di riferimento. Si può prendere come sistema di riferimento, ad esempio, il sistema costituito dagli assi cartesiani. Poiché, come specificato nel paragrafo 2.2, ad ognuno di questi assi è associato un versore, una rotazione nello spazio può essere ottenuta come composizione di tre rotazioni intorno agli assi cartesiani [18]. Per questo motivo, ognuna di queste rotazioni, viene detta “rotazione elementare”.

Dunque, è possibile ruotare un vettore \mathbf{v} nello spazio di un angolo θ intorno ad ognuno degli assi coordinati x, y, z .

Pertanto, la rotazione del vettore \mathbf{v} di angolo θ intorno all'asse x sarà data dalla trasformazione

$$\mathbf{v}' = \mathbf{R}_X(\theta) \mathbf{v} \quad (3.1.1)$$

dove,

$$\mathbf{R}_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (3.1.2)$$

è la matrice che descrive tale rotazione (Fig.3.5a).

In modo analogo si possono ottenere i vettori trasformati \mathbf{v}'' e \mathbf{v}''' , dati rispettivamente dalle rotazioni intorno agli assi y e z . Di conseguenza, la rotazione del vettore \mathbf{v} di angolo θ intorno all'asse y darà il vettore trasformato \mathbf{v}'' , con

$$\mathbf{v}'' = \mathbf{R}_Y(\theta) \mathbf{v} \quad (3.1.3)$$

dove

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.1.4)$$

è la matrice che descrive tale rotazione (Fig.3.5b), mentre la rotazione del vettore \mathbf{v} di angolo θ intorno all'asse z sarà data dal vettore \mathbf{v}'' , con

$$\mathbf{v}'' = \mathbf{R}_Z(\theta) \mathbf{v} \quad (3.1.5)$$

dove

$$\mathbf{R}_Z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1.6)$$

è la matrice che descrive tale rotazione (Fig.3.5c).

Determinata la matrice che rappresenta la rotazione intorno ad un asse cartesiano, si deve stabilire quale sia la direzione della rotazione, ovvero il verso in cui questa avviene. In tal caso, la direzione associata alla rotazione di angolo θ intorno ad uno degli assi cartesiani è data dalla regola della mano destra.

Si pone il pollice nel modo in cui è diretto l'asse di rotazione e le altre dita della mano, stringendosi intorno ad esso, danno il verso in cui avviene la rotazione (Fig.3.6).

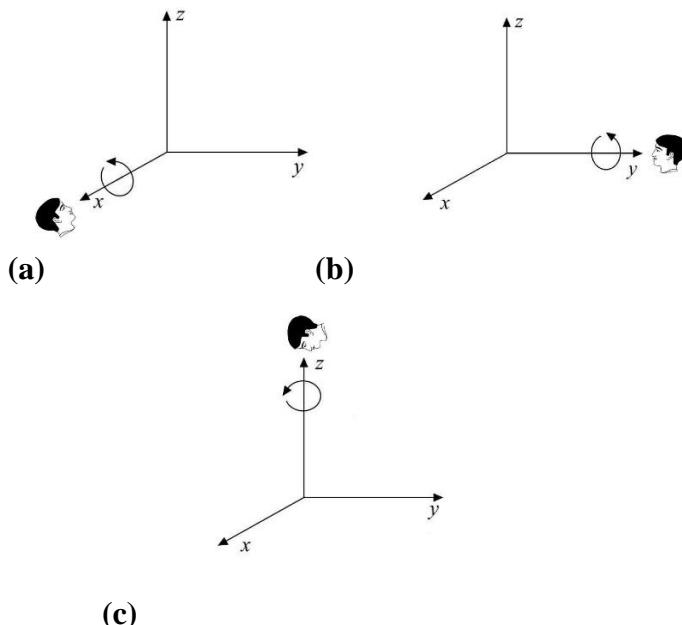


Figura 3.5. (a) Rotazione intorno all'asse x . (b) Rotazione intorno all'asse y . (c) Rotazione intorno all'asse z [18].

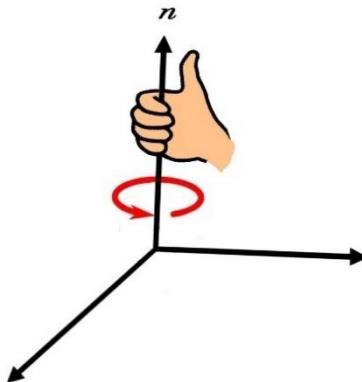


Figura 3.6. Applicazione della regola della mano destra per determinare la direzione di rotazione.

Esempio 3.1.1

Supponiamo di voler ruotare il vettore \mathbf{v} di 90° intorno all'asse z , con

$$\mathbf{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Il vettore trasformato \mathbf{v}'' è dato dalla (3.1.6), per cui

$$\mathbf{v}'' = \mathbf{R}_z(90^\circ) \mathbf{v} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

Ora, il verso di rotazione è dato dalla regola della mano destra. Dunque, posizioniamo il pollice in direzione dell'asse z e stringiamo le altre dita della mano intorno ad esso; le dita si stringono in senso anti-orario, pertanto, il vettore \mathbf{v} verrà ruotato in senso anti-orario di 90° intorno all'asse z .

Esempio 3.1.2

Se, invece, il vettore \mathbf{v} viene ruotato di -90° intorno all'asse z , sempre ricorrendo alla (3.6.1),

$$\mathbf{v}'' = \mathbf{R}_z(-90^\circ) \mathbf{v} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Applicando la regola della mano destra, si ottiene che, ponendo il pollice lungo l'asse z , le dita si stringono in senso anti-orario. Poiché, stavolta, la rotazione è di -90° , dobbiamo considerare il verso opposto rispetto a quello che ci viene dato dalla regola della mano destra; pertanto, in questo caso, la rotazione sarà in senso orario.

Lo stesso risultato si ottiene considerando la rotazione di 90° del vettore \mathbf{v} intorno all'asse $-z$. In questo caso, ponendo il pollice lungo il semiasse negativo delle z , le altre dita della mano destra si stringono in senso orario.

Oltre agli assi cartesiani, può essere scelto come asse di rotazione un vettore qualsiasi dello spazio. Anche in questo caso, sebbene più complesso rispetto a quello presentato

in questo paragrafo, è possibile scrivere ed associare una matrice \mathbf{R} di rotazione a questa trasformazione. Tratteremo questo argomento nel prossimo paragrafo.

3.2 Rotazioni intorno ad un asse arbitrario

Nel paragrafo 3.1 è stato specificato come ruotare un vettore nello spazio ricorrendo a rotazioni elementari. In alcune occasioni, però, occorre ruotare oggetti intorno ad un asse di rotazione che non sia uno degli assi cartesiani. Dunque, si deve affrontare il problema di come poter ruotare, ad esempio, un vettore, intorno ad un asse di rotazione arbitrario. Prima di passare a descrivere la procedura (paragrafi 3.2.2 – 3.2.3), è bene chiarire alcuni concetti di algebra lineare.

3.2.1 Matrici Ortogonali

Una matrice quadrata \mathbf{A} con elementi reali si dice ortogonale se

$$\mathbf{A} \mathbf{A}^T = \mathbf{A}^T \mathbf{A} = \mathbf{I} \quad (3.2.1.1)$$

dove \mathbf{A}^T indica la trasposta della matrice \mathbf{A} , mentre \mathbf{I} è la matrice identità [15].

Dalla definizione di matrice ortogonale, segue, per definizione di matrice inversa \mathbf{A}^{-1} della matrice \mathbf{A} , che

$$\mathbf{A} \mathbf{A}^T = \mathbf{A}^T \mathbf{A} = \mathbf{I} = \mathbf{A} \mathbf{A}^{-1} = \mathbf{A}^{-1} \mathbf{A} \quad (3.2.1.2)$$

Da questa catena di uguaglianze, dunque, si evince che la trasposta di una matrice ortogonale \mathbf{A} coincide con la sua inversa, e, quindi,

$$\mathbf{A}^{-1} = \mathbf{A}^T \quad (3.2.1.3)$$

Enunciamo altre due proprietà delle matrici ortogonali [15]; dunque, se \mathbf{A} è una matrice ortogonale, allora

- i) $\det(\mathbf{A}) = \pm 1$ (3.2.1.4)
dove $\det(\mathbf{A})$ indica il determinante della matrice \mathbf{A} .
- ii) \mathbf{A}^T è anch'essa una matrice ortogonale.
- iii) Data una matrice ortogonale \mathbf{B} , allora anche la matrice $\mathbf{A} \mathbf{B}$, ottenuta con il prodotto riga-per-colonna tra le matrici \mathbf{A} e \mathbf{B} , è una matrice ortogonale.

Per il modo in cui sono definite, un esempio di matrici ortogonali è dato dalle matrici di rotazione [11]. Tutte le matrici di rotazione, infatti, hanno determinante uguale a 1, mentre la loro inversa coincide con la loro trasposta. Pertanto, se \mathbf{R} è la matrice di rotazione in questione, segue che

$$\mathbf{R}^T = \mathbf{R}^{-1} \quad (3.2.1.5)$$

Le matrici ortogonali torneranno utili nel determinare la matrice di rotazione intorno ad un asse arbitrario, la cui procedura verrà descritta nei prossimi paragrafi.

Sia dato un vettore $\mathbf{v} \in \mathbb{R}^3$. Vogliamo calcolare il suo trasformato \mathbf{w} , ottenuto mediante rotazione di angolo θ intorno ad un vettore $\mathbf{l} \in \mathbb{R}^3$, scelto arbitrariamente.

Per calcolare il vettore \mathbf{w} , è necessario costruire una base ortonormale associata al vettore \mathbf{l} (paragrafo 3.2.2), ovvero il nuovo asse di rotazione, per poi determinare la matrice di rotazione che permette di ruotare \mathbf{v} intorno a \mathbf{l} , attraverso un cambiamento di base (paragrafo 3.2.3).

3.2.2 Costruzione di una base ortonormale associata all'asse di rotazione

Prima di passare alla costruzione di una base ortonormale associata all'asse di rotazione \mathbf{l} , delineiamo il concetto di base di uno spazio vettoriale.

Sia V uno spazio vettoriale. Sia $B \subset V$ un suo sottoinsieme. L'insieme B viene definito base dello spazio vettoriale V se è un insieme di vettori linearmente indipendenti che generano tutto lo spazio V . Inoltre, se V è uno spazio vettoriale e B una sua base che lo genera, ogni vettore $\mathbf{v} \in V$ viene definito mediante coordinate espresse rispetto alla base B [14].

Ad esempio, una base che genera lo spazio \mathbb{R}^3 è dato dai versori della base canonica $E = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, dove

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.2.2.1)$$

Mentre il vettore

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad (3.2.2.2)$$

è un esempio di vettore le cui coordinate sono espresse rispetto alla base E , in quanto

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix} = \mathbf{e}_1 + 2\mathbf{e}_2 + 3\mathbf{e}_3 \quad (3.2.2.3)$$

In questo caso si dice che il vettore \mathbf{v} è combinazione lineare dei versori che formano la base E .

Nel caso di rotazioni elementari (paragrafo 3.1), la rotazione avviene intorno ad uno degli assi cartesiani; ad ognuno di questi assi è associato un versore (paragrafo 2.2). Come detto nel capitolo 2, questi versori identificano il sistema di riferimento cartesiano (paragrafo 2.2), rispetto al quale è possibile definire una posizione nello spazio, attraverso delle coordinate.

Ora, possiamo estendere questo concetto con quello di coordinate di un vettore rispetto ad una base (3.2.2.3).

Per la (2.2.2), i versori \mathbf{i} , \mathbf{j} , \mathbf{k} sono proprio i vettori della base canonica E (3.2.2.1), dunque, tali versori costituiscono una base di \mathbb{R}^3 . Pertanto, ogni posizione individuata nel sistema di assi cartesiani può essere espressa mediante un vettore $\mathbf{v} \in \mathbb{R}^3$ con coordinate espresse rispetto la base canonica E .

In modo analogo, quando si calcolano le rotazioni intorno agli assi coordinati, i vettori trasformati ottenuti sono vettori le cui coordinate sono date rispetto alla base canonica E . Di conseguenza, le matrici $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$ rappresentano trasformazioni lineari rispetto alla base canonica E .

Riprendendo la richiesta del quesito posto all'inizio di questo paragrafo, come calcolare la matrice di rotazione di angolo θ intorno ad un asse arbitrario \mathbf{l} rispetto alla base canonica?

Innanzitutto, dato l'asse di rotazione \mathbf{l} , si deve costruire una base ortonormale associata ad \mathbf{l} . Una base ortonormale di vettori è una base costituita da vettori di norma unitaria tra loro ortogonali.

Si supponga che \mathbf{l} sia stato già normalizzato. Per costruire una base ortonormale associata a \mathbf{l} si devono trovare altri due vettori \mathbf{m} , $\mathbf{n} \in \mathbb{R}^3$ tali che

$$\mathbf{l} \perp \mathbf{m} \perp \mathbf{n} \quad (3.2.2.4)$$

Per determinare il vettore \mathbf{m} si può ricorrere al prodotto scalare, il quale va imposto nullo, in quanto i vettori \mathbf{l} e \mathbf{m} devono risultare tra loro ortogonali.

Supponiamo, dunque, che il vettore \mathbf{l} abbia componenti

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \quad (3.2.2.5)$$

ed il vettore \mathbf{m} abbia componenti

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (3.2.2.6)$$

Vogliamo determinare le componenti del vettore \mathbf{m} per cui sia verificata la condizione

$$\mathbf{l} \cdot \mathbf{m} = l_1 m_1 + l_2 m_2 + m_3 l_3 = 0 \quad (3.2.2.7)$$

Un'idea può essere quella di scrivere il vettore \mathbf{m} con le componenti di \mathbf{l} e poi modificarle affinché sia rispettata la (3.2.2.7); si può, ad esempio, porre uguale a 0 la coordinata di \mathbf{m} che risulta essere più piccola in valore assoluto e scambiare tra loro le altre due componenti rimanenti.

Se, ad esempio, la componente di \mathbf{m} più piccola in valore assoluto fosse $m_3 = l_3$, si pone $m_3 = 0$.

Ora, se $m_1 = l_1$ e $m_2 = l_2$, si scambiano i valori di m_1 e m_2 , dunque, si pone $m_1 = l_2$ e $m_2 = l_1$.

Affinché sia rispettata la condizione (3.2.2.7) si deve, però, cambiare il segno ad una coordinata di \mathbf{m} , scelta tra m_1 e m_2 . Supponiamo, dunque, di porre $m_1 = -l_2$.

Il vettore \mathbf{m} così costruito vale

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} -l_2 \\ l_1 \\ 0 \end{bmatrix} \quad (3.2.2.8)$$

nel rispetto della (3.2.2.7).

Determinato il vettore \mathbf{m} , per costruire la base ortonormale si deve definire \mathbf{n} . Il vettore \mathbf{n} deve essere ortogonale sia ad \mathbf{m} che a \mathbf{l} . Una soluzione a tal quesito è data dal prodotto vettoriale tra i vettori \mathbf{l} ed \mathbf{m} , il quale permette di determinare un vettore ad essi ortogonale.

Dunque,

$$\mathbf{n} = \mathbf{l} \times \mathbf{m} \quad (3.2.2.9)$$

A questo punto, avendo determinato i versori \mathbf{m} ed \mathbf{n} , possiamo costruire la base ortonormale associata al vettore \mathbf{l} .

Tale base sarà data da

$$B = \{\mathbf{l}, \mathbf{m}, \mathbf{n}\} \quad (3.2.2.10)$$

dove B è un'altra base di \mathbb{R}^3 .

3.2.3 Cambiamento di base

Una volta determinata la base ortonormale associata a \mathbf{l} (3.2.2.10), l'idea per effettuare rotazioni intorno ad assi arbitrari è quella di un cambiamento di base.

Si vuole determinare il vettore \mathbf{w} , dato dalla rotazione del vettore \mathbf{v} intorno all'asse di rotazione arbitrario \mathbf{l} , esprimere \mathbf{w} in coordinate rispetto alla base canonica E (3.2.2.1) ed esplicitare la matrice che descrive tale rotazione.

La procedura consiste nel passare dalla base canonica E alla nuova base B , in modo da far coincidere l'asse x rispetto ad E con l'asse \mathbf{l} rispetto a B . Il vettore \mathbf{v}' , ovvero il vettore \mathbf{v} espresso in coordinate rispetto a B , viene poi ruotato di angolo θ intorno ad \mathbf{l} ricorrendo alla (3.1.1), ottenendo \mathbf{w}' rispetto a B . Ottenuto il vettore trasformato, si vuole ritornare alla base canonica, ovvero scrivere il vettore \mathbf{w}' , con coordinate rispetto ad E , e non più rispetto alla base B . Dunque, vogliamo ottenere il vettore \mathbf{w} .

Per poter passare dalla base canonica alla base B occorre, innanzitutto, determinare una matrice del cambiamento di base \mathbf{M} . Poiché \mathbf{l} viene dato in coordinate rispetto alla base canonica, allora anche i versori della base B sono espressi in coordinate relative ad E . Dunque, possiamo scegliere come \mathbf{M} , la matrice costituita dai vettori colonna che formano la base B . Di conseguenza, \mathbf{M} sarà la matrice costituita dai vettori di B espressi rispetto ad E .

$$\mathbf{M} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \quad (3.2.3.1)$$

Poiché la matrice \mathbf{M} è costituita da una terna di vettori colonna tra loro ortonormali, questo implica che la matrice \mathbf{M} sarà una matrice ortogonale [15]. Dunque, per la (3.2.1.3) si avrà che

$$\mathbf{M}^{-1} = \mathbf{M}^T \quad (3.2.3.2)$$

Supponiamo che \mathbf{v}' sia il vettore \mathbf{v} espresso in coordinate rispetto alla base B . Poiché \mathbf{M} è la matrice dei vettori di B con coordinate scritte rispetto alla base canonica, la formula

$$\mathbf{v} = \mathbf{M} \mathbf{v}' \quad (3.2.3.3)$$

permette di esprimere il vettore \mathbf{v} rispetto alla base canonica. Di conseguenza, il passaggio inverso da coordinate rispetto ad E a coordinate rispetto alla base B , sarà dato dalla matrice inversa \mathbf{M}^{-1} , e dunque, la formula

$$\mathbf{v}' = \mathbf{M}^{-1} \mathbf{v} \quad (3.2.3.4)$$

permette di esprimere il vettore \mathbf{v} rispetto alla base B , ottenendo, quindi, il vettore \mathbf{v}' . La matrice del cambiamento di base \mathbf{M} , inoltre, permette di trasformare il vettore \mathbf{l} nell'asse x , il vettore \mathbf{m} nell'asse y ed il vettore \mathbf{n} nell'asse z [15]. Dunque, dato il vettore \mathbf{v} , ricaviamo il vettore \mathbf{v}' rispetto alla base ortonormale B con la (3.2.3.4). Con questa trasformazione, l'asse \mathbf{l} è diventato il nuovo asse x . Pertanto, si può ruotare di angolo θ il vettore \mathbf{v}' intorno ad \mathbf{l} , ottenendo il vettore trasformato \mathbf{w}' con coordinate rispetto a B , dato da

$$\mathbf{w}' = \mathbf{R}_l(\theta) \mathbf{v}' \quad (3.2.3.5)$$

dove $\mathbf{R}_l(\theta) = \mathbf{R}_X(\theta)$, con $\mathbf{R}_X(\theta)$ data dalla (3.1.2).

Ricorrendo alla formula (3.2.3.4) possiamo riscrivere la (3.2.3.5) come

$$\mathbf{w}' = \mathbf{R}_l(\theta) \mathbf{M}^{-1} \mathbf{v} \quad (3.2.3.6)$$

Come detto, il vettore \mathbf{w}' è dato rispetto alla base B . Vogliamo, invece, scrivere \mathbf{w}' con coordinate in relazione alla base canonica.

Pertanto, applicando la (3.2.3.3), si ottiene il vettore \mathbf{w} cercato, per cui

$$\mathbf{w} = \mathbf{M} \mathbf{w}' \quad (3.2.3.7)$$

Sostituendo la (3.2.3.6) nella (3.2.3.7) si ottiene che

$$\mathbf{w} = \mathbf{M} \mathbf{R}_l(\theta) \mathbf{M}^{-1} \mathbf{v} \quad (3.2.3.8)$$

Poiché \mathbf{M} è una matrice ortogonale, segue dalla (3.2.3.2) che

$$\mathbf{w} = \mathbf{M} \mathbf{R}_l(\theta) \mathbf{M}^T \mathbf{v} \quad (3.2.3.9)$$

Sia, ora, \mathbf{X} la matrice associata alla rotazione di angolo θ intorno all'asse l espressa rispetto alla base canonica E . Secondo la (3.5), questa trasformazione può essere scritta come

$$\mathbf{w} = \mathbf{X} \mathbf{v} \quad (3.2.3.10)$$

Sostituendo la (3.2.3.9) nella (3.2.3.10), si ottiene la seguente catena di uguaglianze

$$\mathbf{w} = \mathbf{X} \mathbf{v} = \mathbf{M} \mathbf{R}_l(\theta) \mathbf{M}^T \mathbf{v} \quad (3.2.3.11)$$

da cui segue, dunque, che

$$\mathbf{X} = \mathbf{M} \mathbf{R}_l(\theta) \mathbf{M}^T \quad (3.2.3.12)$$

è la matrice che permette di calcolare la rotazione di angolo θ intorno ad un asse arbitrario l [4].

La Figura 3.7 descritta in [15] riassume il procedimento appena mostrato.

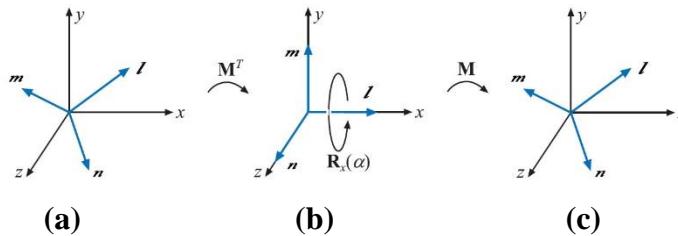


Figura 3.7. I passaggi per effettuare una rotazione intorno ad un asse arbitrario, secondo [15]. (a) La base ortonormale B costituita dai vettori $\{l, m, n\}$ associata al vettore l . (b) Attraverso la matrice \mathbf{M}^T , il cambiamento di base da E a B permette di trattare il vettore l come l'asse x del nuovo sistema di coordinate. Dunque, si può compiere facilmente la rotazione intorno a questo asse. (c) La matrice \mathbf{M} permette di passare dalla base B alla base E e di esprimere il risultato della rotazione rispetto a quest'ultima.

A questo punto, abbiamo tutte le informazioni necessarie per calcolare la matrice X della rotazione di angolo θ intorno ad un asse arbitrario \mathbf{l} , data dalla formula (3.2.3.12). Ci aspettiamo, dunque, che la matrice X sia equivalente a quella presentata da Goldman [15], secondo cui

$$\mathbf{R}_l(\theta) = \begin{bmatrix} \cos(\theta) + (1 - \cos(\theta))l_1^2 & (1 - \cos(\theta))l_1l_2 - l_3\sin(\theta) & (1 - \cos(\theta))l_1l_3 + l_2\sin(\theta) \\ (1 - \cos(\theta))l_1l_2 + l_3\sin(\theta) & \cos(\theta) + (1 - \cos(\theta))l_2^2 & (1 - \cos(\theta))l_2l_3 - l_1\sin(\theta) \\ (1 - \cos(\theta))l_1l_3 - l_2\sin(\theta) & (1 - \cos(\theta))l_2l_3 + l_1\sin(\theta) & \cos(\theta) + (1 - \cos(\theta))l_3^2 \end{bmatrix} \quad (3.2.3.13)$$

dove $\mathbf{R}_l(\theta)$ è la matrice che descrive la rotazione di angolo θ intorno ad un asse arbitrario \mathbf{l} . La matrice $\mathbf{R}_l(\theta)$ può essere riscritta nella forma proposta in [13]

$$\mathbf{R}_l(\theta) = (1 - \cos(\theta)) \mathbf{U}^T + \cos(\theta) \mathbf{I}_3 + \sin(\theta) \mathbf{L} \quad (3.2.3.14)$$

dove \mathbf{L} indica la matrice costituita dai vettori colonna

$$\mathbf{l}' = \mathbf{l} \times \mathbf{e}_1 \quad \mathbf{l}'' = \mathbf{l} \times \mathbf{e}_2 \quad \mathbf{l}''' = \mathbf{l} \times \mathbf{e}_3 \quad (3.2.3.15)$$

ed i vettori $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ sono i vettori della base canonica dati dalla (3.2.2.1).

3.3 Angoli di Eulero

Una rotazione nello spazio può essere rappresentata da tre rotazioni elementari intorno agli assi cartesiani (paragrafo 3.1), a cui possiamo associare una matrice R che descrive tale trasformazione. Possiamo, pertanto, scrivere R come composizione di tre rotazioni elementari di angoli distinti, per cui

$$R = R_{l_1}(\alpha) R_{l_2}(\beta) R_{l_3}(\gamma) \quad (3.3.1)$$

dove i vettori $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ vengono scelti tra gli assi cartesiani x, y, z , e la loro orientazione cambia dopo ogni rotazione.

Gli angoli α, β, γ sono detti angoli di Eulero e permettono di definire l'orientazione di un corpo nello spazio (Fig.3.8).

Dunque, presi gli assi cartesiani x, y, z come sistema di riferimento fisso, attraverso rotazioni elementari con angoli di Eulero si può descrivere la posizione di un oggetto e la sua orientazione, a cui è associato un sistema di riferimento locale solidale con l'oggetto stesso. Questi due sistemi di riferimento hanno la stessa origine ma assi che, in generale, sono orientati in modo differente. In questo caso, la matrice di rotazione, oltre ad indicare rotazioni in un sistema di riferimento, gestisce anche la trasformazione di coordinate da un sistema di riferimento a un altro. Infatti, immaginiamo di avere un oggetto nello spazio, il cui sistema di riferimento locale x_0, y_0, z_0 , solidale con esso, coincide, inizialmente, con il sistema degli assi x, y, z ; dunque, avremo che

$$x_0 = x \quad y_0 = y \quad z_0 = z \quad (3.3.2)$$

Supponiamo di ruotare l'oggetto intorno ad uno degli assi del sistema locale di angolo α , ad esempio, l'asse z_0 (Fig.3.9a). Il sistema locale x_0, y_0, z_0 , verrà trasformato nel sistema x_1, y_1, z_1 , dove

$$x_1 = \mathbf{R}_{z_0}(\alpha) x_0 \quad y_1 = \mathbf{R}_{z_0}(\alpha) y_0 \quad z_1 = \mathbf{R}_{z_0}(\alpha) z_0 \quad (3.3.3)$$

con $\mathbf{R}_{z_0}(\alpha)$ che indica la matrice di rotazione di angolo α intorno all'asse z_0 , mentre l'asse $z_1 = z_0 = z$ in quanto z_0 è stato scelto come asse di rotazione, dunque, rimane invariato, e per la (3.3.2) segue la seconda uguaglianza. La trasformazione (3.3.3), oltre a ruotare l'oggetto in questione, fa ruotare anche gli assi del sistema x_0, y_0, z_0 , mappando tali assi nel nuovo sistema x_1, y_1, z_1 , sempre solidale con l'oggetto dato.

Ruotiamo nuovamente l'oggetto, questa volta, però, intorno al nuovo asse x_1 (Fig.3.9b), compiendo una rotazione di angolo β . Si ottiene, dunque, il sistema x_2, y_2, z_2 , con

$$x_2 = \mathbf{R}_{x_1}(\beta) x_1 \quad y_2 = \mathbf{R}_{x_1}(\beta) y_1 \quad z_2 = \mathbf{R}_{x_1}(\beta) z_1 \quad (3.3.4)$$

dove $\mathbf{R}_{x_1}(\beta)$ è la matrice che descrive questa trasformazione e, analogamente al caso precedente, $x_2 = x_1$ in quanto x_1 è l'asse di questa rotazione. Anche in questo caso, gli assi del sistema x_1, y_1, z_1 sono stati ruotati, ottenendo il sistema x_2, y_2, z_2 , sempre solidale.

Infine, possiamo ruotare il nostro oggetto di angolo γ intorno al nuovo asse z_2 (Fig.3.9c). Anche in questo caso, il sistema x_2, y_2, z_2 verrà mappato nel sistema x_3, y_3, z_3 , con

$$x_3 = \mathbf{R}_{z_2}(\gamma) x_2 \quad y_3 = \mathbf{R}_{z_2}(\gamma) y_2 \quad z_3 = \mathbf{R}_{z_2}(\gamma) z_2 \quad (3.3.5)$$

dove $\mathbf{R}_{z_2}(\gamma)$ è la matrice che descrive tale rotazione, e $z_3 = z_2$ (Fig.3.9d).

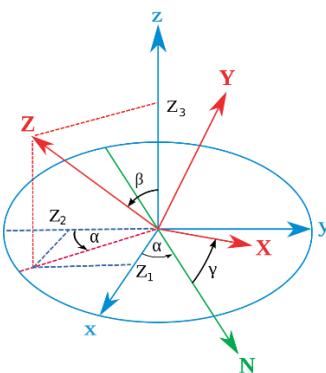


Figura 3.8. Gli angoli di Eulero.

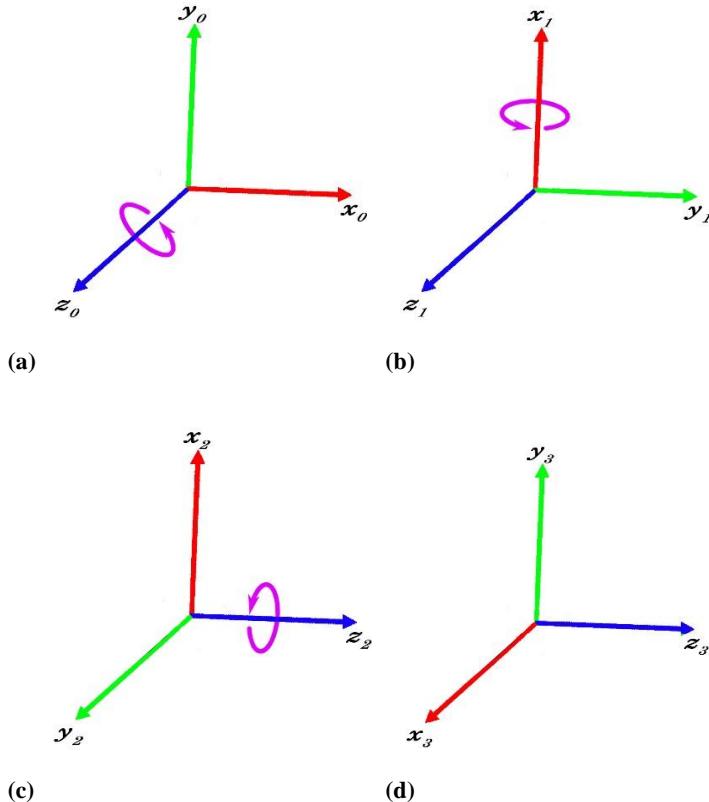


Figura 3.9. Il cambiamento della configurazione degli assi al termine di ogni rotazione. (a) Rotazione di 90° intorno all’asse z_0 . (b) Rotazione di 90° intorno all’asse x_1 . (c) Rotazione di 90° intorno all’asse z_2 . (d) La configurazione finale del sistema dato dagli assi x_3 , y_3 , z_3 [15].

Nelle formule (3.3.3), (3.3.4) e (3.3.5) abbiamo, quindi, descritto una rotazione nello spazio attraverso tre rotazioni elementari con tre differenti angoli. Dunque, possiamo associare a queste trasformazioni la matrice R data da

$$R = R_{z_0}(\alpha) R_{x_1}(\beta) R_{z_2}(\gamma) \quad (3.3.6)$$

E’ bene evidenziare che bisogna maneggiare le composizioni di rotazioni con molta attenzione, in quanto la composizione di isometrie, in generale non è commutativa. Le rotazioni, di solito, sono commutative solo quando vengono effettuate intorno allo stesso asse. Poiché nel caso degli angoli di Eulero si compongono rotazioni intorno ad assi diversi, tale composizione non sarà commutativa, pertanto, ogni composizione darà come risultato una rotazione diversa. Inoltre, è importante sottolineare che, per quanto riguarda le rotazioni con angoli di Eulero, queste devono essere effettuate tutte intorno ad assi diversi. Infatti, se due rotazioni successive sono relative allo stesso asse si perde un grado di libertà. Si devono, dunque, escludere questi casi dalle possibili sequenze di rotazioni. Pertanto, le sequenze possibili ammesse sono 12, le quali vengono divise in due gruppi:

1. Le sequenze proprie degli **angoli di Eulero**, che descrivono rotazioni intorno agli assi del sistema di riferimento locale solidale con l'oggetto, i quali cambiano la loro direzione al termine di ogni rotazione (come nel caso esaminato). In questo gruppo rientrano le sequenze:
 - i) $z - x - z$
 - ii) $x - y - x$
 - iii) $y - z - y$
 - iv) $z - y - z$
 - v) $x - z - x$
 - vi) $y - x - y$

2. Le sequenze relative agli **angoli di Tait-Bryan**, che descrivono rotazioni intorno agli assi x, y, z del sistema che viene preso come sistema di riferimento fisso (assi cartesiani). In questo caso, il sistema locale di un oggetto, solidale con esso, ruota fino a raggiungere un orientamento specifico. A questo gruppo appartengono le sequenze:
 - i) $x - y - z$
 - ii) $y - z - x$
 - iii) $z - x - y$
 - iv) $x - z - y$
 - v) $z - y - x$
 - vi) $y - x - y$

3.3.1 Angoli di Roll, Pitch e Yaw

Gli angoli di Eulero trovano il loro impiego nel settore aeronautico, permettendo di definire l'orientamento di un velivolo e la possibilità di poter tracciare il suo moto. Immagiamo, dunque, di poter costruire un sistema di assi XYZ con origine nel centro di massa del velivolo e che tale sistema sia solidale con esso. In questo caso, gli angoli di Eulero prendono il nome di Roll, Pitch e Yaw.

- **Roll:** è l'angolo α che descrive la rotazione del velivolo intorno al suo asse X. Tale rotazione permette all'aeromobile di virare a destra o a sinistra (Fig.3.10a).
- **Pitch:** è l'angolo β che descrive la rotazione del velivolo intorno al suo asse Y. Tale rotazione permette all'aeromobile di inclinarsi verso l'alto (ad esempio, in fase di decollo) o verso il basso (all'inizio della fase di atterraggio) (Fig.3.10b).
- **Yaw:** è l'angolo γ che descrive la rotazione del velivolo intorno al suo asse Z. Tale rotazione permette all'aeromobile di poter cambiare la sua rotta, puntando verso destra o verso sinistra (Fig.3.10c).

Come vedremo in seguito, faremo uso degli angoli di Eulero per orientare la camera virtuale di Google Earth Studio nella scena e per ricostruire tale orientamento in Maya.

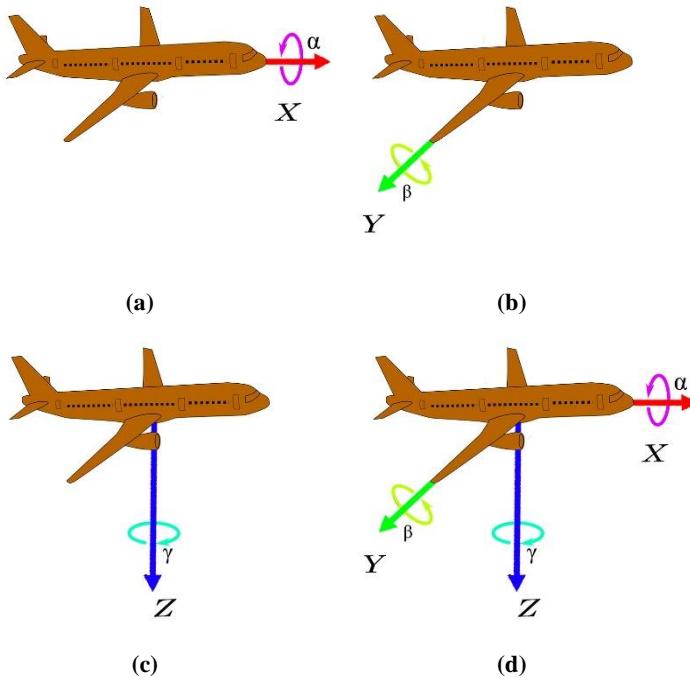


Figura 3.10. Gli angoli di Roll, Pitch e Yaw attraverso cui è possibile definire l'orientazione di un velivolo nello spazio. (a) Angolo di Roll. (b) Angolo di Pitch. (c) Angolo di Yaw. (d) Il sistema XYZ con origine nel centro di massa del velivolo e ad esso solidale.

Capitolo 4

Google Earth Studio

In questo capitolo, verrà descritta la procedura da seguire per creare riprese aeree con il software Google Earth Studio, delineando, inoltre, i vari formati in cui è possibile esportare il progetto. Nel secondo paragrafo invece, verrà data un'interpretazione dei dati forniti da Google Earth Studio e relativi all'animazione creata, contenuti nel file JSON.

Google Earth Studio (GES) è un software basato su browser che permette di realizzare riprese aeree ed animazioni, sfruttando modelli 3D ed immagini satellitari di Google Earth. L'aspetto essenziale che caratterizza la sua interfaccia, rende questo software molto facile da usare, ottenendo come risultato animazioni cinematografiche quasi fotorealistiche.

4.1 Creare un'animazione con Google Earth Studio

In questo paragrafo descriveremo, passo per passo, la procedura da seguire per creare un'animazione con Google Earth Studio. E' possibile accedere al software tramite il sito “ <https://www.google.com/intl/it/earth/studio/> ”. Una volta avviato, la schermata principale mostra alcune impostazioni del progetto da configurare, tra cui il nome del progetto, la risoluzione, il numero dei frame e il frame rate (Fig.4.1). Essendo un software basato su browser, è possibile interagire con l'interfaccia di GES attraverso una connessione ad Internet. Per questo motivo, al fine di ridurre i tempi in fase di rendering della scena creata, si è scelto di realizzare un'animazione costituita da 96 frame con un frame rate di 24fps; questo corrisponde a creare un video della durata di circa 4 secondi.

Creata il progetto, si potrà accedere all'interfaccia del software. La prima cosa da fare è ricercare una località in cui si vuole girare la clip, tramite l'apposita barra di ricerca. Ad, esempio, il nostro scopo è quello di realizzare un video in cui una mongolfiera atterra nello Stadio Olimpico a Roma. Dunque, ambienteremo la nostra clip a Roma, in prossimità del suddetto stadio.

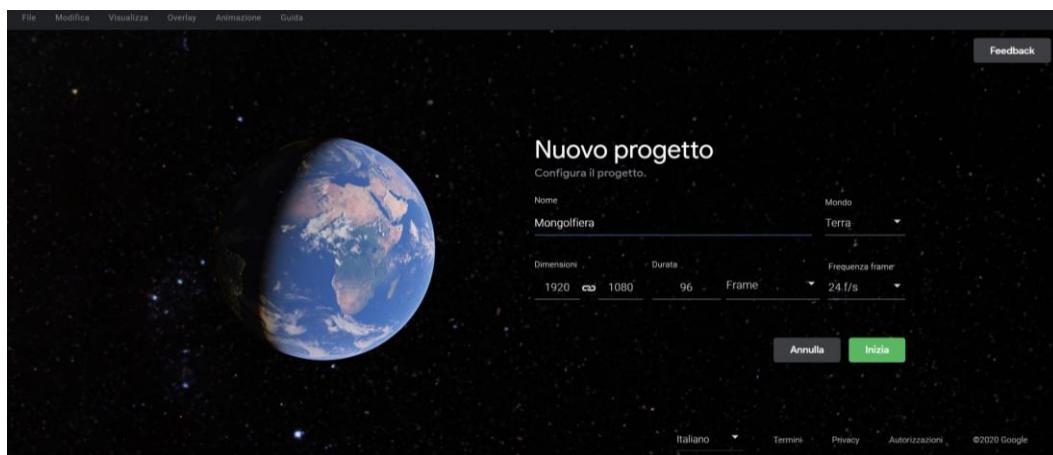


Figura 4.1. Schermata di avvio di Google Earth Studio, in cui è possibile scegliere le impostazioni del progetto da creare.

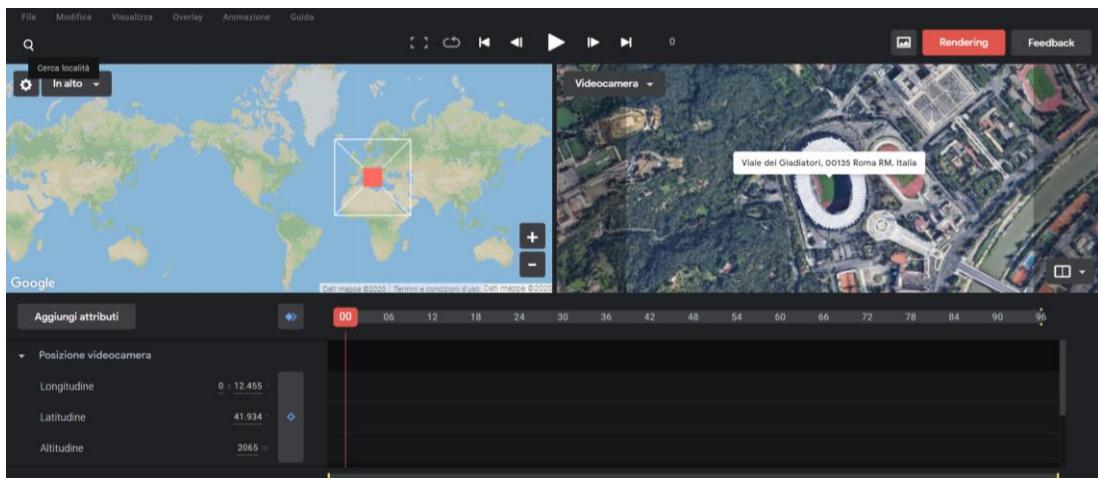


Figura 4.2. Interfaccia di Google Earth Studio. Come impostazione di default la camera riprende la scena in direzione perpendicolare al suolo.

Una volta scelta la località, GES mostra la scena impostando tutti i parametri dell’interfaccia con dei valori di default preimpostati (Fig.4.2).

Per poter creare un’animazione si dovrà scegliere un punto di interesse da riprendere e muovere la camera intorno ad esso. Facendo scorrere lo slider mostrato in Figura 4.3a lungo la timeline, si può impostare la posizione della camera per ogni frame desiderato. In GES, la posizione della camera viene espressa mediante le coordinate geografiche di latitudine, longitudine e altitudine. Cambiando il valore di questi parametri è possibile muovere la camera nella scena (Fig.4.3b). Una volta scelta la posizione della camera, occorre salvare queste informazioni in un fotogramma chiave in corrispondenza del frame corrente, cliccando sul tasto mostrato in Figura 4.3.c. Dunque, impostando vari fotogrammi chiave con diverse posizioni, si ottiene una prima animazione della camera. Ad esempio, una semplice animazione può essere realizzata modificando solo il parametro di longitudine; in questo modo, la camera si muoverà verso destra o verso sinistra. Altrimenti, si può optare per avvicinare la camera al suolo, modificando il valore dell’altitudine.

Se, invece, si vuole realizzare un’animazione più complessa e, dunque, più realistica, occorre orientare la camera nella scena, attraverso gli angoli di Eulero; in questo caso, questi prendono il nome di angoli di Pan (Panoramica), Tilt (Inclinazione) e Roll (Rollio). Se si considera il sistema di riferimento locale della camera (Fig.4.4a), dato dagli assi X, Y, Z,

- il Pan corrisponde a ruotare la camera intorno al suo asse Y,
- il Tilt corrisponde a ruotare la camera intorno al suo asse X,
- il Roll corrisponde a ruotare la camera intorno al suo asse Z.

GES impone questi angoli a 0° come valore di default. Questo significa avere una camera che guarda in direzione perpendicolare al suolo. Se si vuole orientare la camera in direzione parallela al terreno sottostante, ad esempio, occorrerà effettuare un Tilt di 90° (Fig.4.4b); per poter ruotare la camera verso destra o sinistra si dovrà compiere un Pan (Fig.4.4c); infine, per inclinare la camera si dovrà ricorrere ad un Roll (Fig.4.4d).



Figura 4.3. Alcuni dettagli dell’interfaccia di Google Earth Studio. (a) Spostando lo slider è possibile selezionare un frame. (b) Modificando i valori di latitudine, longitudine e altitudine è possibile spostare la camera, cambiando il punto di vista ripreso. (c) Aggiungere un fotogramma chiave consente di animare il movimento della camera.

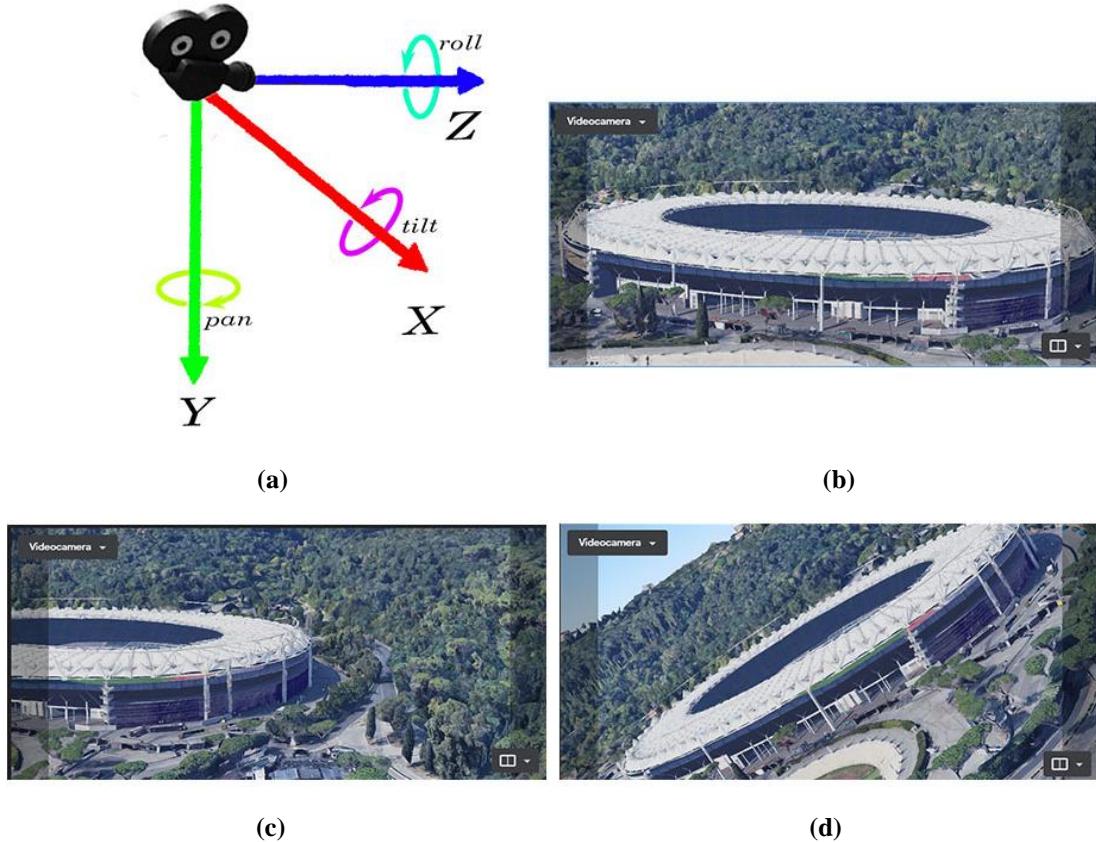


Figura 4.4. Orientare la camera nell’ambiente di Google Earth Studio. (a) Gli angoli di Pan, Tilt e Roll nel sistema di riferimento locale della camera. (b) Esempio di Tilt di 90°. (b) Un esempio di Pan verso destra. (c) Un esempio di Roll.

Oltre a creare animazioni, GES consente di rilevare la posizione degli oggetti nei dintorni attraverso uno strumento chiamato trackpoint (Fig.4.5b). Ad esempio, impostando un trackpoint al centro dello Stadio Olimpico, non solo è possibile conoscerne la posizione espressa in termini di latitudine, longitudine e altitudine, ma

aggiungendo altri trackpoint, possiamo avere un'idea delle proporzioni e delle distanze tra gli oggetti che sono presenti nella nostra scena.

Una volta creata l'animazione, GES permette di esportarla in due formati:

- **in formato JSON:** l'utente può ricavare da questo file le informazioni relative alla posizione della camera e al suo orientamento nella scena per ogni frame. In particolare, l'utente può scegliere se salvare questi dati rispetto ad un'origine globale (dunque, rispetto al centro della Terra), oppure rispetto ad un'origine locale (impostando un trackpoint su un punto della scena che si vuole prendere come origine locale);
- **in formato JSX:** il formato di questo file consente all'utente di poter importare l'animazione creata all'interno di una composizione di Adobe After Effects.

Nel nostro caso, esporteremo l'animazione in formato JSON, sfruttando i dati forniti da GES per poter calibrare la camera, ai fini del matchmoving.

Inoltre, renderizzando la scena, possiamo esportare e salvare in locale i fotogrammi che costituiscono la clip video.

Nel prossimo paragrafo, verrà data un'interpretazione dei dati contenuti nel file JSON generato da GES.

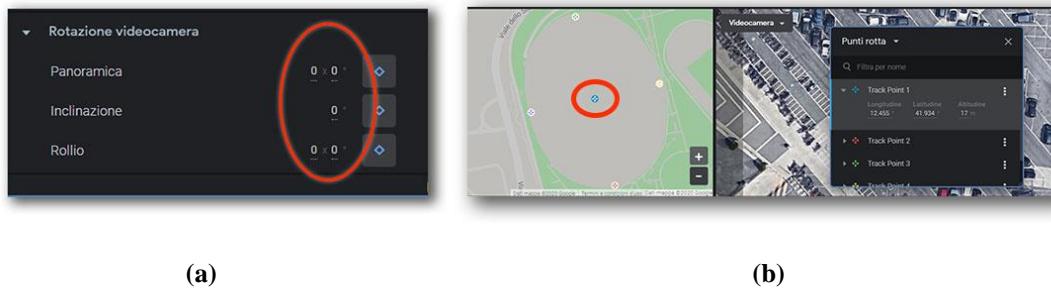


Figura 4.5. (a) Impostazioni per orientare la camera di Google Earth Studio. (b) Esempio di trackpoint.

4.2 Analisi dei dati

In questo paragrafo verrà delineata la struttura del file JSON, formato in cui si è scelto di esportare la clip video realizzata con GES, fornendo, inoltre, un'analisi dei dati in esso contenuti.

Si è scelto di salvare i dati in un file JSON in quanto questo tipo di formato ha una struttura facile da leggere e da interpretare per le persone e, allo stesso tempo, risulta essere facile da generare e da analizzare dalle macchine, essendo costituito da dati raccolti in un insieme di coppie nome/valore.

Il file JSON generato da GES è costituito da 7 sezioni:

- i) **nome del progetto** di GES;
- ii) **risoluzione**;
- iii) **frame rate**, ovvero la frequenza dei fotogrammi per secondo;
- iv) **numero dei frame** che compongono la clip;
- v) **durata della clip** in secondi;
- vi) **dati relativi alla camera per ogni frame**, in cui viene specificata la posizione (espressa in coordinate ECEF), orientazione, coordinate LLH e angolo di FOV verticale della camera per ogni frame che compone la clip;
- vii) **dati relativi ad eventuali trackpoints**, dove sono specificati il nome e le coordinate ECEF del punto in cui è stato posizionato un trackpoint all'interno progetto.

Analizziamo, ora, i dati relativi alla posizione e all'orientamento della camera.

4.2.1 Camera Position

Scorrendo il file JSON, tra i vari parametri in esso riportati troviamo la voce “camera position”; questo parametro indica la posizione in cui si trova la camera frame per frame. Tale posizione è espressa in coordinate cartesiane mediante una terna di valori (**paragrafo 2.2**). In particolare, secondo quanto riportato nella documentazione del software, GES fornisce i dati della camera in coordinate ECEF; questo significa che ogni posizione è data rispetto al centro della Terra, e gli assi x , y , z sono gli assi che costituiscono un sistema di riferimento solidale alla superficie terrestre, dove l'origine coincide con il centro di massa del pianeta (**paragrafo 2.2.1**).

Da questo momento in poi, indicheremo questa configurazione d'assi come il sistema degli assi terrestri (Fig. 4.6).

Per poter interpretare con precisione queste coordinate, dobbiamo capire quale modello GES associa alle coordinate ECEF per rappresentare la forma della Terra (**paragrafo 2.3**).

Per ottenere questa informazione, procediamo nel modo seguente. Sappiamo che il modello più usato per rappresentare la superficie terrestre è l'ellissoide descritto dal WGS84 (**paragrafo 2.3.2.1**); dunque, un'idea è quella di ricorrere al modello ellittico.

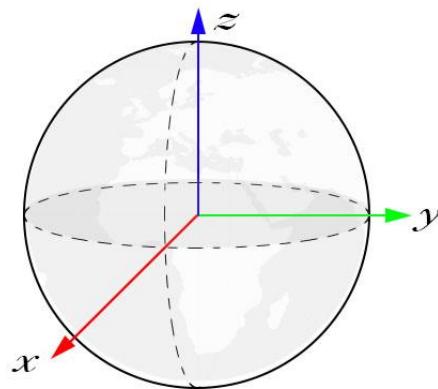


Figura 4.6. Il sistema degli assi terrestri per le coordinate ECEF.

Per verificare se GES si avvale di questo modello ricorriamo al Mapping Toolbox™ di Matlab®, un software per il calcolo numerico.

Supponiamo che al frame 0 la camera sia in posizione P, individuata dall' angolo di latitudine φ , l'angolo di longitudine θ e altezza h, dove φ indica la latitudine geodetica (paragrafo 2.1.1); supponiamo, inoltre, che GES, nel file JSON, ci fornisca la stessa posizione in coordinate ECEF: x, y, z.

Dall'equazione (2.3.2.2.1) fornita dal modello ellittico, è possibile convertire la posizione P espressa mediante coordinate LLH in coordinate ECEF, x_p , y_p , z_p .

Se, effettivamente, GES usa il modello ellittico per approssimare la forma della Terra, ci aspettiamo che le coordinate x_p , y_p , z_p , calcolate con la (2.3.2.2.1) coincidano con le coordinate x, y, z, fornite dal file JSON.

Supponiamo, ad esempio, che il punto P preso in considerazione, corrisponda alla posizione della camera sopra il Colosseo, a Roma. Le coordinate ECEF fornite dal file JSON sono

$$x = 4\ 630\ 471 \text{ m} \quad y = 1\ 025\ 892 \text{ m} \quad z = 4\ 253\ 967 \text{ m} \quad (4.2.1.1)$$

Sfruttando la funzione *geodetic2ecef*, fornita dal Mapping Toolbox™ di Matlab®, possiamo effettuare la conversione di coordinate descritta dalla (2.3.2.2.1). Tale funzione richiede come parametri di input, rispettivamente, l'ellissoide di riferimento associato al modello con cui si vuole approssimare la superficie terrestre (in questo caso l'ellissoide del WGS84), la latitudine geodetica, la longitudine e l'altitudine del luogo considerato. L'output ritornato da questa funzione è dato dalle coordinate

$$x_p = 4\ 642\ 584 \text{ m} \quad y_p = 1\ 028\ 575 \text{ m} \quad z_p = 4\ 236\ 543 \text{ m} \quad (4.2.1.2)$$

Come si può notare, le coordinate x_p , y_p , z_p non coincidono con quelle x, y, z.

Infatti, le coordinate ECEF calcolate secondo il modello ellittico presentano un errore dell'ordine di 10^4 sulle x e sulle z ed un errore dell'ordine di 10^3 sulle y.

Calcolando l'errore relativo percentuale sui dati ottenuti, si ottiene che le coordinate ECEF calcolate secondo il modello ellittico differiscono dello 0,3% da quelle fornite

da GES. Ciò significa che, lo sferoide oblatto del WGS84 differisce dello 0,3% rispetto alla forma della Terra approssimata secondo il modello adottato da GES. Questo ci suggerisce che GES potrebbe ricorrere al modello sferico per rappresentare la superficie terrestre (paragrafo 2.3.1). Possiamo verificare la veridicità di questa ipotesi sfruttando l'equazione (2.3.1.1.3).

Anche in questo caso, prendiamo come riferimento la posizione della camera in prossimità del Colosseo. Se le coordinate ECEF calcolate secondo il modello sferico coincidono con quelle fornite da GES, allora la forma della Terra sarà approssimata da una sfera di raggio medio.

Le coordinate x_p , y_p , z_p , vengono ancora una volta calcolate con la funzione *geodetic2ecef* in Matlab®, questa volta impostando un'approssimazione sferica della superficie terrestre. In questo caso, dunque, la latitudine geodetica φ coincide con quella geocentrica ϕ , quindi non è necessario ricorrere alla relazione (2.3.2.1.1) per esprimere ϕ in termini di φ . L'equazione (2.3.1.1.3) descrive matematicamente la conversione applicata dalla funzione *geodetic2ecef*, ottenendo

$$x_p = 4\ 630\ 472 \text{ m} \quad y_p = 1\ 025\ 892 \text{ m} \quad z_p = 4\ 253\ 968 \text{ m} \quad (4.2.1.3)$$

Come si può notare, le coordinate calcolate secondo il modello sferico coincidono con quelle fornite da GES, con un errore relativo alle unità o a cifre decimali; poiché stiamo prendendo in considerazione misure dell'ordine di 10^6 possiamo considerare questo errore trascurabile.

Dunque, possiamo concludere che le coordinate ECEF fornite da GES vengono calcolate approssimando la superficie terrestre con una superficie sferica, il cui raggio medio vale

$$R = 6\ 371,005076123 \text{ km} \quad (4.2.1.4)$$

4.2.2 Camera Rotation

Un'altra tipologia di dati che GES fornisce per ogni frame in relazione alla camera, riguarda le informazioni relative al suo orientamento, dato dal parametro “camera rotation” nel file JSON. In GES, l'orientamento della camera è descritto attraverso gli angoli di Eulero (paragrafo 3.3), in particolare dagli angoli di Roll, Pitch e Yaw (paragrafo 3.3.1).

Quando gli angoli di Roll, Pitch e Yaw vengono usati per orientare una camera nello spazio, questi prendono il nome, rispettivamente, di angoli di Roll, Tilt, Pan.

Come detto nel paragrafo 3.3, gli angoli di Eulero descrivono rotazioni che avvengono intorno agli assi del sistema di riferimento locale e solidale della camera; tale sistema è un sistema di assi mobili. Prima di passare all'interpretazione dei dati forniti da GES, bisogna fare alcune considerazioni sull'orientazione degli assi che costituiscono il sistema della camera. Indichiamo con XYZ tale configurazione.

Secondo la documentazione relativa alle API di Google Earth, gli assi della camera virtuale in GES sono orientati nel modo seguente (Fig.4.7a):

- l'asse X corrisponde al right vector, pertanto, è diretto nella direzione che si trova a destra della camera;
- l'asse Y corrisponde al up vector, dunque, è diretto verso l'alto;
- l'asse Z è uscente dal centro dello schermo, nella direzione ad esso perpendicolare. Il vettore dello sguardo della camera è diretto lungo il semiasse negativo delle Z.

Siano indicati con x, y, z gli assi cartesiani che descrivono la configurazione del sistema terrestre, descritti nei paragrafi **4.2.1** e **2.4.2**.

Poiché, nel file JSON, gli attributi relativi alla posizione della camera sono dati rispetto al centro della Terra, ci aspettiamo che gli angoli che descrivono l'orientamento della camera nel sistema ECEF siano, in qualche modo, legati agli assi del sistema terrestre.

Supponiamo di considerare un punto situato all'Equatore, di coordinate (0°N, 0°E); supponiamo, inoltre, che la camera stia guardando in direzione perpendicolare al suolo. Dunque, gli angoli di Pan, Tilt e Roll saranno tutti impostati a 0°. Esportando la scena descritta in formato JSON, i seguenti valori descrivono l'orientamento della camera:

$$x = 270^\circ \quad y = -90^\circ \quad z = 0^\circ \quad (4.2.2.1)$$

Questi angoli non coincidono affatto con quelli che sono stati impostati in GES, essendo questi tutti nulli. Questo risultato potrebbe suggerire che tali angoli non facciano riferimento al sistema di assi XYZ della camera ma al sistema xyz degli assi terrestri.

Siano

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.2.2.2)$$

i versori che indicano il modo in cui sono orientati gli assi x, y, z della Terra.

Nella scena considerata, la camera sta guardando in direzione perpendicolare al suolo all'Equatore, dunque, si ha che l'asse Z della camera è diretto come l'asse x della Terra (Fig.4.7b). Sia \mathbf{ze} il versore che indica il vettore dello sguardo della camera. Poiché tale vettore è diretto lungo il semiasse negativo delle Z, segue che, esprimendo \mathbf{ze} rispetto agli assi della Terra,

$$\mathbf{ze} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad (4.2.2.3)$$

Il right vector \mathbf{x}_c punterà a destra della camera, dunque sarà diretto come l'asse y della Terra, pertanto,

$$\mathbf{x}_c = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (4.2.2.4)$$

Per il prodotto vettoriale, segue, che

$$\mathbf{y}_c = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (4.2.2.5)$$

Per la scelta di \mathbf{z}_c , segue che \mathbf{y}_c è diretto lungo il semiasse negativo dell'asse Y della camera, coincidente con l'asse z della Terra. I versori \mathbf{x}_c , \mathbf{y}_c , \mathbf{z}_c , individuano il sistema XYZ ruotato di 180° intorno all'asse X (Fig.4.7c). Verifiche numeriche hanno dimostrato che tale rotazione è necessaria per riprodurre correttamente gli angoli che consentono di orientare la camera in fase di calibrazione (capitolo 5).

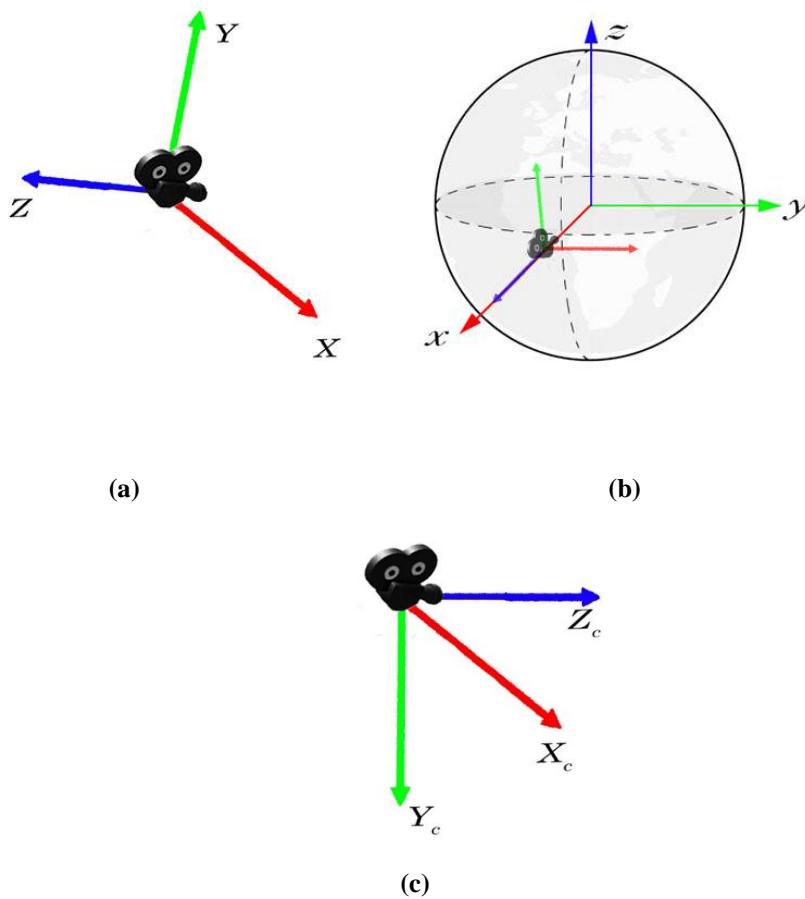


Figura 4.7. (a) Il sistema locale XYZ della camera, descritto secondo le API di Google Earth. (b) Il sistema della camera nel punto (0°N, 0°E). (c) Il sistema della camera ruotato di 180° rispetto ad X, dato dai versori \mathbf{x}_c , \mathbf{y}_c , \mathbf{z}_c .

L'idea è quella di mappare i versori x, y, z , in quelli $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ sfruttando gli angoli dati dalla (4.2.2.1). A tale scopo, ricorriamo al Computer Vision Toolbox™ fornito da Matlab®, che ci consentirà di calcolare le rotazioni intorno ad assi arbitrari.

Ruotando il sistema di versori \mathbf{xyz} intorno al versore \mathbf{x} di 270° , per la (3.6) si ottiene il sistema $\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1$, dato da

$$\mathbf{x}_1 = \mathbf{R}_x(270^\circ) \mathbf{x} \quad \mathbf{y}_1 = \mathbf{R}_x(270^\circ) \mathbf{y} \quad \mathbf{z}_1 = \mathbf{R}_x(270^\circ) \mathbf{z} \quad (4.2.2.6)$$

con

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{y}_1 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{z}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Ruotando il sistema ottenuto intorno al versore \mathbf{y}_1 di -90° , applicando la (3.6), si ottiene

$$\mathbf{x}_2 = \mathbf{R}_{y_1}(-90^\circ) \mathbf{x}_1 \quad \mathbf{y}_2 = \mathbf{R}_{y_1}(-90^\circ) \mathbf{y}_1 \quad \mathbf{z}_2 = \mathbf{R}_{y_1}(-90^\circ) \mathbf{z}_1 \quad (4.2.2.7)$$

con

$$\mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{y}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{z}_2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Ruotando, nuovamente, il sistema ottenuto intorno al versore \mathbf{z}_2 di 0° , sempre per la (3.6), si ottiene

$$\mathbf{x}_3 = \mathbf{R}_{z_2}(0^\circ) \mathbf{x}_2 \quad \mathbf{y}_3 = \mathbf{R}_{z_2}(0^\circ) \mathbf{y}_2 \quad \mathbf{z}_3 = \mathbf{R}_{z_2}(0^\circ) \mathbf{z}_2 \quad (4.2.2.8)$$

con

$$\mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{y}_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{z}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Il sistema di versori ottenuto nella (4.2.2.8) coincide con il sistema dato dai versori $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ in (4.2.2.3), (4.2.2.4), (4.2.2.5). Ruotando i versori $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ di 180° intorno a \mathbf{x}_c , si ottiene il sistema XYZ degli assi della camera, secondo

$$\mathbf{X} = \mathbf{R}_{xc}(180^\circ) \mathbf{x}_c \quad \mathbf{Y} = \mathbf{R}_{xc}(180^\circ) \mathbf{y}_c \quad \mathbf{Z} = \mathbf{R}_{xc}(180^\circ) \mathbf{z}_c \quad (4.2.2.9)$$

con

$$\mathbf{X} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Pertanto, possiamo concludere che gli angoli forniti da GES nel file JSON dati dalla (4.2.2.1) corrispondono agli angoli delle rotazioni necessarie a mappare il sistema degli assi terrestri xyz nel sistema degli assi della camera XYZ. Attraverso esperimenti numerici, si verifica che questa proprietà vale per orientamenti arbitrari della camera.

Capitolo 5

GES per matchmoving non basato su tracking

Calibrare una camera, significa ricavare le informazioni sulla sua posizione, sul modo in cui questa è orientata nello spazio, e su alcune sue proprietà intrinseche, come, ad esempio, lunghezza focale, dimensioni sensore, eventuale distorsione dovuta alle lenti usate.

Come specificato nel capitolo 1, di solito, la calibrazione della camera avviene ricorrendo a tecniche di tracking, sfruttando l'algoritmo di SfM, a partire dalla clip video in cui si vogliono inserire elementi di CG. In questa tesi, vogliamo proporre un metodo che permetta di calibrare la camera senza dover far uso di tecniche di tracking, consentendo, allo stesso tempo, di poter sfruttare tale calibrazione ai fini del matchmoving.

In questo capitolo, verranno descritte in dettaglio le cinque fasi in cui si suddivide la tecnica proposta per matchmoving non basato su tracking (Fig.5.1):

- esportazione della scena creata in GES in formato clip video ed in formato JSON;
- calibrazione della camera;
- animazione della camera virtuale;
- ricostruzione della scena in Maya;
- matchamation.

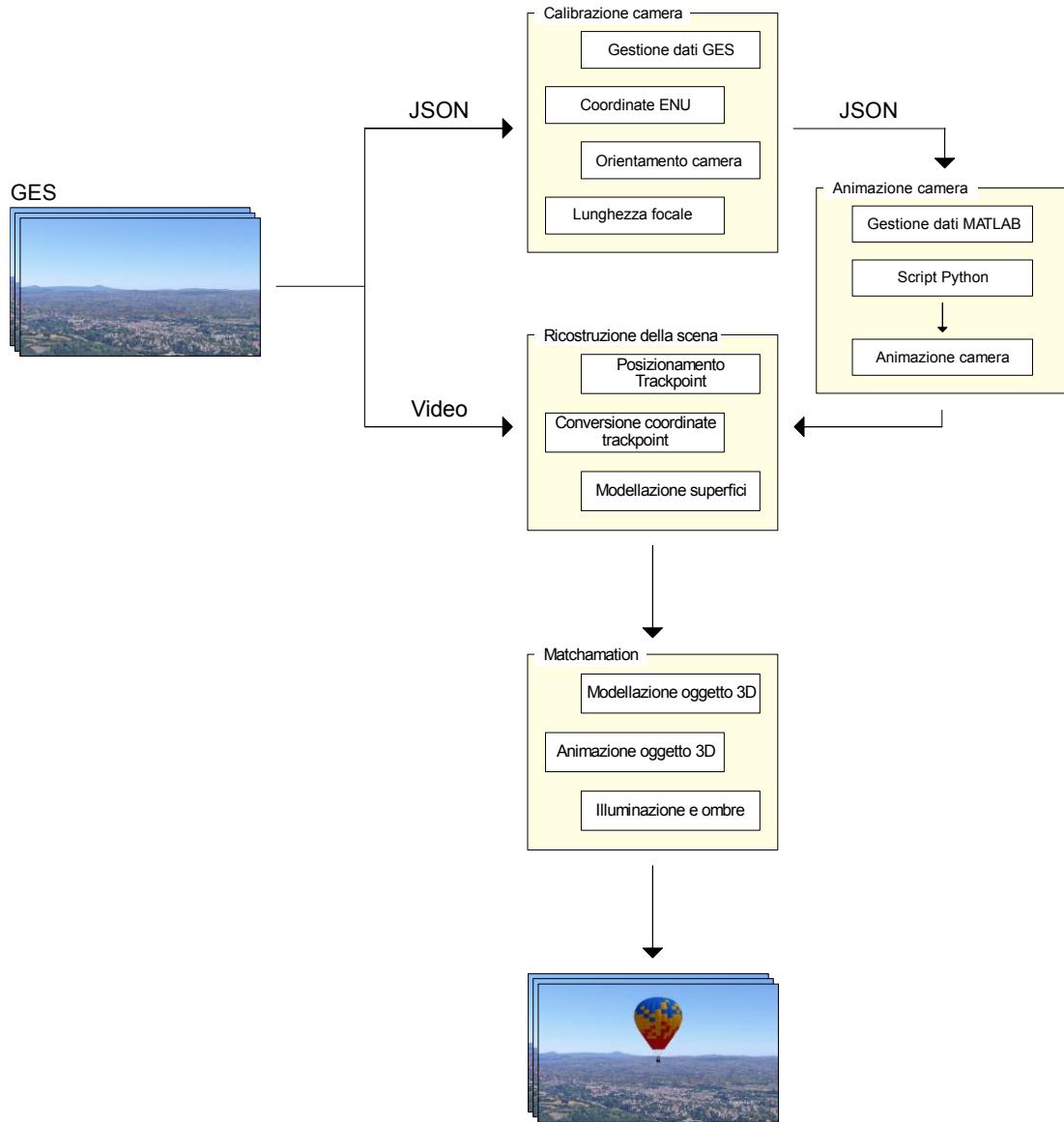


Figura 5.1. Workflow della tecnica proposta in questa tesi per matchmoving non basato su tracking.

5.1 Esportare i dati da Google Earth Studio

L'obbiettivo, dunque, è ottenere una sequenza video in cui inserire un modello 3D animato, le cui posizioni, prospettiva e proporzioni siano corrette e conformi con quelle della scena, come mostrato in Figura 5.2.

Per prima cosa, quindi, è necessario avere una clip video da cui partire. GES offre la possibilità di realizzare una ripresa aerea su una vasta gamma di territori, sfruttando le immagini satellitari di Google Earth, modelli 3D di alcune città e dati geologici. Sotto queste condizioni, dunque, è possibile simulare riprese aeree girate dai droni, senza dover ricorrere a questo tipo di tecnologia (Fig.5.3). Poder sfruttare questa tecnica, infatti, offre molti vantaggi:

- non si devono fronteggiare spese economiche per l'acquisto/noleggio di droni;
- non è necessario, di conseguenza, avere attestati per piloti APR, requisito, invece, obbligatorio per coloro che fanno uso di droni aventi peso superiore ai 250 grammi;
- non occorrono particolari permessi per poter sorvolare alcune zone;
- cosa più importante, GES consente di rilevare dati geologici relativi all'area che si sta riprendendo.

Una volta deciso il luogo da sorvolare e da riprendere, non resta che realizzare la clip video con GES, seguendo il procedimento descritto nel paragrafo 4.1. Come verrà descritto in seguito, il metodo proposto in questa tesi non prevede ricorso alle onerose procedure di tracking, in quanto la camera viene calibrata sfruttando i dati geografici forniti da GES. Ai fini del matchmoving, però, per poter utilizzare questi dati in Maya, occorre convertire le coordinate ECEF in coordinate ENU. Come spiegato nel paragrafo 2.2.2.2, per definire un sistema di coordinate ENU, occorre determinare un piano tangente locale alla superficie terrestre in un punto, preso come origine locale; dunque, tale piano tangente locale deve essere definito in prossimità del luogo geografico in cui vogliamo girare la nostra clip con GES.

Se, ad esempio, si vogliono fare delle riprese nei dintorni dello Stadio Olimpico, per determinare il sistema delle coordinate ENU, si può considerare il punto centrale dello Stadio come origine locale degli assi.



Figura 5.2. Esempi di matchmoving ottenuti applicando la tecnica proposta in questa tesi. (a) Il modello 3D di un'astronave viene inserito in una clip che riprende le piste dell'aeroporto di Fiumicino. (b) Il modello 3D di una mongolfiera viene inserito in una clip che riprende una veduta aerea nei pressi dello Stadio Olimpico a Roma. In entrambi i casi, le clip video sono state realizzate con Google Earth Studio.

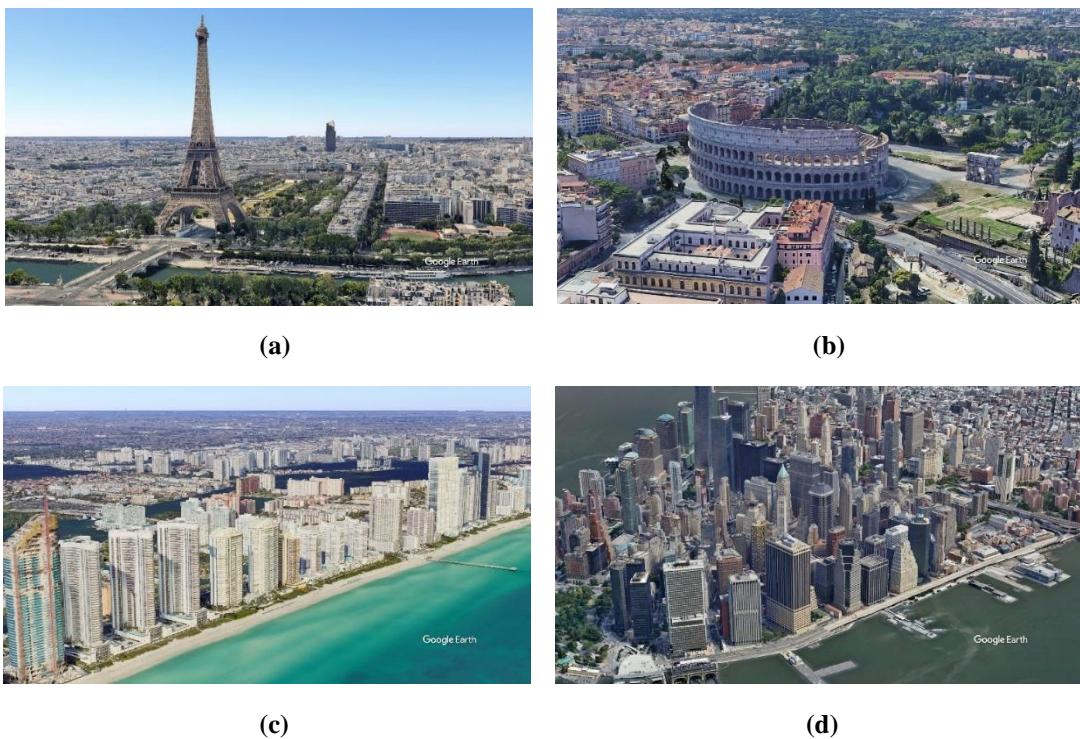


Figura 5.3. Alcune riprese aree realizzate con Google Earth Studio. (a) Una veduta di Parigi. (b) Una veduta di Roma. (c) Una veduta di Miami. (d) Una veduta di New York.

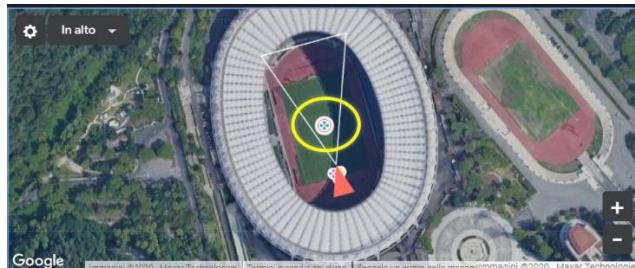


Figura 5.4. Cerchiato in giallo, il trackpoint posto al centro dello Stadio Olimpico in Google Earth Studio. Servirà per determinare l'origine degli assi del sistema ENU.

Per poter convertire le coordinate ECEF date da GES in coordinate locali ENU da usare in Maya, abbiamo bisogno di alcune informazioni geografiche relative al punto scelto come origine locale; in particolare devono essere note la sua latitudine e la sua longitudine. Fortunatamente, è possibile ottenere queste informazioni grazie ai trackpoint; pertanto, impostando un trackpoint nel centro dello Stadio Olimpico, siamo in grado di conoscere la sua posizione. Useremo questo dato importante in seguito per determinare l'origine degli assi ENU (Fig.5.4).

Scelto il punto che si vuole prendere come origine locale, una volta creata la scena in GES, un primo passo da fare è quello di esportare i fotogrammi che costituiscono la sequenza video. Per ridurre i tempi di rendering in GES, verranno create clip da 96 frame con un frame rate di 24fps, ovvero clip video della durata di circa 4 secondi.

Esportati i frame, il secondo passo da fare è quello di esportare le informazioni della scena in un file JSON. Per semplicità di calcoli futuri, sceglieremo di salvare i dati in modalità origine globale. In questo modo, otterremo le posizioni della camera per ogni frame espresse rispetto al centro della Terra.

Terminate queste procedure, si passa a processare i dati per poter ricavare le informazioni necessarie a calibrare la camera.

5.2 Calibrare la camera

La tecnica proposta in questa tesi per calibrare la camera prevede quattro fasi:

- gestione dei dati di GES contenuti nel file JSON;
- calcolo delle coordinate ENU per la posizione della camera;
- orientamento della camera nel sistema ENU;
- stima della lunghezza focale.

5.2.1 Gestione dei dati in input

Nonostante sia stata creata una clip video con soli 96 frame, le informazioni contenute nel file JSON sono tante. Pertanto, occorre una buona strategia per gestire e organizzare la grande quantità di dati prima di passare a processarli. Inoltre, abbiamo bisogno di software che supportino il formato JSON. In base alle esigenze sopra descritte e al fine di velocizzare i tempi di calcolo, l'intero processo di calibrazione della camera verrà svolto in Matlab[®], sfruttando le funzioni contenute nei toolbox aggiuntivi Mapping ToolboxTM e Computer Vision ToolboxTM.

A seguito dell'analisi dati svolta nei paragrafi **4.2.1** e **4.2.2**, possiamo, dunque, importare il file JSON in Matlab[®], usando funzioni apposite previste dall'ambiente. Matlab[®] organizza le varie coppie nome/valore del file JSON in un insieme di strutture annidate, in cui ogni struttura corrisponde ad una coppia specifica.

Una volta importato il file JSON in Matlab[®], la prima cosa da fare, è estrapolare le informazioni relative al numero di frame che compongono la clip e ai trackpoint. In particolare, occorre ricavare il trackpoint corrispondente al punto che si vuole prendere come origine locale degli assi del sistema ENU. GES consente di assegnare una label ad ogni trackpoint che viene aggiunto nel progetto. Dunque, possiamo assegnare al trackpoint in questione il nome “O”; in tal modo, risulterà più facile individuare la struttura relativa al trackpoint richiesto. Fatto questo, occorre trovare una soluzione per gestire i dati forniti da GES.

L'idea è quella di avere una sorta di tabella virtuale da cui è possibile leggere le informazioni della camera per ogni frame. Poiché Matlab[®] consente di manipolare matrici, una buona politica è quella di organizzare i dati contenuti nel file JSON in questo tipo di record.

Se fr indica il numero di frame che compongono la clip, si può pensare di costruire una matrice F in cui salvare i dati del file, avente dimensioni $nf \times 9$, dove le righe corrispondono ai frame e le colonne corrispondono ai dati relativi alla camera per ogni frame, con

$$nf = fr + 1 \quad (5.2.1.1)$$

Il valore di nf segue dal modo in cui GES conta il numero di frame che compongono la clip. Se, ad esempio, si realizza una sequenza video costituita da 2 frame, nel file JSON risulterà che il numero di frame che compongono la clip è 2; nel progetto di GES, invece, verranno creati tre frame: il frame 0, il frame 1 ed il frame 2; di conseguenza, verranno calcolate tre posizioni consecutive della camera, invece di due. Pertanto, il numero effettivo di frame che vengono creati è 3 e non 2. Generalizzando questo concetto, se fr è il numero di frame che compongono la clip, GES aggiungerà sempre il frame 0, per un totale di $fr + 1$ frame. Ciò giustifica la scelta del valore di nf descritto nella (5.2.1.1) e della dimensione della matrice F .

Stabilita la struttura di F , possiamo pensare, quindi, di suddividere i dati della camera per ogni frame in quattro categorie, dove ogni categoria fa riferimento ad una particolare struttura contenuta nel file JSON:

- **Camera Position:** dalla colonna 1 alla colonna 3, corrisponde alla struttura in cui sono contenute le informazioni relative alla posizione della camera espresse in coordinate ECEF, per un dato frame;
- **Camera Rotation:** dalla colonna 4 alla colonna 6, corrisponde alla struttura in cui sono contenute le informazioni relative all'orientamento della camera per un dato frame, ovvero gli angoli di rotazione necessari per allineare il sistema degli assi terrestri con quello locale della camera;
- **Coordinate LLH:** dalla colonna 7 alla colonna 8, corrisponde alla struttura in cui sono contenuti gli angoli di latitudine e longitudine che individuano le posizioni della camera, per un dato frame; ci serviranno queste informazioni per poter ricavare, in seguito, gli angoli di Pan, Tilt e Roll della camera a partire dai dati forniti da GES;
- **FOV:** colonna 9, corrisponde alla struttura in cui è contenuto il valore dell'angolo di FOV Verticale della camera, per un dato frame.

La Tabella 5.1 riassume la struttura della matrice F in cui sono salvati i dati per i primi quattro frame di animazione.

Frame	Camera Position			Camera Rotation			Coordinate LLH		FOV
	x	y	z	x	y	z	lat	lon	
0	4.641402	1.006845	4.246578	93°	10°	131°	41.8°	12.2°	20°
1	4.641398	1.006853	4.246580	93°	10°	131°	41.8°	12.2°	20°
2	4.641394	1.006860	4.246582	93°	11°	131°	41.8°	12.2°	20°
3	4.641391	1.006867	4.246585	94°	12°	130°	41.8°	12.2°	20°

Tabella 5.1. Struttura della matrice F in cui vengono salvati i dati della camera per i primi quattro frame di animazione. Le righe corrispondono ai frame, le colonne ai dati della camera relativi per ogni frame, divisi in quattro categorie. Le misure riportate nelle colonne relative alla “Camera Position” sono dell’ordine di 10^6 m.

Organizzati i dati, si può passare a processarli. Come spiegato all'inizio del capitolo, per evitare di ricorrere alle procedure di tracking, il processo di calibrazione della camera avviene considerando i dati forniti da GES. Inoltre, ai fini del matchmoving, occorre trasformare questi dati in modo opportuno, per poterli importare nel software Maya.

5.2.2 Calcolo delle coordinate ENU per la posizione della camera

Il primo passo da fare ai fini della calibrazione è ricavare le posizioni della camera.

GES fornisce le posizioni della camera frame per frame, esprimendole in coordinate ECEF, ovvero rispetto al centro della Terra. Per poter importare e sfruttare questi dati in Maya, occorre, però, trasformare questo tipo di coordinate in coordinate locali, compatibili con il sistema di riferimento adottato dal software. Un tipo di coordinate che soddisfa le condizioni richieste è dato dalle coordinate ENU. Pertanto, il primo passo da fare sarà quello di convertire le coordinate della camera da ECEF in ENU.

Come detto all'inizio del capitolo, per determinare un sistema di coordinate ENU, è necessario conoscere la posizione del punto che si vuole prendere come origine locale degli assi. A tal scopo, abbiamo impostato nel progetto GES un trackpoint “O” per localizzare il suddetto punto. Nel file JSON, la posizione del trackpoint, come quella della camera, viene espressa in coordinate ECEF. Per poter ricavare le coordinate ENU, però, occorre conoscere gli angoli di latitudine e longitudine che identificano il punto corrispondente ad “O”. A tal fine, ricorriamo alla funzione *ecef2geodetic* del Mapping Toolbox™ fornito da Matlab®. Tale funzione, richiede in input le coordinate ECEF x, y, z del trackpoint ed il modello con cui approssimare la superficie terrestre. Poiché GES ricorre ad un modello sferico per rappresentare matematicamente la forma della Terra, possiamo costruire tale modello con la funzione *referenceSphere* prevista dal Mapping Toolbox™.

In questo modo, dunque, si ottengono l'angolo di latitudine ϕ , l'angolo di longitudine θ , e il valore dell'altitudine h per il punto “O”. Questi parametri verranno indicati con ϕ_0 , θ_0 , h_0 , in quanto fanno riferimento all'origine degli assi ENU.

Prima di convertire le coordinate della camera da ECEF in ENU, è necessario definire gli assi di questo nuovo sistema di riferimento.

Sia **O** il punto scelto come origine del sistema ENU, le cui coordinate sono date ϕ_0 , θ_0 , h_0 e sia indicato con **C** il centro della Terra, origine del sistema ECEF. È possibile costruire gli assi del sistema ENU a partire dal sistema degli assi terrestri attuando una serie di rotazioni (capitolo 2). Per far questo, è necessario avere una configurazione di assi ENU iniziale, definiti rispetto agli assi terrestri, quindi, rispetto a **C**. Sia tale configurazione data dagli assi **E₀**, **N₀**, **U₀** (Fig.5.5a), la cui origine **O₀** coincide con il punto di intersezione tra l'Equatore ed il Meridiano di Greenwich, come suggerito in [13], con

$$\mathbf{E}_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{N}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{U}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.2.2.1)$$

Allora, gli assi **e**, **n**, **u** che costituiscono il sistema ENU, con origine in **O**, sono dati da

$$\mathbf{e} = \mathbf{R}_{N_0}(\theta_O) \mathbf{E}_0 \quad (5.2.2.2)$$

dove \mathbf{R}_{N_0} indica la matrice che descrive la rotazione di angolo θ_O intorno all'asse **N₀**, [13] come mostrato in Figura 5.5b;

$$\mathbf{n} = \mathbf{R}_{-e}(\phi_O) \mathbf{N}_0 \quad (5.2.2.3)$$

dove \mathbf{R}_{-e} indica la matrice che descrive la rotazione di angolo ϕ_O intorno all'asse -**e** (Fig.5.5c). Il motivo per cui la rotazione avviene intorno all'asse -**e** segue dalla regola della mano destra; lo stesso risultato si sarebbe ottenuto ruotando il vettore **N₀** di angolo $-\phi_O$ intorno all'asse **e** [13].

Infine, il vettore **u** è dato da

$$\mathbf{u} = \mathbf{e} \times \mathbf{n} \quad (5.2.2.4)$$

Se, invece, fosse stato calcolato il prodotto vettoriale

$$\mathbf{d} = \mathbf{n} \times \mathbf{e} \quad (5.2.2.5)$$

si sarebbero ottenuti gli assi del sistema NED (paragrafo 2.2.2.1), come conseguenza del fatto che il prodotto vettoriale non è commutativo [13].

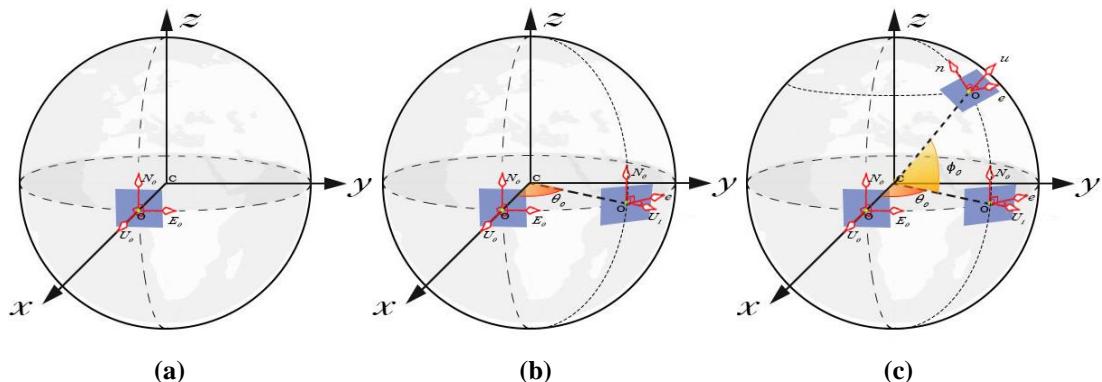


Figura 5.5. Costruzione degli assi ENU a partire dal sistema ECEF. (a) Configurazione iniziale degli assi E_0 , N_0 , U_0 . (b) L'asse **e** ottenuto attraverso la rotazione di angolo θ_O intorno all'asse Z_0 . (c) L'asse **n** ottenuto attraverso la rotazione di angolo ϕ_O intorno all'asse -**e**. Gli assi **e**, **n**, **u** formano il sistema di coordinate ENU con origine locale in **O**.

Dunque, attraverso la (5.2.2.2), (5.2.2.3) e (5.2.2.4) abbiamo costruito il sistema degli assi ENU con origine locale in **O**, a partire dal sistema degli assi terrestri. Avendo definito un piano locale tangente in **O** alla superficie terrestre, le distanze che andremo a determinare in tale sistema di riferimento saranno dell'ordine di circa 10^2 m.

Il sistema ENU creato, costituito da gli assi **e**, **n**, **u**, è un sistema di coordinate cartesiane; per questo motivo, in generale, è consuetudine associare l'asse *x* all'asse **e**, l'asse *y* all'asse **n** e l'asse *z* all'asse **u**. Prima di procedere a calcolare le coordinate della camera, è necessario apportare alcune modifiche al sistema creato, al fine di assicurarne la compatibilità con quello adottato da Maya. Il software, infatti, presenta un sistema di coordinate cartesiane in cui:

- l'asse *x* è diretto verso destra;
- l'asse *y* è diretto verso l'alto;
- l'asse *z* è uscente allo schermo in direzione ad esso perpendicolare.

Nel sistema ENU appena creato, il vettore di “Up” **u** è associato all'asse *z* e non all'asse *y*. Vogliamo, dunque, apportare una piccola modifica agli assi **e**, **n**, **u** affinché nel sistema di coordinate locali che verrà importato in Maya, l'asse **u** di “Up” corrisponda all'asse *y*. Un'idea potrebbe essere quella scambiare tra loro gli assi **n** ed **u**, dunque, creando un sistema costituito dagli assi **E**, **N**, **U**, dove

$$\mathbf{E} = \mathbf{e} \quad \mathbf{N} = \mathbf{u} \quad \mathbf{U} = -\mathbf{n} \quad (5.2.2.6)$$

Da prove sperimentali, per il modo in cui sono orientati gli assi in Maya, segue la necessità di cambiare il segno all'asse **n**; ciò garantisce una completa compatibilità tra il sistema ENU definito e quello adottato da Maya. Prove sperimentali dimostrano che le modifiche apportate consentono di riprodurre fedelmente il movimento della camera di GES in Maya.

Definiti gli assi del nuovo sistema di riferimento, passiamo, ora, a calcolare le posizioni della camera espresse in coordinate locali ENU. Il sistema di coordinate locali definito dalla (5.2.2.6) consente di calcolare delle posizioni rispetto all'origine locale **O**. Se **P** indica la posizione della camera per un dato frame espressa in coordinate ECEF, dunque, rispetto al centro della Terra **C**, allora il vettore **P'** dato da

$$\mathbf{P}' = \mathbf{P} - \mathbf{O} \quad (5.2.2.7)$$

farà riferimento alla posizione della camera rispetto all'origine locale **O**, per un dato frame. Il vettore **P'**, però, è espresso sempre in coordinate ECEF ma questa volta, tali coordinate, invece di essere globali, dunque riferite al centro della Terra, saranno coordinate ECEF locali, quindi riferite al punto **O**. Ciò, dunque, corrisponde a traslare l'origine **C** degli assi terrestri nel punto **O**. Il sistema degli assi terrestri traslato in **O**, però, non coincide con il sistema degli assi locali ENU definito dalla (5.2.2.6).

Ora, dato il vettore **P'**, per ottenere la posizione della camera espressa in coordinate ENU, basta proiettare il vettore **P'** lungo gli assi **E**, **N**, **U** del sistema di coordinate locali definito in precedenza dalla (5.2.2.6). Dunque, traducendo queste proiezioni in termini di prodotto scalare si ha che

$$\mathbf{P}_E = \langle \mathbf{P}', \mathbf{E} \rangle \quad \mathbf{P}_N = \langle \mathbf{P}', \mathbf{N} \rangle \quad \mathbf{P}_U = \langle \mathbf{P}', \mathbf{U} \rangle \quad (5.2.2.8)$$

Le quantità scalari \mathbf{P}_E , \mathbf{P}_N , \mathbf{P}_U della (5.2.2.8) sono le componenti del vettore \mathbf{P}' espresse in coordinate ENU rispetto all'origine locale \mathbf{O} . Pertanto, il vettore

$$\mathbf{P}_{\text{ENU}} = \begin{bmatrix} \mathbf{P}_E \\ \mathbf{P}_N \\ \mathbf{P}_U \end{bmatrix} \quad (5.2.2.9)$$

indica il vettore posizione della camera in coordinate ENU per un dato frame.

5.2.3 Orientare la camera nel sistema ENU

Ricavate le posizioni della camera, occorre determinare il suo orientamento nella scena ai fini della calibrazione.

Come specificato nel capitolo 4, l'orientamento della camera è definito attraverso gli angoli di Eulero, in particolare dagli angoli di Pan, Tilt e Roll. Un problema classico a riguardo consiste nel determinare il valore di questi angoli, noti il sistema di coordinate iniziali $X_0Y_0Z_0$ ed il sistema di coordinate $X_3Y_3Z_3$ al termine delle rotazioni. Si tratta, dunque, di calcolare gli angoli che permettono di ottenere il sistema $X_3Y_3Z_3$ a partire dal sistema $X_0Y_0Z_0$.

Dall'analisi del file JSON svolta nel paragrafo 4.2.2, segue che gli angoli in esso riportati corrispondono agli angoli delle rotazioni necessarie per mappare il sistema degli assi terrestri xyz nel sistema degli assi locali della camera XYZ, per ogni frame. Dunque, questa interpretazione ci suggerisce che, a partire dai dati forniti da GES, possiamo ricavare la configurazione finale del sistema della camera per ogni frame.

Nel paragrafo 4.1 si è detto che se gli angoli di Pan, Tilt e Roll sono nulli, si ottiene una camera che guarda in direzione perpendicolare al suolo; a partire da questa configurazione, cambiando il valore degli angoli di Pan, Tilt e Roll, è possibile modificare l'orientamento della camera nella scena, fino ad ottenere un orientamento finale definitivo, descritto dai dati del file JSON. Partendo da questo presupposto, dunque, siamo in grado di ricostruire la configurazione iniziale del sistema della camera.

Sia data la posizione \mathbf{P}_i della camera, per ogni frame i della clip. Definiamo con CAM_0 la configurazione iniziale degli assi della camera, quando gli angoli di Pan, Tilt e Roll sono tutti nulli; dunque, la camera guarda in direzione perpendicolare al suolo. Definiamo con CAM_1 la configurazione finale degli assi della camera, ovvero la configurazione che si ottiene dopo aver orientato la camera attraverso gli angoli di Pan, Tilt e Roll.

Dunque, l'idea per ricostruire l'orientamento della camera frame per frame è quella di calcolare, per ogni frame, gli angoli α , β , γ , rispettivamente di Pan, Tilt e Roll, che permettono di ottenere il sistema CAM_1 a partire dal sistema CAM_0 .

Ciò significa che, per conoscere l'esatto modo in cui la camera è orientata frame per frame, è necessario determinare i sistemi CAM_0 e CAM_1 per ogni frame.

Determiniamo il sistema CAM_0 .

Siano

$$\mathbf{x}_{\text{Earth}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{y}_{\text{Earth}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{z}_{\text{Earth}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.2.3.1)$$

i versori associati al sistema degli assi terrestri, con origine nel centro della Terra, come descritti nel paragrafo 4.1. Sia \mathbf{P}_{Eq} la posizione della camera nel punto di coordinate (0°N , 0°E), ovvero il punto di intersezione tra l'Equatore ed il Meridiano di Greenwich. Sia CAM_{Eq} la configurazione del sistema della camera in posizione \mathbf{P}_{Eq} data da angoli di Pan, Tilt e Roll tutti nulli (Fig.5.6a). Per le impostazioni di GES, dunque, all'Equatore, la camera guarda in direzione perpendicolare al suolo. Esportando la scena con queste impostazioni da GES, si ottengono i parametri di "Camera Rotation" descritti dalla (4.2.2.1). Rinominiamo questi angoli, in

$$\alpha_{\text{Eq}} = 270^{\circ} \quad \beta_{\text{Eq}} = -90^{\circ} \quad \gamma_{\text{Eq}} = 0^{\circ} \quad (5.2.3.2)$$

Gli angoli descritti in (5.2.3.2) permettono di mappare il sistema degli assi terrestri in (5.2.3.1) negli assi che costituiscono il sistema CAM_{Eq} . Pertanto, il sistema CAM_{Eq} sarà dato dalle seguenti rotazioni descritte in (5.2.3.3), (5.2.3.4), (5.2.3.5):

$$\begin{cases} \mathbf{x}'_{\text{Earth}} = \mathbf{R}_{\mathbf{x}_{\text{Earth}}}(\alpha_{\text{Eq}}) \mathbf{x}_{\text{Earth}} \\ \mathbf{y}'_{\text{Earth}} = \mathbf{R}_{\mathbf{x}_{\text{Earth}}}(\alpha_{\text{Eq}}) \mathbf{y}_{\text{Earth}} \\ \mathbf{z}'_{\text{Earth}} = \mathbf{R}_{\mathbf{x}_{\text{Earth}}}(\alpha_{\text{Eq}}) \mathbf{z}_{\text{Earth}} \end{cases} \quad (5.2.3.3)$$

dove $\mathbf{R}_{\mathbf{x}_{\text{Earth}}}(\alpha_{\text{Eq}})$ indica la matrice di rotazione di angolo α_{Eq} intorno all'asse $\mathbf{x}_{\text{Earth}}$;

$$\begin{cases} \mathbf{x}''_{\text{Earth}} = \mathbf{R}_{\mathbf{y}'_{\text{Earth}}}(\beta_{\text{Eq}}) \mathbf{x}'_{\text{Earth}} \\ \mathbf{y}''_{\text{Earth}} = \mathbf{R}_{\mathbf{y}'_{\text{Earth}}}(\beta_{\text{Eq}}) \mathbf{y}'_{\text{Earth}} \\ \mathbf{z}''_{\text{Earth}} = \mathbf{R}_{\mathbf{y}'_{\text{Earth}}}(\beta_{\text{Eq}}) \mathbf{z}'_{\text{Earth}} \end{cases} \quad (5.2.3.4)$$

dove $\mathbf{R}_{\mathbf{y}'_{\text{Earth}}}(\beta_{\text{Eq}})$ è la matrice che descrive la rotazione di angolo β_{Eq} intorno all'asse $\mathbf{y}'_{\text{Earth}}$;

$$\begin{cases} \mathbf{x}'''_{\text{Earth}} = \mathbf{R}_{\mathbf{z}''_{\text{Earth}}}(\gamma_{\text{Eq}}) \mathbf{x}''_{\text{Earth}} \\ \mathbf{y}'''_{\text{Earth}} = \mathbf{R}_{\mathbf{z}''_{\text{Earth}}}(\gamma_{\text{Eq}}) \mathbf{y}''_{\text{Earth}} \\ \mathbf{z}'''_{\text{Earth}} = \mathbf{R}_{\mathbf{z}''_{\text{Earth}}}(\gamma_{\text{Eq}}) \mathbf{z}''_{\text{Earth}} \end{cases} \quad (5.2.3.5)$$

dove $\mathbf{R}_{\mathbf{z}''_{\text{Earth}}}(\gamma_{\text{Eq}})$ indica la matrice di rotazione di angolo γ_{Eq} intorno all'asse $\mathbf{z}''_{\text{Earth}}$:

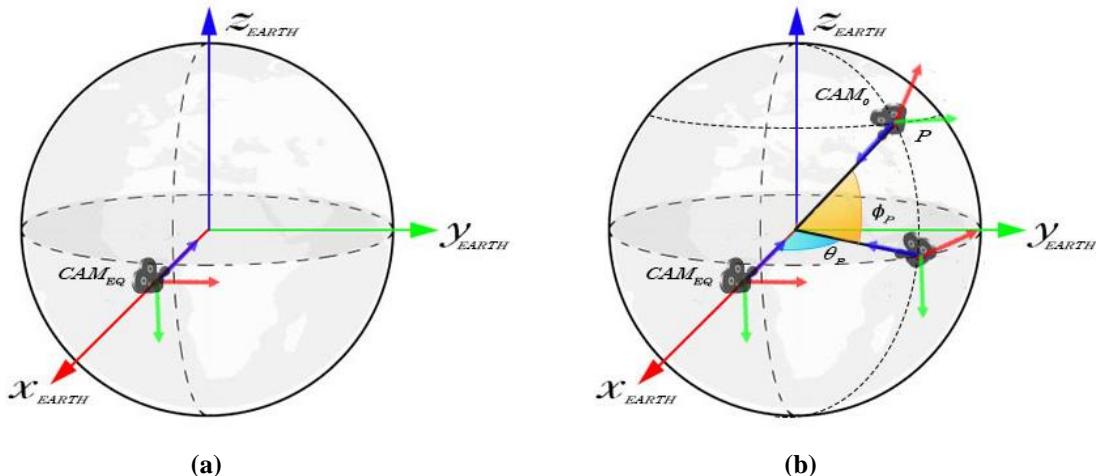


Figura 5.6. Costruzione del sistema CAM₀ a partire dal sistema CAM_{Eq} per ogni frame *i* dell'animazione. (a) Il sistema CAM_{Eq}, per angoli di Pan, Tilt e Roll nulli. (b) Il sistema CAM₀ per angoli di Pan, Tilt e Roll nulli, ottenuto ruotando il sistema CAM_{Eq} prima di angolo θ_P intorno all'asse z, poi di angolo ϕ_P intorno all'asse y.

Dunque, il sistema CAM_{Eq} è definito dagli assi

$$\mathbf{X}_{\text{camEq}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{y}_{\text{camEq}} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{z}_{\text{camEq}} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad (5.2.3.6)$$

Definito il sistema CAM_{Eq} per il frame **i**, a partire da CAM_{Eq} possiamo ricavare il sistema $CAM_{0,i}$ al frame **i**. Per calcolare il sistema $CAM_{0,i}$, l'idea è quella di spostare il sistema CAM_{Eq} nella posizione in cui si trova la camera al frame **i**, sfruttando gli angoli di latitudine e longitudine che descrivono tale posizione. E' bene far notare che il sistema CAM_{Eq} non presenta la dipendenza da **i**. Ciò significa che tale sistema non cambierà mai al variare di ogni frame **i**; pertanto, CAM_{Eq} sarà sempre definito dalla (5.2.3.6) per ogni frame. Invece, il sistema $CAM_{0,i}$ cambierà sempre al variare di **i**, in quanto tale sistema dipende dalla posizione che la camera assume in ogni frame **i**.

Determiniamo, dunque, il sistema $CAM_{0,i}$ al variare della posizione della camera per ogni frame i .

Sia \mathbf{P}_i la posizione della camera relativa al frame i ; siano $\phi_{P,i}$ e $\theta_{P,i}$, rispettivamente, gli angoli di latitudine e longitudine che individuano la posizione della camera \mathbf{P}_i al frame i . Ruotando il sistema $CAM_{Eq,i}$ prima intorno all'asse z della Terra di angolo $\theta_{P,i}$ (5.2.3.7), poi intorno al nuovo asse y terrestre con una rotazione di angolo $\phi_{P,i}$ (5.2.3.8), si ottiene il sistema $CAM_{0,i}$ (Fig.5.6b).

Dunque, le seguenti equazioni fanno riferimento alla trasformazione descritta:

$$\begin{cases} \mathbf{x}'_{cam_0,i} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{x}_{cam_{Eq}} \\ \mathbf{y}'_{cam_0,i} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{y}_{cam_{Eq}} \\ \mathbf{z}'_{cam_0,i} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{z}_{cam_{Eq}} \end{cases} \quad (5.2.3.7)$$

$$CAM_{0,i} = \begin{cases} \mathbf{x}_{cam_0,i} = R_{y_{Earth,\theta}}(\phi_{P,i}) \mathbf{x}'_{cam_0,i} \\ \mathbf{y}_{cam_0,i} = R_{y_{Earth,\theta}}(\phi_{P,i}) \mathbf{y}'_{cam_0,i} \\ \mathbf{z}_{cam_0,i} = R_{y_{Earth,\theta}}(\phi_{P,i}) \mathbf{z}'_{cam_0,i} \end{cases} \quad (5.2.3.8)$$

dove

$$\begin{cases} \mathbf{x}_{Earth,\theta} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{x}_{Earth} \\ \mathbf{y}_{Earth,\theta} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{y}_{Earth} \\ \mathbf{z}_{Earth,\theta} = R_{z_{Earth}}(\theta_{P,i}) \mathbf{z}_{Earth} \end{cases} \quad (5.2.3.9)$$

La (5.2.3.8) definisce la configurazione del sistema CAM_0 per il frame \mathbf{i} . La (5.2.3.8), dunque, descrive una camera che guarda in direzione perpendicolare al suolo, con angoli di Pan, Tilt e Roll impostati a 0° nel frame \mathbf{i} (Fig.5.7a). Ciò corrisponde alle impostazioni di default della camera di GES previste in posizione \mathbf{P}_i .

Definito il sistema CAM_0 per il frame \mathbf{i} , dobbiamo costruire il sistema CAM_1 .

Come detto in precedenza, il sistema CAM_1 corrisponde alla configurazione degli assi della camera dopo essere stata orientata attraverso gli angoli di Eulero (Fig.5.7b).

Come il sistema $CAM_{0,i}$ anche il sistema $CAM_{1,i}$ presenta una dipendenza da \mathbf{i} , in quanto l'orientamento della camera può variare ad ogni frame \mathbf{i} .

Il sistema $CAM_{1,i}$ al frame \mathbf{i} può essere ricavato a partire dai dati forniti da GES. Per ogni frame, la struttura “Camera Rotation” del file JSON fornisce gli angoli α_{JSON} , β_{JSON} , γ_{JSON} ; come detto in precedenza, tali angoli corrispondono agli angoli delle rotazioni intorno agli assi x, y, z della Terra necessarie per allineare il sistema degli assi terrestri con quello locale della camera, dopo averla orientata attraverso gli angoli di Eulero. Anche gli angoli $\alpha_{JSON,i}$, $\beta_{JSON,i}$, $\gamma_{JSON,i}$ sono dipendenti da \mathbf{i} , in quanto possono variare per ogni frame \mathbf{i} . Dunque, sostituendo, rispettivamente, i valori degli angoli $\alpha_{JSON,i}$ nella (4.2.2.6), $\beta_{JSON,i}$ nella (4.2.2.7), $\gamma_{JSON,i}$ nella (4.2.2.8), si ricava il sistema $CAM_{1,i}$ al frame \mathbf{i} dato dai versori

$$CAM_{1,i} = \begin{cases} \mathbf{x}_{cam_1,i} = R_{z_2}(\gamma_{JSON,i}) \mathbf{x}_2 \\ \mathbf{y}_{cam_1,i} = R_{z_2}(\gamma_{JSON,i}) \mathbf{y}_2 \\ \mathbf{z}_{cam_1,i} = R_{z_2}(\gamma_{JSON,i}) \mathbf{z}_2 \end{cases} \quad (5.2.3.10)$$

Conoscendo i sistemi di configurazione $CAM_{0,i}$ e $CAM_{1,i}$ della camera al frame \mathbf{i} , possiamo ricavare gli angoli di Eulero $\alpha_{JSON,i}$, $\beta_{JSON,i}$, $\gamma_{JSON,i}$, che permettono di orientare la camera al frame \mathbf{i} .

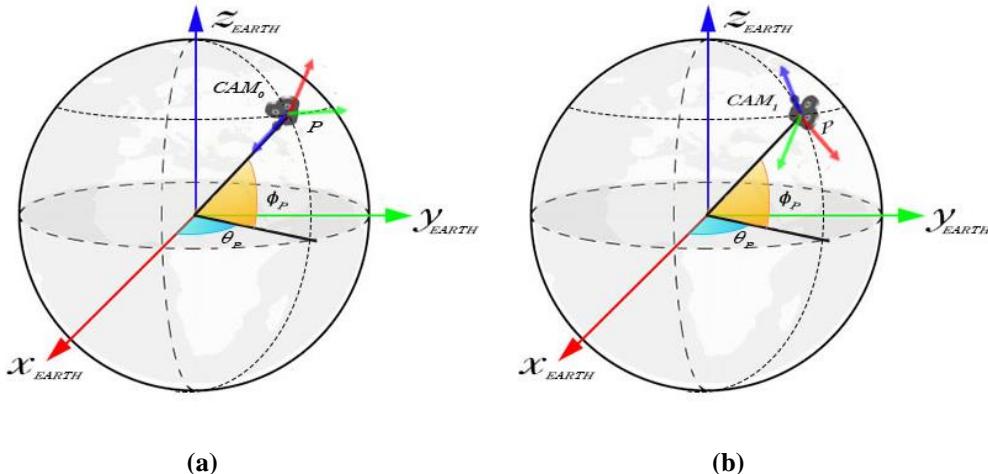


Figura 5.7. I sistemi CAM_0 e CAM_1 per un dato frame i dell’animazione. (a) Il sistema CAM_0 con angoli di Pan, Tilt e Roll nulli. (b) Il sistema CAM_1 ottenuto ruotando la camera con angoli di Pan, Tilt e Roll secondo la procedura descritta dalle API di Google Earth riassunta in (5.2.3.11).

Prima di passare al calcolo di questi angoli, è bene fare alcune considerazioni sulla composizione di rotazioni che mappa il sistema CAM_0 nel sistema CAM_1 .

La documentazione relativa alle API di Google Earth, fornisce una descrizione sull'ordine in cui GES effettua le trasformazioni che consentono di orientare la camera nella scena. L'algoritmo usato dal software per svolgere questo calcolo è suddiviso nelle seguenti fasi:

- ricavare la posizione \mathbf{P} della camera espressa in termini di latitudine ϕ , longitudine θ e altitudine h . Queste informazioni sono impostate dall'utente nei fotogrammi chiave o vengono calcolate per interpolazione se si trattano di informazioni legate a frame intermedi;
 - individuare la posizione di \mathbf{P} della camera sulla sfera di raggio medio R , che approssima la superficie terrestre, attraverso gli angoli di latitudine ϕ e longitudine θ ;
 - traslare la camera e, dunque, il suo sistema di assi, lungo la normale alla superficie terrestre nel punto \mathbf{P} da un'altitudine di 0 m s.l.m. fino alla sua altitudine h stabilita;
 - se α, β, γ sono gli angoli che descrivono le operazioni di Pan, Tilt e Roll, e il sistema locale della camera ad essa solidale è costituito dagli assi $X_0Y_0Z_0$, per orientare la camera nella scena, si ruotano gli assi X_0, Y_0, Z_0 intorno a Z_0 di angolo α , ottenendo il sistema $X_1Y_1Z_1$;
 - si ruota il sistema $X_1Y_1Z_1$ intorno all'asse X_1 di angolo β , ottenendo il sistema $X_2Y_2Z_2$;
 - si ruota il sistema $X_2Y_2Z_2$ intorno all'asse Z_2 di angolo γ , ottenendo il sistema XYZ degli assi della camera orientata nella scena.

Le rotazioni effettuate in questo algoritmo vengono riproposte nello schema in (5.2.3.11):

$$X_0, Y_0, Z_0 \rightarrow_{R_{Z_0}} (\alpha) \rightarrow X_1, Y_1, Z_1 \rightarrow_{R_{X_1}} (\beta) \rightarrow X_2, Y_2, Z_2 \rightarrow_{R_{Z_2}} (\gamma) \rightarrow X, Y, Z \quad (5.2.3.11)$$

Sapere l'ordine in cui vengono effettuate le rotazioni per passare da $X_0Y_0Z_0$ a XYZ , consente di ricavare il valore corretto degli angoli di Pan, Tilt e Roll. Grazie alle informazioni contenute nella documentazione, sappiamo, quindi, che per passare dal sistema CAM_0 al sistema CAM_1 , il sistema di partenza deve essere ruotato intorno all'asse z (Z_0), poi intorno al nuovo asse x (X_1) e poi intorno al nuovo asse z (Z_2), in quanto tali assi cambiano il modo in cui sono orientati, essendo assi mobili. A questo punto, si può procedere a ritroso. Dunque, conoscendo il sistema di partenza $CAM_{0,i}$, il sistema di arrivo $CAM_{1,i}$ e le rotazioni che portano $CAM_{0,i}$ in $CAM_{1,i}$, possiamo ricavare il valore degli angoli di Pan, Tilt e Roll.

Indichiamo con α l'angolo di Pan, β l'angolo di Tilt e con γ l'angolo di Roll. Per semplicità di calcoli, ricaviamo prima β .

Siano dati i sistemi CAM_0 e CAM_1 ; per rendere più chiaro il procedimento, facendo riferimento alla Figura 5.8, indichiamo con,

$$CAM_0 = \begin{cases} \mathbf{x}_{cam_0,i} \\ \mathbf{y}_{cam_0,i} \\ \mathbf{z}_{cam_0,i} \end{cases} = \begin{cases} \mathbf{x}_b \\ \mathbf{y}_b \\ \mathbf{z}_b \end{cases} \quad (5.2.3.12)$$

definito secondo la (5.2.3.8), dove il pedice “b” si riferisce agli assi del sistema di Figura 5.8 colorati in blu, e

$$CAM_1 = \begin{cases} \mathbf{x}_{cam_1,i} \\ \mathbf{y}_{cam_1,i} \\ \mathbf{z}_{cam_1,i} \end{cases} = \begin{cases} \mathbf{x}_r \\ \mathbf{y}_r \\ \mathbf{z}_r \end{cases} \quad (5.2.3.13)$$

definito secondo la (5.2.3.10), dove il pedice “r” si riferisce agli assi del sistema di Figura 5.8 colorati in rosso.

Vogliamo ricavare l'angolo β ; il valore di tale angolo può essere ottenuto ricorrendo alla trigonometria. Sia, dunque, \mathbf{z}_r il vettore di coordinate

$$\mathbf{z}_r = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix} \quad (5.2.3.14)$$

calcolate rispetto al sistema CAM_0 . Dalla Fig.5.8a si osserva che z_3 è la proiezione di \mathbf{z}_r sull'asse \mathbf{z}_b , pertanto, si ha che

$$z_3 = \|\mathbf{z}_r\| \cos(\beta) = \cos(\beta) \quad (5.2.3.15)$$

essendo \mathbf{z}_r un vettore di norma unitaria. Traducendo questo concetto in termini di prodotto scalare, si ottiene

$$z_3 = \langle \mathbf{z}_r, \mathbf{z}_b \rangle \quad (5.2.3.16)$$

Dalla (5.2.3.15) si ricava che

$$z_3 = \langle \mathbf{z}_r, \mathbf{z}_b \rangle = \cos(\beta) \quad (5.2.3.17)$$

e per la (5.2.3.17) segue che

$$\cos(\beta) = \langle \mathbf{z}_r, \mathbf{z}_b \rangle \quad (5.2.3.18)$$

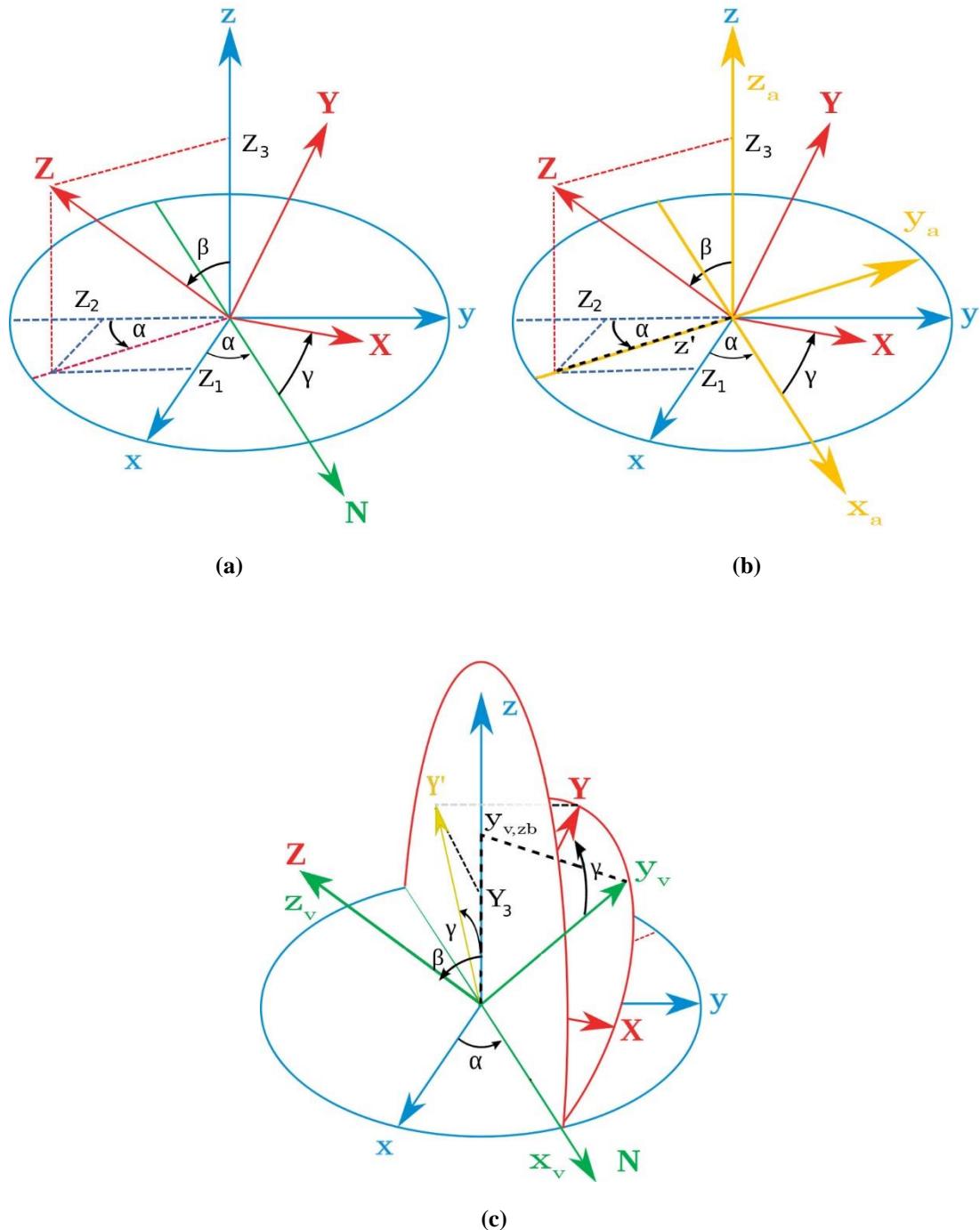


Figura 5.8. Procedura per ricavare gli angoli di Pan, Tilt e Roll a partire dai sistemi CAM_0 e CAM_1 , ricorrendo alla trigonometria. (a) Ricavare l'angolo di Tilt β . (b) Ricavare l'angolo di Pan α . (c) Ricavare l'angolo di Roll γ .

Dalla trigonometria si osserva che

$$\sin(\beta)^2 + \cos(\beta)^2 = 1 \quad (5.2.3.19)$$

pertanto, dalla (5.2.3.19) si può ricavare che

$$\sin(\beta) = \sqrt{1 - \cos^2(\beta)} \quad (5.2.3.20)$$

Sostituendo la (5.2.3.12) e la (5.2.3.13) nella (5.2.3.18) e in (5.2.3.20), si ottiene

$$\cos(\beta) = \langle \mathbf{z}_{\text{cam}_1}, \mathbf{z}_{\text{cam}_0} \rangle \quad (5.2.3.21)$$

$$\sin(\beta) = \sqrt{1 - \langle \mathbf{z}_{\text{cam}_1}, \mathbf{z}_{\text{cam}_0} \rangle^2} \quad (5.2.3.22)$$

Dalla (5.2.3.21) e la (5.2.3.22), possiamo ricavare il valore dell'angolo β , ricorrendo alla funzione *atan2* prevista dall'ambiente Matlab®, come suggerito in [13]. Tale funzione è un'estensione della funzione arcotangente, consentendo di calcolare il valore di β in un range di valori più ampio [15], essendo $-\frac{\pi}{2} < \text{arctg}(\beta) < \frac{\pi}{2}$, mentre $-\pi < \text{atan2}(a, b) < \pi$. La funzione *atan2*, dunque, richiede un valore in input in più rispetto alla funzione arcotangente, per un totale di due parametri.

Poiché la (5.2.3.21) e (5.2.3.22) hanno lo stesso denominatore (e vale 1), allora il valore dell'angolo β sarà dato da

$$\beta = \text{atan2}(\sqrt{1 - \langle \mathbf{z}_{\text{cam}_1}, \mathbf{z}_{\text{cam}_0} \rangle^2}, \langle \mathbf{z}_{\text{cam}_1}, \mathbf{z}_{\text{cam}_0} \rangle) \quad (5.2.3.23)$$

Le relazioni trovate tra l'angolo β e i sistemi CAM₀ e CAM₁ consentono di calcolare α . La rotazione di angolo α intorno a \mathbf{z}_b trasforma il sistema CAM₀ nel sistema CAM_a, secondo la Fig.5.8b, dato da

$$\text{CAM}_a = \begin{cases} \mathbf{x}_a \\ \mathbf{y}_a \\ \mathbf{z}_a \end{cases} \quad (5.2.3.24)$$

Dunque, i vettori \mathbf{x}_b , \mathbf{y}_b , \mathbf{x}_a , \mathbf{y}_a giacciono tutti sullo stesso piano $x_b y_b$. Sia z' la proiezione di \mathbf{z}_r su \mathbf{y}_a , dunque, la proiezione di \mathbf{z}_r sul piano $x_b y_b$ (Fig.5.8b), data da

$$z' = \|\mathbf{z}_r\| \cos\left(\frac{\pi}{2} - \beta\right) = \sin(\beta) \quad (5.2.3.25)$$

essendo \mathbf{z}_r un vettore di norma unitaria. Poiché \mathbf{y}_a è il vettore che si ottiene ruotando \mathbf{y}_b di angolo α intorno a \mathbf{z}_b , si ha che l'angolo tra \mathbf{y}_b e \mathbf{y}_a è proprio α . Questa informazione ci permette di considerare z_2 come la proiezione di z' su \mathbf{y}_b , e, dunque,

$$z_2 = z' \cos(\alpha) \quad (5.2.3.26)$$

Sostituendo il valore di z' dato dalla (5.2.3.25) nella (5.2.3.26), si ottiene che

$$z_2 = \sin(\beta) \cos(\alpha) \quad (5.2.3.27)$$

da cui si ricava che

$$\cos(\alpha) = \frac{z_2}{\sin(\beta)} \quad (5.2.3.28)$$

Poiché \mathbf{z}_2 è ottenuto attraverso una doppia proiezione, possiamo pensare tale vettore come la proiezione di \mathbf{z}_r su \mathbf{y}_b ; dunque, traducendo questo concetto in termini di prodotto scalare, possiamo riscrivere la (5.2.3.28) in

$$\cos(\alpha) = \frac{\langle \mathbf{z}_r, \mathbf{y}_b \rangle}{\sin(\beta)} \quad (5.2.3.29)$$

Sostituendo le (5.2.3.22), (5.2.3.12), (5.2.3.13) nella (5.2.3.29) si ottiene che

$$\cos(\alpha) = \frac{\langle \mathbf{z}_{cam_1}, \mathbf{y}_{cam_0} \rangle}{\sqrt{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.30)$$

La (5.2.3.30) descrive le relazioni tra l'angolo α ed i sistemi CAM_0 , CAM_1 . Dalla trigonometria, per la (5.2.3.19), si ricava che

$$\sin(\alpha) = \sqrt{1 - \cos^2(\alpha)} \quad (5.2.3.31)$$

Sostituendo la (5.2.3.30) nella (5.2.3.31), segue che

$$\sin(\alpha) = \sqrt{1 - \frac{\langle \mathbf{z}_{cam_1}, \mathbf{y}_{cam_0} \rangle^2}{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.32)$$

Portando la (5.2.3.32) e la (5.2.3.30) allo stesso denominatore, per poter calcolare il valore di α con la funzione *atan2* di Matlab®, si ottiene

$$\sin(\alpha) = \sqrt{\frac{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2 - \langle \mathbf{z}_{cam_1}, \mathbf{y}_{cam_0} \rangle^2}{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.33)$$

Dunque, segue che

$$\alpha = atan2(\sqrt{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2 - \langle \mathbf{z}_{cam_1}, \mathbf{y}_{cam_0} \rangle^2}, \langle \mathbf{z}_{cam_1}, \mathbf{y}_{cam_0} \rangle) \quad (5.2.3.34)$$

Determinati α e β , rimane da trovare γ . Dati i sistemi CAM_0 e CAM_1 definiti, rispettivamente, dalla (5.2.3.12) e (5.2.3.13), si definisce linea dei nodi la retta N generata dall'intersezione dei piani $x_b y_b$ e $x_r y_r$ tale che

$$\mathbf{N} = \mathbf{z}_b \times \mathbf{z}_r \quad (5.2.3.35)$$

La rotazione di angolo β intorno a \mathbf{x}_a trasforma il sistema CAM_a nel sistema CAM_v , secondo la Figura 5.7c dato da,

$$CAM_v = \begin{cases} \mathbf{x}_v \\ \mathbf{y}_v \\ \mathbf{z}_v \end{cases} \quad (5.2.3.36)$$

in cui l'asse \mathbf{x}_v coincide sia con sè stesso, in quanto asse di rotazione, che con N ; l'asse \mathbf{z}_v coincide con \mathbf{z}_r , in quanto asse intorno a cui avviene la rotazione di angolo γ .

Segue, dunque, che \mathbf{x}_v giace sul piano $x_r y_r$, per la definizione di linea dei nodi. Si consideri, ora, la rotazione di angolo γ intorno all'asse \mathbf{z}_v . Poiché tale rotazione manda l'asse \mathbf{x}_v in \mathbf{x}_r e \mathbf{y}_v in \mathbf{y}_r , ed i versori \mathbf{x}_r , \mathbf{y}_r , \mathbf{x}_v appartengono al piano $x_r y_r$, essendo CAM_v un sistema di assi ortogonali, segue, intuitivamente, che \mathbf{y}_v giace anch'esso sul piano $x_r y_r$, tra \mathbf{y}_r e \mathbf{x}_r . Consideriamo, ora il piano verticale definito dai vettori \mathbf{z}_b e \mathbf{N} , dunque, il piano $z_b N$. La rotazione di angolo γ trasforma \mathbf{y}_v in \mathbf{y}_r . Sia y' la proiezione di \mathbf{y}_r sul piano $z_b N$. Essendo \mathbf{z}_r e \mathbf{y}_r tra loro ortogonali e, sapendo che l'angolo tra \mathbf{z}_r e \mathbf{z}_b vale β per la rotazione di angolo β , allora l'angolo tra y' ed \mathbf{y}_r vale $\frac{\pi}{2} - \beta$. Dunque, si ha che

$$y' = \|\mathbf{y}_r\| \cos(\frac{\pi}{2} - \beta) = \sin(\beta) \quad (5.2.3.37)$$

essendo \mathbf{y}_r un vettore di norma unitaria. Abbiamo detto che i vettori \mathbf{x}_v , \mathbf{y}_v , \mathbf{z}_v sono ortogonali tra loro, e che $\mathbf{z}_b \perp \mathbf{N}$, $\mathbf{N} \perp \mathbf{y}_v$, in quanto $\mathbf{N} = \mathbf{x}_v$, ma \mathbf{z}_b non è ortogonale a \mathbf{y}_v . Ciò comporta che, calcolando la proiezione di \mathbf{y}_v sul piano $z_b N$, si avrà che la componente di \mathbf{y}_v lungo \mathbf{z}_b è data da

$$y_{v,z_b} = \|\mathbf{y}_v\| \cos(\frac{\pi}{2} - \beta) = \sin(\beta) \quad (5.2.3.38)$$

dove l'angolo tra \mathbf{z}_b e \mathbf{y}_v vale $\frac{\pi}{2} - \beta$, in quanto \mathbf{y}_v giace sullo stesso piano di \mathbf{y}_r . Calcoliamo, ora, la componente di \mathbf{y}_v lungo \mathbf{x}_v . Poiché $\mathbf{x}_v \perp \mathbf{y}_v$, allora

$$y_{v,x_v} = 0 \quad (5.2.3.39)$$

Dunque, le equazioni (5.2.3.38) e (5.2.3.39) suggeriscono che, intuitivamente, la proiezione di \mathbf{y}_v sul piano $z_b N$ giace sull'asse \mathbf{z}_b . Ciò viene suggerito anche dalla Figura 5.8c. L'idea per ricavare γ è quella di verificare, intuitivamente, che la proiezione di \mathbf{y}_v sul piano $z_b N$ giace lungo \mathbf{z}_b e che y' è il vettore che si ottiene ruotando di angolo γ il vettore proiettato di \mathbf{y}_v . La prima condizione viene soddisfatta. Verifichiamo la seconda. Dunque, sia y_3 la proiezione di y' sull'asse \mathbf{z}_b . In questi termini, si può pensare y_3 come la proiezione di \mathbf{y}_r su \mathbf{z}_b . Ma \mathbf{y}_r è ottenuto ruotando \mathbf{y}_v di angolo γ intorno a \mathbf{z}_v . Si sa che i vettori \mathbf{x}_v , \mathbf{y}_v , \mathbf{x}_r , \mathbf{y}_r giacciono tutti sullo stesso piano $x_r y_r$, e tale piano, dunque, rimane sempre ad una distanza fissa rispetto al piano $z_b N$, sia per la rotazione di angolo β , sia per la rotazione di angolo γ . L'angolo formato tra questi due piani misurerà sempre $\frac{\pi}{2} - \beta$. Da ciò segue, intuitivamente, che se si proietta il vettore \mathbf{y}_v sul piano $z_b N$ (la cui proiezione giace su \mathbf{z}_b) e poi si ruota tale vettore di angolo γ intorno all'asse \mathbf{z}_v (ricavando \mathbf{y}_r), si ottiene che anche la proiezione y_{v,z_b} verrà ruotata di angolo γ , ed essa corrisponderà al vettore y' .

Questo procedimento spiega, intuitivamente, la relazione che intercorre tra i vettori y' , y_3 e l'angolo γ . Dunque, ora, si hanno a disposizione tutte le informazioni utili per ricavare γ .

Abbiamo detto che y' è la proiezione di \mathbf{y}_r sul piano $z_b N$ e che y_3 è la proiezione di y' su \mathbf{z}_b . Dunque, si ha che

$$y_3 = y' \cos(\gamma) \quad (5.2.3.40)$$

Dalla definizione di y' secondo la (5.2.3.37) segue che

$$y_3 = \sin(\beta) \cos(\gamma) \quad (5.2.3.41)$$

da cui si ricava che

$$\cos(\gamma) = \frac{y_3}{\sin(\beta)} \quad (5.2.3.42)$$

Poiché y_3 è la proiezione di \mathbf{y}_r su \mathbf{z}_b , traducendo questo concetto in termini di prodotto scalare, con riferimento alla (5.2.3.22), (5.2.3.12) (5.2.3.13), la (5.2.3.42) può essere riscritta nella forma

$$\cos(\gamma) = \frac{\langle \mathbf{y}_r, \mathbf{z}_b \rangle}{\sqrt{1 - \langle \mathbf{z}_r, \mathbf{z}_b \rangle^2}} = \frac{\langle \mathbf{y}_{cam_1}, \mathbf{z}_{cam_0} \rangle}{\sqrt{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.43)$$

Per la trigonometria, segue dalla (5.2.3.19) che

$$\sin(\gamma) = \sqrt{1 - \cos^2(\gamma)} \quad (5.2.3.44)$$

Sostituendo la (5.2.3.43) nella (5.2.3.44) si ottiene,

$$\sin(\gamma) = \sqrt{1 - \frac{\langle \mathbf{y}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.45)$$

Portando la (5.2.3.45) e la (5.2.3.43) allo stesso denominatore, per poter calcolare il valore di γ con la funzione *atan2* di Matlab®, si ottiene

$$\sin(\gamma) = \sqrt{\frac{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2 - \langle \mathbf{y}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}} \quad (5.2.3.46)$$

Dunque, segue che

$$\gamma = \text{atan2}(\sqrt{1 - \langle \mathbf{z}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2 - \langle \mathbf{y}_{cam_1}, \mathbf{z}_{cam_0} \rangle^2}, \langle \mathbf{y}_{cam_1}, \mathbf{z}_{cam_0} \rangle) \quad (5.2.3.47)$$

Pertanto, la (5.2.3.34), la (5.2.3.23) e la (5.2.3.47) permettono di ricavare gli angoli α , β , γ , rispettivamente, di Pan, Tilt e Roll, a partire dai sistemi CAM₀ e CAM₁ per ogni frame **i** che compone la clip.

Determinato il modo in cui la camera è orientata in ogni frame, ai fini della calibrazione, si devono determinare le sue proprietà interne.

5.2.4 Stima della lunghezza focale

Quando si ricorre alla tecnica del matchmoving, in fase di calibrazione, al fine di velocizzare tale procedimento, possono essere usate informazioni aggiuntive relative alla camera usata per girare la scena. Sotto questa condizione, possono essere distinte tre tipologie di camera [5]:

- **camere totalmente calibrate:** si conoscono tutte le informazioni della camera relative alla sua posizione, orientamento, tipologia di lenti usate, apertura diaframma, lunghezza focale, dimensione sensore e altre proprietà specifiche della camera usata;
- **camere parzialmente calibrate:** sono conosciute a priori soltanto alcune informazioni relative alla camera usata; tali informazioni possono essere relative alla sua posizione oppure alle sue proprietà;
- **camere non calibrate:** non è possibile conoscere a priori le informazioni relative alla camera usata.

La camera con cui sono realizzate le clip in GES rientra nella seconda tipologia. Infatti, le informazioni che è possibile conoscere a priori sono solo quelle relative alla sua posizione ed al suo orientamento, ovvero tutte informazioni riportate nel file JSON. Gli unici dati noti relativi alle proprietà intrinseche di tale camera virtuale sono l'angolo di FOV verticale ed il fatto che GES usa una pinhole camera, ovvero una camera senza lenti costituita da una piccola apertura attraverso cui passa la luce. Ciò significa che la camera di GES non crea distorsioni dell'immagine, in quanto non prevede uso di lenti; inoltre, ogni elemento della scena è a fuoco. L'angolo di FOV (Field of View), invece, indica quella porzione della scena che viene ripresa attraverso la camera. Pertanto, ai fini della calibrazione, rimane da determinare il valore della lunghezza focale e le dimensioni del sensore. Tali informazioni di tipo fotografico sono necessarie in Maya per poter ricostruire correttamente la prospettiva della scena. Dunque, ricavate queste ultime informazioni mancanti, i dati conosciuti sono sufficienti per poter riprodurre il movimento della camera di GES in Maya.

Il software di modellazione 3D Maya permette di ricostruire le caratteristiche di una camera con cui si è scattata, ad esempio, una fotografia, consentendo di gestire manualmente un'ampia gamma di parametri relativi alle proprietà della camera. Tra questi, i parametri relativi alla dimensione del sensore e alla lunghezza focale, sono preimpostati dal software con valori che corrispondono al formato Full Frame; dunque, è previsto:

- **un sensore con dimensione 24 x 36 mm**, avente un aspect ratio dell'immagine pari a 3:2, dove 24 mm indica la dimensione verticale del sensore e 36 mm quella orizzontale;
- **una lunghezza focale di 35 mm**;
- **l'angolo di FOV orizzontale di 54,43°**.

Ad un primo confronto tra i dati forniti da Maya e il valore dell'angolo di FOV verticale (20°), fornito da GES, ci aspettiamo che GES non utilizzi una camera con formato Full Frame 3:2 per riprendere la scena.

Per verificare se questa ipotesi è corretta seguiremo la seguente procedura.

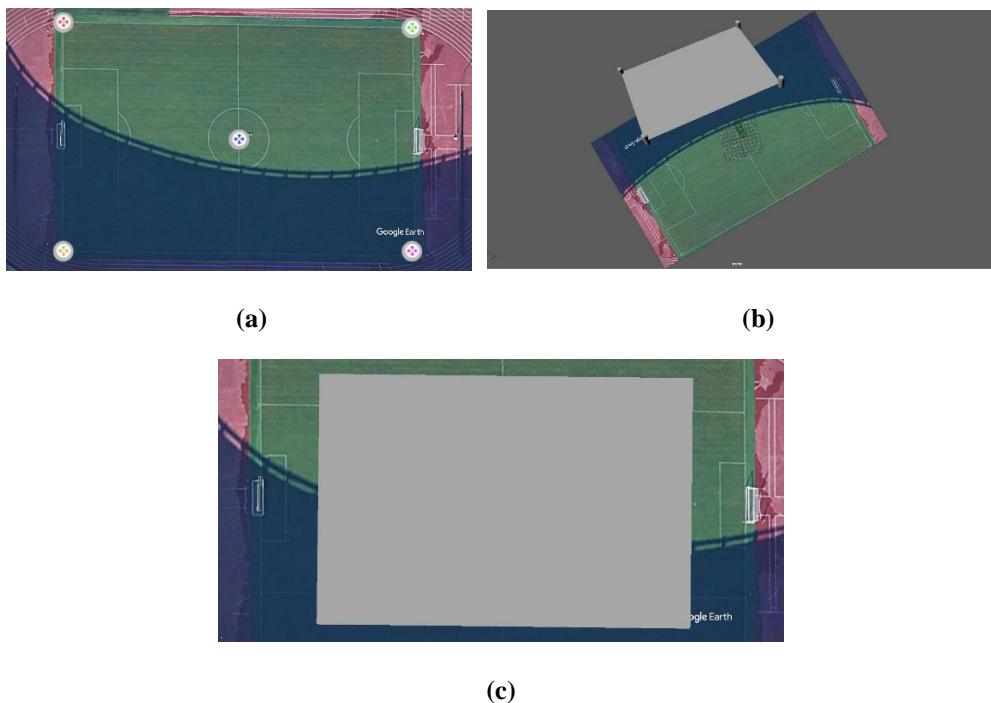


Figura 5.9. Stima della lunghezza focale attraverso un metodo empirico. (a) Quattro trackpoint vengono posizionati in GES in corrispondenza delle quattro bandierine che delimitano i bordi del campo dello Stadio Olimpico. (b) In Maya si ricostruisce il campo dello Stadio Olimpico con le stesse dimensioni reali, fornite dalle coordinate ENU dei trackpoint. (c) La camera di GES non usa un formato Full Frame 3:2, in quanto il rettangolo creato in Maya non si sovrappone al campo dello Stadio Olimpico.

Si realizza in GES una fotografia aerea del campo dello Stadio Olimpico. Per realizzare tale fotografia, si ruota la camera di GES affinchè il campo dello Stadio sia allineato con i bordi del frame, assicurandosi che la camera guardi in direzione perpendicolare al suolo. Si impostano cinque trackpoint; uno al centro dello stadio, considerato come origine locale delle coordinate ENU ed i quattro trackpoint rimanenti ai quattro bordi del campo, in corrispondenza delle quattro bandierine (Fig.5.9a). Vogliamo conoscere l'esatta posizione delle bandierine per poter creare in Maya un piano che abbia le stesse informazioni spaziali e le stesse proporzioni del campo dello Stadio Olimpico; il modo in cui verrà visualizzato tale piano dalla camera creata in Maya permetterà di stabilire se il valore della lunghezza focale adottato da GES corrisponde ad un formato Full Frame 3:2 o meno. Dunque, impostati i trackpoint, esportiamo il fotogramma ed i dati della scena in formato JSON. Le coordinate ENU e l'orientamento della camera da importare in Maya, così come le coordinate ENU dei trackpoint, vengono calcolate, per un solo frame, con il metodo proposto nei paragrafi 5.2.2 e 5.2.3. Si crea, dunque, una camera in Maya avente tali caratteristiche e si associa ad essa un image plane contenente la fotografia scattata precedentemente in GES. Inseriamo nella scena quattro cilindri; al punto centrale (pivot) di ogni cilindro associamo le coordinate ENU di una della quattro bandierine. Costruiamo un piano passante per questi quattro punti, le cui dimensioni coincidano numericamente con quelle del campo dello Stadio Olimpico (Fig.5.9b). Poiché GES fornisce un angolo di FOV verticale di 20°, calcoliamo la rispettiva lunghezza focale supponendo che la dimensione verticale del sensore sia 24 mm. Se GES usa il formato Full Frame 3:2 ci aspettiamo che il piano appena creato coincida perfettamente con il rettangolo che individua il campo dello Stadio nell'image plane. Poiché piano e rettangolo non si sovrappongono, ciò prova

che GES non adotta il formato Full Frame, con un sensore avente aspect ratio di 3:2 (Fig.5.9c).

Dunque, bisogna trovare un valore per le dimensioni del sensore e della lunghezza focale, affichè possa essere riprodotta correttamente in Maya la prospettiva della scena. E' possibile ricavare tali valori attraverso esperimenti empirici, da svolgere direttamente nell'ambiente Maya. Prima di passare a descrivere il metodo adottato per la stima della lunghezza focale, riassumiamo brevemente quali sono i dati noti che possono essere sfruttati a tal fine.

GES ci fornisce una serie di informazioni relative alla scena creata attraverso il file JSON. Scorrendo tale file, oltre al valore dell'angolo di FOV verticale, che si sa essere di 20° , troviamo dei parametri che possono fornire delle informazioni utili; tra questi parametri, vi sono quelli relativi alle dimensioni in pixel di ogni frame. Nel progetto di GES creato, le dimensioni impostate sono 1920 pixel per quella orizzontale e 1080 pixel per quella verticale. Calcolando il rapporto tra la dimensione orizzontale e quella verticale, si ottiene che

$$\frac{1920 \text{ pixel}}{1080 \text{ pixel}} = 1.7 \quad (5.2.4.1)$$

Il valore del rapporto della (5.2.4.1), corrisponde al valore dell'aspect ratio dato dal formato 16:9, infatti

$$\frac{16}{9} = 1.7 \quad (5.2.4.2)$$

Ciò suggerisce che il rapporto tra la dimensione orizzontale del sensore e quella verticale è di 16:9. Pertanto, dalla proporzione

$$16 : 9 = d_H : d_V \quad (5.2.4.3)$$

è possibile ricavare la dimensione orizzontale d_H del sensore, conoscendo quella verticale d_V e viceversa. Inoltre, la relazione che sussiste tra l'angolo di FOV verticale e la dimensione verticale del sensore è data dall'equazione

$$\alpha_V = 2 \operatorname{arctg} \left(\frac{d_V}{2f} \right) \quad (5.2.4.4)$$

in cui α_V indica l'angolo di FOV verticale, d_V la dimensione verticale del sensore e f la lunghezza focale. Pertanto, noti α_V e d_V , dalla (5.2.4.4) si ricava che,

$$f = \frac{d_V}{2 \operatorname{tg} \left(\frac{\alpha_V}{2} \right)} \quad (5.2.4.5)$$

Nel nostro caso, il valore delle dimensioni del sensore e della lunghezza focale non vengono forniti da GES. La (5.2.4.5) suggerisce che per stimare il valore corretto della lunghezza focale occorre conoscere la dimensione verticale del sensore, pertanto, seguiremo un metodo empirico per determinare il suo valore. Tale metodo consiste nel determinare un valore arbitrario della dimensione verticale del sensore e verificare se, per tale valore, il piano modellato coincide con il campo dello Stadio Olimpico.

Attraverso esperimenti pratici, si è dimostrato che il piano in questione si sovrappone perfettamente al campo della Stadio Olimpico impostando la dimensione verticale del sensore della camera a 21 mm.

Dunque, trovato il valore di d_V e conoscendo il valore dell'angolo di FOV verticale α_V , si può calcolare il valore della lunghezza focale secondo la (5.2.4.5). Si ottiene, dunque, una lunghezza focale di 59,5 mm.

Dalla proporzione in (5.2.4.3) si ricava il valore della dimensione orizzontale del sensore, data da

$$d_H = \frac{16}{9} d_V = 1,7 d_V \quad (5.h)$$

da cui segue che la dimensione orizzontale del sensore misura 35,7 mm.

Infine, rimane da calcolare l'angolo di FOV orizzontale α_H , il cui valore si ottiene sostituendo, rispettivamente, α_H e d_H al posto di α_V e d_V nella (5.2.4.4). Segue, dunque, che α_H vale 33,40°.

Impostando in GES valori di angoli di FOV verticali differenti, prove sperimentali dimostrano che i valori trovati per le dimensioni del sensore e lunghezza focale possono considerarsi corretti, in quanto permettono di ottenere una perfetta sovrapposizione tra il campo dello Stadio Olimpico ed il piano creato (Fig.5.10).

La tabella 5.2 mostra il valore della lunghezza focale ottenuto considerando diversi valori di angoli di FOV verticali in GES.

Avendo stimato il valore della lunghezza focale e le dimensioni del sensore della camera è possibile, dunque, riprodurre correttamente la prospettiva della scena. A questo punto, abbiamo determinato tutte le informazioni della camera necessarie ai fini della calibrazione. Calibrata la camera, il prossimo passo sarà quello di importare il movimento della camera in Maya, per testare l'efficacia della tecnica proposta ai fini del matchmoving.

Prima di procedere, però, è necessario fare una considerazione. GES imposta l'angolo di FOV verticale a 20° come valore di default, in quanto tale valore consente di ridurre eventuali distorsioni che potrebbero generarsi quando l'utente naviga attraverso il globo.

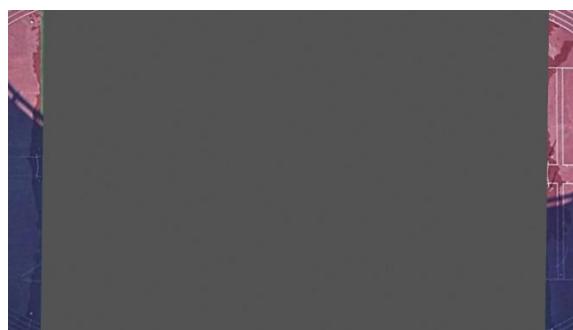


Figura 5.10. Ricostruendo in Maya la camera Full Frame formato 16:9 usata in GES, si ottiene una perfetta sovrapposizione tra il piano modellato ed il campo dello Stadio Olimpico.

α_v	α_H	f
15°	25,2°	79,8 mm
20°	33,4°	59,5 mm
30°	49°	39,2 mm
50°	76,8°	22,5 mm

Tabella 5.2. I valori di angolo di FOV orizzontale (α_H) e della lunghezza focale (f) che si ottengono impostando diversi valori per l'angolo di FOV verticale (α_v), considerando una camera con sensore di dimensioni 21x35,7 mm.

5.3 Animare la camera virtuale

Per poter riprodurre il movimento della camera di GES in Maya, bisogna impostare, per ogni frame, la posizione della camera ed il suo orientamento con i rispettivi valori calcolati durante la calibrazione. Sebbene le clip prese in considerazione in questa tesi siano costituite da soli 96 frame, impostare questi valori manualmente diventa una procedura molto onerosa. Occorre, dunque, trovare una soluzione che possa automatizzare il processo di animazione della camera in Maya, consentendo, allo stesso tempo, di poter trasferire da Matlab® a Maya tutti i dati calcolati in fase di calibrazione per ogni frame. L'incompatibilità tra i due software, però, obbliga a scegliere un linguaggio di programmazione che supporti i formati dei file di output generati in Matlab® e che, allo stesso tempo, sia supportato da Maya. La versione di Autodesk Maya 2016 è compatibile con il linguaggio Python. Sia Matlab® che Python sono in grado di gestire, scrivere ed analizzare file in formato JSON.

L'idea, dunque, è quella di esportare i dati calcolati in Matlab® in un file JSON ed importare tale file in Python, al fine di rendere automatico il processo di animazione della camera in Maya.

5.3.1 Gestione dei dati in output

Per poter esportare i dati calcolati in Matlab® in un file JSON occorre prima organizzare i dati in modo da facilitare la scrittura nel file. Anche in questo caso, la soluzione ideale è quella di poter leggere i dati relativi alla camera per ogni frame da una tabella virtuale. Pertanto, creiamo in Matlab® una matrice F_1 avente dimensione $nf \times 7$, con nf dato dalla (5.2.1.1), in cui le righe corrispondono ai frame e le colonne ai dati relativi alla camera per ogni frame. Dividiamo, nuovamente, i dati della camera in quattro categorie, dove ognuna di queste categorie corrisponderà ad una struttura nel nuovo file JSON che verrà importato in Python:

- **Camera Position:** dalla colonna 1 alla colonna 3, corrisponde alla struttura in cui sono contenute le informazioni relative alla posizione della camera espresse in coordinate ENU, per un dato frame;
- **Camera Rotation:** dalla colonna 4 alla colonna 6, corrisponde alla struttura in cui sono contenute le informazioni relative all'orientamento della camera per un dato frame, ovvero gli angoli di Pan, Tilt e Roll;

- **Lunghezza focale:** colonna 7, corrisponde alla struttura in cui è contenuto il valore della lunghezza focale della camera, per un dato frame.

Le informazioni relative alle dimensioni del sensore, essendo parametri fissi il cui valore non varia al variare dei frame, verranno salvate in una struttura a parte nel file JSON. Non è necessario salvare in questo file il valore dell'angolo di FOV orizzontale, in quanto tale angolo viene calcolato automaticamente da Maya secondo la (5.2.4.4), impostando le dimensioni del sensore e la lunghezza focale della camera.

Dunque, avendo organizzato i dati calcolati in Matlab® in diverse strutture, possiamo esportare tutte queste informazioni nel nuovo file JSON, il quale potrà essere poi utilizzato in Python.

La Tabella 5.3 riassume la struttura della matrice F_1 in cui sono organizzati i dati per i primi 4 frame di animazione.

Frame	Camera Position			Camera Rotation			Lunghezza Focale
	x	y	z	Pan	Tilt	Roll	
0	-6.209734	-0.000352	2.526326	90°	86°	0°	5,95 mm
1	-6.131383	-0.000343	2.497389	91°	86°	0°	5,95 mm
2	-6.053032	-0.000335	2.468452	91°	86°	0°	5,95 mm
3	-5.974682	-0.000326	2.439515	92°	85°	0°	5,95 mm

Tabella 5.3. Struttura della matrice F_1 in cui vengono salvati i dati della camera per i primi quattro frame di animazione. Le righe corrispondono ai frame, le colonne ai dati della camera relativi per ogni frame, divisi in quattro categorie. Le misure riportate nelle colonne relative alla “Camera Position” sono dell’ordine di 10^2 m. Questi dati verranno salvati nel nuovo file JSON da importare in Python.

5.3.2 Realizzare un’animazione in Maya

Prima di descrivere come è stato realizzato il processo di animazione della camera tramite script Python, è bene spiegare la procedura necessaria a creare una semplice animazione in Maya. Supponiamo di avere una scena costituita da un cubo. La scena viene ripresa da una camera in modo tale da vedere il cubo sulla sinistra (Fig.5.11a). Vogliamo realizzare un’animazione in cui il cubo si sposta da sinistra verso destra. Per far questo, aggiungiamo al primo frame della timeline un keyframe, corrispondente alla posizione attuale ed iniziale del cubo. Un keyframe è un particolare fotogramma che consente definire lo stato iniziale, intermedio e finale di un’animazione. Ora, scorriamo la timeline fino ad arrivare all’ultimo frame. Trasliamo il cubo da sinistra verso destra, fino a raggiungere la sua posizione finale desiderata, in corrispondenza della quale si aggiunge un secondo keyframe. A questo punto, abbiamo un set costituito da due keyframe che indicano, rispettivamente, la posizione iniziale e finale del cubo. I frame intermedi che compongono l’animazione vengono calcolati per interpolazione (Fig.5.11b). Dunque, in questo modo è possibile ottenere un’animazione in cui il cubo si sposta da sinistra a destra (Fig.5.11c).

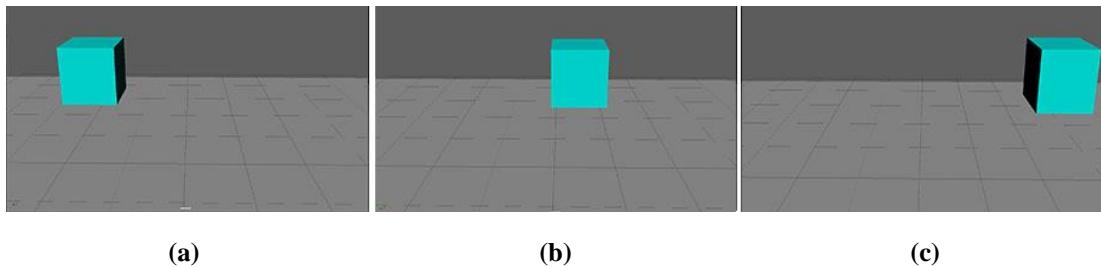


Figura 5.11. Tre frame di un’animazione in Maya. (a) Posizione iniziale del cubo (a sinistra). (b) Posizione intermedia del cubo calcolata per interpolazione. (c) Posizione finale del cubo (a destra).

La procedura appena descritta, è la stessa procedura che occorre seguire per animare la camera di Maya con i dati forniti da GES, elaborati sfruttando Matlab®.

In tal caso, invece di impostare manualmente i keyframe, questi dovranno essere impostati in maniera automatica attraverso uno script, ricorrendo ai comandi che Maya mette a disposizione a tal fine.

5.3.3 Python per l’automatizzazione del processo di animazione della camera

Per riprodurre il movimento della camera di GES in Maya, occorre impostare ogni keyframe dell’animazione con i dati di posizione e orientamento della camera relativi al frame considerato. Poiché la versione di Autodesk Maya 2016 supporta il linguaggio Python, possiamo automatizzare il processo di animazione con uno script implementato in tale linguaggio di programmazione. Prima di procedere, però, è necessario capire come scrivere le istruzioni in Python, affinchè queste possano interagire con gli oggetti di Maya.

Supponiamo, ad esempio, di voler creare, manualmente, una sfera in Maya, sfruttando gli strumenti disponibili nella toolbar (Fig.5.12a). Questa stessa azione può essere riprodotta attraverso un comando MEL (Maya Embedded Language), ovvero il linguaggio di scripting contenuto in Maya (Fig.5.12b). La struttura di ogni comando MEL consiste in una stringa costituita da una serie di flag e dai rispettivi valori che queste assumono [8]. Il comando MEL relativo all’esempio della sfera è dato dalla stringa

```
polySphere -ch on -o on -r 3.015513 ;
```

in cui è possibile riconoscere la struttura

```
commandName -flag1 value1 -flag2 value2 -flag3 value3;
```

dove le flag ch, o, r corrispondono a flag1, flag2, flag3 e on, on, 3.015513 ai rispettivi valori che assumono queste flag. La stringa di comando MEL ci dice che è stata creata una sfera con un raggio che misura 3,015513.

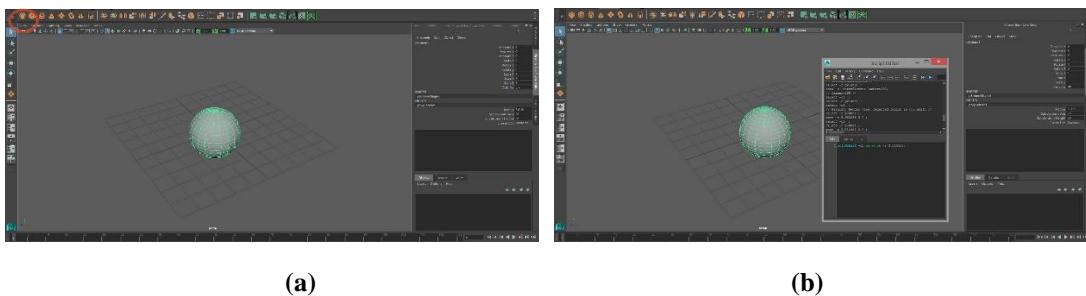


Figura 5.12. Due modalità con cui creare una sfera in Maya. (a) Sfera creata in Maya con gli strumenti a disposizione nella toolbar. (b) Sfera creata in Maya con il comando MEL corrispondente [8].

Dunque, ogni azione effettuata in Maya ha un suo comando MEL corrispondente. Ognuno di questi comandi MEL può essere riscritto con un’istruzione in Python, grazie alla compatibilità tra questo linguaggio di programmazione e Maya.

Ogni istruzione in Python deve essere anch’essa costituita da flag e valori; in questo caso, l’istruzione Python corrispondente all’esempio della sfera sarà

```
maya.cmds.polySphere( radius = 3.015513)
```

Come suggerisce l’istruzione appena mostrata, il comando *polySphere* a cui si ricorre per creare una sfera, fa parte di una libreria (*maya.cmds*) in cui sono contenuti tutti i comandi che consentono l’interazione con Maya; tale interazione avviene tramite script. Dunque, per poter usare i comandi di Maya nell’ambiente Python, è necessario prima importare la libreria in cui tali comandi sono contenuti [8].

Importata la libreria, si può procedere ad importare il file JSON in Python con i dati calcolati in Matlab®. Tale file è costituito da un’unica struttura che comprende:

- **indice del frame**, che permette di ricavare un frame specifico dell’animazione;
- **informazioni relative alla posizione** della camera per ogni frame, espresse in coordinate ENU;
- **informazioni relative all’orientamento** della camera per ogni frame;
- **lunghezza focale** della camera per ogni frame (nel caso la prospettiva della scena cambiasse durante l’animazione);
- **dimensioni del sensore**.

Per poter essere utilizzati in Python, questi dati vengono organizzati in un dizionario, ovvero una struttura dati equivalente ad un array associativo, in cui ogni elemento è individuato dalla coppia chiave/valore. Se, ad esempio, chiamiamo “camera” il dizionario in cui salvare i dati del file JSON, le seguenti istruzioni

```
frm = camera['frame']
pos = camera['position']
rot = camera['rotation']
```

permettono di ricavare, rispettivamente, le informazioni relative all'indice del frame, alla posizione e all'orientamento della camera.

Alla chiave ‘frame’ sarà associato, ad esempio, il valore 1, ovvero l’indice che corrisponde al frame 1, alla chiave ‘position’ sarà associata la lista contenente le coordinate ENU che descrivono la posizione della camera, mentre alla chiave ‘rotation’ sarà associata la lista contenente gli angoli che ne descrivono l’orientamento.

Le proprietà intrinseche della camera, come la lunghezza focale e le dimensioni del sensore, vengono organizzate in un altro dizionario più piccolo, chiamato “cam”.

Organizzati i dati nel dizionario, si passa ad implementare la funzione *setKeyFrame*, che permetterà di automatizzare il processo di animazione della camera in Maya. Tale funzione, consente di aggiungere un keyframe in corrispondenza di un dato frame sulla timeline, il quale viene passato alla funzione come parametro di input. Tra gli altri argomenti che la funzione richiede in input vi sono la lista relativa alla posizione della camera, la lista relativa agli angoli di Pan, Tilt e Roll che descrivono l’orientamento della camera ed il dizionario contenente le proprietà intrinseche della camera. In totale, dunque, sono richiesti quattro parametri di input. Richiamando alcuni comandi di Maya, la funzione *setKeyFrame*, per un dato frame, imposta gli attributi della camera di Maya con i corrispondenti dati della camera di GES processati in Matlab®; come specificato precedentemente, tali dati sono richiesti in input dalla funzione. Dopo aver impostato gli attributi della camera, tale funzione aggiunge un keyframe sulla timeline in corrispondenza del frame considerato.

Dunque, per animare la camera in Maya, l’idea che sta alla base dello script Python è quella di invocare la funzione *setKeyFrame* per ogni frame che compone l’animazione; in questo modo, viene aggiunto un keyframe per ogni frame, ad ognuno dei quali corrispondono particolari attributi di camera. Questa procedura può essere effettuata ricorrendo ad un’istruzione di iterazione.

Eseguendo in Maya lo script realizzato in Python, dunque, si riesce a riprodurre lo stesso movimento di macchina creato in GES. Se, ad esempio, in GES si è creata un’animazione in cui la camera si muove verso destra, lo stesso movimento viene riprodotto dalla camera di Maya attraverso l’esecuzione dello script.

5.4 Ricostruire la scena di GES in Maya

Riprodotto il movimento della camera in Maya, bisogna ricostruire, o almeno in parte, il set in cui è stata girata o ambientata la scena. Questo passaggio è molto importante ai fini del matchmoving e va effettuato prima di modellare l'oggetto 3D che si vuole inserire all'interno della scena. Ricostruire la scena non solo permette di testare la qualità del matchmoving ottenuto, ma significa anche creare dei punti di riferimento che consentiranno una corretta interazione tra i modelli 3D inseriti e gli oggetti reali presenti sul set. Ad esempio, se nella clip girata si vuole inserire un folletto che salta giù da un tavolo per poi correre sul pavimento, basterà ricorrere ad un piano e ad un cubo per ricostruire il set della scena proposta; il piano farà riferimento al pavimento, mentre il cubo corrisponderà al tavolo su cui salterà sopra il folletto. Dunque, ciò che si vuole ottenere in questa fase è un set virtuale che sia ben allineato con quello reale; pertanto, si ricostruisce quella parte della geometria della scena che risulta essere più significativa ai fini di un corretto matchmoving.

Ricostruire parti del set in Maya, talvolta, può rivelarsi una procedura molto complessa da realizzare, in quanto possono non essere note le dimensioni e le proporzioni degli oggetti reali. Una soluzione a tale problema consiste nel ricorrere ad informazioni aggiuntive raccolte sul set reale; tali informazioni vengono fornite misurando le dimensioni di quegli oggetti che dovranno interagire con i modelli 3D, le dimensioni e le posizioni degli elementi che si trovano sul set, le dimensioni delle stanze in cui è ambientata la scena se la ripresa viene girata all'interno, l'altezza di eventuali attori, l'altezza dei muri della stanza e la loro relativa distanza. Si cerca, dunque, di raccogliere tutte quelle misure che possono essere utili e significative ai fini della ricostruzione del set. In questo modo, quindi, è possibile conoscere le proporzioni dei vari oggetti che compongono la scena; ciò consente di modellare gli oggetti o personaggi in CGI con le giuste proporzioni.

In questa tesi, proponiamo clip in cui modelli 3D di astronavi e mongolfiere atterrano su alcune zone di Roma (Fig.5.13a). Dunque, gli unici elementi della scena che interagiscono con gli oggetti realizzati in CGI sono quelle superfici su cui tali oggetti proiettano la loro ombra. Pertanto, la ricostruzione del set prevede solo la modellazione di tali superfici.

Nel nostro caso, però, la scena viene girata interamente nell'ambiente virtuale di GES; pertanto, non è possibile effettuare misure dirette sul set reale per conoscere le dimensioni degli oggetti. Ad ogni modo, è possibile ricavare le informazioni spaziali degli elementi situati nella scena impostando dei trackpoint (paragrafo 4.1).

Dunque, per ricostruire tali superfici in Maya, si impostano nella scena di GES dei trackpoint su quei punti significativi che consentono di modellare la superficie in questione (Fig.5.13b); in questo modo, pertanto, si ricavano le informazioni spaziali in coordinate ECEF dei punti considerati. Convertendo tali coordinate in ENU, è possibile importare questi punti in Maya, modellando la relativa superficie con le corrette proporzioni (Fig.5.13c). Si segue, dunque, un procedimento analogo a quello descritto nel paragrafo 5.2.4 per la stima della lunghezza focale della camera. Questa parte verrà trattata in dettaglio nel capitolo 6, relativo alle clip realizzate per testare la qualità del matchmoving ottenuto applicando la tecnica proposta in questa tesi.

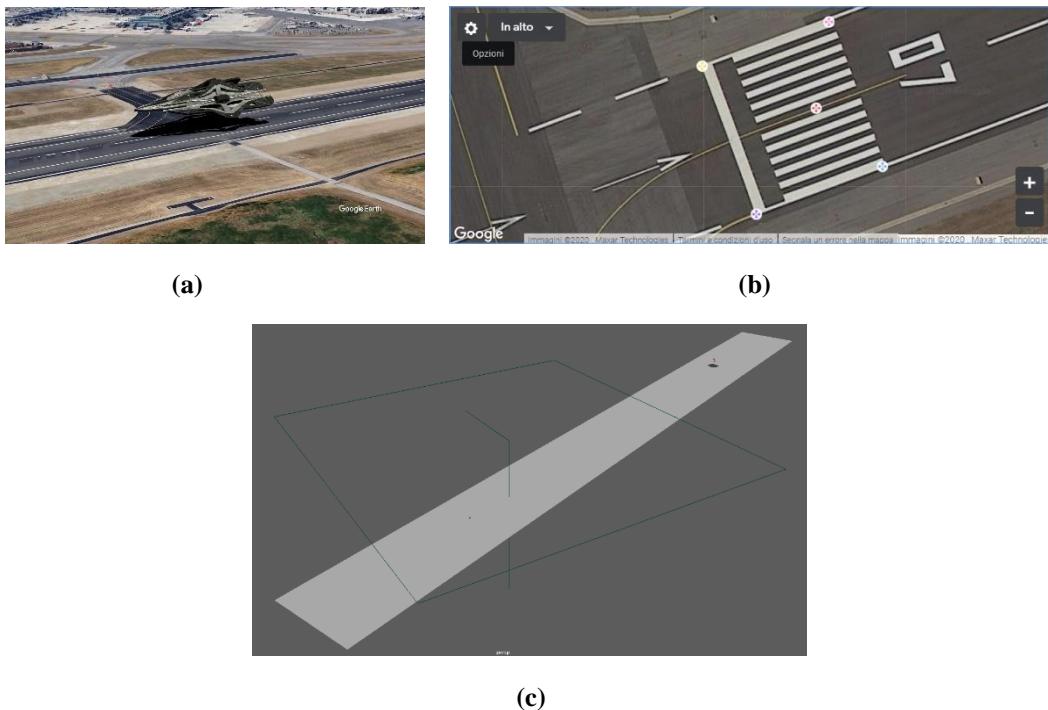


Figura 5.13. (a) Modello 3D di un’astronave atterrata sulla pista dell’Aeroporto di Fiumicino. (b) Trackpoint posizionati in GES per ricavare la posizione spaziale e le dimensioni della pista su cui atterra l’astronave. (c) La pista modellata in Maya, sfruttando le informazioni ricavate dai trackpoint impostati.

Quando si effettua la conversione da coordinate ECEF a coordinate ENU, quest’ultime vengono misurate in metri. Per riprodurre in Maya le dimensioni degli oggetti presenti nella scena in maniera corretta, una buona politica è quella di far corrispondere ogni unità in Maya ad un metro reale. Se, ad esempio, la posizione di un oggetto espressa dalle coordinate ENU

$$\begin{cases} \mathbf{x} = -6.0 \text{ m} \\ \mathbf{y} = 0.0 \text{ m} \\ \mathbf{z} = 2.5 \text{ m} \end{cases} \quad (5.4.1)$$

rispetto all’origine locale di tale sistema, associando un metro ad una unità in Maya, possiamo esprimere la stessa posizione attraverso le coordinate

$$\begin{cases} \mathbf{x} = -6.0 \\ \mathbf{y} = 0.0 \\ \mathbf{z} = 2.5 \end{cases} \quad (5.4.2)$$

Ricostruita la geometria della scena, possiamo passare all’ultima fase della tecnica proposta in questa tesi. Tale fase prevede l’inserimento di un oggetto in CG nel set ricostruito.

5.5 Matchamation

La fase in cui si modella e si anima un oggetto o un personaggio in CGI da inserire nella clip girata è chiamata matchamation. Questa è la fase finale prevista dalla pipeline del matchmoving e consiste in un processo in cui le tecniche di matchmoving, animazione, rotoscoping vengono fuse insieme (da cui il termine matchamation), al fine di creare un risultato finale caratterizzato da un elevato livello di realismo [5]. Ciò che si vuole ottenere, dunque, è un oggetto realizzato e animato in CGI che sia correttamente inserito nella scena in modo tale da risultare parte integrante di quest'ultima.

Per poter ottenere questo risultato, la fase di matchamation richiede alcuni prerequisiti.

Innanzitutto, si deve disporre dell'animazione della camera virtuale, la quale deve riprodurre fedelmente lo stesso movimento della camera reale con cui è stata girata la clip originale (paragrafo 5.3). Un'altra prerogativa necessaria nella fase di matchamation è quella di avere una ricostruzione 3D del set (paragrafo 5.4) in cui muovere la camera virtuale. Soddisfatti questi due prerequisiti, si passa a modellare l'oggetto o il personaggio creato in CGI che si vuole inserire (Fig.5.14a).

In questa tesi, verranno inseriti elementi dinamici in CGI. Dunque, per poter conferire un maggiore realismo al risultato finale, occorre animare il movimento di tali oggetti all'interno della scena. Per realizzare tale animazione, occorre sapere le posizioni iniziali e finali che saranno occupate dall'oggetto, in quali frame tale oggetto sarà visibile e quale sarà il suo moto nel corso dell'animazione. Ad esempio, se vogliamo realizzare una clip in cui un'astronave atterra sulle piste dell'aeroporto di Fiumicino, possiamo individuare due posizioni principali:

- **la posizione iniziale:** l'astronave è in volo sopra la pista; per aumentare la realtà dell'azione, si può pensare di inclinare leggermente l'astronave durante la fase di atterraggio (Fig.5.14b);
- **posizione finale:** l'astronave è a terra, ferma sulla pista (Fig.5.14c).

Tra queste due posizioni, al fine di rendere il movimento dell'oggetto più fluido, si può decidere di introdurre una posizione intermedia, in cui, ad esempio, l'astronave sia atterrata ma risulta ancora in movimento sulla pista e non più inclinata.

Come mostrato nell'esempio, dunque, è importante scegliere posizioni ben definite per le posizioni iniziali e finali, al fine di realizzare un movimento dell'oggetto che sia il più naturale e realistico possibile. Le posizioni intermedie, invece, verranno calcolate per interpolazione.

Un altro aspetto che deve essere preso in considerazione in fase di matchamation riguarda le ombre che gli oggetti realizzati in CGI proiettano sugli elementi della scena. Può accadere, talvolta, che l'illuminazione della scena cambi nel corso della clip; si pensi, ad esempio, a scene costituite da time-laps o scene in cui viene ripreso il passaggio di transizione tra notte e tramonto. A prescindere dal fatto che le condizioni di luce della scena evolvano o rimangano statiche nel corso dell'animazione, si deve tenere conto del modo in cui gli oggetti della scena sono illuminati.

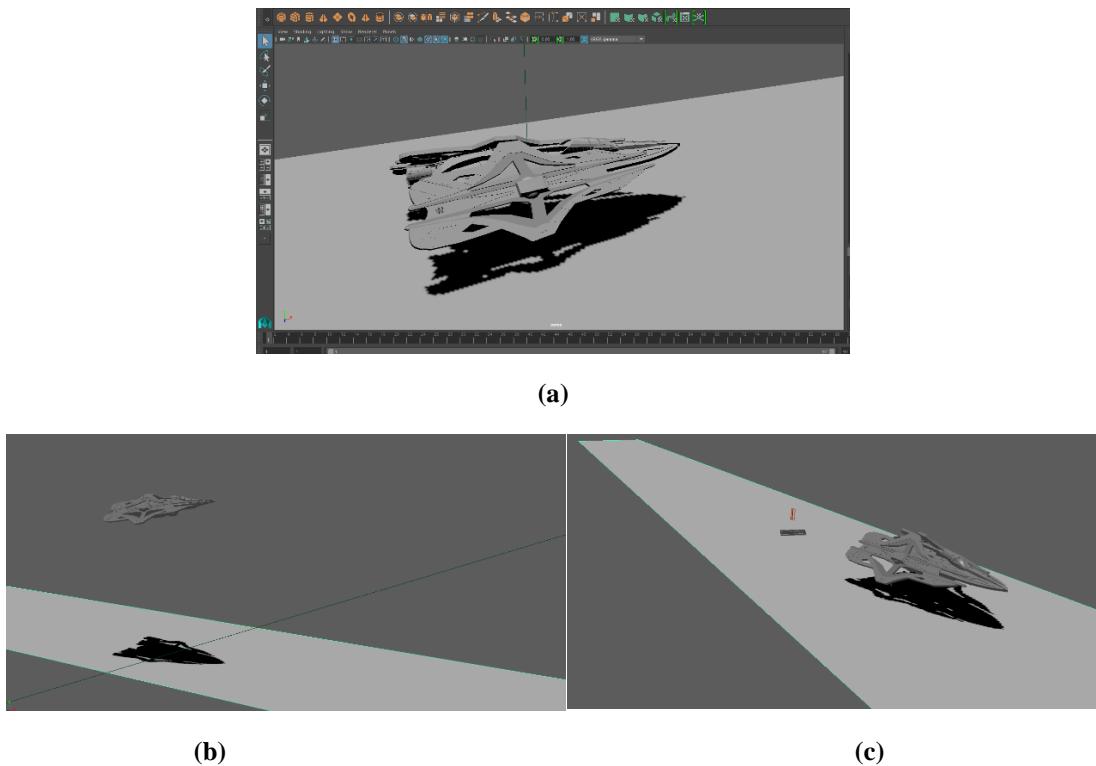


Figura 5.14. (a) Il modello 3D viene inserito nel set virtuale ricostruito in Maya. (b) Posizione iniziale dell’astronave. (c) Posizione finale dell’astronave.

Infatti, per ottenere un risultato sufficientemente realistico nell’immagine finale, tali ombre devono essere proiettate nello stesso modo in cui sono proiettate le ombre nella scena originale. Ciò significa ricostruire l’esatta posizione e l’esatto orientamento della sorgente di luce diretta che illumina la scena. In questa tesi, vengono realizzate clip in cui la scena è ripresa all’esterno, in uno specifico momento del giorno. Dunque, l’illuminazione degli oggetti può essere ricreata in Maya attraverso una luce direzionale. La luce direzionale è un tipo di luce che consente di simulare l’illuminazione che proviene da sorgenti lontane, come ad esempio il Sole. Per riprodurre l’illuminazione presente nella clip, si cerca di orientare la luce direzionale nel modo più coerente possibile alla direzione dei raggi solari nella clip originale. Ad esempio, in Figura 5.15a, osservando l’ombra proiettata dalla parte superiore dello Stadio Olimpico, si può notare come la sorgente di luce provenga da destra. Dunque, in Maya si orienta la luce direzionale, in modo tale per cui l’ombra proiettata dalla mongolfiera sia conforme alle ombre presenti nella scena (Fig.5.15b). Un’errata simulazione dell’illuminazione produce un risultato in cui la coerenza tra gli oggetti in CG e gli oggetti della scena viene a mancare, annullando, quindi quella sensazione di realismo che si voleva trasmettere (Fig.5.15c).

Una volta ricostruito il set, riprodotta correttamente l’illuminazione della scena e modellato e animato l’oggetto 3D da inserire, affinché tale oggetto appaia nella clip finale, bisogna fare in modo che la clip girata in GES funga da sfondo alla scena modellata in Maya. A tal proposito, si può associare la clip in questione all’image plane della camera virtuale che riproduce il movimento della camera reale.

Con queste considerazioni, si conclude l'ultima fase della pipeline prevista dalla procedura descritta in questo capitolo. Dunque, come risultato finale, si ottengono delle clip video in cui è possibile inserire elementi 3D, sfruttando la tecnica di matchmoving senza ricorrere, però, a tecniche di tracking.

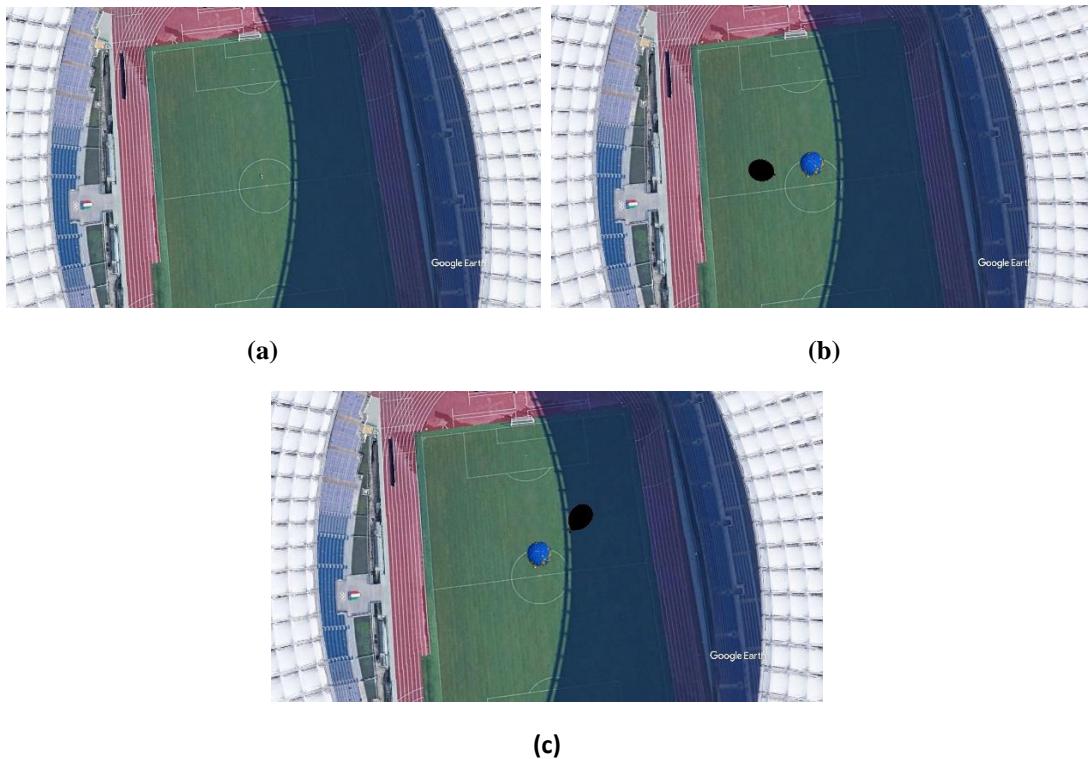


Figura 5.15. (a) Osservando l'ombra proiettata dallo Stadio Olimpico sul campo è possibile capire quale sia la direzione dei raggi solari, per poter riprodurre l'illuminazione della scena. (b) L'illuminazione della scena riprodotta con una luce direzionale per simulare la luce del Sole. L'ombra proiettata dalla mongolfiera è conforme alle ombre presenti nella scena. (c) Un esempio di illuminazione della scena errata. L'ombra proiettata dalla mongolfiera non è conforme alle ombre presenti nella scena.

Capitolo 6

Test e Valutazioni

In questo capitolo verranno analizzate alcune delle clip video realizzate al fine di testare la qualità e la correttezza del matchmoving ottenuto applicando la tecnica proposta in questa tesi.

Ricostruita la geometria della scena, la fase successiva nel workflow per il matchmoving prevede alcuni test che consentano di controllare la qualità del risultato ottenuto. Questi test vengono effettuati inserendo dei modelli 3D all'interno della scena, per verificare se questi siano coerenti agli altri oggetti presenti in essa, in termini di proporzioni, posizioni e prospettiva. Testare un matchmoving significa anche assicurarsi che il movimento della camera sia stato riprodotto correttamente in fase di calibrazione e indagare se gli elementi in CGI inseriti siano soggetti ad eventuali slittamenti, dovuti ad errori di calcolo o di altra natura.

In questa tesi, è stata proposta una procedura per matchmoving non basato su tecniche di tracking, sfruttando i dati forniti da GES. In questo caso, la qualità del risultato ottenuto viene testata realizzando clip video di breve durata, in cui vengono inseriti degli elementi in CGI ed analizzando il modo in cui tali elementi interagiscono con alcune parti della scena. Se, ad esempio, si inserisce un oggetto statico, vogliamo che questo risulti fisso al suolo per tutta la durata della clip.

Come specificato nei capitoli precedenti, al fine di ridurre le tempistiche di rendering sia in GES che in Maya, si è scelto di realizzare animazioni costituite da 96 frame con un frame rate di 24fps, corrispondenti alla durata di 4 secondi. Per controllare la correttezza delle posizioni e la presenza di eventuali slittamenti dei modelli 3D inseriti nella clip, si è deciso di riprendere la scena a basse quote in GES, in modo da rendere ben visibile il terreno. Inoltre, per poter fare una verifica più accurata del risultato ottenuto, si è deciso di allungare la durata della clip da 4 a 12 secondi.

Tuttavia, tutte queste decisioni andranno a limitare l'aspetto realistico della clip finale; infatti, la qualità dei terreni di GES a basse quote non è delle migliori. Pertanto, per ottenere risultati quasi fotorealistici, è consigliabile effettuare riprese ad alte quote. Ma ciò non rientra nell'intento di questa tesi, in quanto il vero obiettivo è quello di dimostrare la possibilità di ottenere un matchmoving abbastanza corretto attraverso la tecnica proposta e, soprattutto, senza dover ricorrere alle onerose procedure di tracking.

La metodologia adottata per realizzare un clip che consenta di testare la qualità del matchmoving ottenuto prevede i seguenti passaggi:

- filmare la scena in GES;
- esportare la sequenza video ed i dati relativi alla scena in un file JSON;
- eseguire lo script Python per poter riprodurre il movimento della camera calibrata in Maya

- ricostruire le superfici della scena che dovranno interagire con i modelli 3D da inserire;
- modellare e animare l'oggetto 3D che si vuole inserire nella scena;
- eseguire il batch rendering con il motore “Maya Software” per ottenere la clip finale.

Il batch rendering è una tecnica particolarmente utile per renderizzare le animazioni, in quanto consente di renderizzare scene multiple in una sola chiamata di rendering.

6.1 Via dei Fori Imperiali

Le prime clip che sono state realizzate in questa fase di test prevedevano l'inserimento di un oggetto statico all'interno della scena, per verificare se tale oggetto rimanesse ancorato al suolo per tutta la durata dell'animazione o se fosse soggetto ad eventuali slittamenti. Si è pensato, dunque, di proporre un contesto surreale, inserendo al centro di Roma, precisamente, in Via dei Fori Imperiali, il modello 3D della Statua della Libertà.

A tal scopo, si è deciso di iniziare questi test con una semplice animazione da realizzare in GES, in cui siano presenti solo movimenti di macchina lineari. Gli angoli di Pan, Tilt e Roll, dunque, sono tutti impostati a 0°, ottenendo una camera che guarda in direzione perpendicolare al suolo, e che, allo stesso tempo, si sposta lungo “Via dei Fori Imperiali”. È stato impostato un trackpoint sul marciapiede posto di fronte alla basilica di Massenzio; in questo modo è possibile estrapolare le informazioni spaziali relative alla posizione in cui si dovrà inserire il modello 3D della Statua della Libertà.

Eseguito lo script Python, la camera sembra riprodurre correttamente il movimento di macchina realizzato in GES. Abbiamo inserito la Statua della Libertà nel punto prestabilito, poggiandola al suolo su un piano rosso; tale piano consentirà di individuare eventuali slittamenti o possibili incongruenze con il resto della scena. Il batch rendering produce un'animazione in cui la Statua della Libertà sembra rimanere ferma al suolo nel punto previsto; osservando attentamente il piano rosso, però, si può notare un piccolo slittamento dell'oggetto 3D nel momento in cui la camera sorvola la Statua, superandola.

La clip prodotta suggerisce una parziale correttezza di questo matchmoving; le proporzioni e la prospettiva della statua sono coerenti con quelle degli oggetti che compaiono nella scena, motivo per il quale si può considerare accettabile questo risultato ottenuto. Tuttavia, il lieve slittamento che si percepisce può essere ricondotto a qualche errore nel calcolo delle posizioni della camera o alla differenza di altitudini presenti nella scena. Ad ogni modo, il matchmoving ottenuto non può considerarsi completamente corretto.

La Figura 6.1 mostra alcuni frame della clip realizzata, in cui è possibile notare lo slittamento descritto.





Figura 6.1. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. In questi frame si può notare il lieve slittamento in avanti del piano rosso.

6.2 Piazza Venezia

Proviamo a testare nuovamente la procedura proposta, scegliendo un altro scorci di Roma. Ambientiamo, dunque, la nostra scena a Piazza Venezia. Questa volta, introduciamo un orientamento della camera differente da quello di default, effettuando un Tilt di 90° ed un Pan fino ad avere il Colosseo sullo sfondo. Impostiamo un trackpoint tra la fine di “Via dei Fori Imperiali” e l’inizio di Piazza Venezia, determinando, quindi, il punto in cui vogliamo inserire la Statua della Libertà. Anche in questo caso, l’animazione realizzata in GES prevede soltanto una traslazione della camera, questa volta, però, osservando la scena da un punto di vista diverso.

Il batch rendering della scena produce un’animazione in cui la Statua della Libertà rimane ferma al suolo.

Anche in questo caso, prospettiva e proporzioni dell’oggetto inserito sembrano coincidere con quelle degli altri elementi presenti nella scena. Osservando attentamente il piano rosso nei frame proposti in Figura 6.2, non si scorgono particolari slittamenti evidenti. Dunque, in base a alle considerazioni fatte, possiamo considerare corretto il matchmoving ottenuto in questa clip.

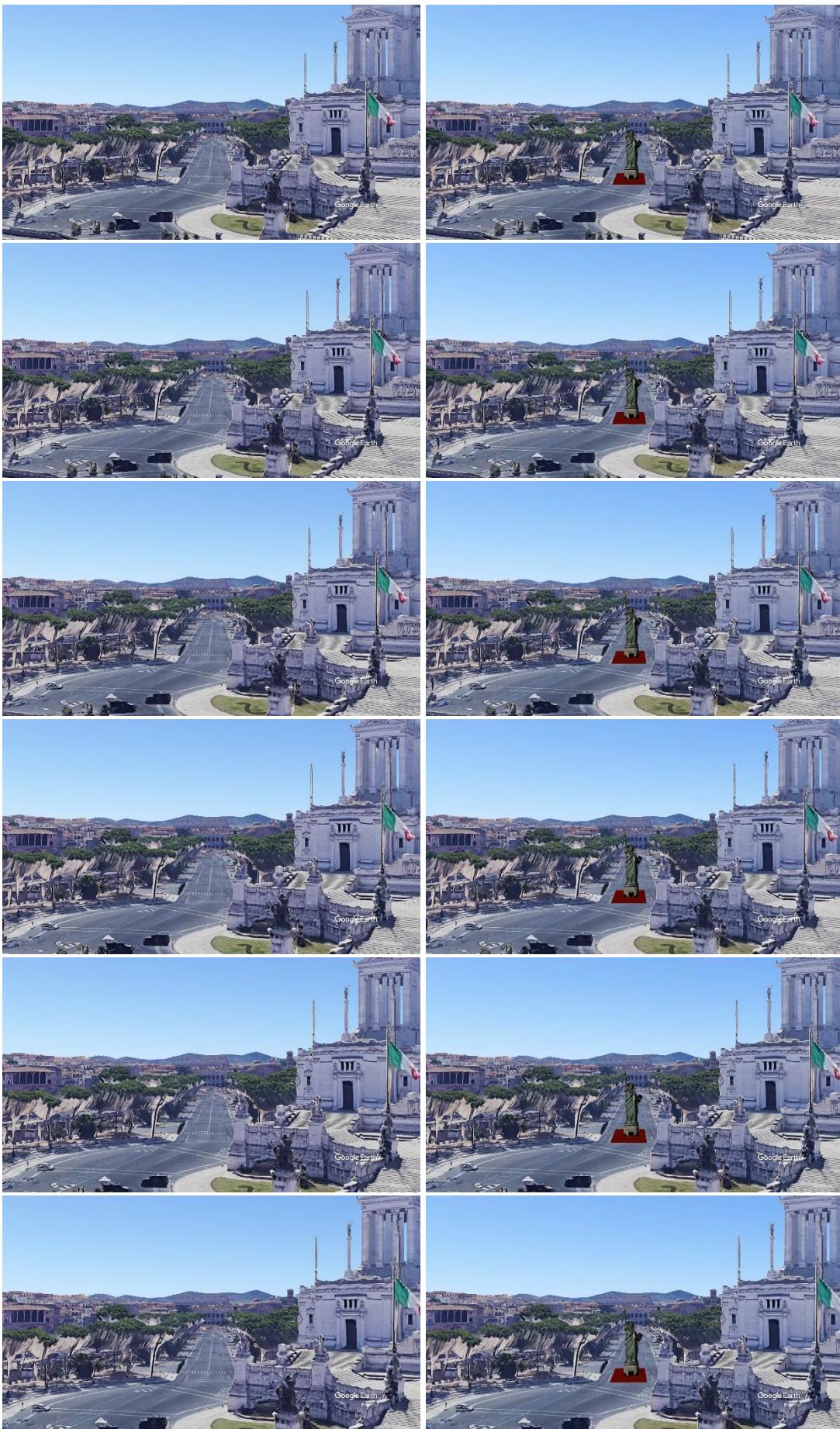




Figura 6.2. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. Osservando attentamente il piano rosso si può notare che la Statua della Libertà rimane fissa a terra.

6.3 Molgolfiere allo Stadio Olimpico

Per effettuare un test che possa verificare il funzionamento della tecnica proposta in questa tesi nel modo più accurato possibile, sceglieremo di ambientare le prossime animazioni in luoghi pianeggianti, in cui le differenze di altitudine siano poco rilevanti. Nelle prossime clip, verranno inseriti modelli 3D animati. In questo paragrafo saranno trattate alcune animazioni rappresentanti una mongolfiera in procinto di atterrare all'interno dello Stadio Olimpico. Anche in questo caso, cominceremo col realizzare semplici animazioni con movimenti lineari, per poi passare animazioni più complesse, che prevedono l'uso di movimenti di macchina più articolati.

Prima di commentare i risultati ottenuti, delineiamo brevemente la procedura con cui è stata ricostruita la superficie dello Stadio Olimpico. Sono stati impostati dei trackpoint nel progetto di GES per ricavare informazioni relative alle dimensioni del terreno, alle dimensioni delle gradinate, alle dimensioni della copertura, all'altezza dello Stadio (Fig.6.3a). In questo modo, è stato possibile modellare in Maya la struttura circolare dello Stadio Olimpico con le giuste proporzioni (Fig.6.3b). Ricostruita la scena virtuale, abbiamo animato la mongolfiera in modo da farla atterrare all'interno dello Stadio, seguendo una discesa verticale in direzione perpendicolare al suolo.

6.3.1 Clip Test 1

Per questa clip è stato previsto in GES un semplice movimento di macchina verticale: si è pensato di ottenere, come risultato finale, un'animazione in cui la camera fosse ancorata alla mongolfiera durante il suo tragitto di discesa all'interno dello Stadio.

Il batch rendering della scena produce come risultato una clip video in cui la mongolfiera non è soggetta ad alcun tipo di slittamento. Come si può notare nei frame mostrati in Fig.6.4, vi è una corretta interazione tra il modello 3D della mongolfiera e gli oggetti che compongono la scena. Infatti, verso la fine dell'animazione, è possibile notare la correttezza del modo in cui la mongolfiera viene occlusa dai pali che compongono la porta del campo da calcio. Inoltre, dai frame finali della clip si può osservare come la prospettiva della mongolfiera e delle ombre che questa proietta sul campo sia coerente con quella della scena.

Dall'analisi dei frame, dunque, si può concludere che il risultato ottenuto mostra un matchmoving corretto.

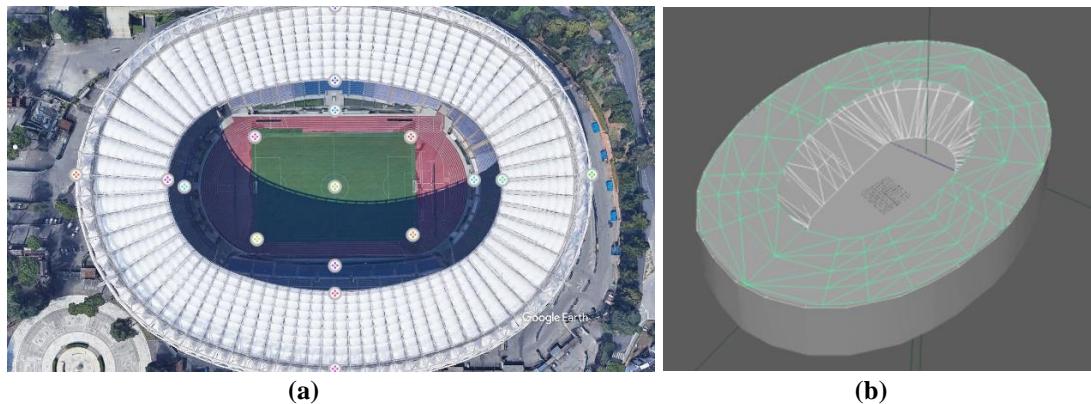
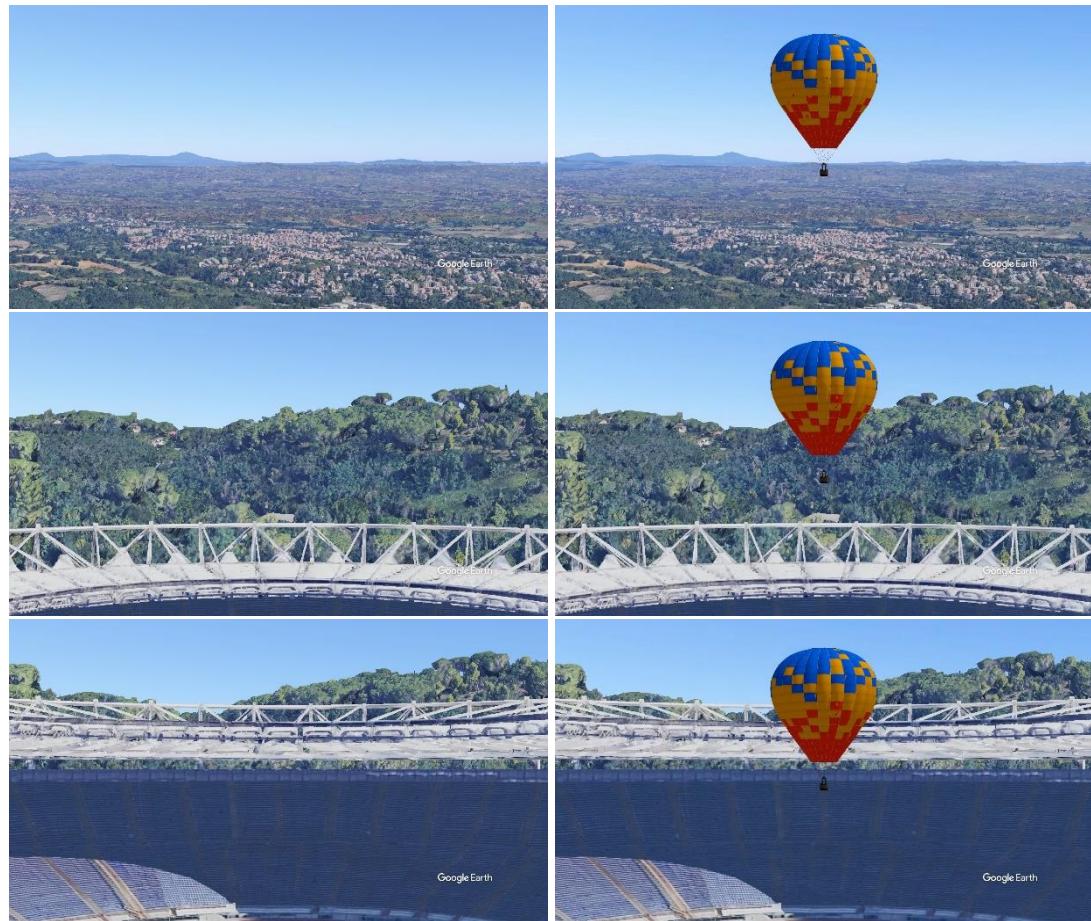


Figura 6.3. Ricostruzione dello Stadio Olimpico. (a) Trackpoint impostati in GES. (b) Lo Stadio Olimpico modellato in Maya sfruttando le informazioni spaziali in coordinate ENU ricavate dai trackpoint impostati in GES.



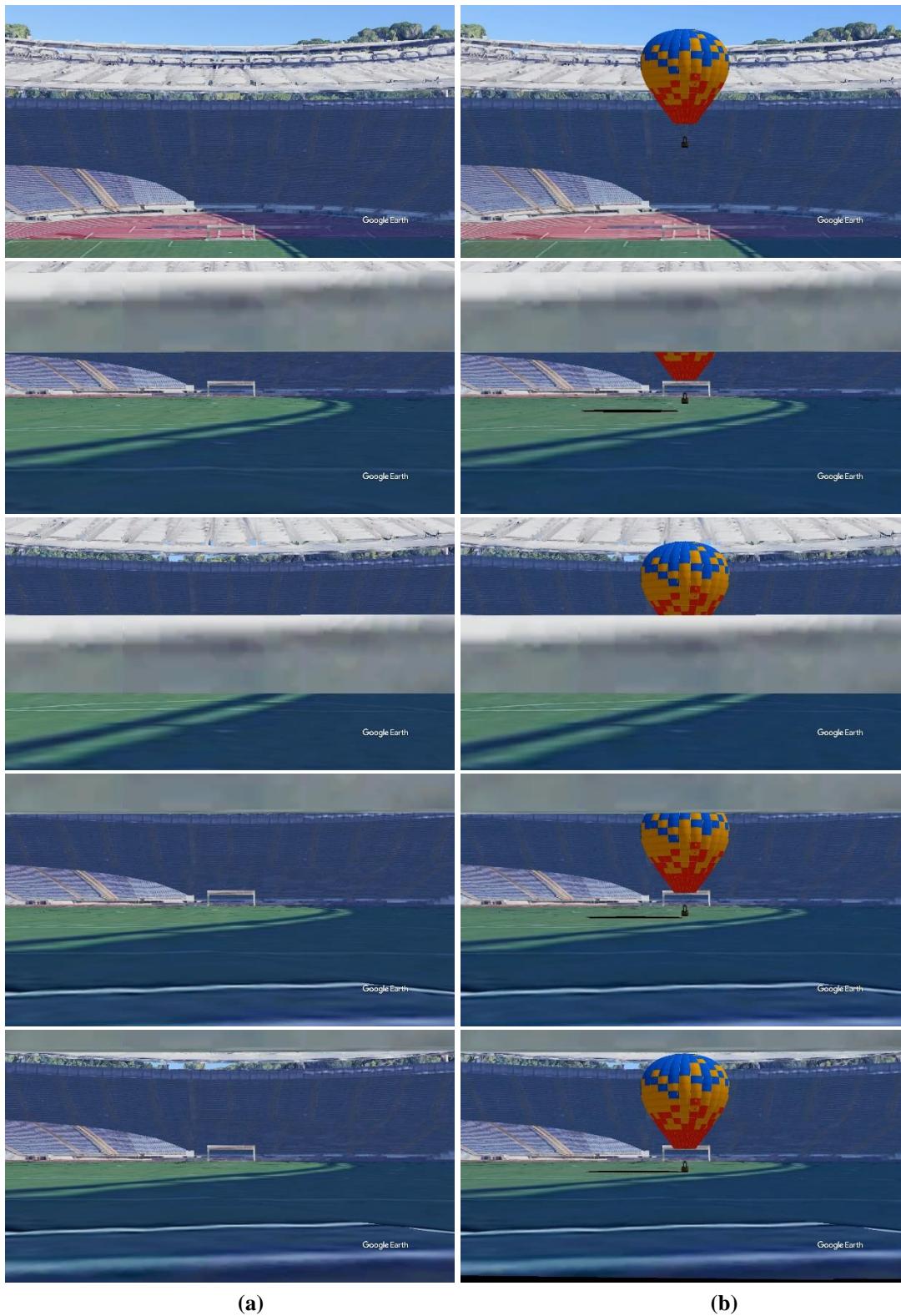


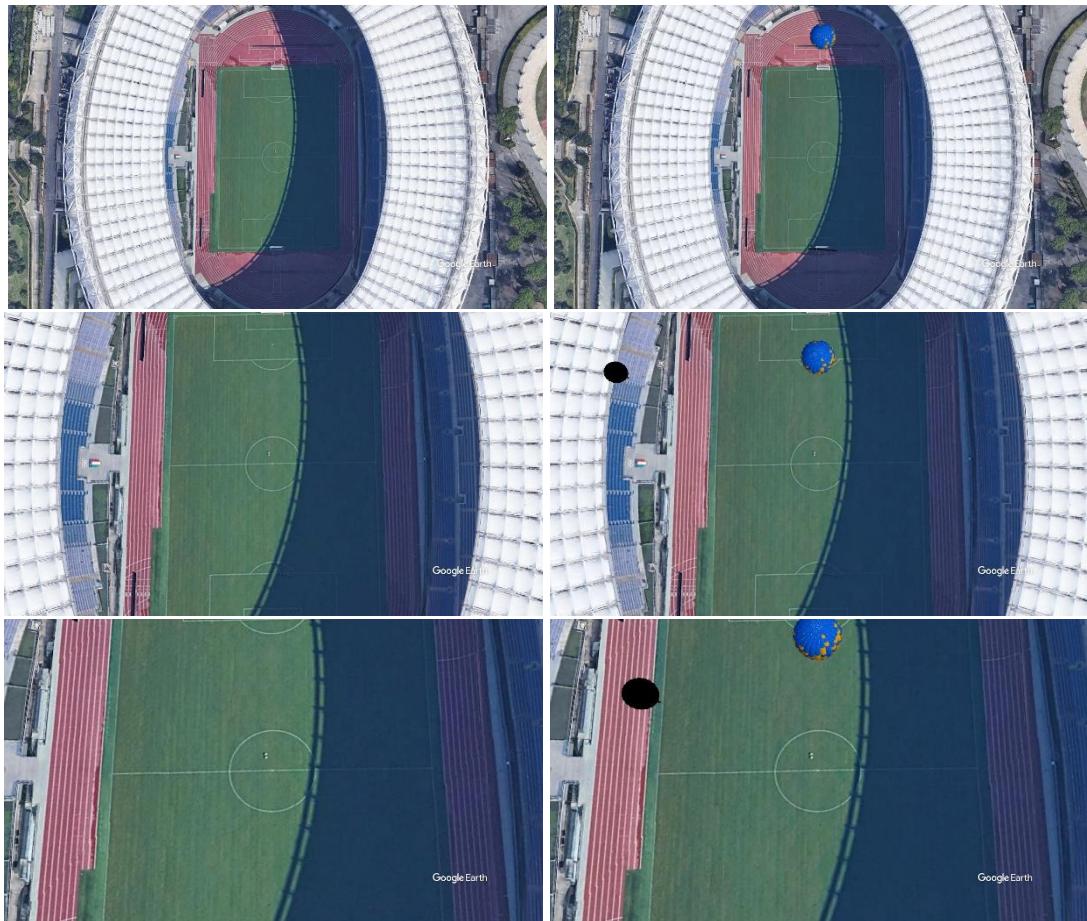
Figura 6.4. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. Nei frame finali è possibile notare la corretta occlusione della mongolfiera.

6.3.2 Clip Test 2

Cominciamo a complicare le clip da realizzare in GES. Questa volta, introdurremo un movimento di macchina che prevede un cambiamento nell'orientamento della camera nel corso dell'animazione. In particolare, vogliamo realizzare un'animazione in cui la mongolfiera viene ripresa prima dall'alto, poi frontalmente, ricorrendo ad un Tilt.

Anche in questo caso, il batch rendering della scena produce una clip in cui non sono evidenti particolari slittamenti e la mongolfiera viene occlusa in maniera corretta dalla porta del campo di calcio, nei frame finali dell'animazione. Proporzioni e prospettiva dell'oggetto 3D appaiono conformi e coerenti a quelle della scena. Pertanto, possiamo considerare corretto il risultato ottenuto in questo matchmoving.

La figura 6.5 mostra alcuni frame della clip proposta.



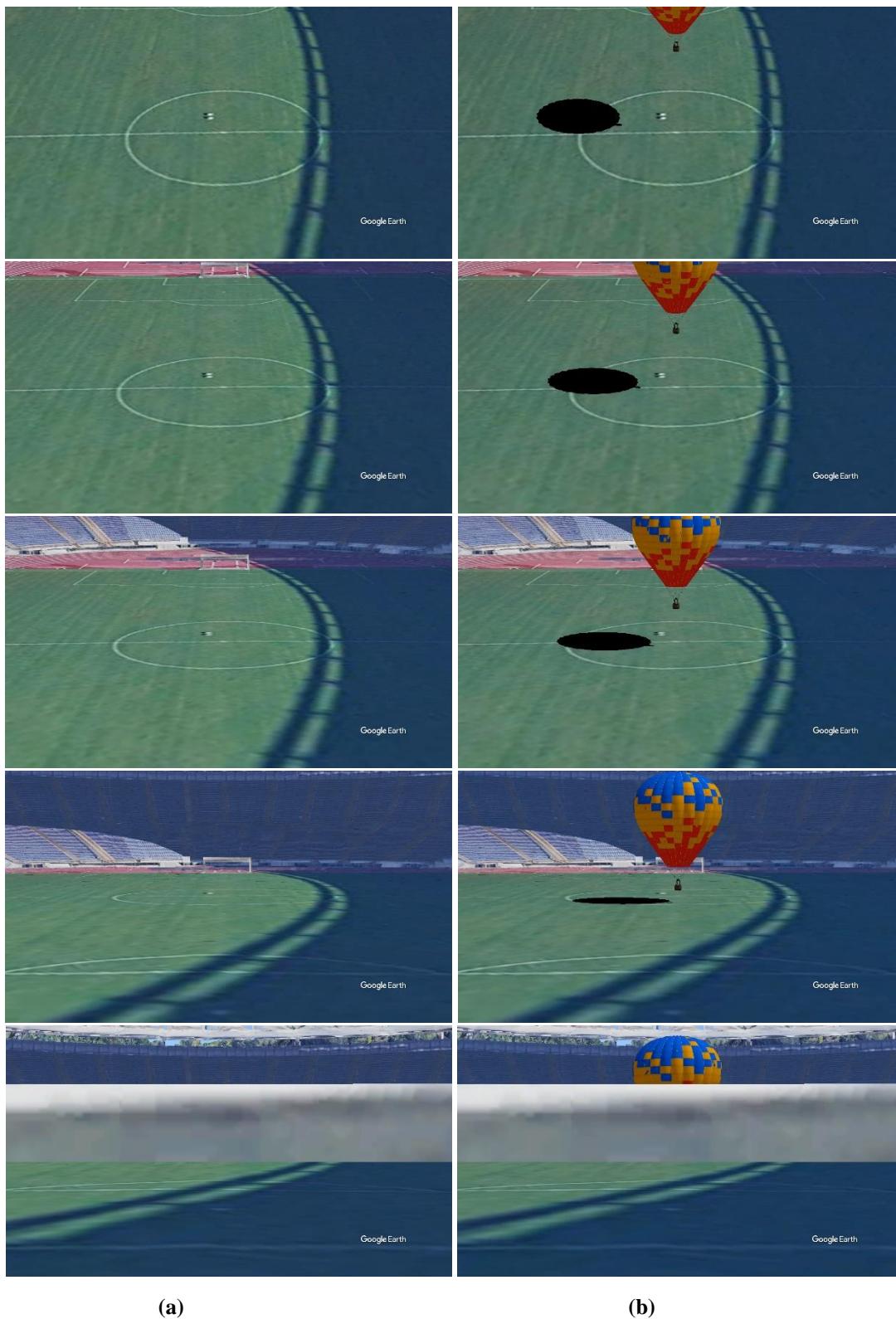


Figura 6.5. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. Nei frame finali è possibile notare la corretta occlusione della mongolfiera. Il Tilt effettuato nel corso dell'animazione non produce slittamenti.

6.4 Astronavi a Fiumicino

Per testare ulteriormente la qualità del matchmoving ottenuto attraverso la tecnica proposta in questa tesi, cambiamo ambientazione della scena, privilegiando, ancora una volta, i luoghi pianeggianti.

In questo paragrafo, verranno analizzate clip in cui sono filmate astronavi in procinto di atterrare sulle piste dell'aeroporto di Roma Fiumicino. Anche in questo caso, cominceremo col realizzare semplici animazioni con movimenti lineari, per poi passare ad animazioni più complesse, che prevedono l'uso di movimenti di macchina più articolati.

Nelle clip proposte, l'unica superficie da modellare con cui interagisce il modello 3D dell'astronave è il piano che rappresenta la pista su cui atterra il velivolo. Dunque, possiamo impostare dei trackpoint sulla pista in questione (Fig.6.6a) per ricavarne le informazioni spaziali e riprodurre il tutto in Maya (Fig.6.6b). Per rendere più realistico il movimento dell'astronave, si è pensato di inclinare l'astronave in fase di decollo e di atterraggio (Fig.6.6c).

6.4.1 Clip Test 1

Come nei casi precedenti, realizziamo una clip caratterizzata da una semplice animazione lineare, per capire come potrebbero variare i risultati in questa nuova scena proposta. In questo caso, la camera riprenderà la scena dall'alto, seguendo l'astronave durante le manovre di decollo lungo la pista. Si è scelto di riprendere la scena dall'alto con un semplice movimento di macchina affinchè si potesse verificare che l'astronave segua il percorso in linea retta tracciato dalle strisce bianche segnate sulla pista. In questo modo, risulterà facile individuare eventuali slittamenti a cui può essere soggetto il modello 3D inserito.

Analizzando i frame di Figura 6.7 prodotti dal batch rendering, si osserva che l'astronave segue correttamente il percorso previsto, senza essere soggetta ad alcun tipo di slittamento. Inoltre, la prospettiva del modello inserito risulta essere coerente con quella della scena proposta. Dunque, anche in questo caso, il risultato ottenuto applicando la tecnica di matchmoving proposta in questa tesi può considerarsi corretto.

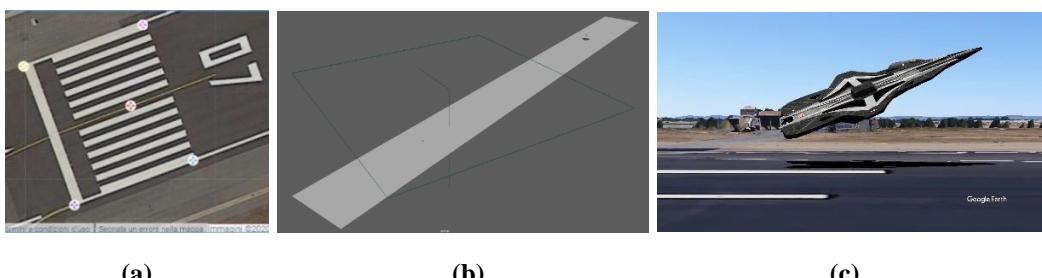


Figura 6.6. Ricostruzione della pista di atterraggio. (a) Trackpoint impostati in GES. (b) La pista di atterraggio modellata in Maya, sfruttando le informazioni spaziali in coordinate ENU ricavate dai trackpoint impostati in GES. (c) L'astronave viene inclinata in fase di decollo per rendere più realistica l'animazione.



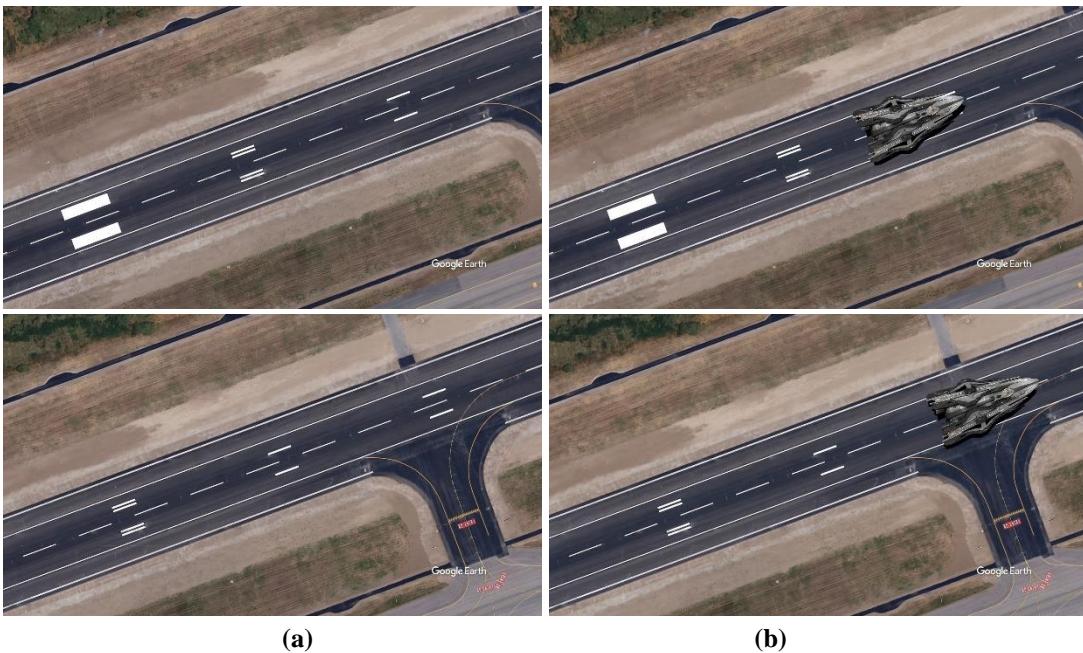


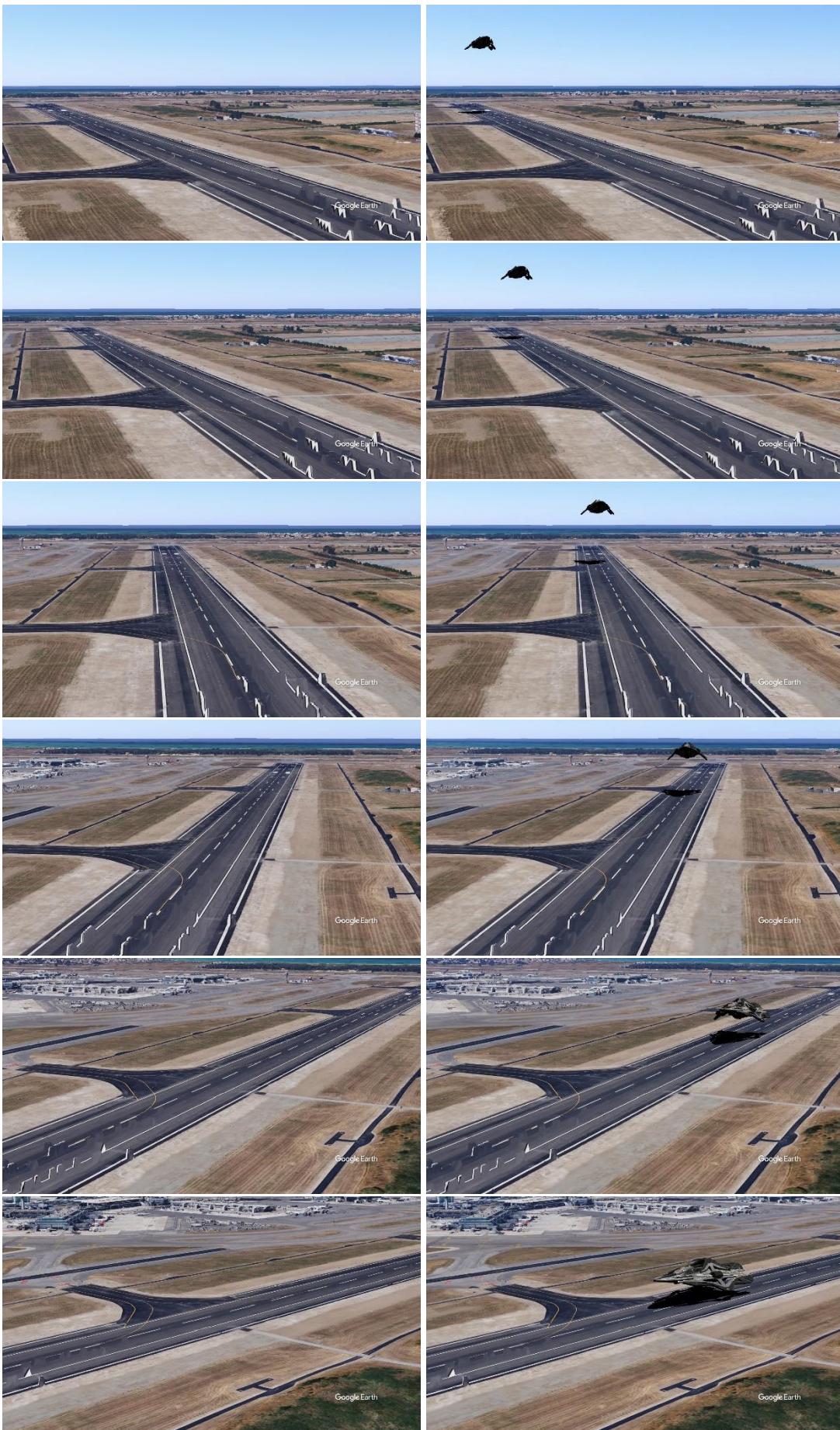
Figura 6.7. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. Si può notare come l'astronave segue correttamente il percorso tracciato dalle strisce bianche segnate sulla pista.

6.4.2 Clip Test 2

Come ultimo test riportato in questo capitolo, proponiamo una clip in cui il movimento della camera in GES segue una traiettoria circolare intorno alla pista, riprendendo il momento in cui l'astronave compie le manovre di atterraggio. Per realizzare quest'animazione, l'orientamento della camera cambia in ogni frame, al fine di mantenere fisso il suo punto di mira in corrispondenza del punto in cui l'astronave tocca il suolo. Al fine di poter verificare la qualità del risultato ottenuto nel modo più accurato possibile, si è deciso di far atterrare l'astronave nel punto che corrisponde al centro della traiettoria circolare descritta dalla camera in movimento; una volta atterrata, l'astronave prosegue il suo moto lungo la pista fino a fermarsi. In questo modo, in base a come viene proiettata l'ombra dell'astronave sulla pista, siamo in grado di stabilire se il velivolo si stia spostando in maniera corretta lungo la pista oppure se è soggetto a qualche slittamento.

Analizziamo i frame della clip, prodotta dal batch rendering. Osservando la traiettoria dell'ombra proiettata sulla pista dall'astronave dopo l'atterraggio, notiamo come questa si muova correttamente lungo le strisce bianche segnate a terra sulla pista. Questo dato ci permette di escludere la presenza di particolari slittamenti evidenti. Ancora una volta, le proporzioni e la prospettiva dell'astronave sono conformi a quelle degli oggetti presenti nella scena. Pertanto, anche in questo caso analizzato, possiamo concludere affermando che il risultato finale ottenuto è corretto.

La Figura 6.8 mostra i frame analizzati in questo paragrafo.





(a)

(b)

Figura 6.8. I frame relativi alla clip analizzata in questo paragrafo. (a) I frame della clip originale. (b) I frame della clip finale ottenuta applicando la tecnica proposta in questa tesi. Si può notare come l'ombra dell'astronave segue correttamente il percorso tracciato dalle strisce bianche segnate sulla pista.

Capitolo 7

Conclusioni

In questa tesi, è stato proposto un metodo che potesse consentire di inserire elementi 3D in clip video attraverso la tecnica di matchmoving, senza dover ricorrere alle onerose procedure di tracking. Abbiamo descritto nei capitoli precedenti, la possibilità di calibrare la camera sfruttando le informazioni fornite da GES. Ciò consente di riprodurre correttamente il movimento della camera reale in Maya evitando il tracking. Inoltre, grazie ai trackpoint è possibile ricavare le informazioni spaziali necessarie a modellare in Maya quelle superfici della scena con cui devono interagire gli elementi in CGI che si vogliono inserire. I dati ottenuti in fase di test dimostrano come questa tecnica proposta permetta di ottenere clip finali di buona qualità, in cui le proporzioni, le posizioni, la prospettiva e le ombre dei modelli 3D inseriti siano coerenti con quelle degli altri elementi presenti nella scena originale.

Pertanto, in base ai risultati finali ottenuti, possiamo concludere che la tecnica proposta in questa tesi consente di ottenere un matchmoving corretto pur non essendo basato su tracking.

7.1 Sviluppi futuri

Tra gli sviluppi futuri che si possono apportare a questo lavoro proposto vi è la possibilità di raffinare e migliorare la tecnica presentata in questa tesi, affinchè tale tecnica possa essere estesa anche al di fuori dell'ambiente di Google Earth Studio. Infatti, la tipologia di matchmoving non basata su tracking proposta potrebbe ridurre le tempistiche di calibrazione della camera, in quanto questa fase si limiterebbe ad un semplice calcolo per la conversione di coordinate geografiche.

Inoltre, è possibile sfruttare la tecnica proposta per realizzare uno script da integrare all'ambiente di Google Earth Studio, che possa convertire le coordinate ECEF in coordinate ENU, in modo da poter usare questo tipo di dati direttamente all'interno di software di modellazione 3D, come Maya o Blender.

Un aspetto che può essere migliorato nella tecnica proposta è quello relativo al modo in cui vengono ricavati gli angoli di Pan, Tilt e Roll che descrivono l'orientamento della camera. Si potrebbe pensare di trovare una strategia che consenta di semplificare tale procedura, in quanto la soluzione proposta in questa tesi risulta essere abbastanza artificiosa.

GES consente di impostare il valore dell'angolo di Pan e Roll nel range $[-360^\circ, 360^\circ]$, mentre per l'angolo di Tilt è previsto un valore compreso tra $[0^\circ, 180^\circ]$. La procedura proposta in questa tesi consente di ricavare il valore di questi angoli in maniera

abbastanza corretta. Tuttavia, il valore effettivo che si ricava per gli angoli di Pan, Tilt e Roll rientra nel range $[0^\circ, 180^\circ]$ invece di $[-180^\circ, 180^\circ]$, range previsto dalla funzione *atan2* di Matlab®. Per l'angolo di Tilt non si hanno problemi, in quanto il valore che si ricava con la funzione *atan2* per tale angolo rientra sia nel range definito da GES, che in quello previsto dalla funzione *atan2*. Per gli angoli di Pan e Roll, invece, il range previsto dalla funzione *atan2* non corrisponde a quello definito in GES; ciò significa che ricavando il valore di questi due angoli usando la funzione *atan2* di Matlab®, si perdono quegli angoli che in GES sono impostati con un valore negativo oppure maggiore di 180° .

La continuità con cui variano i valori degli angoli di Pan e Roll in ogni frame della clip consente di risolvere tale problema, apportando delle opportune modifiche al codice dello script implementato in Python; per il range di frame in cui si ottengono angoli di Pan e/o Roll con valore negativo o maggiore di 180° , si imposta una condizione per cui viene modificato il valore di tali angoli. Si consideri, ad esempio, un'animazione costituita da 96 frame. Supponiamo che a partire dal frame 50 al frame 96 l'angolo di Pan sia maggiore di 180° e l'angolo di Roll sia negativo. Per correggere tale problema, nel codice dello script verrà posta la condizione per cui, per ogni frame contenuto nel range $[50, 96]$ sia modificato il valore degli angoli di Pan e Roll nel seguente modo:

- $\text{Pan} = 360^\circ - \alpha$
- $\text{Roll} = -\gamma$

dove α, γ sono i valori degli angoli di Pan e Roll che sono stati calcolati in Matlab® e che sono contenuti nel file JSON importato in Python. Con $\alpha, \gamma \in [0^\circ, 180^\circ]$.

Risolvendo in questo modo il problema presentato, si riesce ad ottenere per gli angoli di Pan e Roll lo stesso valore impostato originariamente in GES; dunque, ciò consente di riprodurre correttamente in Maya il movimento della camera di GES. Tuttavia, questo vincolo sul calcolo degli angoli di Pan e Roll non consente di realizzare un processo di animazione della camera virtuale che sia completamente automatizzato, in quanto è necessario modificare il valore di tali angoli per quei frame in cui questi assumono valori negativi o maggiori di 180° . Pertanto, occorrerebbe trovare una soluzione che permetta di risolvere questo problema, consentendo, dunque, di poter automatizzare completamente il procedimento sopra descritto.

Infine, in questa tesi, per realizzare le clip finali è stato usato “Maya Software” come motore di rendering. Tale scelta è stata dettata dal fatto che nella versione Autodesk di Maya 2016 altri motori non consentivano di effettuare batch rendering. Allo stesso tempo, però, motori di rendering come “Mental Ray” offrono prestazioni migliori rispetto a “Maya Software” in termini di qualità.

Si potrebbe indagare, dunque, su quali motori di rendering impiegare per realizzare le clip finali, affinchè possano essere soddisfatte le seguenti condizioni:

- consentire il batch rendering;
- alta qualità del rendering della scena.

Bibliografia

- [1] Barber A., Cosker D., James O., Waine T., “*Camera Tracking in Visual Effects, An Industry Perspective of Structure from Motion*”, 2016.
- [2] Bona B., “*Rotazioni*”, DAUIN – Politecnico di Torino, 2008.
- [3] Campana I., “*Prehistoric House and 3D Reconstruction: Towards a BIM Archaeology*”, 2018.
- [4] Dispense di “*Trasformazioni Geometriche del Piano e dello Spazio*”.
- [5] Dobbert T., “*Matchmoving: The Invisible Art of Camera Tracking*”, Sybex Inc., 2005.
- [6] Etzerodt K., Jensen M. L., “*Matchmoved Architectural Visualization*”, 2011.
- [7] Gabiccini M., “*Descrizione dei moti rigidi*”, Dispense di Robotica I, 2010.
- [8] Herbez A., “*Maya Programming with Python Cookbook*”, Packt Publishing, 2016.
- [9] Huang Y., Essa I., “*Tracking multiple objects through occlusions*”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.
- [10] Jin H., Favaro P., Cipolla R., “*Visual tracking in the presence of motion blur*”, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.
- [11] Klain D., “*Lecture notes on orthogonal matrices*”, Linear Algebra II, 2004.

- [12] Klein G., Drummond T., “*Tightly integrated sensor fusion for robust visual tracking*”, in *Image and Vision Computing* 22, 10 (September), 2004.
- [13] Koks D., “*Using Rotation to Build Aerospace Coordinate System*”, DSTO Systems Sciences Laboratory, 2008.
- [14] Marini G., “*Algebra Lineare e Geometria Euclidea*”, 2013.
- [15] Möller T. A., Haines E., Hoffman N., “*Real-Time Rendering*”, Thrid Edition, CRC Press, 2008.
- [16] Pelosi F., “*Nozioni di Matlab*”.
- [17] Radke R. J., “*Computer Vision fro Visual Effects*”, Cambridge University Press, 2012.
- [18] Scateni R., Cignoni P., Montani C., Scopigno R., “*Fondamenti di Grafica Tridimensionale Interattiva*”, MCGrawHill Italia, 2005.
- [19] Shi J., Tomasi C., “*Good features to track*”, 1994.
- [20] Wei Y., Kang L., Yang B., Wu L., “*Application of Structure from Motion: a survey*”, Journal of Zhejiang University-SCIENCE C (Computers & Electronics), 2013.
- [21] Xue T., Rubinstein M., Liu C., Freeman W. T., “*A computational approach for obstruction-free photography*”, *ACM Transactions on Graphics (TOG)*, 2015.
- [22] Zhang Z., “*A flexible new technique for camera calibration*”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

Software usati:

- i) Autodesk Maya 2016
- ii) Google Earth Studio
- iii) Matlab®
- iv) Computer Vision Toolbox™, Matlab®
- v) Mapping Toolbox™, Matlab®

