Università degli studi di Padova
Facoltà di Ingegneria
Corso di laurea Magistrale in Ingegneria delle Telecomunicazioni

TESI DI LAUREA

Studio di mappe di salienza per l'ottimizzazione della ricostruzione 3D

Relatore: Prof. Calvagno Giancarlo
Correlatore: Dott. Milani Simone

Laureando: Barbiero Luca

12 Marzo 2012

Indice

Sommario

Il lavoro di ricerca presentato in questa tesi riguarda l'ottimizzazione della modellazione tridimensionale di una scena desiderata, ottenuta tramite un particolare algoritmo di *bundle adjustment*.
Con modellazione tridimensionale indichiamo il processo atto a rappresentare un qualsiasi volume tridimensionale tramite un modello virtuale generato al computer. Scene ed oggetti, caratterizzati in un modello 3D, vengono renderizzati utilizzando particolari programmi software.
Nel lavoro di ricerca, è stata utilizzata l'applicazione *Photosynth* per acquisire geometrie tridimensionali di oggetti reali. Questo programma permette la ricostruzione di una scena o di un semplice oggetto, a partire da un insieme di fotografie realizzate da utenti diversi, con punti di vista, fotocamere e in condizioni climatiche differenti.
Il lavoro svolto è stato quindi finalizzato alla ricerca di possibili metodi che permettono di ridurre il numero di fotografie necessarie per realizzare la ricostruzione della scena 3D, in modo che si possa diminuire il costo computazionale senza perdere troppo in qualità.
La scelta per ottenere quanto detto è ricaduta sulle mappe di salienza o *saliency map*. Queste mappe, simulando i meccanismi della percezione visiva umana, forniscono un'immagine a scala di grigi che descrive i particolari salienti di una scena.
È quindi plausibile pensare di sfruttarle per individuare l'oggetto più complesso da modellizzare. Una volta individuato, tramite particolari ordinamenti che analizzano i valori di salienza stimati, si possono utilizzare le fotografie del nostro dataset che meglio rappresentano l'oggetto che si vuole modellizzare. Allo stesso tempo risulta possibile "scartare" le foto che presentano una elevata informazione di salienza solo sul bordo della scena, la quale di certo non risulta fondamentale. Mediante queste considerazioni, si può utilizzare un numero inferiore di foto abbassando i costi computazionali e ottenendo allo stesso tempo una buona modellazione 3D visto che ci si concentra solo sull'oggetto principale.

Nel Capitolo 1 verrà presentato lo stato dell'arte relativo all'applicazione delle diverse tecniche computazionali per l'acquisizione della forma. In particolare focalizzeremo la nostra attenzione sui metodi ottici passivi che permettono di ottenere le proprietà geometriche di un modello tridimensionale a partire da alcune proiezioni bidimensionali.

Nel Capitolo 2 si effettuerà una breve descrizione iniziale sul funzionamento di una telecamera. L'intento è quello di capire come sia possibile invertirne il processo

di acquisizione e realizzare così un modello 3D a partire da semplici immagini digitali. Seguirà l'analisi nel dettaglio delle due tecniche principali (*Stereopsi* e *Structure from Motion*) che consentono di eseguire il procedimento appena descritto.

Nel Capitolo 3 verranno invece esposte le motivazioni dello sviluppo di Photosynth e le operazioni effettuate da questo programma per la generazione del modello tridimensionale. In particolare andremo ad analizzare l'algoritmo e i file prodotti da *Bundler*, il software di cui si avvale Photosynth per l'esecuzione della Structure from Motion.

Nel Capitolo 4 presenteremo inizialmetente l'idea che ci ha condotto ad usare le mappe di salienza per ridurre il costo computazionale del modello 3D. Seguirà la spiegazione di come tali mappe simulino la percezione visiva umana ed inoltre vedremo nel dettaglio le due tecniche sviluppate: il metodo *Itti & Koch* e il metodo *Wavelet*.

Nel Capitolo 5 analizzermo i metodi utilizzati per la realizzazione del modello tridimensionale a partire da un numero ridotto di immagini. Queste soluzioni sfruttando le mappe di salienza, selezionano in maniera ottimale le immagini da fornire in input a Photosynth.
Per ogni ordinamento descritto verranno riportati i risultati sperimentali ottenuti dalle due principali simulazioni effettuate in laboratorio. Sarà inoltre presentato un breve confronto finale tra questi metodi allo scopo di determinarne vantaggi e svantaggi.

Nel Capitolo 6 concluderemo la descrizione riportando i commenti finali e gli sviluppi futuri individuati per questa tesi.

Capitolo 1

Introduzione

La modellazione tridimensionale, termine tradotto dall'inglese *3D modeling* consiste nell'esecuzione di una procedura, la cui finalità è quella di catturare la geometria di un oggetto reale mediante l'uso di appositi macchinari. Al giorno d'oggi, tale processo è di molteplice applicabilità e spazia dall'ambito industriale, medico, odontotecnico, virtuale e per finire anche a quello dei beni culturali.
Nel campo industriale, la naturale applicazione è in tutti quei settori che necessitano di acquisire l'informazione della forma di oggetti per ottenerne una descrizione al calcolatore; nello specifico, ci riferiamo ai settori dell'ingegneria inversa (*reverse engineering*), del controllo qualità e del design. Nell'ultimo periodo, ha avuto un notevole sviluppo l'ambito che riguarda il controllo della qualità, a causa delle recenti normative legate alla certificazione del prodotto aziendale.
In campo medico invece si sfrutta la modellazione 3D per rendere più chiare le informazioni fornite da alcuni strumenti medici. Un esempio di questi è la tomografia computerizzata (TAC), la quale fornisce informazioni che risultano complicate e poco intuitive e soprattutto che richiedono una grossa esperienza per essere comprese. La possibilità di trasformarle in un geometria tridimensionale, dà la facoltà all'operatore di analizzare un qualsiasi organo come se potesse averlo in mano. In questo modo le valutazioni oltre ad essere più semplici, possono essere effettuate senza toccare il paziente.
Nel settore odontoiatrico invece, si realizzano e si studiano protesi ottenute da modelli fisici o direttamente dai pazienti. Quest'ultima operazione risulta possibile grazie alla notevole velocità di acquisizione dati raggiunta, la quale non obbliga il paziente all'immobilità prolungata.
Altri settori importanti che sfruttano la modellazione 3D sono quelli che riguardano la realtà virtuale, come l'industria dei *video games* ed il cinema. Oggigiorno infatti si richiedono prodotti sempre più realistici, i quali possono essere sviluppati a partire da modelli 3D che riproducono fedelmente oggetti, volti e ambientazioni.
Infine non possiamo non considerare il settore dei beni culturali, l'ambito che ci ha fornito l'idea ispiratrice per lo sviluppo di questa tesi. Infatti il programma Photosynth utilizzato in questa tesi per realizzare il modello 3D da ottimizzare, è stato sviluppa-

to con il fine di creare dei percorsi virtuali turistici. La possibilità di ricostruire siti archeologici, statue o edifici di interesse storico, permette ai ricercatori di analizzare queste informazioni senza la necessità del contatto, mentre ai semplici visitatori o curiosi, di osservare gli oggetti o i luoghi culturali desiderati, evocando una coinvolgente sensazione di presenza all'interno della scena.

1.1 Metodi di modellazione 3D

Attualmente sono disponibili moltissimi metodi per l'acquisizione della forma di un oggetto, una possibile classificazione [1] è descritta in Fig. 1.1. Come è possibile notare, la suddivisione principale si basa tra le tecniche che utilizzano il contatto e quelle che non lo necessitano.
I digitalizzatori a contatto consistono tipicamente in sonde che riescono a rilevare la loro posizione nello spazio. In alcuni casi sono composte da un singolo braccio snodato in grado di muoversi su un qualsiasi asse, in altre invece, sono delle vere e proprie macchine che vanno a tastare la superficie nei punti di acquisizione. Ogni volta che la sonda tocca la superficie, ne viene registrata la sua posizione. Questo tipo di digitalizzatori è molto preciso ma oltre ad essere molto costoso, è lento e richiede la presenza continua di un operatore. Inoltre il fatto stesso che la sonda debba toccare l'oggetto fa si che sia inapplicabile su superfici fragili o inconsistenti.
I digitalizzatori senza contatto invece, operano osservando le radiazioni di diversa natura provenienti dalla superficie. Queste radiazioni possono essere riflesse dall'oggetto o trasmesse dallo stesso.
Visto che, in questa tesi, le immagini digitali sono il punto iniziale dal quale partire per ottenere la descrizione della geometria tridimensionale di una scena, ci occuperemo di un particolare ramo di quest'ultima categoria, ovvero le tecniche ottiche riflessive. Queste tecniche, oltre ad avere il vantaggio di non richiedere il contatto, hanno la caratteristica di essere veloci ed economiche. D'altro canto però, per tali metodi risulta possibile acquisire solo la parte visibile delle superfici del modello ed inoltre risultano sensibili a trasparenza, brillantezza e colore.
Per la realizzazione del modello si sfrutta quindi il fatto che gli oggetti irradiano luce visibile, le cui caratteristiche dipendono da vari fattori come l'illuminazione della scena, la riflettanza (proporzione di luce incidente che una data superficie è in grado di riflettere) e la geometria della superficie. La fotocamera quindi ha la funzione di catturare questa luce e in un secondo momento, mediante il calcolatore, è possibile analizzare tutte le varie componenti per identificare la struttura 3D della scena in analisi.

1.2 Metodi ottici riflessivi

Come già detto nell'introduzione, focalizzeremo la nostra attenzione sulle tecniche ottiche riflessive, le quali sfruttano onde elettromagnetiche in alcune frequenze dello spettro luminoso. La classificazione di tali metodi si suddivide proprio per tale aspetto, infatti se queste ultime sono generate e controllate da una sorgente ausiliaria, parleremo di

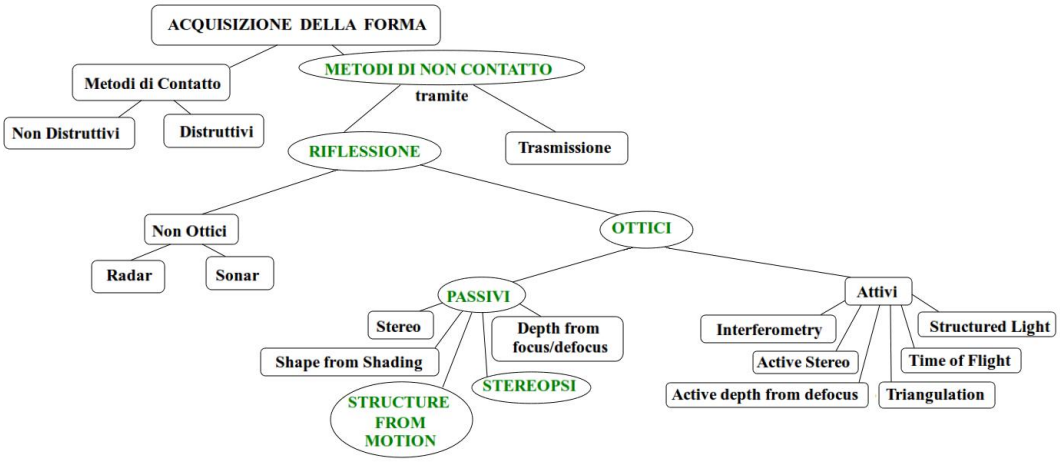


Figura 1.1: *Classificazione dei sistemi di acquisizione della forma*

metodi digitalizzatori attivi, in caso contrario di digitalizzatori passivi [2].
Vediamo ora più in dettaglio le loro caratteristiche:

- **METODI ATTIVI**
  In questi metodi la luce viene proiettata in modo strutturato (pattern luminosi, luce laser, radiazioni infrarosse, ecc) e, misurando la distorsione che essa subisce una volta riflessa si determina la forma dell'oggetto. Il modo in cui viene calcolata l'informazione di profondità determina la sottoclasse di appartenenza del digitalizzatore. Elenchiamo ora alcune delle tecniche attive utilizzabili:

  - *active defocus*
  - *stereo attivo*
  - *triangolazione attiva*
  - *interferometria*
  - *tempo di volo* (ToF)

- **METODI PASSIVI**
  Nei metodi passivi invece, non vengono utilizzate sorgenti ausiliarie bensì la luce naturale riflessa dall'oggetto. Solitamente le caratteristiche di questa sorgente sono sconosciute. Da notare quindi che queste tecniche non interagiscono in modo attivo con l'oggetto in esame nè tramite un contatto fisico, nè tramite un irraggiamento.
  L'analisi dell'oggetto può essere quindi effettuata in base a delle informazioni prelevate anche da persone che non avevano l'intenzione di digitalizzarlo, è sufficiente infatti che esso sia stato fotografato da più punti. Questa caratteristica fa si che possano essere ricostruiti oggetti di cui si possiede solo un'informazione fotografica. Inoltre, l'economicità dell'hardware (una semplice macchina fotografica) e la velocità nel prelevare i dati contraddistinguono questi metodi da tutti gli altri.

Sfortunatamente però tutte queste caratteristiche si pagano in precisione del modello e tempo di calcolo. I relativi algoritmi infatti sono molto complessi e possono impiegare ore e ore di elaborazione per un singolo oggetto. La precisione invece è strettamente legata alla risoluzione di acquisizione dell'immagine, ed allo stato attuale non raggiunge i livelli di alcuni sensori ottici attivi.
Come effettuato per i metodi attivi, elenchiamo ora alcune tecniche passive che si possono utilizzare:

- *depth from focus/defocus*
- *shape from texture*
- *shape from shading*
- *stereo fotometrico*
- ***stereopsi***
- *shape from silhoutte*
- *shape from photo-consistency*
- ***structure from motion***

Capitolo 2

Realizzazione di un modello 3D a partire da immagini digitali

Dopo aver svolto nel capitolo precedente una panoramica sulle attuali tecniche computazionali a disposizione, vedremo ora una dettagliata analisi che descrive le due tecniche principali per l'acquisizione della forma: la *Stereopsi* (Paragrafo 2.2) e la *Structure From Motion* (Paragrafo 2.3). Quest'ultimo metodo risulta anche il più interessante poiché è quello su cui si basa Photosynth, l'applicazione utilizzata in questa tesi per la realizzazione del modello 3D.
Prima però di descrivere queste due tecniche, andiamo ad analizzare in maniera del tutto generale il funzionamento di un semplice apparato per l'acquisizione di immagini, in modo tale da capire come sia possibile invertire tale processo ed ottenere così la geometria tridimensionale di una scena da semplici immagini digitali.

2.1 Geometria della formazione dell'immagine

Nicéphore Niépce dal 1816 compì numerosi tentativi per riuscire a realizzare nel 1826 la prima immagine fotografica della storia utilizzando un semplicissimo "apparecchio" che aveva costruito in modo abbastanza rudimentale chiamato camera a foro stenopeico [3] (Fig. 2.1). Tale apparecchio era conosciuto già nell'antichità e nel XVI secolo veniva usato comunemente da molti pittori per lo studio della prospettiva nei dipinti di paesaggi e vedute.
La camera a foro stenopeico è costituita da una scatola a tenuta di luce dotata su un lato di un piccolo foro (foro stenopeico), dal quale possono entrare i raggi luminosi riflessi dal soggetto che vanno a formare un'immagine invertita sul lato opposto. Normalmente il foro è chiuso con un tappo. Dentro la camera, sul lato opposto al foro, viene collocato un foglio di materiale fotosensibile (una pellicola fotografica o un foglio di carta fotografica, etc.) al buio. Dopo avere inquadrato il soggetto, il tappo viene aperto per un determinato tempo di esposizione e i raggi di luce che entrano attraverso il foro permettono di "impressionare" il foglio di materiale fotosensibile che registra così un'immagine "latente" del soggetto.

Nella camera solo pochi raggi di quelli riflessi dal soggetto riescono a passare attraverso la piccola apertura del foro, formando un'immagine comunque non molto nitida. L'immagine risulta rimpicciolita, proporzionalmente alla distanza tra foro e pellicola, e capovolta rispetto al soggetto in conseguenza del fenomeno della propagazione rettilinea della luce.

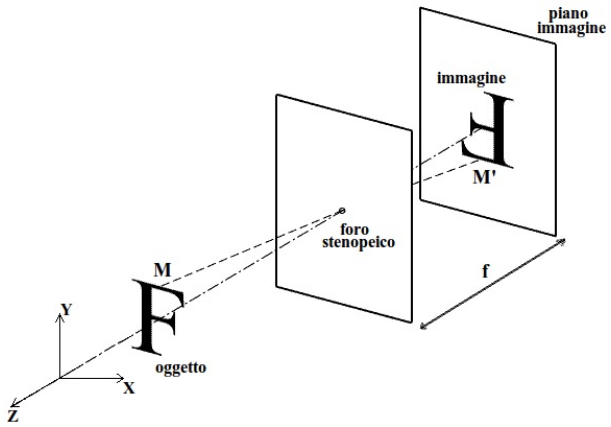


Figura 2.1: *Fotocamera stenopeica*

Vediamo ora le equazioni che definiscono il processo di formazione dell'immagine [4] che prende il nome di proiezione prospettica. Sia M un punto della scena di coordinate (X, Y, Z) e sia M' la sua proiezione sul piano immagine o retina, di coordinate (X', Y', Z'). Se f è la distanza del foro (o centro di proiezione) dal piano immagine (distanza focale) allora dalla similitudine dei triangoli si ottiene:

$$\frac{-X'}{f} = \frac{X}{Z} \quad e \quad \frac{-Y'}{f} = \frac{Y}{Z} \tag{2.1}$$

e quindi:

$$X' = \frac{-f \cdot X}{Z} \qquad Y' = \frac{-f \cdot Y}{Z} \qquad Z' = -f \tag{2.2}$$

È dimostrato quindi anche mediante le formule che l'immagine è invertita rispetto alla scena, sia destra/sinistra che sopra/sotto, come indicato dal segno negativo, inoltre c'è da considerare che la divisione per Z è responsabile dell'effetto scorcio per cui la dimensione dell'immagine di un oggetto dipende dalla distanza di questo dall'osservatore.

2.1.1 Evoluzione della camera con lenti

Nel paragrafo precedente abbiamo parlato della fotocamera stenopeica, la quale simula il funzionamento di un occhio puntiforme, ovvero un occhio senza lenti (il più evoluto è quello del mollusco Nautilus e delle lumache di mare).
Il primo problema che sorge ad un occhio puntiforme è dovuto alla diffrazione: tale

fenomeno dà origine ad una sfocatura che aumenta tanto più il foro è piccolo. Un altro problema da considerare è che affinché l'immagine sia nitida, risulta necessario che il foro sia piccolissimo, questo fatto naturalmente porta ad una riduzione della luce che passa attraverso il foro, l'oggetto allora può essere visto solo se illuminato da una luce di elevata intensità. Per risolvere tale problema occorre allargare il foro però torniamo al problema di partenza.
La camera a foro stenopeico quindi fu in seguito perfezionata con l'aggiunta di una lente frontale convergente al posto del piccolo foro, che essendo di diametro maggiore ne aumentava la luminosità così da ridurre il tempo necessario all'esposizione. La lente inoltre consentiva anche di focalizzare i raggi su uno stesso piano aumentando notevolmente la nitidezza dell'immagine. L'approssimazione che facciamo per l'ottica del sistema di acquisizione, che in generale è molto complessa essendo costituita da più lenti è quella della lente sottile (Fig. 2.2).
Le lenti sottili godono delle seguenti proprietà:

(i) I raggi paralleli all'asse ottico incidenti sulla lente vengono rifratti in modo da passare per un punto dell'asse ottico chiamato fuoco F.

(ii) I raggi che passano per il centro C della lente sono inalterati

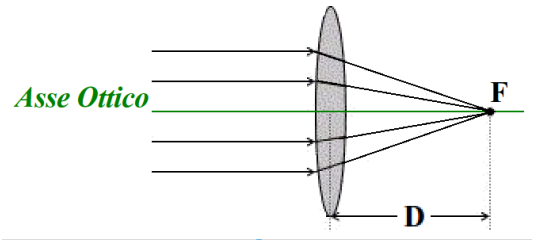


Figura 2.2: *Lente sottile*

La distanza del fuoco F dal centro della lente C prende il nome di distanza focale D. Essa dipende dai raggi di curvatura delle due superfici della lente e dall'indice di rifrazione del materiale.
Solamente i raggi focalizzati forniscono un'immagine nitida perché formano un unico punto piccolissimo.
Dato un punto della scena M è possibile costruirne graficamente l'immagine M' (Fig. 2.3) utilizzando due raggi particolari che partono da M:

- Il raggio parallelo all'asse ottico, che dopo la rifrazione passa per F.

- Il raggio che passa inalterato per C.

Con questa costruzione e grazie alla similitudine dei triangoli, si ottiene la formula dei punti coniugati (o equazione della lente sottile):

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D} \tag{2.3}$$

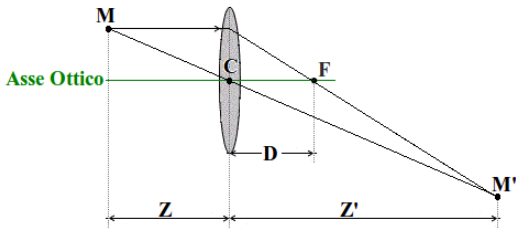


Figura 2.3: *Determinazione del punto coniugato*

L'interpretazione di questa equazione è che l'immagine di un punto della scena, distante Z dalla lente, viene "messo a fuoco"ad una distanza dalla lente Z' che dipende dalla profondità Z del punto e dalla distanza focale D della lente.

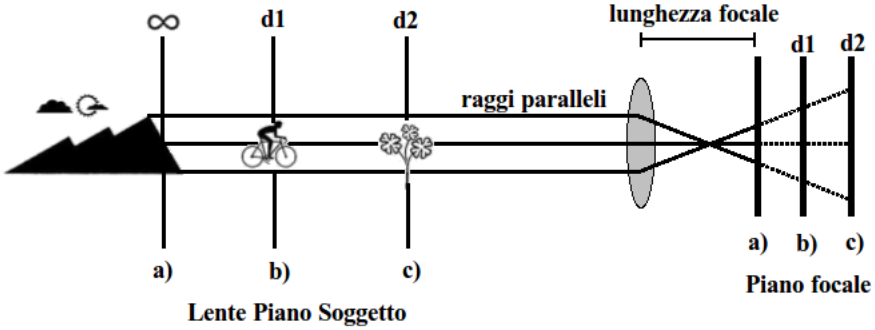


Figura 2.4: *Messa a fuoco di oggetti posti a distanze diverse*

Proprio quest'ultimo concetto è apprezzabile in Fig. 2.4, tutti i punti di un soggetto posti alla stessa distanza dalla lente formano il piano soggetto (che è bidimensionale), e si può immaginare che nello spazio si possono avere infiniti piani soggetto, paralleli tra loro e perpendicolari all'asse ottico della lente, che si avvicinano e si allontanano dalla lente. Ogni piano soggetto viene focalizzato dalla lente solo sul relativo piano focale posto dietro alla lente ad una distanza che è dettata dall'Eq. (2.3).
Ne risulta che ogni piano soggetto ha un unico piano focale dove si riproduce l'immagine del soggetto formata da punti luminosi molto piccoli e quindi sufficientemente nitida. Maggiore è la distanza del piano soggetto dalla lente, minore è la distanza dalla lente al piano focale, fino a uguagliare la lunghezza focale della lente stessa.
La messa a fuoco consiste nello spostamento della lente in avanti o indietro (traslazione lungo Z) rispetto al piano focale della fotocamera (dove si trova la pellicola o il sensore) che invece è fisso ed è vincolato dalla costruzione meccanica dell'apparecchio. La messa a fuoco corretta si raggiunge quando, per un determinato piano soggetto, il relativo piano focale dietro alla lente coincide con il piano focale della fotocamera dove si forma un immagine nitida. Se l'Eq. (2.3) non è verificata si ottiene una immagine sfocata del punto, ovvero un cerchio che prende il nome di cerchio di confusione.
Il piano immagine è coperto da elementi fotosensibili i quali hanno una dimensione piccola ma finita. Finché il cerchio di confusione non supera le dimensioni dell'elemento

fotosensibile l'immagine risulta a fuoco. Esiste quindi un range di profondità per il quale i punti sono a fuoco.

2.1.2 Immagini digitali

Come già detto in precedenza, per la realizzazione di un determinato modello 3D partiremo da semplici immagini digitali che rappresentano l'oggetto in analisi.
Risulta opportuno quindi, dopo aver descritto la fotocamera stenopeica e le lenti sottili, analizzare come avviene la formazione dell'immagine nella fotocamera digitale.
Una fotocamera digitale è composta dall'ottica, che risulta possibile approssimare mediante la coppia lente sottile e matrice di CCD (*Charge-Coupled Device*) o CMOS che costituisce il piano immagine.
Quest'ultimo può essere definito come una matrice $n \times m$ di celle rettangolari fotosensibili, ciascuna delle quali converte l'intensità della radiazione luminosa che vi incide in un potenziale elettrico.
La matrice del CCD viene convertita in una immagine digitale, ovvero in una matrice $N \times M$ (per esempio 1024×768) di valori interi (solitamente $0 \div 255$) a cui corrisponde una determinata colorazione.
Gli elementi della matrice prendono il nome di pixel (*picture element*). Indicheremo con $I(u, v)$ il valore dell'immagine nel pixel individuato dalla riga v e colonna u (tale valore rappresenta la luminosità e ad esso viene associato un colore).
La dimensione $n \times m$ della matrice CCD non è necessariamente la stessa della immagine $N \times M$ (matrice dei pixel); per questo motivo la posizione di un punto del piano immagine è diversa se misurata in elementi CCD piuttosto che pixel.

$$u_{pix} = \frac{N}{n} u_{CCD} \qquad v_{pix} = \frac{M}{m} v_{CCD} \tag{2.4}$$

Ad un pixel corrisponde quindi un'area rettangolare sul CCD array (si chiama anche impronta del pixel), non necessariamente uguale ad una cella del CCD, le cui dimensioni sono le dimensioni efficaci del pixel.

2.1.3 L'intensità luminosa in un'immagine

Dopo aver visto come avviene la formazione dell'immagine in una fotocamera digitale, analizzeremo ora quali sono i fattori che determinano l'intensità luminosa dei pixel che la compongono.
La luminosità $I(p)$ di un pixel p nell'immagine, è proporzionale alla quantità di luce che la superficie S_p centrata in un determinato punto x riflette verso la fotocamera (S_p rappresenta la superficie che si proietta nel pixel p).
Questa luminosità dipende a sua volta da due fattori, dal modo in cui la superficie riflette la luce e dalla distribuzione spaziale delle sorgenti luminose.
La grandezza che solitamente si utilizza per determinare la "quantità di luce"che viene emessa o assorbita da un punto è la radianza. La radianza $L(x, \omega)$ è la potenza della radiazione emessa (o riflessa, o trasmessa) da un punto x, da una superficie di area

unitaria, ed inoltre diretta verso un angolo solido unitario lungo la direzione ω.
Trascurando l'attenuazione dovuta all'atmosfera, si può verificare che la radianza che lascia un punto x della superficie nella direzione del pixel p coincide con la radianza che raggiunge il pixel p dalla medesima direzione.

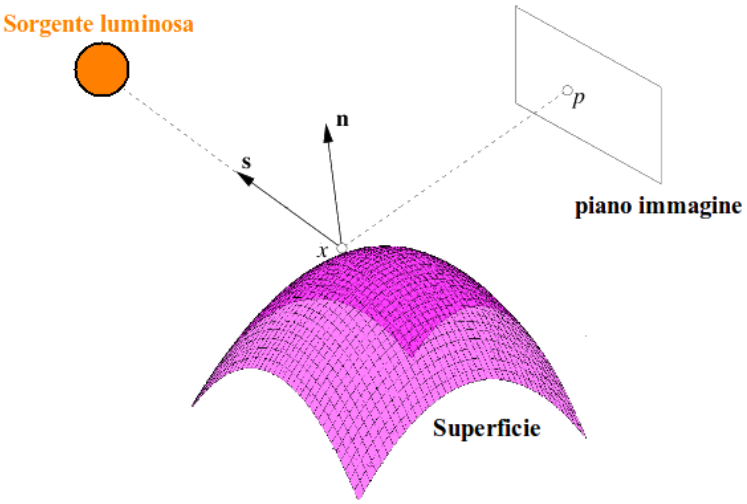


Figura 2.5: *Superficie Lambertiana*

In un modello semplificato, approssimeremo la luminosità del pixel $I(p)$ con la radianza:

$$I(p) = L(x)$$

Come precedentemente detto l'intensità luminosa di un pixel dipende dal modo in cui la superficie riflette la luce e dalla distribuzione spaziale delle sorgenti luminose.
Di particolare importanza risulta il primo fattore, essendo un fenomeno complesso che dipende dal materiale di cui è composta la superficie. La riflettanza rappresenta proprio tale proprietà, ovvero descrive il modo in cui una superficie riflette la luce incidente; analizzeremo ora i due casi estremi:

A- Diffusione
Nella diffusione la luce incidente viene assorbita e riemessa. Consideriamo come superficie quella di tipo lambertiano ovvero una superficie che appare ugualmente luminosa da ogni direzione.
La riflettanza in questo caso si può agevolmente caratterizzare, infatti la radianza $L(x)$ emessa da x segue la cosiddetta legge di Lambert:

$$L(x) = \rho(x)E(x) \tag{2.5}$$

dove:

- $E(x)$ è la irradianza in x, ovvero la potenza della radiazione luminosa per unità di area incidente nel punto x (da tutte le direzioni);

- $\rho(x)$ è l'albedo di x, che varia da 0 (nero) a 1 (bianco).

Considerando una sorgente luminosa puntiforme, si verifica che:

$$E(x) = L(x, \mathbf{s})(\mathbf{s}^T \mathbf{n}) \tag{2.6}$$

dove:

- s è la direzione sotto cui x vede la sorgente luminosa;
- n è la direzione (il versore) della normale in x (si faccia riferimento alla Fig. 2.5).

In questo caso, la legge di Lambert si scrive:

$$L(x) = \rho(x) L(x, \mathbf{s})(\mathbf{s}^T \mathbf{n}) \tag{2.7}$$

B-Riflessione speculare

Nella riflessione speculare la radianza riflessa è concentrata lungo una particolare direzione, quella per cui il raggio riflesso e quello incidente giacciono sullo stesso piano e l'angolo di riflessione è uguale all'angolo di incidenza: è il comportamento di uno specchio perfetto.
La riflettanza di una superficie speculare, diversamente da quella della superficie lambertiana, tiene conto della direzione di incidenza della luce.

2.2 La Stereopsi

Dopo aver visto come funziona l'apparato per l'acquisizione di immagini digitali e prima di affrontare il metodo ottico passivo che è alla base del programma utilizzato per lo sviluppo di questa tesi ovvero lo Structure from Motion, studiamo ora la tecnica principe: la Stereopsi [5].
La Stereopsi è la capacità percettiva che consente di unire le immagini distinte provenienti dai due occhi, che a causa del loro diverso posizionamento strutturale (50-70mm) presentano uno spostamento laterale. Tale disparità viene inoltre sfruttata dal cervello per ottenere informazioni sulla profondità e sulla posizione spaziale dell'oggetto osservato. Di conseguenza questo permette di generare la visione tridimensionale.
In questo paragrafo, dopo una breve introduzione che descriverà a grandi linee la Stereopsi computazionale, esporremo alcuni concetti base che ci saranno utili per descrivere le fasi cruciali di questa tecnica.

2.2.1 Introduzione

La Stereopsi a livello applicativo è realizzata mediante un processo che consente di ottenere informazioni sulla struttura tridimensionale di un modello a partire da due immagini digitali realizzate da fotocamere che inquadrano la scena da posizioni diverse: per quanto detto nella definizione precedente di Stereopsi, queste non possono essere molto distanti.

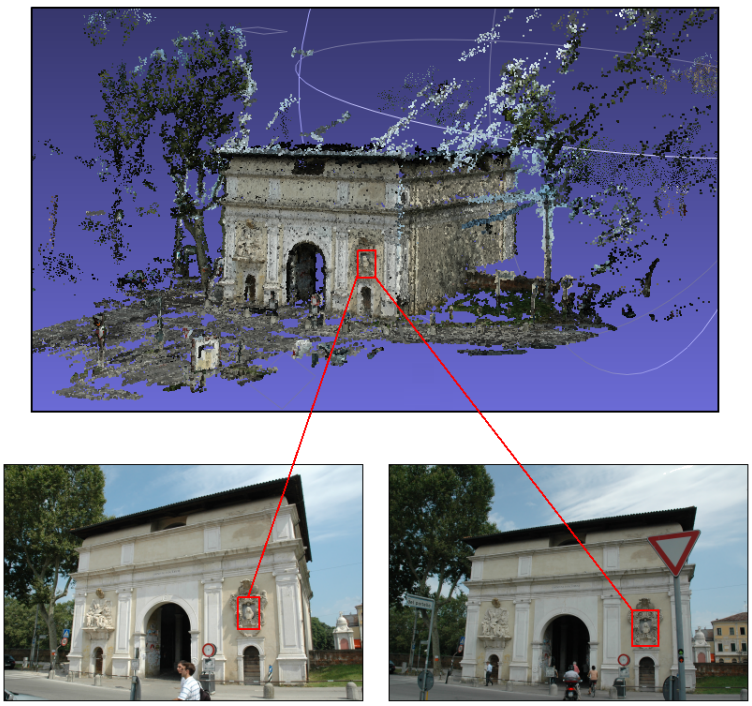


Figura 2.6: *Coppia di punti coniugati corrispondenti nel modello 3D.*

A livello computazionale questa tecnica è caratterizzata da due fasi principali:

- **Calcolo delle corrispondenze**

- **Triangolazione**

La prima fase permette di individuare i cosiddetti punti coniugati, che non sono altro che coppie di punti di due immagini distinte che raffigurano il modello in questione e sono proiezione dello stesso punto nella scena 3D (Fig. 2.6).
Da quanto detto precedentemente, cioè che le immagini che si analizzano non possono essere scattate da punti molto distanti, è facile dedurre che sia possibile individuare molte coppie di punti coniugati visto che molti particolari della scena per ovvie ragioni compariranno in entrambe le foto.
Se però non si pongono alcuni vincoli tra i quali quello fondamentale ovvero il vincolo epipolare (che analizzeremo nel paragrafo successivo), è possibile dimostrare che il processo genererebbe molti falsi accoppiamenti.
Una volta determinati gli accoppiamenti tra i punti delle due immagini è possibile passare alla seconda fase della Stereopsi, la triangolazione. Questa tecnica mediante i parametri estrinseci (posizioni reciproche delle telecamere ottenute tramite la calibrazione) ed i parametri intrinseci del sensore, ricostruisce la posizione tridimensionale nella scena dei punti coniugati delle due immagini.

2.2.2 Concetti base

Prima di affrontare i concetti di triangolazione e del calcolo delle corrispondenze approfondiamo alcuni concetti legati al modello geometrico della formazione dell'immagine [6], ovvero come risulta relazionata la posizione di un punto nella scena e la posizione del punto corrispondente nell'immagine, mediante un opportuno modello geometrico.
Il più comune modello geometrico della fotocamera è il modello stenopeico o prospettico, il quale consiste di un piano immagine R e di un punto C definito centro ottico distante f (lunghezza focale) dal piano.
La retta passante per C ortogonale a R è l'asse ottico (asse Z Fig. 2.7) e la sua intersezione con R prende il nome di punto principale.
Il piano F parallelo ad R che contiene il centro ottico prende il nome di piano focale, i punti di tale piano si proiettano all'infinito sul piano immagine.
Per descrivere in maniera semplice il modello geometrico che si sta studiando scegliamo dei sistemi di riferimento particolari che renderanno le equazioni che studieremo particolarmente semplici.
Per fare ciò introduciamo un sistema di riferimento destrorso (X, Y, Z) per lo spazio tridimensionale definito anche sistema mondo, centrato in C e con l'asse Z coincidente con l'asse ottico. Questo tipo di sistema di riferimento, il quale fissa il riferimento mondo coincidente con il riferimento standard della fotocamera, è anche conosciuto come sistema standard della fotocamera.
Introduciamo inoltre un sistema di riferimento (u, v) per il piano immagine R centrato nel punto principale e con gli assi u e v orientati come X ed Y rispettivamente, come mostrato in Fig. 2.7. Consideriamo ora un punto M di coordinate nello spazio 3D:

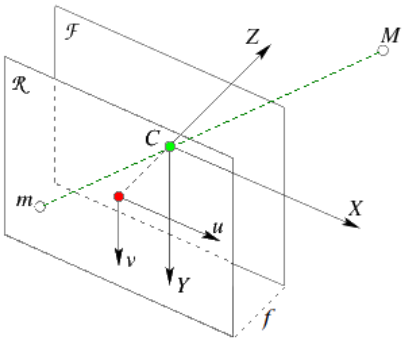


Figura 2.7: *Modello geometrico della fotocamera.*

$$\tilde{\mathbf{M}} = [x, y, z]^T \tag{2.8}$$

(utilizziamo ~ per indicare che stiamo esprimendo il punto in coordinate cartesiane) e sia m di coordinate:

$$\tilde{\mathbf{m}} = [u, v]^T \tag{2.9}$$

la sua proiezione su R attraverso C.
Mediante considerazioni sulla similitudine dei triangoli (Fig. 2.8) e ricordando di tenere conto della inversione di segno delle coordinate, vale la seguente relazione:

$$\frac{f}{z} = \frac{-u}{x} = \frac{-v}{y} \tag{2.10}$$

la quale si può riscrivere come:

$$\begin{cases} u = \dfrac{-f}{z}x \\ v = \dfrac{-f}{z}y \end{cases} \tag{2.11}$$

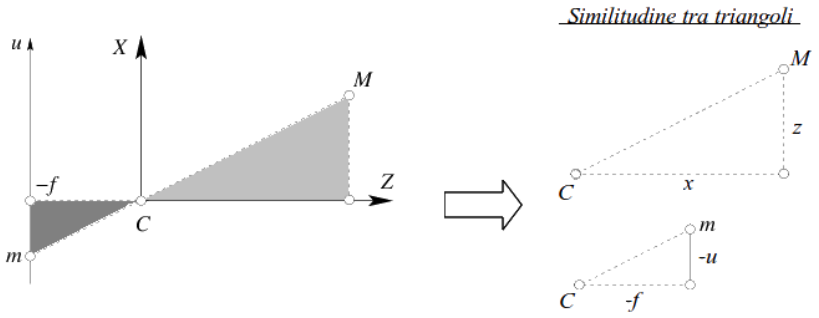


Figura 2.8: *Vista semplificata del modello della fotocamera.*

Questo sistema di equazioni rappresenta appunto la proiezione prospettica, la quale effettua una trasformazione dalle coordinate tridimensionali a quelle bidimensionali.
Come si può inoltre notare il sistema è non lineare a causa della divisione per Z, si può comunque utilizzare le coordinate omogenee (e quindi intendendo la trasformazione come tra spazi proiettivi), in modo tale da ottenere un sistema lineare.
Siano quindi le coordinate omogenee per i punti m ed M:

$$\mathbf{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad e \qquad \mathbf{M} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.12}$$

È possibile notare che ponendo la terza coordinata ad 1, abbiamo escluso i punti all'infinito (per includerli avremmo dovuto usare una terza componente generica). Dunque l'equazione di proiezione prospettica, in questo caso semplificato, si riscrive:

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fx \\ -fy \\ z \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.13}$$

e passando alla notazione matriciale:

$$z\mathbf{m} = P\mathbf{M} \tag{2.14}$$

che a meno di un fattore di scala è possibile scrivere come:

$$\mathbf{m} \simeq P\mathbf{M} \tag{2.15}$$

La matrice P rappresenta il modello geometrico della fotocamera e viene chiamata matrice della fotocamera o *matrice di proiezione prospettica* (MPP).
Vediamo ora un'altra scrittura per la proiezione prospettica che ci sarà utile nei prossimi paragrafi nella descrizione dei vincoli della Stereopsi.
Iniziamo scrivendo la MPP secondo le sue righe:

$$P = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \tag{2.16}$$

e la inseriamo nella Eq. (2.15) trovata precedentemente, ottenendo:

$$\mathbf{m} \simeq \begin{bmatrix} \mathbf{p}_1^T\mathbf{M} \\ \mathbf{p}_2^T\mathbf{M} \\ \mathbf{p}_3^T\mathbf{M} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \mathbf{M} \tag{2.17}$$

quindi l'equazione di proiezione prospettica in coordinate cartesiane, diventa:

$$\begin{cases} u = \dfrac{\mathbf{p}_1^T \mathbf{M}}{\mathbf{p}_3^T \mathbf{M}} \\ v = \dfrac{\mathbf{p}_2^T \mathbf{M}}{\mathbf{p}_3^T \mathbf{M}} \end{cases} \tag{2.18}$$

generalizzazione dell'equazione trovata in precedenza.

PROPRIETÀ MPP: parametri intrinseci ed estrinseci

Per completezza, e visto che saranno nominati spesso nel corso dei prossimi paragrafi, vediamo come vengono definiti i parametri intrinseci ed estrinseci nel modello geometrico della formazione dell'immagine.
Nel modello realistico della fotocamera oltre alla trasformazione prospettica descritta precedentemente bisogna tenere conto di:

A- PARAMETRI INTRINSECI
La pixelizzazione descrive la forma e la dimensione della matrice CCD ed inoltre la posizione della stessa rispetto al centro ottico.
Viene presa in considerazione in un modello realistico mediante una trasformazione affine che tiene conto della traslazione del centro ottico e la riscalatura indipendente degli assi u e v:

$$\begin{cases} u = k_u \dfrac{-f}{z} x + u_0 \\ v = k_v \dfrac{-f}{z} y + v_0 \end{cases} \tag{2.19}$$

dove (u_0, v_0) sono le coordinate del punto principale mentre (k_u, k_v) sono l'inverso della dimensione efficace del pixel lungo la direzione u e v. Sostituendo le nuove equazioni la MPP diventa:

$$P = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K[I|\mathbf{0}] \tag{2.20}$$

dove:

$$K = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

Effettuando il seguente cambio di variabili:

$$\begin{cases} \alpha_u = -fk_u \\ \alpha_v = -fk_v \end{cases} \tag{2.22}$$

si esprime la lunghezza focale in pixel orizzontali e verticali, i parametri intrinseci codificati dalla matrice K sono dunque i seguenti 4: $\alpha_u, \alpha_v, u_0, v_0$

B- PARAMETRI ESTRINSECI

Per descrivere un modello realistico bisogna, oltre alla pixelizzazione, tenere presente il fatto che il sistema di riferimento mondo non coincide con il sistema di riferimento standard della fotocamera, appare così necessario introdurre una trasformazione rigida che lega i due sistemi di riferimento. Si introduce così un cambio di coordinate, costituito da una rotazione R seguita da una traslazione $\mathbf{t}$, e definiamo con $\mathbf{M_c}$ le coordinate omogenee di un punto nel sistema di riferimento standard mentre con $\mathbf{M}$ le coordinate omogenee dello stesso punto nel sistema di riferimento mondo.
Risulta quindi possibile scrivere la seguente relazione:

$$\mathbf{M_c} = G\mathbf{M} \tag{2.23}$$

dove:

$$G = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.24}$$

la matrice G codifica i 6 parametri estrinseci della fotocamera (3 per la rotazione e 3 per la traslazione).
In conclusione è possibile dimostrare che la matrice di proiezione prospettica può essere scritta nella seguente forma:

$$P = K[I|\mathbf{0}]G \tag{2.25}$$

dove K rappresenta i parametri intrinseci, G quelli estrinseci e la matrice $[I|\mathbf{0}]$ è la matrice di proiezione prospettica quando si introduce il seguente cambio di coordinate (chiamate coordinate normalizzate):

$$\mathbf{p} = K^{-1}\mathbf{m} \tag{2.26}$$

2.2.3 Vincolo Epipolare

Come già detto nell'introduzione il calcolo delle corrispondenze consiste nell'individuare coppie di punti coniugati nelle due immagini che sono proiezione dello stesso punto della

scena. Inoltre abbiamo affermato che oltre al fatto che le immagini devono differire lievemente è necessario introdurre altri vincoli che rendano il calcolo delle corrispondenze trattabile.
Il più importante di questi è il vincolo epipolare, il quale afferma che il corrispondente di un punto in un'immagine può trovarsi solo su una retta (retta epipolare) nell'altra immagine. Grazie a questa considerazione la ricerca delle corrispondenze diventa unidimensionale invece che bidimensionale, in modo tale da semplificare notevolmente le cose.

Geometria Epipolare e matrice fondamentale

Come è già stato detto nella premessa grazie ad alcune considerazioni geometriche, il punto coniugato di m deve giacere su di una linea retta nella seconda immagine, chiamata retta epipolare di m. La geometria epipolare è importante soprattutto perché descrive la relazione tra due viste di una stessa scena, dunque è fondamentale per la ricostruzione del modello 3D dell'oggetto inquadrato.
Consideriamo il caso illustrato in Fig. 2.9. Dato un punto m nella prima immagine, il suo coniugato m' nella seconda immagine è vincolato a giacere sull'intersezione del piano immagine con il piano determinato da m, C e C', detto piano epipolare.

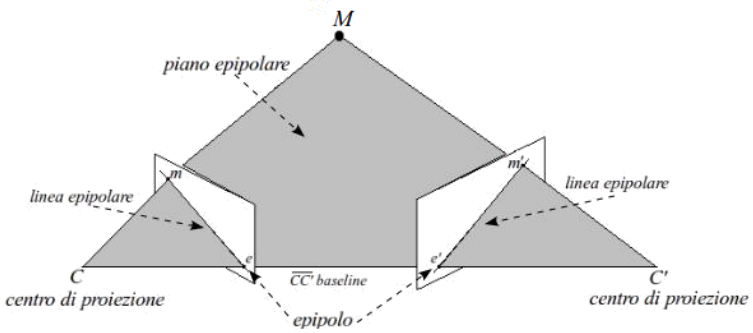


Figura 2.9: *Vista semplificata del modello della fotocamera.*

Si osserva inoltre che tutte le linee epipolari di una immagine passano per uno stesso punto, denominato epipolo, e che i piani epipolari costituiscono un fascio di piani che hanno in comune la retta passante per i centri ottici $\mathbf{C}$ e $\mathbf{C}'$. La retta $\mathbf{CC}'$ prende il nome di linea di base o baseline.
Per arrivare alla formulazione della retta epipolare si fissa un sistema di riferimento assoluto e date le due matrici di proiezione prospettica P e P' descritte dall'Eq. (2.25), si riscrivono esplicitando la sottomatrice 3×3 Q e Q':

$$P = [Q|\mathbf{q}] \quad e \quad P' = [Q'|\mathbf{q'}] \tag{2.27}$$

Inoltre grazie alla Eq. (2.15) sappiamo che:

$$\begin{cases} \mathbf{m} \simeq P\mathbf{M} \\ \mathbf{m}' \simeq P'\mathbf{M} \end{cases} \tag{2.28}$$

La linea epipolare corrispondente ad $\mathbf{m}$ è la proiezione secondo P' del raggio ottico di $\mathbf{m}$. Il raggio ottico M del punto m è la linea retta che passa per il centro ottico $\mathbf{C}$ ed $\mathbf{m}$ stesso, ovvero il luogo geometrico dei punti $\mathbf{M}$.
Sul raggio ottico giacciono tutti i punti dello spazio dei quali il punto $\mathbf{m}$ è proiezione (uno di questi per definizione è il punto $\mathbf{C}$ che come vedremo comparirà nell'equazione parametrica ed inoltre sarà presente un secondo punto ideale che assumerà il valore $\begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix}$)
Il raggio ottico è quindi descritto dalla seguente equazione parametrica:

$$\mathbf{M} = \mathbf{C} + \lambda \begin{bmatrix} Q^{-1}\mathbf{m} \\ 0 \end{bmatrix} \qquad \lambda \in \Re \cup \{\infty\} \tag{2.29}$$

il centro ottico $\mathbf{C}$ è definito invece dalla seguente equazione:

$$\mathbf{C} = \begin{bmatrix} \tilde{\mathbf{C}} \\ 1 \end{bmatrix} \qquad \tilde{\mathbf{C}} = -Q^{-1}\mathbf{q} \tag{2.30}$$

Per arrivare ad ottenere $\mathbf{m}'$, sostituiamo i valori all'Eq. (2.28) ed otteniamo:

$$P'\mathbf{C} = P' \begin{bmatrix} -Q^{-1}\mathbf{q} \\ 1 \end{bmatrix} \tag{2.31}$$

e ricordando che:

$$P' \begin{bmatrix} -Q^{-1}\mathbf{q} \\ 1 \end{bmatrix} = Q'Q^{-1}\mathbf{m} \tag{2.32}$$

allora:

$$P'\mathbf{C} = P' \begin{bmatrix} -Q^{-1}\mathbf{q} \\ 1 \end{bmatrix} = \mathbf{q}' - Q'Q^{-1}\mathbf{q} \triangleq \mathbf{e}' \tag{2.33}$$

la retta epipolare di m ha equazione:

$$\mathbf{m}' \simeq \lambda Q'Q^{-1}\mathbf{m} + \mathbf{e}' \tag{2.34}$$

Questa è l'equazione della retta passante per i punti $\mathbf{e}'$ (l'epipolo) e $Q'Q^{-1}\mathbf{m}$.

Per vedere che esiste una relazione bilineare tra i punti coniugati dobbiamo elaborare ulteriormente l'equazione. Moltiplichiamo a destra e sinistra per $[\mathbf{e}']_x$, ovvero per:

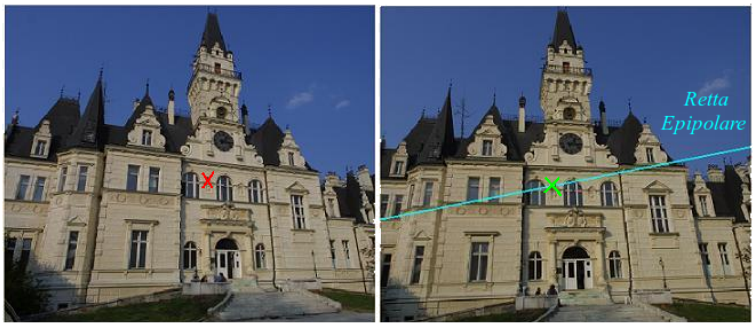


Figura 2.10: *A destra è disegnata le retta epipolare corrispondenti ai due punti coniugati*

$$[\mathbf{e}']_x = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix} \tag{2.35}$$

la quale agisce come il prodotto esterno per $\mathbf{e}'$: $[\mathbf{e}']_x\mathbf{b} = \mathbf{e}' \times \mathbf{b}$, osservando inoltre che $[\mathbf{e}']_x\mathbf{e}' = 0$ otteniamo:

$$[\mathbf{e}']_x\mathbf{m}' \simeq \lambda[\mathbf{e}']_x Q'Q^{-1}\mathbf{m} \tag{2.36}$$

La parte sinistra è un vettore ortogonale a $\mathbf{m}'$, quindi se moltiplichiamo a destra e sinistra per $\mathbf{m}'^T$, risulta:

$$0 = \mathbf{m}'^T[\mathbf{e}']_x Q'Q^{-1}\mathbf{m} \tag{2.37}$$

Questa equazione, che prende anche il nome di equazione di *Longuet-Higgins*, rappresenta la relazione bilineare tra i punti coniugati $\mathbf{m}$ ed $\mathbf{m}'$.
La seguente matrice F:

$$F = [\mathbf{e}']_x Q'Q^{-1} \tag{2.38}$$

contiene i coefficienti della forma bilineare e prende il nome di *matrice fondamentale*. L'equazione di Longuet-Higgins quindi si riscrive come:

$$\mathbf{m}'^T F\mathbf{m} = 0 \tag{2.39}$$

La matrice fondamentale contiene tutta l'informazione relativa alla geometria epipolare, conoscendola risulta possibile tracciare la retta epipolare di un qualsiasi punto, infatti scelto un punto arbitrario m, la retta definita da Fm è la corrispondente retta epipolare.
Per quanto descritto fino ad ora la matrice fondamentale serve solo a tracciare le rette epipolari, ed è calcolata come funzione delle due matrici di proiezione prospettica.
Vedremo poi quando studieremo l'altro fondamentale metodo ottico passivo, lo Struc-

ture from Motion il quale non dispone della calibrazione (quindi delle MPP), che la matrice fondamentale può essere calcolata direttamente dalle immagini.

2.2.4 Calcolo delle corrispondenze

Introduzione

Dopo aver affrontato il vincolo epipolare, possiamo ora concentrarci sulla prima fase della Stereopsi, il calcolo delle corrispondenze.
Abbiamo detto che la Stereopsi riesce ad unire due immagini grazie al fatto che queste tra loro hanno un minimo spostamento laterale, ecco quindi che nel calcolo delle corrispondenze si assume che le immagini non siano troppo diverse, ovvero che un particolare della scena appaia simile in entrambe le foto.
Grazie a questa similarità, un punto di una immagine può essere messo in corrispondenza con molti punti dell'altra immagine, questo può creare un problema, ovvero quello delle false corrispondenze, ciò che rende difficile l'identificazione delle coppie coniugate.
Oltre ai falsi accoppiamenti, vi sono altri problemi che affliggono il calcolo delle corrispondenze, in particolare dovuti al fatto che la scena viene inquadrata da due diversi punti di vista, vediamo i più importanti:

- **Occlusioni**: parti della scena che compaiono in una sola delle immagini, ovvero esistono punti in una immagine che non hanno il corrispondente nell'altra.

- **Distorsione radiometrica**: a causa di superfici non perfettamente lambertiane (l'energia incidente non riflette in modo eguale in tutte le direzioni), l'intensità osservata dalle due fotocamere è diversa per lo stesso punto della scena.

- **Distorsione prospettica**: a causa della proiezione prospettica, un oggetto proiettato assume forme diverse nelle due immagini.

Tutti questi problemi ovviamente si aggravano tanto più quanto più le fotocamere sono distanti. Un aiuto per il calcolo delle corrispondenze ci arriva da alcuni vincoli che possono essere sfruttati, alcuni di questi sono:

- **Geometria epipolare**: visto dettagliatamente nel paragrafo precedente, impone che il punto coniugato giaccia su una retta epipolare determinata dai parametri intrinseci e dalla reciproca posizione delle fotocamere.

- **Somiglianza**: un particolare della scena appare simile nelle due immagini.

- **Liscezza**: lontano dai bordi, la profondità dei punti di una superficie liscia varia lentamente. Questo pone un limite al gradiente della disparità.

- **Unicità**: un punto dell'immagine di sinistra può essere messo in corrispondenza con un solo punto nell'immagine di destra, e viceversa (fallisce se ci sono oggetti trasparenti o in presenza di occlusioni).

- **Ordinamento monotono**: se il punto m_1 in una immagine corrisponde a m'_1 nell'altra, il corrispondente di un punto m_2 che giace alla destra (sinistra) di m_1 deve trovarsi alla destra (sinistra) di m'_1.

Metodi di accoppiamento locali

Il calcolo delle corrispondenze equivale al calcolo della disparità per i punti (tutti o alcuni) dell'immagine di riferimento, per disparità si intende la differenza tra due punti coniugati immaginando di sovrapporre le due immagini, così facendo si ottiene una nuova immagine chiamata mappa disparità.
Tutti i metodi che effettuano il calcolo delle corrispondenze cercano di accoppiare pixel di una immagine con quelli dell'altra sfruttando alcuni dei vincoli elencati sopra.
In particolare queste tecniche si suddividono in locali e globali. Le prime impongono il vincolo ad un piccolo numero di pixel che circondano quello che si vuole accoppiare mentre quelle globali estendono tale vincolo ad un numero di pixel elevato.
Siccome i metodi globali hanno un costo computazionale maggiore vengono preferiti quelli locali (anche in Photosynth, utilizzato in questa tesi per la realizzazione del modello 3D), per questo motivo ci limiteremo allo studio solo di quest'ultimi. Nell'analisi di questi metodi ipotizzeremo che le linee epipolari siano parallele e orizzontali nelle due immagini sicché i punti coniugati verranno ricercati lungo le linee orizzontali delle immagini, mediante questa supposizione non vi sarà perdita di generalità poiché tale condizione risulta possibile mediante una tecnica chiamata *rettificazione epipolare*.
La mappa di disparità mediante tale ipotesi si riduce ad un campo scalare: in ogni pixel viene registrata la distanza orizzontale che separa il corrispondente punto dell'immagine di riferimento dal suo punto coniugato.

Accoppiamento di finestre

Gli algoritmi locali considerano una piccola area rettangolare in una immagine e cercano l'area equivalente più simile nell'altra immagine. Tale somiglianza viene misurata da una determinata funzione di distorsione.
In particolare questi tipi di tecniche consistono nel calcolare per ogni pixel (u, v) della prima immagine (I_1) il suo corrispondente $(u + d, v)$ nella seconda (I_2) . Si consideri quindi una finestra centrata in (u, v) di dimensioni $(2n + 1) \times (2m + 1)$.
Questa viene confrontata con una finestra delle stesse dimensioni in I_2 che si muove lungo la linea epipolare corrispondente ad (u, v); essendo le immagini rettificate secondo l'ipotesi della rettificazione epipolare, si considerano solo le traslazioni laterali ovvero le posizioni $(u + d, v)$, $d \in [d_{min}, d_{max}]$, (Fig. 2.11).
La disparità calcolata è lo spostamento che corrisponde alla massima somiglianza delle due finestre.

Possibili confronti tra finestre

Abbiamo detto che nelle tecniche locali per il calcolo delle corrispondenze è necessario

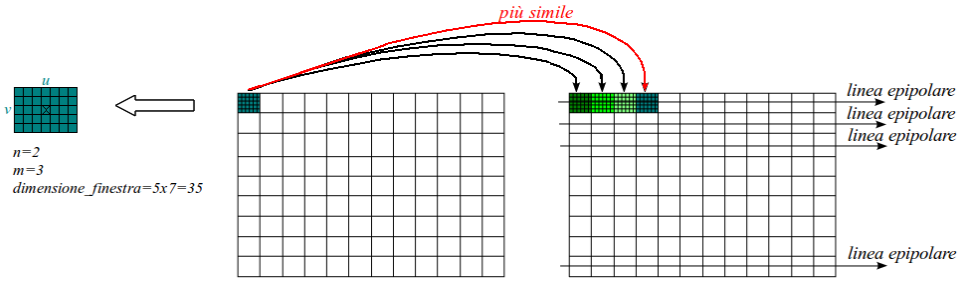


Figura 2.11: *Confronto tra finestre nel caso rettificato*

effettuare un confronto di somiglianza tra le finestre considerate, vi sono vari criteri che si possono classificare sostanzialmente in tre categorie:

A- basati sulle differenze di intensità (SSD, SAD)

B- basati su correlazione (NCC, ZNCC)

C- basati su operatori di rango (trasformata Census)

A- Criteri di somiglianza basati sulle differenze di intensità

La cosiddetta *Sum of Squared Differences* (SSD) è la tecnica più comune ed è descritta dalla seguente funzione:

$$SSD(u,v,d) = \sum_{(k,l)} (I_1(u+k, v+l) - I_2(u+k+d, v+l))^2 \tag{2.40}$$

dove $k \in [-n, n]$, $l \in [-m, m]$ e $I(u, v)$ indica il livello di intensità del pixel (u, v). Più piccolo è il valore di tale equazione, più le porzioni delle immagini considerate sono simili. La disparità calcolata è l'argomento del minimo della funzione errore:

$$d_0(u,v) = argmin_d \;\; SSD(u,v,d) \tag{2.41}$$

In questo modo viene calcolata la disparità con precisione di un pixel. Simile al SSD è il SAD (*Sum of Absolute Differences*), dove il quadrato viene sostituito dal valore assoluto. In tal modo la metrica risulta meno sensibile a rumore di tipo impulsivo: due finestre che sono uguali in tutti i pixel tranne uno risultano più simili secondo SAD che secondo SSD, poiché il quadrato pesa molto di più le differenze del valore assoluto.

$$SAD(u,v,d) = \sum_{(k,l)} |I_1(u+k, v+l) - I_2(u+k+d, v+l)| \tag{2.42}$$

B- Criteri di somiglianza basati su correlazione

La *Normalized Cross Correlation* (NCC), essendo una misura di similarità, non è alla ricerca di un minimo ma risulta una funzione da massimizzare.

In pratica viene vista come il prodotto scalare delle due finestre diviso il prodotto delle norme:

$$NCC(u,v,d) = \frac{\sum_{(k,l)} I_1(u+k, v+l) I_2(u+k+d, v+l)}{\sqrt{\sum_{(k,l)} (I_1(u+k, v+l))^2} \sqrt{\sum_{(k,l)} (I_2(u+k+d, v+l))^2}} \tag{2.43}$$

Se si sottrae a ciascun pixel la media della finestra si ottiene la ZNCC (*Zero-mean* NCC), la cui caratteristica principale è l'invarianza a cambi di luminosità tra le due immagini.

C- Criteri di somiglianza basati su operatori di Rango

La terza metrica è quella che fa uso della *trasformata Census*, la quale utilizza precedentemente una trasformazione delle immagini basata sull'ordinamento locale dei livelli di intensità e in una seconda fase misura la similarità delle finestre con *distanza di Hamming* sulle immagini così trasformate.
La trasformata Census si basa sul seguente operatore di confronto:

$$\xi(I,p,p') = \begin{cases} 1 & se\ I(p) < I(p') \\ 0 & altrimenti \end{cases} \tag{2.44}$$

dove $I(p)$e $I(p')$ sono, rispettivamente, i valori dell'intensità dei pixel p e p' . Se identifichiamo la concatenazione di bit col simbolo $\odot$, la trasformata Census per un pixel p nell'immagine I è:

$$C[I(p)] = \bigodot_{p' \in S(p,\beta)} \xi(I,p,p') \tag{2.45}$$

dove $S(p,\beta)$ rappresenta una finestra chiamata finestra di trasformazione, di raggio β centrata in p.

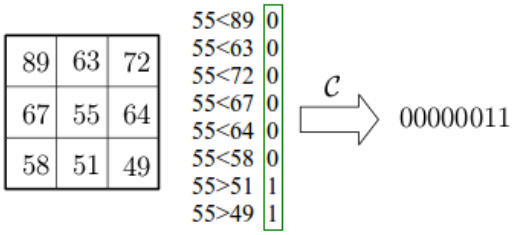


Figura 2.12: *Esempio di trasformata Census con β=1 e centrata sul valore 55*

La trasformata Census associa ad una finestra di trasformazione una stringa di bit che codifica i pixel che hanno un'intensità più bassa (o più alta) rispetto al pixel centrale (Fig. 2.12), in tal modo riassume la struttura spaziale locale.
L'accoppiamento avviene su finestre di immagini trasformate, confrontando le stringhe

di bit. Per effettuare il confronto di somiglianza si utilizza la metrica SCH (*Sum of Census Hamming distances*) che è possibile scrivere come:

$$SCH(u,v,d) = \sum_{(k,l)} C[I_1(u+k, v+l)] \ominus C[I_2(u+k+d, v+l)] \tag{2.46}$$

dove col simbolo $\ominus$ si è indicata la distanza di Hamming tra due stringhe di bit, ovvero il numero di bit nei quali esse differiscono.
Questo metodo risulta particolarmente interessante poiché risulta:

- Invariante nei confronti di qualsiasi distorsione che conservi l'ordine delle intensità.
- Robusto (tollerante nei confronti degli errori dovuti a occlusioni).
- Veloce (le operazioni sono semplici confronti di numeri interi).

Scelta ottima della finestra di confronto

Quando si ha a che fare con questi tipi di algoritmi che sono basati su finestre, risulta indispensabile tenere conto del problema legato alla discontinuità di profondità.
Quando si utilizza una finestra di correlazione che copre una regione in cui la profondità varia, la disparità calcolata sarà inevitabilmente affetta da errore, infatti non esiste una disparità unica per tutta la finestra. A questo punto si potrebbe pensare di ridurre progressivamente la dimensione della finestra per evitare questo problema, ma se la finestra risulta troppo piccola, si otterrebbe un rapporto segnale (variazione di intensità) su rumore basso e la disparità che si ottiene è poco affidabile.
In definitiva si deve far fronte a due richieste contrapposte: l'accuratezza richiede finestre piccole mentre per l'affidabilità si vorrebbero finestre grandi che permettono di avere molta variabilità nella finestra. Vediamo ora due possibili soluzioni a tale problema.

Finestre adattative/eccentriche

La prima soluzione che analizzeremo è quella proposta da *Kanade e Okutomi* [7], la quale prevede una finestra le cui dimensioni sono selezionate adattativamente in base al rapporto segnale/rumore locale e alla variazione locale di disparità. L'idea è che la finestra ideale comprenda più variazione di intensità e meno variazione di disparità possibile.
Poiché la disparità è inizialmente ignota, si parte con una stima ottenuta con finestra fissa e si itera, approssimando ad ogni passo la finestra ottima per ciascun punto, fino alla possibile convergenza.

Metodi multirisoluzione

La seconda soluzione è quella che sfrutta i metodi gerarchici, i quali hanno la caratteristica di operare a differenti risoluzioni (Fig. 2.13).
L'idea è che per il livello grossolano si utilizzino finestre larghe in modo tale da fornire

un risultato affidabile ma allo stesso tempo inaccurato.
Per i livelli più fini si utilizzano invece finestre ed intervalli di ricerca più piccoli migliorando così l'accuratezza. Per tali metodi esistono due tecniche principali:

- **Coarse-to-fine**: questa tecnica utilizza un unico intervallo di ricerca, ma opera su immagini con risoluzioni crescenti. La disparità ottenuta ad un livello è utilizzata come centro per quello successivo.

- **Fine-to-fine**: questa tecnica invece utilizza sempre la medesima immagine ma con finestre ed intervalli via via decrescenti.
  Come nella tecnica precedente, la disparità ottenuta ad un livello viene usata come centro per l'intervallo al livello successivo.

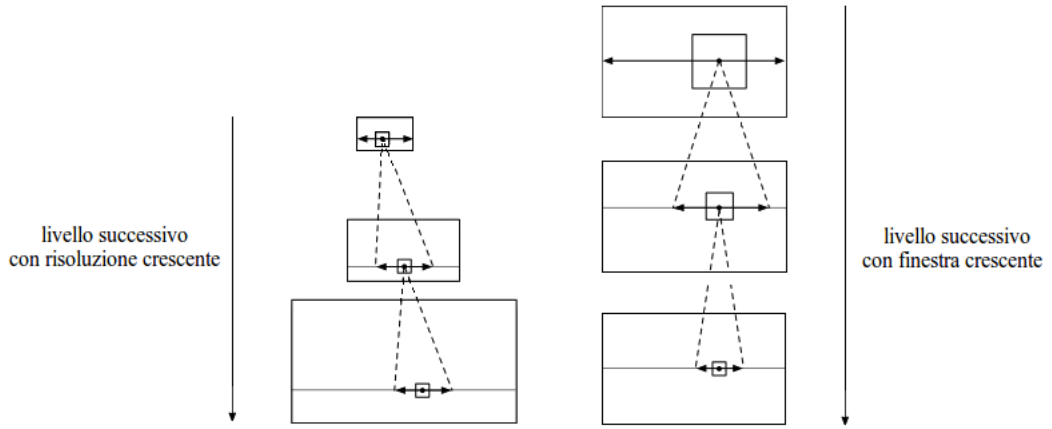


Figura 2.13: *Metodo coarse-to-fine (sinistra) e fine-to-fine (destra)*

La presenza di occlusioni

Come già detto nell'introduzione, uno dei possibili problemi nel calcolo delle corrispondenze sono le occlusioni ovvero punti della scena che compaiono in una sola immagine e quindi non hanno il corrispondente nell'altra.
Per prima cosa bisogna individuare le occlusioni, evitando che si creino false corrispondenze. Il vincolo di ordinamento monotono, quando applicabile, può servire allo scopo. Più efficace però è il vincolo di coerenza destra-sinistra, il quale definisce che se p viene accoppiato con p' effettuando la ricerca da I_1 a I_2 , allora p' deve essere accoppiato a p effettuando la ricerca da I_2 a I_1.
La Fig. 2.14 descrive il procedimento di accoppiamento nel quale per ogni punto di I_1 viene cercato il corrispondente in I_2. Se per esempio, come accade nella figura, una porzione della scena è visibile in I_1 ma non in I_2 (zona arancione), un pixel $p^o \in I_1$, il cui corrispondente è occluso, verrà accoppiato ad un certo pixel $p' \in I_2$ secondo la metrica prescelta (1° fase). Se il vero corrispondente di p' è $p \in I_1$ anche p viene accoppiato a p' (2°fase), violando il vincolo di unicità. Per capire quale dei due è corretto si

effettua la ricerca degli accoppiamenti a partire da I_2 (3° fase). In questo caso, p' viene accoppiato con il suo vero corrispondente p, dunque il punto p^o si può lasciare senza corrispondente eliminando la falsa corrispondenza. Per i punti privi di corrispondente si può stimare una disparità per interpolazione dei valori vicini, oppure lasciarli come sono.

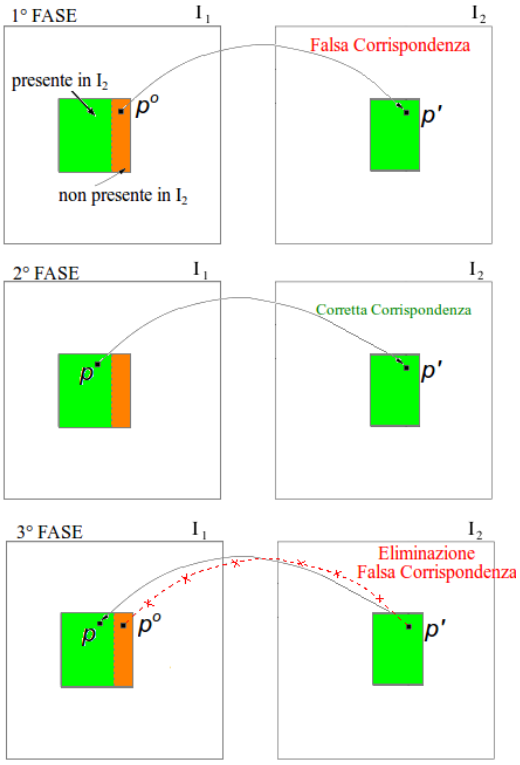


Figura 2.14: *Coerenza destra-sinistra*

Altri metodi locali

Precedentemente abbiamo descritto i principali metodi locali, vi sono però altri metodi di accoppiamento che sono basati su:

- Gradiente
- Segmentazione
- Caratteristiche salienti (feature) dell'immagine

Di questi tre metodi analizziamo il più interessante anche perché verrà sfruttato da Photosynth, programma utilizzato nella tesi per lo sviluppo del modello tridimensionale, ovvero quello che si basa sulle caratteristiche salienti dell'immagine.
I metodi feature-based estraggono dalle immagini appunto queste caratteristiche (feature) come spigoli, angoli, segmenti rettilinei e curvi che devono essere possibilmente stabili rispetto al cambio del punto di vista, alle variazioni di scala, al rumore e all'illuminazione.

Il processo di accoppiamento (matching) sfrutta una misura di distanza tra i descrittori delle feature.
Vedremo più in dettaglio un classico metodo per l'estrazione delle feature denominato SIFT quando parleremo appunto di Photosynth.

2.2.5 Triangolazione 3D

Nei paragrafi precedenti abbiamo descritto la prima fase della Stereopsi ovvero il calcolo delle corrispondenze, passiamo ora alla seconda fase, la triangolazione.
Questa tecnica mediante la matrice di proiezione prospettica che descrive i parametri estrinseci (posizioni reciproche delle telecamere ottenute tramite la calibrazione) ed i parametri intrinseci del sensore, ricostruisce la posizione tridimensionale nella scena ($\mathbf{M}$) dei punti coniugati ($\mathbf{m}$ ed $\mathbf{m}'$) delle due immagini.

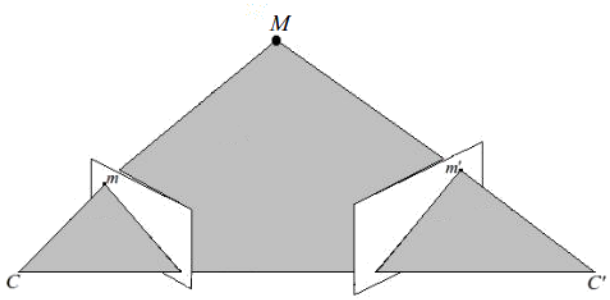


Figura 2.15: *Triangolazione*

Date:

(i) le coordinate (in pixel) di due punti coniugati $\mathbf{m}$ ed $\mathbf{m}'$;

(ii) le due matrici di proiezione prospettica relative alle due fotocamere;

vediamo ora come sia possibile mediante queste condizioni iniziali ricostruire facilmente la posizione in coordinate assolute del punto $\mathbf{M}$.
Consideriamo $\mathbf{m} = [u, v, 1]^T$, la proiezione del punto $\mathbf{M}$ sulla fotocamera che ha MPP P, (Fig. 2.15). Dalla equazione di proiezione prospettiva Eq. (2.18) si ricava:

$$\begin{cases} u = \dfrac{\mathbf{p}_1^T \mathbf{M}}{\mathbf{p}_3^T \mathbf{M}} \\ v = \dfrac{\mathbf{p}_2^T \mathbf{M}}{\mathbf{p}_3^T \mathbf{M}} \end{cases} \tag{2.47}$$

che possiamo scrivere anche come:

$$\begin{cases} \mathbf{p}_3^T \mathbf{M} u = \mathbf{p}_1^T \mathbf{M} \\ \mathbf{p}_3^T \mathbf{M} v = \mathbf{p}_2^T \mathbf{M} \end{cases} \tag{2.48}$$

ottenendo:

$$\begin{cases} (\mathbf{p}_1 - u\mathbf{p}_3)^T\mathbf{M} = 0 \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T\mathbf{M} = 0 \end{cases} \tag{2.49}$$

e quindi, in forma matriciale:

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^T \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T \end{bmatrix} \mathbf{M} = \mathbf{0}_{2x1} \tag{2.50}$$

Da quanto si può notare dall'ultima equazione, un punto ($\mathbf{m}$) fornisce due equazioni. Consideriamo quindi anche il coniugato di $\mathbf{m}$: $\mathbf{m'} = [u', v', 1]^T$ con la corrispondente matrice di proiezione prospettica P'.
Si ottengono così altre due equazioni similari per $\mathbf{m'}$ ed essendo i due punti coniugati proiezione del medesimo punto $\mathbf{M}$, le due equazioni ottenute da $\mathbf{m}$ e le altre due ottenute da $\mathbf{m'}$ possono essere impilate ottenendo un sistema lineare omogeneo di quattro equazioni in quattro incognite:

$$\begin{bmatrix} (\mathbf{p}_1 - u\mathbf{p}_3)^T \\ (\mathbf{p}_2 - v\mathbf{p}_3)^T \\ (\mathbf{p}'_1 - u'\mathbf{p}'_3)^T \\ (\mathbf{p}'_2 - v'\mathbf{p}'_3)^T \end{bmatrix} \mathbf{M} = \mathbf{0}_{4x1} \tag{2.51}$$

La soluzione del sistema è il nucleo della matrice 4×4 dei coefficienti, che deve possedere rango tre, altrimenti si avrebbe la soluzione banale $\mathbf{M} = 0$.
In presenza di rumore, la condizione sul rango non viene soddisfatta esattamente e dunque si cerca una soluzione ai minimi quadrati con la *decomposizione ai valori singolari* (SVD). *Hartley e Sturm* [8] chiamano questo metodo *linear-eigen*. Quanto esposto è stato dimostrato partendo dall'idea di avere solamente due fotocamere, la tecnica si può ovviamente generalizzare al caso di $N > 2$ fotocamere.
In questo caso ogni fotocamera aggiunge altre due equazioni e si ottiene così un sistema omogeneo di $2N$ equazioni in quattro incognite.

2.3 Structure from Motion

Nel paragrafo precedente si è descritta la Stereopsi, la quale mediante due fasi, calcolo delle corrispondenze e triangolazione, permetteva la ricostruzione tridimensionale della scena.
Abbiamo anche detto che per effettuare la triangolazione erano necessari sia i parametri intrinseci sia i parametri estrinseci ovvero le posizioni reciproche delle fotocamere.
Analizzeremo ora un'altra tecnica che, a differenza della Stereopsi, considera solo una fotocamera in movimento dove i parametri intrinseci sono ancora una volta noti, risulta ignoto invece il moto della fotocamera: mancano cioè i parametri estrinseci.
Questa metodologia è definita Structure From Motion (SfM) [9], risulta particolarmente importante perché è alla base del programma Photosynth utilizzato nello sviluppo della tesi. Infatti consideriamo di avere diverse viste di una scena ottenute mediante una fotocamera in movimento e, dato un insieme di punti corrispondenti, dobbiamo ricostruire il moto della fotocamera e la struttura.

Introduzione

Lo scopo della SfM è quello di ricavare la struttura di una scena mediante una fotocamera con movimento incognito, tale problema è stato ampiamente studiato nel passato [10].
Vi sono fondamentalmente due possibili approcci a tale questione, si possono utilizzare i metodi differenziali che prendono come dati d'ingresso le velocità dei punti nell'immagine oppure i metodi discreti che invece utilizzano un insieme di punti corrispondenti.
Uno dei metodi più interessanti, che usa il modello prospettico della fotocamera, fu proposto da *Longuet e Higgins* nel 1981 [11]. Questa tecnica si basa sulla *matrice essenziale*, che descrive la geometria epipolare di due immagini prospettiche (con i parametri intrinseci noti).
La matrice essenziale fondamentalmente codifica il moto rigido della fotocamera, infatti mediante il teorema di *Huang e Faugeras* [12] risulta possibile fattorizzarla in una matrice di rotazione R ed in una di traslazione $\boldsymbol{t}$.
Come già detto più volte, siccome i parametri intrinseci sono noti, conoscendo anche i parametri estrinseci (moto rigido della fotocamera) grazie alla matrice essenziale, si ha la completa conoscenza delle MPP delle fotocamere e quindi mediante la fase di triangolazione è possibile ricavare la struttura della scena.
Bisogna però considerare che la componente $\boldsymbol{t}$ traslazionale può essere calcolata solo a meno di un fattore di scala (ambiguità profondità-velocità), perché è impossibile determinare se il moto che si misura dall'immagine è causato da un oggetto vicino che si sposta lentamente o al contrario da un oggetto lontano che si muove rapidamente.

Determinazione della matrice essenziale

Abbiamo visto nell'introduzione l'importanza della matrice essenziale, tale matrice la possiamo considerare come una specializzazione della matrice fondamentale ove l'as-

sunzione essenziale di matrici calibrate è stata rimossa.
La matrice essenziale ha un numero inferiore di gradi di libertà e proprietà aggiuntive rispetto alla matrice fondamentale. Consideriamo ora due immagini della scena scattate dalla fotocamera in istanti di tempo differenti ed assumiamo che vi siano un certo numero di punti corrispondenti tra le due immagini.
Siano P e P' le MPP delle fotocamere che corrispondono ai due istanti di tempo e $\mathbf{p} = K^{-1}\mathbf{m}$, $\mathbf{p}' = K'^{-1}\mathbf{m}'$ le coordinate normalizzate descritte dei due punti immagine corrispondenti, Eq. (2.26) .
Lavorando in coordinate normalizzate e prendendo il sistema di riferimento della prima fotocamera come riferimento mondo, possiamo scrivere le seguenti due MPP:

$$P = [I|\mathbf{0}] \qquad P' = [I|\mathbf{0}]G = [R|\mathbf{t}] \tag{2.52}$$

Sostituendo queste due particolari MPP nell'Eq. (2.37) di Longuet- Higgins si ottiene la seguente relazione che lega punti coniugati $\mathbf{p}$ e $\mathbf{p}'$ in coordinate normalizzate:

$$\mathbf{p}'^{T}[t]_x R\mathbf{p} = 0 \tag{2.53}$$

Si definisce la matrice E, contenente i coefficienti della forma, matrice essenziale:

$$E \triangleq [\mathbf{t}]_x R \tag{2.54}$$

Sostituendo per $\mathbf{p}$ e $\mathbf{p}'$, risulta $\mathbf{m}'^{T}K'^{-T}EK^{-1}\mathbf{m} = 0$ che confrontata con la relazione $\mathbf{m}'^{T}F\mathbf{m} = 0$ per la matrice fondamentale, identifica che il rapporto tra la matrice essenziale e quella fondamentale è:

$$E = K'^{T}FK \tag{2.55}$$

Abbiamo detto che la matrice essenziale ha un numero minore di gradi di libertà rispetto a quella fondamentale, si può notare infatti che la matrice E dipende da tre parametri per la rotazione e da due per la traslazione essendo omogenea rispetto a $\mathbf{t}$ (il modulo del vettore non conta).
Perciò una matrice essenziale ha solo cinque gradi di libertà, che tengono conto della rotazione (tre parametri) e traslazione a meno di un fattore di scala (due parametri), a differenza della matrice fondamentale che ne possedeva sei. Questo riflette l'ambiguità profondità-velocità, ovvero che la componente $\mathbf{t}$ traslazionale può essere calcolata solo a meno di un fattore di scala.
In termini di vincoli, possiamo osservare che la matrice essenziale è definita quindi a meno di un fattore di scala ed è singolare, poiché $det[\mathbf{t}]_\times = 0$. Per arrivare ai cinque gradi di libertà ottenuti ragionando sulla parametrizzazione bisogna poter esibire altri due vincoli. Vedremo che questi due vincoli saranno forniti dall'eguaglianza dei due valori singolari non nulli di E, ottenendo un polinomio negli elementi della matrice essenziale, il quale garantisce due vincoli indipendenti.

2.3.1 Fattorizzazione della matrice essenziale

Come già accennato in precedenza avevamo detto che mediante il teorema di Huang e Faugeras risultava possibile fattorizzare la matrice essenziale nel prodotto di due matrici, una non nulla antisimmetrica ed una di rotazione.
Il teorema in questione è il seguente:

TEOREMA 1
Una matrice E 3×3 è una matrice essenziale, quindi fattorizzabile come prodotto di una matrice non nulla antisimmetrica e di una matrice di rotazione se e soltanto se due dei suoi valori singolari sono uguali ed il terzo è nullo.

La matrice essenziale può essere calcolata direttamente dall'Eq. (2.54) usando le coordinate normalizzate oppure dalla matrice fondamentale attraverso l'Eq. (2.55).
Vediamo ora mediante due proposizioni che nota la matrice essenziale, le MPP delle telecamere possono essere recuperate da E fino ad un fattore di scala e un'ambiguità di quattro possibili soluzioni; con la matrice fondamentale invece si avrebbe avuto un'ambiguità proiettiva.
Assumendo che la matrice della prima fotocamera sia $P = [I|\mathbf{0}]$ (in coordinate normalizzate) per calcolare la matrice della seconda telecamera P' è necessario come descritto dalla prima proposizione fattorizzare E nel prodotto SR, dove S risulta una matrice antisimmetrica mentre R una matrice di rotazione.

Proposizione
La decomposizione ai valori singolari SVD di $E \in \mathbb{C}^{m \times n}$ afferma che esiste una fattorizzazione nella stessa forma:

$$E = U\Sigma V^* \tag{2.56}$$

dove:

- U matrice unitaria di dimensioni $m \times m$
- Σ matrice diagonale di dimensioni $m \times m$
- V^* trasposta e coniugata di una matrice unitaria di dimensioni $m \times m$

supponiamo che tale decomposizione fornisca, $E = U\ diag(1, 1, 0)\ V^T$ ignorando i segni vi sono due possibili fattorizzazioni di $E = SR$:

$$S = UZU^T \qquad R = UWV^T \quad oppure \quad R = UW^TV^T \tag{2.57}$$

dove:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.58}$$

Sia $S = [\mathbf{t}]_\times$ con $\| \mathbf{t} \| = 1$ (con nessuna perdita di generalità dato che E è definita a meno di un fattore di scala). Poiché $S\mathbf{t} = 0$ ($[\mathbf{t}]_\times \mathbf{t} = 0$) ne segue che $\mathbf{t} = U(0, 0, 1)^T = u3$ ovvero l'ultima colonna di U. Ciononostante il segno di E e quindi quello di $\mathbf{t}$ non può essere determinato.

Proposizione
Dalla proposizione precedente quindi si può notare che per una data matrice essenziale:

$$E = Udiag(1, 1, 0)V^T \tag{2.59}$$

e una matrice della telecamera $P = [I|\mathbf{0}]$, vi sono quattro possibili scelte per l'altra telecamera P':

$$\begin{aligned} P' &= [UWV^T| + u_3] \\ P' &= [UWV^T| - u_3] \\ P' &= [UW^TV^T| + u_3] \\ P' &= [UW^TV^T| - u_3] \end{aligned} \tag{2.60}$$

Come osservato da Longuet-Higgins, la scelta tra i quattro spostamenti è determinata dalla richiesta che i punti 3D, la cui posizione può essere calcolata costruendo le MPP e triangolando, debbano giacere davanti ad entrambe le fotocamere, cioè la loro terza coordinata deve essere positiva.

2.3.2 Algoritmo per il calcolo della matrice essenziale

In questo paragrafo tratteremo il problema della stima di E attraverso le corrispondenze di punti.
Dato un insieme di corrispondenze di punti (sufficientemente grande) $\{(p_i, p'_i)|i = 1, \ldots, n\}$, in coordinate normalizzate, si vuole determinare la matrice essenziale E che collega i punti nella relazione bilineare:

$$\mathbf{p}'^T_i E \mathbf{p}_i = 0 \tag{2.61}$$

La matrice incognita può essere ricavata grazie alla vettorizzazione ed all'impiego del prodotto di Kronecker.
Prima di vederne il risultato diamo le definizioni di questi due operatori ed il legame che intercorre tra essi:

Definizione (Prodotto di Kronecker)
Siano A una matrice $m \times n$ e B una matrice $p \times q$. Il prodotto di Kronecker di A e B è la matrice $mp \times nq$ definita da:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{2.62}$$

Definizione (Vettorizzazione)
La vettorizzazione di una matrice A $m \times n$, denotata da $vec(A)$, è il vettore $mn \times 1$ che si ottiene impilando le colonne di A una sotto l'altra.
La connessione tra il *prodotto di Kronecker* e la vettorizzazione è data dalla seguente (importante) relazione, per matrici A, B, X di dimensioni compatibli:

$$vec(AXB) = (B^T \otimes A)vec(X) \tag{2.63}$$

Utilizzando il prodotto di Kronecker e la vettorizzazione, si ottiene allora:

$$\mathbf{p'}_i^T E \mathbf{p}_i = 0 \iff vec(\mathbf{p'}_i^T E \mathbf{p}_i) = 0 \iff (\mathbf{p}_i^T \otimes \mathbf{p'}_i^T)vec(E) = 0 \tag{2.64}$$

Ogni corrispondenza di punti genera un'equazione omogenea lineare nei nove elementi incogniti della matrice E (letta per colonne). Da n punti corrispondenti otteniamo un sistema lineare di n equazioni:

$$\underbrace{\begin{bmatrix} (\mathbf{p}_1^T \otimes \mathbf{p'}_1^T) \\ (\mathbf{p}_2^T \otimes \mathbf{p'}_2^T) \\ \ddots \\ (\mathbf{p}_n^T \otimes \mathbf{p'}_n^T) \end{bmatrix}}_{U_n} vec(E) = 0 \tag{2.65}$$

La soluzione di questo sistema lineare omogeneo è il nucleo di U_n. Se $n = 8$ il nucleo della matrice ha dimensione uno, quindi come avevamo già detto è possibile determinare la soluzione ovvero la matrice essenziale a meno di una costante moltiplicativa (fattore di scala).
Questo metodo viene chiamato *algoritmo ad 8 punti* (Fig. 2.16), e si tratta di una variante di un altro metodo fondamentale chiamato DLT che viene impiegato in Photosynth, applicazione che vedremo nel capitolo successivo.
Ovviamente nelle ricostruzioni tridimensionali reali sono disponibili ben più di otto punti coniugati, risulta quindi possibile ricavare gli elementi di E risolvendo un problema ai minimi quadrati.
La soluzione è l'autovettore unitario che corrisponde al minimo autovalore di $U_n^T U_n$, che può essere calcolato con la decomposizione ai valori singolari di U_n.

2.3.3 Nel caso di più viste

Abbiamo fino ad ora parlato del calcolo della matrice essenziale partendo da due viste (due immagini), vedremo ora come si può estendere il concetto nel caso di più viste. Per semplicità utilizzeremo il caso con tre viste, poiché risulta facile generalizzarlo

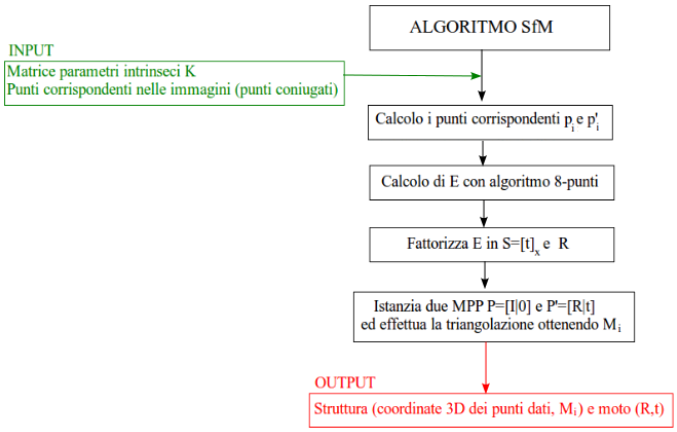


Figura 2.16: *Algoritmo SfM*

al caso di $N > 3$ immagini. Date quindi tre viste: 1, 2 e 3, applicando l'algoritmo visto precedentemente alle coppie (1,2), (1,3) e (2,3) si ottengono tre moti rigidi $(R_{12}, \widehat{\mathbf{t}}_{12})$, $(R_{13}, \widehat{\mathbf{t}}_{13})$ *e* $(R_{23}, \widehat{\mathbf{t}}_{23})$, nei quali ciascuna traslazione ovviamente è nota solo a meno di un fattore di scala (per tale motivo si considera solo il versore, denotato da $\widehat{\ }$). Per recuperare il corretto rapporto tra le norme delle traslazioni, ricordiamo che i moti rigidi devono soddisfare la seguente regola:

$$\mathbf{t}_{13} = R_{23}\mathbf{t}_{12} + \mathbf{t}_{23} \tag{2.66}$$

che si può riscrivere come:

$$||\mathbf{t}_{13}||\widehat{\mathbf{t}}_{13} = ||\mathbf{t}_{12}||R_{23}\widehat{\mathbf{t}}_{12} + ||\mathbf{t}_{23}||\widehat{\mathbf{t}}_{23} \tag{2.67}$$

dividendo m.a.m per $||t_{13}||$:

$$\widehat{\mathbf{t}}_{13} = \mu_1 R_{23}\widehat{\mathbf{t}}_{12} + \mu_2\widehat{\mathbf{t}}_{23} \qquad con\ \mu_1 = \frac{||\mathbf{t}_{12}||}{||\mathbf{t}_{13}||}\ \ e\ \ \mu_2 = \frac{||\mathbf{t}_{23}||}{||\mathbf{t}_{13}||} \tag{2.68}$$

L'equazione si può risolvere rispetto alle incognite μ_1 , μ_2 utilizzando la seguente proposizione:

Proposizione

Dati tre vettori $a, b, c \in \Re^3$ tali che $a^T(b \times c) = 0$ i due scalari μ,λ tali che :

$$\mathbf{c} = \mu\mathbf{a} - \lambda\mathbf{b} \tag{2.69}$$

si calcolano con:

$$\mu = \frac{(\mathbf{c} \times \mathbf{b})^T(\mathbf{a} \times \mathbf{b})}{||\mathbf{a} \times \mathbf{b}||^2} \tag{2.70}$$

$$\lambda = \frac{(\mathbf{c} \times \mathbf{a})^T(\mathbf{a} \times \mathbf{b})}{||\mathbf{a} \times \mathbf{b}||^2} \tag{2.71}$$

utilizzando la proposizione precedente si ottiene:

$$\frac{||\mathbf{t}_{12}||}{||\mathbf{t}_{13}||} = \mu_1 = \frac{(\widehat{\mathbf{t}}_{13} \times \widehat{\mathbf{t}}_{23})^T (R_{23}\widehat{\mathbf{t}}_{12} \times \widehat{\mathbf{t}}_{23})}{||(R_{23}\widehat{\mathbf{t}}_{12} \times \widehat{\mathbf{t}}_{23})||^2} \tag{2.72}$$

Allo stesso modo si ottiene anche μ_2.
Quindi, una volta fissata arbitrariamente una delle tre norme, ad esempio $||\mathbf{t}_{12}||$ mediante i valori calcolati di μ_1 e μ_2 risulta possibile ottenere anche le altre due componenti, $||\mathbf{t}_{13}||$ e $||\mathbf{t}_{23}||$. Ciò permette di istanziare le tre matrici di proiezione prospettica $P_1 = [I|\mathbf{0}]$, $P_2 = [R_{12}|\mathbf{t}_{12}]$ e $P3 = [R_{13}|\mathbf{t}_{13}]$ tra loro coerenti e procedere con la triangolazione.
Si noti che poiché si possono ricavare solo rapporti tra le norme, un fattore di scala globale rimane indeterminato, come nel caso di due viste.

Bundle adjustment

Quando si effettua una ricostruzione tridimensionale di una scena devono essere presenti molte viste per ottenere un risultato soddisfacente, in questo caso il metodo per il calcolo dei parametri estrinseci che abbiamo appena presentato soffre di una accumulazione degli errori che porta ad un progressivo allontanamento da un risultato ragionevole.
Risulta quindi necessario cercare un metodo che raffini il risultato mediante una minimizzazione dell'errore nell'immagine. Questa minimizzazione dell'errore deve quindi essere effettuata sia sulla struttura matrice $\mathbf{M}$ (descrive i punti tridimensionali della scena) sia nel moto effettuato della telecamera, la tecnica che effettua quanto descritto è denominata *bundle adjustment*.
In pratica si cerca di definire la posizione delle N fotocamere e degli n punti 3D affinché la somma delle distanze al quadrato tra il punto j-esimo riproiettato tramite la fotocamera i-esima ($P_i\mathbf{M}^j$) ed il punto misurato $\mathbf{m}_i^j$ sia più piccola possibile in ogni immagine dove il punto appare:

$$min_{R_i,\mathbf{t}_i,\mathbf{M}^j} \sum_{i=1}^{N}\sum_{j=1}^{n} d(K_i[R_i|\mathbf{t}_i]\mathbf{M}^j, \mathbf{m}_i^j)^2 \tag{2.73}$$

Nell'espressione che viene minimizzata bisognerà prestare attenzione nel parametrizzare correttamente R_i , in modo che compaiano solo tre incognite invece che tutti i nove elementi della matrice.
Tipicamente il problema assume notevoli dimensioni, per affrontarlo in maniera adeguata si ricorre ad una strategia che viene effettuata tramite due passi alternati:

(I) Si tengono fermi i punti e si risolve la minimizzazione rispetto alle fotocamere (problema di calibrazione).

(II) Si fissano le fotocamere e si calcolano i punti 3D (problema di triangolazione).

Si itera fino a convergenza.

Capitolo 3

Software utilizzato per la creazione del modello 3D

Nel capitolo precedente abbiamo analizzato nel dettaglio i due principali metodi passivi per l'acquisizione della forma: la Stereopsi e la Structure From Motion. Proprio quest'ultima tecnica è alla base di Photosynth [13], il software da noi utilizzato per la realizzazione del modello 3D da ottimizzare.
In questo capitolo dopo una breve descrizione dei motivi della nascita di Photosynth, vedremo nel dettaglio le operazioni effettuate da questo programma per l'identificazione della geometria tridimensionale di una scena.

3.1 La nascita di Photosynth

Figura 3.1: *A sinistra esempio di alcune immagini digitali utilizzate per la generazione del modello 3D con Photosynth, a destra il risultato ottenuto (184 immagini digitali utilizzate)*

Per lo sviluppo di questa tesi e dell'argomento in questione si è utilizzato un programma di ricostruzione 3D tramite bundle adjustment: Photosynth.
Questo programma è stato inizialmente sviluppato da Microsoft Live Search e dall'U-

niversità di Washington, ed è nato con l'idea di poter navigare ed esplorare un gran numero di foto non ordinate. Con l'indicazione non ordinate intendiamo un insieme di foto realizzate da utenti diversi quindi con punti di vista, fotocamere e in condizioni climatiche differenti. Tuttavia tutte le immagini rappresentano lo stesso oggetto del quale si vuole ottenere un modello 3D.
Il software disponibile, quindi, per prima cosa ha il compito di individuare i punti più significativi dell'immagine in modo tale da poterli confrontare con quelli delle altre fotografie.
Le foto che hanno particolari comuni (punti coniugati) poi vengono corrisposte fra loro in modo da poterle affiancare e ricostruire così la scena 3D, Fig. (3.1). Ciò risulta possibile poiché tale software utilizza una tecnica simile a quelle di Structure from Motion (Paragrafo 2.4). In questi algoritmi, ottenuti una serie di punti coniugati e le informazioni intrinsiche ed entrinseche della fotocamera, mediante la triangolazione (Paragrafo 2.3.5) è possibile generare il modello tridimensionale. In particolare l'applicazione restituisce una nuvola di punti 3D che rappresentano la scena e di questa ne risulta possibile la navigazione.
Nello sviluppo della tesi ci siamo concentrati nella fase di preprocessing, sono presenti infatti altre applicazioni, una di queste ad esempio permette la navigazione all'interno del modello 3D ed allo stesso tempo di esplorare il set di foto utilizzato. Ciò è possibile come è già stato detto in precedenza conoscendo le informazioni ottenute dalle fotocamere (posizione, orientamento e campo visivo) che permettono di posizionare delle miniature nel punto 3D in cui sono scattate tali immagini e grazie ad una particolare tecnica di rendering è possibile effettuare una fluida transizione da una all'altra.
Un'altra applicazione interessante invece è quella che consente di trasferire automaticamente le informazioni (possono essere di qualsiasi tipo: geografico, storico, ecc) di un oggetto a tutte le foto che lo rappresentano anche se esse ne sono prive perché scattate da fotocamera o da cellulare.
Come si può quindi intuire (anche dal nome di questo software), questo sistema rende possibile la creazione di tour fotografici di particolari luoghi panoramici o storici.

3.2 Gli scopi di Photosynth

Uno dei principali fini di Photosynth, oltre a quello base di realizzare dei modelli 3D di oggetti, è la possibilità di creare dei percorsi turistici virtuali di particolari luoghi a livello globale.
Uno degli obiettivi primari del rendering sarà quindi quello di evocare una sensazione di presenza da parte dell'utente all'interno della scena tridimensionale che si realizza.
In questi ultimi anni sono stati fatti dei passi da gigante in questo senso attraverso numerosi metodi di sintesi della vista realizzati da varie comunità di ricerca e anche grazie a prodotti commerciali che si basano su applicazioni per la rappresentazione di scene 3D.
Tutto ciò è stato reso possibile grazie all'accoppiata fotografia digitale ed internet. Infatti, grazie alla facilità di condivisione di foto digitali nella rete, per ogni utente

risulta possibile condividere un grande quantitativo di immagini personali.
Per capire come questo fenomeno sia in espansione, basta cercare su qualsiasi motore di ricerca un qualsiasi oggetto o luogo turistico ben conosciuto dalla comunità e risulta possibile ottenere come risultato migliaia di foto, scattate ovviamente da utenti diversi, con differenti macchine fotografiche, angolazioni e anche condizioni climatiche.
Purtroppo, l'enorme proliferazione di fotografie condivise ha fatto si che tali elementi siano impossibili da gestire ed effettuarne una classificazione o un'ordinazione, la rete infatti restituisce pagine e pagine di miniature che l'utente deve setacciare.
Photosynth, come accennato nell'introduzione, si basa sulle informazioni prese dalle immagini stesse, risulta possibile infatti determinare molti dettagli dalle posizioni ed angolazioni dei fotografi.
In particolare dalla fotocamera e dalle informazioni dedotte dalla scena si riesce a garantire le seguenti funzionalità:

- Visualizzazione di scene in 3D in base all'immagine selezionata.
- Visualizzazione delle foto che contengono un oggetto o una parte della scena.
- Restituzione di informazioni sul luogo in cui è stata scattata la foto.
- Descrizione degli oggetti visibili nella foto grazie al trasferimento delle annotazioni da foto simili.

Il sistema quindi risulta molto robusto ed in grado di gestire grandi collezioni di immagini disorganizzate prese da diverse telecamere in condizioni molto diverse.

3.3 Passi fondamentali per la generazione del modello 3D

Quello su cui focalizzeremo la nostra attenzione, che risulta anche la spina dorsale del sistema in questione, è la tecnica denominata Structure from Motion (SfM) la quale come già descritto (Paragrafo 2.4) genera la struttura dal flusso ottico per ricostruire le informazioni 3D richieste.
L'approccio utilizzato da Photosynth prima di tutto necessita di informazioni precise riguardo la posizione, l'orientamento e i parametri intrinseci (come la lunghezza focale) per ogni fotografia del nostro dataset.
Alcune caratteristiche di Photosynth richiedono la posizione assoluta delle telecamere, in un sistema di coordinate spaziali. Alcune di queste informazioni possono essere fornite con i dispositivi GPS, ma la maggior parte delle fotografie esistenti non possiede tali informazioni. Molte fotocamere digitali incorporano la lunghezza focale e altre informazioni nei *tag EXIF* delle immagini (ecco perché nel capitolo precedente parlando della SfM i parametri intrinseci erano supposti noti). Questi valori sono utili per l'inizializzazione ma a volte sono imprecisi, nel nostro caso il sistema non si basa sulle indicazioni della fotocamera ma determina queste informazioni dalle stesse immagini utilizzando tecniche di Computer Vision, in modo da ottenere una maggior precisione.

Una volta ottenute informazioni precise riguardo la posizione, l'orientamento e i parametri intrinseci per ogni fotografia del nostro dataset si determinano per prima cosa le corrispondenze di punti salienti tra le immagini (punti coniugati), come già descritto a grandi linee nel capitolo precedente (Paragrafo 2.3.4).
In un seconda fase si effettua la SfM che permette di recuperare i parametri delle telecamere e le posizioni 3D dei punti coniugati determinati nella fase precedente in modo tale da generare mediante triangolazione il modello 3D "sparso" della scena.
Passiamo adesso ad analizzare le due fasi descritte in precedenza per la realizzazione del modello 3D.

3.3.1 PRIMA FASE: Rilevamento punti chiave e matching

La prima fase ha il compito di generare delle tracce, un insieme connesso di punti chiave coincidenti su più immagini.
Per fare ciò come prima cosa si determinano i punti chiave di ogni singola immagine utilizzando un particolare algoritmo denominato SIFT.
Una volta determinati questi punti chiave, si cerca di ottenere delle coppie coerenti di punti coniugati effettuando il matching. Ottenute tali corrispondenze risulta possibile finalmente organizzarle in tracce.
Analizziamo ora più in dettaglio queste fasi:

A- Scale Invariant Feature Transform

L'algoritmo SIFT è un particolare metodo locale basato su feature (Paragrafo 2.3.4) ed è stato pubblicato da *David Lowe* [14].
Per feature si intendono caratteristiche locali che possono essere automaticamente individuate come bordi, angoli e punti che avendo una tessitura particolare possono essere individuati e distinti rispetto agli altri all'interno di un'altra immagine.

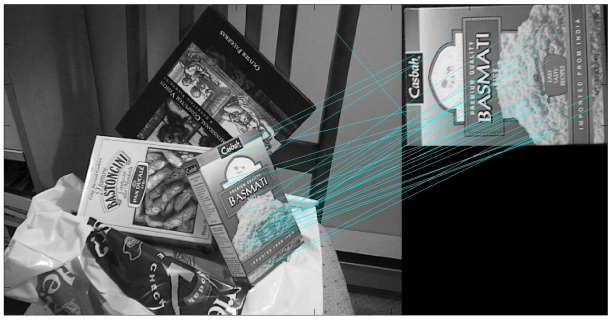


Figura 3.2: *Esempio di matching tra feature di due immagini*

Per ogni oggetto in un'immagine, punti interessanti possono essere estratti per fornire una descrizione delle caratteristiche dell'oggetto stesso. Per un riconoscimento affidabile, è importante che le feature estratte dall'immagine siano rilevabili anche con cambiamenti di scala, rumore e illuminazione. Tali punti di solito sono posizionati in regioni ad alto contrasto dell'immagine, come contorni di oggetti o in regioni in cui il

segnale presenta una forte energia nelle alte frequenze.
Un'altra importante caratteristica delle feature è che le loro posizioni relative, nella scena originale non dovrebbero cambiare tra più immagini, infatti questi tipi di algoritmi funzionano bene su scene statiche mentre se sono applicate ad oggetti articolati o flessibili tipicamente non funzionano.
Uno dei fini quindi di tali applicazioni oltre alla semplice individuazione delle caratteristiche di una immagine è appunto metterle in relazione cercando di capire come è orientata un'immagine rispetto ad un'altra e poter stimare così il vettore di moto, Fig. (3.2).
Abbiamo detto che affinché ci sia un riconoscimento affidabile, le feature debbono avere determinate caratteristiche, SIFT è stato scelto proprio poiché garantisce quelle fondamentali:

- INVARIANZA alle diverse trasformazioni dell'immagine (rotazione, scala, luminosità).
- PRESENZA REGOLARE in immagine successive.
- TRACCIABILITÀ in immagini diverse.
- IMMUNITÀ al rumore.

Vediamo ora quali sono i passi principali dell'algoritmo di estrazione delle SIFT:

1- Individuazione degli estremi locali nello scale-space
Si cercano punti interessanti su tutte le scale e posizioni dell'immagine utilizzando una funzione (DoG) *Difference of Gaussian*.
L'approccio utilizzato è quella del filtraggio in cascata che consente di determinare le posizioni e la scala delle feature candidate ad essere punti chiave e che, in un secondo momento vengono caratterizzate con maggior dettaglio.

2- Localizzazione dei keypoint
Per ciascun punto candidato viene costruito un modello dettagliato per determinarne posizione e scala.
I punti vengono inoltre selezionati secondo le seguenti misure di stabilità:

- analisi a soglia del contrasto.
- verifica della presenza di bordi o linee.
- interpolazione della posizione per aumentare la precisione della localizzazione.

I punti che non superano tutti questi test vengono scartati.

3- Generazione delle orientazioni
Per ottenere l'invarianza rotazionale, ad ogni punto chiave (*keypoint*) vengono associate una o più orientazioni calcolate in base ai gradienti locali dell'immagine.

4- Generazione del descrittore
A partire dai gradienti locali dell'immagine, alla scala selezionata e nell'intorno del punto chiave, viene costruito un descrittore di 128 elementi calcolato in base ai gradienti di tutti i pixel prossimi al punto chiave.
L'algoritmo di costruzione del descrittore permette di ottenere l'invarianza dello stesso a numerose trasformazioni tra le quali illuminazione, punto di ripresa e scala.

B- Matching

Una volta ottenuti i descrittori dei punti chiave per ogni coppia di immagini viene effettuato il matching.
A partire dai descrittori è possibile mettere in relazione i punti trovati nelle varie immagini; dato un punto nell'immagine I_a con descrittore d_{pa} il punto corrispondente nell'immagine I_b è quello che ha il descrittore più vicino (distanza euclidea) a d_{pa}. Può accadere che non tutti i punti presenti in un'immagine hanno un corrispondente nell'altra immagine (occlusioni, nuovi oggetti della scena...).
Occorre dunque avere un criterio che permetta di capire se un punto ha un buon match perché il semplice utilizzo di una soglia fissa sulla distanza tra descrittori non si è dimostrata una buona scelta.
L'algoritmo SIFT utilizza come misura della bontà del matching la seguente relazione:

$$r = \frac{d_1}{d_2} \tag{3.1}$$

dove:

- d_1 distanza euclidea tra il descrittore d_{pa} ed il descrittore con minima distanza (d_{pb1} relativo ai punti dell'immagine I_b) da d_{pa}
- d_2 distanza euclidea tra il descrittore d_{pa} ed il descrittore con minima distanza da d_{pa} dopo d_{pb1}

Nel caso di matching corretto r assumerà valori non elevati, in particolare se $r < 0.8$ SIFT considera il matching corretto altrimenti il punto con descrittore dpa non deve essere preso in considerazione.

C- Determinazione della matrice fondamentale

Successivamente è necessario stimare una matrice fondamentale per tale coppia di punti coniugati, si ricorda che la matrice fondamentale (Paragrafo 2.3.3) è la rappresentazione algebrica della geometria epipolare. Ogni punto $\mathbf{x}$ in un'immagine ha la sua linea epipolare corrispondente ($\mathbf{L}'$) nell'altra immagine; viceversa ogni punto $\mathbf{x}'$ corrispondente al punto $\mathbf{x}$ nella seconda immagine deve cadere sulla linea epipolare $\mathbf{L}'$. Tale linea è la proiezione nella seconda immagine del raggio dal punto $\mathbf{x}$ attraverso il centro $\mathbf{C}$ della prima telecamera; pertanto esiste una mappatura da un punto in un'immagine sulla corrispondente linea epipolare nell'altra immagine $\mathbf{x} \Rightarrow \mathbf{L}'$.

Photosynth determina la matrice fondamentale durante ogni iterazione di ottimizzazione RANSAC [15] (vedremo nel seguito) attraverso un algoritmo otto-punti DLT [16].
Abbiamo già visto nel dettaglio (Paragrafo 2.4.2) come sia possibile mediante un algoritmo otto-punti determinare la matrice essenziale, una volta determinata, mediante l'Eq. (2.55) la quale lega appunto la matrice essenziale e quella fondamentale risulta facile ottenere quest'ultima che è quella che ci interessa.
Siccome l'algoritmo otto-punti è già stato affrontato in dettaglio ci soffermeremo ora sull'algoritmo di ottimizzazione RANSAC.
L'algoritmo RANSAC (*RANdom SAmple Consensus*) è un metodo iterativo per la stima dei parametri di un modello matematico a partire da un insieme di dati contenente *outlier*. È un algoritmo non deterministico ovvero produce un risultato corretto solo con una data probabilità, la quale aumenta al crescere delle iterazioni che si effettuano.
Per il corretto funzionamento si assume come prima cosa che i dati siano suddivisi tra inlier (dati che hanno una distribuzione che può essere associata ad un insieme di parametri di un modello) ed outlier (dati che non sono rappresentabili mediante i parametri del modello).
C'è da considerare inoltre che i dati sono affetti da rumore, gli outlier infatti possono provenire ad esempio da valori estremi di rumore o da ipotesi e misure errate.
RANSAC inoltre assume che, dato un insieme (solitamente ridotto) di inlier, esiste una procedura che può stimare i parametri di un modello che rappresenta in modo ottimale i dati.
A differenza delle classiche tecniche per la stima dei parametri come quella ad esempio a minimi quadrati le quali non hanno meccanismi per rilevare e rigettare errori gravi, RANSAC è in grado di smussare dati contenenti una parte rilevante di errori; esso è particolarmente valido nell'analisi di immagini poiché i rilevatori di punti caratteristici locali, che sono spesso soggetti a errori, sono la sorgente di dati fornita per l'interpretazione dell'algoritmo.
La procedura seguita da RANSAC è opposta a quella delle tecniche convenzionali; invece di utilizzare fin da subito il numero più grande possibile di dati per ottenere una soluzione iniziale da cui scartare i valori non validi, esso utilizza un numero contenuto di punti da arricchire con dati consistenti qualora fosse possibile.
Vediamo ora a grandi linee come funziona l'algoritmo RANSAC.

Descrizione formale algoritmo RANSAC

Per l'utilizzo dell'algoritmo RANSAC è necessario avere un modello che necessiti di un minimo di N punti per generare i suoi parametri liberi e un insieme di dati puntuali M tale che la cardinalità di M sia maggiore di N.
Come prima cosa si sceglie casualmente un sottoinsieme S_1 di N punti da M e si genera il modello. Una volta generato il modello che chiameremo M_1, lo utilizzeremo per determinare un sottoinsieme S_1^* di punti in M che ricadano all'interno di una data soglia di tolleranza d'errore di M_1.
L'insieme S_1^* è chiamato l'insieme consenso di S_1.
Se la cardinalità di S_1^* è maggiore di una soglia $\mathbf{t}$ che è una funzione della stima del

numero degli errori in P, si utilizza S_1^* per calcolare, possibilmente attraverso una ricerca ai minimi quadrati, un nuovo modello M_1^*.
Se invece la cardinalità di S_1^* è minore di $\mathbf{t}$ si sceglie casualmente un nuovo sottoinsieme S_2 e si ripete il processo precedentemente esposto. Se, dopo un predeterminato numero di tentativi, non è stato trovato un insieme di consenso con un numero uguale o superiore a $\mathbf{t}$, allora si stima il modello con l'insieme di consenso maggiore trovato fino a quel punto oppure si termina fallendo l'operazione.
Il paradigma RANSAC quindi, come abbiamo appena visto, contiene tre parametri non specificati:

(i) La tolleranza d'errore usata per determinare se un punto è compatibile con un modello.

(ii) Il numero di sottoinsiemi da testare.

(iii) La soglia $\mathbf{t}$, che è il numero di punti compatibili per stabilire se è stato trovato un modello corretto.

Dopo aver visto in maniera molto formale come lavora l'algoritmo possiamo dire che uno dei suoi vantaggi è proprio la sua capacità di stimare in maniera robusta i parametri del modello, nonostante possa essere influenzato da un elevato numero di outlier.
Uno svantaggio invece è quello che risulta necessaria l'impostazione di soglie che variano in base al problema che si sta affrontando ed inoltre non vi è un tempo limite richiesto per la stima dei parametri. Quando il numero di iterazioni è limitato la soluzione ottenuta potrebbe non essere ottima, la soluzione è eseguire un numero elevato di iterazioni in modo che la probabilità di stima del modello cresca.

D- Eliminazione outlier e generazione tracce

L'ultima operazione della prima fase ha il compito di eliminare i matches che sono valori anomali per la matrice fondamentale. Se il numero di matches rimanenti è minore di venti, tutte le corrispondenze non vengono considerate.
A questo punto abbiamo trovato una serie di match geometricamente coerenti tra ogni coppia di immagini, come già detto non rimane che organizzare i matches in tracce.
Per la seconda fase di ricostruzione del modello 3D manteniamo solo le tracce con almeno due punti chiave e che non contengono più di un punto chiave per la stessa immagine, in caso contrario vengono considerate inconsistenti e quindi eliminate.

3.3.2 SECONDA FASE: Structure from Motion

Una volta organizzati i matches in tracce geometricamente coerenti, la prima cosa che il sistema si preoccupa di fare è ottenere i parametri delle fotocamere e la posizione 3D per ogni traccia.
I parametri recuperati devono essere coerenti, nel senso che l'errore di riproiezione, (ovvero la somma delle distanze tra le proiezioni di ciascuna traccia) sia minimizzato. Il

problema di minimizzazione viene affrontato come un problema non-lineare ai minimi quadrati e viene risolto con specifici algoritmi come quello di *Levenberg-Marquardt* [17]. Bisogna considerare però che la tecnica SfM rischia di essere bloccata da un errata stima del valore minimo essendo una tecnica formulata su larga scala, gli algoritmi per la risoluzione garantiscono solo di trovare minimi locali e quindi risulta fondamentale fornire una buona stima dei parametri iniziali.
Abbiamo detto che come prima cosa il sistema stima i parametri delle telecamere e la posizione 3D di ogni singola traccia, per fare ciò utilizza un metodo di ottimizzazione incrementale ovvero non si stimano i parametri di tutte le telecamere e delle tracce in una sola esecuzione ma si aggiunge una telecamera alla volta.
Si comincia quindi stimando i parametri estrinseci ed intrinseci necessari per la triangolazione 3D di una singola coppia di telecamere. Questa coppia iniziale deve avere un gran numero di corrispondenze e una lunga baseline (Paragrafo 2.3.3) in modo tale che le posizioni 3D dei punti osservati siano coerenti.
Il sistema sceglierà quindi la coppia di immagini che ha il più grande numero di corrispondenze, soggette però alla condizione che queste corrispondenze non possano essere ben modellate da una singola omografia in modo da evitare casi degeneri. Di questa coppia di immagini si determinano mediante triangolazione i primi punti 3D relativi alle corrispondenze. Successivamente si aggiunge un'altra telecamera al metodo di ottimizzazione scegliendo la telecamera che osserva il più grande numero di tracce nelle quali i punti 3D sono già stati stimati ed inizializzeremo i parametri estrinseci di questa nuova telecamera utilizzando l'algoritmo DLT all'interno della procedura RANSAC.
Il DLT fornisce una stima anche dei parametri intrinseci $\mathbf{K}$, quindi si utilizza tale matrice e la lunghezza focale (stimata dai tag EXIF delle immagini) per inizializzare la lunghezza focale di questa telecamera. Infine si aggiungono le tracce osservate dalla telecamera nella procedura di ottimizzazione. Una traccia è aggiunta se viene osservata da al minimo un'altra telecamera (già presente nell'insieme ottimizzato) e se triangolando la traccia da una buona stima della sua posizione.
Questa procedura è ripetuta aggiungendo una telecamera alla volta finché per tutte è stato possibile triangolare il punto 3D delle relative corrispondenze.
Per evitare l'accumulazione degli errori, ad ogni iterazione si utilizza la tecnica bundle adjustment che abbiamo descritto anche nel capitolo precedente la quale permette il raffinamento del risultato minimizzandone l'errore. Al termine del processo SfM, otterremo una rappresentazione della scena 3D organizzata nel seguente modo:

- Un'insieme di punti $P = \{p_1, p_2, p_3, \ldots\}$ ogni punto consiste in una posizione 3D e un colore ottenuti da un punto dell'immagine dove questo viene osservato.

- Una serie di telecamere $C = \{C_1, C_2, C_3, ...\}$ ogni camera C_j consiste in una immagine I_j una matrice di rotazione R_j e traslazione t_j e una lunghezza focale f_j.

- Un mapping denominato *Points*, tra le telecamere e i punti che osservano. Quindi $Points(C)$ è il sottoinsieme di P contenente i punti osservati dalla camera C.

- Un insieme di segmenti 3D, $L = \{l_1, l_2, \ldots, l_m\}$ ed un mapping denominato *Lines*, tra le telecamere e l'insieme delle linee che osservano.

3.4 Algoritmo Bundler per la generazione della SfM

Nel paragrafo precedente abbiamo esposto in maniera descrittiva le due fasi principali utilizzate da PhotoSynth per la generazione del modello tridimensionale.
Nella pratica per effettuare tali operazioni ci si avvale di *Bundler* [18], un particolare sistema Structure from Motion (SfM) che ricostruisce la scena in maniera incrementale, (un po' di telecamere alla volta), utilizzando un'ottimizzazione del pacchetto *Sparse Bundle Adjustment* sviluppato da *Lourakis and Argyros* [19]. Questo programma viene distribuito in linguaggio C e C++ e con licenza *General Public License* (GNU).
Per poter eseguire Bundler è sufficiente fornire in ingresso un dataset di immagini anche non organizzate, calcolare le feature su ogni immagine e successivamente determinare le coppie di punti corrispondenti. Se vengono fornite tali informazioni, risulta possibile ottenere in uscita la ricostruzione 3D delle telecamere e una rappresentazione tridimensionale della geometria della scena.
Poiché Bundler fornisce una nuvola di punti 3D "sparsa", è possibile utilizzare un interessante pacchetto elaborato dal *Dr. Yasutaka Furukawa*, distribuito con il nome di PMVS2 [20], il quale permette di ottenere un modello tridimensionale più denso.
Quindi, solitamente, si eseguono i seguenti step:

1- Esecuzione di Bundler, al fine di ottenere i parametri delle telecamere e un modello 3D "sparso".

2- Esecuzione del pacchetto Bundle2PMVS, permette di convertire i risultati in uscita da Bundler al fine di poterli presentare come ingresso al PMVS2.

3- Esecuzione di PMVS2 con i dati di Bundler convertiti per ottenere un modello 3D più denso.

Per completezza risulta interessante ricordare che nella distribuzione del sorgente di Bundler sono presenti degli algoritmi potenzialmente utili che riguardano concetti fondamentali della *Computer Vision*. Alcuni di questi sono:

- Stima della matrice fondamentale;

- Calibrazione 5-punti della relativa posizione;

- Triangolazione di raggi multipli.

3.4.1 Esecuzione Bundler

Il metodo più semplice per eseguire Bundler è utilizzare lo script RunBundler.sh fornito nella distribuzione del sorgente. Eseguendo questo script, in una cartella contenente un set di immagini nel formato .JPEG, si effettuano in maniera automatica tutti gli step

necessari per svolgere la Structure from Motion.
Come abbiamo già detto precedentemente, l'esecuzione del processo Bundler è l'ultimo step necessario per ottenere la ricostruzione tridimensionale di una scena. Ecco quindi che ci viene in aiuto lo script *RunBundler.sh*, il quale si prende cura di tutte le fasi necessarie per lo sviluppo del modello 3D.
Lo step iniziale di questo script permette di creare una lista delle immagini che contengono negli Exif tags le informazioni sulla lunghezza focale.
Il secondo step invece effettua il calcolo delle feature su ogni immagine; per fare ciò si potrebbe utilizzare qualsiasi tipo di applicazione sviluppata per il calcolo delle feature, Bundler però assume che queste siano nel formato SIFT, quindi che si utilizzi il detector di David Lowe descritto nel paragrafo precedente.
Il terzo ed ultimo step svolto da RunBundler.sh prima dell'esecuzione di Bundler, si occupa di determinare le corrispondenze tra feature su coppie di immagini presenti nel dataset fornito in ingresso.
In definitiva, si possono identificare quattro step necessari per la ricostruzione tridimensionale:

1- Creazione di una lista di immagini utilizzando lo script *extract_focal.pl* (estrae da ogni immagine sfruttando gli EXIF tags le informazioni sulla lunghezza focale e successivamente le memorizza sotto forma di lista).

2- Genera le features per ogni immagine, utilizzando l'algoritmo SIFT di David Lowe.

3- Effettua il match tra coppie di immagini. Le corrispondenze tra le feature determinate sono memorizzate in un file chiamato *matches.init.txt*.

4- Esegue *bundler*.

3.4.2 Formato in uscita

Il programma Bundler, produce in uscita file tipicamente chiamati *bundle_<n>.out*. Ricordando che la SfM Bundler di Photosynth agisce in maniera incrementale, risulta facile capire che questi file contengono lo struttura parziale della scena, ottenuta registrando un numero parziale di telecamere.
Una volta che tutte le telecamere vengono registrate, Bundler fornisce in uscita un file finale denominato *bundle.out*. In aggiunta viene fornito un file di estensione *.ply*, il quale contiene la ricostruzione delle telecamere e dei punti della scena tridimensionale. Questo file .ply può essere visualizzato con programmi come *Meshlab*, i quali permettono la creazione, l'analisi e la modifica di mesh triangolari tridimensionali non strutturate.
Andiamo ora ad analizzare nel dettaglio i file bundle che contengono la stima della scena e la geometria delle telecamere, in particolare, questi file hanno il seguente formato:

```
# Bundle file v0.3
<num_telecamere > <num_punti>
```

<telecamera1>
<telecamera2>
<telecamera3>
<telecamera4>
. . . . . .
<telecameraN>
<punto1>
<punto2>
<punto3>
<punto4>
. . . . . .
<puntoM>

num_telecamere e num_punti assumono valori interi poiché rappresentano il numero di telecamere utilizzate nella simulazione e il numero di vertici 3D determinati eseguendo il processo Bundler.
Per ogni telecamera <telecameraI> con I$\in \{1 \dots N\}$ specificata nell'ordine con la quale appare nella lista immagini (*list.txt*), si definiscono i parametri intrinseci ed estrinseci stimati, nella seguente forma:

$< f > < k1 > < k2 >$
$< R >$
$< t >$

Dove: f rappresenta la lunghezza focale, k_1 e k_2 i coefficienti di distorsione radiale, ed infine, R matrice 3×3 e t vettore di 3 componenti che descrivono rispettivamente la rotazione e la traslazione della telecamera.

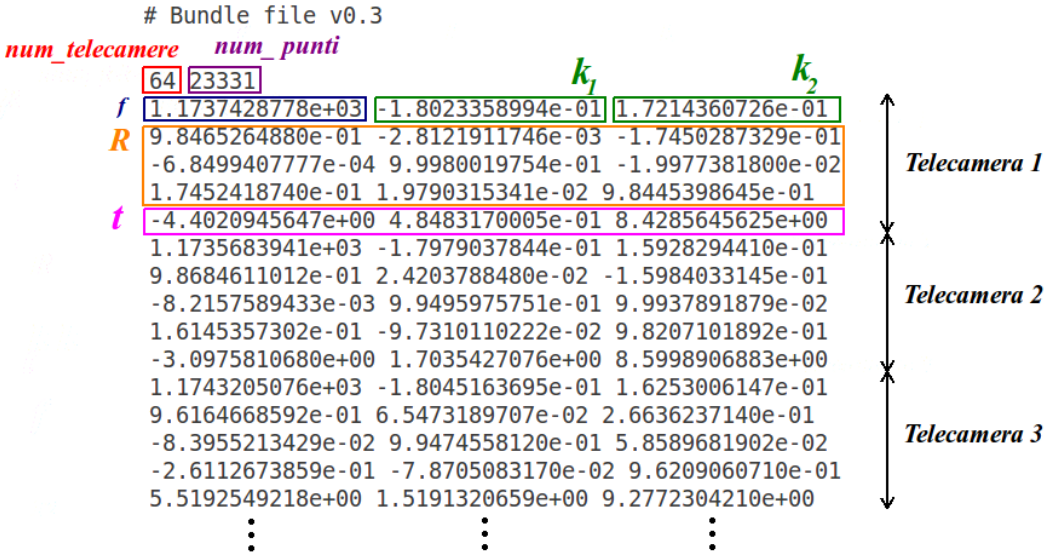


Figura 3.3: *Esempio file "bundle.out"(Telecamere)*

Per ogni punto, invece si definiscono:

<posizione>
<colore>
<lista_viste>

Con il termine posizione, si identifica un vettore di 3 componenti che descrive le coordinate x,y,z della posizione del vertice 3D che rappresenta un punto della scena. Con colore invece, si indica un vettore di 3 componenti che rappresenta appunto la tonalità RGB del vertice.
Infine con l'indicazione lista_viste si descrive una lista di immagini nelle quali il punto 3D in questione è visibile. La lista_viste comincia con il numero di telecamere nelle quali il punto è visibile (<num_viste>), successivamente è seguita da quattro parametri, come nella forma seguente:

<num_viste> <camera> <key> <x> <y>

dove: <camera>$\in \{0 \dots N_{CAM} - 1\}$ è l'indice della telecamera, <key> invece è l'indice della feature ottenuta tramite il programma SIFT, infine <x> e <y> sono le coordinate dell'immagine (ottenuta tramite la telecamera corrispondente) che indicano dove si trova questo pixel.
La posizione del pixel (numero floating point) viene descritta mediante un sistema di coordinate nelle quali l'origine è al centro dell'immagine. Così se indichiamo con w e h rispettivamente la larghezza e l'altezza dell'immagine, $(-w/2,\ -h/2)$ è il terzo quadrante mentre $(w/2,\ h/2)$ corrisponde al primo quadrante.

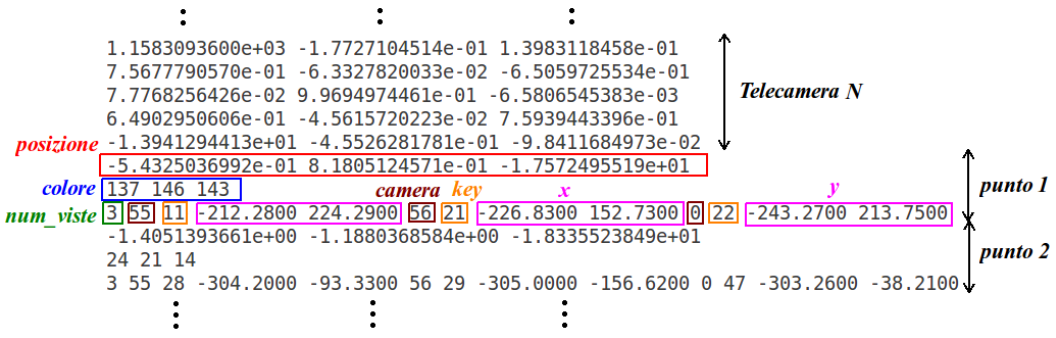


Figura 3.4: *Esempio file "bundle.out"(Vertici 3D)*

3.4.3 Rappresentazione della scena

I vertici 3D che rappresentano la scena 3D in esame, presenti nel file bundle.out, sono stati determinati utilizzando il modello della fotocamera a foro stenopeico (o *pinhole camera*).

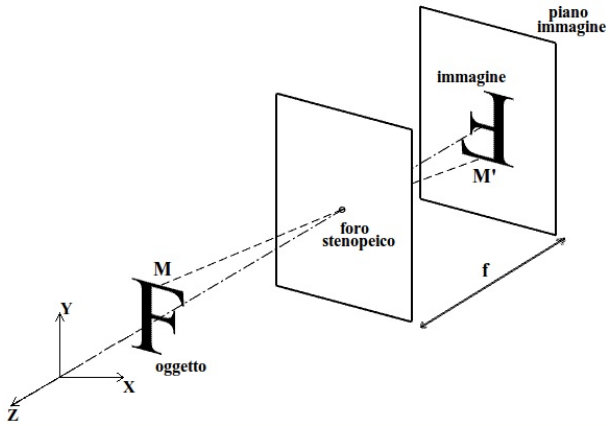


Figura 3.5: *modello della fotocamera a foro stenopeico*

Nel Paragrafo 2.1 avevamo analizzato nel dettaglio il modello reale di questo tipo di fotocamera, il quale, oltre a considerare la trasformazione prospettica, tiene conto anche dei parametri intrinseci ed estrinseci della fotocamera. Oltre a questi parametri, Bundler, per rappresentare un modello più accurato della fotocamera, tiene conto della distorsione radiale delle lenti, stimandone i coefficienti. Questo fatto risulta di fondamentale importanza soprattutto per ottiche a focale corta.
Ecco quindi che, oltre alla lunghezza focale f (parametri intrinseci), alla rotazione $\mathbf{R}$ e traslazione $\mathbf{t}$ della telecamera (parametri estrinseci), bisogna considerare nel modello anche k_1 e k_2 i coefficienti di distorsione radiale.
Per descrivere questo modello, ricaviamo la formula della proiezione 3D del punto $\mathbf{X}$ nella telecamera avente i seguenti parametri intrinseci ed estrinseci: f, $\mathbf{R}$ e $\mathbf{t}$. Per prima cosa consideriamo $\mathbf{X}$ come un punto generico, quindi espresso in coordinate mondo che possono anche non coincidere con quelle della telecamera, Fig. (3.6).

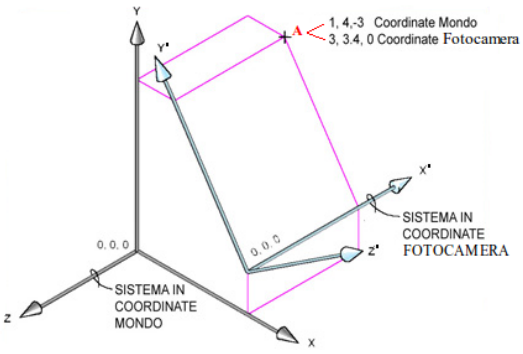


Figura 3.6: *Relazione tra le coordinate mondo e le coordinate della telecamera*

Ecco quindi che risulta necessario effettuare una conversione dal sistema mondo alle coordinate della telecamera, tale operazione viene effettuata moltiplicando il generico punto $\mathbf{X}$ con la matrice di rotazione $\mathbf{R}$, per poi sommare la traslazione $\mathbf{t}$ della teleca-

mera.
In forma algebrica:

$$\mathbf{M} = \mathbf{R} \cdot \mathbf{X} + \mathbf{t} \tag{3.2}$$

Successivamente, come già analizzato nel Paragrafo 2.2.1, per descrivere la proiezione del punto sulla pinhole camera bisogna effettuare la divisione prospettica:

$$\tilde{\mathbf{M}} = -f \frac{\mathbf{M}}{M_z} \tag{3.3}$$

dove f è la lunghezza focale e M_z è la terza coordinata di $\mathbf{M}$. Infine si considera la distorsione radiale moltiplicando $\tilde{\mathbf{M}}$ per $r(\tilde{\mathbf{M}})$, funzione che calcola il fattore di scala al fine di evitare tale distorsione. Questa funzione viene definita come:

$$r(\tilde{\mathbf{M}}) = 1 + k_1 \cdot ||\tilde{\mathbf{M}}||^2 + k_2 \cdot ||\tilde{\mathbf{M}}||^4 \tag{3.4}$$

si ottiene così:

$$\overline{\mathbf{M}} = r(\tilde{\mathbf{M}}) \cdot \tilde{\mathbf{M}} \tag{3.5}$$

$\overline{\mathbf{M}}$ descrive in pixel la proiezione del generico punto $\mathbf{X}$ sulla pinhole camera.

Capitolo 4

Mappe di Salienza

4.1 Premessa

Nel capitolo precedente, abbiamo descritto come risulta possibile, utilizzando il software Photosynth, ottenere un modello tridimensionale a partire da immagini digitali.
Questa tecnica per realizzare il modello tridimensionale (solitamente edifici, statue o monumenti, essendo pensata per realizzare percorsi virtuali turistici), necessita di un discreto quantitativo di immagini. Questo fatto, evidenziato anche nella tabella successiva, comporta un costo computazionale particolarmente elevato.

N° IMMAGINI	Tempo Computazionale	Vertici 3D
24 immagini	17’	8277
44 immagini	43’	16034
64 immagini	1h 23”	22939
84 immagini	2h 5’	31719
104 immagini	2h 42’	38198
124 immagini	3h 32”	45995
144 immagini	3h 55’	52423
164 immagini	4h 22’	59248
184 immagini	4h 54’	64127

Tabella 4.1: *Tempi e vertici 3D corrispondenti al numero di foto utilizzate per la ricostruzione tridimensionale della Porta Ognissanti (PC con processore i7 dual-core, 6Gb di RAM e sistema operativo Ubuntu 10.10)*

Oggigiorno per quasi tutte le applicazioni, comprese quelle che sviluppano modelli 3D, risulta fondamentale definire il giusto trade off tra costo computazionale e qualità.
A volte infatti per alcune applicazioni risulta opportuno avere un’alta qualità, a discapito quindi del tempo di elaborazione del modello, altre volte invece risulta necessario un basso costo computazionale, (soprattutto per applicazioni real time) consapevoli però che il risultato può non essere dei migliori.
Nel lavoro descritto in questa tesi si è cercato un metodo che permetta di ridurre il numero di foto necessario per realizzare il modello 3D (compatibilmente con comples-

sità geometrica della scena) in modo tale da abbassare il livello computazionale senza perdere troppo in qualità.
La scelta per ottenere quanto detto è ricaduta sulle mappe di salienza o saliency map. Queste mappe, che descriveremo in dettaglio nei paragrafi successivi, hanno la capacità di determinare i particolari salienti di una foto simulando i meccanismi della percezione visiva umana.
È plausibile quindi pensare di sfruttare queste tecniche per identificare l'oggetto protagonista di una determinata foto (che sarà anche quello di cui vogliamo realizzare il modello tridimensionale) associandogli una funzione di salienza che descriva l'importanza dei vari punti della scena.
Grazie a ciò risulta possibile definire una "bontà oggettiva"per le immagini. Sfruttando tale caratteristica, grazie a delle considerazioni che vedremo nel capitolo successivo (Paragrafo 5.1) è possibile effettuare dei particolari ordinamenti delle foto, ciò permette (se non si considerano quelle con salienza più bassa) di utilizzarne un numero inferiore, diminuendo così il costo computazionale senza perdere troppo in qualità.
Per capire bene il concetto di fondo nei prossimi paragrafi vedremo: per cominciare una breve descrizione su come le mappe di salienza simulano l'attenzione selettiva (Paragrafo 4.2), successivamente si analizzeranno due particolari tecniche, metodo *Itti & Koch* [21] (Paragrafo 4.3) e metodo *Wavelet* [22] (Paragrafo 4.4).

4.2 La percezione visiva

Nel linguaggio corrente, la salienza è una qualità o una condizione che identifica "l'emergere dalla massa": è saliente ciò che risalta rispetto a ciò che lo circonda. Le mappe di salienza, che descrivono proprio questo aspetto, sono utilizzate nella moderna *Scienza della Visione* per identificare alcuni aspetti della percezione visiva.
La percezione, che comunemente appare come qualcosa di assolutamente immediato, è il risultato di una serie di complessi processi di elaborazione che si realizzano in maniera del tutto automatica.
I primi processi di elaborazione visiva, vengono effettuati dai sistemi sensoriali che sono impegnati sia nella recezione dell'energia luminosa riflessa da un oggetto, sia nella trasduzione, ovvero nella conversione di questa energia fisica (mediante i recettori dell'occhio) in un segnale nervoso che viene trasmesso ai centri visivi del cervello. Questa prima elaborazione dell'informazione composta dall'attività di recezione, trasduzione e trasmissione è chiamata "sensazione".
Le operazioni appena descritte avvengono in maniera del tutto automatica, cioè senza che il soggetto ne sia consapevole o intervenga attivamente nella ricerca dell'informazione nell'ambiente. Una volta che il segnale nervoso raggiunge la corteccia cerebrale, l'informazione viene elaborata da neuroni che sono sensibili sia alle caratteristiche fisiche dello stimolo sia alle sue proprietà cognitive: in questa fase si attua il fondamentale processo della percezione.
Uno degli aspetti che caratterizza maggiormente la percezione visiva è il sovraccarico di informazioni. I sensori ottici generano segnali più o meno continui, provocando un

elevato numero di informazioni che risultano computazionalmente costose da elaborare per un sistema visuale biologico come l'occhio umano.
Il sistema nervoso quindi, è costretto a prendere delle decisioni su quali parti di tali informazioni sia necessaria una elaborazione più dettagliata e quali parti invece possano essere scartate. Inoltre tali stimoli selezionati devono avere una priorità, i più importanti devono essere i primi ad essere elaborati, ciò comporta un trattamento sequenziale delle diverse parti della scena visiva. Questo processo di selezione ed ordinamento si chiama attenzione selettiva (*selective attention*).
Ci si potrebbe chiedere cosa determina quali stimoli devono essere selezionati per l'elaborazione e quali invece debbano essere scartati. La risposta è che vi sono molti fattori che interagiscono e che contribuiscono a questa decisione.
Solitamente si distinguono tali fattori in due categorie: *bottom-up* e *top-down*.

4.2.1 Fattori Bottom-Up e Top-Down

Come già detto precedentemente, nel processo di identificazione di uno stimolo solitamente si possono distinguere due tipi di elaborazione: l'elaborazione "bottom-up"("dal basso verso l'alto") e l'elaborazione "top-down"("dall'alto verso il basso").
L'elaborazione "bottom-up "si basa sull'analisi delle parti che dipendono solo dall'istantaneo input sensoriale, l'elaborazione top-down invece si basa sulle esperienze e sulla storia dell'osservatore, cioè su dati che sono contenuti nella propria memoria.
La percezione richiede un'integrazione tra l'informazione sensoriale e le conoscenze possedute relative allo stimolo. Per capire bene come funziona la percezione, analizziamo la Fig. 4.1 che rappresenta l'opera di Salvador Dalì, "Soldier Take Warning"del 1942.

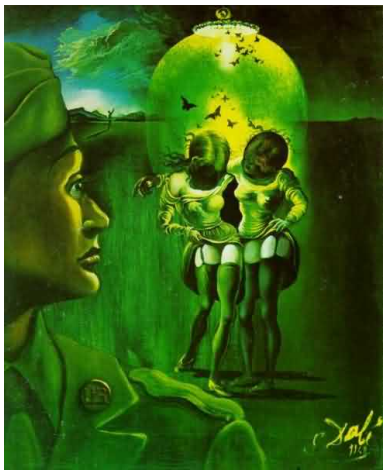


Figura 4.1: *Salvador Dalì "Soldier take warning", 1942*

La prima percezione che abbiamo dell'immagine è un soldato che guarda due ragazze che vengono illuminate da un lampione. Questa prima "impressione"viene ottenuta tramite l'elaborazione bottom-up, la quale semplicemente determina e analizza le informazioni contenute nello stimolo visivo. Se però aggiungiamo l'informazione che le due

ragazze nell'immagine formano un teschio, questa nozione guida l'osservatore ad identificare proprio tale oggetto che probabilmente precedentemente non era stato rilevato. Questo seconda percezione è stata ottenuta tramite l'elaborazione top-down, la quale utilizza le conoscenze acquisite dell'osservatore (in questo caso la presenza del teschio) per guidare l'elaborazione visiva a risultati che vengono ritenuti più significativi.
Vi sono principalmente due fattori che determinano l'utilizzo dell'una o dell'altra strategia di elaborazione.

1- IL GRADO DI CONOSCENZA
Il primo fattore è il grado di conoscenza che l'osservatore ha dell'oggetto in esame. Se l'osservatore ha una buona conoscenza dell'oggetto probabilmente impiegherà un'elaborazione top-down essendo guidato dalle conoscenze precedentemente acquisite. Se invece l'osservatore non conosce o ha informazioni parziali dell'oggetto in questione, dovrà procedere ad un'analisi delle sue caratteristiche (tramite elaborazione bottom-up) per ottenere la sua identificazione.

2- IL CONTESTO
Il secondo fattore che influenza il tipo di elaborazione impiegata nell'identificazione di un oggetto è il contesto in cui esso è inserito. Se l'elemento in esame è congruente con il contesto il riconoscimento dell'oggetto sarà probailmente determinato da un'elaborazione di tipo top-down. Infatti essendo l'oggetto in un contesto congruo, l'osservatore potrà effettuare delle ipotesi in base alle conoscenze acquisite che lo guideranno all'analisi della scena.
In caso contrario l'elaborazione sarà maggiormente guidata dalla semplice analisi delle caratteristiche che lo compongono (tecnica bottom up).
Se si osserva però la Fig. 4.2, la quale rappresenta l'opera di Salvador Dalì "Swans Reflecting Elephants"del 1937 si vedono dei cigni su uno stagno che riflettono degli elefanti.
In questo determinato contesto non ci si aspetta degli elefanti, quindi l'individuazione degli stessi è guidata da una elaborazione di tipo bottom-up.
Se si analizza bene l'immagine, si nota come questi elefanti altro non siano che il riflesso dei cigni e dei tronchi degli alberi, quindi potrebbe sembrare un controsenso non aver rilevato tali riflessi essendo situati in un contesto congruo.
In realtà per riconoscere i cigni, è necessario effettuare una rotazione mentale di 180° degli stessi, in modo da farli coincidere con quelli riflessi nello stagno.
Non essendo per il sistema nervoso una cosa così banale ed immediata, ad una prima percezione prevale l'elaborazione bottom-up.

4.2.2 L'attenzione selettiva a livello computazionale

Da quanto visto precedentemente appare dunque evidente come la percezione oltre ad analizzare le parti che sono presenti nello stimolo tramite i dati sensoriali (elaborazione bottom-up), sia strettamente legata ad altre funzioni cognitive quali l'attenzione, la

Figura 4.2: *Salvador Dalì "Swans Reflecting Elephants", 1937*

memoria e l'immaginazione (elaborazione top-down).
Data la difficoltà di misurare accuratamente o addirittura quantificare gli stati interni dell'osservatore, si è deciso di studiare gli aspetti che riguardano il controllo dell'attenzione che sono indipendenti da questi, ovvero i fattori bottom-up che risultano quindi molto più facili da individuare.

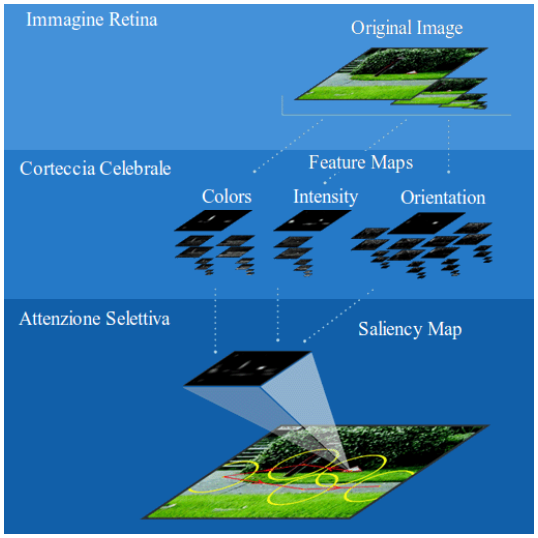


Figura 4.3: *Metodo bottom-up sviluppato da Koch e Ullman*

Il tentativo più influente per la comprensione dei meccanismi neurali dell'attenzione bottom-up è stata fatta da *Christof Koch* e *Shimon Ullman* [23], i quali propongono che le diverse caratteristiche visive che contribuiscono alla selezione di uno stimolo (colore, orientamento, movimento, ecc...) vengano combinate in un singola mappa, la mappa di salienza appunto (Fig. 4.3). Questa mappa integra le informazioni normalizzate provenienti dalle mappe delle caratteristiche individuali (denominate feature map), al fine di individuare una misura globale dell'attenzione selettiva che, come già detto,

seleziona e definisce una priorità per gli stimoli. Le posizioni più salienti di una scena sono quindi le posizioni più probabili nelle quali ricade l'attenzione visiva e saranno determinate principalmente dal modo in cui queste si differenziano dal contorno per colore, orientamento, movimento e profondità.
La mappa di salienza quindi sarà composta da pixel $p'(x, y)$ ai quali è associato un valore discreto di salienza (solitamente 0-255) corrispondenti ai pixel $p(x, y)$ dell'immagine originale. Nella Fig. 4.4 si può vedere un esempio di mappa di salienza.

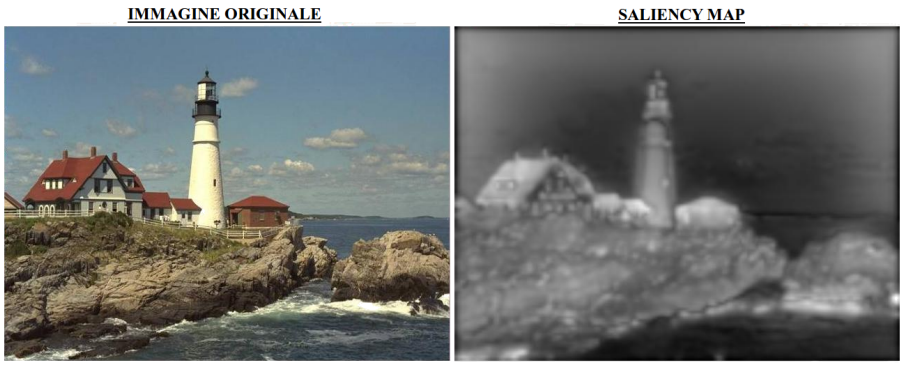
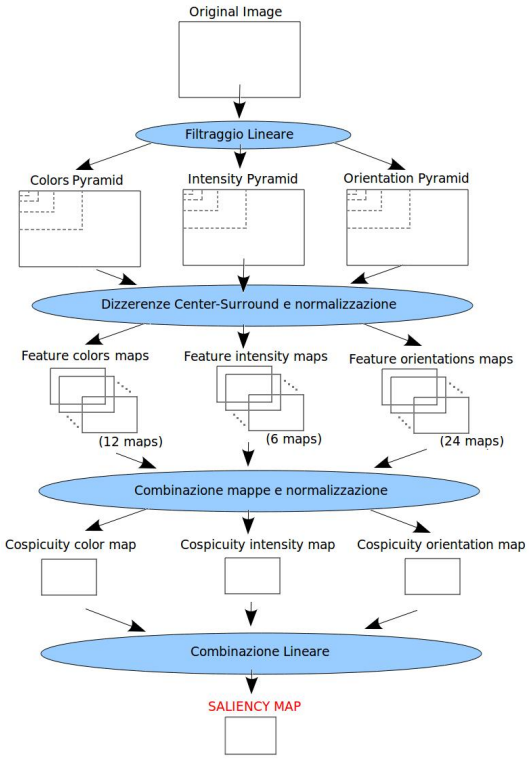


Figura 4.4: *Esempio mappa di salienza*

Risulta facile notare come le parti più salienti dell'immagine siano principalmente: il faro, che è al centro della scena e risulta ben illuminato, ed i tetti delle case, i quali essendo di colore rosso si distinguono in maniera evidente dai colori della scena circostante. Quanto detto va in accordo se si considerano le componenti impiegate per il calcolo della saliency map (intensità, colore ed orientazione).
Dopo questa panoramica sulla percezione visiva ed in particolare su come funziona l'attenzione selettiva, risulta facile capire come le mappe di salienza possano essere utilizzate per identificare le informazioni più importanti dei flussi ottici. Sfruttando tale caratteristica risulta possibile migliorare le prestazioni nella generazione o nella trasmissione di dati visivi.
Questo è proprio il nodo fondamentale sviluppato in questa tesi per migliorare le prestazioni di Photosynth. Prima però di analizzare il modo in cui le saliency map sono state utilizzate per questa applicazione, andiamo a vedere nel dettaglio due particolari tecniche che ci permettono di ottenerle.

4.3 Metodo Itti & Koch

4.3.1 Introduzione



Il primo metodo che andiamo ad analizzare è quello proposto da Itti & Koch. Questo metodo è stato sviluppato basandosi sui concetti riguardanti i meccanismi neurali definiti da *Koch* e *Ullman* nel 1985.
L'algoritmo è legato alla cosiddetta *Feature Integration Theory* (FIT)[24], proprosta per spiegare la strategia di ricerca visiva umana. In questa tecnica, l'input visivo (immagine RGB di dimensioni 640×480) viene suddiviso in immagini multiscala che descrivono le varie componenti percettive, ovvero: intensità, colore ed orientazione. Per ognuna di queste componenti, viene creata una mappa (feature map) che ne descrive la salienza. La salienza viene definita in modo tale che solo i punti che a livello locale si distinguono dai circostanti persistono.
Una volta determinate queste mappe, vengono inglobate mediante tecniche bottom-up (dal basso verso l'alto) in un'unica mappa, la mappa di salienza appunto (immagine a scala di grigi di dimensioni 40×30), che ha il compito di descrivere la *selective attention* umana.
La complessa analisi di una scena viene realizzata mediante questo procedimento, il quale risulta plausibile e computazionalmente efficiente. Inoltre, permette una rapida selezione delle posizioni salienti di una scena, in modo tale che se si vuole un'analisi più approfondita è possibile concentrarsi solo su queste, riducendo di molto il costo computazionale.
Per facilitare la descrizione delle operazioni che devono essere effettuate per ottenere la mappa di salienza, vediamo ora una piccola analisi sul *filtraggio piramidale*.

Il filtraggio Piramidale

Il filtraggio piramidale [25] (Fig. 4.5) prende un'immagine di input e genera 3 piramidi di immagini a risoluzione decrescente. Le piramidi a seconda del tipo di elaborazione, sono di 3 tipi: gaussiana, laplaciana e quella che descrive l'orientazione.
Ognuna delle tre piramidi è costituita da N immagini (altresì denominate stadi o livelli), per la piramide gaussiana il primo di questi (livello 0) è costituito dall'immagine originale. In ognuno degli stadi successivi invece si effettua un filtraggio passa-basso dello stadio precedente ed un campionamento.

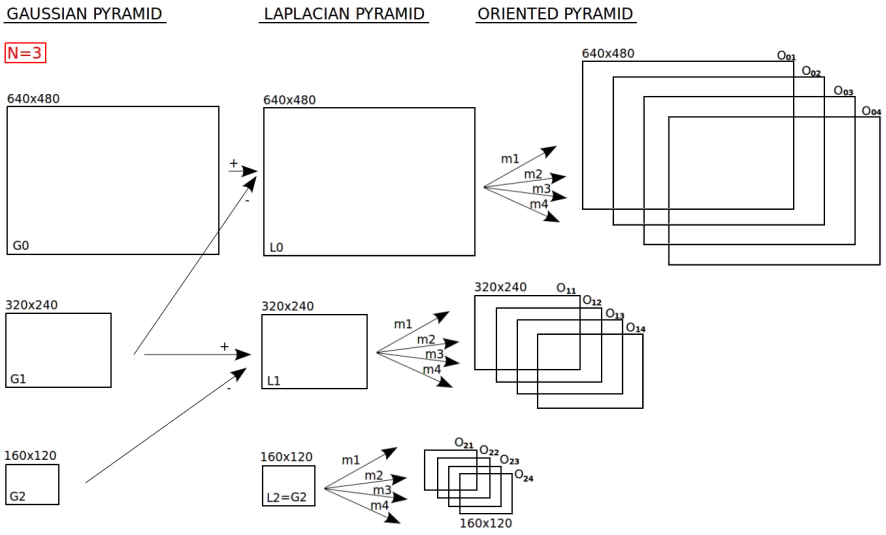


Figura 4.5: *Esempio dello schema a filtraggio piramidale nel caso in cui N=3*

Nella piramide laplaciana invece, ogni livello N-esimo consiste in un filtraggio passa-banda dell'immagine in ingresso, la quale viene ottenuta mediante la sottrazione tra i due corrispondenti livelli adiacenti della piramide gaussiana.
Per comodità, denomineremo G_0 l'immagine originale in ingresso, $G_1, \ldots, G_N$ la versione filtrata mediante il filtro passa-basso con risoluzione decrescente (stadi gaussiani) ed infine la corrispondente versione passa-banda $L_0, \ldots, L_N$ (stadi laplaciani).
Le equazioni che descrivono le immagini della piramide gaussiana e laplaciana utilizzando le denominazioni definite in precedenza, sono le seguenti:

$$G^0_{n+1} = W * G_n \qquad L_n = G_n - G^0_{n+1} \qquad G_{n+1} = SubsampledG^0_{n+1} \tag{4.1}$$

Con il termine W si indica il filtro passa-basso. Abbiamo scelto di utilizzare un filtro gaussiano separabile con 5 campioni [1/16, 1/4, 3/8, 1/4, 1/16], è possibile notare che il filtro risulta normalizzato (somma dei coefficienti pari a 1).
Per concludere, il filtraggio piramidale prevede la definizione della piramide orientazione. Questa viene ottenuta grazie alla modulazione di ogni livello della piramide laplaciana mediante un set di segnali seno, seguiti poi da un altro filtro passa-basso. Le operazioni che la descrivono, sono definite dall'equazione successiva:

$$O_{n\alpha} = LPF[e^{j\,\vec{k}_a \cdot \vec{r}} L_n[x, y]] \tag{4.2}$$

nell'equazione precedente (Eq. 4.2) indichiamo con:

- $O_{n\alpha}$ l'immagine orientata alla scala n-esima e con orientazione α
- $\vec{r} = x\,\vec{i} + y\,\vec{j}$ (dove x e y sono le coordinate dell'immagine laplaciana e l'origine di x e y è presa al centro di tale immagine)
- $\vec{k}_a = (\pi/2) cos\ \theta_\alpha\,\vec{i} + sen\ \theta_\alpha\,\vec{j} \qquad$ con $\quad \theta_\alpha = (\pi/\widehat{N})(\alpha - 1)$

Per la realizzazione della scala orientazione abbiamo utilizzato $\widehat{N} = 4$ quindi è come se ogni livello della piramide laplaciana fosse modulato dalle seguenti sinusoidi complesse:

$$m_1(x, y) = e^{j(\pi/2)x} \tag{4.3}$$

$$m_2(x, y) = e^{j(\pi\sqrt{2}/4)(x+y)} \tag{4.4}$$

$$m_3(x, y) = e^{j(\pi/2)y} \tag{4.5}$$

$$m_4(x, y) = e^{j(\pi\sqrt{2}/4)(y-x)} \tag{4.6}$$

Queste 4 differenti modulazioni differiscono solamente nelle orientazioni, che sono: $0°, 45°, 90°$ e $135°$.

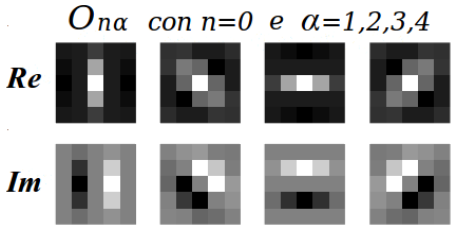


Figura 4.6: *$O_{n\alpha}$ dimensione 5x5*

Dopo la modulazione, le immagini laplaciane sono filtrate tramite passa-basso (lo stesso descritto per la piramide gaussiana), a questo punto sono ettivamente filtrate da un set di filtri.
Questi tipi di filtri sono denominati *Gabor-Filter* e vengono definiti dalla seguente risposta impulsiva:

$$\psi_k(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2} \times m_k(x, y) \qquad k = 1, 2, 3, 4 \tag{4.7}$$

4.3.2 Descrizione del modello

Estrazione delle feature map

Il metodo di Itti & Koch prevede in ingresso un'immagine digitale a colori con risoluzione 640×480. Indichiamo con r,g,b le componenti di rosso, giallo e blu di questa immagine digitale, da queste risulta possibile determinare l'immagine intensità come: $I = (r + g + b)/3$.
L'immagine I viene poi utilizzata per creare la piramide gaussiana $I(\sigma)$ con $\sigma \in [0, \ldots, 8]$. Dai valori che assume σ è facile capire che la piramide è composta da 9 stadi. Come detto nel paragrafo precedente, successivamente al filtraggio passa-basso, si effettua un campionamento dell'immagine I con fattore 2. Vi è quindi una riduzione dell'immagine sia orizzontalmente che verticalmente da un rapporto $1 : 1$ (Scala 0) a

$1:256$ (Scala 8).
Le componenti r, g e b vengono poi normalizzate rispetto ad I, in modo da eliminare la relazione tra colore ed intensità.
Poiché le variazioni di colore a bassa luminanza non sono particolarmente salienti essendo difficilmente percepibili, la normalizzazione viene applicata solo nei punti in cui il valore di I è maggiore di $1/10$ del massimo calcolato nell'intera immagine. Nei punti in cui questa condizione non è verificata i valori di r,g, e b sono posti a zero.
Successivamente, altre quattro componenti di colore vengono calcolate, in particolare:

- $R = b - (r + g)/2$ per il rosso
- $G = g - (r + g)/2$ per il verde
- $B = b - (r + g)/2$ per il blue
- $Y = (r + g)/2 - |r - g|/2 - b$ per il giallo

I punti di queste componenti che assumono valori negativi sono posti a zero. Anche in questo caso per queste componenti si creano le piramidi gaussiane $R(\sigma)$, $G(\sigma)$, $B(\sigma)$ e $Y(\sigma)$ con $\sigma \in [0, \ldots, 8]$.
Come abbiamo già detto nella descrizione generale del metodo Itti, una volta definite le varie componenti risulta necessario definire delle feature map che descrivono la salienza individuale di ogni componente.
Il metodo utilizzato da Itti per determinare le feature map si basa sul meccanismo *center-surround*. Tipicamente i neuroni sono più sensibili (si attivano) nelle piccole regioni centrali dello spazio visivo (centro o *center*), mentre nella regione concentrica antagonista alla zona centrale (contorno o *surround*) vi è una inibizione della risposta neurale.
Il center-surround viene implementato nel modello come la differenza tra scale diverse: il centro è un pixel alla scala $c \in \{2, 3, 4\}$ e il contorno è il corrispondente pixel alla scala $s = c + \delta$ con $\delta \in \{3, 4\}$.
Questa differenza tra mappe con scale diverse, (indicata con "$\ominus$"), sarà ottenuta interpolando la mappa di scala minore alle dimensioni di quella maggiore, per poi effettuare una sottrazione punto punto.
Il primo set di feature map che andiamo a considerare è quello che caratterizza l'intensità. Solitamente l'attenzione visiva dell'osservatore è catturata sia da punti centrali bui su contorni luminosi, che da punti centrali luminosi su contorni bui.
Entrambe le percezioni vengono calcolate simultaneamente mediante un set di sei mappe $I(c, s)$ con $c \in \{2, 3, 4\}$ e $s = c + \sigma$ con $\sigma \in \{3, 4\}$, descritte dalla seguente relazione:

$$I(c, s) = |I(c) \ominus I(s)| \tag{4.8}$$

Per le componenti colore, le feature map vengono calcolate similmente a quanto fatto per l'intensità. Il sistema utilizzato viene denominato *double-opponent*. Al centro del campo recettivo i neuroni sono eccitati da un colore (ad esempio il rosso) e inibiti da un

altro (ad esempio il verde), mentre il contrario è vero per il contorno. Nella percezione visiva umana, tale condizione vale per la coppia di colori rosso/verde, e verde/rosso, blu/giallo e giallo/blu.
In accordo con quanto detto, le mappe $RG(c, s)$ vengono create per l'opposizione della coppia rosso/verde e verde/rosso, lo stesso vale per le mappe $BY(c, s)$ che valgono per il blu/giallo e giallo/blu. Le equazioni che descrivono queste opposizioni di colore sono le seguenti:

$$RG(c, s) = (R(c) - G(c)) \ominus (G(s) - R(s)) \tag{4.9}$$

$$BY(c, s) = (B(c) - Y(c)) \ominus (Y(s) - B(s)) \tag{4.10}$$

L'ultima componente che viene considerata nel metodo Itti & Koch per il calcolo della saliency map è quella di orientazione. L'informazione sull'orientazione locale viene ottenuta da I utilizzando la piramide di orientazione Gabor $O(\sigma, \theta)$ dove $\sigma \in 0, \ldots 8$ e $\theta \in \{0°, 45°, 90°, 135°\}$.
I filtri di Gabor già descritti nel filtraggio piramidale approssimano il campo recettivo dei neuroni rispetto l'orientazione, nella corteccia visiva primaria.
Le feature map per la componente orientazione $O(c, s, \theta)$ sono quindi determinate come:

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)| \tag{4.11}$$

In definitiva vengono determinate ben 42 feature map: 6 per l'intensità, 12 per il colore e 24 per l'orientazione.

Estrazione delle Cospicuity Map

Lo scopo delle saliency map, come già ripetuto più volte, è rappresentare la salienza in ogni posizione del campo visivo tramite una quantità scalare, guidando così la selezione di punti importanti nell'immagine in base alla distribuzione spaziale delle saliency.
Una combinazione delle feature map permette la costruzione della saliency map modellata come una rete neurale dinamica.
Una difficoltà nel combinare diverse feature map è che rappresentano componenti non comparabili, con range dinamici e meccanismi di estrazione differenti. Inoltre c'è da considerare il fatto che alcuni oggetti che appaiono estremamente salienti in alcune mappe possono essere mascherati dal rumore o da oggetti con minor salienza che però sono presenti in numerose mappe.
In assenza di una supervisione top-down, è necessario effettuare una normalizzazione delle mappe, in questo metodo si utilizza l'operatore di normalizzazione $N(\cdot)$.
Dalla Fig. 4.7 risulta possibile notare come tale operatore enfatizzi le mappe nelle quali è presente un ridotto numero di valori elevati, mentre globalmente sopprima le mappe che contengono un numero comparabile di picchi.

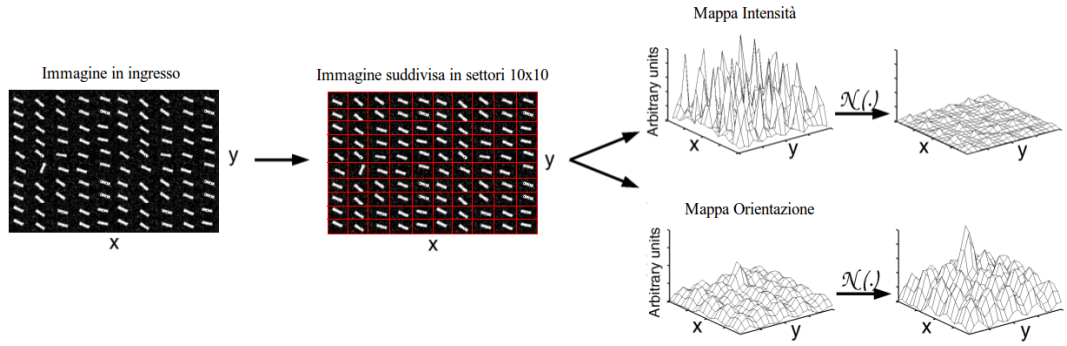


Figura 4.7: *Normalizzazione utilizzata dal metodo Itti & Koch*

In definitiva, $N(\cdot)$ effettua le seguenti operazioni:

1- Normalizza i valori della mappa su di un range fisso $[0, \ldots, M]$ in modo da eliminare la possibile differenza di ampiezza tra mappe che rappresentano componenti differenti.

2- Trova la posizione del massimo globale nella mappa M e determina la media $\overline{m}$ di tutti gli altri massimi locali (la mappa $m \times n$ viene suddivisa in 100 settori di dimensioni $(m/10) \times (n/10)$).

3- Moltiplica i valori dell'intera mappa per $(M - \overline{m})^2$

Nelle operazioni effettuate da $N(\cdot)$ appena descritte si nota come vengano considerati solo i massimi locali in modo da confrontare solo i valori associati ai "punti di attivazione", ignorando invece le aree omogenee.
Confrontando il valore massimo M dell'intera mappa con la media dei massimi locali si determina la differenza tra il maggior punto di "attivazione"e la media di questi punti. Quando questa differenza è alta, il valore di "attivazione"maggiore si distingue fortemente e quindi si enfatizza la mappa, se invece tale differenza risulta piccola, la mappa viene soppressa supponendo che non vi sia nulla di così evidente. La motivazione biologica dell'utilizzo di $N(\cdot)$ è che tale operatore replica il meccanismo center-surround di inibizione laterale.
Le feature map normalizzate, vengono poi combinate per ogni componente in un'unica mappa alla scala ($\sigma = 4$) di dimensioni 40×30, tale mappa viene anche denominata cospicuity map. Si ottengono quindi 3 cospicuity map: $\overline{I}$ per l'intensità, $\overline{C}$ per il colore e $\overline{O}$ per l'orientazione.
Tutte queste mappe sono ottenute tramite un'addizione tra feature map con scale differenti ("$\oplus$"), tale operazione viene effettuata riducendo ogni mappa alla scala 4 per poi effettuare una addizione punto-punto.
Le relazioni che descrivono le cospicuity map, sono le seguenti:

$$\overline{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(I(c,s)) \tag{4.12}$$

$$\overline{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))] \tag{4.13}$$

$$\overline{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} N\left(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(I(c,s)) \right) \tag{4.14}$$

La motivazione per la quale si creano tre canali separati, $\overline{I}$, $\overline{C}$ e $\overline{O}$ effettuando una normalizzazione individuale, si basa sull'ipotesi che componenti simili competono fortemente per la saliency e che i diversi contributi sono indipendenti ai fini della creazione della mappa di salienza.

Estrazione della Saliency Map

Una volta ottenute le tre cospicuity map, $\overline{I}$, $\overline{C}$ e $\overline{O}$, risulta possibile ottenere la saliency map desiderata mediante la relazione:

$$S = \frac{1}{3}(N(\overline{I}) + N(\overline{C}) + N(\overline{O})) \tag{4.15}$$

Si effettua quindi una media dei valori normalizzati delle cospicuity map ottenendo in uscita un'immagine a scala di grigi allo stadio $\sigma = 4$ (dimensioni 40×30). In ogni momento, il massimo della saliency map (SM) definisce i punti più salienti dell'immagine, nei quali l'attenzione focale (f*ocus of attentio*n, FOA) dovrebbe essere diretta.
A questo punto la rete neurale *Winner Take All* (WTA) [26] che descrive il criterio di selezione di uno stimolo, esplora la saliency map e sequenzialmente segnala gli oggetti candidati in ordine di importanza. Si sceglie quindi il punto più saliente, lo si segnala ed infine lo si inibisce tramite un meccanismo denominato *Inibition of Return* (IOR), e si continua nella ricerca dei successivi punti salienti con il medesimo procedimento.

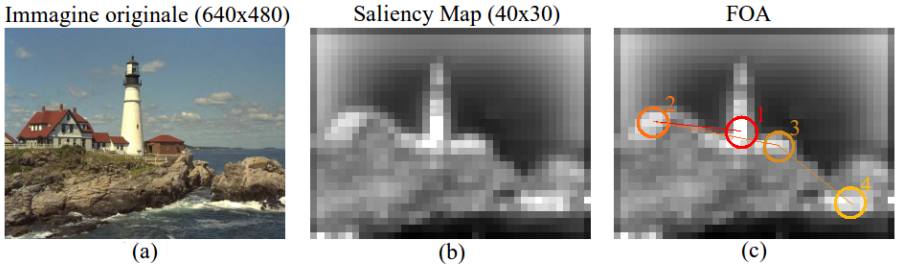


Figura 4.8: *Esempio di estrazione della Saliency Map: (a) Immagine originale, (b) Saliency Map ottenuta , (c)Descrizione della FOA*

Nell'immagine di Fig. 4.8.b è possibile vedere un esempio di saliency map ottenuta mediante il metodo Itti & Koch. Nella Fig. 4.8.c è mostrato invece il metodo WTA che descrive la Focus of Attention. Il primo candidato che viene individuato (1) è il faro nel quale la salienza è più elevata e per prima ricadrà l'attenzione dell'osservatore, una volta inibito lo sguardo ricadrà sui tetti delle case (2) e (3) i quali hanno una componente di

colore molto accesa rispetto al contorno, per finire l'attenzione si sposterà sulle onde del mare (4) le quali hanno una elevata componente d'intensità.

4.3.3 Modifica al metodo di Itti & koch

Abbiamo visto nella descrizione del metodo Itti & Koch come in ingresso si richiedesse un'immagine digitale a colori di dimensioni 480×320 ottenendo in uscita una saliency map alla scala $\sigma = 4$, di dimensioni 40×30.
Per lo sviluppo dei metodi di ordinamento, che vedremo nel capitolo successivo, e per una maggiore flessibilità dell'algoritmo, abbiamo deciso di apportare alcune modifiche che consentono di sfruttarlo con immagini di qualsiasi dimensione ed allo stesso tempo di ottenere in uscita una mappa di salienza della stesse dimensioni dell'immagine in ingresso.
Per ottenere tale risultato come prima cosa abbiamo dovuto prestare particolare attenzione al riscalamento delle immagini nel filtraggio piramidale, poiché nel campionamento dei vari stadi è possibile non avere immagini multiple di 2. Inoltre, nella estrazione delle Cospicuity Map, che come visto si ottiene effettuato un'addizione tra feature map con scale differenti, abbiamo utilizzato una nuova operazione ("$\otimes$"), la quale interpola ogni mappa alla scala 1 per poi effettuare una addizione punto-punto.
Le relazioni ottenute sono del tutto similari a quelle ottenute in precedenza:

$$\overline{I} = \bigotimes_{c=2}^{4} \bigotimes_{s=c+3}^{c+4} N(I(c,s)) \tag{4.16}$$

$$\overline{C} = \bigotimes_{c=2}^{4} \bigotimes_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))] \tag{4.17}$$

$$\overline{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} N\left(\bigotimes_{c=2}^{4} \bigotimes_{s=c+3}^{c+4} N(I(c,s)) \right) \tag{4.18}$$

Nell'immagine successiva (Fig. 4.9) vediamo due esempi di estrazione della saliency map con il metodo di Itti & Koch modificato. In ingresso abbiamo due immagini di dimensioni 1504×1000, in uscita tre cospicuity map ed una mappa di salienza delle stesse dimensioni.
Nella immagine in alto è raffigurata la Porta Ognissanti a Padova: come risulta possibile notare dalla saliency map vi è grande salienza nell'entrata (essendo di colore scuro su contorno chiaro) e nel cielo che presenta una elevata illuminazione.
Nell'immagine in basso invece è raffigurata una pianta in laboratorio. In questo caso, sempre dalla saliency map, si nota come la salienza sia concentrata proprio sulla pianta, la quale possiede un colore evidente rispetto al contorno.
La differenza principale tra le due immagini, al di là della scena rappresentata, è che: una è stata scattata all'aperto e quindi è affetta da tutte quelle componenti ambientali che deviano la salienza (come la luminosità ed il traffico), l'altra invece è stata scattata in un ambiente chiuso e quindi la scena subisce meno tali fattori.

Il risultato è che se nell'immagine della pianta si riesce ad individuare con estrema salienza l'oggetto principale, in quella che rappresenta la Porta Ognissanti ciò risulta possibile solo per alcune parti.

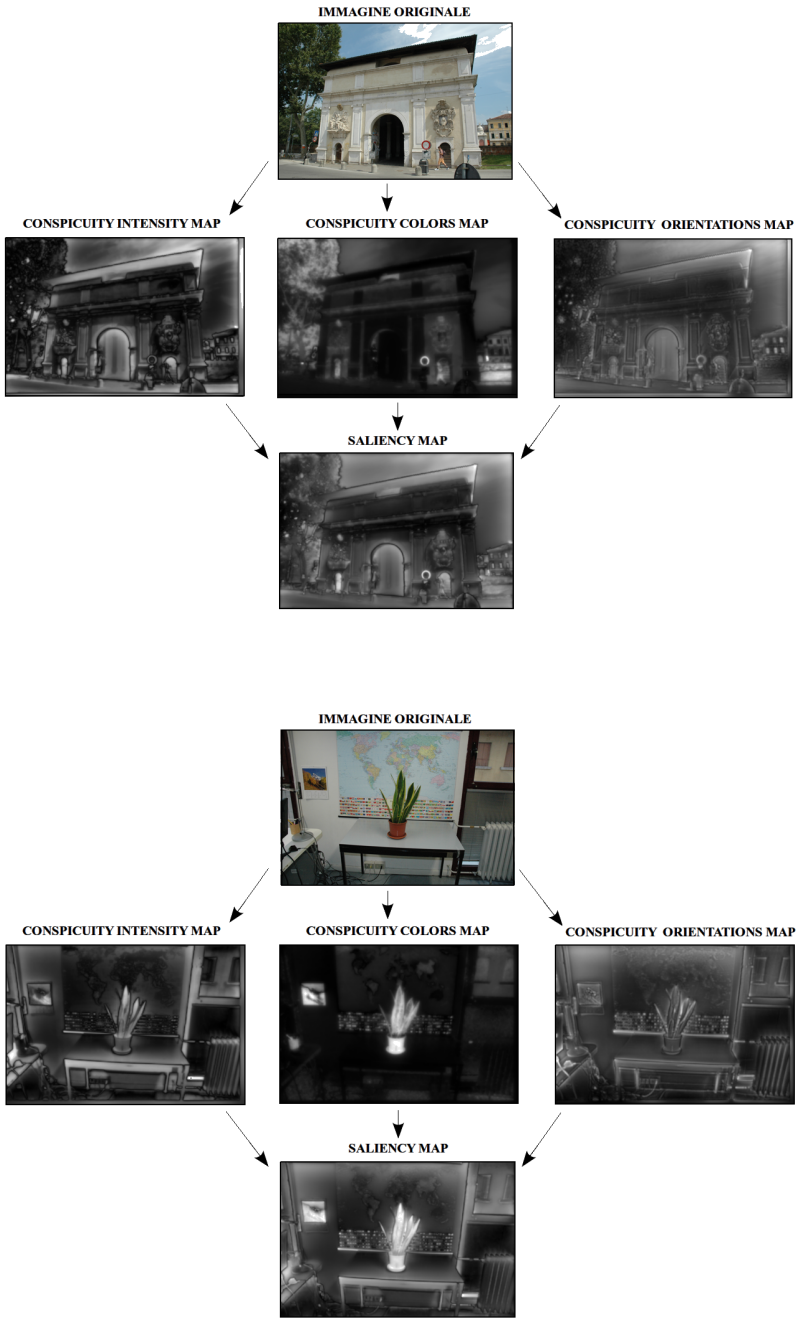


Figura 4.9: *Esempio di estrazione di saliency map in un ambiente aperto (a) ed in un ambiente chiuso (b)*

4.4 Metodo Wavelet

4.4.1 Introduzione

Dopo aver visto nel dettaglio il metodo di Itti & Koch, analizziamo ora un altro metodo basato su wavelet che consente una efficiente determinazione delle mappe di salienza. Basandosi su tale teoria e su quella che descrive la multirisoluzione, risulta possibile mimare il processo center-surround della percezione visiva umana. Tale metodo utilizza la decomposizione tramite wavelet per determinare le componenti che attirano l'attenzione visiva, come: colore, intensità e orientazione. Queste componenti sono fondamentali per la creazione delle feature map, le quali una volta determinate mediante una particolare funzione vengono combinate per produrre la mappa di interesse, la saliency map finale.

4.4.2 Descrizione del modello

Il metodo Wavelet segue la filosofia computazionale proposta da Itti & Koch basata sulla Feature Integration Teory, descritta nel paragrafo precedente.
In questa tecnica però si utilizza il modello a colori $YCrCb$ (invece del classico RGB) e la *decomposizione wavelet* di *Mallat* [27], per permettere un'efficiente via per il calcolo delle saliency map di immagini statiche e sequenze video.
In un'immagine f a colori, rappresentata usando il modello $YCrCb$, mediante il canale Y che corrisponde alla luminanza si possono identificare regioni di interesse in base alle componenti di illuminazione e orientazione, invece tramite Cr (Crominanza Rosso) e Cb (Crominanza Blu) che corrispondono alle componenti cromatiche si possono identificare le regioni di interesse in base alla componente colore.
Anche in questo caso, come nel metodo di Itti & Koch, le aree salienti vengono determinate su scale diverse in base alle componenti di intensità, orientazione e colore. In questo modo dalle feature map possono essere rilevati oggetti rilevanti di diverse misure.
Combinando il risultato delle feature map con scale differenti si ottengono le cospicuity map per l'intensità C_I, orientazione C_O e colore C_C.
La motivazione della creazione di cospicuity map separate è data dall'ipotesi secondo la quale componenti simili competono fortemente per la saliency mentre quelle differenti contribuiscono indipendentemente.
Una volta calcolate le cospicuity map, vengono normalizzate e sommate utilizzando una funzione "sigma"di saturazione che preserva tale indipendenza. Si ottiene così un'unica mappa, la saliency map desiderata.

Estrazione delle Feature Map

Al fine di realizzare la tecnica multiscala, vengono applicati un filtro passa basso $h_\phi(\cdot)$ e un filtro passa alto $h_\psi(\cdot)$ ad ognuna delle tre componenti: Y,C_r e C_b sia nella direzione verticale che in quella orizzontale.

Il risultato in uscita al filtro viene poi campionato di un fattore due, in modo da generare:

- 3 bande passa alto
  H (coefficienti verticali)
  V (coefficienti orizzontali)
  D (coefficienti diagonali)

- A sottobanda passa basso

Il medesimo processo è poi ripetuto per la sottobanda A, al fine di generare il livello successivo della decomposizione.
Le seguenti equazioni descrivono matematicamente l'intero preocesso per il canale luminanza Y, ovviamente lo stesso procedimento è applicato anche ai canali cromatici C_r e C_b:

$$\begin{aligned}
Y_A^{-(j+1)}(m,n) &= \left(h_\phi(-m) * \left(Y_A^{-j}(m,n) * h_\phi(-n)\right) \downarrow^{2n}\right) \downarrow^{2m} \\
Y_H^{-(j+1)}(m,n) &= \left(h_\psi(-m) * \left(Y_A^{-j}(m,n) * h_\phi(-n)\right) \downarrow^{2n}\right) \downarrow^{2m} \\
Y_V^{-(j+1)}(m,n) &= \left(h_\phi(-m) * \left(Y_A^{-j}(m,n) * h_\psi(-n)\right) \downarrow^{2n}\right) \downarrow^{2m} \\
Y_D^{-(j+1)}(m,n) &= \left(h_\psi(-m) * \left(Y_A^{-j}(m,n) * h_\psi(-n)\right) \downarrow^{2n}\right) \downarrow^{2m}
\end{aligned} \tag{4.19}$$

Nelle equazioni precedenti, con $*$ si identifica la convoluzione, con $Y_A^{-j}(m,n)$ l'approssimazione del canale luminanza Y al livello j-esimo (ovviamente $Y_A^{-0}(m,n) = Y$), infine $\downarrow^{2m}$ e $\downarrow^{2n}$ definiscono il campionamento di un fattore due rispetto le righe e colonne.
Una volta eseguita la decomposizione ad un determinato livello, per ogni canale Y, Cr, e Cb, si utilizzano le differenze center-surround per migliorare le regioni che localmente si distinguono dal contorno.
Queste differenze sono effettuate su una particolare scala (livello j-esimo) utilizzando il gradiente morfologico (differenza tra l'apertura e chiusura morfologica).
Utilizzando tale tecnica si ottengono le feature map dell'intensità, del colore e dell'orientazione (ottenuta come somma delle differenze tra le bande ottenute in precedenza), le relazioni che descrivono queste mappe sono le seguenti:

$$I^{-j}(m,n) = Y_A^{-j}(m,n) \bullet b - Y_A^{-j}(m,n) \circ b \tag{4.20}$$

$$O^{-j}(m,n) = |Y_D^{-j}(m,n) - Y_H^{-j}(m,n)| + |Y_D^{-j}(m,n) - Y_V^{-j}(m,n)| + |Y_V^{-j}(m,n) - Y_H^{-j}(m,n)| \tag{4.21}$$

$$CR^{-j}(m,n) = Cr_A^{-j}(m,n) \bullet b - Cr_A^{-j}(m,n) \circ b \tag{4.22}$$

$$CB^{-j}(m,n) = Cb_A^{-j}(m,n) \bullet b - Cb_A^{-j}(m,n) \circ b \tag{4.23}$$

$$C^{-j} = CR^{-j} + CB^{-j} \tag{4.24}$$

Nelle equazioni $I^{-j}(m,n)$, $O^{-j}(m,n)$ e $C^{-j}(m,n)$ identificano le feature map rispettivamente dell'intensità, orientazione e colore determinate alla scala j-esima, mentre $\bullet$ e $\circ$ identificano l'operatore di *chiusura* e *apertura*. Vediamo come vengono definiti questi due operatori:

- APERTURA: Operatore che smussa il contorno di un oggetto, rompe piccoli canali di connessione ed elimina piccole protuberanze.

$$A \circ B = (A \ominus B) \oplus B = \bigcup \{(B)_z : (B)_z \subseteq A\}$$

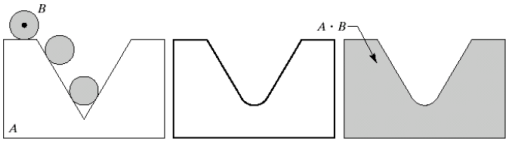


Figura 4.10: *Esempio apertura*

dove:

$(A)_z$: Traslazione di un immagine A per mezzo di un punto z=(z_1,z_2) $\Rightarrow$ $(A)_z = \{b + z : b \in A\}$

$\widehat{A}$: Riflessione di un immagine $A \Rightarrow \widehat{A} = \{-b : b \in A\}$

$A \ominus B$: Erosione di A per mezzo di B $\Rightarrow A \ominus B = \{z : (B)_z \cap A^c = \emptyset\}$

$A \oplus B$: Dilatazione di A per mezzo di B $\Rightarrow A \oplus B = \{z : (\widehat{B})_z \cap A \neq \emptyset\}$

- CHIUSURA: Operatore che fonde sia piccole spaccature, che strette e lunghe insenature. Inoltre elimina piccoli buchi e li riempie lungo il contorno.

$$A \bullet B = (A \circ B) \circ B$$

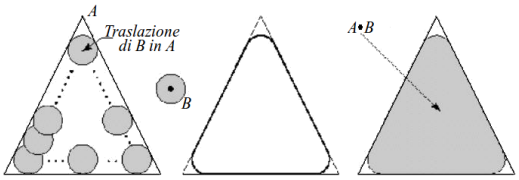


Figura 4.11: *Esempio chiusura*

Nelle equazioni precedenti che descrivono le feature map manca da definire b. L'elemento b corrisponde ad un disco di raggio uguale a J_{max} dove J_{max} è la profondità di analisi massima, che viene determinata come:

$$J_{max} = \left\lfloor \frac{1}{2} log_2 N \right\rfloor \qquad con \;\; N = min(R, C) \tag{4.25}$$

dove in $y = \lfloor x \rfloor$, y è l'intero di valore maggiore nel quale $x \geq y$, e R, C sono rispettivamente i numeri di righe e colonne dell'immagine.
È possibile verificare (Fig. 4.12) che le aree che si distinguono dal contorno (alla scala o livello 3) sono proporzionalmente molto minori rispetto a quelle ottenute ad un livello superiore come ad esempio la scala 1.

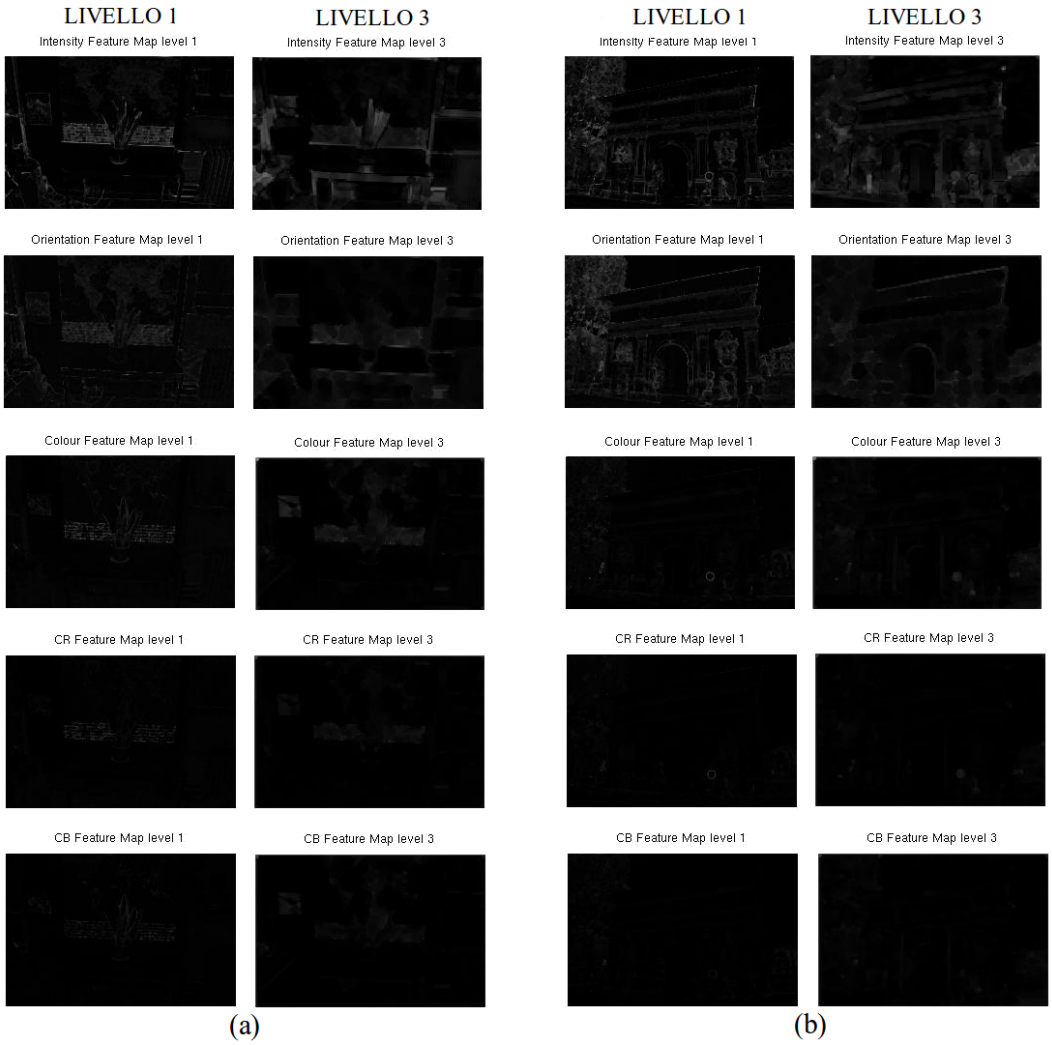


Figura 4.12: *(a) Pianta in laboratorio alla scala 1 e 3, (b) Porta Ognissanti alla scala 1 e 3*

Per tale motivo la combinazione di feature map a scale diverse risulta necessaria per riuscire a determinare sia le piccole parti salienti della scena sia quelle più grandi. La combinazione di scale differenti è ottenuta mediante l'interpolazione delle feature map a risoluzione maggiore, in modo tale da poter eseguire tra queste una sottrazione punto-punto, al risultato poi si applica la funzione di saturazione.
Le seguenti due equazioni (Eq. 4.26 - 4.27) descrivono il processo matematico che combina il risultato di due scale successive per la cospicuity map dell'orientazione.
È ovvio che lo stesso processo viene applicato anche alla cospicuity map dell'intensità e del colore:

$$\widehat{C}_O^{-j}(m,n) = \left(\left(C_O^{-j+1}(m,n)\right) \uparrow^{2m} * h_\phi(m)\right) \uparrow^{2n} * h_\phi(n) \tag{4.26}$$

$$C_O^{-j}(m,n) = \frac{2}{1 + e^{(-(\widehat{C}_O^{-j}(m,n) + O^{-j}(m,n))}} - 1 \tag{4.27}$$

dove $O^{-j}(m,n)$ e $C_O^{-j}(m,n)$ sono rispettivamente a feature map (Eq. 4.21) e cospicuity map dell'orientazione a livello j-esimo, $\widehat{C}_O^{-j}(m,n)$ è l'interpolazione di $C_O^{-j+1}(m,n)$ alla $j-esima$ più "fine", infine $\uparrow^{2m}$ e $\uparrow^{2n}$ definiscono l'interpolazione rispettivamente attraverso le righe e le colonne.

Estrazione della Saliency Map

Dopo aver creato le cospicuity map per le componenti di intensità, orientazione e colore non resta che determinare la mappa di salienza desiderata. Per fare ciò si combinano le mappe delle varie componenti, utilizzando una particolare funzione che ne preserva l'indipendenza.
Questo processo è descritto matematicamente dalla seguente equazione:

$$S(m,n) = \frac{2}{1 + e^{(-(C_I^{-0}(m,n) + C_O^{-0}(m,n) + C_C^{-0}(m,n)))}} - 1 \tag{4.28}$$

dove: $S(m,n)$ sono i valori assunti dalla mappa di salienza, $C_I^{-0}(m,n)$, $C_O^{-0}(m,n)$ e $C_C^{-0}(m,n)$ sono le cospicuity map rispettivamente dell'intensità, orientazione e colore.
Nell'immagine successiva (Fig. 4.13) sono riportati due esempi di saliency map ottenute utilizzando il metodo Wavelet.
A differenza del metodo di Itti & Koch visto nel paragrafo precedente, questo metodo restituisce grande informazioni di salienza per quanto riguarda la componente orientazione rispetto alle componenti colore ed intensità. Per tale motivo, come si può anche notare dai risultati, vi è una buona identificazione dei bordi dei vari oggetti che compongono la scena a discapito dell'individuazione dell'oggetto principale.
Questa caratteristica è la motivazione principale per la quale nell'integrazione delle saliency map in Photosynth che vedremo nel capitolo successivo, si è preferito utilizzare il metodo di Itti & Koch invece della tecnica con wavelet.

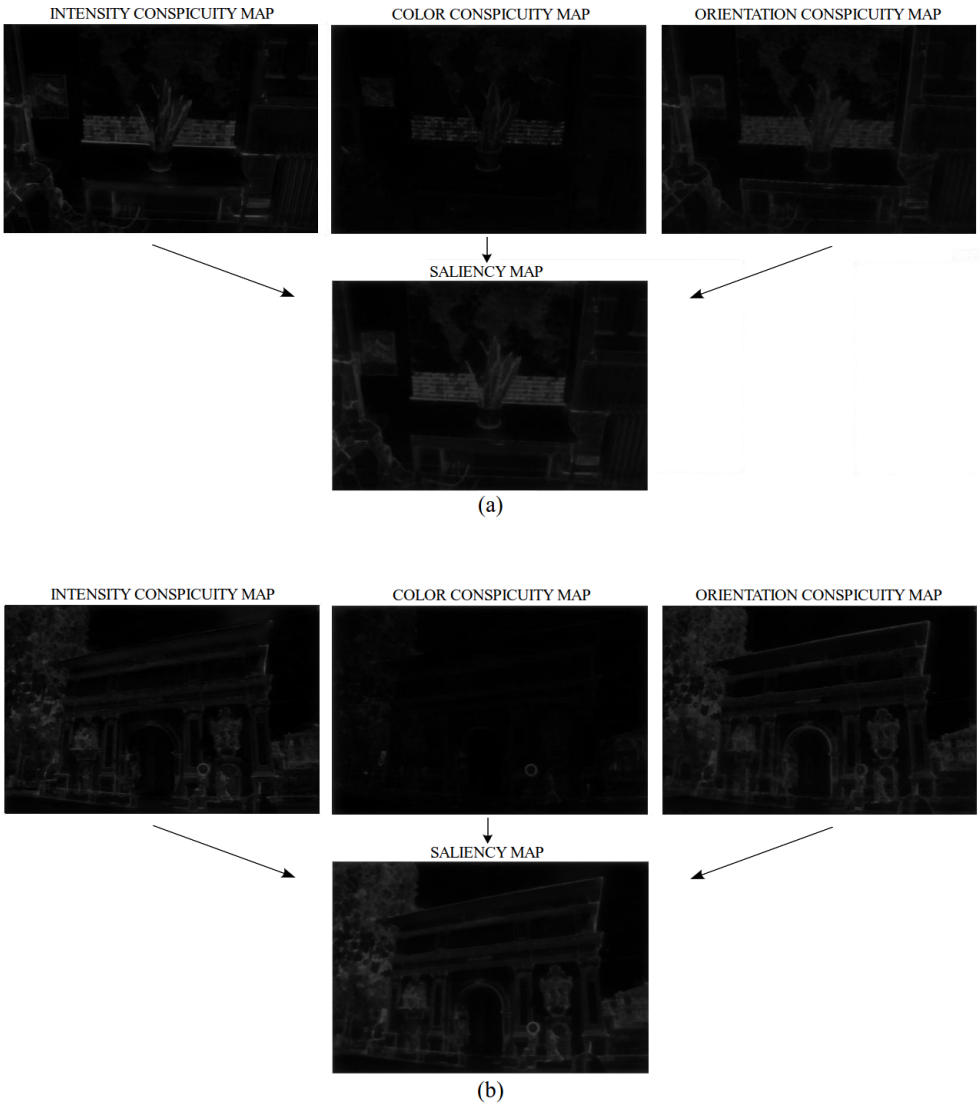


Figura 4.13: *Esempio di estrazione di saliency map in un ambiente aperto (a) ed in un ambiente chiuso (b)*

Capitolo 5

Metodi e risultati sperimentali

In questo capitolo presenteremo i metodi sviluppati per la realizzazione del modello tridimensionale a partire da un numero ridotto di immagini. Tali soluzioni sfruttano le mappe di salienza in modo tale da poter selezionare in maniera ottimale le immagini da fornire in input al Bundler.
Così facendo, si riduce il costo computazionale (visto che si utilizza un numero inferiore di foto) e allo stesso tempo si garantisce una buona qualità del modello realizzato.
Prima di analizzare le tecniche utilizzate, risulta necessario sia descrivere la routine che ci ha permesso di organizzare in maniera efficiente i dati in uscita dal Bundler, sia esaminare l'algoritmo sviluppato per effettuare il confronto di modelli diversi della stessa scena.

5.1 Funzione Matlab "get_camera_proximity.m"

Nel Paragrafo 3.4.2 abbiamo analizzato il formato del file *bundle.out* in uscita al Bundler. Questo file, oltre a contenere tutte le informazioni necessarie per la ricostruzione tridimensionale della scena, fornisce una descrizione completa della geometria delle telecamere.
Come prima cosa, quindi, ci siamo preoccupati di sviluppare una funzione Matlab, denominata *get_camera_proximity*, al fine di organizzare in maniera ordinata i dati forniti da Bundler.

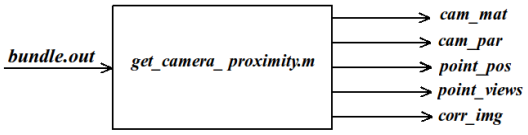


Figura 5.1: *funzione get_camera_proximity.m*

In ingresso alla funzione *get_camera_proximity* (Fig. 5.1), forniamo il file bundle.out, ottenuto utilizzando un numero di telecamere N_{CAM}. In uscita invece è possibile organizzare i dati di questo file nel seguente modo:

- ***CAM_MAT***: Matrice di dimensioni $N_{CAM} \times N_{CAM}$, contiene i punti in comune tra coppie di telecamere.

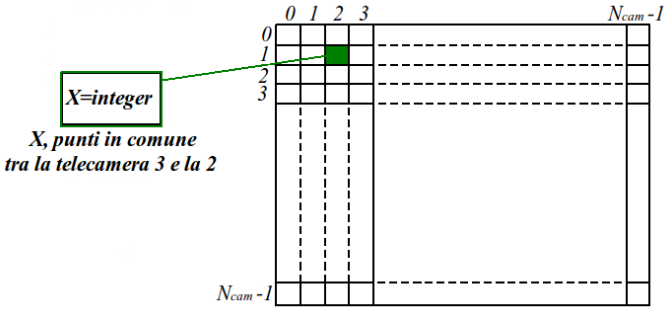


Figura 5.2: *Esempio di matrice CAM_MAT*

- ***CAM_PAR***: Matrice di dimensioni $5 \times 3 \times N_{CAM}$, contiene i parametri intrinseci ed estrinseci associati alle telecamere. Per ogni telecamera c, la prima riga della matrice, alle coordinate $(:,:,c)$, riporta i valori della focale f e della distorsione radiale k_1 e k_2. La seconda, terza e quarta riga invece contengono le componenti della matrice di rotazione $\mathbf{R}$ della telecamera, mentre l'ultima riga il vettore traslazione $\mathbf{t}$.

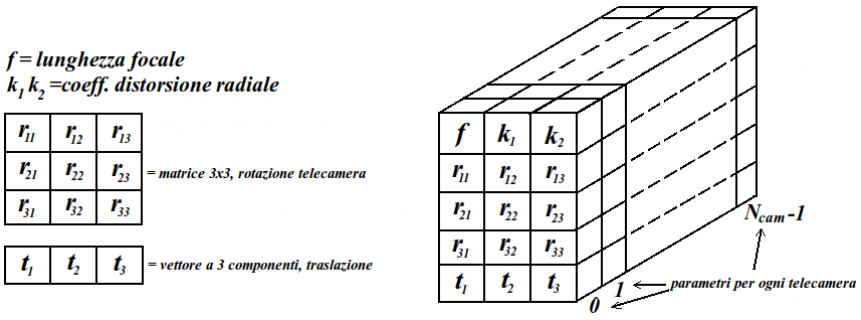


Figura 5.3: *Esempio di matrice CAM_PAR*

- ***POINT_POS***: Matrice di dimensioni $N_{points} \times 7$, dove N_{points} è il numero di vertici 3D stimati nella simulazione. Nelle prime tre colonne si riportano le coordinate (x,y,z) del punto tridimensionale. La quarta colonna memorizza il numero di telecamere nelle quali questo punto appare. Infine le colonne 5,6 e 7 ne identificano l'informazione di colore (RGB).

- ***POINT_VIEWS***: Matrice di dimensioni $3 \times N_{CAM} \times N_{points}$, riporta le informazione delle proiezioni 2D dei vertici nelle differenti telecamere. Per ogni colonna di coordinate (:,c,p), la matrice memorizza l'indice della feature associata al punto p, ed inoltre le coordinate (x,y) del pixel corrispondente al vertice nella telecamera c.

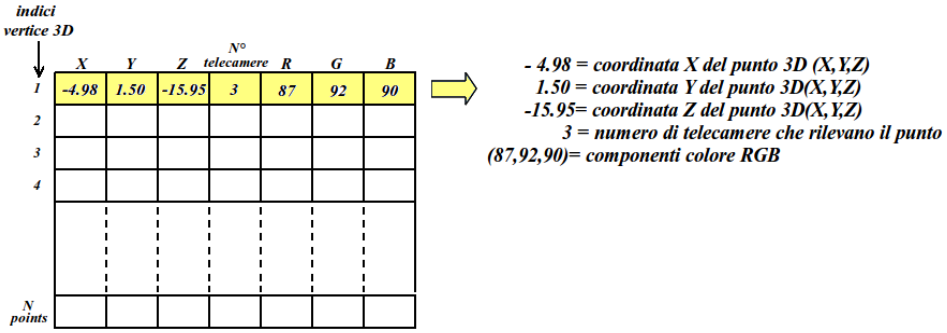


Figura 5.4: *Esempio di matrice POINT_POS*

- ***CORR_IMG***: Matrice di dimensioni $N_{points} \times (3 \cdot N_{CAM})$ contenente tutte le proiezioni del vertice tridimensionale nelle varie immagini. Per ogni vertice 3d vengono memorizzate nella tabella le telecamere in cui compare e le coordinate pixel (x,y) corrispondenti.

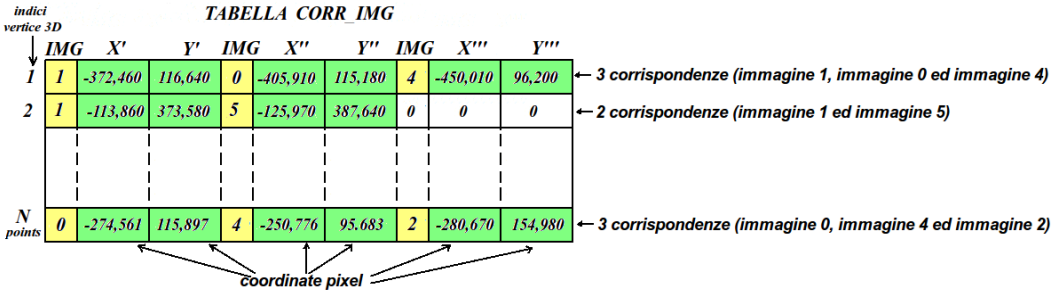


Figura 5.5: *Esempio di matrice CORR_IMG*

Dopo aver analizzato nel dettaglio i dati in uscita alla funzione Matlab, vediamo ora in Fig. 5.6, un esempio della relazione che intercorre tra la matrice *POINT_POS* e la matrice *CORR_IMG*.
Come si nota nella figura, il punto tridimensionale con indice 2, di coordinate (-1.61,4.28,-14.61), è stato identificato grazie alla corrispondenza di solo due immagini del dataset in ingresso. Tali immagini con le rispettive coordinate pixel, possono essere identificate nella matrice *CORR_IMG* grazie alla relazione esistente tra gli indici delle due tabelle. Ecco quindi che il vertice tridimensionale di indice 2 compare nell'immagine 5 alla posizione (-113,373) e nell'immagine 6 in (-125,387).

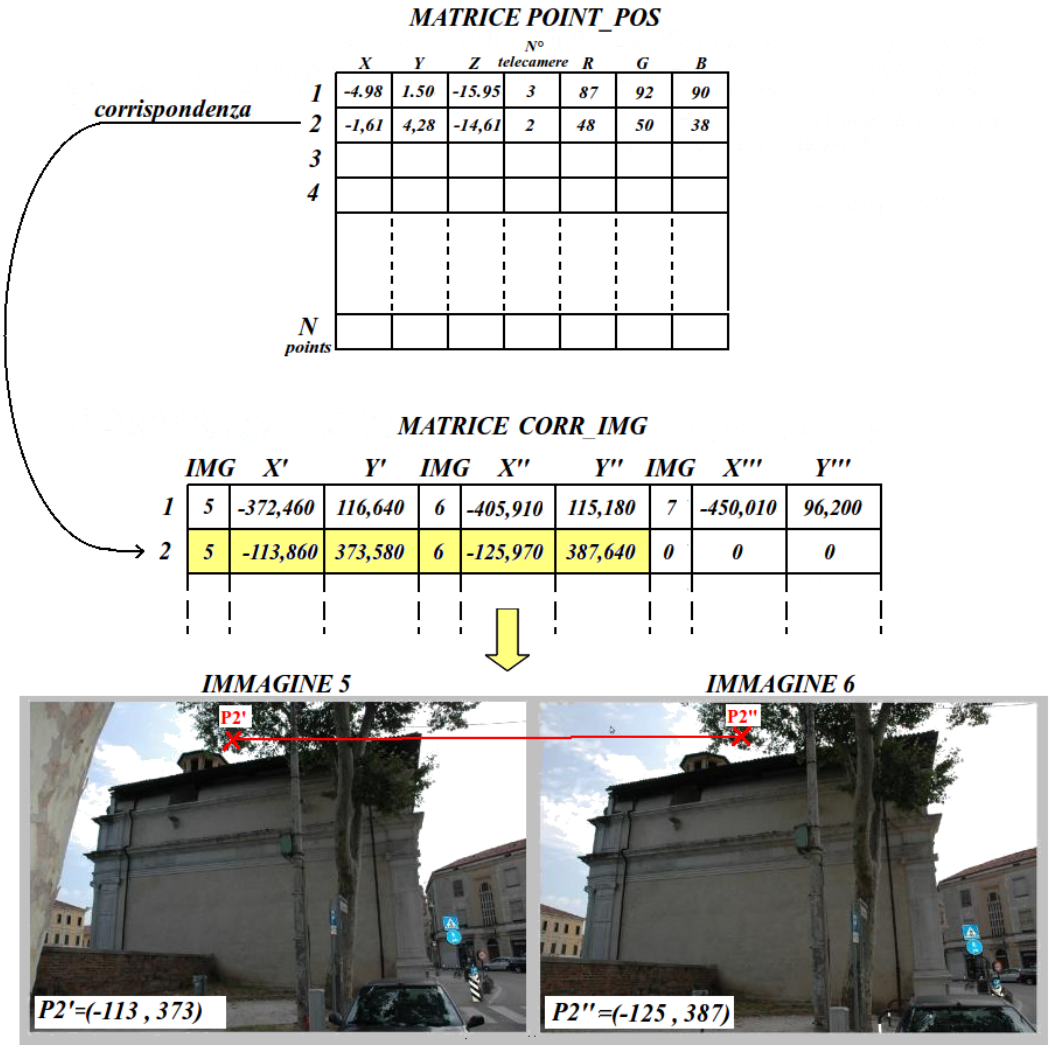


Figura 5.6: *Relazione tra la matrice POINT_POS e la matrice CORR_IMG*

5.2 Corrispondenze tra file bundle.out

Nel paragrafo precedente abbiamo descritto la funzione Matlab *get_camera_proximity.m*, la quale permette di organizzare i dati del file finale di Bundler.
Questa funzione si rende utile poiché facilita la ricerca delle corrispondenze tra vertici 3D trovati in file bundle.out che utilizzano l'intero dataset d'immagini, e simulazioni parziali che invece ne utilizzano un numero inferiore.
Il motivo di tale confronto, e soprattutto dell'utilizzo di dataset parziali, è, come già stato descritto nel Capitolo 4, la riduzione del costo computazionale pur mantenendo una discreta qualità del modello.
Per ricercare quindi le corrispondenze dei vertici 3D, tra simulazioni parziali e totali, abbiamo sviluppato un algoritmo Matlab, organizzato nel seguente modo:

A- Organizzazione dei file bundle.out della simulazione parziale e totale

B- Relazione tra le immagini delle due simulazioni

C- Matching

D- Analisi delle corrispondenze

A-Organizzazione dei file bundle.out della simulazione parziale e totale

Il calcolo di corrispondenze, tra due file bundle.out che utilizzano un numero diverso di immagini, provenienti dallo stesso dataset, non è un semplice matching tra i vertici 3D stimati.
Infatti, l'aumento del numero di telecamere produce una maggior accuratezza del modello, con una conseguente variazione nella stima dei parametri intrinseci ed estrinseci delle telecamere.
Ecco quindi che avendo parametri differenti, la stima dei vertici 3D (Paragrafo 3.4.3) per le due simulazioni sarà diversa.
Per determinare le corrispondenze tra questi vertici, ci concentreremo sulla matrice *CORR_IMG*, la quale contiene gli indici delle immagini e le relative coordinate delle feature. A differenza dei vertici 3D, il calcolo e la corrispondenza tra feature nelle due simulazioni, sarà uguale per medesime immagini, quindi trovando nelle due simulazioni le stesse corrispondenze tra feature, troveremo anche il legame tra i vertici 3D dei due modelli.
Per comprendere meglio la descrizione delle operazioni effettuate dall'algoritmo Matlab in questo primo step, proporremo un semplice esempio pratico, nel quale la simulazione totale è composta da 16 immagini mentre quella parziale solamente da 6 di queste (Fig. 5.7).

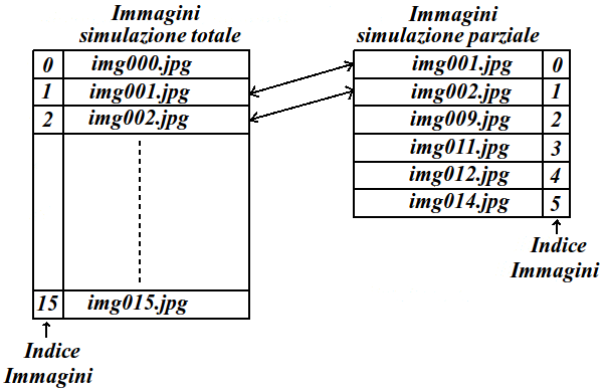


Figura 5.7: *Esempio di simulazione totale e parziale*

A1- Operazioni per la simulazione parziale

Nella matrice *CORR_IMG* ottenuta dallo script *get_camera_proximity.m*, gli indici delle immagini corrispondenti ad un vertice 3D non sono ordinati. Per facilitare il

calcolo delle corrispondenze tra simulazione parziale e totale si è deciso di ordinarli. In fig. 5.8 si può osservare un semplice esempio di tale ordinamento.

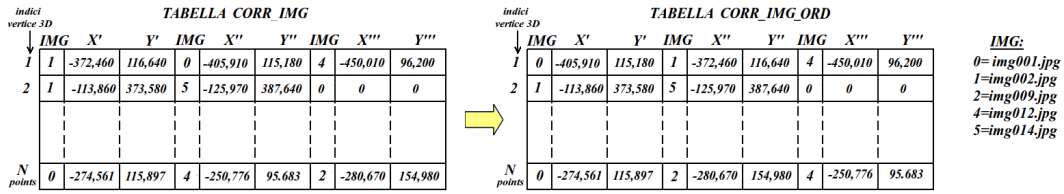


Figura 5.8: *Ordinamento degli indici nella matrice $CORR\_IMG$*

A2- Operazioni per la simulazione totale

Per poter trovare le corrispondenze tra queste due simulazioni, risulta necessario mantenere nella simulazione totale solo i vertici 3D che potrebbero essere stati stimati anche dalla simulazione parziale.
Per fare ciò, sapendo che per stimare un vertice 3D sono necessarie almeno 2 viste, si eliminano quei vertici e le rispettive corrispondenze nella matrice $CORR\_IMG$ che non utilizzano almeno 2 delle immagini della simulazione parziale. Nell'esempio riportato in Fig. 5.9, si elimina la corrispondenza di indice 2 poiché per la determinazione del corrispondente vertice 3D solo una immagine (*img002.jpg*) è presente nella simulazione parziale.

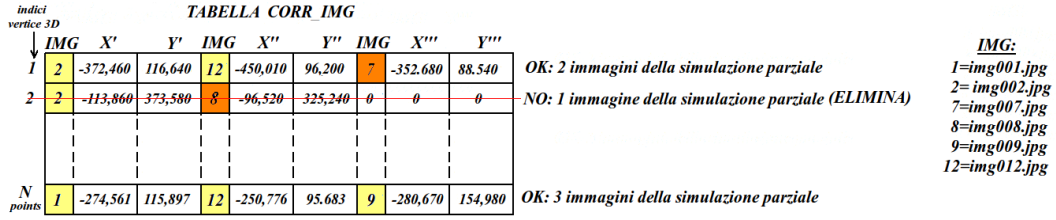


Figura 5.9: *Eliminazione dei vertici nella matrice $CORR\_IMG$ che non utilizzano almeno 2 immagini della simulazione parziale*

Dei vertici 3D rimanenti, si eliminano poi nella matrice $CORR\_IMG$ gli indici immagine, con le relative coordinate pixel, delle immagini presenti nella simulazione totale che non sono presenti in quella parziale. Nell'esempio di Fig. 5.10 si elimina nella corrispondenza di indice 1, l'indice immagine 7 e le relative coordinate pixel poichè *img007.jpg* non è presente nella simulazione parziale.

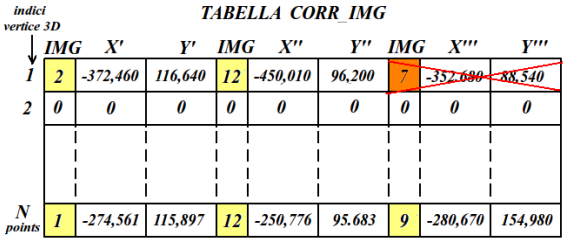


Figura 5.10: *Eliminazioni nella matrice $CORR\_IMG$ delle corrispondenze non plausibili*

Infine si ordinano gli indici delle immagini della matrice $CORR\_IMG$ (Fig. 5.11), come descritto nella simulazione parziale.

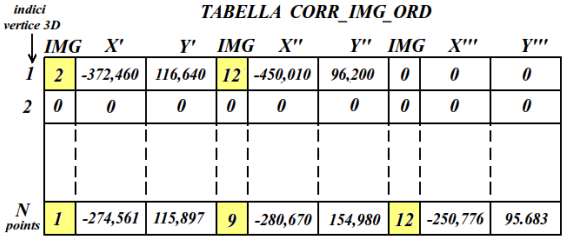


Figura 5.11: *Ordinamento degli indici nella matrice $CORR\_IMG$*

B-Relazione tra le immagini delle due simulazioni

Bundler indicizza le immagini in ingresso partendo da zero. Quindi, se si utilizza un dataset con N immagini, queste saranno indicizzate da $0, \ldots, N-1$.
Se si vogliono determinare le corrispondenze tra simulazioni che utilizzano lo stesso dataset d'immagini ma in numero differente, risulta necessario modificare l'indice delle immagini altrimenti corrisponderà per le due simulazioni ad immagini differenti.
È possibile notare nell'esempio precedente (Fig. 5.7) come effettivamente l'indice tra le due simulazioni sia diverso. Ad esempio, all'immagine *img001.jpg* è associato l'indice 1 nella simulazione totale, 0 invece in quella parziale.
Nell'esempio quindi sarà necessario modificare l'indice immagine per la matrice $CORR\_IMG$ nel seguente modo:

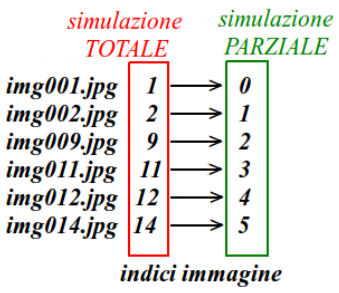


Figura 5.12: *Variazione degli indici immagine*

C-Matching

Una volta adeguati i dati, siamo pronti a determinare le corrispondenze. Come già detto, le triangolazioni dei vertici 3D, corrispondenti a feature su più immagini, risulteranno diverse nella simulazione parziale e totale. Per questo motivo, la ricerca delle corrispondenze si baserà sulla matrice $CORR\_IMG$, che contiene, gli indici delle immagini e le relative coordinate delle feature.
L'algoritmo sviluppato effettua due fasi di matching:

C1- Prima fase di matching

La prima fase di matching, è una semplice ricerca delle medesime righe della matrice $CORR\_IMG$. Andremo ad effettuare il matching tra gli indici delle matrici $CORR\_IMG$ che contengono gli stessi indici immagine ma anche le medesime coordinate della feature individuata. Una volta determinata una corrispondenza sarà quindi possibile determinare un legame tra due vertici 3D delle due simulazioni che corrispondono alle stesse feature, ma che come già detto assumono valori differenti.
Nell'esempio pratico descritto, è possibile notare Fig. 5.13, come il matching venga effettuato tra gli indici 1 delle due tabelle, ma non per l'indice 3 della matrice della simulazione totale che pur contenendo gli stessi indici immagine, ha per tali feature differenti coordinate.

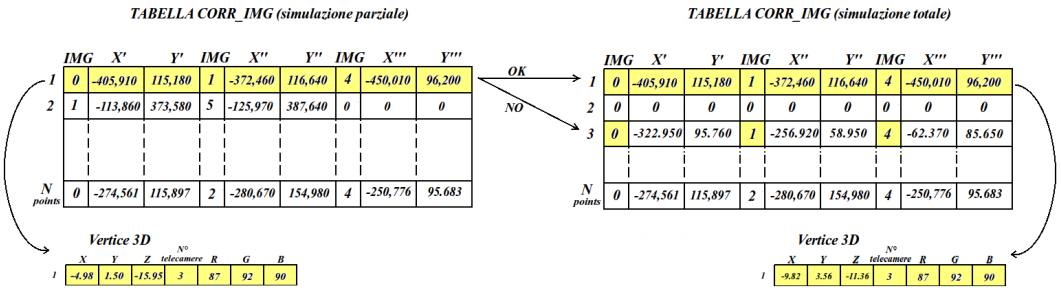


Figura 5.13: *Prima fase di matching*

C2- Seconda fase di matching

Eseguita la prima fase di matching, molte corrispondenze non vengono individuate poiché la simulazione totale, utilizzando un numero maggiore di telecamere, riesce a determinare più corrispondenze anche tra feature appartenenti al dataset della simulazione parziale.
Dato quindi un vertice 3D della simulazione parziale non ancora associato, si cerca nella tabella $CORR\_IMG$ della simulazione totale, l'indice che contiene le medesime immagini e coordinate pixel.
Una volta trovati tutti i possibili matching (è possibile trovarne più di uno), si sceglie come corrispondenza quello stimato grazie al maggior numero di telecamere. Questo criterio garantisce una maggiore precisione nella localizzazione 3D del punto.
Nell'esempio pratico, come riportato in Fig. 5.13, il vertice di indice 2 non ha trovato una corrispondenza nella prima fase di matching come accaduto per quello di indice 1.

Si analizzano quindi le righe della matrice $CORR\_IMG$ e si nota come siano possibili due corrispondenze visto che entrambi contengono lo stesso indice immagine (1 e 5) ed anche le medesime coordinate pixel.
Si nota inoltre come la simulazione totale abbia trovato altre corrispondenze per feature di immagini che appartengono all'insieme di quelle parziali (in questo caso l'immagine 3 e 4 per la corrispondenza di indice 37 e l'immagine 2 per quella di indice 52).
A questo punto, la scelta ricade sulla corrispondenza di indice 37, la quale utilizza ben 4 viste per la stima del vertice 3D a differenza di quella di indice 52 che ne utilizza solamente 3.

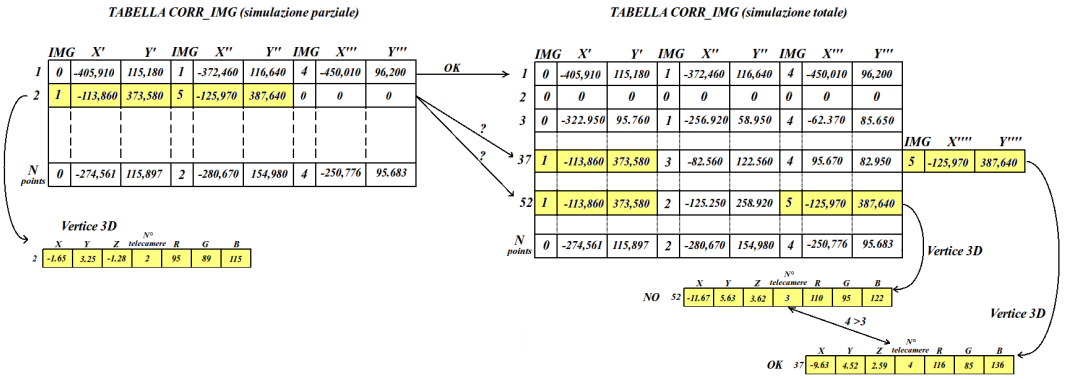


Figura 5.14: *Seconda fase di matching*

D- Analisi delle corrispondenze

Una volta effettuate le due fasi di matching otterremo una tabella contenente gli indici corrispondenti ai vertici 3D equivalenti nelle due simulazioni.
Nell'esempio proposto e da quanto visto dalle operazioni precedenti, si ottiene una matrice corrispondenze dove all'indice 1 del vertice 3D della simulazione parziale corrisponde il vertice di indice 1 di quella totale (prima fase di matching), allo stesso modo il vertice di indice 2 corrisponde a quello di indice 37 (seconda fase di matching).

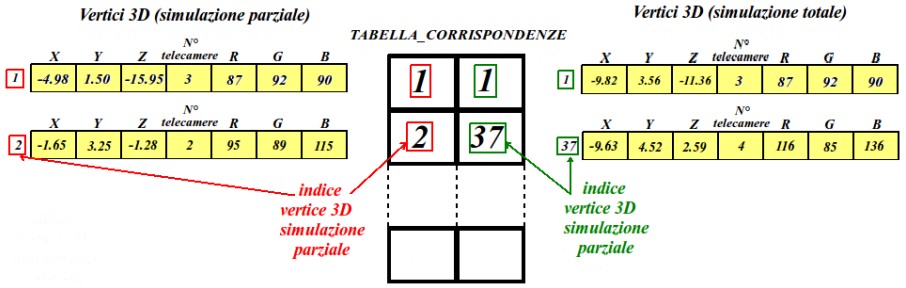


Figura 5.15: *Analisi delle corrispondenze*

Da questa matrice sarà possibile determinare:

- **Numero di vertici 3D corrispondenti, tra la simulazione parziale e totale.**

- **Numero di vertici 3D corrispondenti, tra la simulazione parziale e totale all'interno di un range di interesse.**
  In questo caso, non si considerano tutti i vertici 3D che hanno trovato corrispondenza ma solo quelli che sono stati stimati da feature che appartengono all'oggetto principale della scena. In Fig. 5.16 vengono riportati due possibili esempi di range, che come già detto comprendono gli oggetti protagonisti della scena, ovvero la pianta di laboratorio e la Porta Ognissanti.

Figura 5.16: *Possibili range per la simulazione "Pianta"e la simulazione "Porta Ognissanti"*

- **Errore quadratico medio (MSE), tra le due simulazioni.**
  Siano $V_k^T = (x_k, y_k, z_k)$ e $V_k^P = (x'_k, y'_k, z'_k)$ generici vertici 3D rispettivamente della simulazione totale e parziale che hanno trovato corrispondenza, allora, si calcola l'errore quadratico medio come:

$$MSE = \frac{1}{N_{corr}} \sum_{k=1}^{N_{corr}} \|V_k^T - V_k^P\|^2 \tag{5.1}$$

- **Errore quadratico medio (MSE), tra le due simulazioni utilizzando l'algoritmo ICP.**
  Come abbiamo già detto, utilizzando un numero di telecamere diverso i parametri intrinseci ed estrinseci variano. Ciò si riflette sulla triangolazione dei vertici 3D delle due simulazioni che quindi assumono valori differenti.
  Per trovare una qualche corrispondenza tra la geometria descritta dall'insieme dei vertici 3D della simulazione parziale V_k^P con $k \in \{1, \ldots, N_{corr}\}$ e la geometria definita invece dai corrispondenti vertici della simulazione totale V_k^T con $k \in \{1, \ldots, N_{corr}\}$, si è utilizzato l'algoritmo ICP.
  Quest'ultimo calcola la distanza tra le due geometrie utilizzando il metodo dei minimi quadrati e itera finché non determina la trasformazione che minimizza questa distanza.
  In ingresso richiede gli insiemi V_k^P e V_k^T con $k \in \{1, \ldots, N_{corr}\}$ e restituisce una matrice $\mathbf{R}$ che descrive la rotazione della trasformazione e un vettore $\mathbf{t}$ che ne

descrive la traslazione.
La nuova geometria che minimizza la distanza sarà quindi data da:

$$V_k^{ICP} = \mathbf{R} \cdot V_k^T + \mathbf{t}$$

Dalla geometria precedente si determina poi l'errore quadratico medio:

$$MSE_{ICP} = \frac{1}{N_{corr}} \sum_{k=1}^{N_{corr}} \|V_k^{ICP} - V_k^P\|^2 \tag{5.2}$$

5.3 Tecniche per l'ottimizzazione dei modelli 3D

Dopo aver descritto la funzione Matlab che ci permette di confrontare la geometria tridimensionale fornita da Bundler, presentiamo ora le due principali simulazioni effettuate per la generazione del modello tridimensionale da ottimizzare. Successivamente analizzeremo le tecniche sfruttate per raggiungere tale scopo.

5.3.1 Modelli tridimensionali

Il primo modello tridimensionale che abbiamo analizzato è stato realizzato in laboratorio con il fine di rappresentare una pianta in laboratorio. Compatibilmente con la complessità della geometria della scena, si è deciso di utilizzare un dataset di 46 immagini che ben descrive l'oggetto protagonista in ogni sua angolazione.
Come si può vedere in Fig. 5.17 le foto sono state scattate in un ambiente non controllato (a differenza ad esempio di una camera di acquisizione), quindi nelle immagini non compare solo l'oggetto della scansione ma anche altre componenti esterne: fisse (cartina topografica, tavolo, ecc) e variabili (persone).

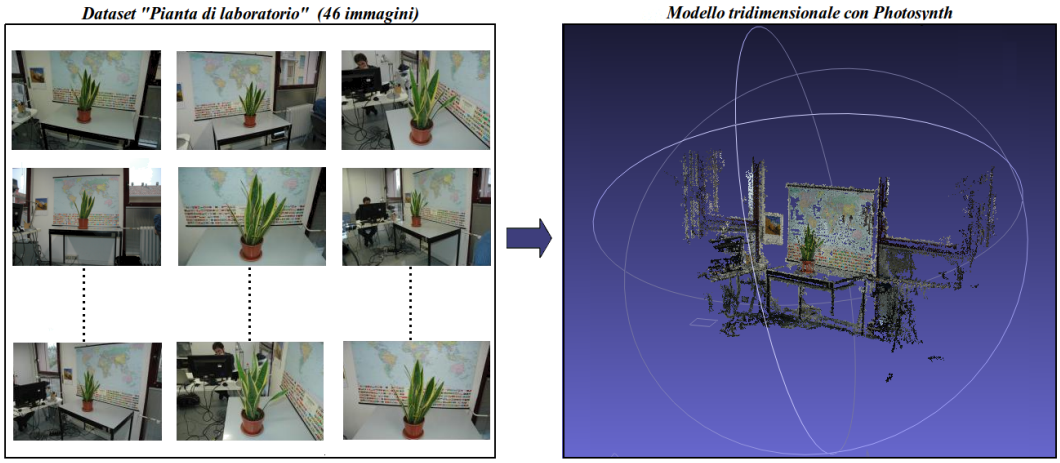


Figura 5.17: *Dataset e modellazione 3D per la simulazione "Pianta"*

Il secondo modello tridimensionale invece è stato realizzato coerentemente con la destinazione d'uso principale del programma Photosynth, ovvero la creazione di percorsi turistici virtuali. Ecco quindi che la nostra scelta è ricaduta sulla Porta Ognissanti, uno dei numerosi edifici storici di Padova.
Vista la complessità della scena si è scelto di utilizzare in questo caso un dataset più corposo, ben 186 immagini che rappresentano la Porta Ognissanti in ogni sua angolazione e dettaglio.
Le immagini, a differenza del primo modello, sono state scattate all'aperto ma anche in questo caso come si nota in Fig. 5.18, oltre alla rappresentazione dell'oggetto protagonista della scena compaiono altre componenti esterne: fisse (cartelli, altri edifici, ecc...) e variabili (persone, macchine, alberi, ecc...).

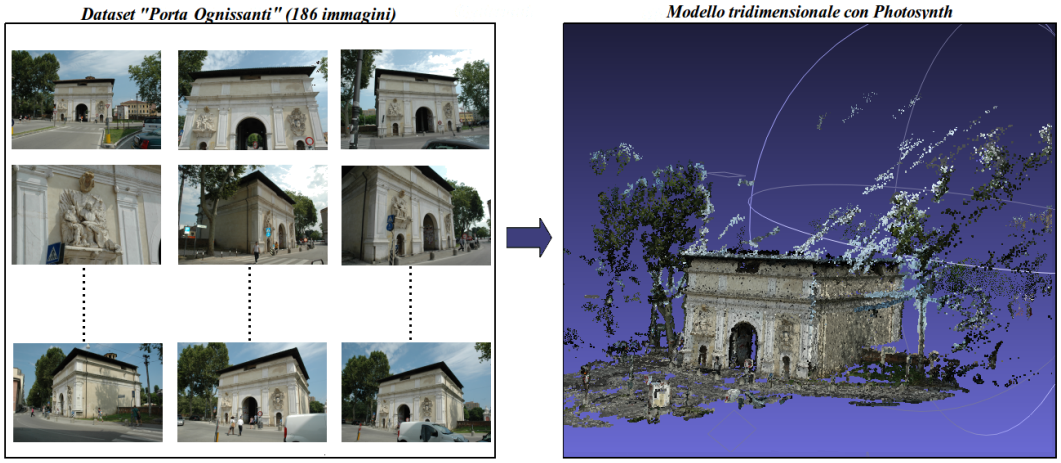


Figura 5.18: *Dataset e modellazione 3D per la simulazione "Porta Ognissanti"*

5.3.2 Tecniche e risultati degli ordinamenti delle immagini tramite mappe di salienza

Presentiamo ora le tecniche che sfruttano le informazioni fornite dalla saliency map per ottimizzare il modello 3D. In particolare vedremo:

A- Ordinamento delle immagini in maniera random

B- Ordinamento delle immagini in base al valore di salienza

C- Ordinamento clustering delle immagini in base al valore di salienza

D- Ordinamento delle immagini con range in base al valore di salienza

E- Ordinamento clustering delle immagini con range in base al valore di salienza

F- Ordinamento delle immagini utilizzando saliency 2.5D

A-Ordinamento delle immagini in maniera random

Per poter effettuare un confronto che descriva la bontà dei successivi ordinamenti, sono state effettuate delle simulazioni parziali con ordinamenti delle immagini di tipo casuale. Per la simulazione "Pianta", sono state effettuate 8 simulazioni parziali, con un numero crescente di immagini, in particolare abbiamo utilizzato {14, 16, 18, 20, 22, 24, 26, 36} $\in$ 46 iniziali. Per la seconda simulazione, "Porta Ognissanti", avendo un numero maggiore di immagini, abbiamo deciso di distribuirle in maniera differente, ovvero {24, 44, 64, 84, 104, 124, 144} $\in$ 186 iniziali.
Come avevamo descritto nel Paragrafo 5.2, dalla corrispondenza dei vertici 3D tra simulazione parziale e totale, riporteremo i valori stimati di:

- MSE= Errore quadratico medio (Mean Square Error)
- MSE_{ICP}= Errore quadratico medio ottenuto utilizzando l'algoritmo ICP
- V_{CORR}= Numero dei vertici 3D della simulazione parziale che hanno trovato corrispondenza in quella totale
- V_{TOT}^{P}= Numero di tutti i vertici 3D stimati dalla simulazione parziale

Prima di riportare i risultati ottenuti dalle simulazioni, risulta opportuno ricordare che i valori determinati sono stati acquisiti dalla media di 3 simulazioni in maniera tale da descrivere in modo attendibile la scelta random delle immagini.

A1-Risultati ordinamento random per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	0.913	0.373	5353/6082
16 immagini	40.759	0.463	6717/7706
18 immagini	1.220	0.694	7070/8670
20 immagini	34.298	0.411	8553/10309
22 immagini	66.064	0.307	8906/10738
24 immagini	26.830	0.646	9294/11175
26 immagini	1.470	0.719	10173/12146
36 immagini	3.329	1.198	14219/16394

Tabella 5.1

Come già detto nel Paragrafo 5.2 è possibile definire dei range di interesse per le immagini del nostro dataset. In questo modo si possono considerare solo le feature e quindi i

vertici 3D, che ricadono nell'oggetto protagonista della scena (nelle nostre due simulazioni la pianta e la Porta Ognissanti). Definendo i range per la simulazione "Pianta"si ottengono i seguenti risultati:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V^P_{TOT}
14 immagini	0.706	0.267	256/336
16 immagini	39.747	0.235	358/471
18 immagini	0.590	0.247	455/610
20 immagini	32.917	0.198	573/748
22 immagini	64.176	0.115	600/785
24 immagini	24.881	0.288	588/801
26 immagini	1.052	0.362	723/976
36 immagini	2.590	0.900	1026/1389

Tabella 5.2

A2-Risultati ordinamento random per la simulazione "Porta Ognissanti"
Al pari di quanto fatto per la simulazione "Pianta", riportiamo i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti".

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V^P_{TOT}
24 immagini	1134.320	636.503	269/521
44 immagini	686.953	411.285	10422/15335
64 immagini	928.678	442.424	16955/24439
84 immagini	569.264	447.905	24291/32993
104 immagini	447.446	316.268	30596/40532
124 immagini	549.560	407.448	35992/47039
144 immagini	351.587	251.598	41981/52587
164 immagini	640.374	552.913	47347/59096

Tabella 5.3

Se consideriamo anche in questo caso un range di interesse che identifichi la Porta Ognissanti otteniamo i seguenti risultati:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	820.446	234.009	183/307
44 immagini	432.582	131.635	9103/12858
64 immagini	583.277	119.748	14177/19561
84 immagini	277.363	104.216	19471/25785
104 immagini	214.597	71.525	23661/30603
124 immagini	265.407	97.856	27471/34881
144 immagini	178.938	229.581	31243/38298
164 immagini	275.851	106.297	34712/42134

Tabella 5.4

B-Ordinamento delle immagini in base al valore di salienza

In questo tipo di ordinamento la prima operazione effettuata è stata quella di determinare le saliency map per tutte le immagini del nostro dataset, mediante l'algoritmo di Itti & Koch (Paragrafo 4.3). Successivamente abbiamo analizzato i valori assunti da ogni singola mappa, suddividendoli in settori che ne descrivono la salienza. In particolare, il pixel generico p' che può assumere valori tra 0 e 255, appartiene al settore:

- E: se il valore di salienza di $p' \in \{0 \div 49\}$
- D: se il valore di salienza di $p' \in \{50 \div 99\}$
- C: se il valore di salienza di $p' \in \{100 \div 149\}$
- B: se il valore di salienza di $p' \in \{150 \div 199\}$
- A: se il valore di salienza di $p' \in \{200 \div 255\}$

Figura 5.19: *Pixel e settori corrispondenti per la simulazione "Pianta"e "Porta Ognissanti"*

Dalle assunzioni fatte in precedenza, è facile osservare come il settore A contenga i punti di maggior salienza; risulta quindi possibile definire una bontà oggettiva per le immagini ordinandole in base al numero di pixel contenuti in tale settore.
Maggiore infatti sarà il numero di pixel $p' \in \{200 \div 255\}$, maggiore sarà la salienza

dell'immagine e l'importanza che essa avrà ai fini della ricostruzione del modello tridimensionale.
Quindi, in definitiva, le N foto selezionate per effettuare le simulazioni parziali, saranno le N immagini con maggior pixel $p' \in \{200 \div 255\}$.

B1-Risultati ordinamento in base al valore di salienza per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	183.336	1.043	7052/7921
16 immagini	131.721	3.745	8066/9065
18 immagini	112.490	0.276	8732/9792
20 immagini	95.630	0.552	9273/10458
22 immagini	119.526	4.651	9373/10634
24 immagini	0.286	0.035	9892/11194
26 immagini	0.281	0.152	9982/11141
36 immagini	0.152	0.013	12278/15837

Tabella 5.5

Per i vertici 3D considerati nel range che descrive la "Pianta"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	175.944	1.076	501/620
16 immagini	120.606	1.634	638/815
18 immagini	109.016	0.290	587/807
20 immagini	91.219	0.455	611/831
22 immagini	106.716	1.552	636/859
24 immagini	0.203	0.005	600/837
26 immagini	0.227	0.001	705/916
36 immagini	0.111	0.002	1088/1508

Tabella 5.6

È possibile notare che a parità di immagini utilizzate l'ordinamento in base al valore di salienza permette di ottenere modelli più accurati. Ad esempio, analizzando i risultati per 26 immagini in Tabella 5.5 e in Tabella 5.1, si osserva come il metodo random permetta di avere $MSE_{ICP} = 0.719$ mentre il metodo basato sui valori di salienza $MSE_{ICP} = 0.152$. Un notevole miglioramento di qualità si nota anche confrontando

la Tabella 5.3 e Tabella 5.6.

B2-Risultati ordinamento in base al valore di salienza per la simulazione "Porta Ognissanti"

Riportiamo i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti"per questo tipo di ordinamento.

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	1258.850	658.193	5475/8277
44 immagini	1495.573	837.048	10691/16034
64 immagini	1501.844	1564.652	15860/22939
84 immagini	1448.278	1118.117	23082/31719
104 immagini	1539.504	1270.778	28363/38189
124 immagini	857.522	632.581	36612/45995
144 immagini	1319.609	577.857	40779/52423
164 immagini	603.368	6580.815	48179/59248

Tabella 5.7

Per i vertici 3D considerati nel range che descrive la "Porta Ognissanti"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	913.956	196.435	4950/7053
44 immagini	1002.072	203.563	9105/13078
64 immagini	834.952	245.133	13290/18177
84 immagini	792.617	232.337	18965/24916
104 immagini	804.880	209.457	22600/29094
124 immagini	389.287	144.023	28110/34115
144 immagini	439.365	118.469	30320/37491
164 immagini	305.430	212.741	34442/41903

Tabella 5.8

Per quanto riguarda il dataset "Porta Ognissanti"le prestazioni del metodo basato sui valori di salienza sono leggermente inferiori a causa del gran numero di oggetti esterni. Tuttavia è possibile notare come tale metodo permetta di controllare con il numero di immagini la precisione del modello generato. Infatti la Tabella 5.8 mostra come il valore di MSE progressivamente decresce all'aumentare del numero di immagini, in Tabella 5.4 invece è molto variabile.

C-Ordinamento clustering delle immagini in base al valore di salienza

Il metodo visto precedentemente ha la caratteristica di ordinare le immagini del nostro dataset per valori di salienza. Il semplice ordinamento decrescente però, porta ad identificare immagini (con maggior salienza) che hanno caratteristiche molto simili, soprattutto rispetto l'angolazione con la quale vengono scattatate. Ecco quindi che, soprattutto in simulazioni parziali che utilizzano un numero molto ristretto di immagini, questo può presentare un problema. Infatti se si utilizzano immagini scattate tutte con la stessa direzione, non risulta possibile realizzare un buon modello 3D che permetta di rappresentare l'oggetto protagonista in ogni sua angolazione.

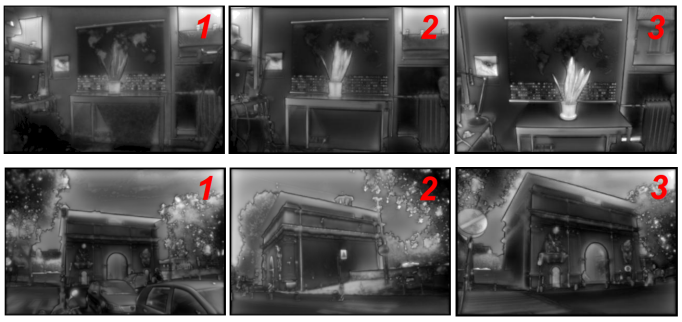


Figura 5.20: *Prime 3 immagini selezionate dall'ordinamento in base al valore di salienza*

In Fig. 5.20 si nota come quanto detto precedentemente non avvenga nella simulazione "Porta Ognissanti", poiché il calcolo delle saliency map per questo tipo di scena (che si svolge in un ambiente "aperto") è molto soggetto alla salienza degli agenti esterni come alberi e intensità dello sfondo.
Nella simulazione "Pianta"invece, le prime immagini ordinate per valore di salienza sono tutte scattate da una vista centrale. Per questo motivo, invece di utilizzare un semplice ordinamento decresente delle foto, abbiamo deciso di suddividere le immagini in base alla posizione in cui sono state scattate, vediamo ora le operazioni effettuate.

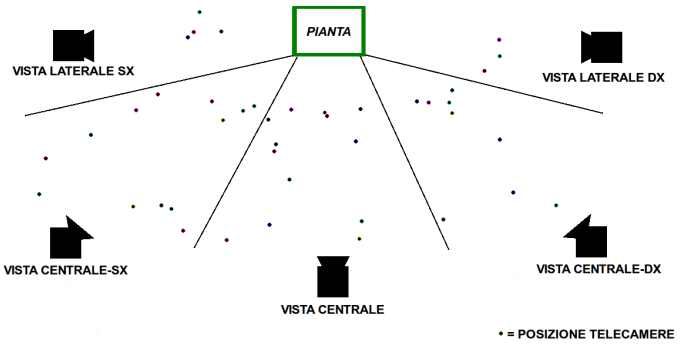


Figura 5.21: *Suddivisione clustering delle telecamere*

Il file Bundle.out fornisce la stima dei parametri estrinseci delle telecamere (matrice

R di rotazione, e vettore **t** di traslazione), da questi risulta possibile identificare le posizioni delle telecamere all'interno della scena.
Una volta determinate queste posizioni, si suddivide la scena e quindi le immagini, in cinque insiemi che rappresentano le viste di osservazione (Fig. 5.21). Per ognuno di questi si effettua successivamente l'ordinamento delle immagini in base al numero di pixel $p' \in \{200 \div 255\}$, effettuando così cinque ordinamenti indipendenti. A questo punto, le N foto da selezionare per effettuare le simulazioni parziali saranno prelevate da ogni vista per valore di salienza, proporzionalmente al numero di telecamere presenti per ciascuna. In questo modo avremo un set d'immagini prese da tutte e cinque le possibili viste, in maniera tale da poter rappresentare la scena 3D in ogni sua angolazione.

C1-Risultati ordinamento clustering delle immagini in base al valore di salienza per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	12.128	3.991	5011/5726
16 immagini	11.378	3.396	5609/6567
18 immagini	11.937	3.975	6079/7142
20 immagini	11.958	4.087	7427/8581
22 immagini	13.365	6.918	6402/9412
24 immagini	9.181	2.157	7461/10644
26 immagini	11.904	5.413	8452/11941
36 immagini	0.002	0.0018	13177/16902

Tabella 5.9

Per i vertici 3D considerati nel range che descrive la "Pianta"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	10.064	2.117	204/274
16 immagini	9.582	1.308	217/297
18 immagini	9.457	1.787	220/334
20 immagini	10.074	1.733	240/513
22 immagini	8.732	2.210	271/615
24 immagini	7.219	1.001	365/615
26 immagini	7.633	2.347	446/815
36 immagini	0.00015	0.00002	1153/1564

Tabella 5.10

C2-Risultati ordinamento clustering delle immagini in base al valore di salienza per la simulazione "Porta Ognissanti"
Riportiamo ora i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti"per questo tipo di ordinamento.

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	1124.571	809.937	4447/7064
44 immagini	1116.540	969.391	11718/16871
64 immagini	1522.875	179.881	17206/23976
84 immagini	839.367	669.002	23105/31290
104 immagini	928.998	733.581	30738/40154
124 immagini	394.972	340.704	35853/45766
144 immagini	447.216	320.412	41944/54119
164 immagini	1066.229	910.926	45766/59464

Tabella 5.11

Per i vertici 3D considerati nel range che descrive la "Porta Ognissanti"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	851.675	191.778	3866/5771
44 immagini	677.013	315.911	10362/13990
64 immagini	1421.528	54.547	14547/19355
84 immagini	316.846	140.953	18376/23702
104 immagini	348.630	159.777	23844/29987
124 immagini	173.010	92.856	27217/34005
144 immagini	280.309	59.937	30807/3872
164 immagini	412.533	217.289	33885/41874

Tabella 5.12

È possibile notare (mettendo in confronto la Tabella 5.7 con la Tabella 5.11 e Tabella 5.8 con Tabella 5.12) che l'operazione di clustering ha portato qualche miglioramento nel dataset "Porta Ognissanti"come in quello "Pianta". Tuttavia il metodo random funziona ancora in in modo migliore in termini di MSE.

D-Ordinamento delle immagini con range in base al valore di salienza

Abbiamo visto nell'ordinamento (B) per valore di salienza, in particolare nella Fig. 5.20, come la selezione delle immagini più salienti sia affetta da disturbi di agenti esterni.

La simulazione "Porta Ognissanti" rappresentando un ambiente "aperto" ne è fortemente soggetta, infatti il cielo (grazie alla componente intensità), gli alberi e i cartelli stradali (grazie alla componente colore) contribuiscono in maniera essenziale alla salienza dell'immagine.
Quanto detto vale, anche se in maniera minore, per la simulazione "Pianta", la quale nonostante rappresenti un ambiente "chiuso" è soggetta all'intensità della luce proveniente dalla finestra e dalla componente colore del quadro.
Per questi motivi, abbiamo deciso di effettuare il medesimo ordinamento per valore di salienza visto precedentemente, applicato però ai soli pixel che giacciono all'interno di un range che definisce l'oggetto protagonista della scena (Fig. 5.22).

Figura 5.22: *Scelta dei soli pixel che appartengono al range*

Come si può notare (Fig. 5.23), a differenza dell'ordinamento precedente, le immagini più salienti descrivono in maniera migliore l'oggetto di cui si vuole realizzare il modello tridimensionale.

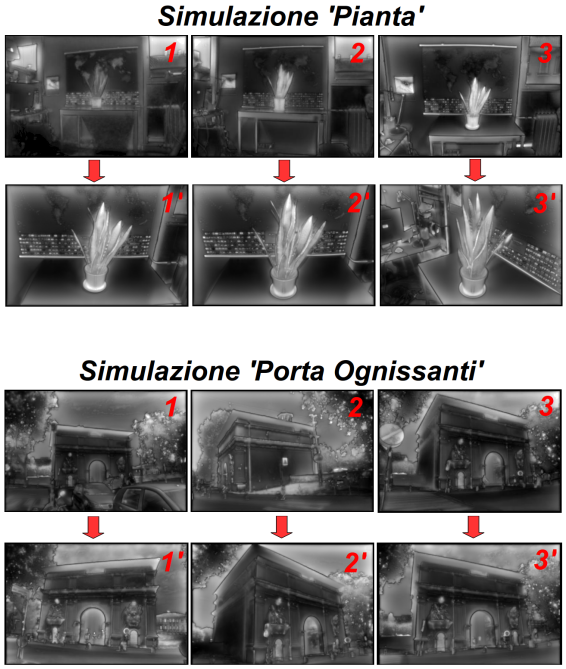


Figura 5.23: *Nuovo ordinamento delle immagini grazie all'utilizzo del range*

D1-Risultati ordinamento con range in base al valore di salienza per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	55.069	9.9067	9683/10536
16 immagini	88.239	14.54	10354/11403
18 immagini	0.385	0.168	11106/12141
20 immagini	0.322	0.127	12112/13360
22 immagini	1.339	1.115	12215/13387
24 immagini	0.755	0.233	13458/14657
26 immagini	0.748	0.265	12012/15186
36 immagini	0.844	0.138	14931/18340

Tabella 5.13

Per i vertici 3D considerati nel range che descrive la "Pianta"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	36.222	6.519	741/870
16 immagini	83.033	13.512	736/870
18 immagini	0.252	0.081	725/972
20 immagini	0.271	0.094	920/1233
22 immagini	1.201	0.508	944/1244
24 immagini	0.635	0.128	1102/1342
26 immagini	0.581	0.142	1066/1414
36 immagini	0.716	0.301	1346/1661

Tabella 5.14

È possibile notare che rispetto l'ordinamento random i valori di MSE sono un pò più alti ma il numero di punti è aumentato (confronto tra la Tabella 5.13 e la Tabella 5.3). La stessa conclusione si può trarre per i valori di MSE che corrispondono ai vertici 3D che ricadono nel range di interesse (confronto tra la Tabella 5.4 e la Tabella 5.14).

D2-Risultati ordinamento con range in base al valore di salienza per la simulazione "Porta Ognissanti"

Riportiamo i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti"per questo tipo di ordinamento.

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	1110.008	370.880	7639/10346
44 immagini	1021.726	660.305	13106/17488
64 immagini	213.548	130.451	20914/26658
84 immagini	771.807	651.882	26290/34982
104 immagini	310.852	125.345	30594/39921
124 immagini	546.851	439.770	36460/47901
144 immagini	635.561	502.797	41846/54325
164 immagini	616.170	400.580	46586/59424

Tabella 5.15

Per i vertici 3D considerati nel range che descrive la "Porta Ognissanti"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	847.059	167.068	7024/9452
44 immagini	598.641	164.208	11664/15467
64 immagini	127.554	34.098	17708/22424
84 immagini	329.203	166.347	22442/29306
104 immagini	229.841	30.859	25338/32625
124 immagini	270.557	107.796	28816/36895
144 immagini	283.333	112.401	31946/40155
164 immagini	372.883	91.020	34022/42300

Tabella 5.16

Per quanto riguarda il dataset "Porta Ognissanti"(Tabella 5.15 e Tabella 5.16) è possibile notare un miglioramento significativo sia in termini di precisione sia in termini di densità di punti. Con 104 immagini l'ordinamento con range in base al valore di salienza ottiene un $MSE_{ICP} = 30.859$ su 25338 vertici, il metodo random invece prevede un $MSE_{ICP} = 71.525$ su 23661 vertici. A tal proposito è possibile concludere che il modello ottenuto è più preciso e denso.

E-Ordinamento clustering delle immagini con range in base al valore di salienza

L'ordinamento (C) clustering delle immagini in base al valore di salienza, può essere applicato anche al metodo precedente (D), il quale considera solo i pixel appartenenti al range di interesse.

E1-Risultati ordinamento clustering delle immagini con range in base al valore di salienza per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	13.006	4.333	6879/7731
16 immagini	10.519	3.558	8277/9237
18 immagini	11.766	4.678	9321/10399
20 immagini	77.528	64.872	9556/10742
22 immagini	11.183	4.187	11350/12731
24 immagini	11.122	4.348	10805/14025
26 immagini	8.881	2.861	11576/14849
36 immagini	0.8823	0.194	14974/18341

Tabella 5.17

Per i vertici 3D considerati nel range che descrive la "Pianta"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
14 immagini	10.098	2.708	401/590
16 immagini	8.329	2.365	532/772
18 immagini	8.026	3.447	649/917
20 immagini	54.010	41.556	550/804
22 immagini	8.723	2.698	899/1213
24 immagini	7.615	3.061	971/1386
26 immagini	6.322	1.896	1082/1474
36 immagini	0.691	0.129	1355/1689

Tabella 5.18

E2-Risultati ordinamento clustering delle immagini con range in base al valore di salienza per la simulazione "Porta Ognissanti"

Riportiamo i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti"per questo tipo di ordinamento.

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	1162.901	447.883	555/798
44 immagini	571.289	360.543	14034/18287
64 immagini	814.947	652.466	18731/24438
84 immagini	548.292	521.551	25096/33468
104 immagini	323.741	89.390	31850/41879
124 immagini	606.560	382.280	36752/48198
144 immagini	660.591	576.851	41415/53800
164 immagini	630.976	560.254	46278/59306

Tabella 5.19

Per i vertici 3D considerati nel range che descrive la "Porta Ognissanti"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	831.745	188.862	438/543
44 immagini	311.099	99.650	12144/15801
64 immagini	352.831	172.738	15750/20372
84 immagini	285.847	100.641	20370/26481
104 immagini	254.538	22.621	25553/32836
124 immagini	348.979	96.408	28790/36636
144 immagini	316.312	119.564	31440/39459
164 immagini	302.876	104.542	34470/42275

Tabella 5.20

In questo caso non si notano miglioramenti significativi per il dataset "Porta Ognissanti". Per il dataset "Pianta"è possibile notare un peggioramento dell'MSE che è comunque comprensibile visto l'aumento del numero di punti (Tabella 5.18).

F-Ordinamento delle immagini utilizzando saliency 2.5D

Nelle tecniche precedenti abbiamo ordinato le immagini in base alla loro informazione di salienza, determinata tramite il metodo di Itti & Koch che considera le componenti intensità, colore ed orientazione.
In questo tipo di ordinamento invece vogliamo fare di più, vogliamo aggiungere a queste componenti l'informazione di profondità.
L'idea è quella di definire una relazione di proporzionalità tra vicinanza e salienza, considerando un oggetto vicino più importante rispetto ad uno più lontano. Questa assunzione è giustificata dal fatto che solitamente un osservatore si colloca in una posizione vantaggiosa per osservarlo.

Per definire un'informazione di profondità ci siamo basati sulla stereopsi, la tecnica descritta nel Paragrafo 2.3, la quale, lo ricordiamo, utilizza un meccanismo che sfrutta le informazioni che arrivano dalle due retine.
Quando si fissa un oggetto, i due occhi non lo osservano dal medesimo punto di osservazione, tra essi c'è una distanza di alcuni centimetri, sufficiente a creare un differente angolo visuale. Di conseguenza l'immagine dell'oggetto che si forma sulla retina di sinistra sarà leggermente diversa da quella che si forma sulla retina di destra.
Proprio tale disparità, variando a seconda della distanza degli oggetti, (più sono distanti meno disparità c'è tra le immagini) ci fornisce le informazioni sulla profondità.
Nel nostro caso non abbiamo immagini provenienti dalle due retine; abbiamo però un'insieme di foto molto simili poiché vengono scattate da posizioni molto vicine tra loro.
Per simulare quindi il meccanismo della disparità, abbiamo deciso come prima cosa di suddividere le immagini come effettuato nell'ordinamento clustering e di scegliere per ogni visuale un'immagine "campione"che meglio rappresenta l'oggetto da quel punto di vista (Fig. 5.24).

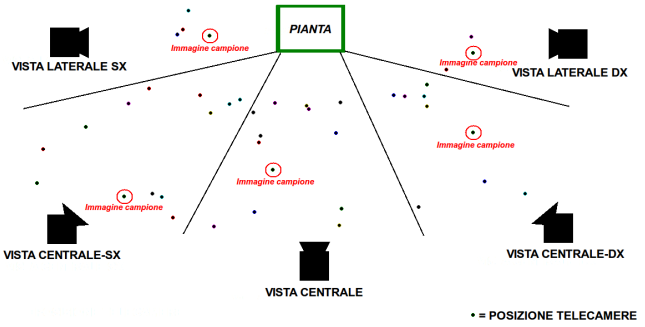


Figura 5.24: *Suddivisione clustering delle telecamere e scelta di un'immagine campione*

A questo punto, per tutte e cinque le viste determiniamo la disparità effettuando un confronto tra la mappa di salienza dell'immagine campione e la mappa di salienza di ogni immagine che appartiene a tale visuale.
Per capire come ciò sia possibile, vediamo ora un'esempio supponendo di effettuare il confronto tra la mappa di salienza dell'immagine campione I_c della vista centrale e la mappa di salienza di una qualsiasi altra immagine I' che appartiene tale insieme.
La prima operazione che si effettua è quella di determinare le feature tra queste due immagini mediante l'algoritmo SIFT visto nel Paragrafo 3.3.1. Supponiamo per semplicità, che si siano determinati 3 punti corrispondenti $\{P1, P2, P3\} \in I_c$ e $\{P1', P2', P3'\} \in I'$ (Fig.5.25).

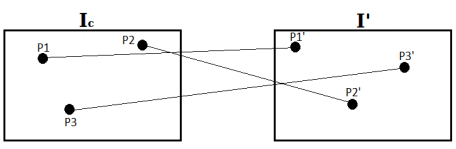


Figura 5.25: *Esempio di 3 punti corrispondenti stimati tramite SIFT*

Una volta determinata la corrispondenza tra feature, è possibile stimare tramite la trasformazione affine i punti $\{P_a, P_b, P_c\} \in I_c$ che sono le proiezioni sull'immagine I' dei punti $\{P1, P2, P3\} \in I_c$ (Fig.5.26).

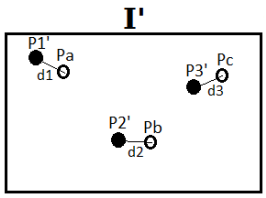


Figura 5.26: *Trasformazione affine di* $\{P1, P2, P3\}$

In questo modo, possiamo pensare alla distanza tra queste coppie di punti $d1, d2$ e $d3$ (Fig.5.26), come alla disparità presente nelle due retine descritta precedentemente:

$$d1 = \sqrt{(P_{ax} - P1'_x)^2 - (P_{ay} - P1'_y)^2}$$

$$d2 = \sqrt{(P_{bx} - P2'_x)^2 - (P_{by} - P2'_y)^2}$$

$$d3 = \sqrt{(P_{cx} - P3'_x)^2 - (P_{cy} - P3'_y)^2}$$

Inoltre possiamo calcolare l'errore quadratico medio, come:

$$MSE = E[d] = \frac{d1^2 + d2^2 + d3^2}{3}$$

A questo punto possiamo definire la relazione che lega la disparità d_i e la salienza S_i del punto corrispondente P'_i, come:

$$S'_i = S_i + \lambda \frac{d_i^2}{MSE_{medio}} \qquad con \;\; i \in \{1, 2, 3\}$$

dove:

- λ= coefficiente moltiplicativo
- MSE_{medio}=MSE medio di tutti quelli determinati tra le coppie di immagini per ogni vista

Nella relazione precedente si nota come il nuovo valore di salienza S'_i relativo al punto Pi' nell'immagine I', aumenti all'aumentare della distanza tra la proiezione di Pi ed il punto Pi' (disparità). Questo va in accordo con quanto detto precedentemente, infatti maggiore è la disparità, minore è la profondità dell'oggetto e di conseguenza risulta più importante all'interno della scena descritta.
Una volta stimati i nuovi valori di salienza dei punti corrispondenti alle feature, possiamo (considerando solo tali punti) effettuare un ordinamento di tipo clustering come

visto nell'ordinamento (C).
Durante lo studio di questo metodo abbiamo cercato di ottimizzare il coefficiente moltiplicativo λ, al fine di definire il corretto legame tra salienza e profondità. In particolare abbiamo utilizzato $\lambda = \{5, 10, 15, 20, 25, 30\}$ ed i risultati migliori si sono ottenuti con $\lambda = 15$.
Per tale motivo presentiamo ora solo i valori ottenuti per le due simulazione "Pianta"e "Porta Ognissanti"con $\lambda = 15$.

E1-Risultati ordinamento delle immagini utilizzando saliency 2.5D per la simulazione "Pianta"

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V^P_{TOT}
14 immagini	14.563	6.606	5605/6308
16 immagini	11.095	3.651	6717/7593
18 immagini	12.325	4.418	8004/9061
20 immagini	11.255	3.974	8371/9490
22 immagini	14.001	5.926	9545/10282
24 immagini	14.161	8.512	10107/11313
26 immagini	1.054	1.399	8950/12098
36 immagini	0.004	0.003	13074/16610

Tabella 5.21

Per i vertici 3D considerati nel range che descrive la "Pianta"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V^P_{TOT}
14 immagini	14.837	2.491	173/227
16 immagini	7.874	3.322	374/448
18 immagini	8.702	2.398	323/426
20 immagini	9.367	3.299	315/416
22 immagini	9.073	3.605	469/577
24 immagini	10.205	2.203	455/582
26 immagini	0.637	0.043	574/794
36 immagini	0.0007	0.0005	1038/1338

Tabella 5.22

E2-Risultati ordinamento delle immagini utilizzando saliency 2.5D per la simulazione "Porta Ognissanti"
Riportiamo i risultati ottenuti per la simulazione tridimensionale della "Porta Ognissanti"con questo tipo di ordinamento.

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	916.501	834.603	329/546
44 immagini	1511.936	881.378	9477/14409
64 immagini	831.070	666.182	15576/23331
84 immagini	1143.32693	934.88637	20111/29210
104 immagini	1229.938	1071.403	26210/36623
124 immagini	848.615	750.134	34264/45711
144 immagini	752.834	660.325	41858/54228
164 immagini	625.917	525.167	46272/59312

Tabella 5.23

Per i vertici 3D considerati nel range che descrive la "Porta Ognissanti"si ottiene:

N° IMMAGINI	MSE	MSE_{ICP}	V_{CORR}/V_{TOT}^{P}
24 immagini	230.352	134.253	258/343
44 immagini	877.357	258.340	8305/11809
64 immagini	389.712	170.279	12784/18056
84 immagini	469.982	253.672	15968/22229
104 immagini	554.243	292.591	20095/26954
124 immagini	349.160	168.071	24877/32310
144 immagini	322.742	148.876	29375/36528
164 immagini	310.153	129.924	33285/41572

Tabella 5.24

In questo caso non c'è un aumento significativo della precisione ma è possibile notare un andamento progressivo della qualità rispetto al numero di immagini.

5.4 Analisi dei risultati

Dopo aver presentato le tecniche di ordinamento effettuate, risulta opportuno attuare un confronto tra i valori ottenuti dall'analisi di questi metodi, in modo da poterne analizzare vantaggi e svantaggi.

Per comprendere meglio i grafici che seguiranno, ricordiamo le tecniche viste in precedenza:

A- Ordinamento delle immagini in maniera random

B- Ordinamento delle immagini in base al valore di salienza

C- Ordinamento clustering delle immagini in base al valore di salienza

D- Ordinamento delle immagini con range in base al valore di salienza

E- Ordinamento clustering delle immagini con range in base al valore di salienza

F- Ordinamento delle immagini utilizzando saliency 2.5D

5.4.1 Analisi della densità del modello tridimensionale

Per analizzare la densità del modello tridimensionale "Pianta", nelle quattro figure successive sono riportati rispettivamente:

- (Fig. 5.27) Numero di vertici totali stimati nella simulazione parziale
- (Fig. 5.28) Numero di vertici corrispondenti tra le simulazioni parziali e quella totale
- (Fig. 5.29) Numero di vertici totali stimati nella simulazione parziale appartenenti al range di interesse
- (Fig. 5.30) Numero di vertici corrispondenti tra le simulazioni parziali e quella totale appartenenti al range di interesse

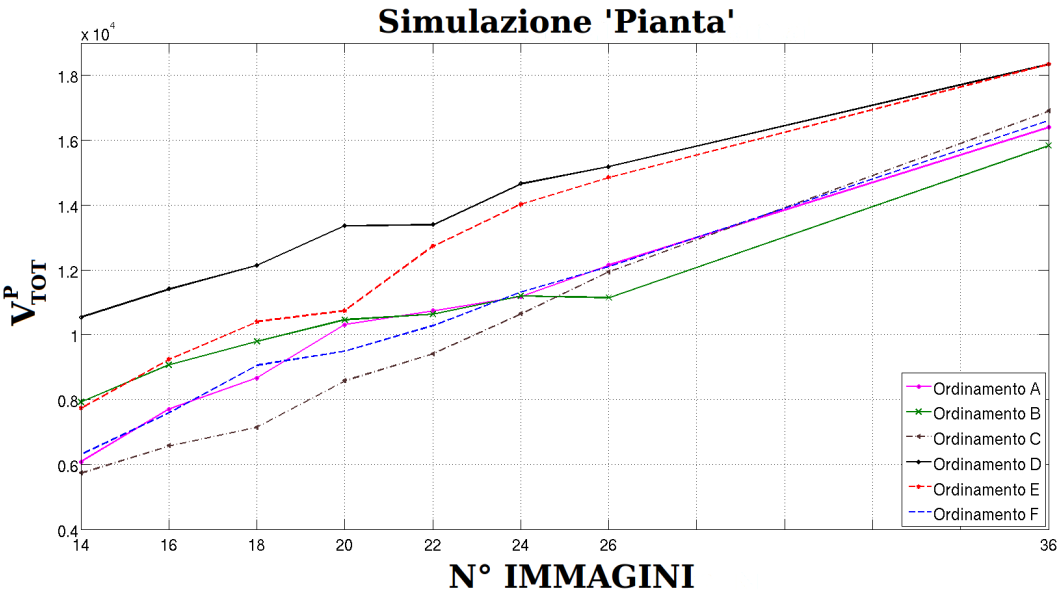


Figura 5.27: *Numero di vertici totali stimati nella simulazione parziale "Pianta"*

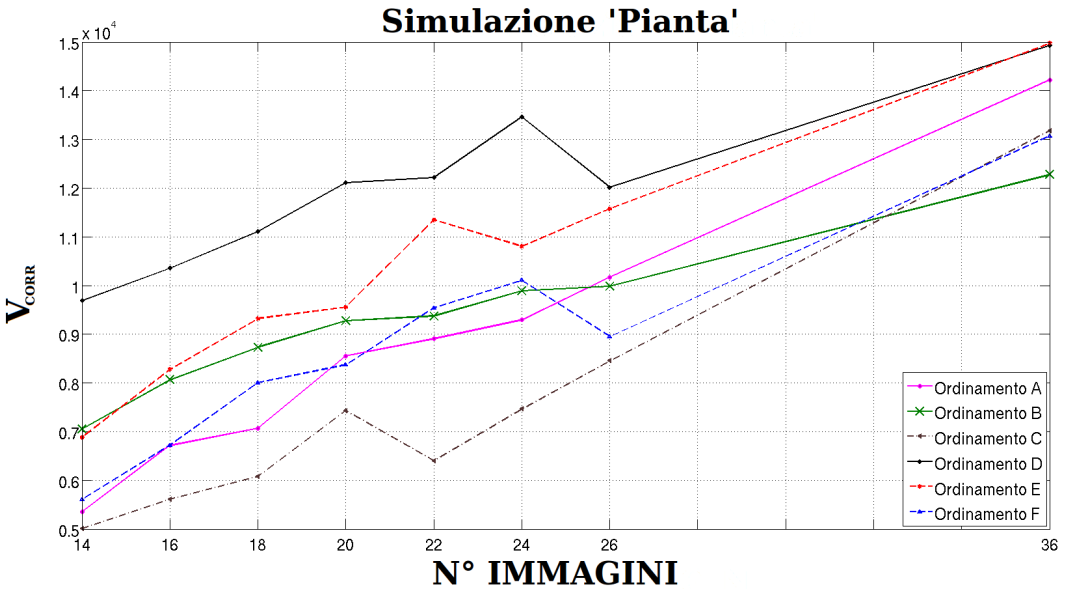


Figura 5.28: *Numero di vertici corrispondenti tra le simulazioni parziali e quella totale "Pianta"*

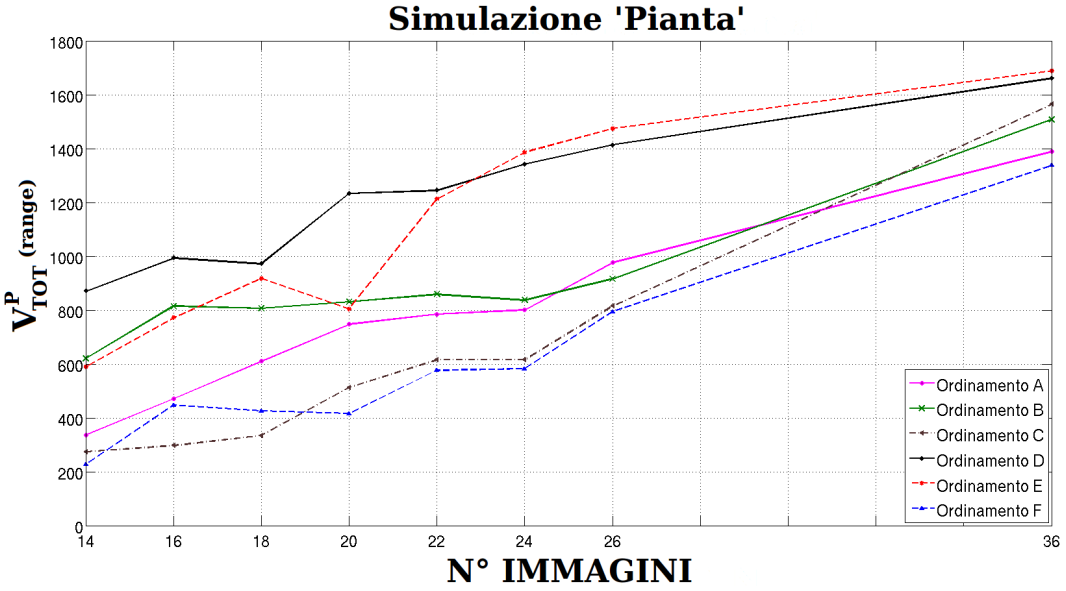


Figura 5.29: *Numero di vertici totali stimati nella simulazione parziale appartenenti al range di interesse "Pianta"*

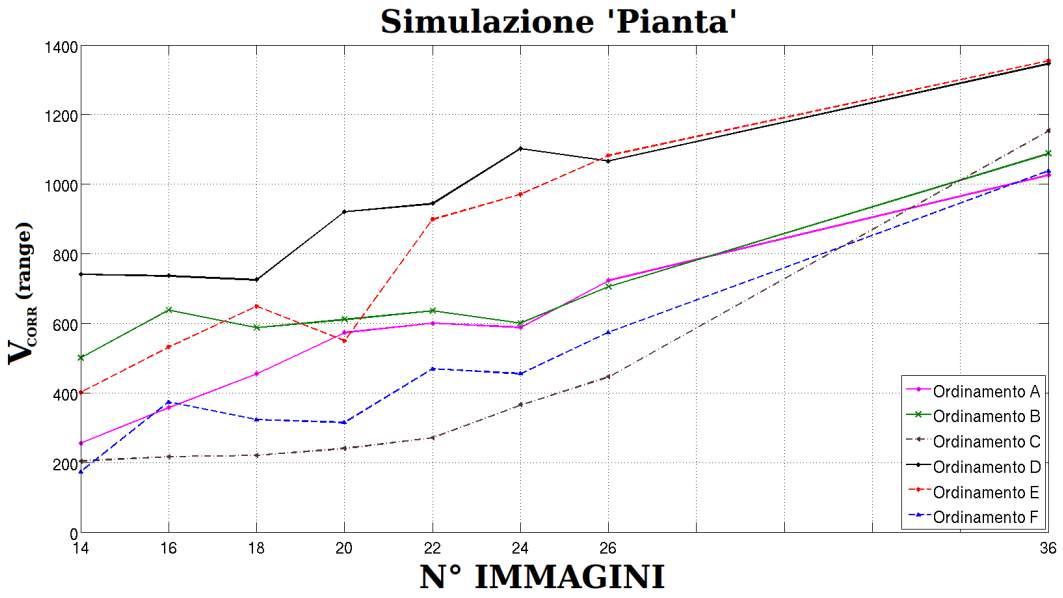


Figura 5.30: *Numero di vertici corrispondenti tra le simulazioni parziali e quella totale appartenenti al range di interesse "Pianta"*

Dai quattro grafici precedenti osserviamo che rispetto l'ordinamento di tipo casuale (Ordinamento A), l'ordinamento con range in base al valore di salienza (Ordinamento D), permette un incremento di circa il 70% del numero dei vertici stimati. Similmente, l'ordinamento clustering delle immagini con range (Ordinamento E), permette un incremento di circa il 30%.
A differenza di questi due, l'ordinamento per valore di salienza che non utilizza il range (Ordinamento B) permette un buon incremento (circa il 30%) per simulazioni che utilizzano un numero molto ridotto di immagini; aumentandole, si allinea ai valori ottenuti per l'ordinamento casuale.
Se invece utilizziamo la tecnica saliency 2.5D (Ordinamento F), otteniamo delle prestazioni molto simili all'ordinamento casuale, l'unica differenza è che il numero di vertici 3D appartenenti all'oggetto protagonista (la pianta), risulta decisamente inferiore.
Concludiamo l'analisi osservando come l'ordinamento clustering senza range (Ordinamento C), abbia, per un numero molto ridotto di foto, delle prestazioni addirittura inferiori rispetto l'ordinamento casuale. Con l'aumentare del numero di immagini tale ordinamento segue l'andamento random.
Ciò, molto probabilmente, è dovuto al fatto che se selezioniamo immagini provenienti da cinque viste differenti, facilmente queste saranno scattate da posizioni molto distanti tra loro, di conseguenza il numero di punti corrispondenti tra queste immagini sarà molto basso.
Ora, al pari di quanto fatto per la simulazione "Pianta", riportiamo i valori stimati per la simulazione "Porta Ognissanti".

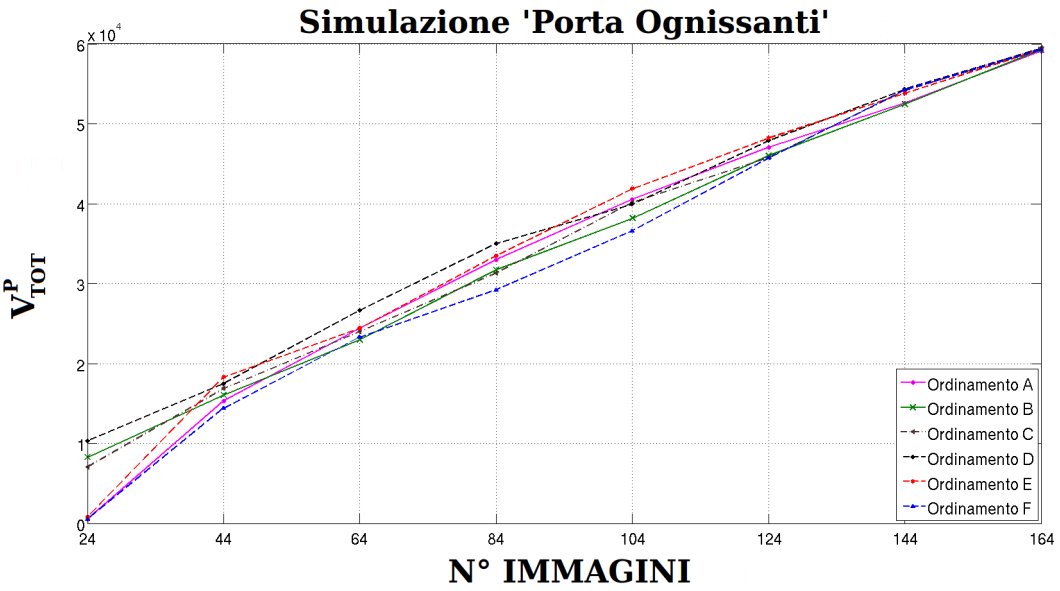


Figura 5.31: *Numero di vertici totali stimati nella simulazione parziale "Porta Ognissanti"*

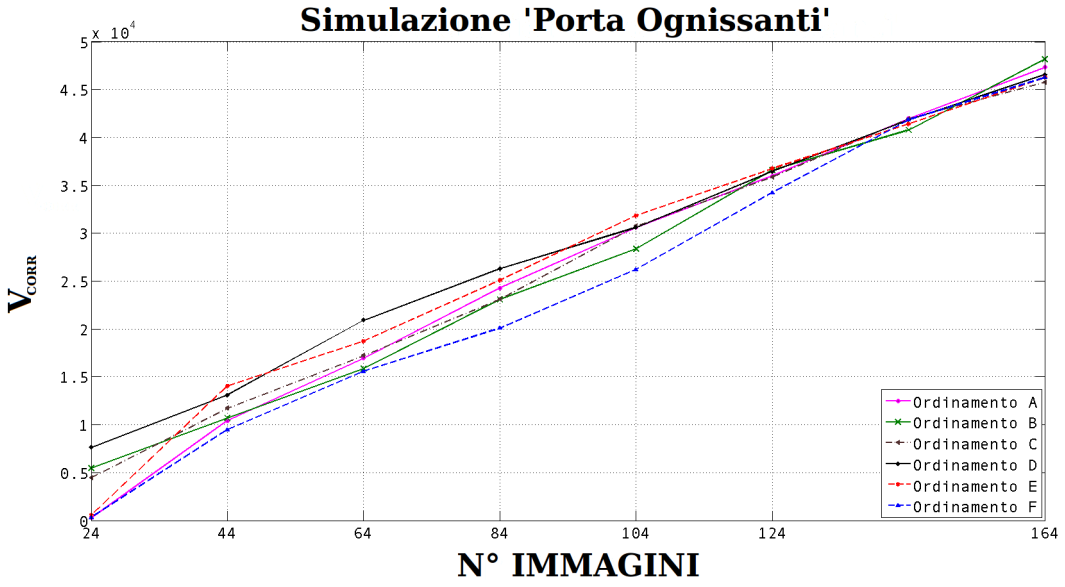


Figura 5.32: *Numero di vertici corrispondenti tra le simulazioni parziali e quella totale "Porta Ognissanti"*

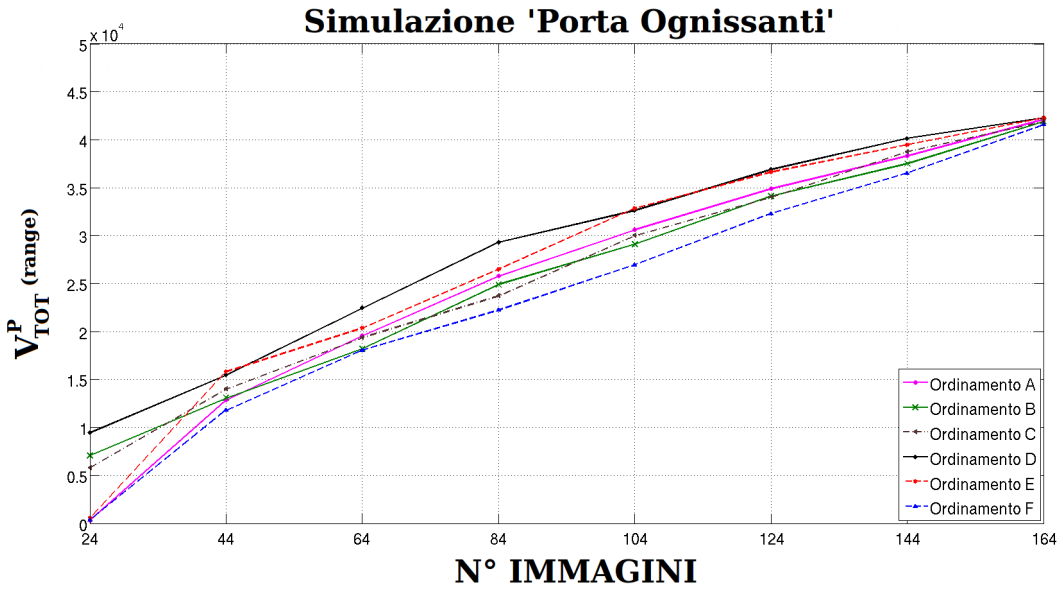


Figura 5.33: *Numero di vertici totali stimati nella simulazione parziale appartenenti al range di interesse "Porta Ognissanti"*

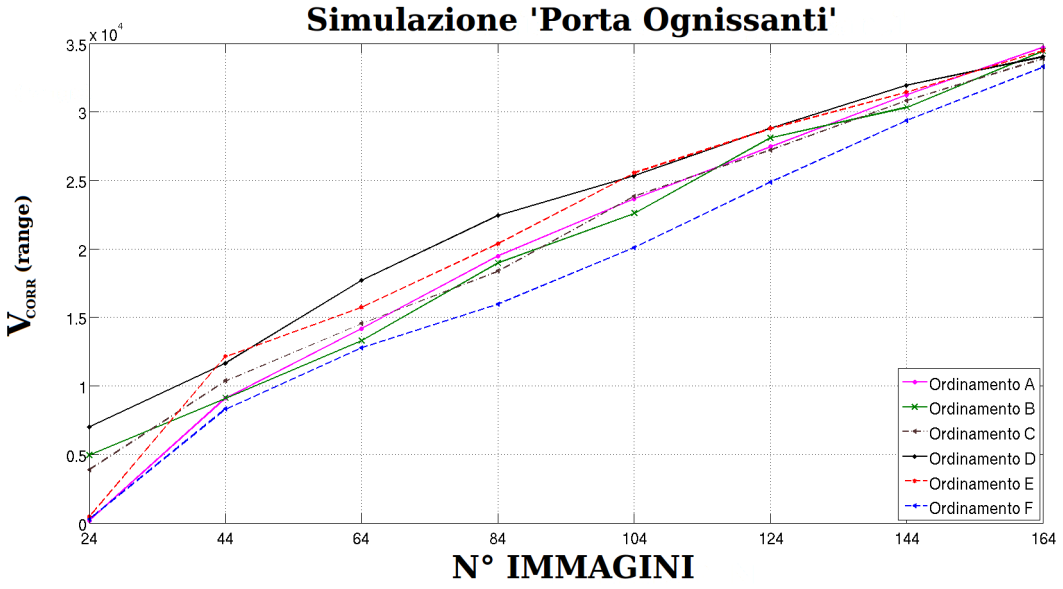


Figura 5.34: *Numero di vertici corrispondenti tra le simulazioni parziali e quella totale appartenenti al range di interesse "Porta Ognissanti"*

Dai quattro grafici precedenti possiamo notare come ci sia un notevole vantaggio per alcuni tipi di ordinamenti quando il numero di immagini sia molto ridotto.
Infatti per l'ordinamento clustering (Ordinamento C) e per gli ordinamenti in base al valore di salienza, con e senza range (Ordinamento D e B); risulta possibile effettuare un discreto modello tridimensionale con solo 24 immagini.
Aumentando il numero di immagini però, a differenza dei risultati visti in precedenza

per la simulazione “Pianta”, non otteniamo prestazioni decisamente migliori rispetto l’ordinamento di tipo casuale, il massimo che riusciamo ad ottenere è un incremento del numero stimato di vertici del 10% (Ordinamento D e E).

5.4.2 Analisi della precisione del modello tridimensionale

Per analizzare la precisione del modello tridimensionale “Pianta”, nelle quattro figure successive sono riportati rispettivamente:

- (Fig. 5.35) Errore quadratico medio (MSE) dei vertici corrispondenti
- (Fig. 5.36) Errore quadratico medio (MSE) dei vertici corrispondenti utilizzando l’algoritmo ICP
- (Fig. 5.37) Errore quadratico medio (MSE) dei vertici corrispondenti appartenenti al range di interesse
- (Fig. 5.38) Errore quadratico medio (MSE) dei vertici corrispondenti appartenetni al range di interesse utilizzando l’algoritmo ICP

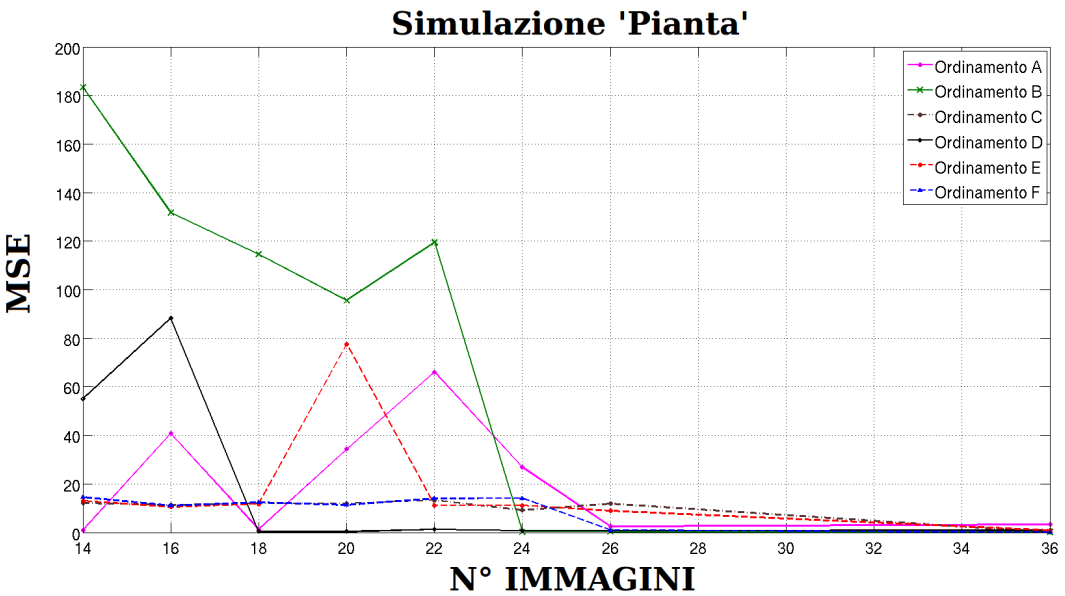


Figura 5.35: *Errore quadratico medio (MSE) dei vertici corrispondenti (simulazione “Pianta”)*

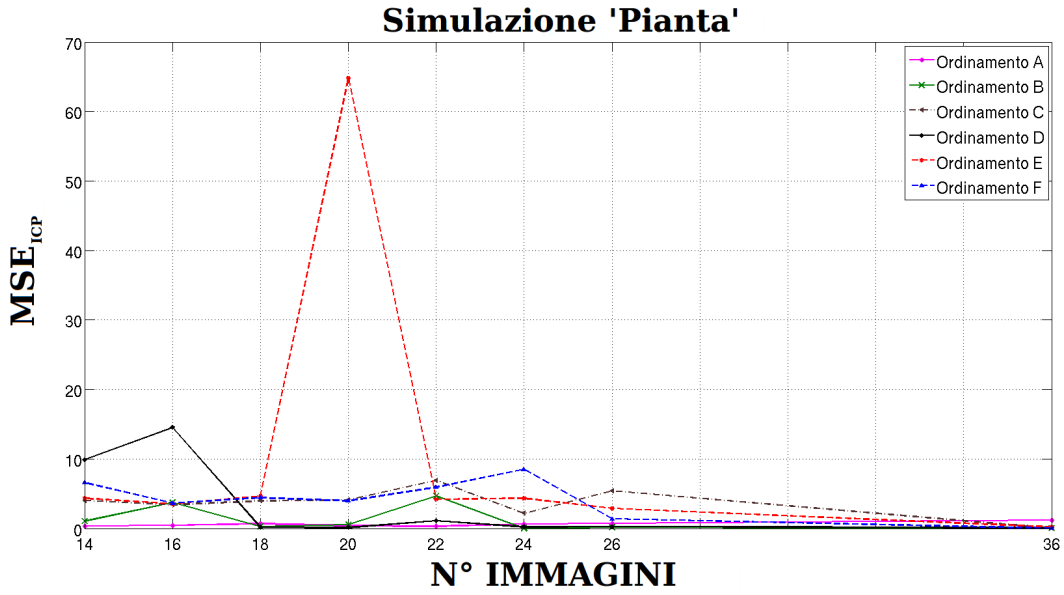


Figura 5.36: *Errore quadratico medio (MSE) dei vertici corrispondenti utilizzando l'algoritmo ICP (simulazione "Pianta")*

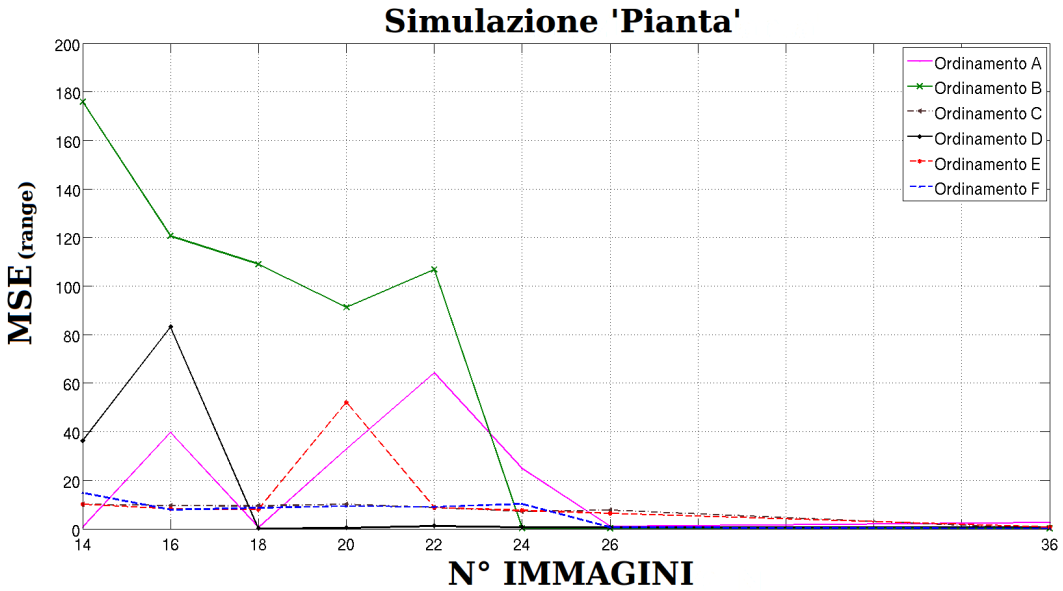


Figura 5.37: *Errore quadratico medio (MSE) dei vertici corrispondenti appartenenti al range di interesse (simulazione "Pianta")*

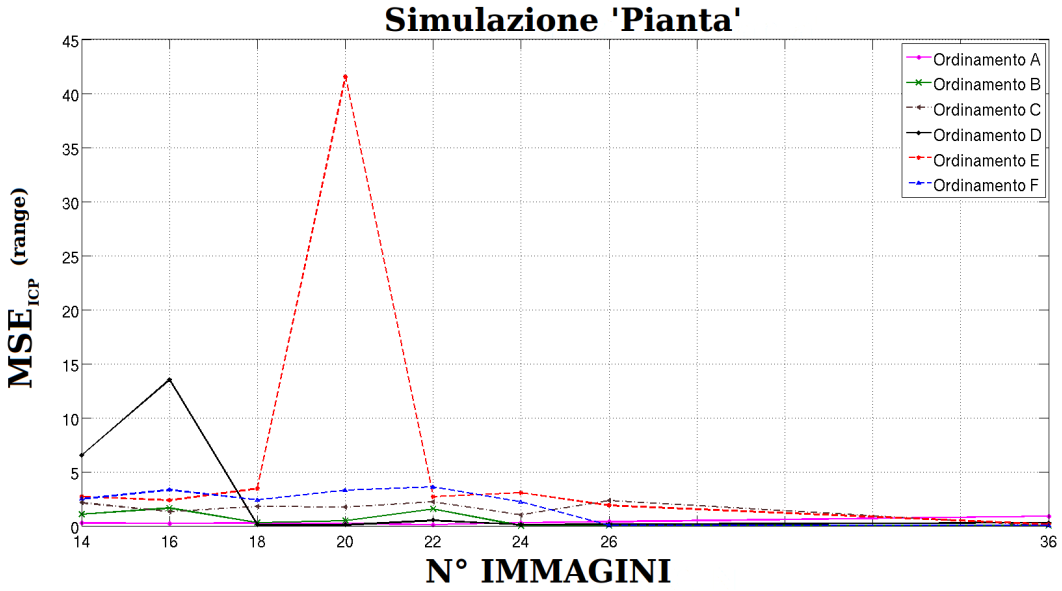


Figura 5.38: *Errore quadratico medio (MSE) dei vertici corrispondenti appartenenti al range di interesse utilizzando l'algoritmo ICP (simulazione "Pianta")*

Dai quattro grafici precedenti possiamo notare come l'andamento dell'errore quadratico medio non sia lineare ma dipenda molto da ogni immagine utilizzata per la simulazione parziale.
In particolare osserviamo gli andamenti dell'MSE per gli ordinamenti B, D ed E che come visto in precedenza garantiscono nella simulazione "Pianta", la stima di un maggior numero di vertici 3D rispetto l'ordinamento casuale.
Abbiamo visto in precedenza che l'ordinamento B garantisce un incremento del 30% dei vertici 3D per un numero ridotto di immagini. Purtroppo però in tale range possiede un elevato valore di MSE.
È possibile notare inoltre come tale MSE decada drasticamente quando si utilizza un numero di immagini superiore a 24; l'andamento dei vertici però, per tali valori è allineato a quello random, quindi non si hanno grossi vantaggi.
L'ordinamento D invece, garantisce un incremento del numero di vertici di circa il 70%, osservando i grafici precedenti si nota come tale ordinamento abbia un MSE elevato solo per un numero molto ridotto di immagini. Se si supera la soglia delle 18 foto utilizzate, l'MSE diminuisce sensibilmente. Utilizzando quindi un numero maggiore di immagini si possono avere dei modelli tridimensionali più densi e precisi.
Le osservazioni fatte per quest'ultimo ordinamento, possono essere espresse anche per l'ordinamento E. Le due differenze principali sono l'incremento del numero di vertici, in questo caso circa il 30% e la soglia che definisce il decadimento dell'MSE, 24 foto.
Ora, al pari di quanto appena descritto per la simulazione "Pianta", riportiamo i valori stimati per la simulazione "Porta Ognissanti".

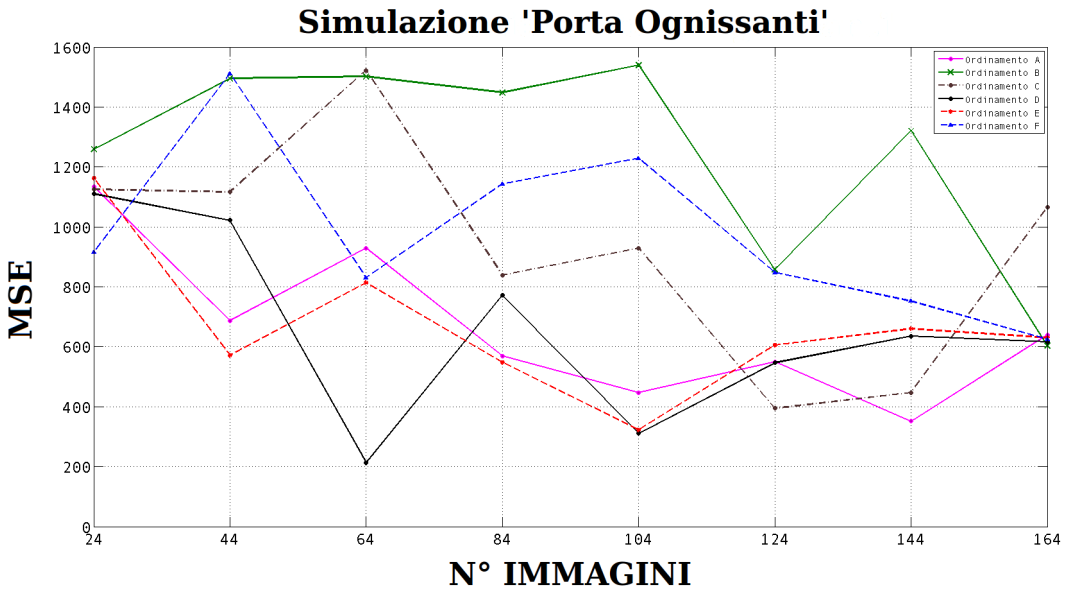


Figura 5.39: *Errore quadratico medio (MSE) dei vertici corrispondenti (simulazione "Porta Ognissanti")*

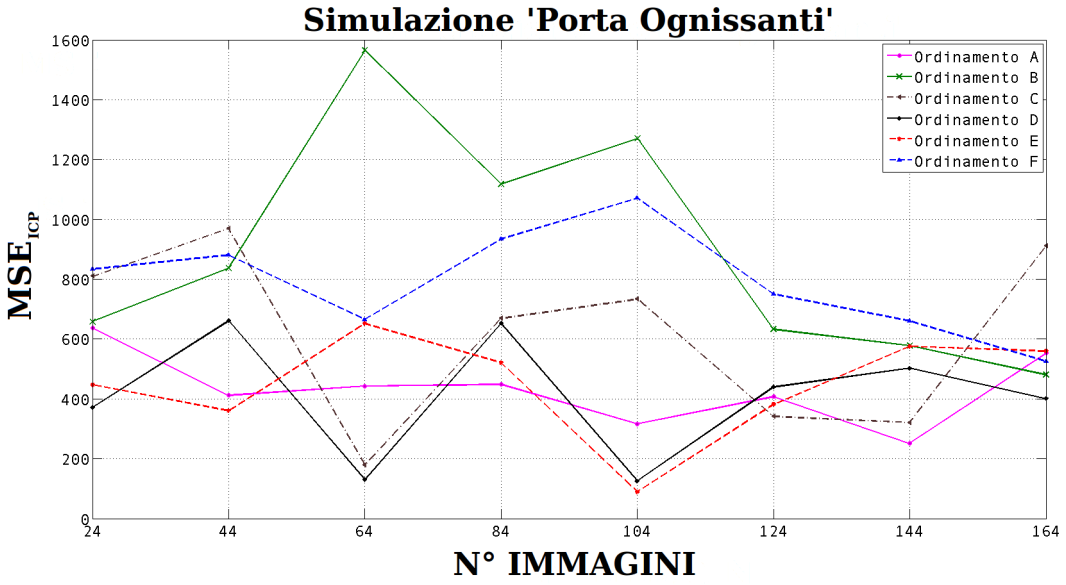


Figura 5.40: *Errore quadratico medio (MSE) dei vertici corrispondenti utilizzando l'algoritmo ICP (simulazione "Porta Ognissanti")*

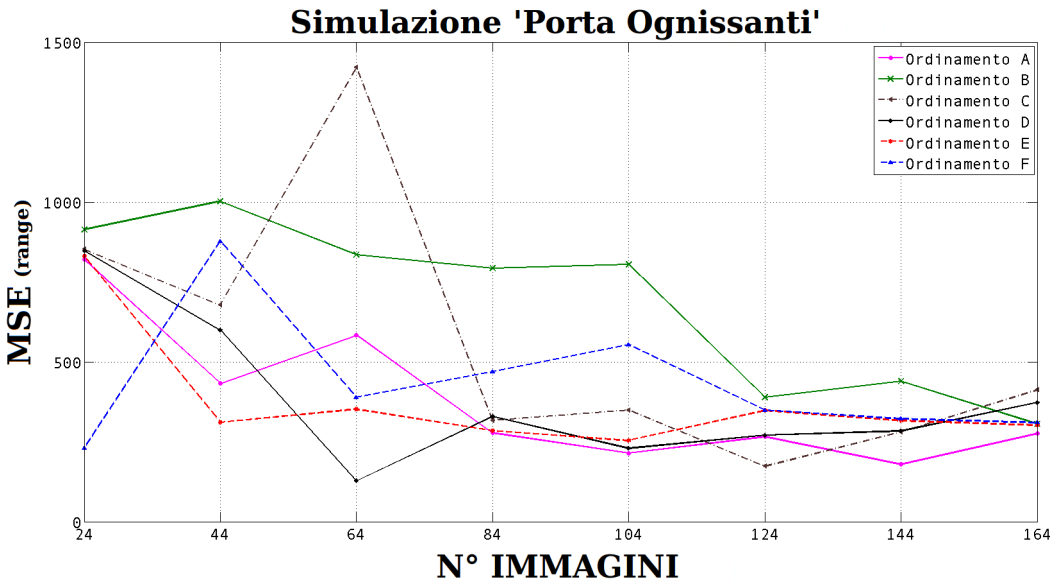


Figura 5.41: *Errore quadratico medio (MSE) dei vertici corrispondenti appartenenti al range di interesse (simulazione "Porta Ognissanti")*

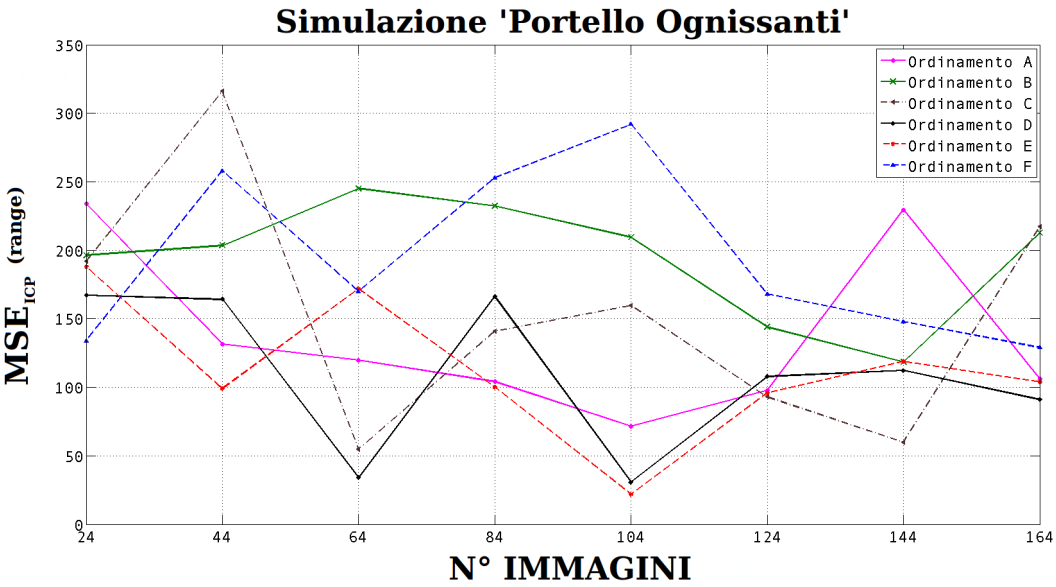


Figura 5.42: *Errore quadratico medio (MSE) dei vertici corrispondenti appartenenti al range di interesse utilizzando l'algoritmo ICP (simulazione "Porta Ognissanti")*

In precedenza avevamo visto come per la simulazione "Porta Ognissanti"solo gli ordinamento C, D e B permettessero di realizzare con solo 24 immagini un modello tridimensionale denso.
Per tale numero di foto, questi tre ordinamenti presentano valori di MSE soddisfacenti, quindi oltre alla densità è possibile avere anche una buona precisione della geometria del modello.
Un'ultima osservazione che possiamo fare riguarda gli ordinamenti D ed E che garanti-

scono un incremento dei vertici 3D stimati di circa il 10%. Come è possibile osservare nei 4 grafici precedenti, i valori di MSE per queste due tecniche sono analoghi a quelli determinati per l'ordinamento casuale. A parità quindi di precisione, utilizzando gli ordinamenti D ed E possiamo avere un modello più denso del 10%.

Capitolo 6

Conclusioni e sviluppi futuri

In questa tesi sono stati presentati i metodi sviluppati per la realizzazione di un buon modello tridimensionale a partire da un numero ridotto di foto.
Per perseguire il nostro obiettivo siamo partiti dall'idea che si potessero sfruttare le mappe di salienza per selezionare in maniera "ottimale "le immagini. Queste mappe che simulano la percezione visiva umana, permettono di indentificare l'oggetto principale della scena, il quale sarà lo stesso di cui si vuole effettuare la modellazione 3D. Una volta individuato, selezionando le foto che lo rappresentano nel modo migliore, abbiamo supposto fosse possibile ridurre il costo computazionale (visto che si utilizza un numero parziale di foto) ed inoltre garantire una discreta qualità del modello realizzato.
Le considerazioni esposte, hanno poi trovato riscontro nei risultati sperimentali ottenuti. Infatti come analizzato nel capitolo precedente per alcuni di questi algoritmi risulta possibile ottenere (rispetto ad una scelta delle foto di tipo random), dei modelli nettamente più densi e con precisione comparabile. In particolare, per i metodi che ordinano le immagini in base al valore di salienza con e senza range, si ottiene per la simulazione "Pianta"un incremento del numero di vertici 3D dal 30% al 70%, per la simulazione "Porta Ognissanti "invece l'incremento è inferiore, circa il 10%.
Nel range d'immagini in cui questi ordinamenti parziali presentano questa sostanziale crescita, corrispondono valori di MSE paragonabili con quelli determinati per l'ordinamento di tipo casuale. Quindi a parità di precisione del modello, i metodi basati sulla salienza hanno una geometria notevolmente più densa.
Dal differente incremento ottenuto per le due simulazioni è possibile rimarcare l'importanza delle mappe di salienza ai fini dell'ottimizzazione del modello 3D. Infatti come avevamo visto nel capitolo 4, le saliency map computate per la simulazione "Pianta"riescono ad identificare in maniera univoca l'oggetto protagonista della scena, la pianta appunto. Ciò è dovuto al fatto che la salienza stimata per questa scena (che si svolge in un ambiente chiuso) è affetta da poche componenti esterne.
Nella simulazione "Porta Ognissanti"invece, che rappresenta un ambiente aperto soggetto a numerosi fattori di rumore, l'individuazione non è così inequivocabile. Ecco quindi che queste considerazioni giustificano i diversi incrementi stimati per le due simulazioni. Migliore è l'individuazione da parte delle saliency dell'oggetto di interesse

migliore sarà anche la densità del modello 3D.
Con l'obiettivo di migliorare le prestazioni descritte si potrebbe sviluppare un algoritmo di riconoscimento di forme, in modo da selezionare automaticamente l'oggetto di cui si vuole effettuare la modellazione tridimensionale in maniera sicuramente più stringente di quanto fatto in questa tesi.
Meritano delle considerazioni anche i metodi presentati che non hanno portato grossi vantaggi rispetto all'ordinamento di tipo casuale. Ci riferiamo all'ordinamento di tipo clustering e a quello che sfrutta la saliency 2.5D. Abbiamo visto infatti che per un numero molto ridotto di foto, questo primo algoritmo porta alla scelta di immagini così distanti tra loro da presentare pochi punti corrispondenti. È plausibile quindi pensare di miglioralo ottimizzando la suddivisione delle viste. Una soluzione effettuabile potrebbe essere quella di aumentare il numero di visuali. In questo modo si descrive facilmente l'oggetto da ogni sua angolazioni ed inoltre le immagini, risultando più vicine, garantiscono l'individuazione di un numero molto più corposo di punti corrispondenti.
Per quanto riguarda invece l'ordinamento che sfrutta le saliency 2.5D, si potrebbe cercare di sviluppare una metodologia che, compatibilmente con la complessità della scena, permetta di ottimizzare la stima di λ. Così facendo risulterebbe possibile definire il corretto legame tra profondità e salienza per qualsiasi modello.
In ottica futura lo sviluppo degli algoritmi presentati in questa tesi può risultare interessante principalmente per i campi applicativi della robotica e della modellazione 3D su dispositivi portatili.
Nel primo infatti se si utilizza un dispositivo robot per la generazione di una geometria tridimensionale, risulta lecito pensare di sfruttare gli algoritmi presentati in maniera tale da poterne guidarne i movimenti verso i punti più salienti. In questo modo si può selezionare e ridurre l'enorme quantitativo di dati proveniente dall'intera scena andando ad acquisire con maggior dettaglio solo le parti più interessanti, tralasciando così le informazioni inutili del contorno.
Per quanto riguarda invece la modellazione su dispositivi portatili come tablet o cellulari di ultima generazione, è verosimile pensare di applicare gli algoritmi presentati a set di foto realizzate mediante la fotocamera del dispositivo. L'utente quindi scattando un numero molto esiguo di foto vista la ridotta potenza di calcolo di questi dispositivi, può selezionando o meno nell'immagine l'oggetto di interesse (ordinamento con e senza range) sfruttare le mappe di salienza per generare un buon modello 3D.

Bibliografia

[1] Brian Curless, "Overview of active vision techniques", in Course on 3D Photography, SIGGRAPH 2000.

[2] Andrea Fusiello, "Metodi ottici", in book "Visione Computazionale (appunti delle lezioni)", pp.5-7 ,(3rd revision) 2009.

[3] Paolo Chistè, "La fotocamera", in paper "La fotografia di documentazione archeologica in digitale: introduzione alla fotografia digitale e alle tecniche fotografiche", lezione 5, 2009-2010.

[4] Andrea Fusiello, "Formazione dell'immagine ", in book "Visione Computazionale (appunti delle lezioni)", pp.26-33 ,(3rd revision) 2009.

[5] Andrea Fusiello, "Stereopsi", in book "Visione Computazionale (appunti delle lezioni)", pp.68-81 ,(3rd revision) 2009.

[6] Andrea Fusiello, "Calibrazione della fotocamera ", in book "Visione Computazionale (appunti delle lezioni)", pp.34-51 ,(3rd revision) 2009.

[7] Kanade T. & Okutomi M. "A stereo matching algorithm with an adaptive window: Theory and experiments". IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 920-932, 1994.

[8] Richard Hartley, "Self-calibration of stationary cameras". International Journal of Computer Vision, pp. 5-24, 1997.

[9] Andrea Fusiello, "Moto e struttura", in book "Visione Computazionale (appunti delle lezioni)", pp.115-130 ,(3rd revision) 2009.

[10] Huang T. S. & Netravali A. N. "Motion and structure from feature correspondences". A review Proceedings of IEEE, pp. 252-267, 1994.

[11] Christopher Longuet-Higgins "A computer algorithm for reconstructing a scene from two projections". Nature, pp. 133-135, 1981.

[12] Huang T. & Faugeras O. "Some properties of the E matrix in two-view motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence", pp. 1310-1312 , 1989.

[13] Noah Snavely, Steven M. Seitz, Richard Szeliski, "Photo tourism: Exploring photo collections in 3D"ACM Transactions on Graphics (SIGGRAPH Proceedings), 2006,

[14] Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150-1157.

[15] Martin A. Fischler and Robert C. Bolles "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". Comm. of the ACM 24: 381-395, June 1981.

[16] H ARTLEY, R. I., AND Z ISSERMAN, "Multiple View Geometry". Cambridge University Press, Cambridge, UK, 2004.

[17] NOCEDAL , J. & W. RIGHT, S. J. "Numerical Optimization. Springer Series in Operations Research". Springer-Verlag, New York, NY, 1999.

[18] Noah Snavely, Steven M. Seitz, Richard Szeliski. "Modeling the World from Internet Photo Collections". International Journal of Computer Vision (to appear), 2007.

[19] M.I.A. Lourakis and A.A. Argyros. "The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm". Tech. Rep. 340, Inst. of Computer Science-FORTH, Heraklion, Crete, Greece.

[20] Yasutaka Furukawa and Jean Ponce, "Accurate, dense, and robust multiview stereopsis,"IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 8, pp. 1362-1376, 2010

[21] Itti, L. and Koch, C. and Niebur, E. and others, "A model of saliency-based visual attention for rapid scene analysis". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998 pp. 1254-1259.

[22] Nicolas Tsapatsoulis, Konstantinos Rapantzikos "Wavelet Based Estimation of Saliency Maps in Visual Attention Algorithms". ICANN pp.538-547, 2006

[23] C. Koch and S. Ullman, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry,"Human Neurobiology, vol. 4, pp. 219-227, 1985.

[24] A.M. Treisman and G. Gelade, "A Feature-Integration Theory of Attention,"Cognitive Psychology, vol. 12, no. 1, pp. 97-136, Jan. 1980.

[25] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C.H. Anderson, "Overcomplete Steerable Pyramid Filters and Rotation Invariance"Proc. IEEE Computer Vision and Pattern Recognition, pp. 222-228, Seattle, Wash., June 1994.

[26] C. Koch and S. Ullman, “Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry,”Human Neurobiology, vol. 4, pp. 219-227, 1985.

[27] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation”, IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, pp 674-693, 1989.