UNIVERSITÀ DEGLI STUDI DI PADOVA

Stima di modelli tridimensionali da immagini bidimensionali

Marco Donà

Facoltà di Ingegneria
CORSO DI LAUREA IN INGEGNERIA DELLE TELECOMUNICAZIONI

17 Luglio 2012

"Ricerca è ciò che faccio quando non so che cosa sto facendo."

Wernher von Braun

Abstract

Il lavoro presentato in questa tesi si è focalizzato sull'analisi dello stato dell'arte e delle tecniche attualmente utilizzate nella ricostruzione di modelli tridimensionali. Fra tutte le possibili soluzioni, consideriamo il caso estremo nel quale non sia possibile recuperare dati di profondità attraverso sensori 3D e in cui sia a nostra disposizione un solo frame/immagine della scena. Tale condizione richiede di affrontare il problema in maniera sostanzialmente diversa dalle soluzioni classiche. La singola immagine offre una quantità di informazioni ridotto. Un esempio su tutti è rappresentato dalle occlusioni presenti nella scena, ovvero parti della scena tridimensionale rese non visibili a causa della prospettiva di osservazione, o di oggetti interposti tra l'occlusione stessa e il punto dal quale è osservata la scena. La soluzione trovata non sarà dunque univoca, ma dovrà esser scelta tra un'infinità di soluzioni possibili. Da un lato va presa in considerazione la qualità visiva della soluzione scelta, dall'altra va considerata un'analisi prestazionale dell'algoritmo, in termini di errore sui dati, calcolati su apposite immagini di training delle quali è disponibile il dato di profondità, precedentemente acquisito tramite scansione laser. Dopo aver visto quali sono gli algoritmi usati in tale scenario, si pone particolare attenzione ad uno di essi: Make 3D. Successivamente si andranno a sviluppare alcune idee e concetti poi utilizzati nella modifica del codice e si confronteranno i risultati ottenuti con quelli di partenza. I due tipi di ottimizzazione non necessariamente presentano un andamento lineare: quella che dati alla mano risulta essere la soluzione ottimale non necessariamente risulta anche visivamente migliore di una soluzione più povera, che magari va ad "appiattire" dettagli irrilevanti o, addirittura, fastidiosi all'occhio umano. In questo tipo di ricostruzione dunque va cercato il compromesso migliore tra qualità visiva/soggettiva e qualità reale/oggettiva.

Indice

Elenco delle figure

Ai miei genitori...

Capitolo 1

Modelli 3D: principi base e creazione

1.1 Generazione di modelli 3D

Nell'ambito della visione computazionale la generazione di strutture tridimensionali a partire da proiezioni bidimensionali, è, ancora oggi, un problema aperto. Diversi sono i metodi usati, e di nuovi ne vengono sempre proposti. Questo primo capitolo analizza i metodi per la generazione di modelli tridimensionali di oggetti reali. Le proprietà intrinseche all'immagine, l'analisi delle diverse componenti che collegano proiezioni 2D e modello 3D associato, vengono qui presentati.

I metodi per l'acquisizione automatica dei volumi di un oggetto sono svariati. Una prima classificazione può dividere sistemi riflessivi e trasmissivi (es. raggi X). Per le finalità e gli scopi di questa tesi verranno presi in esame solo i sistemi di tipo riflessivo, in particolare quelli ottici, quelli che cioè operano con la luce riflessa dagli oggetti, allo stesso modo del nostro sistema visivo.

Il principio base dell'*image base modeling* è molto semplice: gli oggetti irradiano luce visibile, questa può essere catturata attraverso l'uso di una convenzionale telecamera. Le caratteristiche della luce dipendono da diversi fattori quali: illuminazione della scena, geometria delle superfici, riflettanza delle superfici stesse. L'analisi al calcolatore permette poi di stimare la natura 3D degli oggetti. Le diverse tecniche vengono classificate soprattutto in base all'impiego (o meno) di

diverse fonti di illuminazione esterne. Distinguiamo dunque tra due importanti categorie:

> *metodi attivi:* sistemi che irradiano la scena di interesse con opportune radiazioni elettromagnetiche (n.b: volendo anche a frequenze non visibili all'occhio umano, es. infrarossi). In questo caso si ricorre all'uso di pattern luminosi, luce laser, radiazioni IR, etc.
>
> *metodi passivi:* si basano esclusivamente sull'analisi di immagini di colore della scena cosi come è, sfruttando una o più riprese della telecamera, unico strumento di analisi.

I primi hanno il vantaggio di raggiungere risoluzioni elevate, con precisione adatta anche ad applicazioni industriali, richiedendo però un costo notevole e permettendo un'applicabilità ristretta a determinati ambiti. I secondi, pur avendo prestazioni esponenzialmente inferiori, presentano una maggiore duttilità, ampia versatilità e applicabilità in diversi contesti, oltre ad un costo ridotto.

1.2 Metodi ottici

Pur non potendo ricreare fedelmente i processi alla base della ricostruzione tridimensionale del sistema visivo umano, la visione computazionale deve analizzare tutti gli aspetti che in un'immagine sono legati alla percezione della profondità di una scena: *sfocamento, parallasse(disparità), chiaroscuro, tessiture.* Tutte le tecniche computazionali, attive o passive che siano, devono fare riferimento a questi "indizi" e ad altri fenomeni ottici, nel processo di modellazione. Tra la famiglia dei metodi ottici passivi abbiamo, ad esempio:

- depth from focus/defocus
- shape from texture
- shape from shading
- stereo fotometrico
- stereopsi

- shape from silhouette
- shape from photo-consistency

Tra quella dei metodi ottici attivi invece:

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

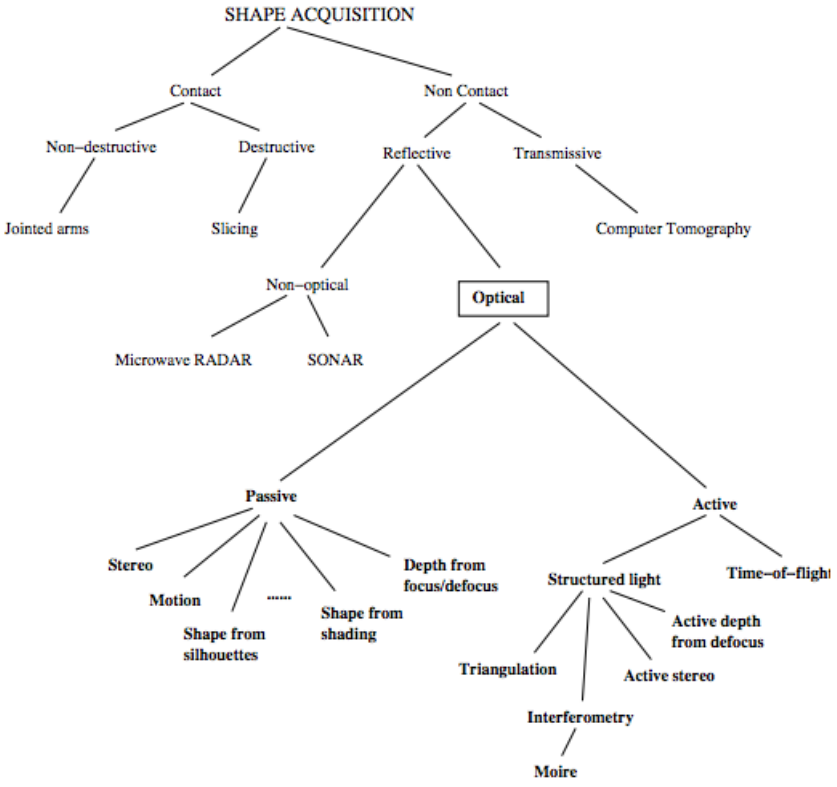

FIGURA 1.1: Tassonomia dei sistemi di acquisizione.

1.3 Immagini Range

Molti dei dispositivi ottici per l'acquisizione della superficie 3D di un oggetto o di una scena restituiscono un'*immagine range*, cioè un'immagine nella quale a ciascuna coordinata bidimensionale *(x,y)* è associata la relativa distanza dal sensore al primo punto visibile della scena.

FIGURA 1.2: Esempio di immagine range.

Un'immagine range è costituita da misure discrete di una superficie 3D, rispetto ad un piano di riferimento (il piano della telecamera). Questa di solito viene anche chiamata immagine 2.5D. Un sensore range è un dispositivo (apparecchiature e software) che produce per l'appunto un'immagine range. La qualità di tale immagine si misura secondo i seguenti parametri:

- **risoluzione**: la più piccola variazione di profondità alla quale è sensibile il sensore
- **accuratezza**: errore tra valore rilevato e valore esatto
- **precisione**: deviazione standard su una serie di misure ripetute sullo stesso valore
- **velocità**: misure ottenibili nell'unità di tempo

Tra i tipi di sensori più utilizzati possiamo citarne alcuni:

(i) **Laser Scanner 3D**: I sistemi di misura basati sul laser sfruttano una lama di luce (Laser) per digitalizzare le parti. Attualmente, gli scanner laser più utilizzati sono di due tipologie distinte:

a) Scanner manuali

b) Scanner laser fissi

Con gli scanner laser manuali, attraverso una movimento manuale è possibile acquisire i dati spazzolando la parte su e giù oppure svolgendo un movimento da destra verso sinistra. Gli scanner fissi montati su un cavalletto sono in linea di massima più precisi degli scanner laser manuali e offrono una migliore accuratezza. Entrambi i sistemi possono far uso di targets in modo da allineare progressivamente le varie misure;

(ii) **Luce strutturata**: Gli Scanner a luce strutturata sono composti generalmente da una testa ottica montata su un treppiede. Alcune applicazioni particolari prevedono il montaggio della testa ottica su un robot antropomorfo con diversi gradi di libertà. Questo dispositivo di misura 3D genera, attraverso l'utilizzo di un proiettore, un pattern che viene proiettato direttamente sulla superficie da digitalizzare. Nell'arco di qualche secondo le dimensioni del pattern (o frangia) vengono ridotte in larghezza; attraverso questa operazione il software è in grado di stabilire ed estrarre le coordinate 3D dei punti in modo da restituire velocemente una nuvola di punti;

(iii) **Time of Flight**: Una camera Time of Flight (ToF Camera) è un sistema per l'acquisizione di immagini range che cerca di risolvere le distanze basandosi sulla conoscenza della velocità alla quale viaggia la luce, misurando con accuratezza il tempo di volo tra la telecamera ed il soggetto di un segnale luminoso, per ciascun punto dell'immagine. Le camere ToF rappresentano una classe nell'insieme dei sistemi LIDAR (Laser Imaging Detection and Ranging) che si contraddistinguono per l'assenza di scansione. Questo tipo di sistemi ha cominciato a diffondersi solamente nell'ultimo decennio[1], con l'aumentare della velocità disponibile in alcuni dispositivi attraverso la tecnologia a semiconduttore. Le telecamere ricoprono distanze che vanno da alcuni metri fino a profondità massime di 60 metri. La risoluzione è di circa 1 cm. La risoluzione laterale risulta sicuramente bassa se comparata con telecamere 2D standard, arrivando attualmente a risoluzioni massime di 320x240 pixel[2] [3] . Comparate però con i metodi di laser scanning, le camere ToF operano ad alta velocità, restituendo sino a 100 immagini al secondo.

1.3.1 Comparazione tra sensori Laser e Luce strutturata

Spesso si è soliti confondere i due tipi di sensori. Può essere utile fare un breve confronto tra i due in modo da capire pregi e difetti di ognuno:

`Tecnologia:` gli scanner che utilizzano il laser campionano la parte da misurare utilizzando generalmente una sola lama di luce 3D mentre gli scanner a luce strutturata campionano l'oggetto proiettando progressivamente una serie di frange con larghezze diverse. A causa della ripetibilità delle letture (o campionamenti) è dimostrato che uno scanner a luce strutturata offre una qualità migliore rispetto a un laser.

`Velocità:` rispetto agli scanner a luce strutturata gli scanner laser hanno avuto per diverso tempo un potenziale vantaggio in termini di velocità in quanto era possibile ottenere con un unico movimento una lettura più veloce. Grazie però all'introduzione di una nuova serie di telecamere, un'elettronica rinnovata e a processori sempre più potenti la luce strutturata è in grado di offrire tempi di misura nell'ordine del secondo producendo nuvole dell'ordine di un milione di punti. Gli scanner a luce strutturata diventano un ottimo strumento per acquisire volti umani e per effettuare attività di body scanning recuperando, tra l'altro, anche le informazioni relative al colore.

`Area di scansione:` gli scanner laser effettuano generalmente la misurazione dividendo la lama di luce in una serie di punti disposti nello spazio. L'acquisizione dei punti è bidimensionale e si ottiene per effetto dello stiramento della lama proiettata sulla parte da digitalizzare. Gli scanner a luce strutturata sono invece in grado di acquisire i punti ordinati direttamente in 3D producendo quindi nuvole di punti intrinsecamente migliori dei sistemi basati sull'acquisizione tramite scansione laser.

`Condizioni di illuminazione:` gli scanner laser hanno la capacità di alzare il guadagno (gain) per ottenere le informazioni anche in quegli ambienti le cui condizioni di illuminazione sono precarie (illuminazione diffusa per esempio). I dati sono in genere rumorosi e spesso imprecisi. Gli scanner a luce strutturata richiedono che le condizioni di illuminazione ambientale siano controllate in quanto la lettura dei dati è determinata dalle prestazioni del proiettore. Scansioni in ambienti all'aperto e con luce diurna producono di

norma scarsi risultati e il laser è sicuramente la scelta migliore. Se le acquisizioni vengono invece effettuate in uffici e/o locali chiusi in cui è possibile controllare le sorgenti luminose la qualità delle misurazioni ottenute con la luce strutturata sono certamente migliori in termini di qualità superficiale e accuratezza rispetto alla tecnologia al laser.

`Sicurezza:` i Laser, grazie alla loro capacità di concentrare l'intensità della luce ed energia in uno spazio molto piccolo, presentano diversi problemi legati alla sicurezza, in particolare in tutte quelle circostante in cui il raggio laser entra in contatto con l'occhio. I sistemi di scansione al laser devono essere certificati secondo rigide normative e comunque rimanere al di sotto della Classe 2D per essere destinati all'acquisizione di corpi umani, visi, piedi e così via. La luce strutturata è basata semplicemente sulla luce bianca o blu (recentemente sono stati proposti anche algoritmi con luce strutturata con frequenze diverse, ma sempre nel range della luce visibile (380-760 nm))pertanto pienamente compatibile per effettuare misurazioni anche direttamente sul volto di una persona. In sintesi, ogni tecnologia ha i suoi pro e contro. Individuare il tipo di tecnologia più adatto dipende in gran parte delle esigenze del progetto (es. oggetto da digitalizzare, ambiente, colore del materiale, precisione, risoluzione).

1.4 Elaborazione dei dati

Il processo di acquisizione del volume non si esaurisce però con la semplice acquisizione della profondità anche se ne costituisce il passo fondamentale: per un modello completo infatti vanno acquisite più immagini range, da diverse prospettive. I dati poi vanno allineati, corretti e fusi tra di loro, ottenendo così una maglia poligonale. Possiamo dividere il processo in 3 fasi:

(i) **registrazione**: l'allineamento che viene fatto per trasformare le varie misure in misure con un unico sistema di riferimento comune;

(ii) **fusione geometrica**: volta ad ottenere un'unica superficie tridimensionale dalle varie immagini range;

(iii) **semplificazione della maglia**: i punti ridondanti vengono eliminati, viene semplificato il sistema in modo da renderlo più maneggevole.

Un'immagine range *Z(X,Y)* dunque definisce un insieme di punti 3D del tipo *(X,Y,Z(X,Y))*. Per ottenere una superficie nello spazio quindi, basta connettere tra di loro i punti ottenuti: il più delle volte questo viene ottenuto attraverso l'uso di facce triangolari. Va fatta particolare attenzione all'esistenza di discontinuità (buchi, bordi occludenti) che non si vuole vengano coperti da triangoli. A questo scopo si evita di connettere punti tra loro troppo distanti, nel qual caso si otterebbero facce con lati eccessivamente lunghi o angoli troppo acuti.

1.4.1 Registrazione

Utilizzando i sensori range è possibile acquisire delle mappe range di un oggetto da diverse angolazioni, ottenendo così immagini differenti della superficie. A ciascuna immagine è legato il proprio sistema di riferimento rispetto alla posizione planare del sensore. Allo scopo di unificare sotto un unico sistema di riferimento le diverse immagini range interviene proprio la registrazione, attraverso opportune rotazioni o traslazioni tridimensionali (idealmente trasformazioni rigide). Finchè i diversi sistemi di riferimento sono noti, il calcolo risulta banale. Nel caso invece che i nostri sistemi di riferimento siano incognite del problema, è necessario calcolare le opportune trasformazioni a partire unicamente dai dati stessi del problema (i punti ottenuti): l'algoritmo adatto a questa finalità si chiama *ICP: Iterated Closest Point*.

1.4.2 Fusione geometrica

Una volta compiuto il processo di registrazione, il passo successivo consiste nella fusione geometrica: la fusione di tutti i dati in un'unica forma, quale, ad esempio, una maglia triangolare. La superficie si ricostruisce, sebbene non sia nota a priori nessuna informazione di connettività. I metodi di fusione geometrica possono essere suddivisi come segue:

Integrazione di maglie: si uniscono le maglie triangolari appartenenti alle singole superfici range;

Fusione volumetrica: i dati sono fusi in una rappresentazione volumetrica, dalla quale poi si estrae una maglia triangolare.

1.4.3 Integrazione di maglie

Queste tecniche mirano ad unire maglie triangolari diverse e sovrapposte. Ad esempio la tecnica detta di *zippering* [4] erode le superfici sovrapposte fino ad eliminare la ridondanza, e usa una triangolazione bidimensionale per ricucire gli eventuali bordi. In tale processo non vi è perdita di accuratezza nei dati iniziali. Allo stesso tempo però gli errori presenti in fase di misura si propagano anche al modello 3D.

1.4.4 Fusione volumetrica

Questo metodo costruisce una superficie implicita intermedia che unisce le misurazioni sovrapposte in un'unica misurazione. Diversi algoritmi poi sono stati studiati per passare da questa rappresentazione ad una maglia triangolare. In questo caso può esserci perdita di accuratezza nei dati, come ad esempio nei dettagli della superficie. Inoltre lo spazio richiesto per la rappresentazione volumetrica cresce rapidamente al crescere della risoluzione.

1.5 Formazione dell'immagine

Questo paragrafo e i successivi vogliono trattare in prima analisi i processi alla base della formazione e la cattura dell'immagine. Risulta utile enunciare questi sia per completezza, sia per una miglior comprensione d'insieme quando successivamente verranno affrontati argomenti nei capitoli successivi che prevedono queste conoscenze base. Il modello geometrico elementare per la formazione di un'immagine all'interno di un modello è descritto dalla telecamera pinhole. In figura è presentato lo schema base:

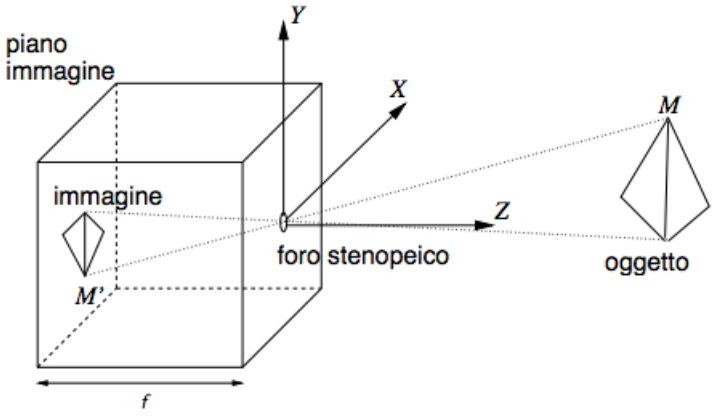

FIGURA 1.3: Telecamera pinhole.

Dato un punto M dell'oggetto, di coordinate *(X,Y,Z)*, sul piano della telecamera otteniamo la sua proiezione, di coordinate *(X',Y',Z')*. Indicata con f la distanza dal foro del piano focale, otteniamo le uguaglianze:

$$\frac{-X'}{f} = \frac{X}{Z}, \qquad \frac{-Y'}{f} = \frac{Y}{Z} \tag{1.1}$$

$$X' = \frac{-fX}{Z}, \qquad Y' = \frac{-fY}{Z}, \qquad Z' = -f \tag{1.2}$$

Il segno meno indica che l'immagine viene invertita rispetto all'originale. Questo processo di formazione dell'immagine è detto **proiezione prospettica**. La divisione per Z è invece responsabile del fenomeno per cui oggetti distanti appaiono più piccoli una volta proiettati.

1.5.1 Lenti sottili

Se invece di un foro viene usata una lente per l'acquisizione, la porzione di luce che può essere raccolta incrementa notevolmente. Per contro in questo caso non tutta l'immagine può essere contemporaneamente a fuoco. Anche nel caso di più lenti il sistema può essere approssimato come un'unica lente (lente sottile), in cui tutti i raggi convergenti convergono in un unico punto, detto fuoco: i raggi che incidono sulla lente vengono quindi deviati dalla lente stessa, fatta eccezione per i raggi incidenti sul centro della lente, che passano inalterati. La distanza tra centro della lente e fuoco della telecamera prende il nome di distanza focale, parametro fondamentale nel processo di acquisizione. Presa in considerazione la figura (1.4) otteniamo la formula dei punti coniugati, che vincola i punti:

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D} \tag{1.3}$$

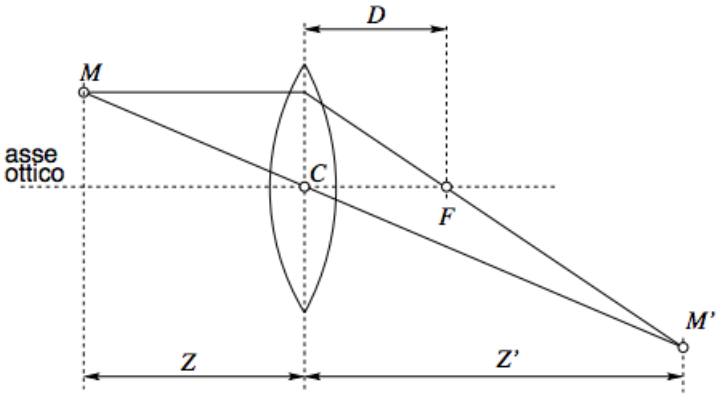

FIGURA 1.4: Lente sottile

Questo significa che un'immagine a distanza Z dalla lente, viene messa a fuoco ad una distanza Z', dipendente da Z: vi sarà quindi nella scena un'unico piano focale, a distanza Z, tutti i punti giacenti su un piano diverso, a distanza Z", saranno "fuori fuoco" e produrranno un cerchio invece che un punto, detto cerchio di confusione, dando la sensazione di punto sfocato. Finchè il cerchio prodotto non ha diametro superiore all'elemento fotosensibile questo risulta a fuoco.
Si parla dunque di profondità di campo: l'intervallo di distanze in cui gli oggetti

sono percepiti ancora a fuoco. Per mettere a fuoco oggetti a distanze diverse è sufficiente cambiare la distanza Z' (cosa che accade negli obiettivi delle fotocamere), oppure, in maniera più complessa, cambiare la forma della lente, come avviene nell'occhio umano. Una telecamera pinhole dunque ha teoreticamente una profondità di campo infinita, mentre gli obiettivi comunemente usati hanno una profondità di campo inversamente proporzionale al diametro della lente: per questo obiettivi più luminosi presentano una capacità di messa a fuoco limitata in profondità

1.5.2 Radiometria della formazione dell'immagine

Ci soffermiamo ora sulle caratteristiche che contraddistinguono un'immagine in relazione alla sua profondità. La luminosità *I(p)* di un pixel p dell'immagine è proporzionale alla quantità di luce che la superficie, centrata in un punto x, riflette in direzione dell'apparato di acquisizione. Questa a sua volta dipende sia dal modo in cui la superficie riflette la luce, sia dalla distribuzione e posizione delle fonti luminose che irradiano la scena.

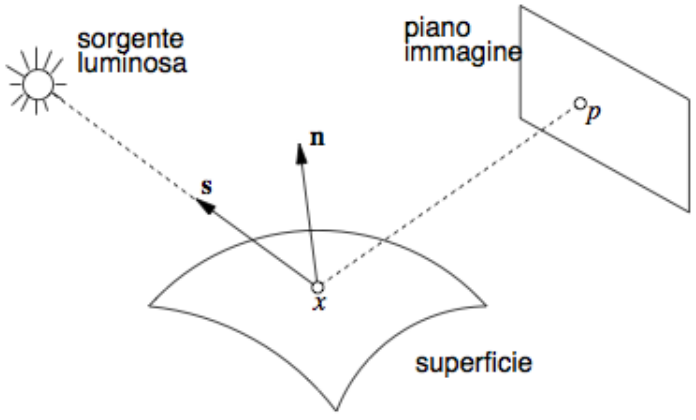

FIGURA 1.5: Radiometria della formazione dell'immagine

La quantità di luce che un punto emette o assorbe si misura formalmente mediante la radianza *L(x,ω)*: potenza di radiazione luminosa per unità di area per unità di angolo solido emessa dal punto x lungo la direzione ω. E si calcola secondo la seguente formula

$$L = \frac{d^2P}{dA\, d\Omega\, \cos\theta} \simeq \frac{P}{\Omega\, A\, \cos\theta} \tag{1.4}$$

Dove θ rappresenta l'angolo compreso tra la normale alla superficie e la direzione specificata, A è la superficie emittente, P è la potenza ed Ω è l'angolo solido. Parleremo di diffusione nel caso la luce venga riflessa in modo omogeneo verso ogni direzione, e di riflessione speculare nel caso la radianza riflessa sia concentrata lungo una determinata direzione.

1.6 Chiaroscuro, tessitura, sfocamento

1.6.1 Chiaroscuro

Dato un oggetto illuminato, attraverso lo *Shape from shading* [5] è possibile calcolarne la forma, sfruttando le informazioni connesse alle variazioni di luminosità della superficie. Se per semplicità analizziamo un immagine B/N con forme semplici, la distribuzione dei livelli di grigio reca con sé informazione utile riguardo a forma della superficie e direzione di illuminazione.

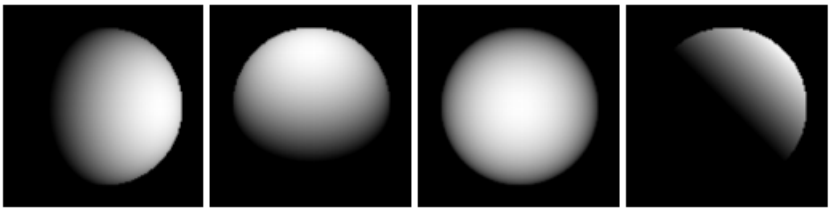

FIGURA 1.6: Sfere lambertiane in diverse condizioni di illuminazione

Seguendo questo metodo, punti della superficie con retta normale coincidente, avranno lo stesso valore di grigio, con ipotesi di sorgente puntiforme a distanza infinita (illuminazione parallela). Quella che si va a costruire dunque è una mappa di riflettanza. Ruolo cruciale giocano inoltre la determinazione della direzione di illuminazione e dell'albedo (la percentuale di luce riflessa da una superficie, in tutte le direzioni).

In maniera più complessa, per risolvere l'ambiguità circa la normale ad una superficie, si usano diverse direzioni di illuminazione: questo è detto stereo fotometrico e, a costo di una maggior numero di riprese con illuminazione differente, semplifica notevolmente il problema, intersecando le mappe di riflettanza.

1.6.2 Tessitura

Con tessitura si indica la particolarità di oggetti che presentano superfici ricorsive, con pattern ripetuti (un muro di mattoni, un prato d'erba, etc.). Gli elementi che si ripetono vengono detti *texel*, questi devono essere abbastanza piccoli da non poter esser considerati come oggetti distinti. Vi possono essere texture deterministiche (strutture artificiali) o statistiche (presenti in natura). Qualsiasi immagine che presenti delle ripetizioni geometriche oppure variazioni che seguono un andamento ricorsivo, quali ad esempio i quadrati di una scacchiera o di un tessuto, possono essere visti come texture. Non necessariamente dunque classifichiamo come texture una forma che presenti piccole variazioni di colore e di frequenza, ma, in modo più generale, qualsiasi andamento ricorsivo all'interno dell'immagine. I texel, seppur uguali, in una delle due forme, deterministica o statistica, presentano caratteristiche diverse nell'immagine acquisita, a causa della proiezione prospettica. Questo fatto può essere opportunamente utilizzato ai nostri scopi, una volta identificata la texture. Elementi distintivi di una texture sono dunque:

- *cambiamento della forma di ciascun texel, in rapporto all'inclinazione rispetto al piano di riferimento (ellitticita')*

- *cambiamento della dimensione apparente, in base alla distanza*

- *velocita'di cambiamento (gradiente di tessitura)*

Questi sono indizi che ci consentono di risalire all'inclinazione del piano ai quali fa riferimento la texture. Il concetto risulta più chiaro citando un'esempio di algoritmo per l'estrazione delle texture. Il metodo proposto da Leung and Malik [6] utilizza un banco di filtri su delle immagini texture di training per ogni materiale con illuminazione e angolo prospettico noto. Le risposte dei filtri vengono salvate in un cluster, che contiene così le proprietà peculiari del materiale, dette textons, creando così un dizionario dei materiali possibili: ogni materiale è dunque rappresentato da una particolare funzione di densità probabilistica, come ad esempio la distribuzione delle frequenze delle textons. Possiamo così associare a ciascun elemento dell'immagine un particolare materiale, come rappresentato in figura.

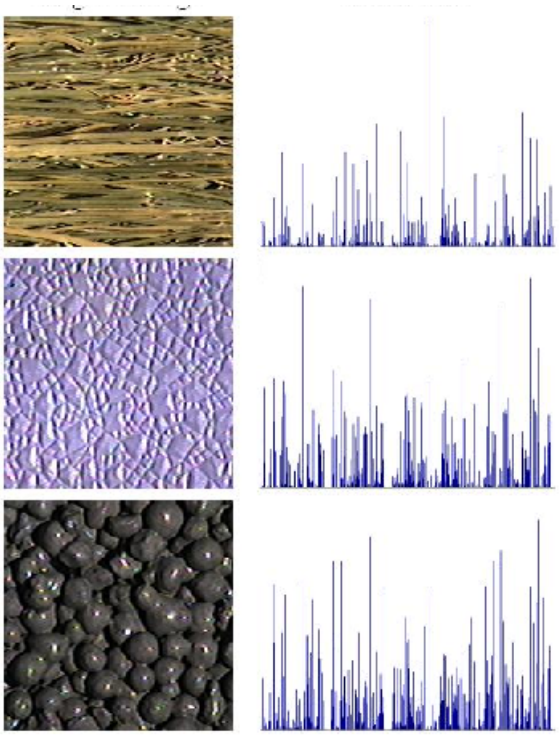

FIGURA 1.7: Texture e relative pdf delle texton

1.6.3 Messa a fuoco

Ultimo elemento che viene analizzato, utile alla deduzione della maschera 3D, è la messa a fuoco. In questo caso possiamo distinguere tra due diversi comportamenti: fochettatura e sfocamento.

1.6.3.1 Fochettatura

Il metodo della fochettatura, meglio conosciuto come *Depth from focus*, si basa sul miglioramento della messa a fuoco, in immagini successive, dello stesso soggetto: sapere che un punto risulta a fuoco ci fornisce informazioni sulla sua distanza, attraverso la legge delle lenti sottili. Questa metodologia risulta ovviamente molto lenta, e richiede una successione di immagini, oltre alla ricerca di un criterio ottimo con il quale cambiare la messa a fuoco, e con il quale trovare la posizione ottimale. Questo si ottiene con un calcolo sulle alte frequenze spaziali dell'immagine rilevata. Tale problema di calcolo si risolve con il metodo di bisezione o di Fibonacci, che, come noto, soffre di problemi di estremi locali, che possono invalidare il risultato.

1.6.3.2 Sfocamento

Nota come *Depth from defocus* si basa sull'analisi di almeno due immagini con diversi parametri di fuoco. Sfruttando la relazione diretta tra parametri di ottica, profondità e sfocamento, possiamo ricavare la profondità di tutti i punti. I problemi maggiori si hanno in fase di misura dello sfocamento stimato e di calibrazione della relazione depth/sfocamento.

1.7 Stereopsi

Tra le diverse tecniche di computer vision note in letteratura e mirate alla ricostruzione della struttura tridimensionale di una scena osservata da una o piu' telecamere (sd esempio shape from motion, shape from shading, shape from texture) la visione stereoscopica è quella che ha riscosso la maggiore attenzione principalmente perche non impone alcun vincolo sulle caratteristiche degli oggetti presenti nella scena (come presenza o meno di oggetti in movimento, presenza o meno di particolari condizioni di illuminazione). La visione stereoscopica consente di inferire la struttura tridimensionale di una scena osservata da due o piu' telecamere (nel caso di due telecamere si parla di visione binoculare). Il principio alla basa della visione stereoscopica, noto sin dal rinascimento, consiste in una triangolazione mirata a mettere in relazione la proiezione di un punto della scena sui due (o più) piani immagine delle telecamere (tali punti sono denominati punti omologhi) che compongono il sistema di visione stereoscopico. L'individuazione dei punti omologhi, problema noto in letteratura come il problema della corrispondenza (correspondence problem o matching stereo), consente di ottenere una grandezza denominata disparità (disparity) mediante la quale, conoscendo opportuni parametri del sistema stereoscopico, è possibile risalire alla posizione 3D del punto considerato. Il problema delle corrispondenze rimane ancora aperto e produce tuttora un'ampia attivita di ricerca nonostante siano stati proposti sin dagli anni '60 innumerevoli algoritmi.

La stereopsi dunque è la capacità percettiva che consente di unire le immagini provenienti da due occhi, che, grazie al loro diverso posizionamento strutturale, presentano uno scostamento laterale. Questa disparità viene sfruttata dal cervello umano per trarre informazioni sulla profondità e sulla posizione spaziale dell'oggetto di interesse. Di conseguenza la stereopsi permette di generare la visione

tridimensionale, oltre a diminuire la probabilità di occlusioni non visibili ad entrambi i punti di vista. Per quanto riguarda la visione computazionale invece si parla di stereopsi (computazionale) quando si sfruttano le informazioni provenienti da una coppia di immagini della stessa scena, che presentano la caratteristica di differenziarsi per posizione dalla quale la scena viene ripresa. Vi è dunque uno sfasamento angolare laterale di qualche grado tra le due immagini, in modo che esse,pur rappresentando la stessa scena, contengano dati sufficientemente diversi da poter esser fusi insieme. Un angolazione troppo ristretta porterebbe poca informazione, un'angolazione viceversa troppo enfatizzata farebbe si che le due immagini riprendano una struttura talmente diversa da non poter esser confrontata. La fusione avviene tramite il processo di **calcolo delle corrispondenze** e successiva **triangolazione**. Il calcolo delle corrispondenze prevede di indentificare nelle due immagini i punti comuni, lo stesso riferimento nella scena reale: tali punti vengono detti punti coinugati. Il calcolo dell'accoppiamento è possibile grazie al fatto che le due immagini differiscono solo lievemente, cosicché un particolare della scena appare simile nelle due proiezioni. Così facendo però si creano diversi falsi accoppiamenti. Introducendo nuovi vincoli, quali il vincolo epipolare, che vincola ogni corrispondenza a trovarsi su una determinata retta, per ciascun punto, si riesce ad evitare tale problema. Noti gli accoppiamenti, la posizione relativa delle telecamere, i parametri interni dei sensori, si ricava la posizione nella scena dei punti. Questo processo di triangolazione necessita della calibrazione dell'apparato stereo, cioè del calcolo dei parametri intrinseci (singoli) ed estriseci (duali) delle telecamere.

1.7.1 Geometria di un sistema stereoscopico

La trasformazione prospettica che mappa un punto dello spazio nel piano immagine di una telecamera implica la perdita dell'informazione relativa alla distanza. Questo può essere facilmente rilevato osservando la figura (1.8) nella quale due distinti punti (P e Q) nello spazio intersecati dallo stesso raggio passante per il centro ottico O_l della telecamera corrispondono allo stesso punto (pq) nel piano immagine.

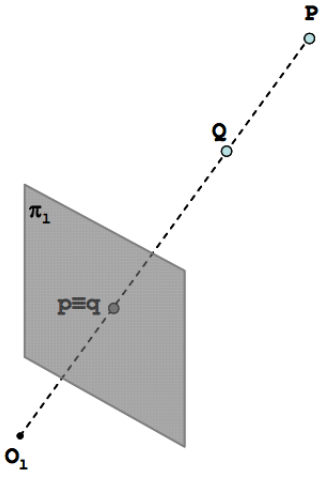

FIGURA 1.8: Proiezione di due distinti punti nello spazio sul piano immagine

Un metodo per poter risalire a quale punto dello spazio corrisponda la proiezione di un punto sul piano immagine di una telecamera, consiste nell'utilizzo di due o piu telecamere. Infatti, come mostrato nella figura seguente nel caso di un sistema composto da due telecamere, tra tutti punti nello spazio che giacciono sul raggio che passa per il centro ottico O_l e il punto q, proiezione di Q sul piano immagine π_l, al più un solo punto (*punto omologo*) viene proiettato (q') anche sul piano immagine π_r. La determinazione dei punti omologhi consente di mettere in relazione le proiezioni dello stesso punto sui due piani immagini e di risalire, mediante una procedure denominata triangolazione, alle coordinate dei punti dello spazio rispetto ad un sistema di riferimento opportuno. Sia dato un punto q su un piano immagine π_l, proiezione del punto Q appartenente allo spazio 3D: per ottenere, attraverso la triangolazione, le coordinate 3D del punto nello spazio è necessario determinare (problema delle corrispondenze) il punto omologo q' nel piano immagine π_l. Tale problema, dato il punto q nel piano immagine π_l, richiederebbe una ricerca bidimensionale del punto omologo q' all'interno del piano immagine π_r.

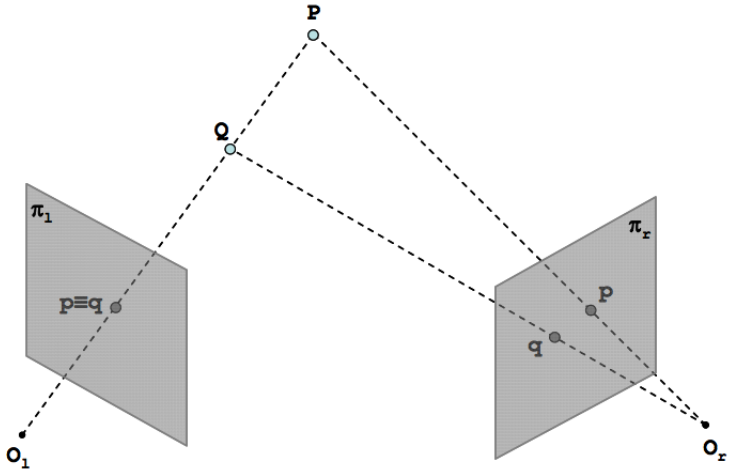

FIGURA 1.9: Principio alle base di un sistema stereoscopico

In realtà, sfruttando una particolare caratteristica della geometria del sistema stereoscopico, è possibile effettuare la ricerca del punto omologo in uno spazio monodimensionale. Infatti, come mostrato nella figura (1.9), gli omologhi di tutti i punti dello spazio che potrebbero risultare proiezione nello stesso punto q del piano immagine π_l, (ad esempio il punto p proiezione di P o lo stesso punto q proiezione di Q) giacciono sulla retta generata dall'intersezione tra il piano immagine π_r e il piano (denominato piano epipolare) che passa per la retta O_lQP e i due centri ottici O_l e O_r. Tale vincolo, denominato vincolo epipolare, consente di limitare lo spazio di ricerca dei punti omologhi ad un segmento di retta semplificando considerevolmente il problema delle corrispondenze sia da un punto di vista della complessità algoritmica sia per quanto concerne la correttezza della soluzione.

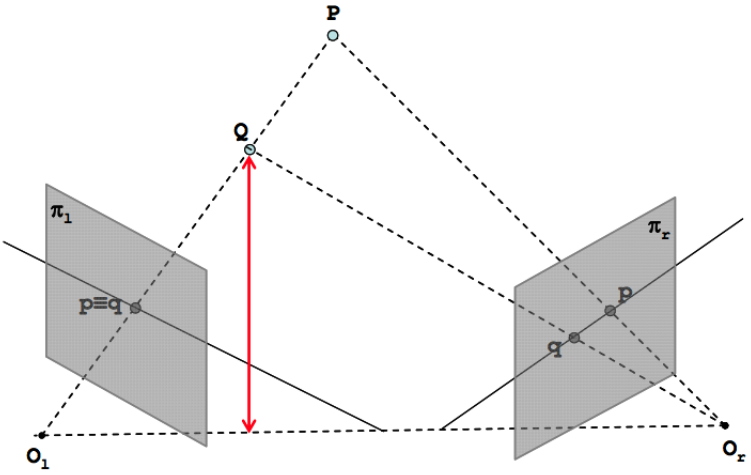

FIGURA 1.10: Vincolo epipolare

Si fa notare che il problema delle corrispondenze non necessariamente ha soluzione: infatti, a causa della diversa posizione delle telecamere che compongono un sistema di visione stereoscopico nello spazio, è possibile che un punto non risulti proiettato su tutti i piani immagini delle telecamere. In tal caso il problema delle corrispondenze non ha soluzione e non è possibile determinare la distanza del punto esaminato dalle telecamere (occlusioni). Un sistema di visione stereoscopico è completamente caratterizzato mediante i parametri intrinseci ed estrinseci. I parametri intrinseci consentono di definire la trasformazione che mappa un punto dello spazio 3D nelle coordinate del piano immagine di ogni telecamera e risultano le coordinate relative al piano immagine del punto principale (punto di intersezione tra il piano immagine e la retta ortogonale al piano immagine stesso passante per il centro ottico), la distanza focale, ed eventualmente altri parametri che descrivono altre caratteristiche del sensore (distorsione delle lenti, forma dei pixels, etc). I parametri estrinseci invece rappresentano le posizioni di ogni telecamera rispetto ad una sistema di riferimento noto. La determinazione dei parametri intrinseci ed estrinseci, ottenuta mediante la procedura di calibrazione, consente quindi di descrivere completamente il sistema stereoscopico ed in particolare di inferire informazioni relative alle coordinate dei punti nello spazio mediante la triangolazione di punti omologhi. La conoscenza dei parametri intrinseci ed estrinseci consente anche di trasformare le immagini acquisite dal sistema stereoscopico al fine di produrre un sistema virtuale nel quale i piani immagine delle telecamere giacciono sullo stesso piano e nel quale la ricerca dei punti omologhi avviene esaminando le medesime righe nei diversi piani immagine. Tale configurazione del sistema stereoscopico ottenuta mediante una procedura denominata rettificazione, ed è mostrata in figura (1.11).

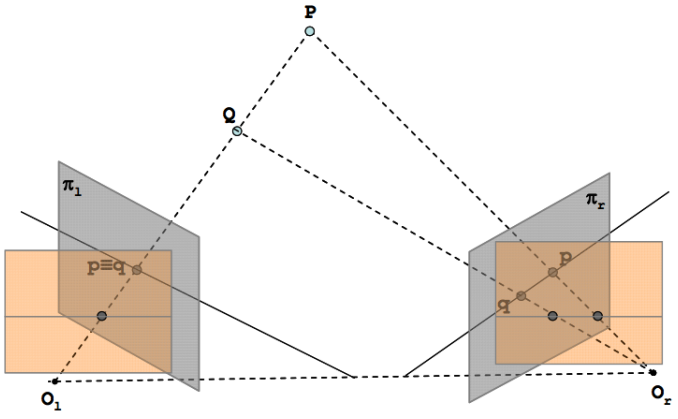

FIGURA 1.11: Immagini in forma standard

Osservando lo schema si può notare come il sistema risultante dopo la rettificazione sia composto da due piani immagine virtuali θ_l e θ_r giacenti sullo stesso piano. Le immagini stereoscopiche ottenute da una sistema rettificato sono denominate immagini in forma standard. Si osserva infine che nelle immagini in forma standard i piani xy dei sistemi di riferimento centrati nei centri ottici delle due telecamere sono coplanari.

1.7.2 Calibrazione

La calibrazione è una procedura eseguita offline ed è mirata all'individuazione dei parametri che caratterizzano un sistema di visione stereoscopico. Tali parametri, denominati parametri intrinseci ed estrinseci, sono utilizzati dalla procedura di rettificazione per trasformare le immagini acquisite dal sistema stereoscopico in modo da ottenere immagini stereoscopiche in una forma particolare (forma standard) e per ottenere le coordinate 3D dei punti mediante la procedura di triangolazione. Esistono in letteratura diverse tecniche per effettuare la calibrazione di un sistema stereoscopico (per maggiori dettagli si consideri [7]). Tipicamente però questa operazione viene eseguita con tecniche basate sull'utilizzo di pattern geometrici dei quali sono note con precisione le caratteristiche (dimensione e posizione delle features presenti nel pattern, etc). Mediante l'acquisizione di tali pattern (generalmente contenenti features simili a scacchiere) in diverse posizioni e utilizzando procedure ampiamente note [8], è possibile stimare i parametri intrinseci ed estrinseci che caratterizzano il sistema stereoscopico. Nella figura seguente sono mostrate una serie di immagini utilizzate per la calibrazione di un sistema stereoscopico mediante l'utilizzo di un pattern planare (scacchiera).

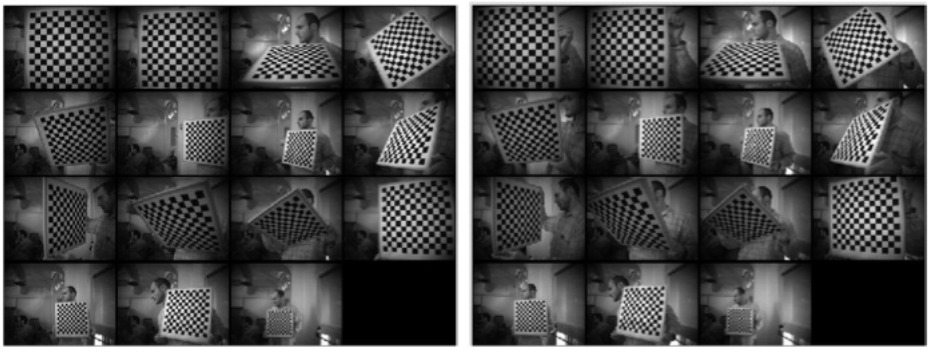

FIGURA 1.12: Calibrazione mediante l'acquisizione di un pattern noto in 15 posizioni diverse

1.7.3 Acquisizione

In un sistema stereoscopico mirato ad ottenere informazioni tridimensionali di punti in movimento è necessario che l'acquisizione delle due immagini che costituiscono l'immagine stereo (stereo pair) sia simultanea. Questo vincolo può essere eliminato nel caso di scene statiche che però rivestono un limitato interesse pratico. L'acquisizione simultanea di più immagini provenienti da una sorgente video analogica o digitale può avvenire mediante diverse tecnologie. Nel caso di segnali video anologici si utilizzano frame-grabbers capaci di acquisizioni contemporanee di più segnali video o frame-grabbers che acquisiscono una sola sorgente video analogica nella quale le due o più immagini sono state preventivamente interallacciate (interlaced) via hardware. Quast'ultima soluzione è facilmente realizzabile ed esistono soluzioni commerciali che consistono in un sistema integrato composto da telecamere e sistema che interlaccia le immagini. Utilizzando sorgenti video analogiche risulta che le dimensioni delle immagini acquisite sono limitate dalle specifiche del segnale video analogico medesimo. Per cui, nel caso di sistemi binoculari, utilizzando la codifica NTSC (PAL) la massima risoluzione di ogni immagine della coppia stereo per acquisizioni multiple è pari a 640x240 pixels (768x288 pixels PAL) mentre nel caso di segnale interallacciato è pari a 640x120 pixels (768x144 pixels nel caso PAL). In entrambi i casi la massima frequenza di acquisizione è di 60 frame per secondo (fps) nel caso NTSC e 50 fps nel caso PAL. Un vantaggio del segnale video analogico è dovuto al fatto che le telecamere e il dispositivo di acquisizione possono essere poste a notevole distanze (dell'ordine di centinaia di metri) utilizzando dei semplici ed economici amplificatori di segnale. Nel caso di acquisizione da una sorgente di segnale digitale i limiti alla risoluzione delle immagini stereo acquisite sono legati alle risoluzioni delle telecamere e dall'ampiezza di banda del canale digitale di comunicazione digitale che tipicamente è di tipo seriale USB 2.0 (480 Mbps) e firewire (400 Mbps), noto anche come IEEE 1394a o i-Link. Attualmente esistono in commercio sistemi commerciali ([9] [10]) che integrano le telecamere e il modulo firewire. Nel caso di sistemi binoculari e di utilizzo del protocollo firewire è possibile acquisire stereo-pair di dimensioni 640x480 pixels a 30 fps e di dimensione 1280x960 a 7.5 fps [11]. Mediante tecnologie USB 2.0 e firewire possiamo realizzare un sistema di acquisizione stereo utilizzando delle normali telecamere dotate di ingresso di trigger esterno mediante il quale è possibile sincronizzare l'acquisizione inviando un semplice segnale di sincronizzazione. Allo stato attuale i limiti maggiori della tecnologia firewire (IEEE

1394a) risultano nelle distanze massime tra le telecamere e il computer che acquisisce le immagini, limitate ad alcuni metri. Si fa notare che recentemente è stato definito un nuovo standard IEEE1394b che consente di disporre di una banda massima pari a 800 Mbps in grado di coprire distanze di centinaia di metri utilizzando cavi in fibra ottica. Lo standard 1394b è compatibile con lo standard 1394a e in presenza di un dispositivo a 400 Mbps tutti i dispositivi IEEE1394b limitano la banda a tale velocità. Non esistono attualmente sistemi stereo integrati dotati di interfaccia IEEE1394b ma cominciano ad apparire sul mercato telecamere dotate di tale interfaccia.

1.7.4 Rettificazione

La rettificazione è una procedura, che sfruttando i parametri intrinseci ed estrinseci ottenuti mediante calibrazione, mira a trasformare le immagini stereoscopiche provenienti dal dispositivo di acquisizione in modo che risultino soddisfatti alcuni vincoli. Tra questi, la rettificazione consente di trasformare le immagini in forma standard: nel caso di sistemi binoculari questo assicura che dato un punto in un'immagine il suo omologo possa essere rintracciato sulla stessa riga dell'altra immagine consentendo una notevole riduzione dei calcoli ed una maggiore affidabilità nella soluzione del problema delle corrispondenze.

FIGURA 1.13: Immagini prima e dopo la procedura di rettificazione

Mediante la rettificazione è possibile trasformare le immagini in modo che i punti omologhi di una riga (scanline) dell'immagine possano essere ricercati nella corrispondente riga dell'altra immagine. La rettificazione consente anche di ridurre i problemi legati alla distorsione provocata dalle ottiche, e di ottenere immagini con la stessa distanza focale. A titolo di esempio, si mostra nelle immagini in figura (1.13) l'effetto della rettificazione di immagini acquisite da un sistema stereoscopico alla risoluzione di 640x480 pixel. Nelle immagini superiori sono visualizzate le immagini acquisite dal sistema stereo, mentre in quelle inferiori è mostrato il risultato della rettificazione ottenuta mediante i parametri stimati con la procedura di calibrazione.

1.7.5 Triangolazione 3D

Al fine di capire il problema della triangolazione è utile partire dal caso basilare, considerando due telecamere parallele ed allineate, in modo da riportarci al caso bidimensionale come in figura (1.14)

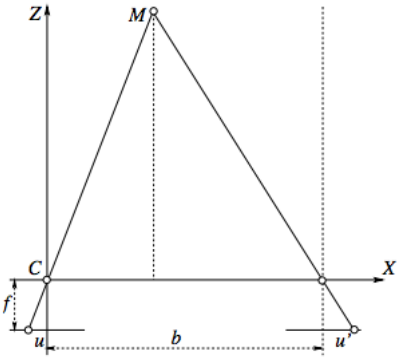

FIGURA 1.14: Triangolazione stereoscopica

Possiamo scrivere le seguenti equazioni per il sistema di riferimento:

$$\frac{f}{z} = \frac{-u}{y} \tag{1.5}$$

$$\frac{f}{z} = \frac{-u'}{y-b} \tag{1.6}$$

Dalle quali otteniamo:

$$z = \frac{bf}{u'-u} \tag{1.7}$$

Questo ci dice quindi che nota la geometria del sistema di riferimento (b ed f) e la disparità (u-u') siamo in grado di ricavare la profondità z.

1.7.6 Calcolo delle corrispondenze

Massima importanza va data infine al calcolo delle corrispondenze (disparità). Diamo la definizione di:

Coppia coniugata : costituita da due punti in due immagini diverse che sono proiezione dello stesso punto della scena reale

Disparità : differenza vettoriale tra due punti coniugati, sovrapponendo le due immagini

Il calcolo delle corrispondenze equivale al calcolo delle disparità per i punti dell'immagine riferimento. Otteniamo una mappa di disparità come in figura (1.15):

FIGURA 1.15: Mappa di disparita'

La stereoscopia computazionale ha bisogno di due immagini, come già detto, diverse, al fine di ricevere informazione aggiuntiva, ma allo stesso tempo abbastanza simili, in modo da poter mettere in corrispondenza le informazioni, allineando su un riferimento comune le due immagini. Vi devono dunque essere un numero sufficiente di coppie di punti corrispondenti tra un frame e l'altro. Oltre a questo abbiamo il problema delle false corrispondenze: casi in cui non è possibile trovare la coppia di punti, principalmente a causa di:

occlusioni: vi sono punti che sono visibili da solo una delle due immagini, ovviamente risulta impossibile trovare la disparità;

distorsione radiometrica: differenza di radianza osservata dalle due telecamere;

distorsione prospettica: a causa della distorsione prospettica un oggetto assume forme diverse da diversi punti di vista, proiettandosi differentemente.

Vengono usati dei vincoli per agevolare la soluzione delle corrispondenze:

somiglianza: un oggetto appare simile nelle due immagini;

geometria epipolare: il punto coniugato giace su una retta detta epipolare determinata attraverso i parametri intriseci ed estrinseci;

lisciezza: lontano dai bordi, la profondità dei punti di una superficie liscia varia lentamente. Vi è un vincolo sul gradiente;

unicità: ad ogni punto corrisponde biettivamente uno ed un solo punto;

ordinamento monotono: fatta eccezione per casi particolari, due punti nell'immagine A mantengono l'ordine nell'immagine B.

Come metodo locale per il calcolo della disparità va menzionato l'accoppiamento tra finestre: si considera una piccola area di una delle due immagini, e in base ai livelli di grigio, o di una funzione basata su questi, si cerca la corrispondente finestra nell'altra immagine stereoscopica. I metodi di questa classe sono basati ad esempio sulla correlazione o sulle differenze di intensità. Particolare attenzione va posta nella scelta della dimensione della finestra: se la finestra di correlazione copre una regione in cui la profondità varia, la disparità sarà inevitabilmente affetta da

errore. D'altro canto, se la finestra è troppo piccola, il rapporto SNR sarà basso e la disparità ottenuta è poco affidabile. Per sopperire a questo inconveniente sono state proposte soluzioni che fanno uso di finestre adattative, o scelte tra diverse finestre di area fissa, ma traslate, scegliendo quella che presenta meno variazione ipotetica di disparità

1.8 Occlusioni

Le occlusioni si hanno quando un punto non è visibile in entrambe le immagini, creando così una corrispondenza mancante. Innanzitutto per gestire questo fenomeno è necessario rilevare le occlusioni, evitando così false corrispondenze. Indispensabile risulta il vincolo di unicità: supponendo che l'algoritmo di accoppiamento dei punti operi correttamente, dette I_1 e I_2 le due immagini, e preso un punto $p \in I_1$, che risulta occluso in I_2, questo genererà una falsa corrispondenza, trovando un determinato pixel p' in I_2. Tuttavia, p' sarà anche la soluzione corrispondente applicando l'algoritmo al punto $p_{real} \in I_1$. In tal modo viene violato il vincolo di unicità avendo due punti di I_1 che hanno come soluzione p'. L'incertezza si risolve applicando l'algoritmo nel verso opposto, da I_2 ad I_1. Trovando un'unica soluzione per p', data dal punto p_{real}. Fatto questo p può essere etichettato come punto con corrispondente occluso.

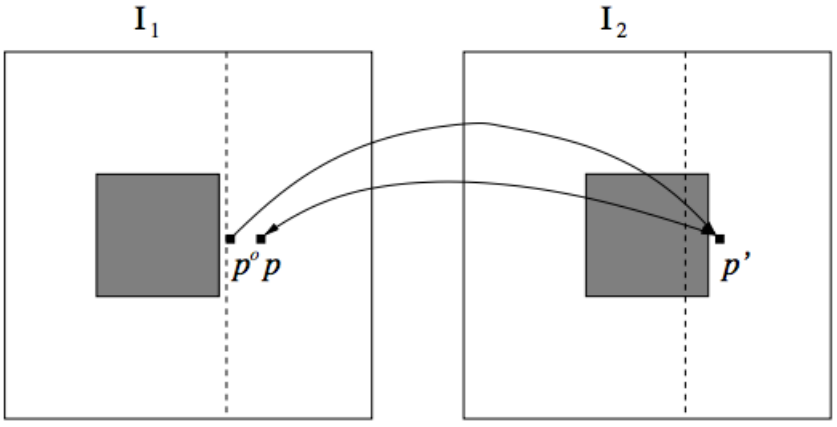

FIGURA 1.16: Verifica della coerenza in presenza di occlusioni

Sempre nella classe di metodi che sfruttano le proprietà a livello locale delle foto vi sono quelli basati sul gradiente, quelli sulla segmentazione (che ampiamente verrà discussa nei capitoli successivi) e quelli basati su alcune feature dell'immagine,

quali edge(spigoli), angoli e rette, possibilmente stabili rispetto ad un cambio di prospettiva dell'immagine.

1.9 Metodi di accoppiamento globali

Oltre alle caratteristiche locali dell'immagine che abbiamo brevemente visto, vi sono anche delle caratteristiche globali, che possono essere utilizzate efficacemente per l'accoppiamento. Queste risultano particolarmente utili in zone locali dove l'accoppiamento, attraverso i metodi già visti, risulta poco efficace. Si può far riferimento alla cosidetta *Disparity Space Image* (DSI), un'immagine tridimensionale che contiene in terza dimensione i valori della metrica di accoppiamento, che va studiata attraverso una funzione costo, sia a livello bidimensionale, che, come spesso utilizzato, per scan line, ovvero il costo lungo una determinata riga o colonna, ove le discontinuità possono trovarsi solo in corrispondenza delle occlusioni o dei salti di discontinuità

1.10 Stereo Attivo

La situazione in cui ci siamo posti sinora è detta stereo passivo, in quanto l'analisi opera sulla scena originale, senza introdurre nuovi elementi. Nel campo della modellazione 3D però spesso si utilizzano fonti luminose esterne, in modo da facilitare tutti i processi visti finora. Se siamo interessati ad ottenere il miglior risultato possibile, o se l'applicazione che pensiamo ha applicazioni industriali, cercando una scannerizzazione degli oggetti della scena, non possiamo fare a meno di ricorrere allo stereo attivo. Il tipo di illuminazione usata viene definita illuminazione strutturata, ed è di vari tipi:

Tessitura artificiale: viene proiettata sulla scena un'insieme di punti luminosi, creando così una tessitura detta "sale e pepe" che va ad agevolare la valutazione delle corrispondenze

Raggio laser: la scena viene scannerizzata da un raggio laser che proietta un punto, che viene messo facilmente in corrispondenza nelle due immagini. Così facendo il processo va però reiterato per ogni punto della scena, aumentando notevolmente il peso computazionale e di memoria.

Lama di luce: vengono proiettati segmenti (generalmente orizzontali) di luce laser, l'intersezione delle linee curve proiettate con le rette epipolari fornisce i punti corrispondenti. Sebbene più leggero del sistema precedente, richiede comunque di riprendere molte coppie di immagini. In questo caso il laser può svolgere il ruolo anche di punto di triangolazione, rendendo non più indispensabile una seconda telecamera per la triangolazione.

Luce codificata: Per rendere ancora più snello e veloce il sistema è possibile, attraverso un proiettore, utilizzare la proiezione di più piani nello stesso momento, codificando in qualche modo le bande proiettate.

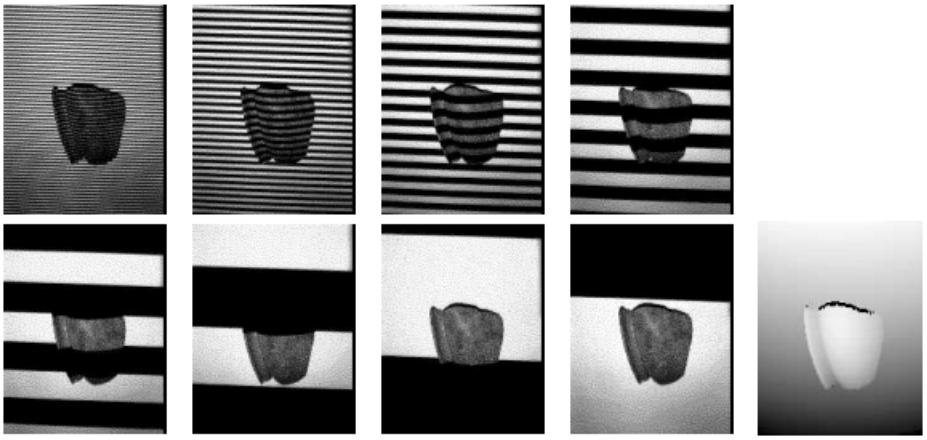

FIGURA 1.17: Luce codificata

1.10.1 Triangolazione con luce strutturata e singola camera

Nel caso si voglia triangolare l'immagine con una singola camera, sfruttando le informazioni della luce strutturata proiettata sulla scena, si procede in maniera molto simile al caso di due camere:

Detto M un punto della scena visto dalla camera, di coordinate *(x,y,z)*, attraverso una trasformazione rigida possiamo portare questo punto sul piano del proiettore, *(R,t)*. le coordinate del punto per il proiettore saranno: $M_p = RM_c + t$. La proiezione del punto M sul piano della telecamera invece, in coordinate normalizzate risulta: $p_c = [u_c, v_c, 1]^T$

Ottenendo:

$$p_c = \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix} = \frac{1}{z_c} M_c \tag{1.8}$$

Il proiettore invece viene modellato come una telecamera, con coordinata verticale del punto proiettato incognita (bande verticali). Sia u_p la coordinata del piano che illumina M, allora M si proietta sul punto $p_p = \frac{1}{z_p} M_p$ ottenendo attraverso le equazioni di prima:

$$z_p p_p - z_c R p_c = t \tag{1.9}$$

in forma matriciale, scomponibile in 3 equazioni.

Dopo qualche passaggio, otteniamo che la profondità del punto M nel piano della camera risulta:

$$z_c = \frac{t_1 - t_3 u_p}{(u_p r_3^T - r_1^T) p_c} \tag{1.10}$$

Capitolo 2

Conversione da 2D a 3D: algoritmi ed ambiti di utilizzo

2.1 Introduzione

Il sistema visivo umano ha un'innata capacità nell'interpretare la struttura tridimensionale degli oggetti partendo semplicemente dalle linee prospettiche rilevabili analizzando un'immagine bidimensionale e le caratteristiche globali e locali che la contraddistinguono, come ad esempio le relazioni tra elementi diversi di una stessa immagine, la luminosità oppure il gradiente di colore. Gli algoritmi di visione computazionale cercano di riprodurre in modo artificiale queste capacità di deduzione di profondità dalla singola immagine. Se l'informazione in un'immagine bidimensionale è rappresentata dall'insieme dei pixel, i punti dell'immagine, nel caso tridimensionale va introdotto il concetto di Voxel. Un voxel (volumetric pixel) è un punto nello spazio tridimensionale al quale è associata un'informazione di colore e di intensità luminosa (analogamente al pixel). Partendo dunque dai pixel l'obiettivo è quello di stimare i voxel relativi, appartenenti allo stesso raggio, come illustrato in figura (2.1).

Nell'ambito della visione computazionale, la ricostruzione tridimensionale di una scena a partire da una singola immagine bidimensionale rimane un problema ancora aperto, al quale sono state presentate diverse soluzioni. Di queste nessuna risulta ottimale, dato che il problema risulta mal formulato: la completa ricostruzione infatti è impossibile, dato che una sola immagine bidimensionale è priva delle informazioni di profondità presenti nella scena reale.

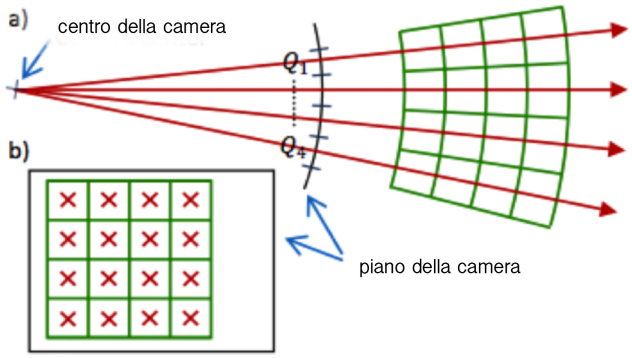

FIGURA 2.1: Definizione di pixel, voxel, raggio, e relazione geometrica tra questi elementi in un piano normale(a) o parallelo(b) al piano della camera. I voxel(in verde) sono allineati ai pixel cosicchè le loro proiezioni coincidono perfettamente, e sono delimitati agli altri due lati da linee concentriche ed equidistanti, con centro il punto di origine della camera. Un raggio (in rosso) con origine nel centro della camera passa per un pixel ed il suo corrispondente voxel.

La stessa ricostruzione di profondità nel processo umano prevede infatti una visione stereoscopica della scena, avendo così a disposizione copie di immagini della stessa scena, riprese da angolazioni leggermente differenti tra di loro. Proprio queste differenze permettono di recuperare informazione. Allo stesso modo, anche nell'ambito della visione computazionale, ove possibile, si ricorre ad una visione stereoscopica della scena di interesse. In questo modo è possibile, conoscendo i parametri di calibrazione e la posizione delle telecamere, triangolare i dati ottenuti. Riproducendo invece la capacità umana di recuperare informazione di profondità attraverso l'analisi delle variazioni della scena nel tempo, molti algoritmi ricorrono alla Structure from Motion (SfM), analizzando i segnali locali di movimento nella sequenza temporale, utilizzando dunque più immagini successive provenienti anche dalla stessa telecamera. In molti casi però non è possibile ricorrere a sistemi ad hoc per la ricostruzione tridimensionale. Basti pensare alla conversione di materiale già presente, oppure ad utilizzi in mancanza di dispositivi adeguati. Si rende dunque necessario cercare di estrapolare informazioni dalla proiezione bidimensionale (l'immagine 2D) della scena 3D. È facile intuire le problematiche introdotte, visto che l'immagine non è in rapporto univoco con l'immagine reale. Mancando l'informazione di profondità, l'immagine può essere associata ad un'infinità di strutture tridimensionali corrette. In particolare ogni punto dell'immagine può essere associato ad una retta di punti tridimensionali. Non esiste infatti una soluzione matematica rigorosa al problema. Basti pensare che risulta impossibile capire se una figura rappresenti una scena reale tridimensionale oppure sia a sua volta una semplice fotografia della scena stessa, risultando in questo caso priva di

profondità. È comunque possibile sfruttare molte delle caratteristiche monocolari che contraddistinguono un'immagine, quali linee, luminosità zone di colore, texture, messa a fuoco. Il cervello umano le percepisce attraverso la vista e attraverso l'esperienza, è abituato ad estrapolare informazioni di tridimensionalità. Molti algoritmi di conversione 2D/3D cercano di riprodurre artificialmente tale processo. Se da un lato è vero che nella comprensione della profondità nel processo umano gioca un ruolo fondamentale la visione stereoscopica di occhio destro e sinistro, dall'altro sono altrettanto importanti, e per certi versi ancor più fondamentali, le analisi che il nostro cervello compie sulle immagini ricevute come input: nella percezione umana delle dimensioni intervengono diversi fattori, come parallasse di movimento, parallasse stereoscopica, ed altre caratteristiche che captiamo semplicemente dalle informazioni monocolari: convergenza delle linee parallele, riduzioni prospettiche, occlusioni, ombre, offuscamento, conoscenza delle geometrie degli oggetti. Sicuramente tra tutte queste le prime due meritano di essere messe in primo piano per importanza nel processo cognitivo. Per capire quanto siamo abituati a queste ricostruzioni è sufficiente guardare l'immagine in figura (2.2): anche in caso di oggetto monocromatico e texture uniforme, sono sufficienti le differenze di illuminazione delle diverse facce e l'analisi delle linee prospettiche affinchè la nostra mente ci suggerisca la volumetria del cubo.

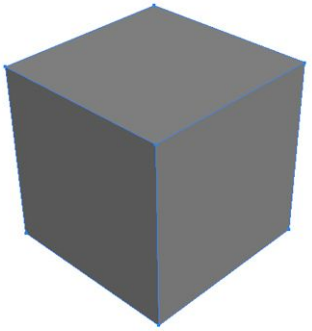

FIGURA 2.2: Percezione di un cubo attraverso proprietà monocolari.

In ambito computazionale possiamo dunque basarci anche solo su questi particolari per tentare di stimare l'ambiente 3D di partenza, simulando il processo umano. I risultati ottenibili non sono di certo equiparabili a quelli che si ottengono con le metodologie classiche, e risultano molto più restrittive nel loro campo di utilizzo. Diversi metodi risultano funzionare solo in un limitato campo d'azione, per il quale sono stati progettati: ambienti indoor od outdoor, oppure con una limitata serie di oggetti. Questi ancora il più delle volte richiedono l'intervento umano nel

definire delle zone di confine o degli oggetti in primo piano, in modo da garantire risultati più funzionali, od evitare errori grossolani. Molti algoritmi inoltre partono da assunzioni sul campo di lavoro molto stringenti: presenza di soli oggetti con profondità verticale od orrizzontale, presenza di un ground con segmentazione manuale dello stesso, divisione dell'immagine in macrozone semplici (ground/ wall/ sky).

2.2 Ambiti d'uso

2.2.1 Condizioni di applicabilità

Nonostante le limitazioni appena viste, la conversione da due a tre dimensioni risulta sicuramente interessante per alcune applicazioni, per due motivi principali:

1. Facilità di utilizzo: applicabile anche in situazioni ove non sia possibile una ricostruzione efficiente, risulta una soluzione certamente meno sofisticata, ma è sufficiente una singola telecamera/immagine. Questo la rende una soluzione eterogenea ed economica, in ambiti dove non si necessita di alta fedeltà nella ricostruzione. Si pensi ad applicazioni nella robotica, in cui si è più interessati a riconoscere "cosa" e "dove" sia presente sulla scena, in modo da potervi interagire.

2. Focalizzazione sugli aspetti intrinseci dell'immagine: gli algoritmi sviluppati recuperano informazioni partendo da una sola immagine, aspetto che viene spesso trascurato nella ricostruzione stereoscopica. Questo tipo di ricostruzione non esclude quindi una possibile successiva ricostruzione sfruttando anche la triangolazione da più immagini di input. I risultati ottenuti in questo campo potrebbero andare a coadiuvare gli algoritmi basati sulla sola triangolazione, dove necessario. Nulla vieta di eseguire una stima basata su visione stereoscopica, ed applicare algoritmi di post processing attraverso le pecularietà delle immagini prese una ad una. Questo renderebbe il processo similare a quello della nostra esperienza quotidiana: visione stereoscopica filtrata attraverso l'analisi delle caratteristiche monocolari.

2.2.2 Utilizzo, stato dell'arte e possibili scenari d'uso futuri

La conversione di modelli bidimensionali a tridimensionali è utilizzata in diversi ambiti: partendo dalla fotografia, numerose sono le applicazioni che permettono di convertire le proprie immagini in copie tridimensionali, sia creando immagini rosso(ciano)/ verde(magenta), sia creando delle mesh navigabili ed interattive. Vengono offerti dei servizi di conversione delle immagini e di successiva stampa del modello ottenuto, tramite apposite stampanti provviste di frese a 3 assi su modello plastico, come quello offerto da Kodak [12]. Lo scopo di queste applicazioni risulta sicuramente più ludico, e volto ad incrementare l'interattività con l'utente. A volte l'effetto 3D è anche ottenuto mettendo in sequenza diverse copie dell'immagine ricostruite, con angolazione leggermente differente. Sicuramente interessante risulta lo sviluppo di ambienti navigabili, in cui l'utente può muoversi all'interno di un'immagine, ed avere così la sensazione di visitare un dato luogo. Questo tipo di conversione sta inoltre prendendo sempre più piede nella ricostruzione 3D video. I televisori 3D di ultima generazione (stereoscopici), in mancanza ancora di un'offerta ampia di contenuti video 3D nativi, integrano una funzione di ricostruzione tridimensionale al volo dei filmati standard. Partendo dal segnale video classico viene ricostruita la depthmap dei singoli frame, che viene utilizzata per generare una vista parallela, necessaria alla visione stereoscopica. In questo caso particolare però va ricordato come la ricostruzione video sia fortemente avvantaggiata rispetto a quella fotografica, potendo sfruttare la forte correlazione che esiste tra frame successivi, disponendo in qualche modo di più visuali della stessa scena: lo Stereo from Motion menzionato prima, dove fondamentale risulta trovare la correlazione tra i frame successivi della scena. Legato a questo ultimo settore va infine ricordata la riconversione 3D di film che in realtà sono stati girati con telecamere 2D tradizionali. Si parla dunque di ricostruzione virtuale delle multicamere. Viene cioè costruita una seconda (o più) visuale, con la quale è possibile creare due canali diversi della stessa scena. Questa soluzione seppur molto costosa, risulta sicuramente vantaggiosa, oltre all'unica possibile per determinate pellicole. La qualità per quanto riguarda questi sistemi non può certo competere con quella dei film realizzati nativamente in 3D. Particolare attenzione merita inoltre lo sviluppo di algoritmi per il riconoscimento di oggetti: nel campo della robotica sono stati sviluppati diversi progetti con l'intento di analizzare e riconoscere l'ambiente circostante partendo dalla cattura di immagini attraverso una semplice webcam.

Attraverso l'utilizzo di un database di oggetti predefiniti, da confrontare con le scansioni ottenute, è possibile ottenere ottimi risultati in ambienti relativamente poveri, dove è presente una quantità limitata di possibili oggetti differenti. In ambito domestico ad esempio questo permetterebbe una navigazione completamente autonoma ed indipendente dell'ambiente, andando ad interagire con gli ostacoli, riconoscendoli o evitando la collisione con essi, senza richiedere l'uso di sistemi complessi come scansione laser o confronto stereoscopico. In futuro tali soluzioni potrebbero essere usate anche nel campo della guida automatizzata, con ruolo primario o secondario (nel riconoscimento di ostacoli o segnali). Tali soluzioni, già ampiamente implementate (riconoscimento automatico dei pedoni [13]) attraverso soluzioni tecnologiche radar, potrebbero portare ad una riduzione dei costi e quindi un mercato più ampio.

2.3 Algoritmi proposti

Nel corso degli anni sono stati sviluppati diversi sistemi che sfruttano alcune pecularietà tipiche delle immagini.Alcuni di essi si basano sull'informazione di intensità, sulle linee orrizzontali/verticali, e sui punti prospettici [14] [15] ai più sofisticati come Make 3D[16].

Tuttora non esiste il sistema perfetto, bensì algoritmi studiati ad hoc per settori diversi, che presentano notevoli limiti di applicabilità o realizzazione. Un sistema che pretenda di essere eterogeneo dovrebbe prescindere da molte delle supposizioni fatte e sfruttare un'insieme delle caratteristiche usate nei vari settori. Tra i diversi metodi proposti meritano sicuramente di essere citati alcuni, per importanza o motivazioni storiche. Va certamente menzionato il *Shape from Shading*[10]: metodo che cerca di ricostruire la profondità basandosi sulla diffusione della sorgente luminosa: le variazioni di luminanza presenti all'interno di un'immagine possono essere sfruttate per ricavarne informazioni utili sulla dispozione in terza dimensione degli oggetti. Gioca ovviamente un ruolo fondamentale l'origine della fonte luminosa, che risulta incognita nel problema. Nel 1997 Horry et al. [17] svilupparono un algoritmo, basato sui punti e le linee prospettiche, in grado di suddividere l'immagine in 5 diverse zone: *floor, right wall, left wall, rear wall* e *ceiling*, e creare così un modello semplice ma capace di rendere l'effetto tridimensionale, il cosidetto TIP: *Tour into the picture*.

Più recentemente, basandosi sempre sui principi del pioneristico lavoro di Horry et al., sempre con l'obiettivo di ricreare una scena tridimensionale e navigabile dell'immagine, troviamo il lavoro di Iizuka [18] che sfrutta questa volta la segmentazione in superpixel: questa consiste nello scomporre ogni singolo pixel come vettore a cinque dimensioni, contenente l'informazione sulla posizione (x,y) e sul colore (L, u, v), ed aggregare insieme pixel coerenti appartenenti con una certa probabilità ad una stessa regione. Il metodo prevede dei semplici input da parte dell'utente (che dovrà cerchiare gli oggetti *foreground* e segnare la linea di demarcazione tra *background* e *floor*) al fine di evitare i frequenti errori di riconoscimento, spesso comuni a tutti gli algoritmi attuali. In questo modo si è in grado di dare un'animazione 3D di qualità certamente superiore, provvedendo anche a stimare le occlusioni (*Patch Match*). Di contro per ottenere questi risultati è presente una forte interattività con l'utente, che deve provvedere a segnare le zone di interesse, confinando e coadiuvando così il lavoro dell'algoritmo, che risulta tutt'altro che automatizzato.

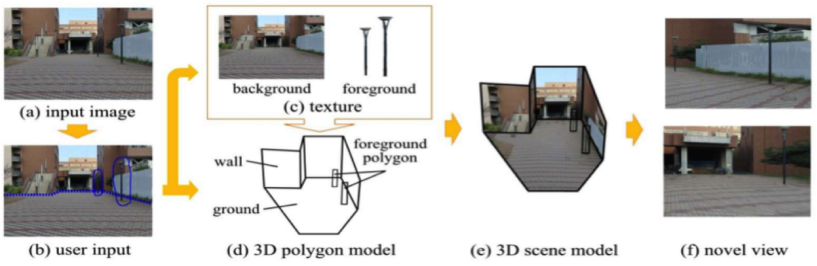

FIGURA 2.3: Costruzione della scena 3D (a) data una singola immagine, l'utente specifica attraverso l'inserimento di polilinee le figure in primo piano e delimita il piano di background.I diversi elementi vengono combinati per ottenere il modello 3D che l'utente può navigare (f)

Altro lavoro che va citato è quello di Derek Hoiem in [19] che sfrutta le informazioni a livello macroscopico di un'immagine. Per primo egli capì che era possibile utilizzare le conoscenze geometriche a priori date dal calcolo statistico in un determinato ambito: restringendo il campo alle scene outdoor è quasi sempre possibile distiguere diverse zone di interesse: *ground, sky* e *vertical*, sfruttando così le informazioni regionali dell'immagine per il recupero dell'informazione di profondità, in maniera completamente automatizzata. Questo modello è stato poi ripreso anche in lavori successivi.

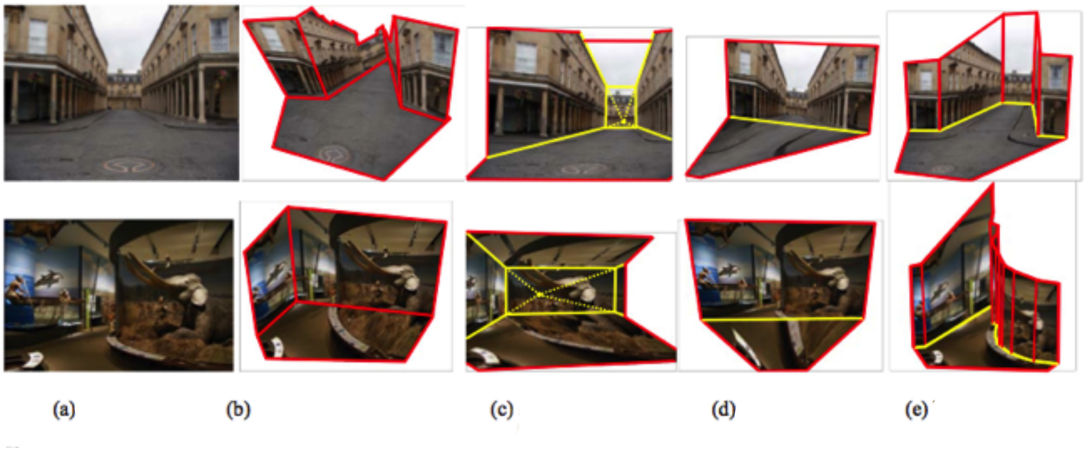

FIGURA 2.4: Risultati tipici del sistema di Horry confrontato con altre soluzioni per creare ambienti 3D navigabili: a) immagine iniziale b) photo pop-up (Hoiem 2005) c)spider mesh (Horry 1996) d) vanishing line (Kang 2001) e) boundary lines. In giallo le linee date come input dall'utente, in rosso le linee di confine rimanenti

Questo sistema, ed in minor modo quello elaborato da Saxena [16][20], pur essendo completamente automatizzati, portano ad errori piuttosto evidenti nel caso di errata segmentazione o stima della profondità, la quale, seguendo un modello probabilistico, non garantisce correttezza nella totalità dei casi.

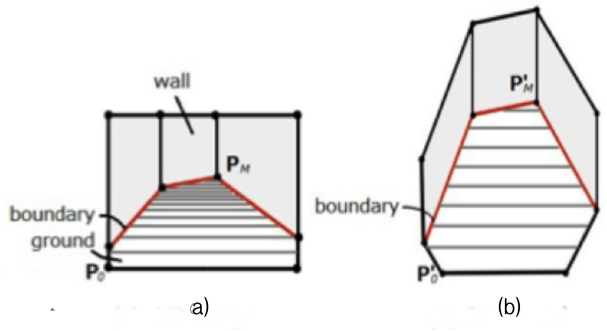

FIGURA 2.5: a) immagine bidimensionale b) modello 3D

Con finalità completamente diverse si presenta invece la categoria degli algoritmi che fanno uso di un database di forme base e di oggetti predefiniti: in questo caso lo scopo principale non è quello di garantire la fedeltà tra immagine e modello 3D, bensì solo quello di riconoscere le forme, scomponendo l'immagine in segmenti elementari bidimensionali e trovando un riscontro tra le forme già presenti nella libreria: un metodo simile viene proposto in [21] e trova applicazione nel campo della robotica, per il riconoscimento degli oggetti e l'interazione con questi. Un concetto simile viene anche usato in [22], dove la ricostruzione dei volti a partire da un'immagine è fatta attraverso la compensazione della luminosità, la *detection* del volto, l'estrazione dei punti base e seguente applicazione di *patch* da un database di volti 3D scannerizzati precedentemente al laser. Queste classi basate su modelli risultano adatte a rappresentare oggetti solo nel campo per le quali sono state studiate, e trovano poco riscontro al di fuori del loro contesto, essendo difficile una generalizzazione.

Capitolo 3

L' algoritmo di Saxena: Make 3D

3.1 Presentazione dell' algoritmo

Nato da un progetto dell'università di Stanford, Make 3D [23] è un'algoritmo, oltre ad essere un'applicazione online, capace di trasformare qualsiasi foto in un modello tridimensionale del suo contenuto, permettendo così la stima della struttura tridimensionale degli oggetti. Questo tipo di tecniche rappresenta sicuramente un ambito di ricerca emergente. Make 3D però fa un grosso passo avanti rispetto agli algoritmi precedenti. Abbandonando alcune delle assunzioni più frequenti sull'immagine (ad esempio sull'esistenza di una linea marcata dell'orizzonte) ed effettuando invece un' analisi più approfondita di tutti i dettagli dell'immagine legati alla profondità (gli stessi utilizzati dall' occhio umano), si è arrivati ad ottenere un algoritmo che fornisce risultati apprezzabili in un numero quasi radoppiato dei casi. Il grosso limite finora infatti non era tanto rappresentato dalle prestazioni ottenibili, bensì il limitato numero di casi e condizioni favorevoli necessarie per ottenerle. Nel lavoro di Saxena l'algoritmo compie una sorta di training utilizzando delle immagini rappresentative del tipo di dati sul quale verrà utilizzato (panorami, ambienti urbani, etc), in modo da ottimizzare i parametri, confrontandoli con quelli ottenuti con misure di profondità come ad esempio uno scanner laser.

3.2 Principi alla base dell'algoritmo

Lo studio svolto dal gruppo di Saxena si pone come obiettivo quello di stimare strutture 3D complesse, sempre a partire da una singola immagine, anche prive di strutture geometriche ben definite. Lo scopo è quello di ottenere modelli sia quantitativamente accurati, sia piacevoli a livello visivo. L'algoritmo sfrutta una quantità di segnali visivi per stimare gli aspetti tridimensionali dei superpixel. Per prima cosa l'immagine viene scomposta in diverse piccole sezioni. Per ognuna di queste viene usato un MRF (Markov Random Field) per desumere un insieme di parametri planimetrici che stabiliscono sia la posizione che l'orientamento della singola sezione (patch).

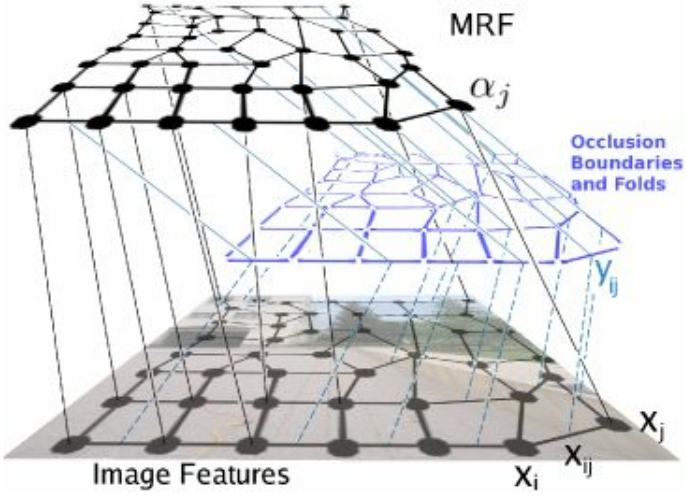

FIGURA 3.1: Schema visivo della segmentazione con MRF

Make 3D provvede poi a stimare anche le relazioni che intercorrono tra le diverse parti dell'immagine. Il modello risulta molto eterogeneo, dal momento che non assume nessun presupposto sulla struttura in esame, al di fuori del fatto che essa possa essere scomposta come un'insieme finito di piccoli piani. Questo permette l'utilizzo dell'algoritmo in ambienti che vanno anche al di fuori da quelli presi in esame, o considerati come situazione standard, sulle quali il sistema è stato testato.

3.2.1 Informazioni monocolari di profondità

In generale i dettagli che ci permettono la ricostruzione tridimensionale a partire da un'immagine sono diversi. Attraverso un'attenta analisi di questi e cercando

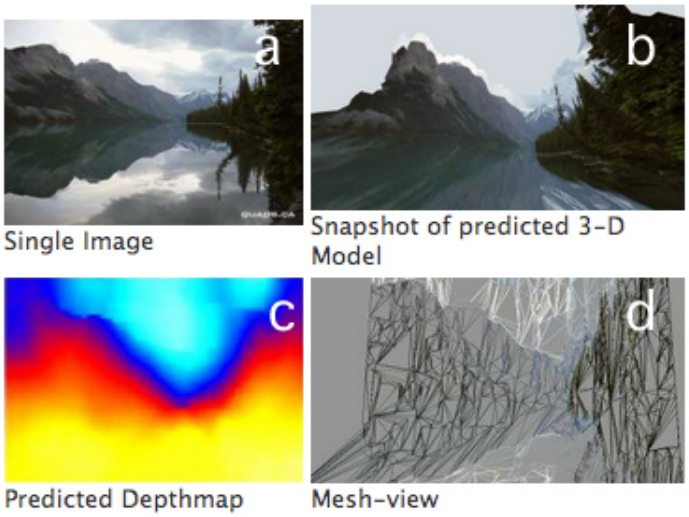

FIGURA 3.2: Risultato prodotto dall'algoritmo, con (a) immagine iniziale, (b) immagine rilevata dal modello 3D, (c) depthmap relativa e (d) visione del modello come mesh

di sfruttarne appieno le potenzialità, è possibile derivare le informazioni contenute in esse ed operare anche in assenza di triangolazione, cioè utilizzando una singola immagine. I principali sono:

- *variazioni di tessitura (texture)*: ove l'immagine presenta un forte cambiamento di colore e luminanza è facile che si presenti un cambio di piano, ovvero i punti considerati giacciono su superfici piane diverse. Questo elemento riguarda le variazioni ad alte frequenze nell'immagine.

- *gradiente di texture*: la presenza di un cambiamento graduale della texture complessiva può essere indice di un cambiamento graduale di profondità della superficie in esame. Esempio tipico quello di un prato che, visto a distanze diverse, presenta componenti ad alta frequenza a distanze ravvicinate, e solamente componenti a bassa frequenza a distanze ragguardevoli o fuori dal fuoco dell'immagine, pur rappresentando lo stesso elemento e la stessa crominanza globale. A differenza delle variazioni prese in esame precedentemente, per il gradiente vengono considerati quegli elementi che variano lentamente lungo un certo asse, sia per profondità, sia per caratteristiche.

- *interposizione*: le caratteristiche che presenta un'oggetto collocato su un altro piano focale rispetto a quello di fondo.

- *occlusioni*: le occlusioni rappresentano un limite dell'analisi attraverso l'uso di una singola immagine. Non è possibile recuperare le informazioni di oggetti che non sono visibili nell'immagine, se non è disponibile un'altra visuale che ne è priva. Ad ogni modo saper identificare quando queste avvengono può prevenire eventuali errori di ricostruzione, ed in alcuni casi si può stimare la natura di quest'ultima.

- *conoscenza delle dimensioni degli oggetti*: il nostro cervello è in grado di posizionare correttamente sull'asse della profondità oggetti di cui ha esperienza. Un albero o un edificio che nell'immagine presentano una certa dimensione possono essere collocati in un range piuttosto veritiero di distanza reale, senza ulteriori analisi.

- *luci ed ombre*: l'occhio umano è in grado di stimare la direzione sorgente della fonte (o delle fonti) di illuminazione della scena: l'illuminazione delle superfici e le ombre che si creano aiutano notevolmente il processo di analisi. Questo fattore risulta molto complesso e di difficile gestione per l'analisi automatizzata.

- *defocus*: le parti a fuoco e le parti sfocate di una scena presentano elementi distintivi e sono perciò identificabili. Nella fotografia classica il piano focale di un'immagine è unico, in corrispondenza dei soggetti della foto. A seconda della lunghezza focale utilizzata questo fenomeno è presente in maniera più o meno evidente. Soggetti pienamente a fuoco avranno una distanza simile dal centro della camera.

3.2.2 Caratteristiche dell'immagine analizzate dall'algoritmo

Come già sottolineato un'immagine rappresenta la proiezione bidimensionale su un piano di una struttura a tre dimensioni. Si capisce quindi come da questa la struttura 3D originale risulti ambigua. Si possono dedurre infinite proiezioni da essa, alcune più probabili, altre meno, a seconda delle informazioni che si possono dedurre in maniera piuttosto complessa attraverso il nostro cervello. Alcune di queste comunque possono essere elaborate computazionalmente. In particolare si porge particolare attenzione a:

- *Caratteristiche dell'immagine e profondità*: le caratteristiche di un'immagine di un oggetto (es. texture) e la sua profondità risultano essere fortemente correlate.

- *Connettività*: ad esclusione delle occlusioni (da qui l'importanza di saperle identificare) tutte le patch (supersegmentazioni) hanno una forte probabilità di essere connesse tra loro. Questo assunzione limita notevolmente il numero di soluzioni possibili, restringendo di molto il nostro campo di ricerca (teoreticamente infinito se non facessimo nessuna ipotesi sulla connessione di patch adiacenti).

- *Coplanarità*: due patch adiacenti, e che presentano caratteristiche simili, sono molto probabilmente coplanari.

- *Colinearità*: lunghe linee dritte nell'immagine bidimensionale possono essere spigoli di elementi nel tridimensionale (lati di edifici, finestre).

È da notare come nessuno di questi preso singolarmente sia un'elemento in grado di ricondurci univocamente al modello 3D, ma se ad ognuno di essi viene dato un'indice di affidabilità e si usano congiuntamente in un modello Markoviano, è possibile ricondurci ad una struttura fedele all'originale.

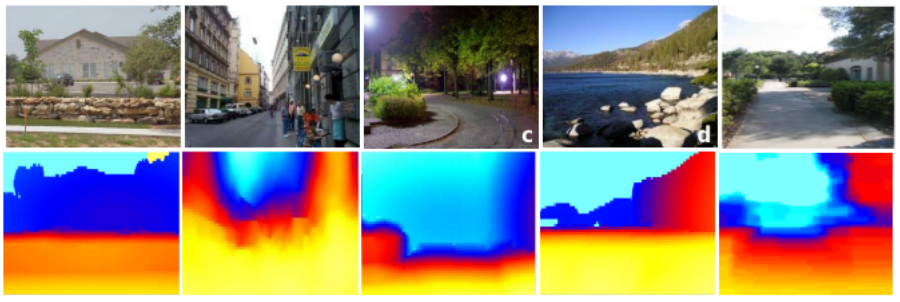

FIGURA 3.3: Depth map prodotte dall'algoritmo (riga inferiore) per le relative immagini. In giallo gli elementi con profondita' minore, in blu gli elementi con profondita' maggiore

3.2.3 Segmentazione e superpixel

Il primo passo che viene svolto è quello di sfruttare un'algoritmo di segmentazione e suddividere l'immagine in tante piccolissime aree. La suddivisione operata dalla segmentazione potrebbe risultare eccessivamente fine in presenza di ampie regioni

uniformi, risultando così coperte da molti superpixel. Tuttavia ognuno di essi presenterà caratteristiche uniformi, cioè riporterà gli stessi parametri planari.

L'algoritmo a questo punto cerca di stimare la posizione e l'orientazione di ciascun superpixel. Un elemento distintivo del lavoro svolto dall'università di Stanford risulta certamente la possibilità di non avere solo orientazioni verticali od orizzontali, bensì anche inclinate obliquamente.

Successivamente viene utilizzato un'edge detector, in modo da filtrare i risultati ottenuti tramite l'analisi del gradiente: pur in presenza di un forte gradiente infatti non sempre vi è uno spigolo o un cambio di orientazione. Si pensi ad esempio ad un ombra marcata proiettata su una superficie. Seppur tutti questi fattori non siano sufficienti da soli ad un'accurata stima del modello, se uniti al processo Markoviano rappresentano una componente essenziale, andando a migliorare notevolmente le prestazioni dello stesso processo eseguito in assenza di questi parametri.

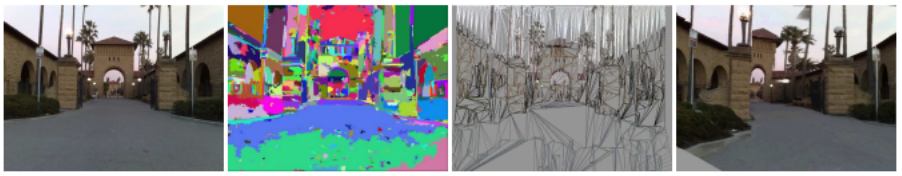

FIGURA 3.4: da sinistra verso destra: immagine originale, sovrasegmentazione dell'immagine per ottenere i superpixel, modello 3D predetto, screenshot del modello 3D

3.3 Calcolo della profondità assoluta

Data una patch i-esima dell'immagine $I(x,y)$, vengono calcolate le seguenti feature. Attraverso l'uso di 17 differenti filtri (9 maschere di Law, 2 canali di colore, e 6 gradienti relativi alla texture) viene elaborata la patch corrente e calcolate le varie uscite.

FIGURA 3.5: Filtri di Law e filtri gradiente utilizzati

Da queste vengono ricavate l'energia e la curtosi relativa a ciascun filtro convoluto per i pixel della patch in questione, secondo la formula

$$E_i(n) = \sum_{(x,y)\in superpixel(i)} |I(x,y) * F_n(x,y)|^k \tag{3.1}$$

con k= {*1,2*} dove con k=*1* abbiamo la somma dell' inviluppo dell' uscita e con k=*2* la somma quadratica ovvero l' energia. I rappresenta i valori dell'immagine per il superpixel, mentre $F(n)$ rappresenta la matrice del filtro. Al fine di stimare la profondità assoluta di una patch, le informazioni locali centrate attorno ad essa sono insufficienti, e sono necessarie più informazioni globali. Si ovvia a questo problema catturando tali informazioni a differenti scale/risoluzioni. Oltre a risolvere parzialmente questo problema, così facendo è possibile catturare oggetti di dimensione completamente diversa: un oggetto grande sarà catturato nella scala a più bassa risoluzione, mentre quelli più piccoli nella scala ristretta ad alta risoluzione. Al fine di ottenere informazioni addizionali il calcolo delle feature della patch avviene sia dallo studio della patch stessa che dalle 4 patch vicine. Questo viene ripetuto per tutte e tre le scale utilizzate, cosicchè il vettore relativo ad una patch contiene anche le informazioni delle patch adiacenti, anche se queste rappresentano pure punti spazialmente distanti nel caso della scala maggiore. Ripetendo dunque il calcolo delle feature su 3 livelli sia per il superpixel in questione che per i 4 adiacenti, otteniamo un vettore delle features di dimensione $34 * (4 + 1) * 3$ alle quali si aggiungono 14 feature riguardanti la forma calcolate unicamente sul superpixel, per un totale di 524 dimensioni

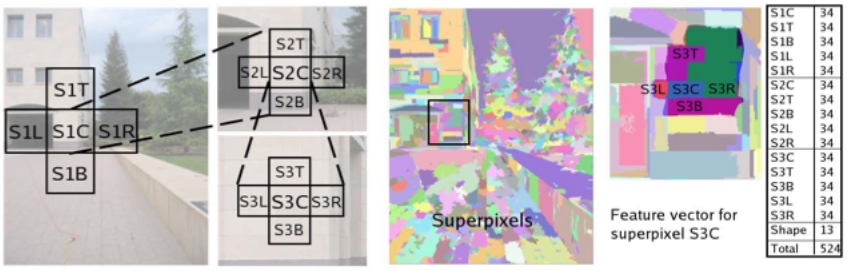

FIGURA 3.6: Vettore delle feature di un superpixel, che include le informazioni sui superpixel adiacenti a diversi livelli di scala

3.4 Calcolo della profondità relativa

In modo simile viene svolto il confronto relativo di profondità tra 2 patch: usando gli stessi filtri, e salvando (in modo più accurato) le uscite su entrambe le patch,

vengono messe in relazione le depth relative. Vengono poi salvate le differenze tra i due istogrammi.

3.5 Modello Markoviano: descrizione e motivazioni di utilizzo

Nei problemi di elaborazione delle immagini occorre valutare un'entità (pixel, oggetto o altro) all'interno di un certo contesto (pixel vicini od oggetti vicini). La valutazione ha lo scopo di associare un nuovo valore (colore o significato) all'entità. Il principio del vicinato è quindi ben noto in Computer Vision (CV), ed è presente in tutte le classiche tecniche di elaborazione dell'informazione visiva. I Random Field di Markov (MRF) riescono a mescolare tre caratteristiche interessanti tanto da renderli adatti alla CV: vicinato, unicità teorica e ottimalità. Cioè con un'unica base teorica, quella dei MRF appunto, si possono sviluppare tecniche diverse, dal low level all'high level processing, per trovare soluzione ottime a diversi problemi. L'ottimalità è valutata probabilisticamente per cui si trova la soluzione "più probabile" al problema. Tutto ciò rende gli MRF particolarmente attraenti. Il lato negativo della teoria MRF è che mostra una struttura alquanto complessa poiché mescola tecniche probabilistiche a tecniche proprie della ricerca operativa. La risoluzione di problemi di computer vision (CV) ha avuto un'evoluzione da approcci euristici verso una più sistematica ricerca di teorie unificanti [24]. In questo processo i ricercatori si sono resi conto che un problema dovrebbe essere risolto in termini di ottimalità tenendo conto dei vincoli di contesto presenti. Questi ultimi, in particolare, sono necessari per la corretta interpretazione dell'informazione [25], non solo nel campo della CV. La ragione principale per l'uso dell'ottimalità si ritrova nell'esistenza di incertezza in ogni problema legato alla CV, ma non solo: rumore o altri fattori degradanti derivanti da quantizzazione o disturbi; diversa illuminazione degli oggetti e loro parziale/totale occlusione in una scena; deformazione della loro struttura dovuta alle ottiche usate per catturare l'immagine; anomalia nell'ingegnerizzazione della conoscenza in generale. È abbastanza evidente che situazioni di questo tipo rendono quasi impossibile la ricerca della soluzione perfetta: ha più senso ricercare una soluzione ottima dati certi criteri di ottimalità. Ci sono tre questioni fondamentali nei problemi di ottimizzazione in generale: rappresentazione del problema; definizione della funzione obiettivo; algoritmo di ottimizzazione. Nella rappresentazione del problema, un

ruolo fondamentale, come vedremo, è assunto dal concetto di legame di vicinato: questo determina che la forma più generale per rappresentare un problema per MRF consiste in un grafo. La rappresentazione a matrice o a vettore, spesso utilizzate in CV, sono particolari modi di intendere i grafi. La funzione obiettivo mappa istanze di soluzioni verso numeri reali che ne misurano l'ottimalità. La formulazione determina come i diversi vincoli (proprietà dei pixel e relazioni tra essi) sono codificate in una funzione. Come vedremo più avanti, la teoria dei MRF consente di formulare, a partire da considerazioni statistiche, una funzione obiettivo probabilistica. L'ottimizzazione della funzione obiettivo consente di ricercare una soluzione ottima all'interno di uno suo spazio ammissibile. La ricerca dell'ottimo offre spunti di studio notevoli: l'esistenza di minimi locali; la presenza di funzioni obiettivo non convesse; l'efficienza degli algoritmi nel tempo e nello spazio. In questo ambito risulta impossibile definire una metodologia ottimale in ogni situazione ma, piuttosto, regole che consentono in determinate situazioni di fornire soluzioni adeguate. In genere una suddivisione grossolana dei metodi viene compiuta tra algoritmi che producono soluzioni locali o globali. Per capire meglio la rappresentazione della MRF come grafo,vanno introdotti tutti quegli elementi che consentono di arrivare a formulare una funzione obiettivo per un problema di CV.

3.5.1 Labeling

Un problema di labeling è specificato in termini di elementi (site) e label: il labeling assegna una label a ciascun sito. È in generale una mappatura:

$$f : S \to L$$

dove S è l'insieme degli elementi e L quello delle label. Un esempio di insieme di elementi può essere l'insieme dei pixel in un'immagine o, più in generale, l'insieme dei nodi in un grafo. Per le label invece si possono considerare i valori (colori) assumibili da un pixel o i valori (anche simbolici) assumibili dal nodo di un grafo. È ovvio che lo spazio delle configurazioni totali è identificato da:

$$F = L^{|S|}$$

In certe circostanze le label ammissibili per un sito possono essere diverse da quelle ammissibili per un altro sito: il problema si generalizza e per la sua soluzione basta tenerne conto nell'insieme dei vincoli caratterizzanti.

3.5.2 Vicinato

In S gli elementi sono collegati da un sistema di vicinato N. Per il sito i, N_i ha le seguenti proprietà

$$i \notin N_i$$

$$j \in N_i \Leftrightarrow i \in N_j$$

Un esempio di vicinato può essere la distanza euclidea. Ne deriva che la coppia *(S,N)* è un grafo.

Nel caso della struttura a matrice, si può visualizzare la distanza di vicinato ad un pixel dato (X) come in figura (3.7).

5	**4**	**3**	**4**	**5**
4	**2**	**1**	**2**	**4**
3	**1**	**X**	**1**	**3**
4	**2**	**1**	**2**	**4**
5	**4**	**3**	**4**	**5**

FIGURA 3.7: Distanza di vicinato nel caso di una matrice

3.5.3 Clique

Una clique è una sottoparte del grafo formato dai siti e dal sistema di vicinato: $c \subseteq (S, N)$

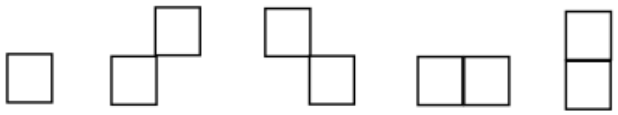

FIGURA 3.8: Forme possibili di Clique per finestre 3x3

In genere si usa identificare le clique non solo per la loro forma (quando si ha a che fare con strutture regolari) ma anche per la loro cardinalità. Per convenzione la cardinalità di una clique è espressa a pedice:

$$C_1 = \{i | i \in S\};\ C_2 = \{\{i, i'\} | i' \in N, i \in S\};\ C_3 = \{...\}$$

E l'insieme delle clique è dato ovviamente da:

$$C = \bigcup_k Ck$$

La forma delle clique è invece molto importante soprattutto per questioni legate a tecniche di elaborazione low-level nella CV. Si hanno varie forme di clique (si pensi ad una finestra 3x3) come in figura (3.8).

Le clique del primo tipo sono solo un pixel; le altre sono invece formate dalle 4 direzioni fondamentali (2 diagonali e 2 assiali) e così via.

L'uso di conoscenza di contesto è necessaria per catalogare correttamente una entità [25], qualunque sia lo spazio dei problemi considerato. A questo proposito le informazioni provenienti dal vicinato sono usate come contesto. In termini probabilistici è semplice considerare i vincoli di contesto poiché si possono codificare usando la probabilità condizionata

$$P(f_i | \{f_i'\}) = P(f_i' | \{f_i\}) \frac{P(f_i}{P(f_i')} \tag{3.2}$$

che può essere letto come: probabilità che il sito i assuma il valore f_i a fronte del fatto che nel suo vicinato vi è l'insieme di valori f_i'. Il problema non è complesso se ciascuna label è indipendente:

$$P(f_i|\{f_i'\}) = P(f_i)$$

e la probabilità globale (data da tutte le label nel loro complesso) è banalmente:

$$P(f) = \prod_{i \in S} P(f_i)$$

Il problema diventa complesso quando dalla probabilità condizionata si vuole calcolare la forma di quella non condizionata e le label sono tra loro interagenti. MRF fornisce la soluzione per trattare questo caso [26].

Le incertezze intrinseche nelle problematiche legate alla CV, così come anche in altre tipologie di problemi, giustificano l'utilizzo dell'ottimalità per il calcolo della soluzione. La soluzione ottima si trova in corrispondenza della soluzione a probabilità massima. Conoscendo le distribuzioni dei dati a disposizione o della soluzione che si intende ritrovare, vi sono vari metodi statistici che possono aiutare nella ricerca della soluzione [27][28]. Specificatamente:

- *Maximum Likelihood (ML)*: quando si conoscono le caratteristiche della distribuzione dei dati a disposizione ma non si ha idea su come si distribuirà la soluzione.
- *Maximum Entropy (ME)* : quando la situazione è opposta rispetto al ML.
- *Maximum A Posteriori (MAP)* : quando si conoscono entrambe le distribuzioni. Le precedenti sono casi particolare di questa.
- *Minimum Description Length (MDL)* : in base al principio per cui la soluzione al problema è quella che necessita del minor insieme di vocaboli di un linguaggio per la sua descrizione. È dimostrata l'equivalenza con MAP.

3.6 Markov Random Field

Il processo Markoviano, che porta a determinare le coordinate tridimensionali del modello voluto, viene eseguito per ciascuna delle 3 segmentazioni. Per ogni superpixel viene calcolato il vettore normale al piano, e questo viene messo in relazione ai superpixel vicini. A seconda che l'andamento di una segmentazione sia congruo con quelle delle altre dimensioni, è possibile pesare in maniera differente ciascuna soluzione, fondendone in un secondo momento insieme i dati. Per la ricostruzione 3D assume un'importanza fondamentale la definizione di errore relativo (frazionario) di profondità. Detta d la profondità reale di un punto, e $\hat{d}$ la profondità da noi stimata per quel medesimo punto, l'errore relativo si definisce come:

$$(\hat{d} - d)/d = \hat{d}/d - 1 \tag{3.3}$$

La minimizzazione della quantità appena definita assume un ruolo fondamentale nel nostro processo Markoviano.

Al fine di catturare le relazioni presenti tra i parametri di ogni piano, le caratteristiche dell'immagine, e le altre proprietà quali coplanarità, connettività e colinearità, formuliamo il nostro MRF secondo la seguente equazione:

$$P(\alpha|X, v, y, R; \theta) = \frac{1}{Z} \prod_i f_1(\alpha_i|X_i, v_i, R_i; \theta) \prod_{i,j} f_2(\alpha_i, \alpha_i|y_{ij}, R_i, R_j) \tag{3.4}$$

dove, con α_i indichiamo i parametri planimetrici del superpixel i-esimo, rappresentato da S_i punti. Usiamo x_{i,s_i} per denotare le features del punto s_i nel superpixel i-esimo. Z è semplicemente una costante di normalizzazione. $X_i = \{x_{i,s_i} \in \mathbb{R}^{524} : s_i = 1, ..., S_i\}$ sono le caratteristiche (features) del superpixel i-esimo. Similmente, $R_i = \{R_{i,s_i} : s_i = 1, ..., S_i\}$ è l'insieme dei raggi normali alla camera per il superpixel i-esimo, v indica la confidenza che diamo alle caratteristiche locali calcolate nel predirre la depth. Il primo termine, $f_1(.)$, modella i parametri planimetrici come funzione delle caratteristiche dell'immagine x_{i,s_i}. Abbiamo dunque che, definita la profonditale come d_{i,s_i}, possiamo scrivere:

$$R_{i,s_i}^T \alpha_i = 1/d_{i,s_i} \tag{3.5}$$

la profondità stimata inoltre può essere scritta come:

$$\hat{d}_{i,s_i} = x_{i,s_i}^T \theta_r \tag{3.6}$$

e finalmente il nostro errore frazionario risulta:

$$\frac{\hat{d}_{i,s_i} - d_{i,s_i}}{d_{i,s_i}} = \frac{1}{d_{i,s_i}}(\hat{d}_{i,s_i}) - 1 = R_{i,s_i}^T \alpha_i(x_{i,s_i}^T, \theta_r) - 1 \tag{3.7}$$

Fatto questo, per minimizzare l'errore su tutti i punti del superpixel in esame, modelliamo la relazione tra parametri planimetrici e feature dell'immagine come:

$$f_1(\alpha_i|X_i, v_i, R_i; \theta) = exp\left(-\sum_{s_i}^{S_i} v_{i,s_i}|R_{i,s_i}^T \alpha_i(x_{i,s_i}^T) - 1|\right) \tag{3.8}$$

I θ_r rappresentano parametri del modello, per ogni riga dell'immagine usiamo un differente valore di questo parametro, dato che le diverse righe presentano andamenti statistici diversi. Nel caso poi che le features calcolate per un determinato punto non risultino attendibili, il parametro v_{i,s_i} viene posto a 0, andando ad annullare l'effetto del termine $|R_{i,s_i}^T \alpha_i(x_{i,s_i}^T) - 1|$. Il secondo termine, $f_2(.)$, modella le relazioni tra i parametri planimetrici di due superpixel differenti, i e j. Esso usa due punti s_i ed s_j secondo la formula:

$$f_2(.) = \prod_{s_i, s_j \in N} h_{s_i,s_j}(.) \tag{3.9}$$

Con differenti scelte di $h(.)e\{s_i, s_j\}$ cattureremo coplanarità colinearità e connettività.

3.6.1 Vincolo di connettività

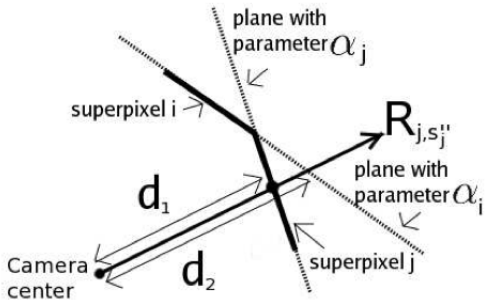

FIGURA 3.9: Vincolo di connessione

Rafforziamo inoltre il vincolo sulla connettività scegliendo s_i e s_j in modo tale che essi siano posti in prossimità di uno spigolo comune (detto edgel) dei rispettivi superpixel, penalizzando la distanza relativa tra i due punti. Così facendo otteniamo un vincolo di connessione tra superpixel diversi, secondo la formula (3.10)

$$h_{s_i,s_j}(\alpha_i, \alpha_j, y_{ij}, R_i, R_j) = exp(-y_{ij}|(R^T_{is_i}\alpha_i - R^T_{js_j}\alpha_j)\hat{d}|) \tag{3.10}$$

In dettaglio, $R^T_{is_i}\alpha_i = 1/d_{i,s_i}$ e $R^T_{js_j}\alpha_j = 1/d_{j,s_j}$. Dunque, il termine $R^T_{is_i}\alpha_i - R^T_{js_j}\alpha_j)\hat{d}$ rivela la distanza frazionale $|(d_{i,s_i} - d_{j,s_j})/\sqrt{d_{i,s_i}d_{j,s_j}}|$ per la distanza stimata $\hat{d} = \sqrt{d_{s_i}d_{s_j}}$. Va notato come in caso di occlusioni la variabile $y_{i,j}$ venga posta uguale a 0, evitando così di forzare la connessione dei superpixel con punti adiacenti.

3.6.2 Vincolo di coplanarità

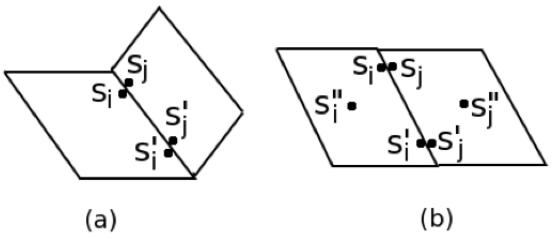

FIGURA 3.10: Vincolo di coplanarita'

Per rafforzare il vincolo di coplanarità tra superpixel adiacenti scegliamo due coppie di punti di confine come appena visto, ed una terza coppia di punti s_i'' e s_j'', questa volta punti interni dei rispettivi superpixel. Penalizziamo dunque la distanza relativa del punto s_j'' dal piano del superpixel i-esimo, calcolato attraverso l'uso del suo raggio normale.

$$h''_{s_j}(\alpha_i, \alpha_j, y_{ij}, R_j, s''_j) = exp\,(-y_{ij}|(R^T_{i,s''_i}\alpha_i - R^T_{j,s''_j}\alpha_j)\hat{d}_{s''_j}|) \tag{3.11}$$

3.6.3 Vincolo di colinearità

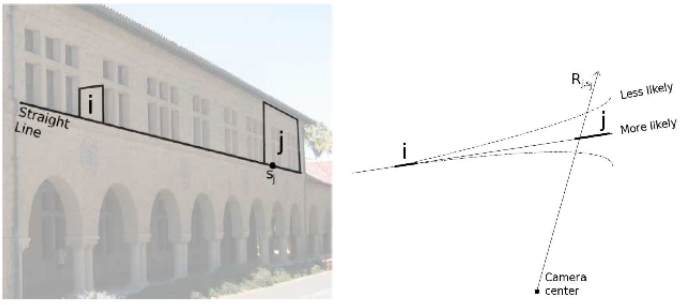

FIGURA 3.11: Concetto di colinearita'.

Consideriamo due superpixel i e j, che risiedono su uno stesso segmento retto nell'immagine 2D: vi è un'infinità di linee curve nello spazio tridimensionale che si proiettano nello spazio bidimensionale dell'immagine come tale segmento rettilineo, anche se statisticamente risulta più probabile che tale proiezione sia data da una curva che risulta essere anche essa rettilinea nel 3D. Allo stesso modo, siamo portati a penalizzare dunque le soluzioni che presentano linee curve in presenza di segmenti rettilinei nell'immagine 2D, discostandosi così da quella che reputiamo

essere la soluzione ottima. Definiamo dunque α_i e α_j i parametri planimetrici di due superpixel i e j che risultano proiettarsi su segmento retto nell'immagine bidimensionale: per un punto s_j appartenente al superpixel j-esimo, penalizziamo la distanza frazionale lungo il raggio R_j dalla linea retta passante per lo stesso superpixel. Secondo la formula:

$$h_{s_j}(\alpha_i, \alpha_j, y_{ij}, R_j, s_j) = exp\,(-y_{ij}|(R^T_{i,s_i}\alpha_i - R^T_{j,s_j}\alpha_j)\hat{d}|) \tag{3.12}$$

abbiamo inoltre che $h_{s_i,s_j}(.) = h_{s_i}h_{s_j}$. Nel dettaglio, $R^T_{j,s_j}\alpha_j = 1/dj, s_j$, $R^T_{j,s_j}\alpha_i = 1/d'j, s_j$. Quindi il termine $(R^T_{i,s_i}\alpha_i - R^T_{j,s_j}\alpha_j)\hat{d}$ da la distanza frazionale $|(d_{j,s_j} - d'_{j,s_j})/\sqrt{d_{j,s_j}d'_{j,s_j}}|$ per $\hat{d} = \sqrt{\hat{d}_{j,s_j}\hat{d}'_{j,s_j}}$ La confidenza che diamo al vincolo è espressa dal termine y_{ij} e dipende dalla lunghezza della linea e dalla sua curvatura: una linea retta e longilinea nell'immagine bidimensionale ha probabilità maggiore di rappresentare a sua volta una linea retta nel 3D.

3.7 Il modello probabilistico

Come già ricordato la profondità di una specifica patch dipende dalla stima della profondità assoluta calcolata sulle caratteristiche della patch stessa, ma anche dalla profondità relativa alle patch adiacenti. Due patch appartenenti allo stesso oggetto, come ad esempio un edificio, avranno profondità fortemente correlata. Per fare questo viene utilizzato un modello Markoviano di stima. Allo stesso modo del calcolo delle features, anche la stima della depth avviene su differenti scale: questo per ovviare al problema di patch che a livello microscopico non sembrano correlate, ma invece lo sono a livello macroscopico. Si pensi ad un edificio contentente delle finestre: pur non trovando correlazione tra piccole patch in prossimità della finestra. Possiamo comunque individuare la continuità presente nella struttura se analizziamo punti maggiormente distanti con patch di dimensioni maggiori impedendo così un'analisi sfalsata dalle discontinuità presenti nella struttura. Definiamo con s= 1,2,3 ciascuna delle 3 scale, otteniamo:

$$d_i(s+1) = \frac{1}{5}\sum_{j \in N_s(i) \cup \{i\}} d_j(s) \tag{3.13}$$

dove detta d_i la depth della patch i-esima, con $N_s(i)$ indichiamo le 4 patch vicine alla patch i-esima alla scala s. Così facendo le patch ad una scala superiore sono forzate ad essere una media delle patch ad una scala di dimensione inferiore. Il modello sulle profondità risulta il seguente

$$P(d \Big| X;\theta,\sigma) = \frac{1}{Z}\left(-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}\right) \quad (3.14)$$

Dove, con M si indica il numero totale di patch, x_i indica il vettore della profondità assoluta della patch i-esima, θ e σ sono parametri del modello. Si usano differenti parametri, dato che, con una telecamera montata orizzontalmente, ogni riga presenta differenti proprietà statistiche. Infine Z rappresenta una costante di normalizzazione.

3.8 Finalità dell'algoritmo

I campi di utilizzo risultano dunque molteplici. Grazie proprio alla generalizzazione fatta risulta possibile l'impiego dell'algoritmo per finalità diverse. Alcuni obiettivi che possono essere raggiunti sono:

- predizione della profondità a partire da una singola immagine
- incorporare sull'immagine le informazioni rilevate dai dettagli monocolari, in modo da poter permettere sia la ricostruzione della profondità monocolare indipendente, sia quella stereo coadiuvata da questa prima analisi sulle singole immagini.
- creare modelli 3D navigabili, garantendo una ricostruzione piacevole del modello.
- creare grossi modelli 3D anche partendo da un esiguo numero di immagini disponibili
- permettere la guida autonoma di una macchina radiocomandata

Questi sono ovviamente solo alcuni esempi, la duttilità dell'algoritmo ne permette l'applicazione in tutti quei settori dove si presentano problematiche con la ricostruzione 3D classica, oltre a quei campi dove è auspicabile poter incrementare le prestazioni raggiunte con i modelli standard.

3.9 Possibili sviluppi futuri

Pur presentando ottimi risultati nell'ambito in cui si colloca, l'algoritmo è ben lontano dall'essere perfetto. Presenta un ottimo andamento nel caso di paesaggi, mentre riscontra difficoltà nel caso di strutture architettoniche o in presenza di oggetti in primo piano. Un possibile miglioramento dell'algoritmo si otterrebbe introducendo il riconoscimento di alcune forme tramite l'uso di alcune librerie contenenti modelli di riferimento: potendo ad esempio riconoscere una sagoma umana sarebbe semplice dedurne la profondità di campo in base alle dimensioni. Altri possibili metodi volti a migliorare le prestazioni attuali vengono poi presentati nel capitolo successivo, come l'uso dei filtri sobeliani in combinazione della trasformata di Hough, o la segmentazione coadiuvata dalle informazioni sul colore. Qualora disponibili, si potrebbe inoltre sfruttare le informazioni EXIF presenti in numerosi file .jpeg, come nelle stesse immagini test utilizzate nel lavoro di Saxena. Lunghezza focale ed apertura del diaframma (F) possono essere utilizzate in qualche modo per stimare delle proprietà dell'immagine, in combinazione con un'analisi, tramite opportuno filtraggio, delle zone a fuoco e delle zone sfuocate nell'immagine. Pensando poi di poter disporre sempre di una sola telecamera, ma potendo prelevare frame diversi in un certo intervallo temporale, potremmo ricondurci al caso di *shape from motion*, utilizzando Make 3D per una prima stima del modello, velocizzando e migliorando anche il processo di acquisizione, rendendo necessario un minor numero di immagini prima di poter stimare un modello veritiero, e andando ad irrobustire la tecnica di predizione.

Capitolo 4

Make 3D: il codice e le modifiche apportate

4.1 Introduzione

Il codice, scritto in MATLAB, e scaricabile gratuitamente [23], contiene diversi script e funzioni annidate (richiamate dall'esecuzione dello script principale '`Oneshot3DEfficient.m`') ognuna con un diverso compito computazionale sui dati in ingresso (la nostra immagine). Sempre sul sito sono disponibili diverse immagini test, a risoluzione 1704x2272, e le relative mappe di profondità, ricavate tramite scansione laser della scena. Per rendere l'algoritmo applicabile anche in condizione di diversa risoluzione, si è provveduto ad un riscalamento dell'immagine iniziale, in modo da poter esser testato con qualsiasi tipo di immagine a nostra disposizione. Va notato come, in caso di riscalamento, introduciamo un errore sull'immagine stessa, dato che essa subisce degli stiramenti. In uscita all'algoritmo otteniamo un file .wrl, o, in seguito alle modifiche apportate, un file .ply. Entrambi sono estensioni che rappresentano semplici file di testo con le informazioni per la grafica vettoriale. Abbiamo dunque un poligono (solitamente viene usato un collage di triangoli) che descrive la nostra approssimazione della superficie tridimensionale stimata dall'immagine 2D. Con questi file mesh è possibile riportare vertici, spigoli, colore della superficie, texture,brillantezza e trasparenza .Essi sono visualizzabili con qualsiasi software per la manipolazione di modelli 3D. Per la facilità di utilizzo e la capacità di riconoscere diversi formati quasi sempre si è utilizzato l'open source MeshLab [29], che permette di navigare all'interno del modello.

4.2 Lettura e ridimensionamento

L'immagine di input viene letta da MATLAB, e salvata come matrice RGB, dove ogni colore è rappresentato da una matrice singola di dimensioni equivalenti a quelle dell'immagine, con i valori del colore per ciascun pixel, nell'intervallo 0-255. Come già accennato, prima di compiere qualsiasi operazione, si è provveduto ad eseguire un casting delle dimensioni, attraverso il comando `imresize`, garantendo così la corretta elaborazione della nostra immagine. La dimensione corretta dell'immagine dunque risulta pari a 900x1200 pixel, ridimensionamento congruo con quanto accade in fase di elaborazione per le immagini campione fornite assieme al codice. L'immagine così manipolata viene quindi salvata in formato .jpg, in modo da poter esser richiamata anche nelle funzioni annidate, qualora fosse necessaria.

4.3 Riconoscimento delle linee

Make 3D, dopo queste operazioni preliminari di lettura, procede immediatamente con la segmentazione dell'immagine. Al fine di migliorare le prestazioni però, si è subito cercato di ricavare delle informazioni preliminari basandosi unicamente su calcoli elementari eseguiti sull'immagine, da utilizzare poi in una seconda fase, assieme ai risultati del corpo vero e proprio di Make 3D. In risposta a ciò si è deciso di utilizzare una matrice riportante la media dei valori di colore sui tre canali (una sorta di immagine in bianco e nero, recante informazioni sulle diverse zone presenti nella scena), e di servirsi della trasformata di Hough descritta nel paragrafo successivo.

FIGURA 4.1: Immagine con valore medio di colore.

4.3.1 Trasformata di Hough

La trasformata di Hough è una tecnica di estrazione utilizzata nel campo dell'elaborazione digitale delle immagini. Nella sua forma classica si basa sul riconoscimento delle linee di un'immagine, ma è stata estesa anche al riconoscimento di altre forme arbitrariamente definite. Fu scoperta da Paul Hough nel 1959, anche se la forma universalmente riconosciuta ed utilizzata al giorno d' oggi è quella rivista da Richard Duda e Peter Hart nel 1972. È una tecnica che permette di riconoscere particolari configurazioni di punti presenti nell'immagine, come segmenti, curve o altre forme prefissate, rappresentando un tipico operatore globale. Il principio fondamentale è che la forma cercata può essere espressa tramite una funzione nota che fa uso di un insieme di parametri. Una particolare istanza della forma cercata è quindi completamente precisata dal valore assunto dall'insieme di parametri. Per esempio, assumendo come rappresentazione della retta la forma $y = ax + b$, qualunque retta è completamente specificata dal valore dei parametri (a, b). Se si assume un tipo di rappresentazione diversa, quale la forma normale polare $\rho = xcos\theta + ysin\theta$, l'insieme di parametri varia di conseguenza. In questo caso la retta è completamente specificata dalla coppia (ρ, θ).

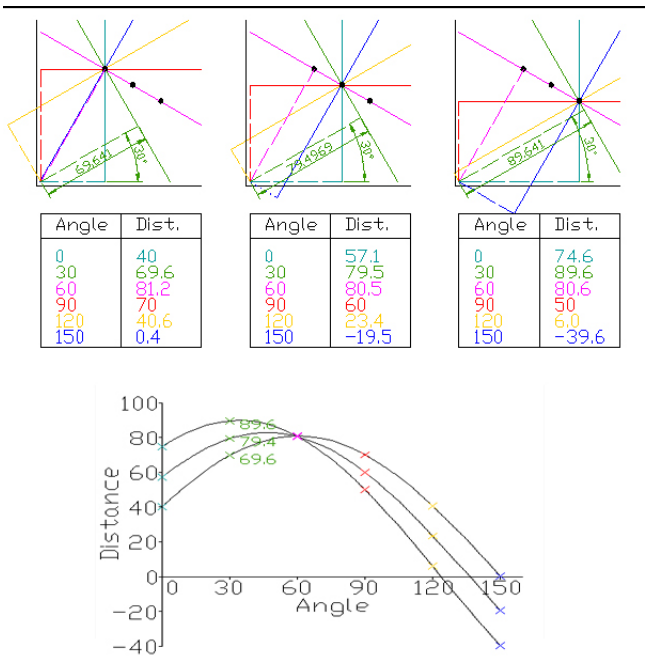

FIGURA 4.2: Per ogni punto sono disegnate in alto le possibili linee che lo attraversano, insieme alle perpendicolari relative che intersecano l' origine. Di ciascuna di queste perpendicolari viene misurata la lunghezza e l' angolo. Tali valori vengono riportati nel grafo di Hough

Fissata la forma di interesse (p.es. il segmento di retta) e la sua rappresentazione (p.es. la forma polare), è possibile considerare una trasformazione dal piano dell'immagine (su cui la forma è rappresentata) allo spazio dei parametri. In questo modo, una particolare istanza di retta viene rappresentata da un punto nello spazio dei parametri. Nel piano dell'immagine, un punto è identificato dall'intersezione di più rette. Quindi, ad ogni punto P corrisponde, nel piano dei parametri, la curva formata dai punti immagine delle rette passanti per P. Se nell'immagine sono presenti dei punti allineati su una stessa retta, sul piano dei parametri, le curve che corrispondono alle trasformazioni dei vari punti si intersecano in un punto del piano trasformato che è l'immagine della retta su cui giacciono i punti. In questo modo, è possibile individuare i segmenti di retta presenti sull'immagine originale. L'approccio è robusto al rumore e ad eventuali interruzioni che dovessero essere presenti sul segmento nell'immagine. La trasformata di Hough è stata calcolata con l'opportuna funzione MATLAB, facendo attenzione a filtrare prima l'immagine attraverso un filtro sobeliano, che altro non è che un filtro che calcola il gradiente dell'immagine, restituendo il valore 1 in presenza di picchi di alta frequenza, e 0 altrimenti: Detta I la nostra immagine, G_x il gradiente orizzontale, e G_y il gradiente verticale, questi ultimi sono calcolati attraverso la convoluzione per le matrici:

$$\mathrm{G_x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathrm{I} \tag{4.1}$$

$$\mathrm{G_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathrm{I} \tag{4.2}$$

L ingresso della trasformata di Hough dunque è rappresentato dal sobeliano successivamente binarizzato. L' immagine di ingresso così ottenuta è stata salvata in modo da essere disponibile per eventuali elaborazioni successive. Nel calcolo della trasformata di Houghè possibile stabilire la soglia di tolleranza, per rilevare i punti che rappresentano le linee con un forte gradiente, oltre al numero massimo e alla lunghezza minima delle linee da prendere in considerazione.

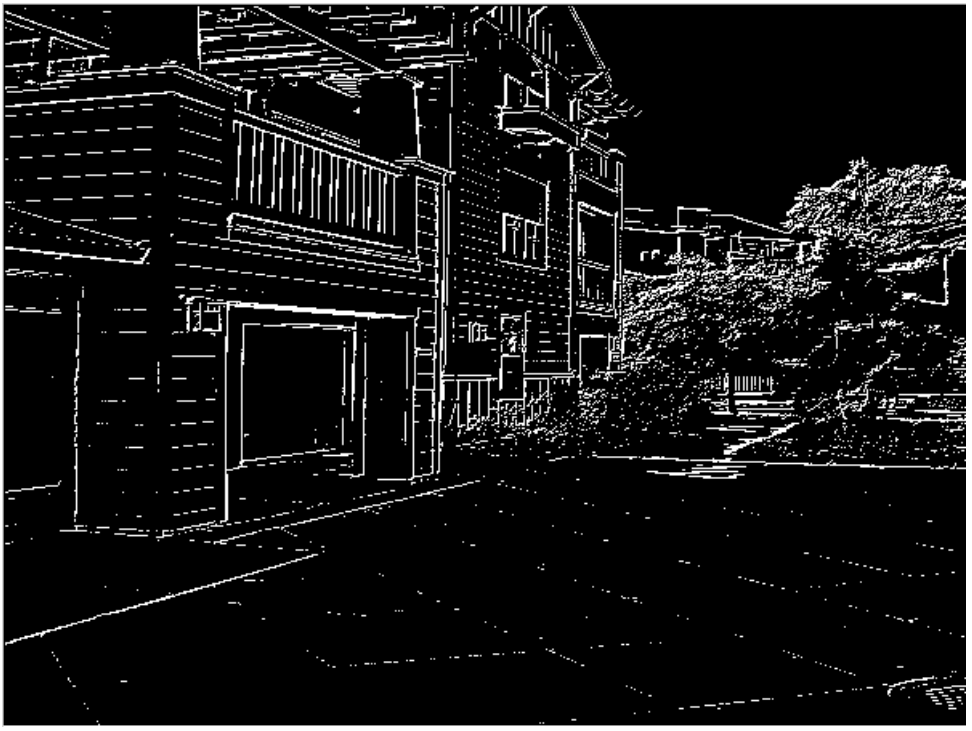

FIGURA 4.3: Uscita del filtro Sobeliano.

Nella figura sopra viene rappresentato il calcolo del filtro sobeliano, facendo riferimento sempre alla stessa immagine di partenza in figura (4.1).

Va ricordato come ogni singola immagine richiederebbe dei parametri configurati ad hoc, che rispecchino la natura stessa dell'immagine. Oltre a questo va sottolineato come, in caso di immagini "difficili", con forti gradienti in presenza anche di superifici continue, il calcolo della trasformata di Hough potrebbe non portar alcun vantaggio, o addirittura potrebbe forviare la stima.

In figura (4.4) è illustrato l'andamento nel piano (ρ, θ) dei punti corrispondenti alle linee passanti per i punti dell'immagine, in presenza di un punto di accumulazione abbiamo un houghpoint, segnato da un quadratino nero. A questo corrisponde ovviamente un segmento. Possiamo sovraporre l'immagine iniziale alla nostra stima delle linee di Hough, verificandone così la correttezza.

È facile notare come per la prima immagine a sinistra, le linee risultino sovrabbondanti, andando a delineare confini che non sempre rappresentano discontinuità. Per l'immagine a destra invece si percepisce subito l'informazione utile che reca il calcolo della trasformata di Hough, che va a delimitare zone appartenenti a piani prospettici differenti. In generale quindi risulterebbe controproducente utilizzare questa stima per una divisione dei superpixel (informazione forte). D'altro canto

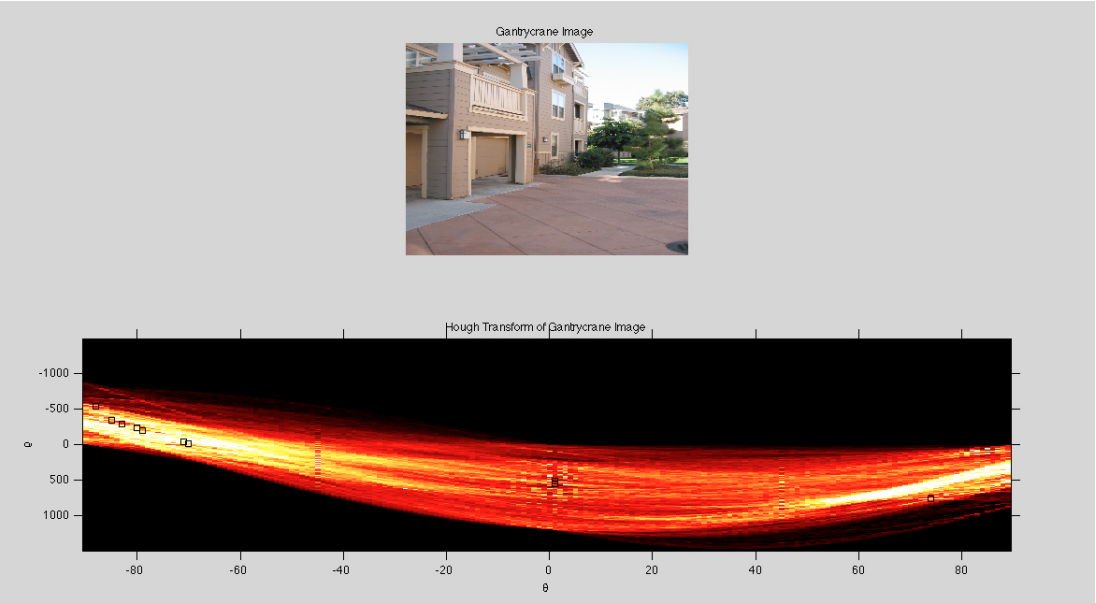

FIGURA 4.4: Mappatura della trasformata di Hough.

FIGURA 4.5: houghlines riportate nel piano di riferimento dell'immagine.

possiamo sfruttare quanto visto considerando le linee di Hough come informazione debole: aggregando cioè segmenti che giacciono su una stessa linea parallela alla linea di Hough. Non ricorreremo dunque ad un processo di segmentazione, bensì ad un processo di fusione, utile soprattutto nel caso di segmentazione fine dei superpixel, ove la divisione è spesso sovradimensionata alle caratteristiche reali dell'immagine.

4.4 Generazione dei superpixel

La segmentazione come già ricordato avviene a diversi livelli. Sono previsti tre diversi tipi di segmentazione: la prima, fine, si occupa di segmentare l'immagine in una moltitudine di superpixel, e serve a tener conto delle variazioni a livello

microscopico, oppure di oggetti relativamente piccoli, quali una finestra od un arco vista in lontananza, che risulterebbero appiattiti da una segmentazione più grossolana. La segmentazione media si occupa invece di rendere l'andamento generale dell'insieme, tralasciando i dettagli ma tenendo ancora conto di variazioni locali, ed una segmentazione macroscopica, che ritorna le informazioni sulle macrozone, e, accoppiata alle altre due, può rivelarsi indispensabile a definire quali sono le vere zone che presentano un cambiamento forte in termini di profondità ed inclinazione. L'algoritmo è stato modificato in modo da visualizzare per ognuno dei 3 tipi di divisione la relativa segmentazione, assegnando dei valori casuali di colore ai superpixel. In caso di errori evidenti è possibile rifiutare la segmentazione ottenuta e ripeterla variando alcuni parametri, quali dimensione minima della segmentazione ed omogeneità .

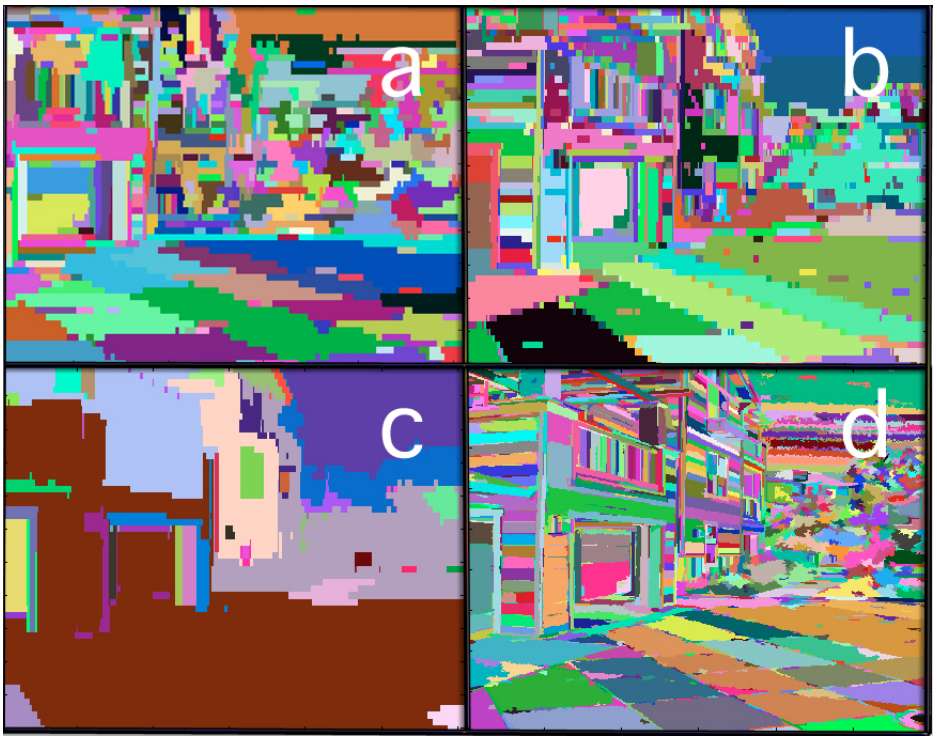

FIGURA 4.6: Generazione dei superpixel: (a) fine, (b) media, (c) grossolana. Il riquadro (d) invece rappresenta la media dei precedenti a risoluzione elevata.

Eseguito questo processo di supersegmentazione iniziale si prosegue attraverso il calcolo delle features dell'immagine, in relazione alle patch calcolate: vengono a questo scopo usati i 17 filtri presentati nel capitolo 3: maschere di Law, canali colore e gradienti relativi alla texture. Inoltre sono presenti delle funzioni che cercano di ripulire, specialmente le segmentazioni più piccole, attraverso l'uso di maschere, con l'intento di identificare zone di cielo o di terreno piatto, in modo da eliminare il più possibile le segmentazioni inutili, sfruttando le conoscenze a

priori della natura delle immagini. È stato inoltre inserita un'ulteriore maschera, in modo da rafforzare il comportamento nel caso in cui l'immagine catturi anche una parte di cielo sullo sfondo. Analizzando esclusivamente la parte superiore dell'immagine e andando ad identificare i pixel contigui che presentano un range di colori plausibile dell'azzurro, è possibile identificare i pixel rappresentanti porzioni di cielo. Particolare attenzione è stata posta nell'evitare di includere pixel che presentino discontinuità troppo accentuate o gradienti ad alte frequenze, in modo da minimizzare la probabilità di includere oggetti non appartenenti allo sfondo voluto. I voxel corrispondenti a tale maschera sono stati infine posti tutti alla distanza massima rilevata nella matrice di profondità

4.5 Calcolo dell'errore

Ponendo come obbiettivo quello di andare ad incrementare in termini prestazionali l'algoritmo, si è posta la necessità di confrontare i dati stimati con quelli reali, in modo da poter sia analizzare l'efficenza dell'algoritmo con diverse immagini, sia poter confrontare i risultati in seguito alle modifiche apportate al codice. I dati disponibili, frutto della scansione laser della scena, non riportano unità di misura, seppur risultino proporzionali ai dati stimati dall'algoritmo stesso. Si è reso dunque necessaria una conversione dei dati laser, per renderli confrontabili con i valori ottenuti. A tale scopo si è utilizzato un'algoritmo ICP. L'ICP (Iterative Closest Point) è utilizzato spesso nella ricostruzione 3D, per poter confrontare dati provenienti da differenti scansioni di una stessa scena. Iterativamente, sulla nuvola di dati tridimensionali (X,Y,Z) si procede come segue:

1. Si trova una corrispondenza tra le due superfici (mapping di punti, superfici, linee, curve)
2. Si calcola la distanza tra le due superfici con il metodo dei minimi quadrati
3. Si calcolare la trasformazione che minimizza questa distanza
4. Si effettua la trasformazione e si reitera la procedura finché la distanza non sia minore di un threshold, o finchè non si è raggiunto il numero massimo di iterazioni.

L'algoritmo sostanzialmente trova la traslazione e rotazione dei dati che minimizza l'errore tra i due dataset, oltre all'errore quadratico medio per ciascuna iterazione.

4.6 Cancellazione dei picchi

Un evidente limite dell'algoritmo Make 3D risulta certamente l'introduzione di picchi di profondità in realtà inesistenti, portando piccoli oggetti su un piano molto più vicino o lontano rispetto a quello reale. Si nota infatti come spesso accada che un numero limitato di punti sia affetto da un forte errore, o per la valutazione errata su un oggetto di piccole dimensioni in realtà appartenente allo stesso piano dell'oggetto in cui è inserito, oppure per un'errata connessione tra punti adiacenti. Si è dunque provveduto a risolvere questo tipo di problematica andando ad analizzare i risultati di profondità ottenuti ed eliminando le discontinuità spurie ove queste plausibilmente non esistano.

A tal fine è stata introdotta un'ulteriore segmentazione dell'immagine, ad un livello intermedio diverso dai precedenti, che descriva in modo piuttosto preciso le diverse zone, senza perdere l'informazione di segmentazione globale dei macroblocchi. I valori usati per questa suddivisione sono di solito intermedi tra quelli delle due segmentazioni maggiori usate nell'algoritmo. Questa suddivisione in blocchi è poi raffinata attraverso l'uso dell'informazione di colore: un'ulteriore scissione in due blocchi distinti è eseguita qualora siano presenti due tonalità distinte di colore, una fusione invece qualora due segmenti presentino colore univoco. Ad ogni segmento viene dunque assegnato un valore numerico, comune a tutti i pixel facente parte della stessa zona. Per ciascun superpixel viene calcolata la media e la varianza dei valori di profondità corrispondenti. Se sono presenti valori numericamente non coerenti con i valori adiacenti, questi vengono ignorati. A questo punto viene eseguita la regressione lineare tra valori di coordinate XY e i valori numerici di profondità Z. Vengono così trovati i coefficenti di regressione lineare attraverso i quali possiamo ricalcolare sulla zona i valori di profondità. Questo appiattimento dei valori potrebbe sembrare penalizzante, dato che teoreticamente potremmo avere una perdita di dettaglio, che però risulta trascurabile se realizzato facendo attenzione alla segmentazione scelta per la suddivisione. Da calcoli eseguiti su un set ampio di immagini otteniamo un miglioramento delle prestazioni, sia in termini di errore numerico, sia in termini di errore visivo, ottenendo ottimi risultati in termini di attenuazione dei picchi.

Capitolo 5

Conclusioni e risultati

5.1 Introduzione

Negli ultimi anni, le metodologie basate sulla visione stereo e sulla triangolazione, sono state applicate a diversi problemi, quali la navigazione di automi, la ricostruzione 3D di strutture architettoniche, oppure il riconoscimento di oggetti, solo per citarne alcuni. Diversamente dagli algoritmi basati sulla visione stereoscopica o sulla *structure from motion*, questa tesi ha preso in esame e sviluppato concetti attorno agli algoritmi che sfruttano un ampio insieme di indizi monocolari. Abbiamo quindi presentato un modello basato su MRF multiscala in grado di stimare la profondità a partire dalla semplice analisi di una singola immagine. Queste informazioni non solo possono essere integrate con quelle di triangolazione classiche, bensì permettono una duttilità di utilizzo altrimenti inimmaginabile in diversi contesti.

5.2 Confronto dell'errore per Make3D con algoritmi simili

I risultati ottenuti, vanno analizzati e confrontati, in termine di errore, all'interno della stessa classe di algoritmi. Il lavoro di Saxena dunque è stato confrontato con altri esempi algoritmi di stima tridimensionale da singola immagine. In particolare si sono considerati:

- Il lavoro di Hoiem et. al. (HEH) [19];
- SCN, un algoritmo precedentemente formulato dagli stessi autori di Make3D [30];
- Point Wise MRF (BaseLine 1) senza vincoli di coplanarità colinearità connettività [16];
- Point Wise MRF (BaseLine 1) con vincoli di coplanarità colinearità connettività [16];
- Planar MRF (BaseLine 2) senza vincoli [16];
- Planar MRF (BaseLine 2) con il solo vincolo di coplanarità [16];
- L'algoritmo completo Make3D [16].

E si è valutato l'errore in diversi modi:

- Percentuale di modelli corretti rilevati;
- Percentuale di singoli piani (profondità e inclinazione) correttamente rilevati;
- errore percentuale in scala logaritmica $|logd - log\hat{d}|$;
- errore relativo percentuale medio $\frac{|d - \hat{d}|}{d}$.

Per il calcolo è stato utilizzato un dataset di 107 immagini differenti, scelte, come riportato negli articoli, da terze persone esterne al lavoro svolto.

Nel caso dell'algoritmo di HEH i dati sono stati prima scalati e ruotati, per poter esser confrontati con i dati di profondità ricavati tramite la scansione laser della

Tabella 5.1: Tabella di confronto dell'errore rilevato

Metodo	modelli corretti	piani corretti	errore log	errore %
SCN	NA	NA	0.198	0.530
HEH	33.1%	50.3%	0.320	1.423
BS1 senza vinc.	0%	NA	0.300	0.698
BS1 con vincolo	23%	NA	0.149	0.458
BS2 senza vinc.	0%	0%	0.334	0.516
BS2 con coplan.	45.7%	57.1%	0.191	0.373
BS2 con vincolo	**64.9**%	**71.2**%	0.187	**0.370**

scena. Il laser utilizzato è definito come autocostruito, è stato quindi impossibile ricavarne i parametri di funzionamento dai datasheet, rendendo necessario per il confronto dell'algoritmo modificato l'applicazione dell'algoritmo ICP, indistintamente sia sui dati di Make3D, sia sui dati da noi ottenuti. In tabella sono riportati i valori stimati nel lavoro di Saxena per i diversi metodi.

Da questa si nota facilmente come sia l'utilizzo dell'algoritmo Point-wise MRF sia Plane Parameter MRF superano SCN ed HEH nel predirre la profondità per quanto riguarda l'accuratezza da un punto di vista quantitativo. Va messo in risalto come passare da un algoritmo Point based ad uno Plane based contribuisca molto all'accuratezza relativa della profondità, producendo depthmap più definite. L'introduzione delle proprietà di coplanarità, connettività e colinearità, aumenta prestazionalmente l'algoritmo, oltre a migliorare significatamente i modelli 3D prodotti, andando ad eliminare quelle discontinuità che, pur producendo un lieve errore quantitativo, risultano inaccettabili per il modello qualitativo.

5.3 Confronto dell'errore dopo le modifiche

Una volta apportate le modifiche descritte nel capitolo 4 al codice, il confronto quantitativo sull'errore è stato eseguito direttamente tra i valori ottenuti e quelli disponibili sul sito, frutto della scansione laser, per il dataset di immagini disponibili. Si è potuto così confrontare l'errore del codice originale con quello ottenuto in seguito alle variazioni descritte. Per una migliore visione d'insieme si è deciso di riportare l'errore percentuale globale $|d - \hat{d_{norm}}|$ e l'errore MSE (Mean Square Error), secondo il codice riportato:

```
function MSE = MeanSquareError(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);

[M N] = size(origImg);
error = origImg - distImg;
MSE = sum(sum(error .* error)) / (M * N);
```

L'errore è stato valutato per 40 immagini, riportando sempre risultati coerenti, e garantendo dunque una certa robustezza delle modifiche effettuate, nelle tabelle alle pagine seguenti sono riportati i valori ottenuti per alcune di esse.

Viene riportato l'errore prima e dopo la trasformazione dei punti attraverso l'algoritmo ICP. Similmente a quanto riportato in [16] l'errore tipico risulta del 36% per l'algoritmo Make 3D (M3D), mentre troviamo un errore medio del 31% per l'algoritmo da noi modificato, indicato in tabella con la sigla M3D+. In seguito alla trasformazione dei dati, in modo da essere adattati ai valori reali ottenuti tramite scansione, l'errore medio rivelato risulta pari al 31%, per attestarsi su valori del 27-28% in seguito alle modifiche apportate. Va notato come, nelle immagini prese in analisi, non sono stati riscontrati casi in cui le modifiche abbiano portato ad un peggioramento prestazionale. A seconda del tipo di immagine il miglioramento ottenuto oscilla molto: in alcuni casi non vi sono state sostanziali modifiche al modello ottenuto, in altri invece siamo arrivati ad ottenere anche miglioramenti del 7-8%. Il confronto degli MSE invece rivela un sostanziale dimmezzamento dell'errore quadratico. In alcuni (rari) casi, in cui l'errore a monte della trasformazione ICP risultava basso ($< 7\%$), i dati sono stati trascurati, dato che questi portavano ad un'errore nullo una volta trasformati, a causa della limitata precisione computazionale. La causa è da attribuirsi ad immagini con depthmap semplici, prive di strutture complesse, come ad esempio un'edificio in lontananza, in cui anche la scansione laser presenta solamente due piani principali: un'insieme di punti tutti equidistanti all'orizzonte ed un piano riferito alla pavimentazione. Questi dunque non sono stati considerati esempi tipici per gli scopi prefissi e si è ritenuto opportuno tralasciarli nelle stime. Per ogni immagine viene riportata una icona, in modo da sottolineare le diverse prestazioni in rapporto alla diversa natura delle immagini in questione.

Nome immagine	Errore M3D iniziale	Errore M3D+ iniziale	M3D+ vs M3D	Errore M3D finale	Errore M3D+ finale	M3D+ vs M3D	MSE M3D	MSE M3D+	.jpg
Test1	39,81	36,23	-3,58	27,1	24,44	-2,66	748,82	528,16	
Test2	39,10	33,21	-5,89	24,75	18,73	-6,02	964,51	585,16	
Test3	40,38	36,23	-4,15	27,01	25,18	-1,83	791,20	563,82	
Test4	39,12	35,00	-4,12	26,57	24,48	-2,09	841,61	578,11	
Test5	39,06	34,80	-4,26	26,30	25,69	-0,61	867,12	552,22	
Test6	36,83	34,12	-2,71	27,05	25,00	-2,05	643,85	524,89	
Test8	40,35	37,43	-2,92	28,89	23,17	-5,72	912,67	588,94	
Test5	30,48	26,42	-4,06	20,04	17,27	-2,77	618,11	422,51	
Test6	38,39	33,92	-4,47	25,91	25,13	-0,78	833,50	551,19	
Test7	27,20	24,90	-2,30	20,46	19,18	-1,28	423,34	289,98	
Test8	5,12	4,13	-0,99	3,49	3,42	-0,07	274,72	57,94	
Test9	32,56	28,91	-3,65	23,39	22,29	-1,10	593.70	376.05	
Test10	44,52	37,22	-7,30	22,79	15,28	-7,51	1248	1141	
Test11	19.85	14,00	-5,85	13.97	13.25	-0,72	315.34	161.01	
Test12	30.71	27.06	-3,65	21.5	20.63	-0,87	505.19	320.91	
Test13	43.87	37.47	-6,40	24.66	19.81	-4,85	1118,1	822,78	

Nome immagine	Errore M3D iniziale	Errore M3D+ iniziale	M3D+ vs M3D	Errore M3D finale	Errore M3D+ finale	M3D+ vs M3D	MSE M3D	MSE M3D+	.jpg
Test14	54.13	41.16	-12,97	21.39	17.60	-3,79	2274,3	1455,5	
Test15	21.87	17.75	-4,12	13.90	13.05	-0,85	428,01	198.75	
Test16	22.42	19.89	-2,53	17.33	17.09	-0,84	328.51	182.03	
Test17	19.77	14,27	-5,50	15.57	14.26	-1,31	306.67	195.02	
Test18	35,56	30,44	-5,12	22,44	19,32	-3,12	672.65	522.54	
Test19	28.16	21.89	-6,27	15.87	14.34	-1,53	717.93	359.65	
Test20	39,50	32.78	-6,72	21,76	18,25	-3,51	849.25	668.48	
Test21	18.50	14.97	-3,53	12.89	11.67	-1,22	323.42	144.19	
Test22	26.19	22.85	-3,34	18.85	16.86	-1,99	453.03	300.67	
Test23	23.86	20.57	-3,29	16.24	14.84	-1,40	393.07	219.62	
Test24	23.78	19.80	-3,98	16.01	13.59	-2,42	413.56	217.64	
Test25	42.91	39.83	-3,08	31.17	28.14	-3,03	824.51	619.74	

FIGURA 5.1: Confronto dell' errore sulla predizione per l'algoritmo Make 3D prima e dopo le modifiche

5.4 Make3D contro HEH

Per quanto riguarda l'errore qualitativo, cioè l'errore visivo presente nel modello 3D generato, risulta estremamente arduo ricavare dei dati percentuale. Un calcolo quantitativo infatti difficilmente può render conto della bontà del modello: pensiamo a due immagini con lo stesso numero di superfici errate, o con delle discrepanze di omogeneità. In base a dove si collocano e come si collocano tali errori all'interno della scena, possiamo trovare accettabile il primo modello, mentre potremmo trovare visivamente non corretto il secondo. Il lavoro di Saxena è stato fatto valutare dagli utenti, sia in termine di numero di superfici inferite non correttamente, sia in termine di risultato a livello globale. In [16] viene riportata una percentuale di modelli corretti pari al 64.9%, mentre nello stesso articolo la percentuale di modelli corretti per il lavoro di HEH scende al 33.1%. Sempre in [16] viene eseguito un confronto alla pari tra i due algoritmi, facendo valutare a persone estranee al progetto i modelli ottenuti per le stesse immagini. Nel 62.1% dei casi è risultato vincente Make3D, nel 22.1% HEH, nel resto vi è stata una sostanziale parità. Da notare come la percentuale di modelli corretti scenda dal 64.9% al 48.1% per lo stesso algoritmo di riferimento, come riportato in [31]. Sottolineamo ancora una volta come queste stime siano fortemente vincolate a fattori soggettivi e ai diversi parametri di riferimento, oltre al diverso set di immagini prese in considerazione. Per tali valutazioni dunque sarebbe necessario fissare un set unico di immagini eterogenee, e fissare dei parametri ricavabili direttamente dal modello che rispecchino la capacità di proporre una mesh visualmente piacevole, da utilizzare per qualsiasi algoritmo si ponga lo scopo di ricreare modelli 3D della realtà circostante, sia essa frutto di triangolazione, di scansione laser o di indizi monocolari. L'ultimo dato fa riferimento ad un esperimento web su larga scala, dove gli utenti potevano eseguire l'upload delle loro foto e visualizzare il modello 3D generato da esse. Di seguito in figura è riportato un confronto visivo tra modelli di HEH e di Make3D.

Confrontando le immagini in figura (5.1) possiamo vedere le differenze tra i due modelli. A discapito di quanto riportato, notiamo come HEH fornisca quasi sempre un modello meno dettagliato della struttura 3D. Quest'ultima però spesso non risulta la scelta ottima: sebbene a livello quantitativo la stima sia certamente migliore, a livello qualitativo si paga molto, specie in caso di inferenza erronea. Si prenda in esame l'immagine 5: entrambi i modelli risultano sbagliati: HEH porta ad un appiattimento della scena, Make3D invece crea degli spigoli vivi irreali, ancor più fastidiosi a livello visivo.

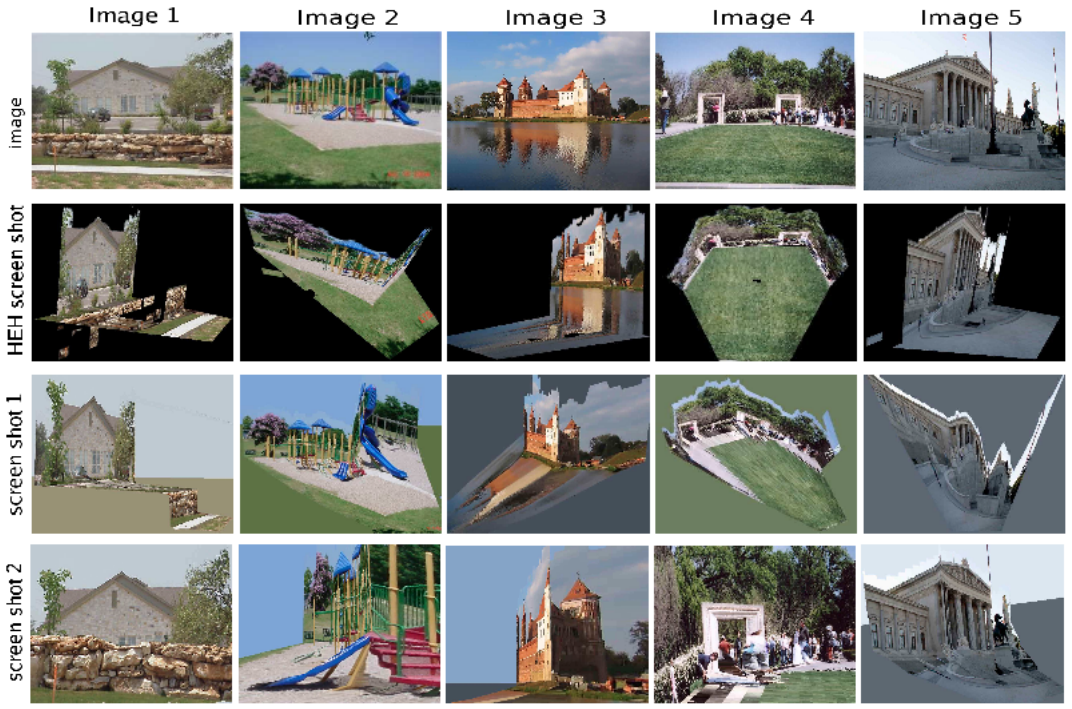

FIGURA 5.2: Confronto tra algoritmo HEH e Make3D: Nella prima riga l'immagine originale, nella seconda il modello predetto da HEH, nella terza e quarta riga prospettive del modello per Make3D

5.5 Miglioramento del livello visivo

Analizzando il comportamento dei diversi algoritmi livello visivo dunque, si è cercato dunque di attenuare la peculiarità della creazione di spigoli vivi non reali tipica di Make3D, senza andar a perdere per questo la sua capacità maggiore nel riconoscimento di piani diversi in una scena rispetto ad HEH. La perdita di dettaglio a livello microscopico risulta del tutto accettabile all'occhio umano, mentre risulta molto fastidioso un andamento spigoloso irreale. Altro fattore essenziale a cui si è cercato di porre rimedio è l'assenza di profondità a lunga distanza: entrambi gli algoritmi tendono a schiacciare gli elementi in secondo piano sullo sfondo, appiattendoli contro il cielo. Ove applicabile dunque si è modificato l'algoritmo in modo da rilevare la presenza di uno sfondo a distanza infinita (il cielo) tramite opportuni controlli di colore ed omogeneità su punti determinati dell'immagine (vicino ai margini superiori sinistro, destro e centrale, oltre ad un punto più vicino al centro dell'immagine). Se alcuni di questi punti hanno caratteristiche confrontabili con quelle che generalmente caratterizzano il cielo in un'immagine, si itera il procedimento ai punti adiacenti, sino ad individuare la totalità dei punti. La

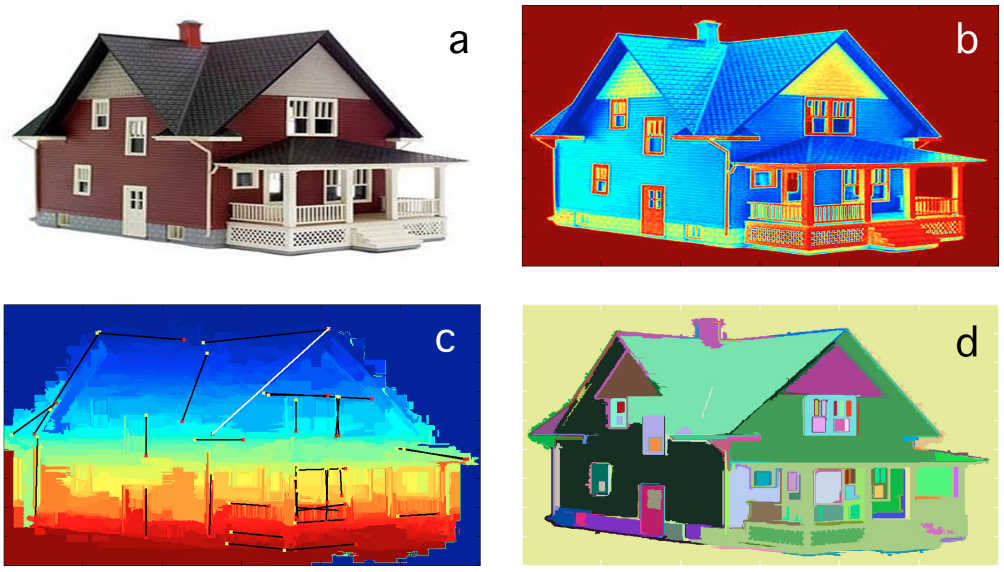

FIGURA 5.3: Alcune informazioni usate nella modifica del codice: (a) immagine originale (b) valore medio di colore (c) trasformata di Hough e houghlines (d) supersegmentazione

maschera ottenuta viene dunque forzata ad assumere per ciascun punto la profondità massima presente nell'immagine, creando un ulteriore livello nel modello, di forte impatto visivo. Il processo è eseguito anche in presenza di sfondo bianco dell'immagine, in modo da poter processare correttamente un numero non esiguo di potenziali immagini frutto di grafica vettoriale, per fare un esempio. A differenza di quanto visto nell'algoritmo originale, i modelli creati dopo la modifica del codice, partono dal presupposto che non sempre a soluzioni quantitativamente migliori corrispondono soluzioni qualitativamente migliori. Se infatti la prima fase delle modifiche apportate al codice si è concentrata sull'ottimizzazione dell'errore numerico, la seconda si è posta come obiettivo quello di ottimizzare la visualizzazione del modello, pur partendo dai dati ottenuti nella prima fase. Analizzando diversi modelli ottenuti sono stati identificati i principali problemi caratteristici dell'algoritmo:

- *Presenza di microzone con orientamento e profondità errata*: piccoli elementi della scena, quali ad esempio le persone interposte tra il piano principale e lo sfondo, spesso vengono correttamente predette in profondità solo per un limitato numero di punti, mentre i rimanenti vengono accomunati alla profondità della scena. Capita così di vedere figure estendersi per un range

di profondità elevato, creando un effetto poco piacevole. Sia tramite rimappamento della profondità con l'uso di una supersegmentazione ad hoc, sia tramite filtraggio, si è cercato di porre rimedio a questo effetto antiestetico, preferendo perdere l'informazione sull'oggetto, a costo di un appiattimento sullo sfondo, trascurabile per oggetti di queste dimensioni;

- *Appiattimento della scena sullo sfondo*: se da un lato questo andamento è stato volutamente cercato per elementi di piccole dimensioni, si è riscontrato che spesso elementi in secondo piano venivano accomunati a punti di distanza infinita. Per ovviare almeno in parte a ciò si è implementata una funzione `findsky.m`, in grado di identificare all'interno di un certo range di valori la maschera del cielo, ponendola così su un piano a se stante.

- *Presenza di picchi locali con profondità errata*: Per alcuni punti o linee all'interno dell'immagine si è notato come talvolta l'algoritmo tendesse, nonostante il vincolo Markoviano, ad inferire una profondità completamente errata, identificandoli come oggetti singoli. Per questo tipo di errori si è ovviato tramite filtraggio della mappa di profondità attraverso la funzione `medfilt2.m`: un filtro di media, un'operazione non lineare spesso usata nell'image processing per ridurre il rumore 'sale e pepe'. Un filtro mediano risulta più efficiente quando l'obiettivo è quello di ridurre il rumore e simultaneamente preservare gli edge. La maschera usata risulta di 5x5 pixel.

- *Errore su elementi all'interno di elementi di dimensione maggiore*: (ad es. una finestra di una casa). In tal caso il problema risulta risolvibile attraverso il filtraggio eseguito tramite la supersegmentazione modificata, che rivela le zone di maggior interesse. Se l'errore però è dato dal fatto che l'elemento sia associato ad una patch sbagliata (l'esempio comune è quello di una porta di un edificio, fusa con la pavimentazione) l'errore persiste, e potrebbe essere teoreticamente rilevabile tramite l'uso della trasformata di Hough con divisione delle supersegmentazioni. Tale procedura, pur portando ad un miglioramento teorico di questo fenomeno, comporterebbe l'introduzione di nuovi errori.

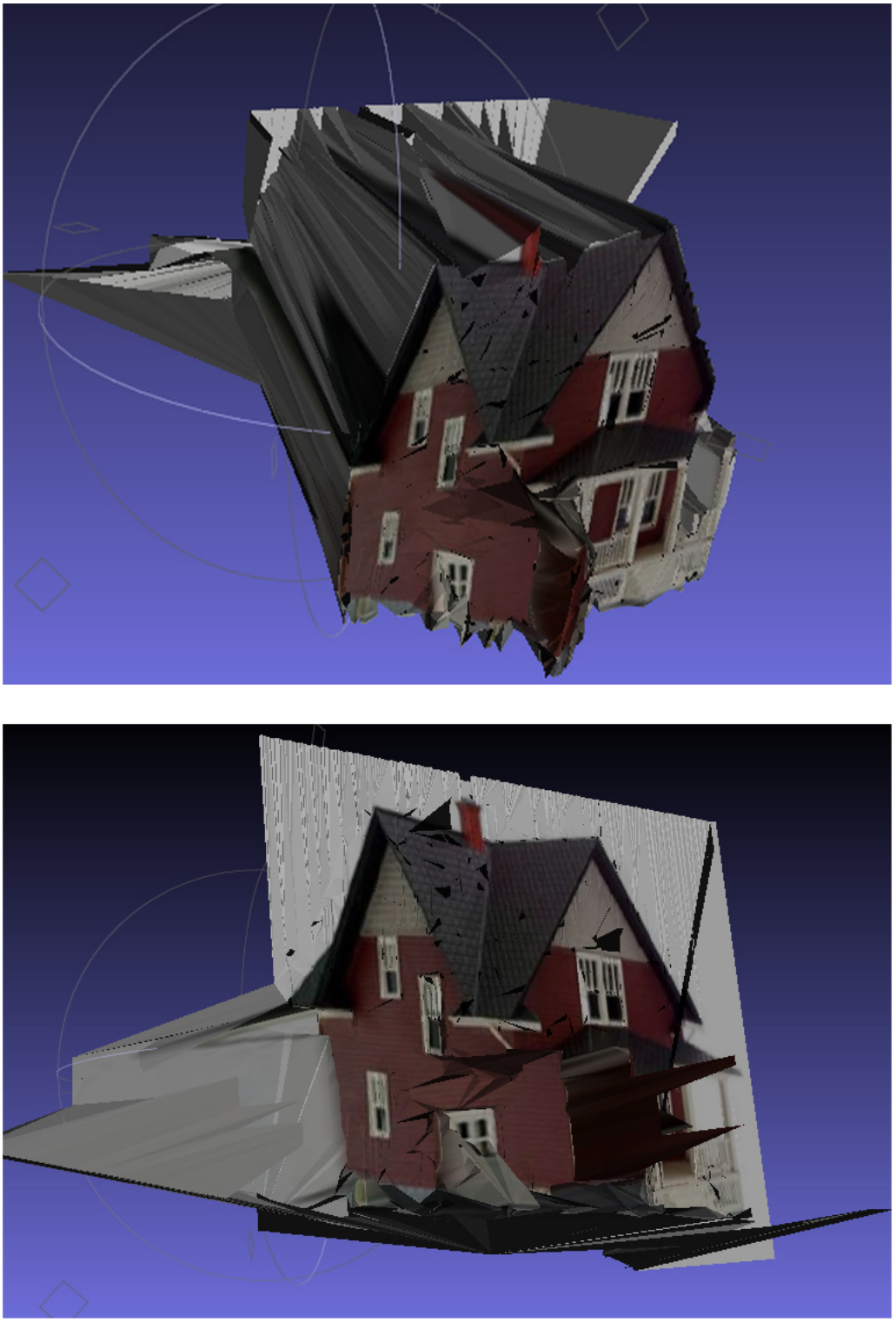

FIGURA 5.4: Confronto tra il nostro modello (sopra) e il modello originale: si nota un netto miglioramento sia come percezione di profondita' sia come eliminazione di picchi di errore

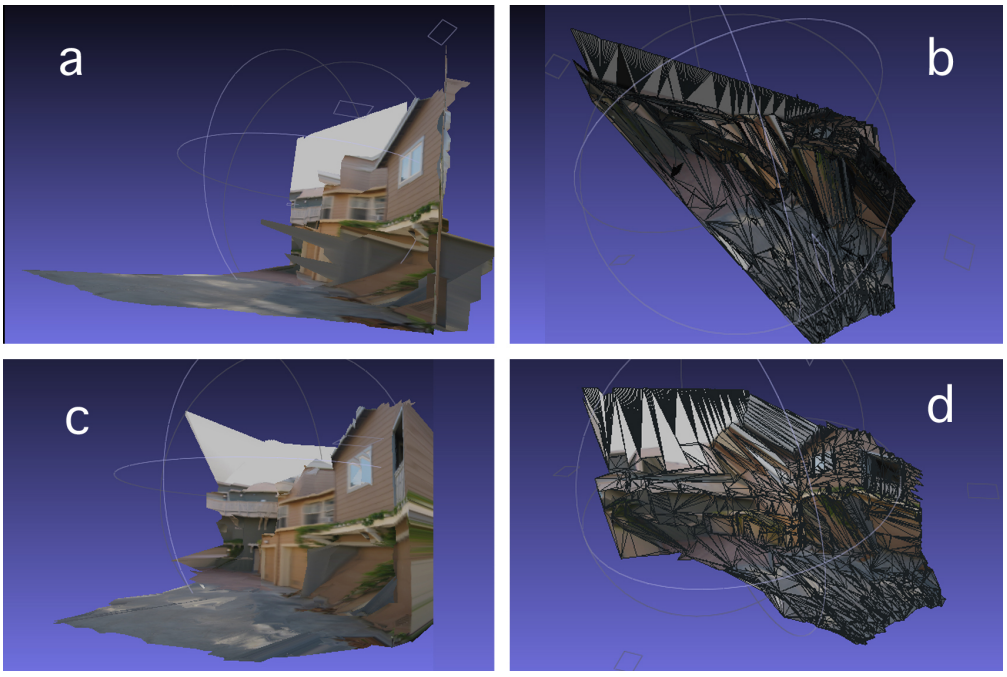

FIGURA 5.5: Confronto tra modelli prima e dopo le modifiche: (a) snapshot del modello originale (b) mesh strutturata del modello originale (c) snapshot del modello modificato (d) mesh strutturata del modello modificato. Nel confronto tra le snapshot si nota un andamento maggiormente lineare, oltre ad un aumento della sensazione di profondita', nella mesh strutturata invece si pone attenzione sul maggiore dettaglio acquisito sugli elementi di sfondo. Gli errori a triangolo sono frutto della triangolazione in fase di creazione della mesh

5.6 Conclusioni

L'algoritmo presentato è dunque in grado di stimare strutture 3D piuttosto dettagliate, estraendo informazioni da una singola immagine. Si è modellata sia la locazione che l'orientamento di piccole zone omogenee nell'immagine, dette superpixel, usando un MRF e sfruttando le relazioni e le feature presenti nella scena. Nonostante gli ottimi risultati ottenuti, e la complessità computazionale presente in Make3D, è stato possibile incrementare le prestazioni dell'algoritmo, per lo più attraverso l'uso di semplici strumenti quali la trasformata di Hough, il calcolo del sobeliano, l'analisi della segmentazione e del colore. Inoltre, dato l'ambiente in cui si va a lavorare, e notando una tendenza all'errore spurio in presenza di segmentazioni non banali, si è preferito adattare l'algoritmo in modo da ottenere un andamento maggiormente lineare, sacrificando la cura del dettaglio estremo a favore di un debole appiattimento in caso di zona incerta, ma evitando così i frequenti errori riscontrati. Oltre ad un netto miglioramento quantitativo questo ha

prodotto un notevole miglioramento dei modelli tridimensionali ottenuti. Su entrambi i fronti sono possibili ulteriori margini di miglioramento, dal punto di vista matematico si potrebbero introdurre ulteriori funzioni, in grado di incrementare l'ottimalità della segmentazione, oppure di post processare i dati in uscita. Il miglioramento, seppur possibile, risulta tutt'altro che banale. Ogni funzione, presa singolarmente, non può sicuramente portare ad un calo drastico sull'errore. Inoltre ogni modifica va valutata su un ampio set eterogeneo di immagini, in modo da verificarne la ripetibilità, ed evitare che si presentino casi in cui il suo uso introduca ulteriore errore. Per quanto riguarda la ricostruzione dei modelli invece, si apre uno scenario piuttosto vasto. Il riconoscimento di forme geometriche ben definite nell'immagine potrebbe essere piuttosto utile come parametro di decisione, permettendo o meno l'uso di forme spigolose nel modello associato, ed usando una funzione di smoothing sui diversi segmenti trovati, nel caso di segmentazione irregolare. Una ricerca delle linee di foreground e background (solo in fase di modellazione) agevolerebbe ulteriormente il processo. La rappresentazione del modello tramite piccoli elementi triangolari potrebbe essere rivista: spesso l'unione lungo un piano perpendicolare al piano della telecamera (piano occluso) di elementi a profondità diversa avviene in modo piuttosto casuale: in primo luogo si potrebbe forzare tale superficie ad assumere i valori di texture del segmento più vicino al piano della camera, in secondo luogo si potrebbe vincolare la profondità massima di tale oggetto, in relazione alla sua dimensione frontale. La versione del modello realizzata in formato .ply, contenente solo i punti effettivamente rilevati e stimati dell'immagine, rappresenta un'ottima base di partenza per uno sviluppo in tal senso.

Appendice A

Creazione di un anaglifo

Un anàglifo è un'immagine stereoscopica, o stereogramma, che, se osservata mediante opportuni occhiali dotati di due filtri di colore complementare l'uno rispetto all'altro, fornisce una illusione di tridimensionalità. Un anaglifo contiene due immagini sovrapposte, riprese alla stessa distanza degli occhi umani. Per la piena fruizione di un'immagine anaglifica è necessario indossare dei caratteristici occhialini con lenti dotate di filtri colorati, che assegnano una porzione ben specifica dello spettro a ciascun occhio, porzione che era stata stabilita in fase di preparazione dell'anaglifo. L'anaglifo è tornato recentemente in voga grazie al suo utilizzo per la presentazione di immagini stereoscopiche in Internet, in Blu-ray ad alta definizione, CD e nella stampa. Gli occhialini di cartoncino con lenti di plastica che utilizzano i filtri colorati, dopo il 2002, permettono l'utilizzo dei tre colori primari. Lo standard corrente dei colori per le lenti dell'anaglifo sono il rosso e il ciano, con il rosso utilizzato per il canale sinistro e il ciano per il destro. Il materiale utilizzato per i filtri è una gelatina monocromatica, etichettata rosso e blu per convenienza e costo.

Come già visto, la visione della natura nella sua tridimensionalità è ottenuta attraverso due immagini parallele scostate orizzontalmente di pochi centimetri l'una dall'altra (tra i 5,5 e i 7,5 nell'occhio umano). Per poter ottenere una immagine stereoscopica che fornisca una illusione di tridimensionalità è perciò necessario riprendere un soggetto con due fotocamere, cineprese o videocamere parallele che restituiscano una doppia immagine del medesimo soggetto alla medesima distanza degli occhi umani. Il medesimo principio di funzionamento dello stereogramma parallelo sta alla base anche dell'anaglifo,

le due immagini parallele sono però riprodotte una sull'altra. La discriminazione delle due immagini destinate separatamente ai due occhi, avvengono però attraverso un filtraggio cromatico: le due immagini vengono filtrate attraverso due gelatine di colore complementare (rosso/verde, blu/giallo o, più comunemente, rosso/ciano) e sovrimpresse sul medesimo supporto, stampa fotografica o pellicola positiva. Per poter osservare le due immagini separatamente, viene successivamente richiesto l'uso di appositi occhiali filtrati con i medesimi colori della stampa: l'occhio che vede attraverso il filtro rosso vedrà le parti rosse dell'immagine come chiare/bianche e le componenti ciano (blu e verdi) come scure/nere. Allo stesso modo l'occhio che vede attraverso il filtro ciano scarterà le componenti rosse, vedendo solamente quelle ciano (verde+blu). Le parti bianche, nere o grige (prive di crominanza), verranno percepite sia dall'occhio destro che dal sinistro. Il cervello unisce le due immagini e interpreta le differenze visive come il risultato della differente distanza tra i soggetti in primo piano, in secondo piano e sullo sfondo. Questo permette di creare una immagine stereoscopica, senza il bisogno di ausili quali un visore stereoscopico, per permettere a entrambi gli occhi di vedere l'immagine che gli compete.

Alla base dunque della creazione di un anaglifo vi è una coppia stereoscopica della scena, ripresa da diverse visuali.

Attraverso le informazioni ricavate attraverso l' algoritmo make3D modificato però, è possibile creare un anaglifo a partire da una singola immagine bidimensionale. Si è pensato dunque di sfruttare le informazioni di profondità ricavate per dividere la nostra immagine in diverse zone di profondità (nel nostro caso si è provato a creare un modello a 3 e a 4 livelli). Definite queste zone si è suddivisa l' immagine per canali colore, isolando dapprima la sola componente R (Red) ed in seguito l' insieme delle componenti G+B (Green e Blue). Per ciascun livello si è assegnato un diverso scostamento laterale, dell' ordine di 50/20/10 pixel a seconda della zona. I canali così discostati sono stati infine fusi insieme, ottenendo per esempio l' anaglifo in figura.

FIGURA A.1: Esempio di anaglifo ottenuto in MATLAB

Bibliografia

[1] G.J. Iddan G. Yahav. 3d imaging in the studio (and elsewhere...). *www.3dvsystems.com.il*, 2005. URL `http://classes.soe.ucsc.edu/cmps290b/Fall05/readings/iddan.pdf`.

[2] Schuon Sebastian Theobalt Christian Davis James Thrun Sebastian. High-quality scanning using time-of-flight depth superresolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008.

[3] Canesta. Canesta's latest 3d sensor cobra 320 x 200 depth sensor, capable of 1mm depth resolution, usb powered, 30 to 100 fps.

[4] Greg Turk Marc Levoy. Zippered polygon meshes from range images in proceedings of siggraph '94. *Graphics Proceedings, Annual Conference Series*, 1994.

[5] Horn B.; Ikeuchi K. Numerical shape from shading and occluding boundaries. *Articial Intelligence*, 1981.

[6] T. K. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *JCV*, 2001.

[7] www.vision.caltech.edu/bouguetj/calibdoc/. Camera calibration toolbox.

[8] *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[9] www.videredesign.com. Videre design.

[10] www.ptgrey.com. Point grey research.

[11] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *JCV*, 2002.

[12] 3D Photo Printing at the Kodak Booth at CES. Ces 2011, 6-9 january.

[13] Pedestrian detection with full auto brake debuts on the allnew volvo S60. Volvo cars of north america press release date of issue mar 02, 2010.

[14] Y.; Mitani J.; Fukui Y. Iizuka, S.; Kanamori. An efficient modeling system for 3d scenes from a single image. *Computer Graphics and Applications, IEEE*, 2011.

[15] S. Mohan. Automated 3d modeling and rendering from single view images. *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on.*

[16] Min Sun Ashutosh Saxena and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. .

[17] Ltd. Youichi Horry Ken-ichi Anjyo Kiyoshi Arai Hitachi. Tour into the picture: Using a spidery mesh interface to make animation from a single image.

[18] T. Aida T. Kurita M. Kawakita, K. Iizuka and H. Kikuchi. Real-time three-dimensional video image composition by depth information. *IEIxpress,CE Electronics*, 2004.

[19] A.A.; Hebert M. Hoiem, D.; Efros. Geometric context from a single image. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005.

[20] Sung H. Chung Ashutosh Saxena and Andrew Y. Ng. Learning depth from single monocular images. .

[21] Leslie Pack Kaelbling Tomas Lozano-Perez Han-Pang Chiu, Huan Liu. Class-specific grasping of 3d objects from a single 2d image. 2007.

[22] Xiaoli Li Vue Ming, Qiuqi Ruan. 3d face reconstruction using a single 2d face image.

[23] Saxena et al. Make 3d, 2008. URL `http://make3d.cs.cornell.edu/`.

[24] *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[25] T Pavlidis. A critical survey of image analysis methods. *ICPR, 502-511*, 1992.

[26] Geman D. Geman, S. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1984.

[27] *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1991.

[28] *Decision, Estimation and Classification: an Introduction to Pattern Recognition and Related Topics*. Wiley, 1989.

[29] Meshlab:open source, portable, and extensible system for the processing and editing of unstructured 3d triangular meshes. URL `http://meshlab.sourceforge.net/`.

[30] S. H. Chung A. Saxena and A. Y. Ng. Learning depth from single monocular images. *Neural Information Processing Systems (NIPS)*, 2005.

[31] Savil Srivastava Ashutosh Saxena, Nuwan Senaratna and Andrew Y. Ng. Rapid interactive 3d reconstruction from a single still image. .