UNIVERSITÀ DEGLI STUDI DI CAGLIARI
FACOLTÀ DI INGEGNERIA
Corso di Laurea Triennale in Ingegneria Elettrica ed Elettronica

RICOSTRUZIONE 3D DA FRAMES MULTIPLI ATTRAVERSO LA TECNICA MULTIVIEW

Tesi di Laurea

Relatore:
Prof. Giuseppe Rodriguez

Candidato:
Alessia Follesa

Anno Accademico 2014/2015

Ai miei genitori e a mia sorella dedico questo lavoro perchè, senza il loro sostegno, questo sogno non si sarebbe avverato.

Indice

Introduzione

La visione è forse il senso più importante che l'uomo possiede. Permette di interpretare il mondo tridimensionale, di riconoscere e localizzare gli oggetti presenti in una scena e percepire i rapidi cambiamenti dell'ambiente. Per questo è facile avere un'idea della struttura 3D mostrata in un'immagine.
Per un dispositivo invece, a causa della perdita di una dimensione nel processo di proiezione dell'oggetto sul piano dell'immagine (Frame), la stima della vera geometria 3D è difficile. E' un cosiddetto problema mal posto, perché di solito infinite superfici 3D differenti possono produrre lo stesso insieme di immagini sul frame.
La Computer Vision è la disciplina che studia come abilitare i computer alla comprensione e interpretazione delle informazioni visuali presenti in immagini e video. Negli ultimi anni sono stati sviluppati un certo numero di algoritmi di alta qualità, permettendo allo stato dell'arte di migliorare rapidamente. Una delle applicazioni principali delle tecniche della Computer Vision è nel campo dell'archeologia. Esse sono diventate straordinariamente importanti da quando gli studiosi, con l'entrata in vigore delle nuove normative per la preservazione dei beni culturali, non possono più avvalersi delle vecchie tecniche invasive per la documentazione di siti, incisioni, bassorilievi e reperti e si sono orientati verso metodi più all'avanguardia, come scanner laser o multiview , meno invasivi per la mappatura tridimensionale sia di oggetti singoli che di scenari interi.
L'obiettivo di questo lavoro è analizzare uno dei metodi per lo sviluppo della Computer Vision, il Multiview, e metterlo a confronto con un altro metodo di ricostruzione 3D, il Photometric Stereo.
Il Multiview è una tecnica che consente, in base a modelli matematici dedicati, di ricostruire un modello 3D completo di una scena attraverso gli scatti (frames) catturati da punti di vista differenti della fotocamera. Della fotocamera solitamente si conosce la posizione oppure la si può ricavare dal DataSet di immagini.
Il primo step consisterà nel rilevamento dei punti caratteristici dell'immagine 2d, confrontando diversi algoritmi. Successivamente ci imbatteremo nel

problema della corrispondenza di tali punti in tutti i successivi frames, informazione necessaria ai fini della ricostruzione. Spiegheremo infine l'algoritmo della fattorizzazione, ideato da Lucas C. e Kanade T., adatto per estrapolare le informazioni riguardanti le coordinate 3D della scena.

Capitolo 1

Computer Vision

1.1 Introduzione

La Computer Vision è un insieme di tecniche di visione artificiale o computazionale volte a riprodurre le abilità della visione umana in dispositivi elettronici per l'acquisizione e la comprensione delle immagini. Il suo scopo è quello di interpretare gli effetti della visione umana attraverso l'acquisizione, l'elaborazione al calcolatore e la successiva comprensione delle immagini. Sono coinvolti molti aspetti, ma il più importante è *l'Image Processing*.
Il problema principale per questa disciplina è quello di riuscire a elaborare e riportare sul dispositivo le proprietà 3D reali da una o più immagini digitali. Le proprietà che ci interessando maggiormente sono di natura *geometrica* (ad esempio forma e posizione degli oggetti solidi) e *dinamiche* (ad esempio la velocità degli oggetti).

1.2 Discipline correlate

Risulta assai difficile elaborare una lista esaustiva di tutti i campi a cui la *Computer Vision* può essere applicata, perchè l'argomento è vasto, multidisciplinare e in continua espansione: nuove applicazioni nascono continuamente. Per questo ci soffermeremo a descrivere solo gli aspetti e le discipline strettamente correlate a ciò che riguarda il nostro lavoro(1.1).

Image Processing. L'Image processing è la disciplina che si occupa di rielaborare le immagini utilizzando operazioni matematiche e qualsiasi forma di elaborazione. Per i nostri scopi è una disciplina che differisce dalla Computer Vision in ciò che concerne le proprietà dell'immagine, potemmo dire che quest'ultima è considerata a un livello di elaborazione più alto rispetto

all'image processing. Poichè molti algoritmi per la Computer Vision richiedono come fase preliminare l'image processing è facile sovrapporre le due discipline. Particolari aree di applicazione dell'Image Processing includono: *l'image enhancement* (calcola un immagine con una qualità migliore rispetto all'originale), *image compression* (rappresentazione compatta dell'immagine, utilizzata per la trasmissione), *image restoration* (elimina gli effetti noti dovuti alla degradazione), *feature extraction* (individuazione di elementi caratteristici dell'immagine come contorni, o aree strutturate).

Pattern Recognition. La Pattern Recognition studia tecniche per il riconoscimento e la classificazione degli oggetti servendosi delle immagini digitali. Molti dei metodi che sono stati sviluppati in passato non erano adatti per il mondo 3D. Questo ha portato, negli anni successivi, gran parte della ricerca verso il campo della Computer Vision sebbene qualche problema ci sia ancora.

Fotogrammetria. La Fotogrammetria è la scienza che si occupa di ottenere misure accurate e affidabili da fotografie. L'output è tipicamente una mappa, un disegno, una misura oppure un modello 3D di qualche oggetto o scena reale. Questa disciplina si sovrappone meno alla Computer Vision rispetto alle due descritte in precedenza. La differenza principale è che la fotogrammetria si propone di raggiungere un'accuratezza molto elevata rispetto alla computer vision, e non tutti i metodi della computer vision sono legati alla misurazione.

1.3 Applicazioni

Le *Applicazioni* coprono uno spettro molto ampio. Ad esempio:

- Il fax 3D è un'apparecchiatura in grado di realizzare a distanza una copia fisica di un oggetto tridimensionale. La stazione trasmittente è costituita da un dispositivo in grado di collezionare informazioni sulla struttura dell'oggetto, informazioni che il dispositivo ricevente utilizza per guidare la costruzione della copia, ad esempio tramite stereolitografia.

- L'archeologia e l'arte sono caratterizzate dalle contrastanti necessità di proteggere e di rendere fruibili rari reperti o irripetibili opere d'arte.

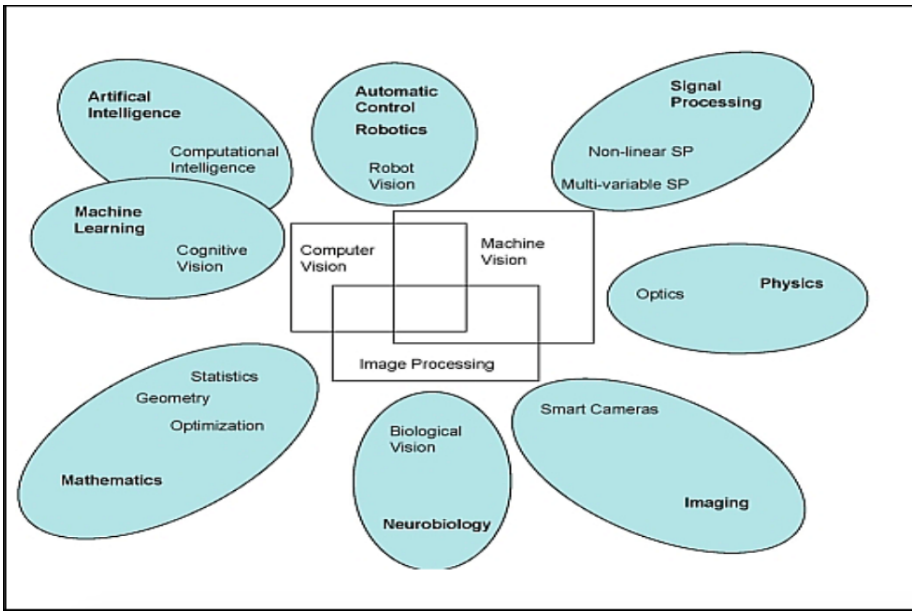

Figura 1.1: Relazione tra Computer Vision e altri campi di ricerca.

L'acquisizione di questi oggetti (in realtà possono essere interi complessi) consente da una parte una misura ed una catalogazione più accurata, e dall'altra permette l'accesso alle copie digitali per scopo di studio o per visite culturali ad un numero elevato di persone, senza rischio per i reperti originali.

- La realtà virtuale può essere utilizzata per addestrare, in situazione controllabile, all'esecuzione di compiti critici. Per esempio, in medicina e chirurgia è necessario acquisire una certa esperienza prima di poter operare correttamente. L'inesperienza, cioè il trovarsi a gestire una situazione mai conosciuta prima, può in questo campo comportare un danno grave al paziente. Esiste un'applicazione che permette di simulare operazioni ortopediche. Il sistema è in grado di simulare svariate situazioni di frattura del femore e gli strumenti e gli impianti per la ricomposizione delle fratture.

Le differenti applicazioni hanno caratteristiche computazionali e ambientali differenti. Per esempio, l'archeologia virtuale richiede una grande accuratezza nella ricostruzione, una bassa invasività delle tecniche di acquisizione, ma generalmente non impone limitazioni di tempo per l'elaborazione.

Il fax 3D presuppone che ad utilizzarlo sia un operatore senza particolari conoscenze tecnologiche. La facilità di utilizzo e il basso costo del dispositivo (nonchè la sua compattezza) sono quindi le caratteristiche principalmente desiderate.

1.4 Tecniche maggiormente usate nella Computer Vision

Tante sono le sfaccettature che contraddistinguono un metodo dall'altro. L'obiettivo in questo paragrafo è quello di descriverne brevemente solo due, indicando le principali differenze, vantaggi e svantaggi, soprattutto dal punto di vista del carico computazionale.

1.4.1 Photometric Stereo

La *Photometric Stereo* è una tecnica che consente, in base a modelli matematici dedicati, di ricostruire delle superfici 3-D, a partire semplicemente da delle fotografie acquisite secondo condizioni di illuminazione differenti. Risulta molto economica e facile da implementare quando la posizione delle luci è nota e molto precisa.
Per la ricostruzione si serve di una tecnica che consente di analizzare il flusso della radiazione elettromagnetica di un oggetto: la *fotometria*. Una delle informazioni che si perde durante l'acquisizione di un'immagine è proprio la profondità che in realtà può essere ricostruita a partire dalla conoscenza delle caratteristiche fotometriche dell'oggetto, ad esempio lo spettro di radiazione. Nel caso della Photometric Stereo il punto di osservazione è sempre il medesimo e ciò presenta un grosso vantaggio sia in termini di costo, perché in questo caso è sufficiente una solo fotocamera, ma anche computazionali in quanto, rispetto alla fotometria binoculare, non sussiste il problema della corrispondenza biunivoca. Ogni singolo frame viene infatti acquisito illuminando gli oggetti che si vogliono ricostruire da diverse angolazioni. In altri termini in questa circostanza è la sorgente luminosa che viene spostata intorno all'oggetto. La finalità è dunque quella di studiare il comportamento fotometrico dell'oggetto, eccitato in condizioni differenti, ricostruendo una mappa dei gradienti e una mappa delle normali che ci consentono di stabilire l'orientazione punto per punto dell'oggetto che può essere poi in un secondo momento utilizzata per ricostruire la superficie stessa.

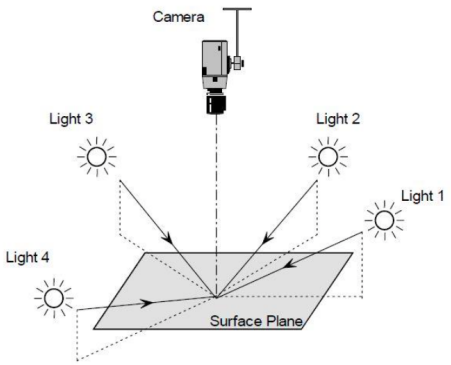

Figura 1.2: Illustrazione grafica del processo di acquisizione

Esistono però tutta una serie di limiti piuttosto importanti che a volte incidono pesantemente nella ricostruzione 3D delle superfici. Prima fra tutte le proprietà fisiche degli oggetti, infatti per ottenere buoni risultati, quest'ultimi dovrebbero essere il più regolare possibili, quindi caratterizzati da geometrie non troppo complesse. A tal proposito si definiscono Lambertiani tutti quei corpi che rispettano la legge di Lambert, meglio nota come la legge del coseno. Quest'ultima afferma che l'illuminamento prodotto da una sorgente su una superficie è direttamente proporzionale all'intensità luminosa della sorgente e al coseno dell'angolo che la normale alla superficie forma con la direzione dei raggi luminosi.
In altri termini una superficie può essere definita Lambertiana, se indipendentemente dal punto di osservazione in cui viene osservata, quest'ultima riflette la luce con egual intensità. Empiricamente questa situazione è tanto più verificata quanto più le superfici presentano geometrie non troppo complesse, quindi poco spigolose, e caratterizzate da materiali altamente riflettenti. Purtroppo tale proprietà spesso non viene mai rispettata, in quanto gli oggetti che si vogliono ricostruire il più delle volte sono geometricamente complessi e caratterizzati da superfici ruvide. Tutti questi fenomeni di non idealità si riflettono sulla propagazione degli errori. Poiché comportano come input, dati piuttosto perturbati, per cui nostri output saranno di conseguenza affetti da una grossa componente di rumore. Possono però essere adottate delle precauzioni per limitare il mal condizionamento dei dati. Una di queste sicuramente consta nello scegliere il numero corretto di immagini. Infatti maggiore sarà il numero dei frame minore sarà il rumore presente in uscita, ma non a caso precedentemente abbiamo utilizzato l'espressione: "numero corretto". Infatti al crescere della quantità di immagini aumenta anche il carico computazionale. In altri termini dobbiamo trovare un giusto compro-

messo tra la fedeltà dei risultati e i tempi di calcolo.
La Photometric Stereo è un metodo di ricostruzione delle superfici piuttosto valido. In realtà rimane un piccolo punto da chiarire. É una tecnica che consente di ricostruire delle superfici 3D, parlare però di vere e proprie ricostruzioni 3D non è corretto perchè non si ha una rappresentazione a tutto tondo degli oggetti.
Quando si ha la necessità di una rappresentazione a tutto tondo la Photometric Stereo non è sufficiente. In questi casi si può ricorrere alla strumentazione Laser(rileva la distanza di una superficie cronometrando il tempo di andata e ritorno di un impulso di luce (1.3), che però ha dei limiti che sono principalmente legati al costo e ai lunghi tempi di acquisizione.

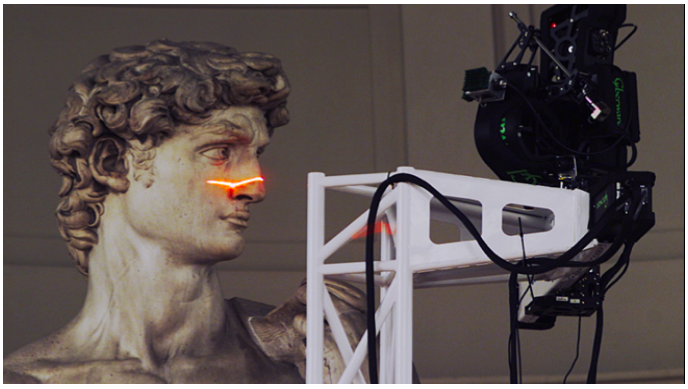

Figura 1.3: Laser Scanning

Esiste un modo per integrare l'utilizzo della Photometric Stereo con qualcos'altro che consenta di ricostruire una superficie a 360°? La risposta è ovviamente si. Infatti si potrebbe pensare di ricorrere al ***Multiview.***

1.4.2 Multiview

Possiamo sicuramente affermare che la photometric stereo è un metodo di ricostruzione delle superfici 3D piuttosto valido in quanto molto accurato e capace, grazie ad algoritmi dedicati, di catturare molti dettagli ma è soggetto a un limite in quanto riesce a ricostruire solamente una facciata della superficie dell'oggetto. Una soluzione alternativa potrebbe essere il laser scanning ma ha delle problematiche che riguardano il grande carico di dati (BigData), quindi un tempo di acquisizione ed elaborazione molto elevato, inoltre la strumentazione laser è molto ingombrante non utilizzabile quindi in applicazioni particolari come l'archeologia.
Si ha l'esigenza di ricostruire superfici a 360° con l'acquisizione di un numero di dati adeguato, in modo da non avere un carico computazionale troppo elevato, con una strumentazione minimale. Questa viene soddisfatta grazie alla tecnica del **MultiView**.
L'obiettivo del *MultiView* è quello di ricostruire un modello completo di oggetti in 3D da un DataSet di immagini scattate da punti di vista differenti e noti(1.4).

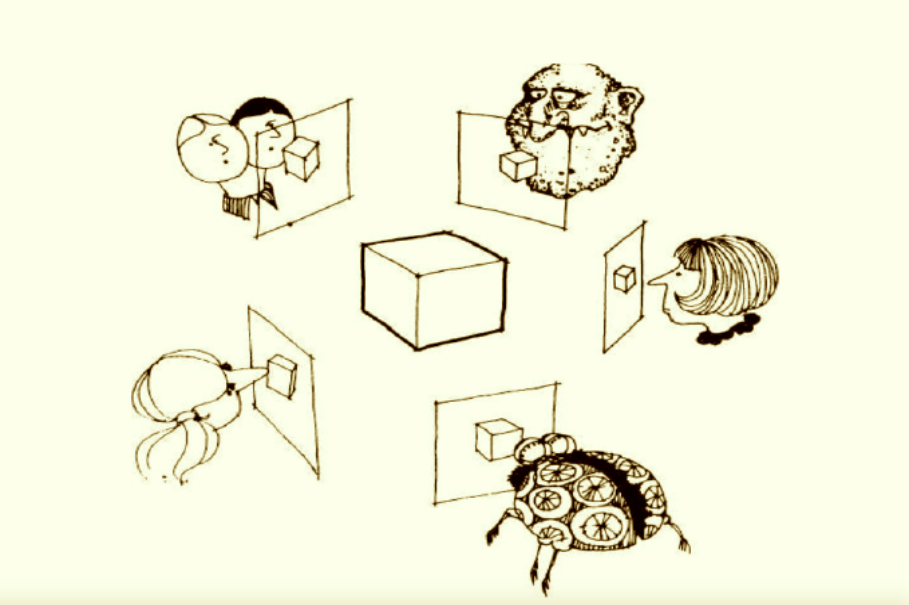

Figura 1.4: MultiView Stereo

Negli ultimi anni sono stati sviluppati un certo numero di algoritmi di alta qualità che tendenzialmente miglioreranno ancora.
La tecnica del Multiview viene spesso confusa con il ***Binocular Stereo***(1.5),

o MultiView Stereo, in cui l'obiettivo è quello di ricavare una mappa di di profondità di un oggetto a partire da una coppia di fotografie (una per ogni occhio) scattate da diverse angolazioni note. Questo è un problema di *Stereo Matching* ovvero immaginando di avere solo due scatti S_1 e S_2, ogni singolo punto di S_1 deve trovare una corrispondenza biunivoca in S_2, e questo deve valere reciprocamente per i punti S_2. Una volta ricreata una mappa di corrispondenze sarà possibile ricostruire la superficie 3D dell'oggetto.

$$z = \frac{df}{x1 - x2}$$

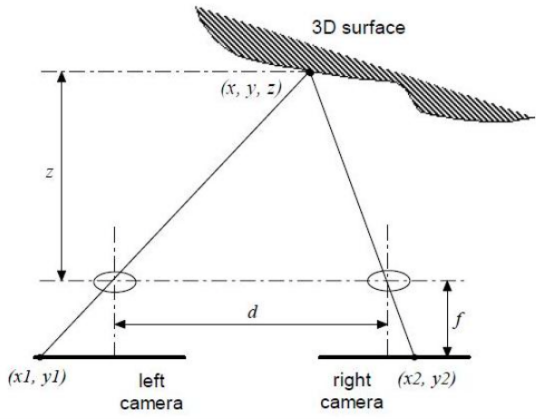

Figura 1.5: Binocular Stereo

L'obiettivo del MultiView è lo stesso, ma ci si arriva in una maniera diversa, è una generalizzazione della visione binoculare. Si ha a disposizione una telecamera che muovendosi cattura $\boldsymbol{f}$ immagini (DataSet) al medesimo istante, ovviamente maggiore è il numero di frame $\boldsymbol{f}$ maggiore sarà l'accuratezza della ricostruzione. Per ricondurci al caso del binocular stereo potremo pensarlo un po' come un essere dotato di molti occhi.

Capitolo 2

MultiView

Introduzione

L'essenza di un'immagine è una proiezione da una scena 3D su un piano 2D, ma durante questo processo la profondità viene persa. Esistono numerosi algoritmi che riguardano la tecnica del Multiview. Queste variano notevolmente nelle loro ipotesi, nei campi di funzionamento e per il comportamento. In generale i metodi esistenti si classificano in base alle proprietà fondamentali che differenziano gli algoritmi principali.

2.1 Formazione dell'immagine e calibrazione della camera

La risoluzione di molte problematiche di Computer Vision parte dall'analisi del processo di formazione dell'immagine in una scena.
Gli oggetti che compongono una scena 3D riflettono la luce, creando un'immagine 2D su un piano di immagine. Per definirla occorre studiarne la geometria (il modo in cui la luce si riflette rimbalzando sugli oggetti del mondo fino a un punto sul piano dell'immagine) e la fotometria (il modo in cui la luce nella scena determina la luminosità dei punti sull'immagine).
Vogliamo far in modo che ogni punto della scena "influisca" su di un solo punto del piano immagine, una possibilità è quella di costringere tutti i raggi a passare per un foro molto piccolo (pinhole). La scatola entro cui si forma l'immagine viene della *camera pinhole*, questo tipo di dispositivo richiede una superficie con un range di sensibilità molto elevato per questo non è pratica da utilizzare. La soluzione migliore sono le *lenti sottili*.
Le lenti sottili sono un dispositivo ottico più complesso e flessibile per mettere

a fuoco l'immagine di una scena, possiamo immaginarle come un sottile disco di vetro di un materiale trasparente in cui vengono definiti un asse ottico e due fuochi (punti particolari dell'asse ottico esterni alla lente).

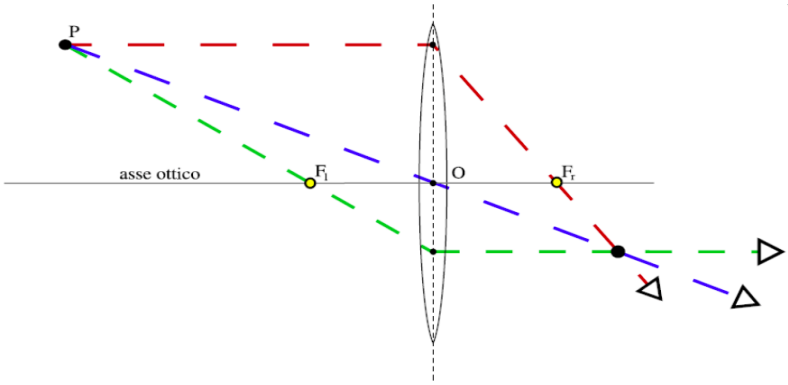

Figura 2.1: Lenti Sottili

Le lenti deviano i raggi di luce seguendo due regole:

1. ogni raggio di luce che entra da un lato della lente parallelamente all'asse ottico viene deviato verso il fuoco che si trova dall'altro lato:

2. ogni raggio di luce che entra da un lato della lente passando per il fuoco, esce dall'altro lato parallelamente all'asse ottico;

Equazione fondamentale lenti sottili:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \tag{2.1}$$

in cui:

Z : distanza dell'oggetto dalla lente

z : distanza d'immagine dalla lente

f : lunghezza focale (distanza del fuoco dal centro)

L'equazione fondamentale ha una conseguenza importante: affinchè l'immagine sia a fuoco, a parità di lunghezza focale i punti devono trovarsi alla stessa distanza dalla lente. In altri termini, una lente è in grado di mettere a fuoco solo una sezione della scena parallela al piano d'immagine.

La camera non è il centro del mondo: non abbiamo più un unico sistema di riferimento. Inizialmente le coordinate dei punti che vengono considerate

si trovano in un sistema di riferimento solidale con la camera nei suoi movimenti, detto sistema di riferimento della camera stessa (*camera reference*). In generale però i punti sono forniti in un sistema di riferimento "assoluto", detto sistema di riferimento del mondo (*world reference*), la cui relazione con il sistema camera è spesso sconosciuta e quindi deve essere ricostruita sulla base delle immagini disponibili.
Questa relazione viene regolata dalle leggi della prospettiva, che indicano il modo di trasformazione dei punti della scena nei riferimenti solidali alla camera, e dalle trasformazioni tra camera reference e world reference.
Per passare da un sistema di riferimento all'altro serve una *rototraslazione* nello spazio che è descrivibile con un vettore **T** che esprime gli offset di traslazione (3 valori) e una matrice di rotazione **R** che esprime la rotazione (3 gradi di libertà).
Di seguito la relazione tra un punto espresso nel sistema di riferimento del mondo (P_w) e l'equivalente del sistema di riferimento della camera (P_c)

$$P_c = R(P_w - T) \tag{2.2}$$

I sei parametri (tre angoli di rotazione e tre coordinate) che descrivono questa trasformazione vengono detti ***parametri estrinseci*** della camera e consentono di passare dal sistema di riferimento del mondo a quello della camera.
Per definite completamente la proiezione cui ogni punto è sottoposto servono altri parametri:

1. lunghezza focale;
2. piano d'immagine con un proprio sistema di riferimento esprimibile in pixel;
3. l'unità di misura del mondo 3D;
4. distorsione radiale introdotta dalle lenti reali (k_1,k_2);

Queste grandezze sono dette ***parametri intrinseci*** della camera ed esprimono la mappatura tra le coordinate geometriche e le coordinate in pixel nell'immagine digitale prodotta.
La determinazione con accuratezza accettabile di tutti i parametri di una camera è detta ***calibrazione della camera*** ed è un'esigenza fondamentale nella maggior parte delle applicazioni di Computer Vision.
Le lenti reali distorcono l'immagine planare che si forma. Tale distorsione è tipicamente piccola vicino al centro dell'immagine, ma diventa significativa nella periferia. Un modello matematico per tener conto di tale distorsione è detto *radiale*.

Se (x,y) sono le coordinate geometriche e (x_d,y_d) sono le coordinate dopo la distorsione, si ha:

$$r^2 = x_d^2 + y_d^2 \tag{2.3}$$

$$x = x_d(1 + k_1 r^2 + k_2 r^4) \tag{2.4}$$

$$y = y_d(1 + k_1 r^2 + k_2 r^4) \tag{2.5}$$

dove r è la distanza del punto dal centro dell'immagine.

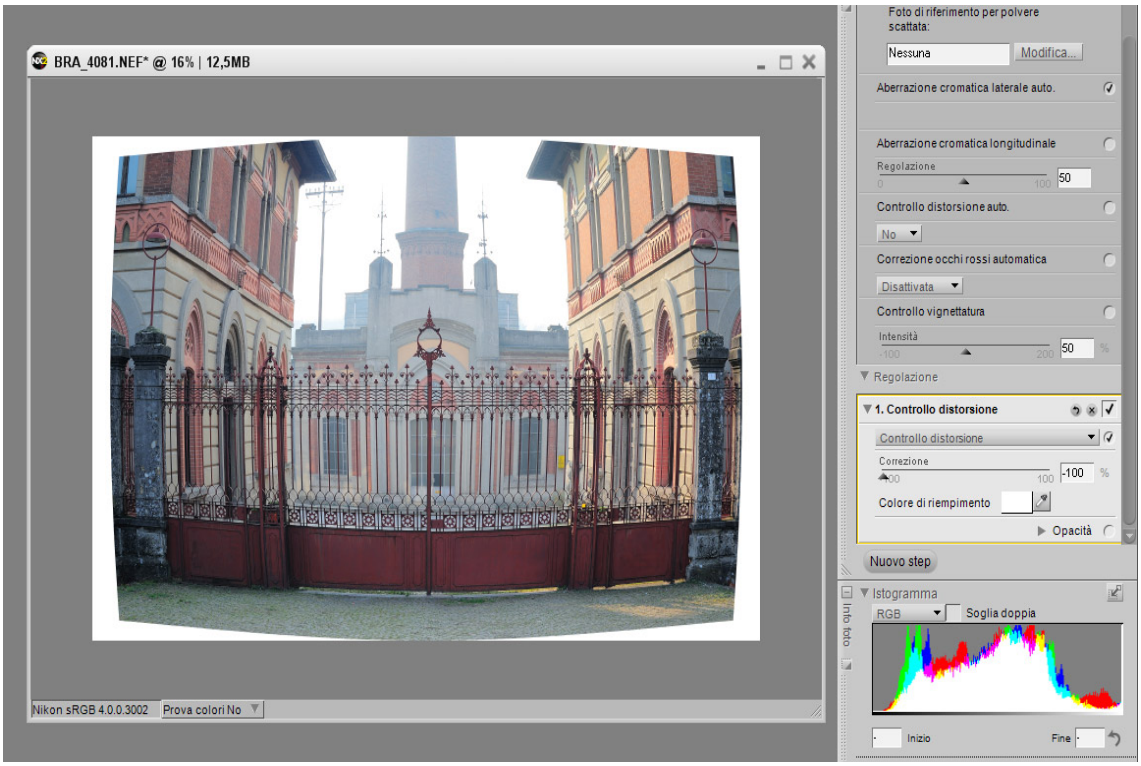

Figura 2.2: Esempio Distorsione

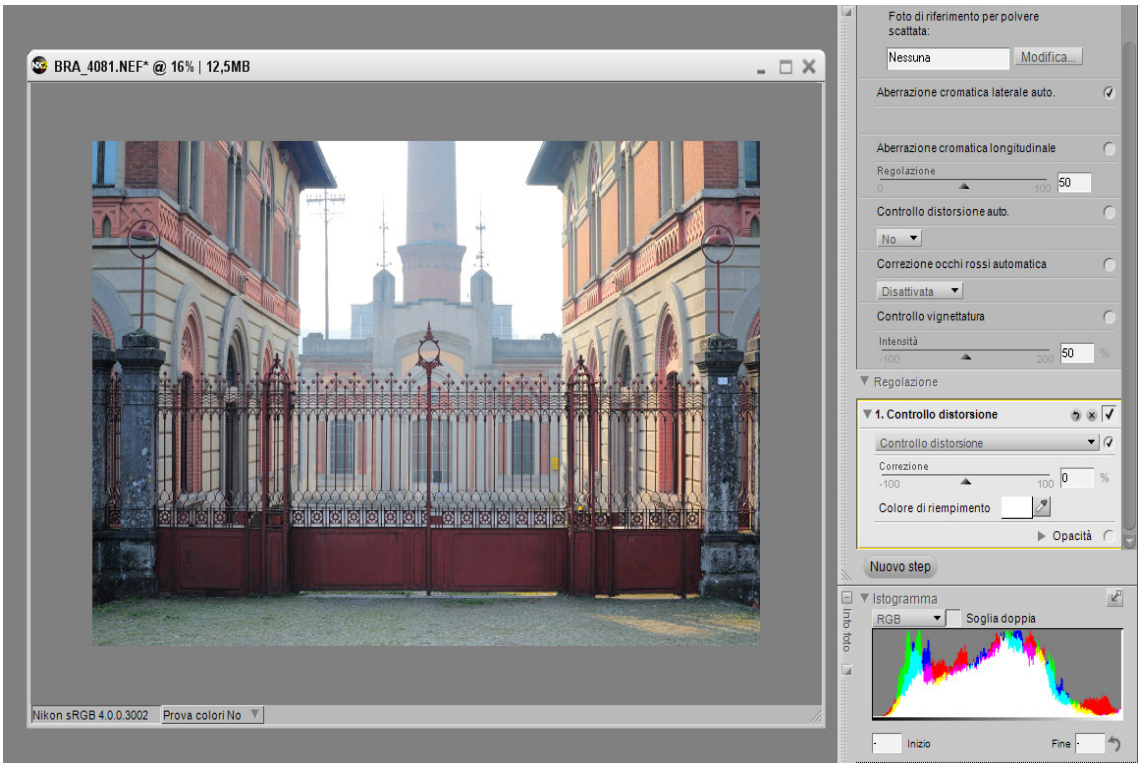

Figura 2.3: Esempio Correzione

Definiti i parametri estrinseci ed intrinseci della camera, siamo in grado di tradurre il sistema di riferimento del mondo in quello della camera.
Per proiettare un punto della scena P_w sul piano immagine di una camera arbitrariamente orientata occorre:

1. tradurre le coordinate di P_w nel sistema di riferimento della camera tramite i parametri estrinseci;

2. proiettare P_c con le equazioni fondamentali della proiezione prospettica;

3. tradurre le coordinate geometriche del punto proiettato in pixel tramite i parametri intrinseci;

Il nostro lavoro è partito proprio dalla creazione di immagine 2D, utilizzando il software MATLAB.
Il programma inizialmente ha come input una matrice M, che contiene le coordinate $\mathbf{P}$ dei vertici 3D di un poliedro, e un vettore k che rappresenta il raggio di proiezione ovvero il punto di vista della camera. Ricevuti in input questi dati il primo passo consiste nella proiezione dei punti 3D in un piano

d'immagine. In questo modo:

```
function[U, i, j]=proiezione(M, k)

P = size(M,2);

if k(2)~= 0
    it=[1; -k(1)/k(2); 0];
    i=it/norm(it);
    j=cross(k,i);
elseif k(1)~=0
    jt=[-k(2)/k(1); 1; 0];
    j=jt/norm(jt);
    i=cross(j,k);
else
    i=[1; 0; 0 ];
    j=[0; 1; 0];
end


U=zeros(2,P);

for n=1:1:P
    Qp=M(:,n);
    Q=Qp-((Qp'*k)*k);
    U(:,n) = [i j]'*Q;
end
```

dove i e j, versori ortogonali al raggio di proiezione k, dentificano l'orientamento del piano dell'immagine del frame in questione.
Questo script genera una matrice U che contiene le coordinate 2D dei vertici del poliedro.
É stata in questo modo creata l'immagine.

Abbiamo visto come si forma l'immagine di una scena: se conosciamo i dati 3D e i parametri della camera, sappiamo calcolare ciò che "vede" la camera. Ma l'obiettivo della computer Vision è risolvere il problema opposto, ovvero, risalire da una o più immagini alle informazioni 3D riguardanti la scena.
Se conosciamo già i parametri della camera la soluzione è relativamente semplice attraverso il metodo della *triangolazione* o metodi affini. Se invece i

parametri non sono noti è necessario ricavarli partendo da una sola immagine, a patto però che si conoscano a priori un certo numero n di corrispondenze tra punti della scena e punti dell'immagine 2D.

2.2 Caratteristiche dell'immagine

Che cosa distingue un'immagine digitale dalle altre? Con digitale si vuole esprimere il concetto di un qualcosa che può essere compreso da un calcolatore, cioè una rappresentazione numerica. Digitale deriva da digit che in inglese significa cifra; a sua volta digit deriva dal latino digitus che significa dito. In definitiva, è digitale ciò che è rappresentato con i numeri che si contano, appunto, con le dita.
Per immagine digitale s'intende l'immagine che è stata digitalizzata, ossia rappresentata in numeri. Come avviene questo processo?
Il primo passo è dividere il soggetto in unità distinte. L'immagine è divisa in una griglia di elementi della figura, detti anche pixel (contrazione della locuzione inglese picture element); se ne fa, tecnicamente, un campionamento spaziale. Il dettaglio raggiungibile e la complessità della griglia (ossia la sua risoluzione) variano a seconda di quanto è sofisticato il sistema di acquisizione.
Se pensiamo dapprima a un'immagine in bianco e nero, senza ombreggiature o livelli di chiaroscuro, e la suddividiamo in una griglia formata da righe orizzontali e verticali a distanza costante ogni quadratino che ne deriva è un pixel e può essere codificato in numero binario. Il simbolo "0" viene utilizzato per la codifica di un pixel corrispondente a un quadratino bianco (in cui il bianco è dominante), mentre "1" per la codifica di un pixel corrispondente a un quadratino nero (in cui il nero è dominante). Poiché una sequenza di bit è lineare, è necessario definire delle convenzioni per ordinare la griglia dei pixel in una sequenza: assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra. Non sempre il contorno della figura coincide con le linee della griglia. Quella che si ottiene nella codifica è un'approssimazione della figura originaria, la digitalizzazione comporta quindi perdita di qualità. La rappresentazione sarà più fedele all'aumentare del numero di pixel, ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine.
Per codificare le immagini con diversi livelli di grigio si usa la stessa tecnica: per ogni pixel si stabilisce il livello medio di grigio cui viene assegnata convenzionalmente una rappresentazione binaria.
Per memorizzare un pixel non è più sufficiente un solo bit. Ad esempio, se utilizziamo quattro bit possiamo rappresentare $2^4 = 16$ livelli di grigio, men-

tre con otto bit ne possiamo distinguere $2^8 = 256$, ecc.
Analogamente possiamo codificare le immagini a colori. In questo caso si tratta di individuare un certo numero di sfumature, gradazioni di colore differenti, e di codificare ognuna mediante un'opportuna sequenza di bit. Qualsiasi colore può essere rappresentato dalla composizione del rosso, del verde e del blu: *codifica* ***RGB*** (Red, Green, Blu - Rosso, Verde, Blu ovvero i tre colori primari). Per ogni colore primario si usa un certo numero di bit per rappresentarne la gradazione; ad esempio, utilizzando 8 bit per colore primario, otteniamo 256 diverse gradazioni, ovvero $256 \times 256 \times 256 = 16777216$ colori diversi.
Nel caso della codifica del colore quindi un pixel richiede tre byte di informazione.

Queste matrici numeriche bidimensionali possono a loro volta dare luogo a due tipi d'immagine:

Immagini a modulazione d'intensità o luminosità: le normali fotografie che codificano l'intensità della luce, acquisite tramite i normali sensori alla luce (misurano la quantità di luce che si imprime sul sensore).

Immagini spaziali: codificano la forma e la distanza: (stimano direttamente le strutture 3D della scena osservata attraverso varie tecniche). Sono ottenute con l'utilizzo di sonar o scanner laser.

A seconda della natura dell'immagine i numeri possono quindi rappresentare valori diversi, quali l'intensità della luce, le distanze o altre quantità fisiche. La prima considerazione che si può trarre è che la relazione tra l'immagine e il mondo rappresentato dipende dal processo di acquisizione e quindi dal sensore utilizzato. La seconda considerazione riguarda il fatto che ogni informazione contenuta nell'immagine deve essere ricavata da una matrice numerica.
É importante comprendere la relazione tra la geometria della formazione dell'immagine e la rappresentazione delle immagini nel computer.
Un pixel è un campione dell'intensità dell'immagine quantizzato a un valore intero e l'immagine è una matrice bidimensionale di pixel. Gli indici [i,j] di un pixel sono valori interi che specificano le righe e le colonne della matrice. Il pixel [0,0] è posizionato nell'angolo in alto a sinistra dell'immagine. I valori dell'indice i si susseguono dall'alto verso il basso, mentre quelli di j si dirigono da sinistra verso destra. Questo tipo di notazione corrisponde strettamente alla sintassi della matrice utilizzata nei programmi. La posizione dei punti sul piano dell'immagine ha come coordinate x e y. La coordinata y (ordinate)

corrisponde alla direzione verticale, la x (ascisse) a quella orizzontale. L'asse delle y va dal basso verso l'alto, quella delle x sinistra verso destra. Quindi i valori riportati dalla matrice degli indici i e j sono in ordine rovescio rispetto ai valori riportati dalla matrice delle coordinate relative alle posizioni x e y. È necessario quindi eseguire degli algoritmi per passare da un sistema di coordinate all'altro.
In un sistema per la formazione dell'immagine, ogni pixel occupa un'area definita sul piano dell'immagine. Le posizioni sul piano dell'immagine possono essere quindi rappresentate da frazioni di pixel. La matrice dei pixel del software corrisponde alla griglia di posizioni sul piano dell'immagine da cui è ottenuta. Questa premessa ci permette di comprendere quali elaborazioni possono essere compiute sull'immagine, o meglio sulla matrice di valori che la rappresenta. Tuttavia è necessario considerare il livello, o meglio la posizione, su cui si opera. Quindi occorre considerare ogni algoritmo in base alle trasformazioni che pone in essere, a quali sono i dati richiesti e i risultati forniti. Certamente l'elaborazione si svolge sull'immagine, ma come risultato si hanno dei simboli rappresentanti, per esempio, l'identità e la posizione di un oggetto.
A seconda di quali dati si ha la necessità di trattare possiamo distinguere dei livelli, di cui in seguito una breve descrizione:

Livello puntuale. Operazioni che si basano solo su un punto dell'immagine, ad esempio l'operazione di soglia.

Livello locale. L'intensità dei punti nell'immagine risultato dipende non solo da un singolo punto dell'immagine di partenza, ma anche da quelli che gli sono adiacenti/vicini. La ridistribuzione dei punti (smoothing) e la rilevazione dei contorni sono operazioni locali.

Livello globale. Un istogramma dei valori d'intensità o una trasformata di Fourier sono esempi di operazioni globali.

Livello oggetto. Questo livello è il più specifico della Computer Vision, dato che gli altri sono la base anche per altre materie come l'elaborazione dell'immagine. Le dimensioni, l'intensità media, la forma, e altre caratteristiche dell'oggetto devono essere valutate perché il sistema le riconosca. Al fine di determinare queste proprietà vengono effettuate delle operazioni solamente sui pixel appartenenti all'oggetto. Ma il punto centrale è: cos'è un oggetto? Come si può rilevare?
Gli oggetti sono normalmente definiti dal loro particolare contesto. In effetti, molte operazioni in computer vision sono svolte per trovare la

posizione di un oggetto nell'immagine. Tuttavia, definire cos'è un oggetto è come trovarsi nella condizione del "gatto che si morde la coda". Per valutare le caratteristiche dell'oggetto abbiamo bisogno di sapere quali punti appartengono a tale oggetto, ma per identificarli è necessario sapere da quali caratteristiche sono contraddistinti. Sono stati fatti molti sforzi per distinguere le figure dallo sfondo, o raggruppare i punti in oggetti.

2.2.1 Movimento

Introducendo il movimento si amplia la prospettiva in quanto l'elaborazione dell'immagine abbraccia la dimensione temporale. Più precisamente, si può disporre delle informazioni visive che possono essere estratte da variazioni spaziali e temporali presenti in una sequenza d'immagini.

Sequenza di immagini: É una serie di N immagini, o *frames*, acquisite in istanti di tempo

$$t_{\mathrm{k}} = t_0 + k\Delta t \tag{2.6}$$

dove Δt è un intervallo di tempo fissato e

$$k = 0, 1, ..., N-1 \tag{2.7}$$

.
Al fine di acquisire una sequenza di immagini, è necessario un frame grabber (dispositivo analogico-digitale, il cui compito è quello di digitalizzare le informazioni) in grado di memorizzare i frame a un ritmo veloce. In genere i rates tipico sono il cosiddetto frame rate e field rate, corrispondenti rispettivamente ad un intervallo di tempo Δt di $\frac{1}{24}$sec e $\frac{1}{30}$sec rispettivamente.
É importante assicurarsi che Δt sia abbastanza piccolo da garantire che la sequenza discreta sia un campione rappresentativo dell'immagine in continua evoluzione nel tempo; come regola generale, ciò significa che gli spostamenti apparenti sul piano dell'immagine tra i frames dovrebbero essere al massimo di pochi pixel.
Assumeremo che le condizioni di illuminazione siano costanti e che i cambiamenti che riguardano l'immagine sono causati da un relativo movimento tra fotocamera e scena. Si possono presentare le seguenti situazioni, ognuna delle quali richiede una diversa tecnica d'analisi:

- **Camera fissa, un singolo oggetto fisso, sfondo fisso.** É una semplice scena statica.

- **Camera fissa, un singolo oggetto in movimento, sfondo fisso.** L'oggetto in movimento sullo sfondo comporta dei movimenti dei pixel nell'immagine associati all'oggetto. La rilevazione di questi pixel può svelare la forma dell'oggetto così come la sua velocità e percorso. Questo tipo di sensori è normalmente utilizzato per la sicurezza e la sorveglianza.

- **Camera fissa, più oggetti in movimento, sfondo fisso.** Il movimento di uno o più oggetti può essere tracciato per ottenere una traiettoria o un percorso dai quali sarà possibile trarre indicazioni sul comportamento dell'oggetto. È il caso di una telecamera usata per analizzare il comportamento di alcune persone che entrano in un edificio per affari o altro lavoro. Diverse telecamere possono essere utilizzate per ottenere diversi punti di vista dello stesso oggetto, permettendo quindi di elaborare percorsi tridimensionali. Possibili applicazioni sono l'analisi del movimento di atleti o di pazienti in riabilitazione. Vi è anche un sistema in via di sviluppo che traccia i movimenti, durante un incontro di tennis, della palla e dei giocatori fornendo l'analisi degli elementi del gioco.

- **Camera in movimento, diversi oggetti in movimento.** Questa situazione presenta i problemi relativi al movimento più difficili da risolvere e probabilmente più importanti in quanto riguardano situazioni in cui sono in movimento i sensori ma anche una grande quantità di oggetti nella scena osservata. È il caso di un veicolo che si muove nel traffico di punta, o di alcune telecamere che devono seguire automaticamente degli oggetti in movimento.

- **Camera in movimento, scena relativamente costante.** Una telecamera in movimento provoca dei cambiamenti nelle immagini dovuti al suo stesso movimento, anche se l'ambiente non cambia. Si può utilizzare questo movimento in modi diversi. Ad esempio si può ottenere una più ampia visione dell'ambiente rispetto all'osservazione da un singolo punto fisso: è il caso di un movimento panoramico di macchina. Il movimento della camera può anche fornire informazioni sulla profondità relativa degli oggetti, in quanto le immagini di quelli vicini cambiano più velocemente di quelle relative ai lontani. In terzo luogo, esso può dare la percezione o la misurazione della forma di oggetti 3D vicini: i molteplici punti di vista permettono infatti di effettuare calcoli trigonometrici simili alla visione stereo.
  Questo è il caso che verrà analizzato in questo lavoro di tesi.

La descrizione del lavoro di codifica si è fermato alla creazione di un solo frame. Adesso abbiamo la necessità di muovere la camera secondo una precisa traiettoria e di far corrispondere a ogni spostamento un frame.

L'idea è questa:

```
K = traiettoria(tipo,h,t0,tf,nt);

for n=1:nphi


    [Up, i, j]=proiezione(M, K(:,n));

    U(n,:) = Up(1,:);
    V(n,:) = Up(2,:);
end
```

Ogni colonna della matrice K rappresenta un diverso punto di vista della camera. Il parametro n_{phi} rappresenta il numero di angolazioni diverse, perciò la matrice K avrà dimensioni $3 \times n_{phi}$. La funzione 'proiezione' restituisce una matrice U_p utilizzata per la formazione delle matrici U e V che contengono rispettivamente le coordinate 2D dei punti orizzontali e verticali (u_p, v_p) (2.4).

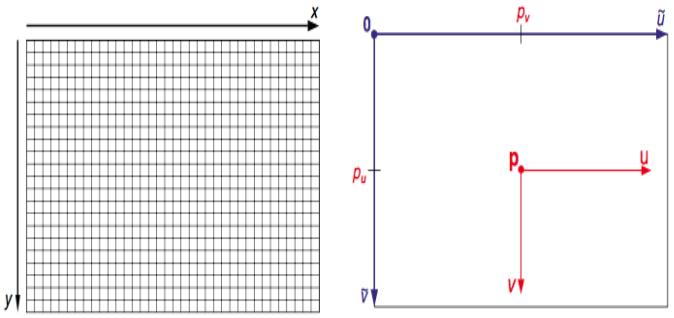

Figura 2.4: *Sinistra:* in un immagine digitale la posizione di un punto dell'immagine è indicata dalle sue coordinate in pixel, definite rispetto all'angolo superiore sinistro dell'immagine. *Destra:* le coordinate (u,v) della proiezione di un punto nell'immagine sono definite con riferimento al punto principale p.

Per quanto riguarda la traiettoria della camera ne abbiamo implementato 3:

1. **circolare**

$$\begin{cases} x = r\cos\phi \\ y = r\sin\phi \\ z = h \end{cases} \tag{2.8}$$

2. **lineare**

$$\begin{cases} x = t \\ y = y_0 \\ z = h \end{cases} \tag{2.9}$$

3. **parabolica**

$$\begin{cases} x = t \\ y = -t^2 + y_0 \\ z = h \end{cases} \tag{2.10}$$

Il codice MatLab utilizzato per la scelta della traiettoria della camera è il seguente:

```
function K = traiettoria(tipo,h,t0,tf,nt)

K = zeros(3,nt);
t = linspace(t0,tf,nt)';

switch tipo
    case 'circle'
        x = 0;
        y = 0;
        r = 2;
        for n = 1:nt
            K(:,n) = [x+r*cos(t(n)) y+r*sin(t(n)) h]';
            K(:,n) = K(:,n)/norm(K(:,n));

        end
```

```
16      case 'line'
            y0=2;
18          for n = 1:nt

20              K(:,n)= [t(n) y0 h]';
                K(:,n)= K(:,n)/norm(K(:,n));
22          end
        case 'parabola'
24          y0=0;
            for n=1:nt
26              K(:,n)= [t(n) -t(n)^2+y0 h]';
                K(:,n) = K(:,n)/norm(K(:,n));
28          end

30      otherwise
            error('unknown type')
32  end
```

La dimensione temporale in Computer Vision è importante per due ragioni. Primo, il movimento apparente degli oggetti sul piano dell'immagine è un forte segnale visivo per la comprensione della struttura e del movimento 3D. Secondo, i sistemi visivi biologici usano il movimento visivo per capire le proprietà del mondo 3D con delle piccole conoscenze a priori su di esso.

L'analisi del movimento porta con se dei problemi legati anche alle analogie con il mondo Stereo. I problemi principali sono:

Corrispondenza: quali elementi di un frame corrispondono agli elementi del frame successivo della sequenza?

Ricostruzione: dato un numero di elementi corrispondenti, conoscendo possibilmente i parametri intrinseci della camera, cosa possiamo dire riguardo il movimento e la struttura 3D dell'oggetto osservato?

Invece le principali differenze con il mondo Stereo sono:

Corrispondenza: le sequenze di immagini vengono campionate temporalmente a tassi generalmente elevati, per questo le differenze spaziali tra fotogrammi consecutivi sono, in media, molto inferiori a quelle tipiche delle coppie stereo.

Ricostruzione: diversamente dallo stereo, lo spostamento relativo tra il punto di vista della camera e la scena non è necessariamente causato da una singola trasformazione rigida 3D.

Il fatto che le sequenze di immagini posseggono molti frames strettamente campionati e disponibili per l'analisi presenta un grosso vantaggio sul caso stereo per almeno due ragioni. Primo, l'approccio *feature-based* può essere più efficace con l'uso di tecniche di *tracking*, che sfruttano la storia passata del moto delle caratteristiche per prevedere le disparità nel fotogramma successivo. Secondo, a causa delle lievi differenze spaziali e temporali tra frames consecutivi, il problema della corrispondenza può essere lanciato come il problema della stima del moto apparente del modello di luminosità dell'immagine, comunemente chiamato *Optical Flow*.
Il problema della ricostruzione è molto più difficile nel Multiview che nello Stereo. Anche con un solo movimento tra il punto di vista della camera e la scena, il recupero del movimento e della struttura 3D *frame-by-frame* risulta essere più sensibile al rumore.
Per risolvere i problemi del Multiview un utile punto di partenza è la stima del moto. Le numerose tecniche sono state divise dalla comunità della Computer Vision in due grandi classi principali:

Tecnica differenziale: basata sulle variazioni temporali e spaziali dell'illuminazione dell'immagine a ogni pixel. É considerato un metodo per l'Optical Flow.

Tecnica della corrispondeza (Matching Technique): stima la differenza tra determinati punti dell'immagine (features) tra un frame e quello successivo.

In questo lavoro di tesi verrà utilizzata la seconda tecnica.

2.2.2 Caratteristiche (features) dell'immagine

In Computer Vision il termine *image features* si riferisce a due possibili entità:

1. *una proprietà globale dell'immagine*, ad esempio il livello medio di grigio oppure l'area in pixel (global features);

2. *una parte dell'immagine con qualche proprietà speciale*, per esempio un cerchio, una linea, oppure una particolare superficie dell'immagine (local features);

La sequenza delle operazioni nella maggior parte dei sistemi di Computer Vision iniziano con la ricerca e la locazione di punti caratteristici nelle immagini di input. Le caratteristiche dell'immagine sono:

Significative, ovvero che le caratteristiche sono associate a un elemento della scena interessante. Ad esempio la variazione del livello di grigio nell'immagine.

Rilevabili, significa che l'algoritmo di locazione deve esistere. Differenti caratteristiche vengono associate a un differente algoritmo di rilevazione; in generale ciascun algoritmo specifica la posizione e altre proprietà essenziali dell'immagine.

É necessario precisare che nel 3D Computer Vision l'estrazione delle caratteristiche è solo uno step intermedio, non l'intero obiettivo. Non vengono estratte linee solo per avere una mappa delle linee; vengono estratte per guidare un robot in un corridoio, per decidere se l'immagine contiene un certo oggetto, per calibrare i parametri intrinseci ed estrinseci della camera, e così via. La cosa importante è che *non ha senso ricavare delle features perfette* in quanto l'adeguatezza dell'algoritmo deve essere valutata nel contesto dell'intero sistema. Ovviamente deve essere applicato un criterio di performance ragionevole per gli algoritmi che si occupano di estrazione delle features.
L'algoritmo più semplice si occupa dell'estrazione dei punti di bordo (edge detection), i quali non sono altro che pixel in cui l'immagine subisce brusche variazioni. L'interesse in questi punti ha una ragione precisa, i contorni di un elemento di una scena di potenziale interesse, come oggetti solidi, generano dei forti punti di bordo che vengono poi utilizzati per la ricostruzione degli oggetti. Esistono poi degli algoritmi più complessi che si occupano della rilevazione ed estrazione di tutti i punti (non solo di quelli ai bordi) che caratterizzano l'immagine.
Il concetto di punto chiave richiama il fatto che, non tutti, ma solo alcuni punti dell'immagine hanno una probabilità elevata di essere individuati senza ambiguità durante un confronto. Sono punti notevoli, stabili, facilmente individuabili. Nell'ultima decade, come in quasi tutti i campi della Visione Computazionale, sono stati fatti grandi passi in avanti nello sviluppo di local invariant features, punti caratteristici che permettono alle applicazioni di definire una geometria locale dell'immagine e codificarla in maniera che sia invariante alle trasformazioni dell'immagine, quali traslazione, rotazione, scala e deformazioni affini.
Quando però si ha a che fare con un flusso di immagini, come nel caso del Multiview, nasce un problema: la corrispondenza tra i punti caratteristici di un frame e quelli del frame successivo, quindi tra una coppia di immagini analogamente al caso Stereo. É necessario fare due assunzioni: primo, la maggior parte dei punti della scena devono essere visibili da entrambi i punti di vista; secondo, le regioni corrispondenti sono simili. Generalmente però

entrambe le assunzioni risultano false e il problema della corrispondenza diventa considerevolmente più difficile. Per iniziare possiamo prenderle come valide e vedere il problema della corrispondenza come un problema di ricerca: dato un elemento dell'immagine sinistra cercare l'elemento corrispondente in quella destra. Questo implica due decisioni, quali elementi dell'immagine confrontare e che misura adottare. Per convenienza gli algoritmi di corrispondenza sono classificati in due classi: *correlation-based* e *feature-based*. Noi ci soffermeremo solo ad uno studio del secondo metodo.

2.3 Estrazione dati 2D da immagini digitali

La tecnica *Matching* fa una stima del moto lavorando solamente con i punti caratteristici dell'immagine (features). Il problema sta proprio nel determinare queste features e la loro corrispondenza tra frames consecutivi.
La maggior parte dei metodi tendono a restringere il numero di possibili corrispondenze per ogni feature facendo rispettare dei vincoli adeguati. Questi vincoli possono essere:

- *geometrici*, vincoli epipolari;
- *analitici*, ad esempio il vincolo di unicità (ogni feature può avere al massimo una corrispondenza), o il vincolo di continuità (lo scarto varia in modo continuo in tutta l'immagine);

Sfortunatamente non esiste un metodo per la corrispondenza prestabilito che dia risultati ottimali in ogni circostanza. La scelta dipende da fattori come il campo di applicazione, l'hardware o il software a disposizione.
Detto ciò, può essere utile fare alcune considerazioni. Il metodo feature-based è adatto quando si conoscono a priori delle informazioni che riguardano l'immagine, solo così si possono ottenere delle features ottimali. Si può inoltre rivelare più veloce del metodo correlation-based, un altro vantaggio è che risulta essere relativamente insensibile alla variazioni di illuminazione.

Geometria epipolare. lo scopo del Multiview è quello di ricostruire la struttura di una scena 3D a partire da un numero sufficiente di viste. Possiamo pensare al mondo del Multiview come tante visioni stereo, e quindi trattare il problema della corrispondenza tra due frames consecutivi (quindi la corrispondenza tra 1° e 2° frame, tra 2° e 3°, ...). La geometria epipolare descrive le relazioni e i vincoli geometrici che legano due immagini 2D della stessa scena 3D catturata da fotocamere

con posizione e orientamento distinto. Immaginiamo di voler fotografare un elemento nello spazio 3D nel punto P tramite due fotocamere centrate in O_l e O_r. Il punto in questione verrà rispettivamente proiettato sul piano immagine della fotocamera di sinistra in π_l e π_r nel piano dell'immagine della fotocamera destra.

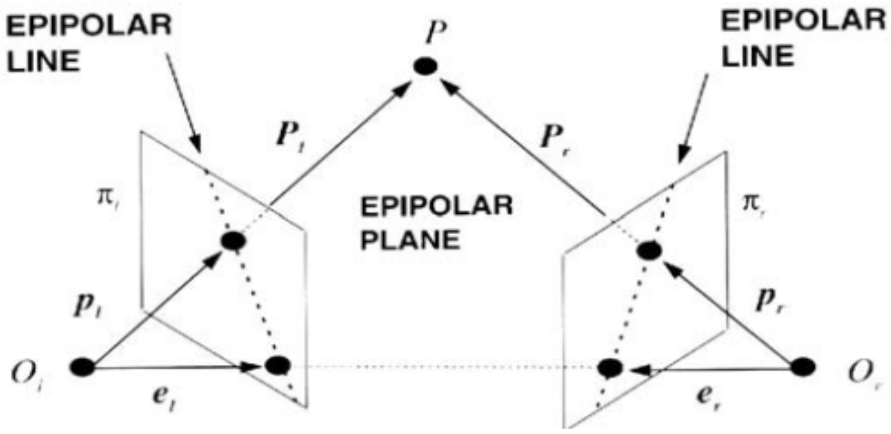

Figura 2.5: Geometria Epipolare

Gli **epipoli** e_l e e_r sono rispettivamente le proiezioni dei centri di proiezione O_l e O_r di ciascuna camera, l'uno sul piano immagine dell'altro, in due istanti successivi (vedi figura(2.5)).
I punti P**,** p_l e p_r stanno sullo stesso piano: *piano epipolare.*
La geometria epipolare ricopre un ruolo fondamentale nella ricostruzione di una scena tridimensionale a partire da una coppia di immagini. La ricostruzione avviene tramite i seguenti passi:

1. Data una lista di punti corrispondenti nelle due immagini viene calcolata la matrice fondamentale F ovvero una matrice 3×3 di rango 2 tale che $\pi_l^T F \pi_r = 0$.
2. Dalla matrice fondamentale vengono ricavate le matrici che rappresentano le trasformazioni proiettive delle due camere tramite le relazioni:

   $$P_l = [I|0]$$

   e

   $$P_r = [e_r \times F|e_r]$$

3. Per ogni coppia di punti corrispondenti nell'immagine si stima il punto P tramite le informazioni ricavate dalle due matrici di proiezione.

Vincolo epipolare: dato un punto sul primo piano immagine, il corrispondente sul secondo piano immagine deve giacere sulla retta epipolare (*epipolar line*). Il vantaggio del vincolo epipolare è che ci permettedi ridurre il problema della ricerca delle corrispondenze, in origine bidimensionale, a una dimensione.

Le performance di ogni metodo di corrispondenza è compromessa da *occlusioni* (punti che non hanno una controparte nel frame successivo), e *corrispondenze spurie* (false corrispondenze create dal rumore). Gli effetti di questo tipo di problemi possono essere ridotti con l'assunzione dei vincoli sopracitati.

Esistono molte tecniche per estrarre automaticamente da un'immagine punti salienti. Il software MATLAB che abbiamo utilizzato possiede un toolbox *"Computer Vision"* i cui pacchetti permettono numerose operazioni su immagini e video, tra queste troviamo algoritmi dedicati all'estrazione di punti chiave (**keypoint detection**), la loro caratterizzazione (**feature description**) e infine il confronto (**matching**).
Un elenco, non esaustivo, di algoritmi per individuare punti chiave è:

Harris Corner. Harris formalizza, da un punto di vista matematico, il concetto di bordo e, attraverso lo studio degli autovalori della matrice di covarianza nell'intorno di un punto, permette di ricavare la presenza o meno di uno spigolo. È invariante rispetto a cambiamenti di luminosità, a trasformazioni geometriche quali traslazioni e rotazioni, e minimamente a variazioni di scala;

KLT. Il Kanade-Lucas-Tomasi sfrutta una variante di Harris (Shi-Tomasi) come *corner detector* ed esegue il confronto sfruttando rappresentazioni piramidali della scena;

AST. La classe degli *Advance Segment Test* identifica un punto caratteristico osservando la differenza di luminosità dei punti su una circonferenza;

Durante il nostro studio abbiamo generato $n_{ph}i$ frames del nostro oggetto, il cubo, dopodiché con la funzione MATLAB 'getframe' e 'movie' abbiamo trasformato il flusso di frames in un filmato in questo modo:

```
for n=1:nphi;
    figure(1)
     A(:,n) = getframe;

end
 movie(A);

 movie2avi(A, 'myCube.avi', 'compression', 'None');

 [X, map]= frame2im(A(:,1));

 corners = detectFASTFeatures(X(:,:,1));
```

La necessità di trasformare i frames in un video nasce dal fatto che per la ricerca dei punti corrispondenti in un flusso di frames, il toolbox "Computer Vision" mette a disposizione un algoritmo chiamato ***vision.PointTracker System object*** il quale traccia una serie di punti usando l'algoritmo KLT, ma per farlo in maniera ottimale ha bisogno appunto di un video con estensione *.avi* e di una collezione di punti caratteristici da cui partire, esattamente quelli del primo frame del video.
I punti caratteristici di un singolo frame vengono recuperati grazie ad algoritmi dedicati. Questi permettono inoltre di rilevare oggetti, fare una stima del movimento e consentono inoltre di trattare il problema della rotazione e dell'occlusione. Il Computer Vision System Toolbox^{TM} offre vari rivelatori veloci quali Harris e Shi-Tomasi.

I pacchetti contenuti nel Toolbook permettono una miriade di azioni, quelle di interesse per la ricerca delle features sono contenute nel pacchetto***Features Detection and Extraction***. Tutte le funzioni restituiscono una matrice di punti con caratteristiche diverse, principalmente *corner points* (punti di angolo), rilevati dall'immagine 2D in scala di grigio in ingresso.
Abbiamo iniziato testando diverse funzioni sul poliedro perfetto di cui conosciamo tutto a priori, in modo da capire la validità dell'algoritmo, e abbiamo ottenuto questi risultati:

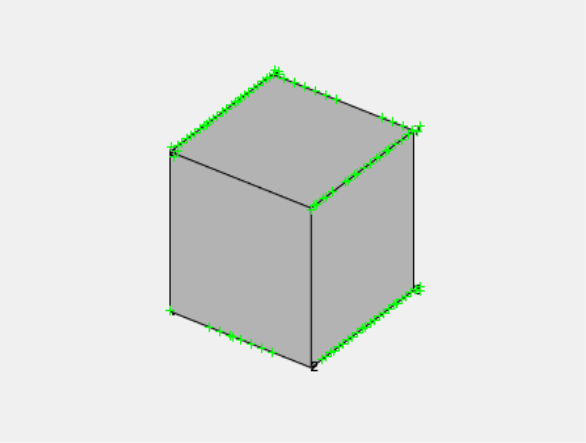

Figura 2.6: Funzione detectFASTFeatures

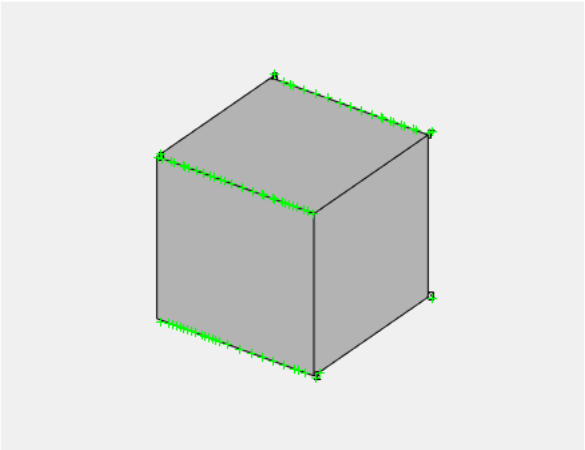

Figura 2.7: Funzione detectHarrisFeatures

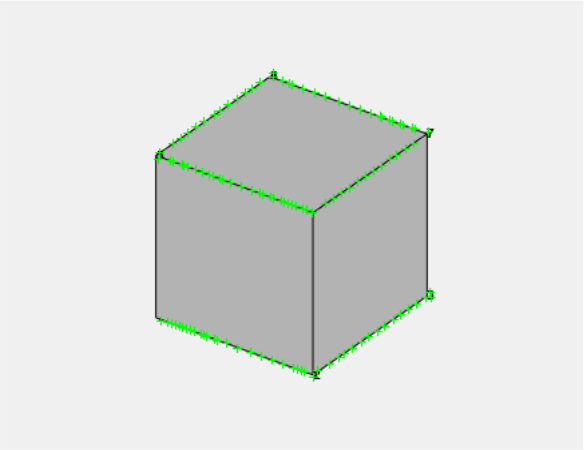

Figura 2.8: Funzione detectMinEigenFeatures

Come possiamo notare dalle immagini la funzione che si comporta meglio, a parità di parametri (abbiamo imposto la rivelazione di 1000 punti) e per i nostri scopi, è *detectMinEigenFeatures* che utilizza l'algoritmo del minimo autovalore e restituisce una matrice di cornerPoints dell'oggetto.

La situazione si complica quando analizziamo immagini reali.
Continuiamo con l'utilizzo della funzione che è stata più efficace nel caso ideale, i risultati sono stati ottenuti, modificando i parametri di rivelazione a 5000, sono mostrati nelle figure2.9 2.10.
Possiamo osservare che se l'immagine contiene molti particolari l'algoritmo non è in grado di distinguere quali siano i punti salienti, e questo può essere dovuto al fatto che non è particolarmente sensibile al cambiamento brusco di colore sulla scala di grigio (vedi Figura2.9).
Mentre se l'immagine è più semplice (Figura2.10) riesce chiaramente a rilevare, anche in modo abbastanza preciso, i punti di maggiore interesse ai fini della ricostruzione.

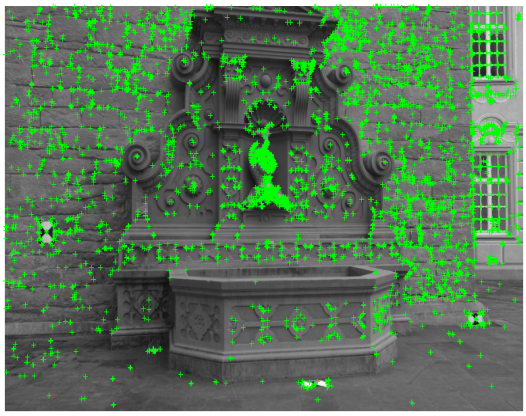

Figura 2.9: Fontana

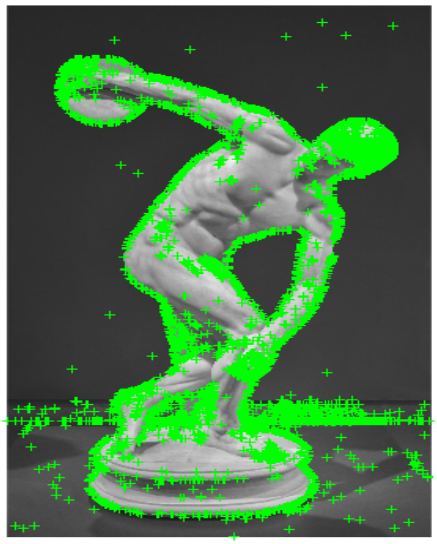

Figura 2.10: Discobolo

2.3.1 Corrispondenze

L'algoritmo che ci permette di comprendere tutti i problemi relativi alla rilevazione delle corrispondenze, contenuto nel Computer Vision Toolbox^{TM} di MALTLAB, è stato elaborato da ***Lucas e Kanade*** nel 1991. Di seguito una breve spiegazione del problema.

Possiamo, in maniera semplice, intendere il campo di moto come un'approssimazione del cosiddetto flusso ottico. Più precisamente di intende la proiezione del vettore velocità di un punto nello spazio tridimensionale sul frame (bidimensionale). Il task diventa quindi stimare questo vettore per poterlo utilizzare in applicazioni come il tracking di punti nel caso di Multiview.
Consideriamo la ricerca di corrispondenza come l'analisi del moto di un punto (x, y) in funzione di t, ipotizzando che l'intorno di (x, y) conservi i valori di I (livelli di grigio).
Scriviamo la derivata totale di I rispetto a t:

$$\frac{dI}{dt} = \frac{\partial I}{\partial t} + \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt}$$

Tale derivata si annulla per l'ipotesi di conservazione dei grigi.
Introducendo i vettori:

$$\Delta I = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}$$

$$\mu = \begin{bmatrix} \frac{dx}{dt} \frac{dy}{dt} \end{bmatrix}$$

si arriva a :

$$\Delta I \cdot \mu = -\frac{\partial I}{\partial t}$$

che è l'equazione del *flusso ottico*.
L'equazione del flusso ottico è valida solo per piccoli intervalli di tempo (o piccoli moti) tali per cui $|\mu| \sim 1$. Essa consente solo il calcolo della componente del vettore di moto nella direzione del gradiente dell'immagine.
La logica che si segue è la seguente:

- Risolvere le equazioni differenziali per ogni punto dell'immagine.
- Calcolare le derivate seconde dell'equazione di costanza ($dI/dt = 0$) della luminosità e risolvere per ogni punto dell'immagine.

- Approssimazione lineare e stima locale ai minimi quadrati (soluzione semplice ed efficace).
- Estrazione di features significative e calcolo del loro spostamento.

Abbiamo applicato quest'algoritmo solamente a un oggetto ideale (il poliedro) per cercare di capire la sua efficacia per lo sviluppo della Multiview.

Capitolo 3

Ricostruzione

Dopo aver illustrato i metodi per risolvere il problema della corrispondenza, possiamo affermare che una ricostruzione 3D robusta può essere ottenuta solo quando si ha un numero di informazioni sufficiente da dare in input all'algoritmo della fattorizzazione.

3.1 Metodo della Fattorizzazione

Data la stima del moto da una sequenza di immagini, è necessario calcolare la *forma* degli oggetti visibili e il loro *movimento* rispetto alla posizione della camera. Questo può essere fatto partendo da un set di immagini con punti caratteristici corrispondenti.
Se l'intervallo di tempo medio tra un frame e il successivo è piccolo la ricostruzione guadagna stabilità e robustezza. Per trovare una soluzione, tra i vari metodi proposti in letteratura, abbiamo scelto il *metodo della fattorizzazione*, semplice da implementare con ottimi (e numericamente stabili) risultati per qualsiasi tipo di oggetto a qualsiasi distanza.

Assunzioni:

1. Il modello della camera è ortografico.
2. La posizione degli n punti dell'immagine, corrispondenti ai punti della scena P_1,P_2...P_n non tutti complanari, sono stati tracciati in N frames, con $N \geq 3$.

Si noti che la seconda assunzione è equivalente all'acquisizione dell'intera sequenza prima di iniziare qualsiasi elaborazione. Questo può, o non può, es-

sere accettabile a seconda dell'applicazione. Si noti inoltre che, dal momento che il modello di fotocamera è ortogonale, la calibrazione può essere del tutto ignorata se accettiamo di ricostruire i punti 3D solo fino ad un certo fattore di scala.

Notazione. sia $p_{\text{fp}} = [u_{\text{fp}}, v_{\text{fp}}]^T$, dove f è il generico frame ($f = 1, ..., F$) e p la generica feature dell'immagine ($p = 1, ..., P$). Scriveremo le coordinate orizzontali della feature u_{fp} in una matrice $U = F \times P$, useremo ciascuna riga per il singolo frame e ogni colonna rappresenterà un punto caratteristico. Allo stesso modo una matrice $V = F \times P$ verrà generata dalle coordinate verticali v_{fp}. La matrice combinazione avrà dimensione $2F \times P$:

$$W = \begin{bmatrix} \text{U} \\ \text{V} \end{bmatrix}$$

W viene chiamata *matrice delle misure.*
Per ragioni che saranno chiare a breve, verrà sottratta la media degli elementi della stessa riga per ogni u_{fp} e v_{fp} in questo modo:

$$\tilde{u}_{fp} = u_{fp} - \bar{u}_{fp} \tag{3.1}$$

$$\tilde{v}_{fp} = v_{fp} - \bar{v}_{fp} \tag{3.2}$$

dove:

$$\bar{u}_f = \frac{1}{P} \sum_{p=1}^{n} u_{fp} \tag{3.3}$$

$$\bar{v}_f = \frac{1}{P} \sum_{p=1}^{n} v_{fp} \tag{3.4}$$

Questo produce due nuove matrici $F \times P$:

$$\tilde{U} = [\tilde{u}_{fp}]$$

$$\tilde{V} = [\tilde{v}_{fp}]$$

che generano la matrice $\tilde{W}$, chiamata *matrice delle misure registrata*:

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix}$$

e rappresenta l'input del metodo della fattorizzazione.

3.1.1 Teorema del rango

Il metodo della fattorizzazione è basato sulla dimostrazione di un semplice ma fondamentale risultato. Si tratta di analizzare la relazione tra il movimento della camera, la forma, e gli elementi della matrice $\tilde{W}$. Questa analisi porta al risultato chiave che $\tilde{W}$ è altamente *rank deficient*, ovvero non tutte le colonne sono esattamente indipendenti.
In riferimento alla Figura(3.1), supponiamo di mettere l'origine del sistema di riferimento del mondo (x,y,z) al baricentro dei P punti, in uno spazio che corrisponde ai P punti caratteristici tracciati nel sistema immagine. L'orientamento del sistema di riferimento della camera, corrispondente al frame numero f, è determinato da una coppia di vettori unitari i_f e j_f, che giacciono rispettivamente sul bordo orizzontale e verticale del frame.

Sotto l'ipotesi di ortografia, tutti i raggi di proiezione sono paralleli al prodotto vettoriale di i_f e j_f:

$$k_f = i_f \times j_f$$

Dalla Figura(3.1) possiamo vedere che l'origine delle coordinate del mondo sono poste nel baricentro dell'oggetto:

$$\frac{1}{P}\sum_{p=1}^{P} s_p = 0$$

$$s_p = (x_p, y_p., z_p)^T$$

con

$$p = 1, ..., P$$

Ora possiamo esprimere la matrice delle misure registrate $\tilde{W}$ nella forma:

$$\tilde{W} = RS \tag{3.5}$$

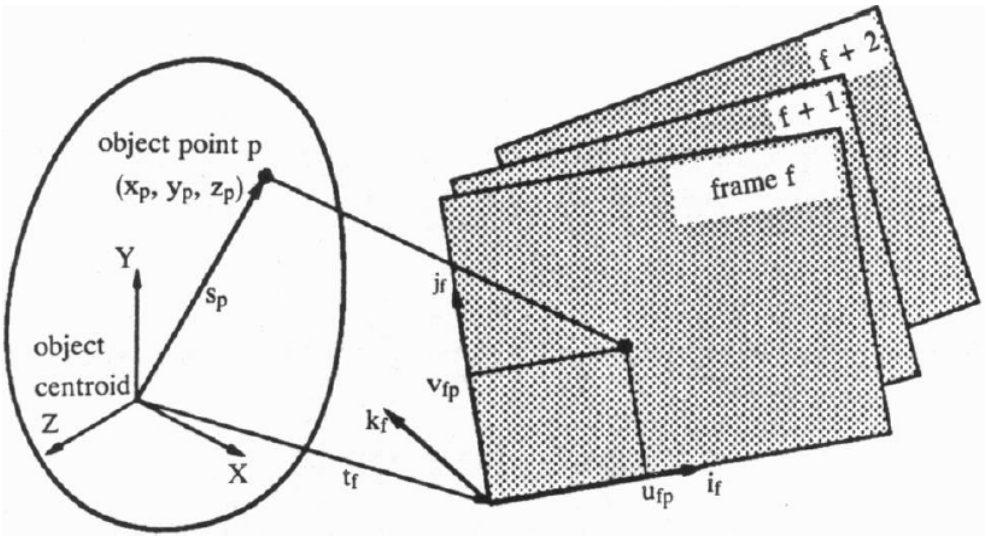

Figura 3.1: Sistema di riferimento usato nella formulazione del nostro problema

dove:

$$R = \begin{bmatrix} i_1^T \\ i_2^T \\ \dots \\ i_f^T \\ j_1^T \\ j_2^T \\ \dots \\ j_f^T \end{bmatrix} \tag{3.6}$$

rappresenta la rotazione della camera, e

$$S = [s_1, ..., s_P] \tag{3.7}$$

è la matrice della forma.

Le righe di R rappresentano gli orientamenti orizzontali e verticali e gli assi di riferimento della camera per tutto il flusso di frames, mentre le colonne

di S sono le coordinate dei P punti caratteristici rispetto al baricentro.
Poichè R è una matrice $2F \times P$ e S è $3 \times P$, Eq(3.5) implica questo:

Teorema 3.1.1 (Teorema del rango). *Senza rumore la matrice delle misure registrate $\tilde{W}$ è al massimo di rango 3.*

Il teorema del rango esprime il fatto che $2F \times P$ misure di immagini sono altamente ridondanti. Infatti, potrebbero tutte essere descritte in modo conciso con F frames e P vettori delle coordinate dei punti, solo se questi sono noti.
Nelle equazione (3.1) e (3.2), i_f e j_f sono vettori unitari mutuamente ortogonali, quindi devono soddisfare questi vincoli:

$$|i_f| = |j_f| = 1 \qquad i_f^T j_f = 0 \tag{3.8}$$

Inoltre, la matrice di rotazione R è unica se il sistema di riferimento per la soluzione è allineato con la posizione della prima camera, in modo che :

$$i_1 = (1, 0, 0)^T \quad e \quad j_1 = (0, 1, 0)^T \tag{3.9}$$

L'importanza del teorema del rango è duplice. In primo luogo, ci dice che c'è una grande quantità di dati dell'immagine ridondanti: non importa quanti punti e punti di vista si stanno prendendo in considerazione, il rango di $\tilde{W}$ non supererà 3. In secondo luogo, e fatto più importante, la fattorizzazione di $\tilde{W}$ come prodotto di S e R suggerisce un metodo per ricostruire la struttura e il moto da una sequenza di punti immagine caratteristici tracciati in precedenza.

Ci troviamo quindi di fronte a un sistema sovradeterminato, problema che può essere affrontato con il metodo ai minimi quadrati.

3.1.2 Problema ai minimi quadrati - decomposizione ai valori singolari (SVD)

In molti casi si incontrano sistemi lineari in cui il numero di equazioni è diverso dal numero delle incognite.
Dato un sistema:

$$Ax = b$$

dove $A \in (R^{m \times n})$, $x \in R^n$, $b \in R^m$.

Si possono verificare due casi:

1. Il numero di equazioni è maggiore di quello delle incognite ($m \geq n$), il sistema risulta sovradeterminato e potrebbero non trovarsi soluzioni.

2. Il numero di incognite è maggiore di quello delle equazioni ($m \leq n$), il sistema si dice sottodeterminato e potrebbe avere infinite soluzioni.

Il problema in questi casi si dice mal posto e non si possono trovare soluzioni in senso classico, è necessario quindi riformularlo in un problema ben posto. Se tutte le equazioni non possono essere verificate contemporaneamente, è ragionevole chiedere che lo scarto quadratico medio tra il primo e il secondo membro del sistema sia minimo.
Si passa così a un ***problema ai minimi quadrati***. La condizione di varianza minima si può esprimere nella forma:

$$\min_{x \in R^n} ||Ax - b| \, |_2$$

Se il minimo risulta essere zero, questo significa che il sistema originale ammette soluzione in senso classico. In caso contrario di ottiene la soluzione nel senso dei minimi quadrati del sistema lineare sovradeterminato, nel nostro caso:

$$\tilde{W} = RS$$

La matrice $\tilde{W}$ viene fuori dal calcolo illustrato nel paragrafo 2.2.1, mentre la matrice R, matrice di rotazione, è nota perchè abbiamo scelto noi il movimento della fotocamera. L'incognita è dunque la matrice della forma S.

Definizione 3.1. Un problema è ***ben posto*** se esso possiede, in un prefissato campo di definizione, una e una sola soluzione e questa dipende con continuità dai dati. In caso contrario, viene detto ***mal posto***.

Nel caso in cui la soluzione non dipenda con continuità dai dati, è possibile che anche una piccola perturbazione su di essi porti ad una soluzione diversa da quella esatta. Questo aspetto è fondamentale nell'ambito di dati sperimentali in quanto i dati misurati sono sempre affetti da errore. Il condizionamento misura quanto un errore nei dati possa essere amplificato nei risultati.

Definizione 3.2. Sia δ_d una perturbazione dei dati d di un problema e sia δ_x la corrispondente perturbazione sulla soluzione x. Sia inoltre $||\cdot||$ una qualsiasi norma vettoriale. Il ***numero di condizionamento assoluto*** $K(d)$ è definito dalla relazione:

$$||\delta_x|| \leq K\,||\delta_d||$$

mentre il ***numero di condizionamento relativo*** $k = k(d)$ verifica la diseguaglianza:

$$\frac{||\delta_x||}{||x||} \leq K\frac{||\delta_d||}{||d||}$$

É necessario quindi utilizzare un metodo che permette di trovare una soluzione approssimata a problemi mal-condizionati, uno di questi è il metodo della ***decomposizione ai valori singolari:***
sia $A \in R^{m\times n}$ una matrice rettangolare con $m \geq n$. Allora la decomposizione a valori singolari della matrice A avrà la forma:

$$A = U\Sigma V^T = \sum_{i=1}^{n} u_i\sigma_i v_i^T \tag{3.10}$$

- $U \in R^{m\times m}$ e $V \in R^{n\times n}$ sono matrici unitarie;
- $\Sigma \in R^{m\times n}$ è una matrice diagonale;
- $A = U\Sigma V^H$

$U = [u_1, ..., u_n]$ e $V = [v_1, ..., v_n]$ sono matrici con colonne ortonormali, $U^TU = V^TV = I_m$ e $\Sigma = diag(\sigma_1, ..., \sigma_n)$ è una matrice diagonale con elementi non-negativi in ordine non-crescente:

$$\sigma_1 \geq \cdots \geq \sigma_n \geq 0$$

Questi σ_i sono i valori singolari di A e i vettori u_i e v_i sono rispettivamente i valori singolari sinistri e destri di A. Il condizionamento della matrice equivale al rapporto tra il più grande valore singolare e il più piccolo non nullo: σ_i/σ_n.
Uno o più valori singolari piccoli implicano che la matrice A è quasi *rank-deficient* cioè non tutte le colonne di A sono esattamente linearmente indipendenti.

3.1.3 Algoritmo di fattorizzazione

La fattorizzazione di $\tilde{W}$ è relativamente lineare. Prima di tutto, questa fattorizzazione non è unica: se R e S fattorizzano $\tilde{W}$, e Q è una matrice invertibile 3×3, allora:

$$(RQ)(Q^{-1}S) = R(QQ^{-1})S = RS = \tilde{W}$$

a ciò vanno aggiunti due vincoli:

1. le righe di R, pensandole come vettori 3D, devono avere una norma unitaria;
2. le prime n righe di R (i_f^T) devono essere ortogonali alle corrispondenti ultime n righe (j_f^T).

Il nostro ultimo sforzo prima di completare l'algoritmo è quello di dimostrare che questi vincoli permettono di calcolare una fattorizzazione di $\tilde{W}$ che sia unica rispetto a un orientamento iniziale del frame sconosciuto. Allo stesso tempo, estenderemo il metodo nel caso in cui, a causa di rumore oppure corrispondenze imperfette, il rango della matrice $\tilde{W}$ sia maggiore di 3. Iniziamo con il considerare la SVD di $\tilde{W}$:

$$\tilde{W} = UDV^T \tag{3.11}$$

Il fatto che il rango di $\tilde{W}$ sia maggiore di 3 significa che ci sono più di tre valori singolari diversi da zero nella diagonale della matrice D.
Il teorema del rango può essere rinforzato semplicemente imponendo che i valori singolari dopo il terzo siano nulli e ricalcolando poi la matrice $\tilde{W}$ corretta dalla (3.11).
Se prestiamo attenzione solo alle prime 3 colonne di U, la prima submatrice 3×3 di D, e le prime tre righe di V^T in questo modo:

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \tag{3.12}$$

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \tag{3.13}$$

$$V^T = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{3.14}$$

abbiamo $UDV^T = U_1 D_1 V_1^T + U_2 D_2 V_2^T$.
Il secondo termine entra in gioco in caso di rumore.
Se il rapporto tra il terzo e il quarto valore singolare non è troppo grande, la (3.11) ci permette di ottenere risultati consistenti.

Nel nostro caso abbiamo a disposizione la matrice $\tilde{W}$ e siamo a conoscenza dei parametri della camera per cui è nota la matrice R, l'incognita è la matrice della forma S le cui colonne contengono le coordinate 3D dei punti dell'oggetto.

```
W = [U;V];
S=pinv(R)*W; % pinv calcola la pseudoinversa di R
             %utilizzando la SVD
```

Dove *pinv(R)* è la *matrice pseudoinversa* di R tale per vale la seguente definizione:

$$(A^T \cdot A)^{-1} \cdot A^T = A^+$$

Con A^+ matrice pseudoinversa.
Data una matrice $A \in \mathbb{R}^{m \times n}$ di rango massimo e la sua pseudoinversa A^+ valgono le seguenti proprioetà:

- $A \cdot A^+ \cdot A = A$;
- $A^+ \cdot A \cdot A^+ = A^+$;
- $(\alpha \cdot A)^+ = \alpha^{-1} \cdot A^+$. Dove $\alpha \neq 0$ è un numero reale.

Se la matrice A è già una matrice quadrata di rango pieno e quindi invertibile allora la sua pseudoinversa è uguale alla sua inversa ($A^{-1} = A^+$). Bisogna fare bene attenzione però di non pensare che la pseudoinversa di una matrice non quadrata sia l'equivalente dell'inversa di una matrice quadrata. Mentre l'inversa viene utilizzata per trovare la soluzione esatta di un sistema determinato, la pseudoinversa viene utilizzata per trovare un vettore che soddisfi il criterio dei minimi quadrati in un sistema sovradeterminato, vettore che non sarà in generale una soluzione esatta del sistema.

Essendo quindi una soluzione non esatta del sistema, abbiamo la necessità di valutare di quanto si discosta la matrice S che abbiamo ottenuto dalla matrice ideale M che abbiamo usato per descrivere il poliedro inizialmente.

L'errore viene ottenuto in questo modo utilizzando la *norma di Frobenius*:

$$||A||_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{i,j}|^2 \right)^{1/2}$$

non è altro che la norma euclidea applicata al vettore:

$$vec(A) = (a_{1,1}, a_{2,1}. \cdots, a_{2,n}, a_{1,2}, \cdots, a_{n,n})^T$$

che si ottiene sovrapponendo una sull'altra le colonne di A.

```
1 err= norm(S-M,'fro')
```

Quando si parla di dati sintetici i risultati ottenuti sono ottimi con una errore dell'ordine di 10^{-14} (praticamente nullo), questo è il caso del poliedro che abbiamo generato. Con DataSet reali non siamo certi che ciò accada a causa degli errori sperimentali. Ad esempio (vedi Tabella), simulando la presenza di un rumore gaussiano con il comando MATLAB $randn(n, m)$ (che genera una matrice di numeri distribuiti scondo una gaussiana da media 0 e larghezza 1) moltiplicato per una data deviazione standard, possiamo notare che la matrice W_n calcolata in presenza di rumore ha un errore rispetto alla stessa calcolata in assenza di rumore (W_n). Abbiamo anche valutato quanto si discosta la matrice della forma S che viene ricostruita rispetto a quella iniziale M che noi abbiamo generato.
Un altro problema che deriva dalla presenza di rumore, è che $\tilde{W}$ non ha più rango 3, condizione necessaria per avere una buona ricostruzione. Questa condizione può essere imposta utilizzando la decomposizione ai valori singolari troncata (TSVD).
La TSVD ci permette di approssimare una matrice, nel nostro caso affetta da disturbo, a una con rango non pieno ottenuta ponendo a zero tutti i valori singolari di interesse (per noi è necessario far andare a zero tutti i valori singolari $\sigma \geq 3$).

Rumore	$\|\|W - W_n\|\|$	$\|\|M - S_n\|\|$
10^{-1}	10^{0}	10^{-1}
10^{-2}	10^{-1}	10^{-2}
10^{-4}	10^{-3}	10^{-4}
10^{-9}	10^{-8}	10^{-9}

Possiamo osservare che le matrici affette da rumore si discostano da quelle calcolate in assenza di noise in modo costante rispetto all'errore, com'era ovvio aspettarsi. In particolare l'ordine di grandezza dell'errore riportato sulla matrice delle coordinate 3D ricostruite S_n, rispetto a quella in assenza di noise M, è esattamente uguale alla deviazione standard applicata.

3.2 Occlusioni

Uno dei problemi tipici della Computer Vision e più in particolare della Multiview è l'occlusione.
Nella realtà, quando la camera si muove, le *features* possono apparire e scomparire dalle immagini consecutive a causa dell'occlusione. Inoltre qualsiasi metodo di feature-tracking non sempre riuscirà a trovare tutti i punti caratteristici nel sistema immagine. Questo fenomeno è abbastanza frequente da rendere il calcolo di forma e movimento quasi irrealizzabile.
La sequenza con la quale le features appaiono e scompaiono fanno si che la matrice $\tilde{W}$ sia parzialmente completa. Comunque, solitamente ci sono informazioni sufficienti nel flusso di immagini che ci consentono di determinare tutte le posizioni della camera e tutte le coordinate 3D dei punti caratteristici dalla matrice $\tilde{W}$ incompleta, non solo, potremo anche riuscire a ricavare gli elementi ignoti che completano la matrice.
Supponiamo che un punto caratteristico non sia visibile in un certo frame. Se la stessa feature è visibile abbastanza spesso in altri frames, la sua posizione nello spazio può essere ricavata. Inoltre se il frame in questione include abbastanza punti caratteristici può essere ricavata anche la corrispondente posizione della camera. In questo modo abbiamo ricavato la feature occlusa e il punto di vista della camera e siamo in grado di ricostruire la matrice $\tilde{W}$ completa.
Formalmente, abbiamo la seguente condizione sufficiente:

> *Condizione di ricostruzione:* In assenza di rumore, una coppia di misure delle immagini ignote (u_{fp}, vfp) in un frame f può essere ricostruito se il punto p è visibile in almeno 3 o più frames f_1, f_2, f_3 e se ci sono almeno tre o più punti p_1, p_2, p_3 visibili in tutti e quattro i frames (il frame originale f e i frames f_1, f_2, f_3 aggiunti).

Basandoci su questo, è stato sviluppato un algoritmo in grado di recuperare la forma tridimensionale di una scena che è parzialmente occlusa nell'immagine in input (Tomasi - Kanade factorization).

Conclusioni

Concludendo possiamo dire che la difficoltà maggiore nello sviluppo del metodo Multiview, ovvero la ricerca dei punti caratteristici nell'immagine 2D e le loro corrispondenze nei frames, permane. Tuttavia abbiamo constatato che il problema è meno complesso quando trattiamo di immagini semplici. Questo accade perchè tutti gli algoritmi utilizzati sono sensibili a forti variazioni sulla scala di grigi, quando queste variazioni non sono abbastanza forti si ha la perdita di informazione in quanto la feature non viene rilevata e la ricostruzione non potrà essere fedele all'originale.
Abbiamo appurato che l'idea iniziale di confrontare la tecnica Multiview con quella Photometric Stereo risulta infondata dal momento che entrambe trattano il problema della ricostruzione in maniera radicalmente differente. Il metodo Multiview permette una ricostruzione a tutto tondo, seguendo il movimento di punti interessanti, ma non dettagliata come la ricostruzione delle faccia di un oggetto ricavata con la Photometric Stereo, che invece lavora secondo le variazioni di intensità di ciascun pixel.
Un interessante proseguo di questo lavoro sarebbe riuscire a fondere le due tecniche in modo da riuscire ad ottenere una ricostruzione estremamente dettagliata a 360^o.

Bibliografia

[1] Rodriguez G. *Algoritmi numerici* (2008).

[2] Trucco E. and Verri A.*Introductory techniques for 3-D Computer Vision* (1998).

[3] Rodriguez, G. and Seatzu, S. *Introduzione alla Matematica Applicata e Computazionale* (2010).

[4] Dessì R., Mannu C., Rodriguez G., Tanda G., Vanzi M. *Recent improvements in photometric stereo for rock art 3D imaging. Digital Applications in Archaeology and Cultural Heritage* (2015).

[5] Tomasi C. and Kanade T. *Shape and motion from image streams: A factorization method* (1993).

Ringraziamenti

Ringrazio innanzitutto Professor Rodriguez per avermi dato l'opportunità di lavorare con lui, per la pazienza e il tempo dedicatomi.
Non basterebbe un intero libro per ringraziare i miei genitori e mia sorella per tutti gli sforzi fatti in questi anni, per aver appoggiato ogni mia scelta anche quelle più difficili che richiedevano un sacrificio economico maggiore. Grazie per aver sopportato i miei sbalzi d'umore giornalieri di quest'ultimo periodo, e per l'impegno che mettete in tutto quello che fate. Senza la vostra educazione e il vostro esempio oggi non sarei la persona che sono.
Un ringraziamento sincero va anche ai miei nonni, per aver sempre ricoperto il ruolo di secondi genitori, garantendomi la loro presenza costante nella mia vita.
Sarebbe necessaria una pagina per ringraziare uno per uno tutti i miei amici, per esserci oggi, e per esserci stati in questi anni. Un ringraziamento particolare lo devo a Ninni, Simona, Serena, Cristina per aver sempre consolato ogni mia delusione, per esserci sempre stati nei momenti di bisogno e in quelli di svago.
Un grazie particolare ai miei colleghi Lorena, Romina, Francesca, Massimiliano per aver reso le giornate di studio più piacevoli, e a Giulia che pur non essendo una mia collega ha sempre cercato di aiutarmi, soprattutto con l'italiano.
Ringrazio la mia curiosità per avermi spinto a partecipare al progetto Erasmus e conoscere persone che l'hanno resa speciale: Mariagrazia e Gabriele.
Ci tengo a ringraziare tutti coloro che hanno preso un aereo, o un permesso a lavoro, solo per essere qui oggi e farmi sentire una persona importante.
Ultimo, ma non per importanza, un grazie immenso a Mattia per sopportarmi ogni giorno da sei anni a questa parte.