

Il seguente file costituisce esclusivamente un supporto alla preparazione dell'esame, al fine unico di ripasso generale: si raccomanda di studiare dalle slide del docente dagli appunti presi a lezione, unico materiale valido per la corretta preparazione dell'esame. Il riassunto è stato redatto da Florio Vincenzo Pio, con l'aiuto di Savoia Davide.

Artificial Vision

Cenni generali

Campo Visivo

Il **campo visivo** rappresenta l'**estensione angolare** dell'area che può essere osservata da un sistema. È misurato in multipli e sottomultipli di metro e determina la quantità di informazioni visibili all'interno dell'immagine catturata. Ai fini dell'esame si valuta la sua estensione lungo l'asse x (**FOVx**) e lungo l'asse delle y (**FOVy**), dove la relazione tra questi due dipende strettamente dal rapporto delle dimensioni lungo i due assi del sensore che usiamo per acquisire l'immagine.

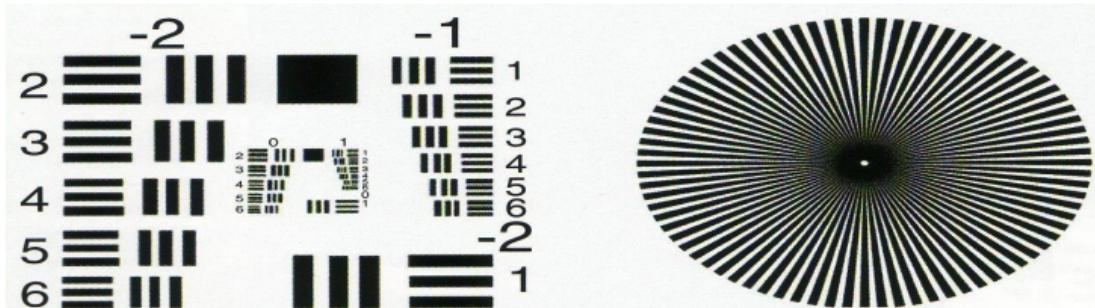
Distanza di lavoro

La **distanza di lavoro** si riferisce alla distanza fisica tra il sistema di artificial vision e l'oggetto che viene osservato o analizzato. Questa distanza può influenzare la **qualità** dell'immagine, la sua **nitidezza** e la **precisione** delle misurazioni. Qualora non ci siano vincoli particolari è necessario lavorare ad una distanza che ci permetta l'utilizzo di **focali intermedie**, **evitando** l'utilizzo di lenti **grandangolari**, che andrebbero a creare **distorsioni** evidenti soprattutto sui bordi. Si usa questa proporzione per calcolare l'incognita rispetto ai parametri noti:

$$\frac{Wd}{FOV} = \frac{f}{\dim Sens}$$

Risoluzione

La **risoluzione** indica la capacità di un sistema di **distinguere dettagli fini all'interno** dell'immagine. Una maggiore risoluzione corrisponde a una migliore capacità di discriminare dettagli più piccoli, fornendo immagini più nitide e dettagliate. Viene definita come coppie di righe/mm. Vengono usate delle immagini standard per valutare questo parametro:



Profondità di campo

La **profondità di campo** è la gamma di distanze focali all'interno della quale gli **oggetti** nell'immagine risultano **nitidi**. È importante per garantire che gli oggetti a diverse distanze dal sistema di artificial vision siano visibili in modo chiaro e definito.



La profondità di campo dipende da 3 fattori:

	Aumenta	Diminuisce
Apertura diaframma	Aperture ristrette (valori alti, es.: f22)	Aperture ampie (valori bassi, es.: f1.8)
Lunghezza Focale	Focali corte, grandangoli	Focali lunghe, teleobiettivi
Tipologia di sensore	Sensori piccoli	Sensori grandi
Distanza soggetto-sfondo	Soggetto vicino allo sfondo	Soggetto lontano dallo sfondo

Ingrandimento primario

L'**ingrandimento primario** rappresenta il **rapporto** tra le **dimensioni** dell'oggetto nell'immagine catturata e le sue **dimensioni reali**. È un fattore importante per la misurazione accurata degli oggetti e può essere espresso come un rapporto o una percentuale.

$$PMAG = \frac{\text{dim Sens}}{\text{FOV}}$$

Dimensione del sensore

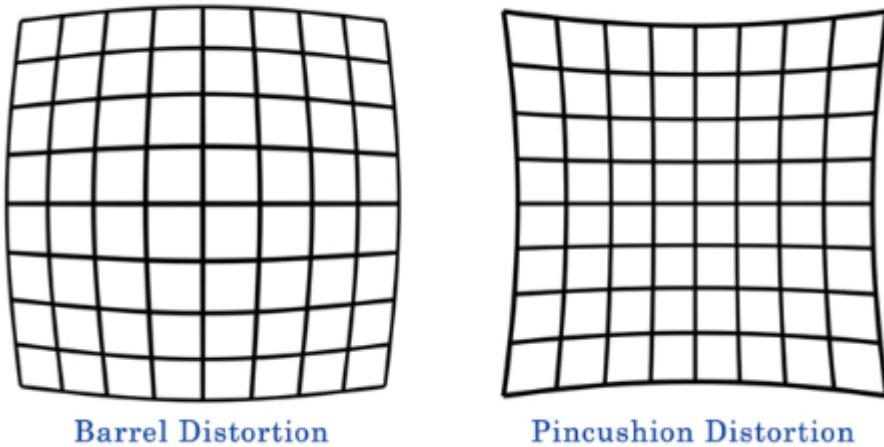
La **dimensione** del **sensore** rappresenta le dimensioni fisiche del sensore d'immagine utilizzato nel sistema di artificial vision. Le dimensioni del sensore possono influenzare la qualità dell'immagine, la sensibilità alla luce e altre caratteristiche dell'immagine catturata.

Contrasto

Il **contrasto** è la **differenza** tra i livelli di **luminosità** nell'immagine. Un contrasto maggiore indica una differenza più marcata tra le aree chiare e scure dell'immagine, fornendo una migliore definizione dei dettagli e una maggiore nitidezza. Questo chiaramente diminuisce all'aumentare della frequenza.

Distorsione

La **distorsione** rappresenta la **deformazione** dell'immagine rispetto all'oggetto reale. Può essere causata da fattori come le proprietà ottiche del sistema di artificial vision e può influenzare la precisione delle misurazioni e la fedeltà dell'immagine. C'è la possibilità di ridurla via software rimappando l'immagine empiricamente.

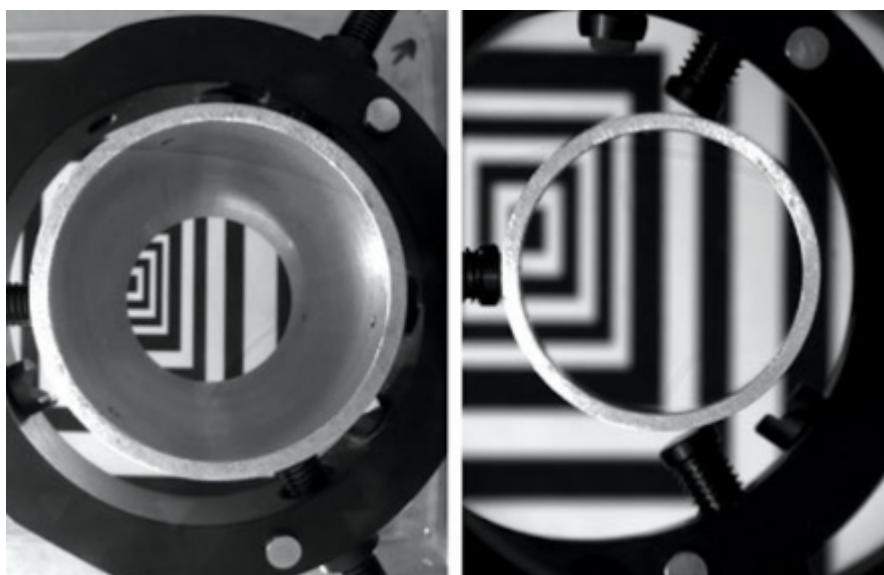


Parallasse

La **parallasse** è l'**effetto** visivo in cui gli oggetti sembrano spostarsi rispetto all'osservatore quando l'osservatore cambia posizione. Nella visione artificiale, la parallasse può influenzare la **percezione della profondità** e richiede un'adeguata calibrazione e compensazione.

Telecentricità

La **telecentricità** si riferisce alla **prospettiva costante** dell'immagine acquisita dal sistema, indipendentemente dalla distanza dell'oggetto. Questo significa che l'immagine mantiene le stesse proporzioni e dimensioni dell'oggetto reale, anche quando l'oggetto si avvicina o si allontana dalla fotocamera. Spesso le lenti però sono grandi e ingombranti.



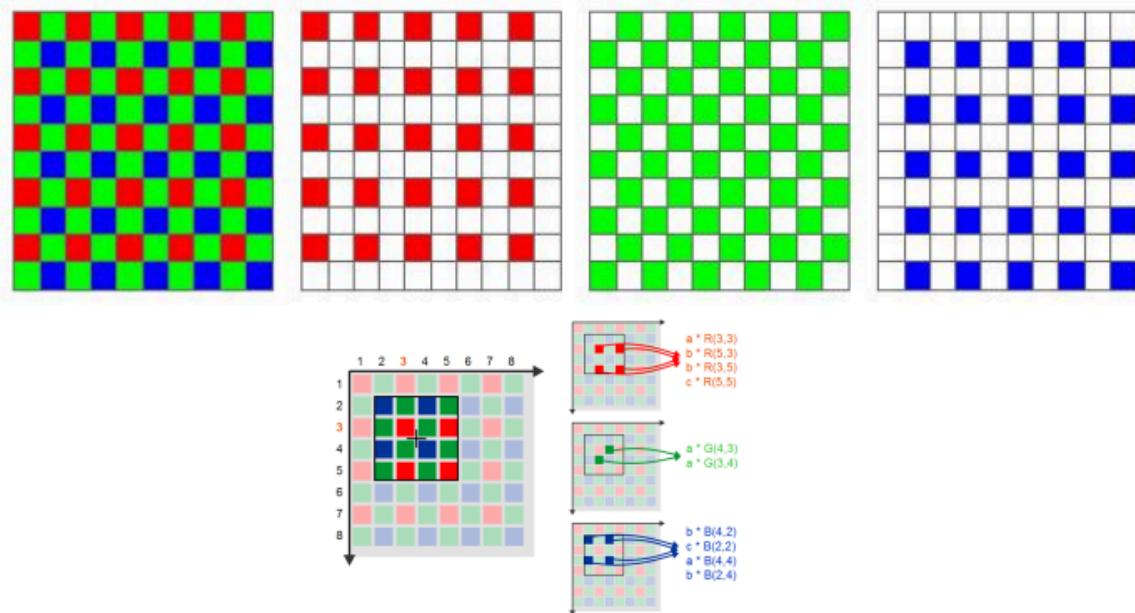
Sensori

Sensori Monocromatici

Sensori che catturano immagini nella **scala di grigi**, senza informazioni sull'aspetto cromatico. Hanno un'**elevata sensibilità** alla luce e sono adatti ad applicazioni dove è necessaria una precisione molto elevata

Pattern di Bayer

E' un **filtro** posto sul sensore che serve ad acquisire le immagini con i **colori**. Utilizza una matrice di filtri dei 3 colori primari (RGB) e richiede che siano applicati degli algoritmi di **demosaicing** per ottenere l'immagine una volta scattata



Sensori HDR

Mediante lo scatto di 3 foto (una sottoesposta, una sovraesposta e una correttamente esposta) si vanno a sommare le informazioni e si prendono soltanto quelle utili, permettendo un livello molto variegato di informazioni visive.



Camere matriciali

Sono le fotocamere che usiamo più comunemente, utilizzano sensori CCD o CMOS e usano una **matrice** di fotosensori per scattare una foto. Di seguito qualche caratteristica:

- Risoluzione da 640x480 a 6000x5000
- Framerate dipendente da risoluzione, sensore e interfaccia di collegamento
- Grandezza dei pixel da 2.5 a 14 um
- Funzionamento in free run o trigger con un segnale elettrico
- Sono tra le più comuni negli esercizi

Camere lineari

Fotocamere che acquisiscono immagini **lungo una linea**, mediante varie combinazioni di filtri di colore. Possono essere di diversi tipi e consentono una **frequenza di scatto molto più alta di quelle matriciali (centinaia di kHz)**. Di seguito qualche caratteristica:

- Risoluzione da 512 punti fino a 16k punti
- Framerate dipendente da risoluzione, sensore e interfaccia di collegamento
- Grandezza dei pixel da 3 a 14 um
- Funzionamento in free run o trigger con un segnale elettrico

Single line with RGB triples



Risoluzione ridotta di 1/3

Scarsa Separazione colori dovuta a filtro bayer

Scarsa sensibilità

Dual line



Risoluzione migliorata

Non ottima Separazione colori dovuta a filtro bayer

Sensibilità migliorata

Trilinear line



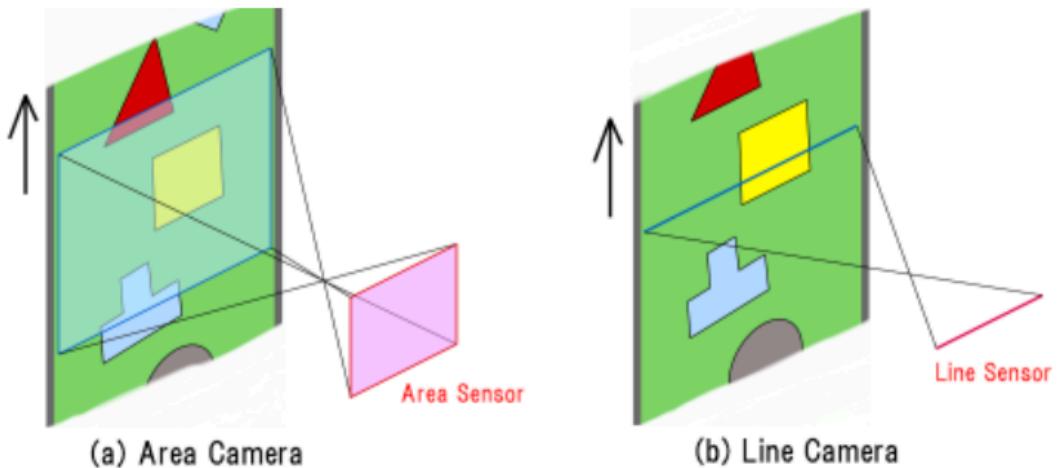
Risoluzione completa

Ottima separazione colori

Maggior complessità

Differenze tra Matriciali e Lineari

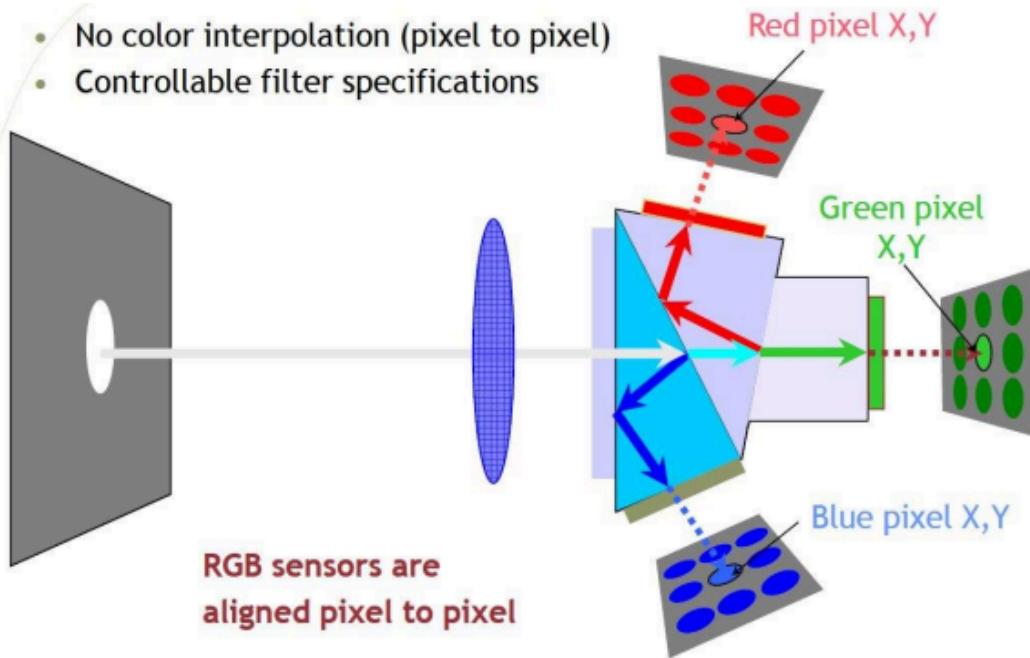
	Matriciali	Lineari
Framerate	<200fps	>50000fps
Sincronizzazione	Piccoli vincoli in base alla velocità del nastro	Sincronizzazione precisa



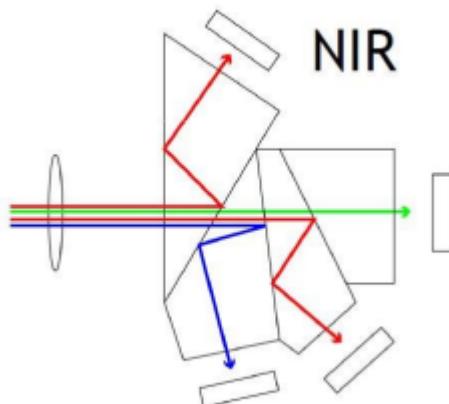
Camere multi-sensori

Fotocamere particolari che usano **3 sensori diversi** e un **prisma** per dividere le frequenze dei colori primari. Permettono di ottenere informazioni spettrali molto più dettagliate per le **bande** interessate.

- No color interpolation (pixel to pixel)
- Controllable filter specifications

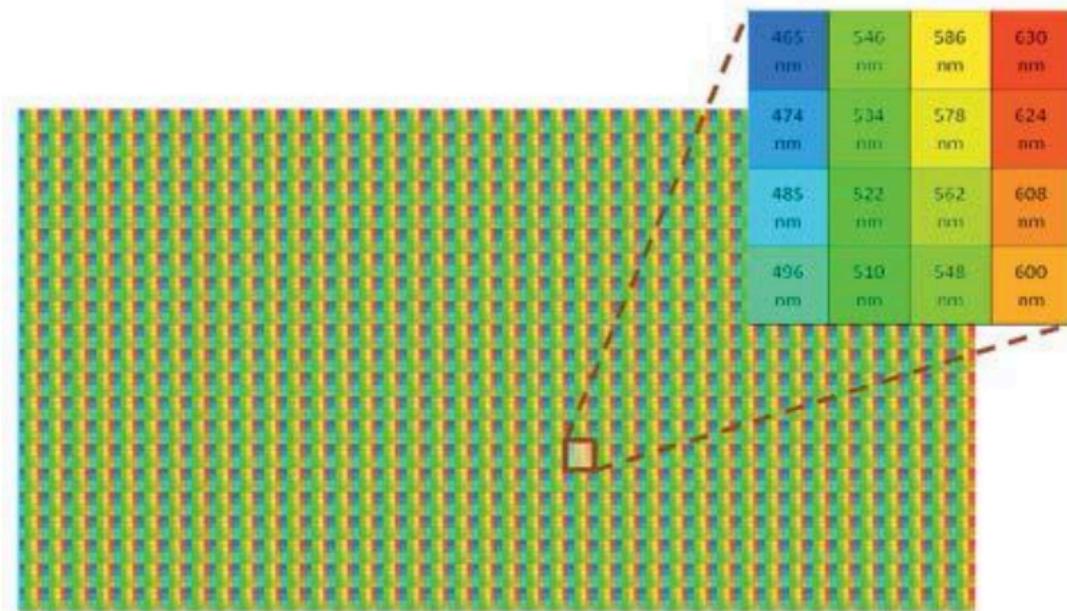


E' anche possibile prevedere un quarto sensore per NIR, in questo caso sarà un po' diversa la struttura:



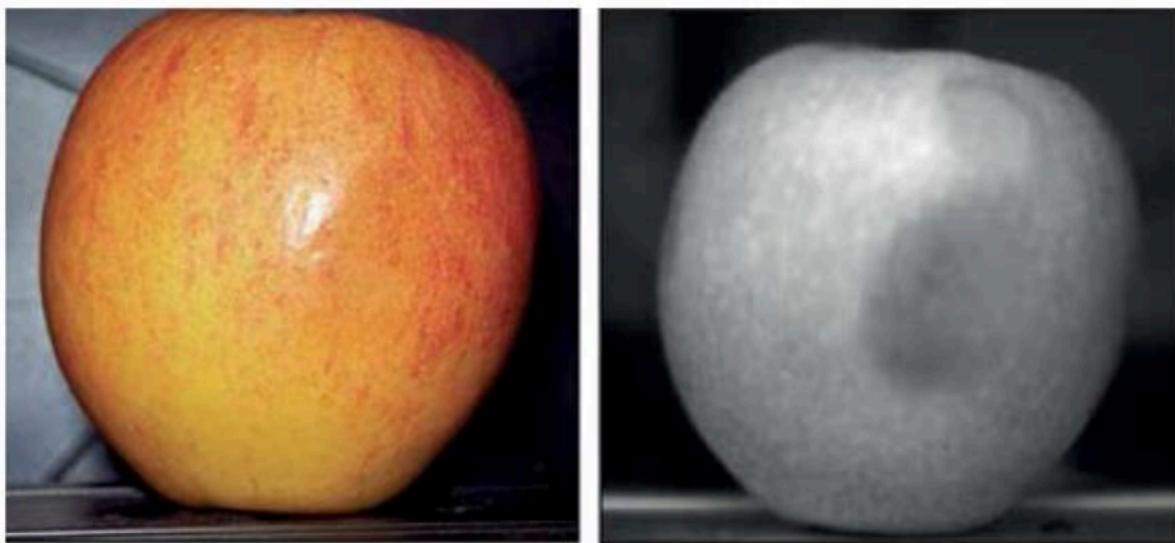
Camere multispettrali

Con la stessa idea nascono le fotocamere multispettrali, che permettono di raccogliere informazioni su diverse frequenze che ci interessano. In questo caso vengono impiegate nei settori più disparati, in particolar modo per informazioni su materiali, vegetazione e per altri tipi di analisi.



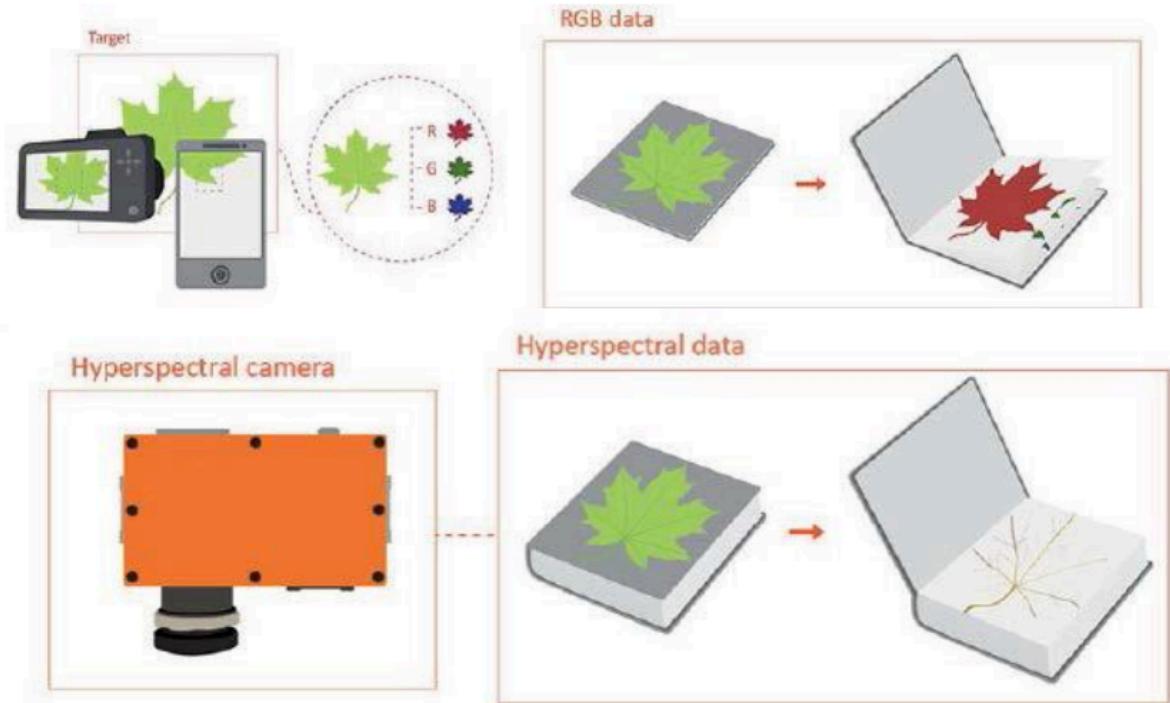
Camere NIR

Per NIR si intende una serie di frequenze **vicino l'infrarosso**, che ci consente di rilevare informazioni non apprezzabili nello spettro del visibile, soprattutto per utilizzi che hanno a che fare con il riconoscimento della vegetazione ed eventuali danni nei vegetali.



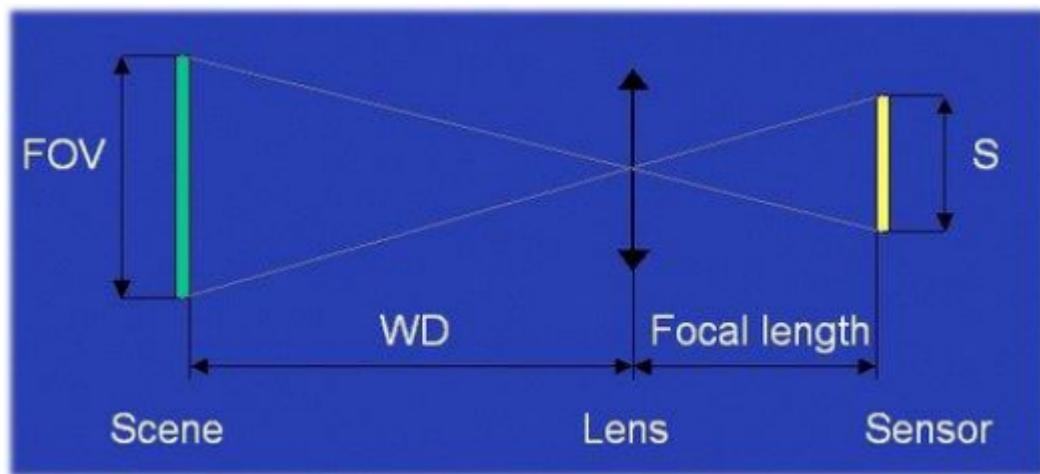
Camere iperspettrali

Le fotocamere iperspettrali acquisiscono immagini in un gran numero di frequenza, permettendo un'analisi molto precisa di determinate caratteristiche chimiche e fisiche di alcuni materiali.



Ottiche (lenti)

Ricordando la formula che ci permette di calcolare il FOV:



Ci sono alcune accortezze particolari che dobbiamo impiegare nella scelta della lente.

Mount

Ogni lente ha il suo **mount**. Questo serve a **collegare** la lente al corpo macchina e occorre che i due siano completamente compatibili per usufruire di tutto quello che ci offre la lente. Questo prevede determinati pin, di cui alcuni ad esempio per la messa a fuoco (se gestita dal sensore), che verrebbe persa qualora si usasse la lente su un corpo macchina non compatibile, attaccandola fisicamente al sensore.



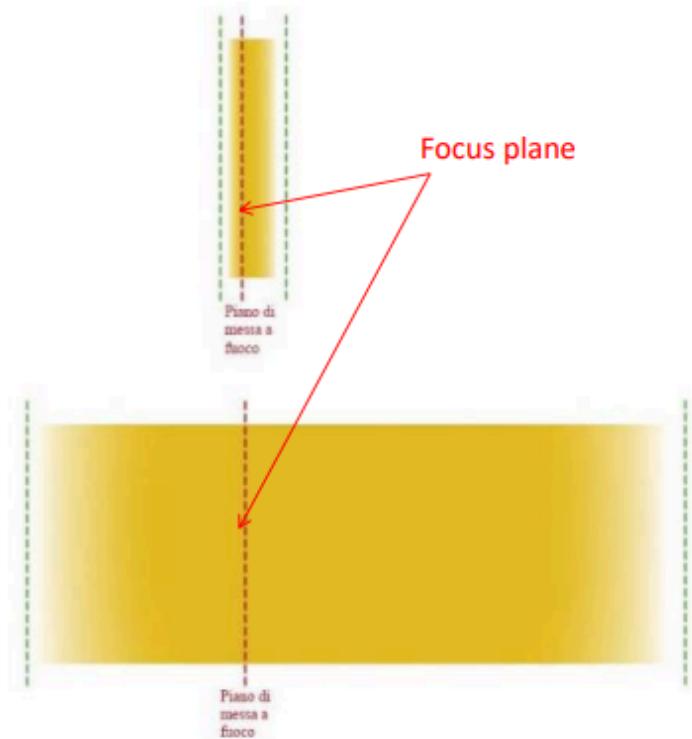
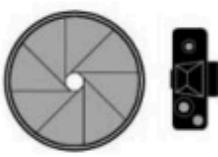
Dimensione del sensore

La **grandezza del sensore** è un vincolo fondamentale per la scelta della lente. Un sensore piccolo può sempre usare una lente creata per un sensore più grande, chiaramente utilizzerà soltanto una porzione centrale, andando di conseguenza a ridurre il FOV (a parità di focale). Chiaramente non è conveniente, in quanto una lente più grande sarà più costosa, ma se fosse necessario avrebbe senso. Al contrario (Sensore grande con una lente per un sensore piccolo), invece, il risultato non sarà accettabile, in quanto parte dell'immagine non sarà visibile e si soffrirebbero i soliti problemi visibili ai bordi della lente.



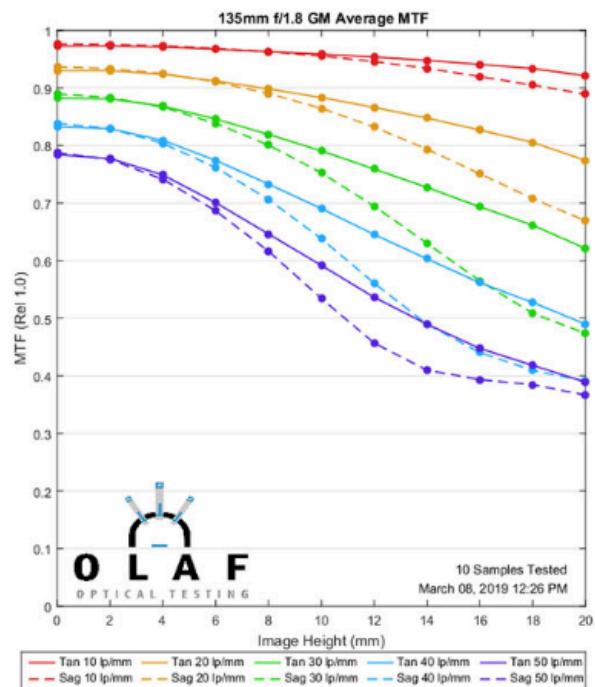
Apertura (e PdC)

Uno dei motivi per cui le lenti sono estremamente costose è l'**apertura massima** consentita dalla lente (il fattore "f" che si trova scritto davanti). Più è alto e maggiore è la luce che entrerà nel sensore. Per quanto non sia consigliabile scattare al valore massimo (il valore numerico minimo), in questo punto si potrà apprezzare la minima profondità di campo.

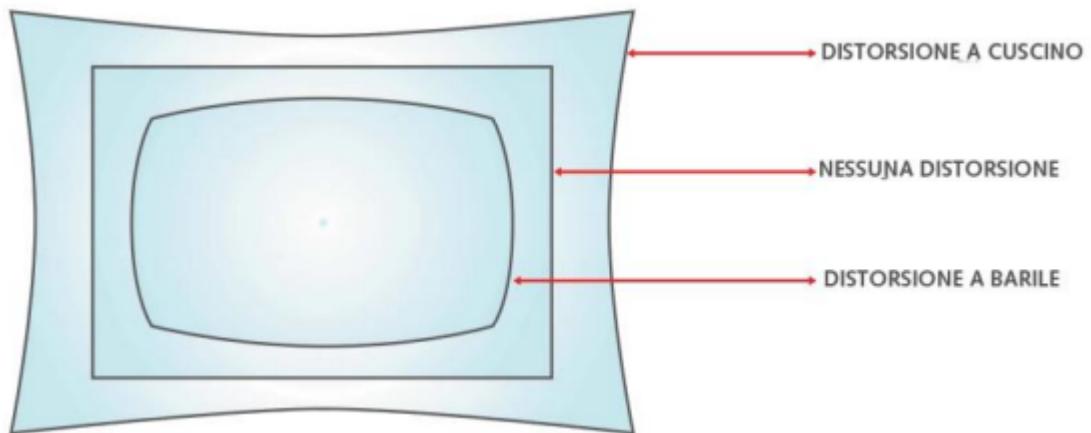


MTF (modulation transfer function)

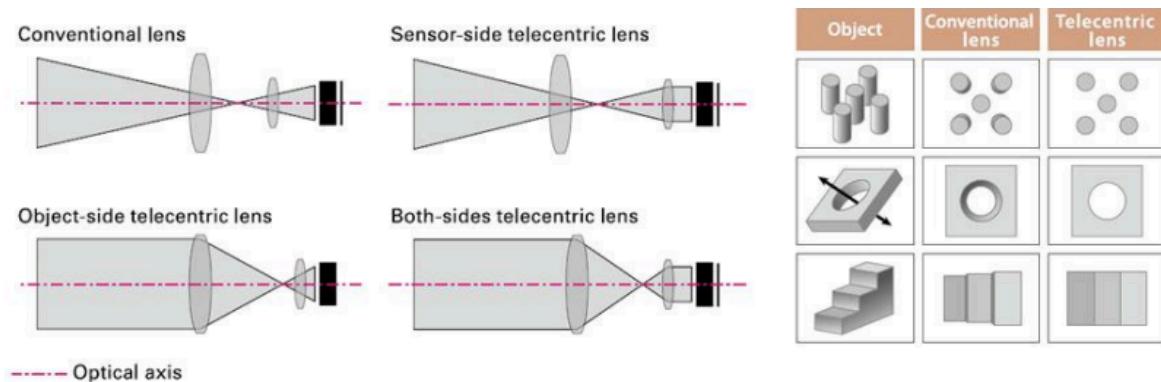
Misura che si utilizza per **valutare** le **prestazioni** di una lente fotografica, ovvero la capacità che la lente ha di **riprodurre dettagli fini** e contrasto nell'immagine. Generalmente è una curva e mostra il contrasto rispetto alla frequenza spaziale. Avrà **buoni valori per frequenze spaziali basse** e via via andranno a **peggiorare per valori crescenti** di frequenze spaziali. Dipende dai materiali utilizzati per la costruzione, dalle aberrazioni ottiche e dai rivestimenti antiriflesso.



Distorsione



Telecentricità



Protocolli di comunicazione

Firewire (IEEE1394)

Introdotto nel 1995 da Apple come standard per la connessione tra PC e periferiche.
Utilizzato come interfaccia standard per la visione industriale.
La banda dati video era di circa 400 Mb/s, ma l'evoluzione a Firewire B (IEEE1394b) nel 2002 ha aumentato la larghezza di banda massima a 800 Mb/s.

Universal Serial Bus (USB)

Introdotto da Intel, Microsoft, Compaq e IBM per consentire la comunicazione tra dispositivi e PC.
Le prime versioni, USB 1.0 (1,5 Mb/s) e USB 1.1 (12 Mb/s), non erano adatte per la visione industriale.
Nel 2001 è stato introdotto USB 2.0 (480 Mb/s), che è diventato uno standard comune nei PC.
Nel 2007 è stato introdotto USB 3.0 (5 Gb/s), con una bassa richiesta di CPU.
USB3 Vision, basato su USB 3.0, è uno standard introdotto nel 2015 con una banda dati di trasferimento immagini di circa 350 MB/s.

GenICam

GenICam (Generic Interface for Cameras) è un'interfaccia di programmazione generica per applicazioni di visione industriale.
Scopo dello standard è separare la tecnologia utilizzata per l'interfaccia fisica della fotocamera dall'interfaccia applicativa presentata all'utente.
GenICam è amministrato dall'European Machine Vision Association (EMVA).
È composto da diversi moduli, tra cui GenTL per l'interfaccia di programmazione dei transport level, GenApi per la descrizione dei parametri della fotocamera e SFNC per la convenzione di denominazione delle funzionalità standardizzate.

GigE Vision

Standard introdotto nel 2006 dall'AIA (Automated Imaging Association).
Basato sul protocollo User Datagram Protocol (UDP) delle schede di rete.
Supporta la lunghezza del cavo fino a 100 m in rame.
Utilizza GenICam con XML all'interno della fotocamera.
Offre una velocità di trasferimento dati di 100 MB/s con una porta singola e 200 MB/s con l'aggregazione di link.

Camera Link

Standardizzato nel 2001 dall'AIA.
Basato sulla comunicazione parallela a bassa tensione (LVDS).
Offre velocità di comunicazione elevate e una larghezza di banda dati fino a 850 MB/s.
Esistono diverse varianti di Camera Link, come Base, Medium, Full ed Extended, che differiscono per il numero di porte, chip e connettori.

Camera Link HS

Standard introdotto dall'AIA nel 2012 come evoluzione del protocollo Camera Link.
Destinato alle fotocamere ad alta velocità.
Offre una velocità di trasferimento dati di 300 MB/s per il canale in rame o 1200 MB/s per il canale in fibra ottica.
Supporta l'aggregazione di link fino a 8 canali.

CoaXPress

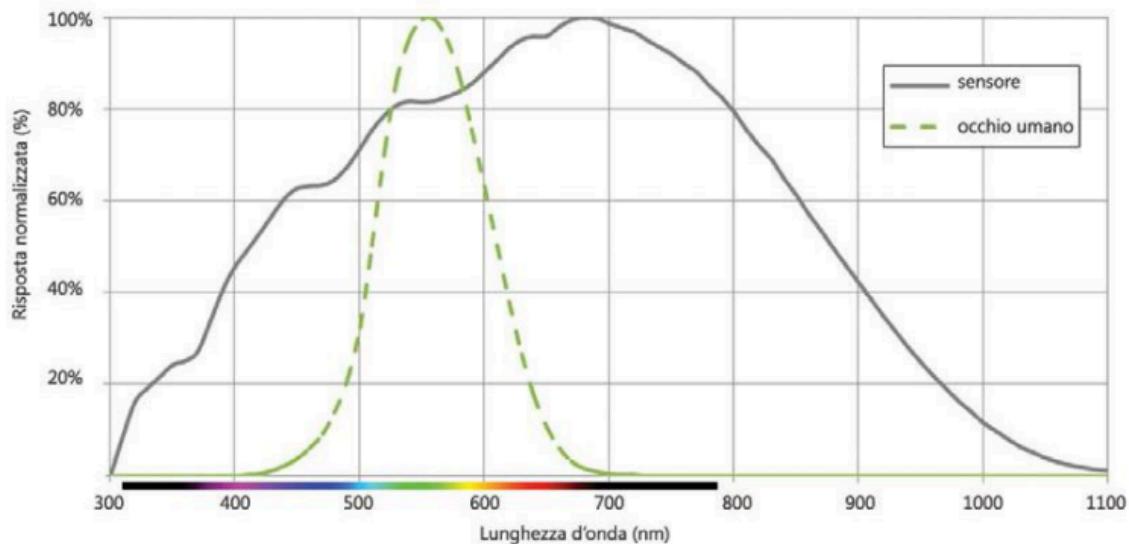
Standard introdotto dall'AIA nel 2010.
Utilizza un protocollo digitale serializzato su un singolo cavo coassiale.
Offre una velocità di trasferimento dati di 6,25 Gb/s per canale, supportando l'aggregazione di link fino a 6 canali.
Supporta l'alimentazione tramite il cavo coassiale.

Quando usare un frame grabber?

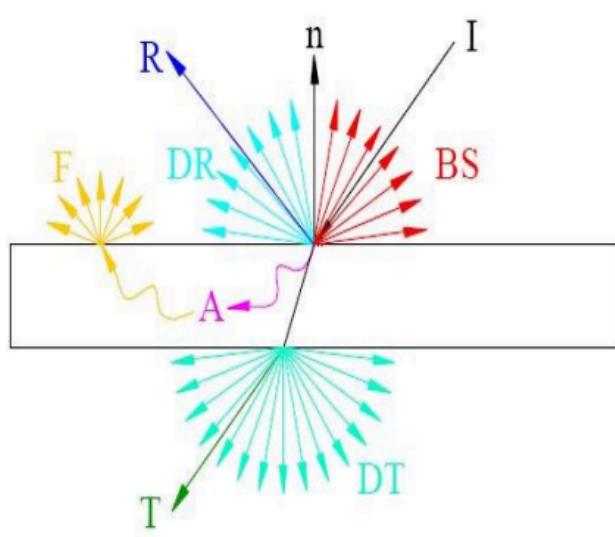
Un frame grabber è un dispositivo elettronico che viene generalmente collegato ad un computer, con il fine di acquisire una serie di immagini, streammando quello che vede il sensore, direttamente sul computer, come una scheda di acquisizione, permettendo quindi la manipolazione del video o di ogni singolo frame. E' necessario l'utilizzo di questi strumenti con alcuni protocolli, come Camera Link, ma in generale, ogni protocollo che utilizza porte che non sono presenti su un pc comune (tipo Ethernet), in quanto potremmo dover utilizzare camere con risoluzioni e framerate molto elevati, avendo quindi bisogno di una velocità di trasmissione elevata. Negli esercizi, è buono scrivere quando è necessario, avendo cura di specificare anche quante camere per ogni frame grabber (solitamente 2-4 camere per frame grabber)

Luce

La luce che noi riusciamo a vedere appartiene allo spettro della luce visibile. Le fotocamere, invece, hanno uno spettro più ampio, che ci consente di vedere spettri più ampi di quelli che riusciamo a vedere. Dunque la risposta di un sensore è molto più ampia su più lunghezze d'onda (solitamente vengono usati filtri per tagliare queste frequenze)

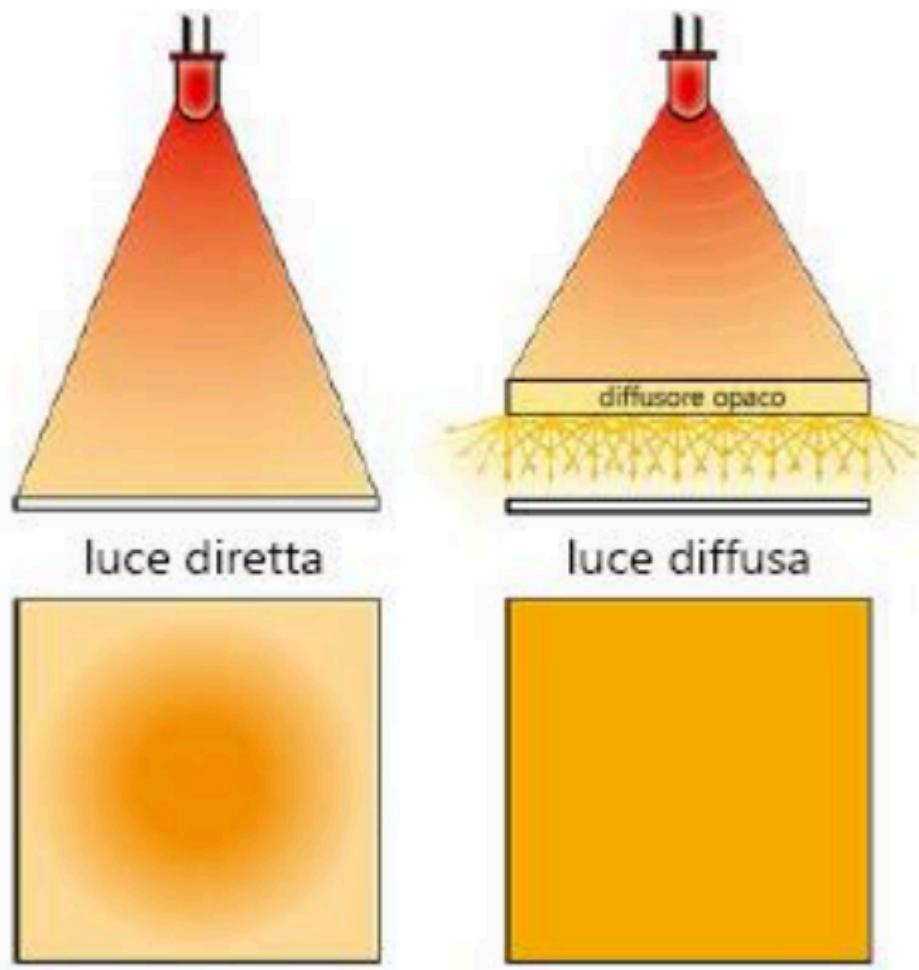


Tipi differenti di luce



Incident
reflected
Fluorescence
Diffuse reflected
Backscatter
Absorbed
Transmitted
Diffuse
Transmitted

Luce diretta e diffusa

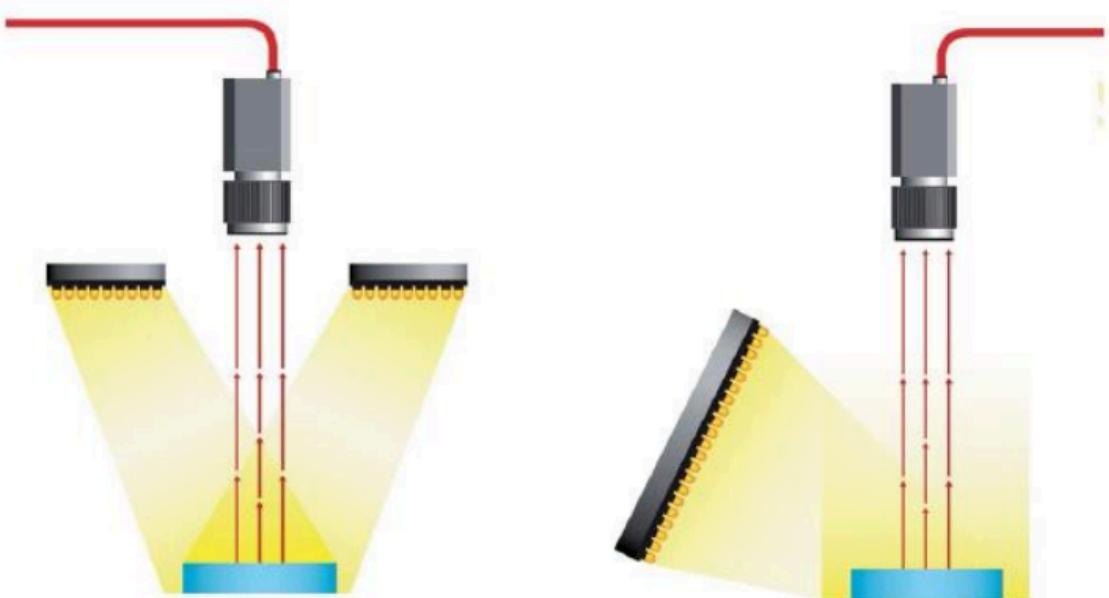


“Diffondere” la luce consente di illuminare omogeneamente tutto il FOV

Campo chiaro vs Campo scuro

Bright field (primo caso) → riflessione **all'interno** del FOV

Dark field (secondo caso) → riflessione **fuori** dal FOV

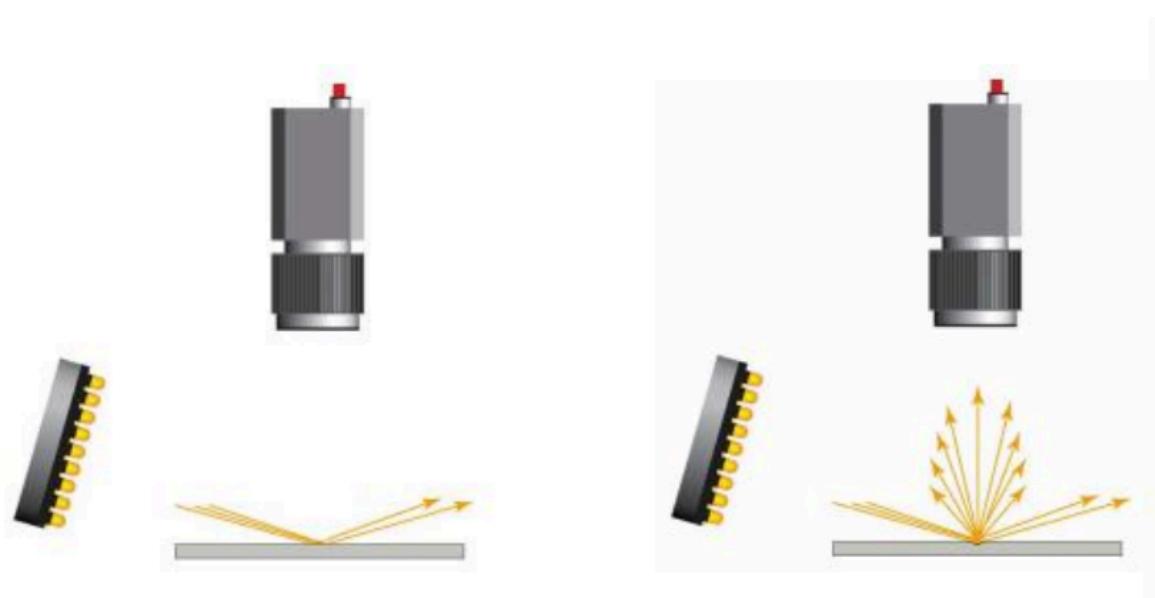


Il campo scuro serve per far risaltare particolarmente incisioni o bassorilievi

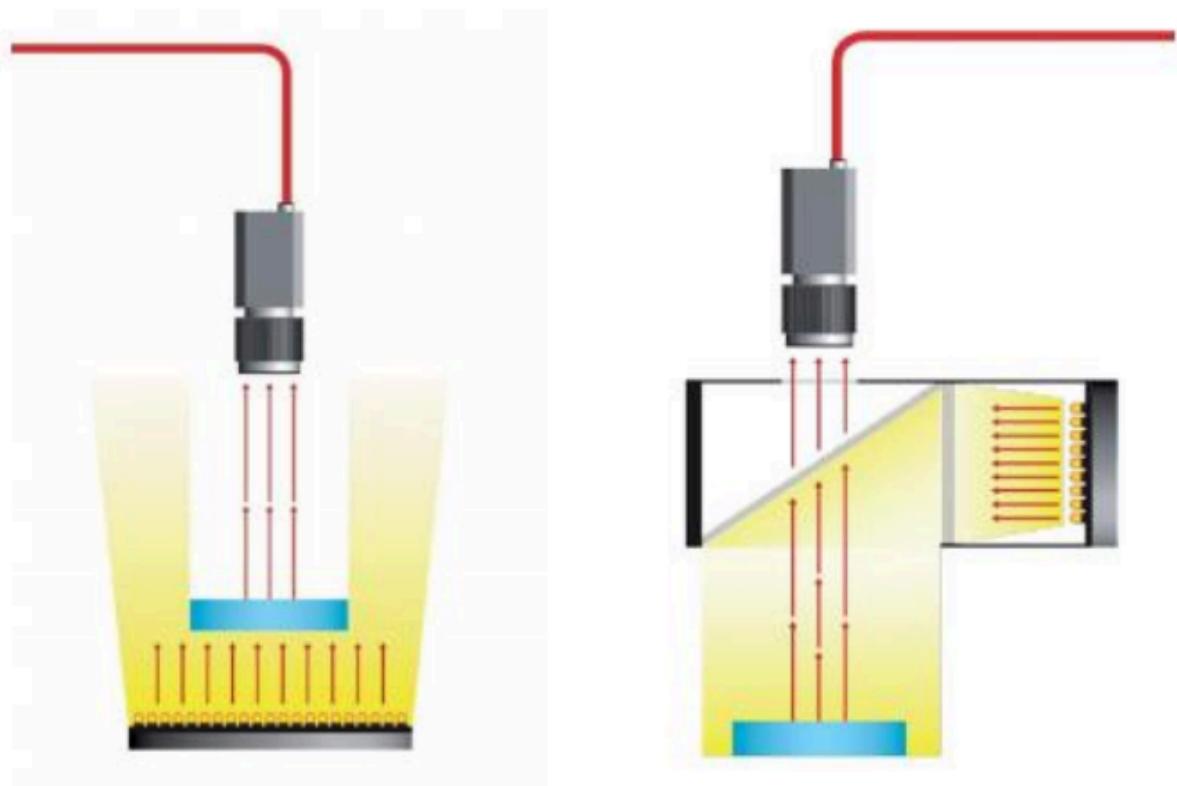


Gazing lighting

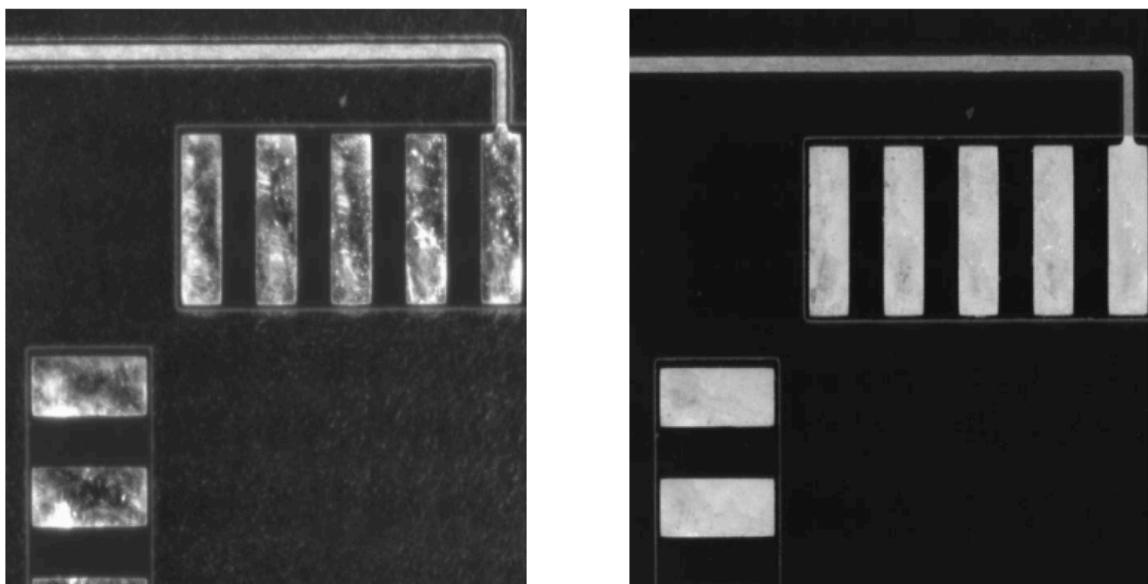
Particolarmente utile per evidenziare eventuali graffi e ruvidità degli oggetti



Back light e luce coassiale (DOAL)

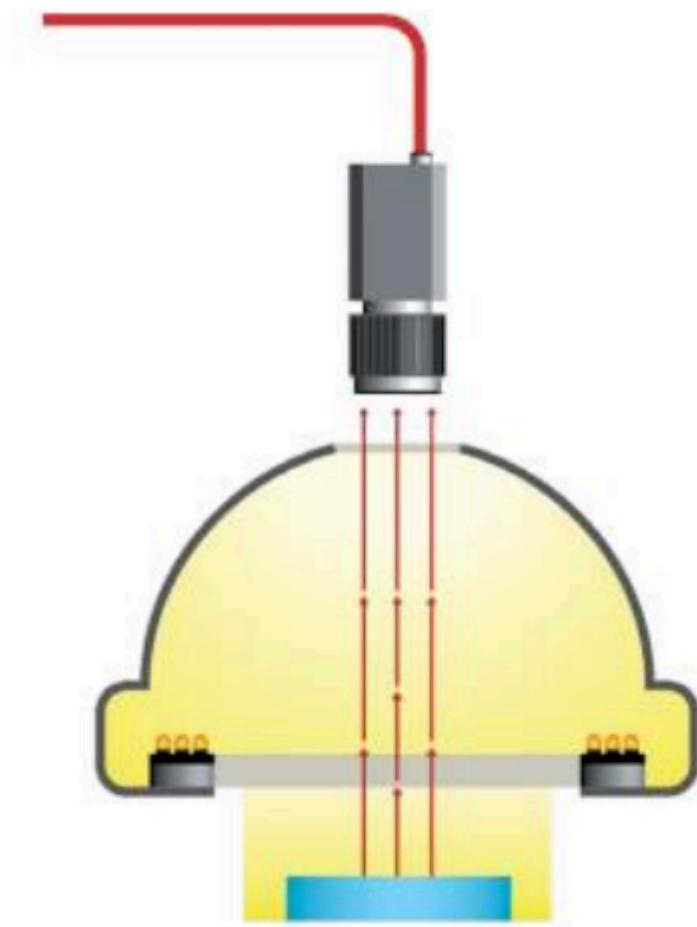


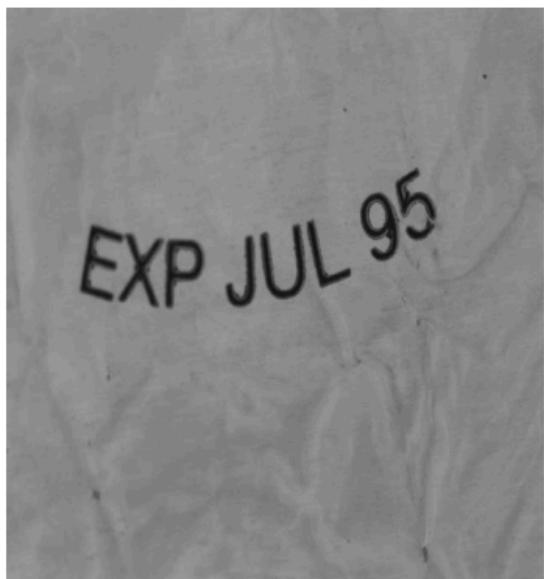
Il DOAL è particolarmente utilizzato per evitare riflessioni indesiderate



Cupola (CDI)

Toglie tutte le ombre e illumina tutto uniformemente, permettendo di leggere eventuali scritte su superfici con increspature





Riassunto dei tipi di illuminazione



direct



diffuse



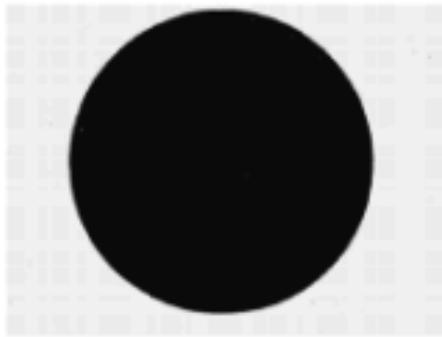
Bright field



CDI



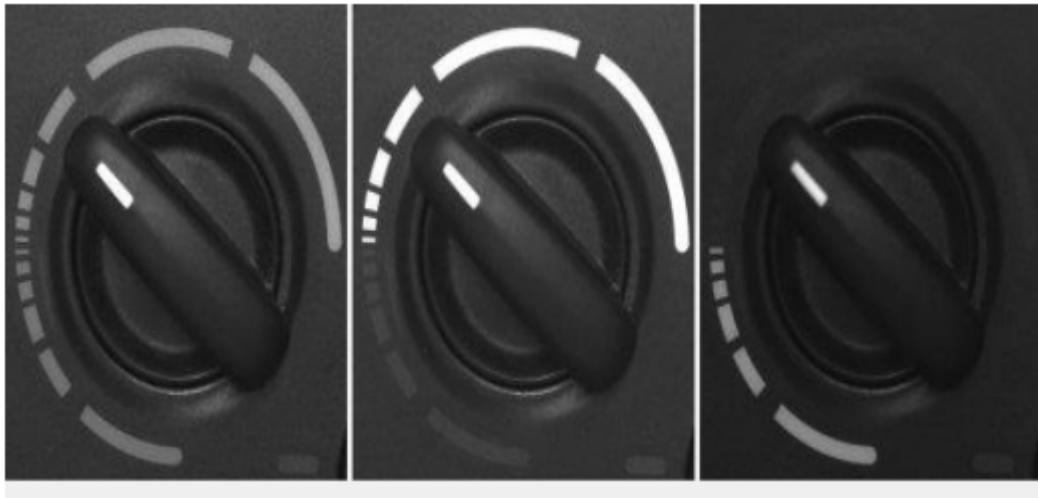
Dark field



Back

Filtri per lunghezze d'onda

Per discernere particolari dettagli di cui abbiamo bisogno dobbiamo utilizzare dei **filtri**, che ci servono per escludere tutto quello che non ci serve. In questo caso la prima immagine è senza filtro, la seconda ha un filtro rosso (635nm) e la terza un filtro blu (470nm)



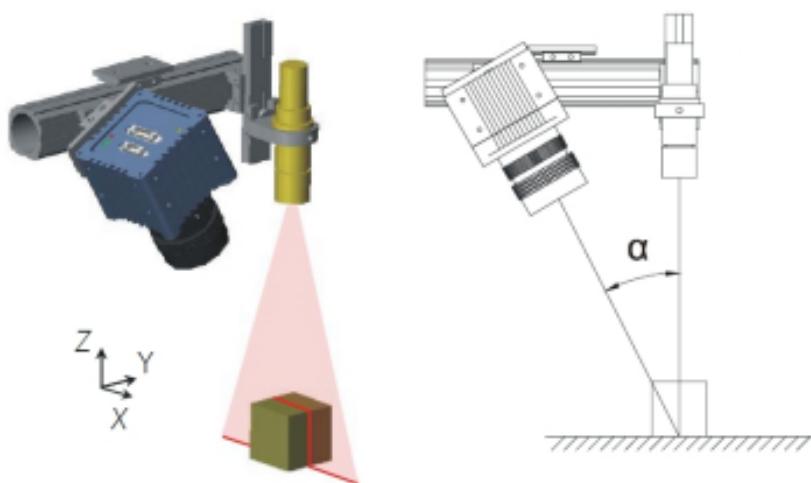
Dunque, nella seconda immagine il filtro si occuperà di far passare soltanto il contributo rosso, mentre nella terza soltanto quello blu, evidenziando le componenti selezionate nelle rispettive foto

Profilometro

Un **profilometro** è uno **strumento** che si usa generalmente per misurare la **topografia** e la **rugosità** di una **superficie** (tridimensionalmente) a partire da una rilevazione bidimensionale. Si possono utilizzare diversi tipi di geometrie, di cui ognuna ha delle caratteristiche ben precise, ovvero:

- 1) una **fotocamera**
- 2) un **laser**
- 3) un **nastro trasportatore** (o un binario qualora sia la fotocamera a muoversi)

Geometria 1



$$\Delta Z = \frac{\Delta X}{\sin \alpha}$$

La risoluzione, per i 3 assi dipende:

- X: dalla **risoluzione** della **fotocamera**
- Y: dal **movimento** relativo tra la fotocamera e il sistema (ovvero dalla frequenza di acquisizione)
- Z: dalla formula scritta sopra

Qualora volessimo aumentare la risoluzione dovremmo agire rispettivamente:

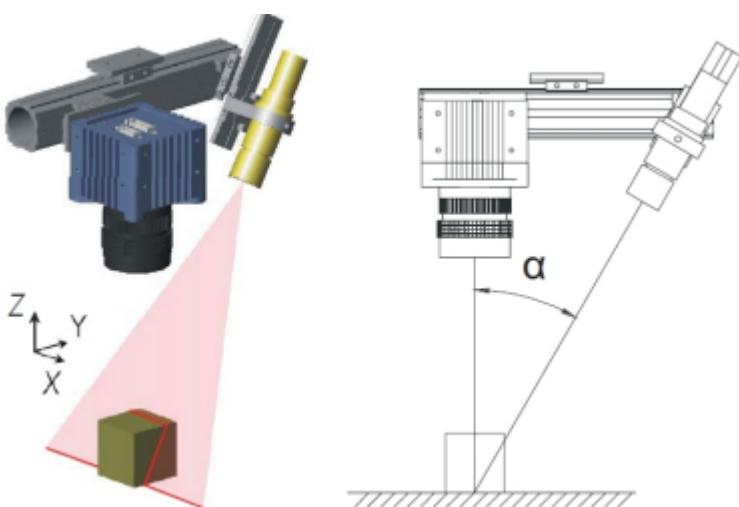
- X: **aumentare la risoluzione della fotocamera**
- Y: **rallentare il movimento o aumentare la frequenza di acquisizione**

Per quanto riguarda l'asse Z, invece c'è da fare un appunto importante, ovvero è necessario arrivare a un compromesso per scegliere l'angolo:

- Se **l'angolo è alto** sarà **migliorata la ricostruzione lungo l'asse Z**, ma **peggiorata lungo l'asse Y** (per la difficoltà intrinseca)
- Se **l'angolo è basso** sarà **migliorata la ricostruzione lungo l'asse Y**, ma **peggiorata lungo l'asse Z**
- Nel caso in cui **l'angolo** valesse **90** la luce non arriverebbe al profilometro, rendendo inutile l'acquisizione
- Nel caso in cui **l'angolo** (per assurdo) valesse **0** non sarebbe visibile alcuna differenza in base all'altezza della misurazione

Ovviamente **non** devono essere presenti **ostacoli** tra la fotocamera e l'oggetto

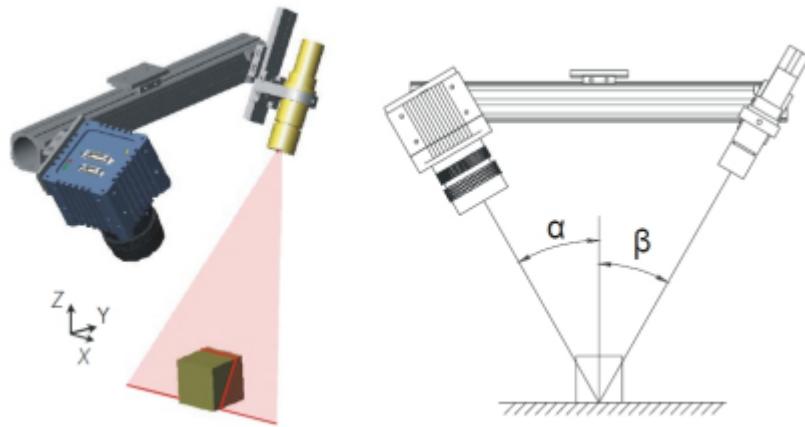
Geometria 2



$$\Delta Z = \frac{\Delta X}{\tan \alpha}$$

Le valutazioni sulla risoluzione, sia per l'asse X che per l'asse Y rimangono identiche. Discorso diverso per l'asse Z, dove si hanno degli evidenti **miglioramenti**: la **risoluzione** sarà **migliorata** a parità di angolo, questo perché al denominatore ci sarà un valore mediamente maggiore rispetto a quello del caso precedente. Tuttavia è da considerare che potrebbe esserci un'acquisizione peggiore in quanto, a causa della configurazione, la faccia frontale potrebbe non essere correttamente valutata, in quanto, nonostante lo scattering non sarebbe riflessa sufficiente luce alla fotocamera

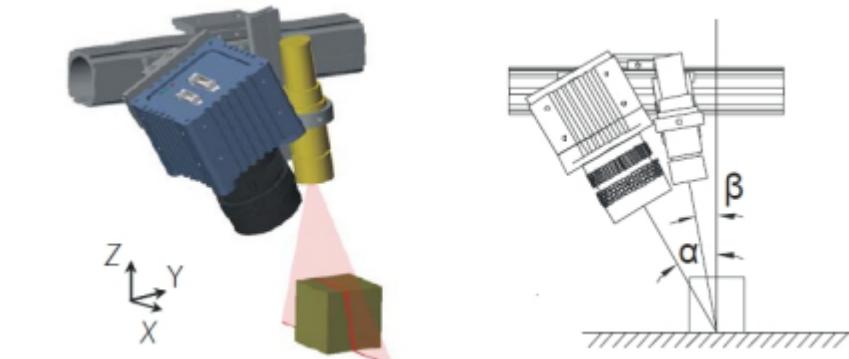
Geometria 3



$$\Delta Z \approx \Delta X \frac{\cos \beta}{\sin(\alpha+\beta)} \approx \frac{\Delta X}{2 \sin(\alpha)}$$

In questo caso bisogna prestare particolari attenzioni a eventuali **occlusioni**, ma il **risultato è migliore** rispetto al primo caso, e la risoluzione arriva a essere circa il **doppio**. In generale si preferisce adottare lo stesso angolo

Geometria 4



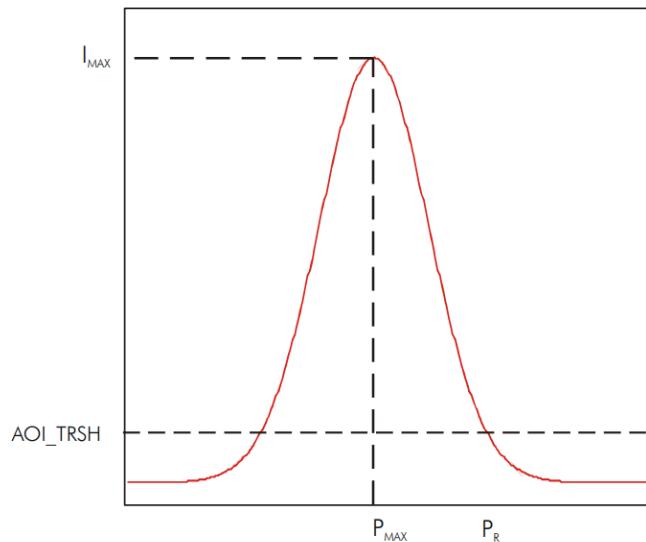
$$\Delta Z \approx \Delta X \frac{\cos \beta}{\cos(\alpha-\beta)}$$

In questo caso c'è **minore** probabilità di **occlusione**, in quanto ci sono piccolissime differenze tra i due angoli, pertanto l'oggetto sarà profilato bene.

Modalità di massima intensità (MAX)

The Maximum Intensity Profile Mode (MAX)

In this mode the position of the maximum intensity of laser beam profile is calculated. The result includes the position value of the maximum (P_{MAX}) as well as the maximum intensity value (I_{MAX}).

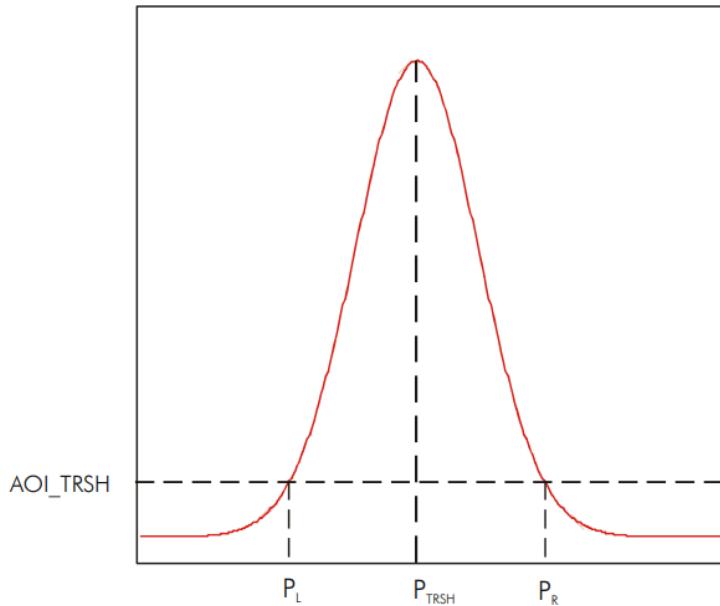


The calculation of position value is performed with simple pixel accuracy, i.e. the evaluation of 1024 rows delivers a position range from 0 to 1023 pixels (10 bit). If there is more than one local maximum, the position of the first maximum (starting from row zero) is used.

Modalità a soglia (THRESHOLD)

The Threshold Mode (TRSH)

In this mode the left (P_L) and (P_R) right edge position of the laser beam profile are calculated for a given threshold value of intensity [AOI_TRSH](#).



The position value of the laser line is approximated: $P_{TRSH} = (P_L + P_R) / 2$. In order to simplify the digital representation the division over 2 is not performed and thus an integer representation with one subpixel is realised. The evaluation of 1024 rows delivers a position range from 0 to 2047 pixels (11 bit).

In threshold mode the camera can output either the left and right threshold position separately or the subpixel position ($P_L + P_R$) and the line width ($P_R - P_L$). Moreover, the maximum intensity value can be optionally delivered.

Modalità a centro di gravità (COG)

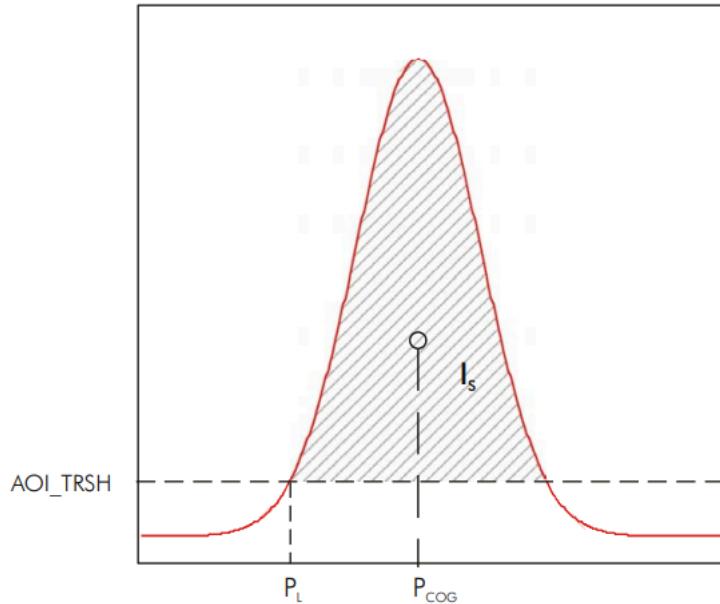
The Center Of Gravity Mode (COG)

In this mode the center of gravity of laser beam profile is calculated. For this purpose the following parameters are computed:

Position value of the left edge of laser beam profile for a given intensity threshold value P_L ,

Sum of intensity value $I_s = \sum I_p$,

Sum of first order moment $M_s = \sum I_p * P$.



The position value of laser line (center of gravity of beam profile) is then obtained from:

$$P_{COG} = P_L + M_s / I_s .$$

In addition the laser line width can be delivered over the [Data Channel DC1](#). The average intensity of the illumination profile can be calculated by normalising the sum of intensity value I_s with the line width.

In questo caso, il risultato (P_{COG}), sarà un numero con molte cifre decimali, ma questo non vuol dire assolutamente che questo metodo ci restituisce un risultato più preciso. Piuttosto, questo risultato varierà nelle varie acquisizioni, mantenendo alcune cifre costanti, che costituiranno la precisione “effettiva”, sotto la quale c’è un tappeto di rumore.

Trasformazioni

Cenni generali

Ogni immagine bidimensionale, come suggerisce il nome, è composta da due coordinate, X e Y, pertanto definiamo il vettore \mathbf{x} (è in grassetto, indica un vettore)

- \mathbf{x} is a point in our image where:
 - $\mathbf{x} = (x, y)$ or in matrix terms

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

\mathbf{x} (IN **BOLD**) VECTOR, x (normal) scalar coordinate, like y

Qualora noi volessimo operare su questo vettore, potremmo moltiplicare per una matrice, ottenendo così un nuovo vettore, ad esempio $\mathbf{x}' = \mathbf{M}\mathbf{x}$. Con la stessa logica possiamo anche scalare (ingrandire) un'immagine

Ingrandimento

Scaling is just a matrix operation

- Map points from one image into another
- Often we can use matrix operations
- Given a point \mathbf{x} , map to new point \mathbf{x}' using \mathbf{M}

$$\mathbf{x}' = \mathbf{S} \mathbf{x}$$

$$\mathbf{x}' = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \mathbf{x}$$

(Note: In case of S_{11} different from S_{22} , deformation along one dimension)

In questo caso, il risultato sarà:

- $x' = S \cdot x + 0 \cdot y = S \cdot x$
- $y' = 0 \cdot x + S \cdot y = S \cdot y$

Questo vuol dire che saranno scalate le componenti proporzionalmente al fattore S scelto.

Traslazione

Nel caso della traslazione abbiamo bisogno di compiere un'operazione particolare al vettore, ovvero "aumentarlo", aggiungendo una terza coordinata, che sarà uguale ad 1

- \bar{x} is x but with an added 1
- *Augmented vector*

$$\bar{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Questo procedimento sarà valido anche per tutte le altre trasformazioni.

- \bar{x} is x but with an added 1
- *Augmented vector*
- Now translation is easy
- (I= Identity; t= translation)

$$\bar{\mathbf{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{x}' = [I \ t] \bar{\mathbf{x}}$$

I è la matrice identità (solo la diagonale principale ha valore 1, tutto il resto 0) e t è un vettore di traslazione. Il risultato, \mathbf{x}' sarà uguale a

- $\bar{\mathbf{x}}$ is \mathbf{x} but with an added 1
- *Augmented vector*
- Now translation is easy

$$\bar{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix} \quad \mathbf{x}' = [\mathbf{I} \ \mathbf{t}] \bar{\mathbf{x}}$$

dove dx e dy rappresentano i gradi di libertà, ovvero i parametri sui quali possiamo giocare.
In questo caso, il risultato sarà:

- $x' = 1 \cdot x + 0 \cdot y + dx \cdot 1 = x + dx$
- $y' = 0 \cdot x + 1 \cdot y + dy \cdot 1 = y + dy$

Per cui l'immagine rimarrà invariata a livello di dimensioni, ma subirà uno spostamento lungo gli assi.

Euclidea

Se volessimo anche **ruotare** l'immagine, oltre a spostarla, si parla di trasformazione euclidea. Qui servono le stesse premesse della traslazione, ovvero si utilizza il vettore aumentato

- Want to translate and rotate at same time
- Still just matrix operation
- \mathbf{R} is rotation matrix, \mathbf{t} is translation

$$\mathbf{x}' = \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix} \quad \mathbf{x}' = [\mathbf{R} \ \mathbf{t}] \bar{\mathbf{x}}$$

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

In questo caso \mathbf{R} è una matrice di rotazione, \mathbf{t} quella di traslazione (già usata prima) e il risultato che otteniamo sarà:

- $x' = \cos\theta \cdot x - \sin\theta \cdot y + dx$
- $y' = \sin\theta \cdot x + \cos\theta \cdot y + dy$

Similarità

In questo caso si tratta di fare una trasformazione euclidea ma scalando anche l'immagine, dunque avremo bisogno di un vettore che moltiplichi la matrice di rotazione

$$\mathbf{x}' = [s\mathbf{R} \quad \mathbf{t}] \bar{\mathbf{x}}$$

$$\mathbf{x}' = \begin{bmatrix} a & -b & dx \\ b & a & dy \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

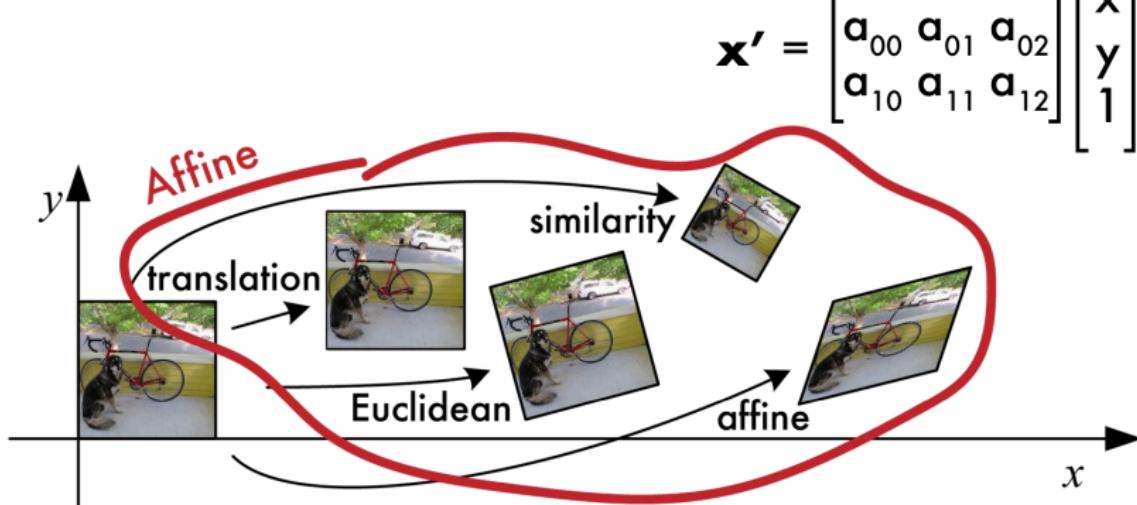
Pertanto i valori di a e b dipenderanno dai coefficienti di ingrandimento (quelli chiamati S, nel caso precedente) e dal valore del seno e del coseno di theta.

Affine

Questo tipo di trasformazione può anche prevedere un insieme generico di tutti gli altri tipi, pertanto in generale prevede la scelta di tutti e 6 i coefficienti e la moltiplicazione per una matrice \mathbf{M} , che rappresenta tutte le operazioni che vogliamo svolgere

General case of 2x3 matrix

$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Chiaramente potrebbero esserci dei problemi, che nascono dal fatto che le matrici finora utilizzate fossero 2×3 , questo problema ha una soluzione:

Say you want to translate, then rotate, then translate back, then scale.

$$\mathbf{x}' = \mathbf{S} \ \mathbf{t} \ \mathbf{R} \ \mathbf{t} \ \mathbf{x} = \mathbf{M} \ \mathbf{x},$$

$$\text{If } \mathbf{M} = (\mathbf{S} \ \mathbf{t} \ \mathbf{R} \ \mathbf{t})$$

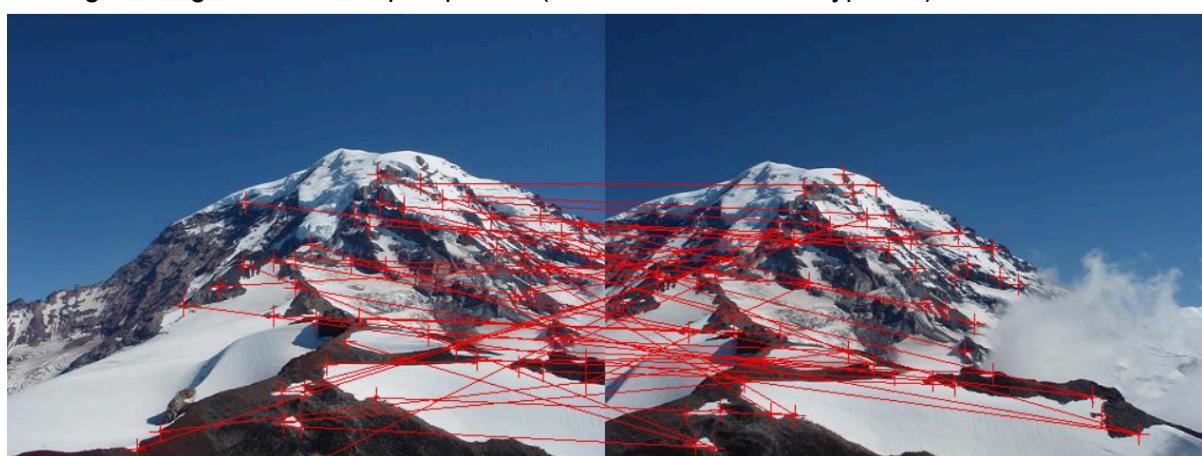
\mathbf{M} is still affine transformation

Wait, but these are all 2×3 , how to we multiply them together?

$$\begin{aligned}\bar{\mathbf{x}}' &= \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} & \bar{\mathbf{x}}' &= \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \bar{\mathbf{x}}' &= \begin{bmatrix} a & -b & dx \\ b & a & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} & \bar{\mathbf{x}}' &= \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\end{aligned}$$

Ovvero aggiungere un'altra riga alla fine, con quelle componenti.

Se per esempio volessi trovare la matrice M di coefficienti a che mi mappa i punti di una montagna fotografata da due prospettive (conoscendo alcuni keypoints)



Affine transformation → 6 Degrees of Freedom

$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Poniamo i punti immagine $x' = n$ e i punti del dominio $x=m$ e la matrice dei coefficienti A quindi: $m^*A=n$, possiamo scrivere 2 equazioni per le 2 coordinate di ciascun punto n:

$$n_x = a_{00}*m_x + a_{01}*m_y + a_{02}*1$$

$$n_y = a_{10}*m_x + a_{11}*m_y + a_{12}*1$$

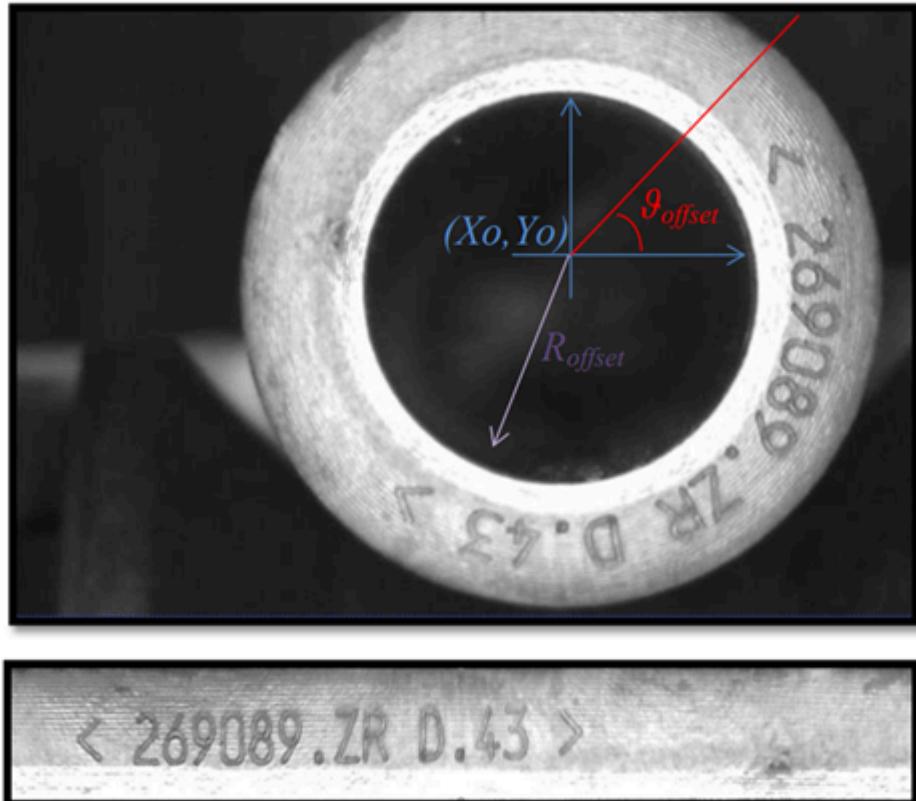
Considerando più punti (n_x_1, n_y_1), (n_x_2, n_y_2), ... posso arrivare a determinare i coefficienti a:

- Solve $\mathbf{M} \mathbf{a} = \mathbf{b}$
- $\mathbf{M}^T \mathbf{M} \mathbf{a} = \mathbf{M}^T \mathbf{b}$
- $\mathbf{a} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{b}$
- Still works if overdetermined

$$\begin{array}{c} \mathbf{M} \quad \mathbf{a} \quad \mathbf{b} \\ \left[\begin{array}{ccccc} m_{x1} & m_{y1} & 1 & 0 & 0 \\ 0 & 0 & 0 & m_{x1} & m_{y1} \\ m_{x2} & m_{y2} & 1 & 0 & 0 \\ 0 & 0 & 0 & m_{x2} & m_{y2} \\ m_{x3} & m_{y3} & 1 & 0 & 0 \\ 0 & 0 & 0 & m_{x3} & m_{y3} \end{array} \right] \left[\begin{array}{c} a_{00} \\ a_{01} \\ a_{02} \\ a_{10} \\ a_{11} \\ a_{12} \end{array} \right] = \left[\begin{array}{c} n_{x1} \\ n_{y1} \\ n_{x2} \\ n_{y2} \\ n_{x3} \\ n_{y3} \end{array} \right] \\ \dots \quad \dots \end{array}$$

Trasformazione in coordinate polari

$$\theta = \arctg((x-X_o)/(y-Y_o))$$
$$r = \sqrt{(x-X_o)^2 + (y-Y_o)^2}$$



Volendo shiftare poi

$$x' = r (\vartheta_{\text{offset}} - \vartheta)$$

$$y' = r - R_{\text{offset}}$$

Proiezione

Per effettuare questo tipo di trasformazioni abbiamo bisogno di utilizzare un vettore con 3 coordinate, in questo caso un vettore aumentato con una modifica, ricordando che una foto è una proiezione bidimensionale in 3 dimensioni

- Homogeneous coordinate system
 - Useful because we can do this kind of transform
- Each point in 2d is actually a vector in 3d
- Equivalent up to scaling factor
- Have to normalize to get back to 2d

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix} \quad \bar{\mathbf{x}} = \tilde{\mathbf{x}} / \tilde{w}$$

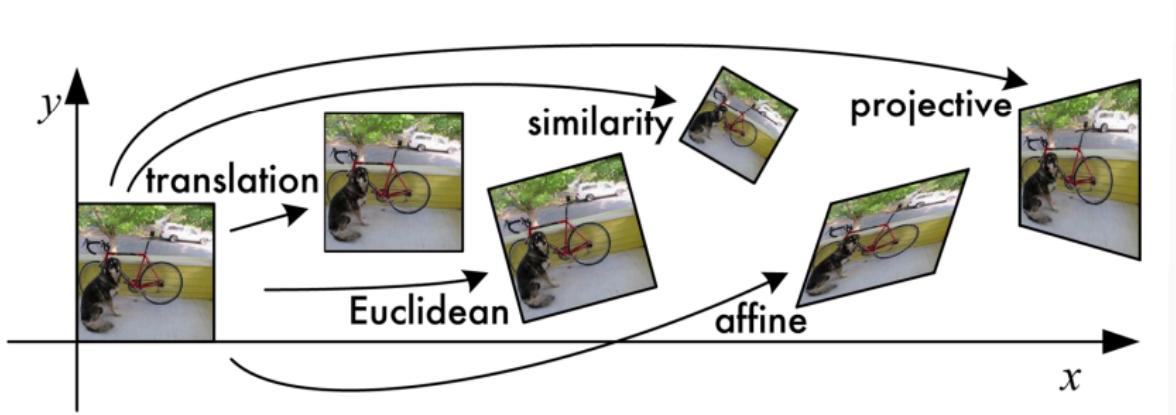
- Multiply $\tilde{\mathbf{x}}$ by \mathbf{H} to get $\tilde{\mathbf{x}'}$
- Convert to $\bar{\mathbf{x}}$ by dividing by w'

$$\tilde{\mathbf{x}'} = \mathbf{H} \tilde{\mathbf{x}}$$

$$\begin{bmatrix} \tilde{x}' \\ \tilde{y}' \\ \tilde{w}' \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix} \quad \bar{\mathbf{x}} = \tilde{\mathbf{x}} / \tilde{w}$$

Dunque abbiamo bisogno di effettuare questo procedimento in quanto la matrice \mathbf{H} è una 3x3 e per ottenere il vettore risultante devo prima effettuare la moltiplicazione con la matrice \mathbf{H} e successivamente dividere tutto per lo scalare w . Così facendo otterò un risultato con 2 coordinate. I gradi di libertà sono 8 e non 9, come si potrebbe immaginare erroneamente, in quanto è possibile usare una scala arbitraria, pertanto soltanto 8 saranno indipendenti

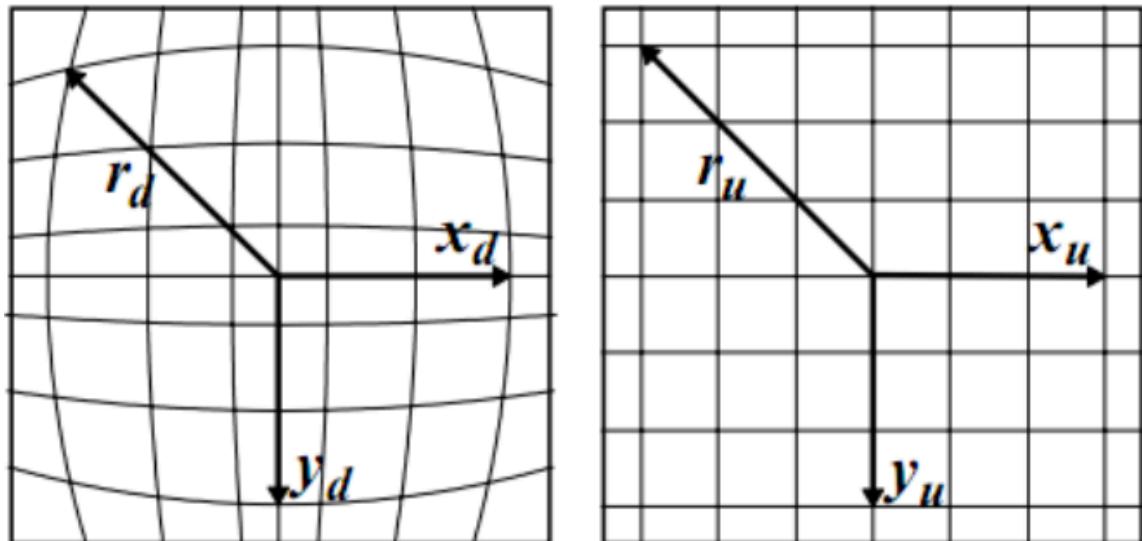
In breve



Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3 \times 3}$	8	straight lines	

Barrel Distortion Removal

Nel caso della distorsione a barile, è possibile correggerla utilizzando una trasformazione delle coordinate (polari) di ciascun punto, secondo la formula



$$\vartheta_u = \vartheta_d$$

$$r_u = r_d + k r_d^3$$

Dove k è una caratteristica della lente.

Una volta effettuata la trasformazione, però, ci sarà il problema di trasferire i dati di intensità luminosa: La matrice (raster) della output image non corrisponderà con la struttura dell'immagine corretta, quindi sarà necessario effettuare:

Interpolazione post correzione della distorsione

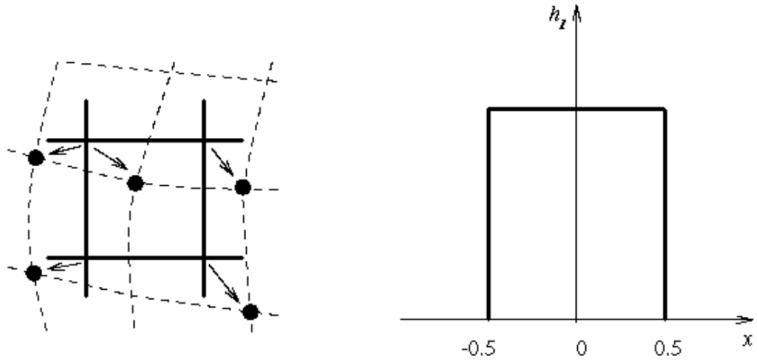
Each pixel value in the output image raster (as it was when the image was distorted) can be obtained by brightness interpolation of some neighboring noninteger samples. Per calcolare il valore di brightness:

$$f_n(x, y) = \sum_{l=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} g_s(l\Delta x, k\Delta y) h_n(x - l\Delta x, y - k\Delta y)$$

Dove f è il risultato dell'interpolazione e h è l'**interpolation Kernel**

- **Nearest Neighbor interpolation**

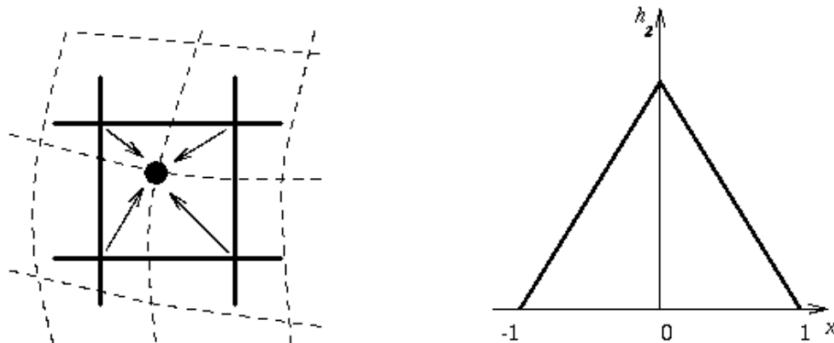
$$f_1(x, y) = g_s(\text{round}(x), \text{round}(y))$$



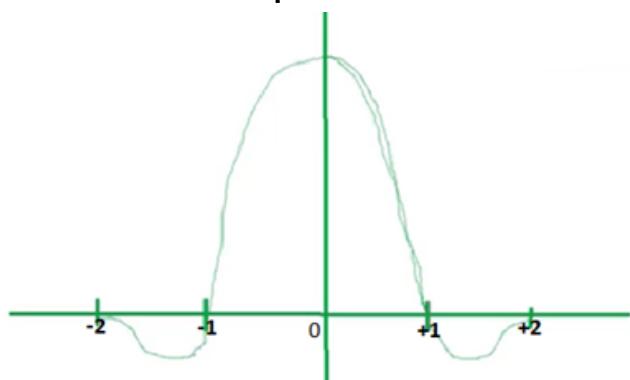
Linee spezzate = raster di output, Linee continue = nuove coordinate non distorte.

Errore di posizione massimo $\frac{1}{2}$ pixel

- **Linear interpolation**



- **Bicubic Interpolation - Mexican Hat**



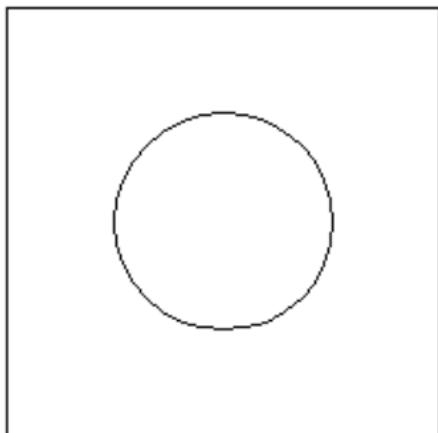
- improves the model of the brightness function by approximating it locally by a bicubic polynomial surface; sixteen neighboring points are used for interpolation.
- interpolation kernel ('Mexican hat') is given by

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$

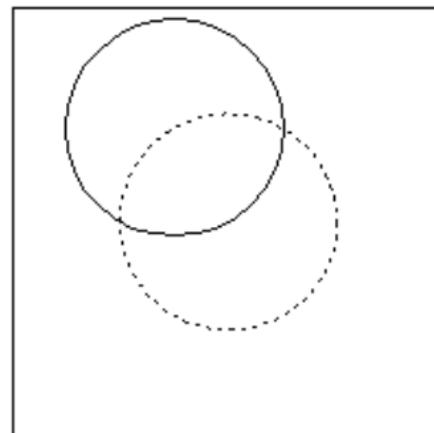
Trasformata di Hough

La Trasformata di Hough è una tecnica utilizzata per studiare i contorni di un'immagine e individuare se i pixel appartengono ad una forma specifica, come una retta o, nel caso della General Hough Transform (GHT), qualsiasi altra forma.

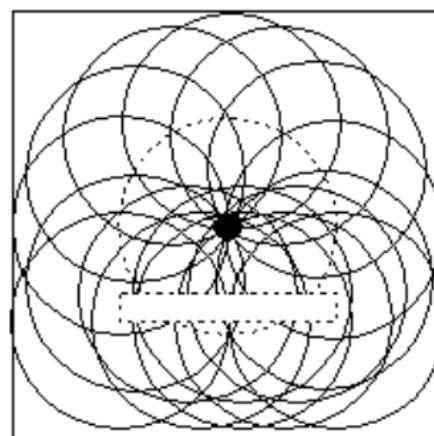
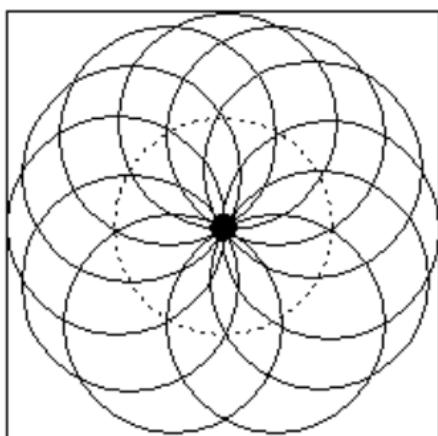
Nel caso del rilevamento dei cerchi con un raggio r noto, disegneremo cerchi di raggio r per ogni pixel “scuro”. Se troviamo un punto comune ad un grande numero di cerchi, quello sarà il centro del cerchio reale. Questo metodo funziona anche se l'immagine contiene altri elementi.



a)

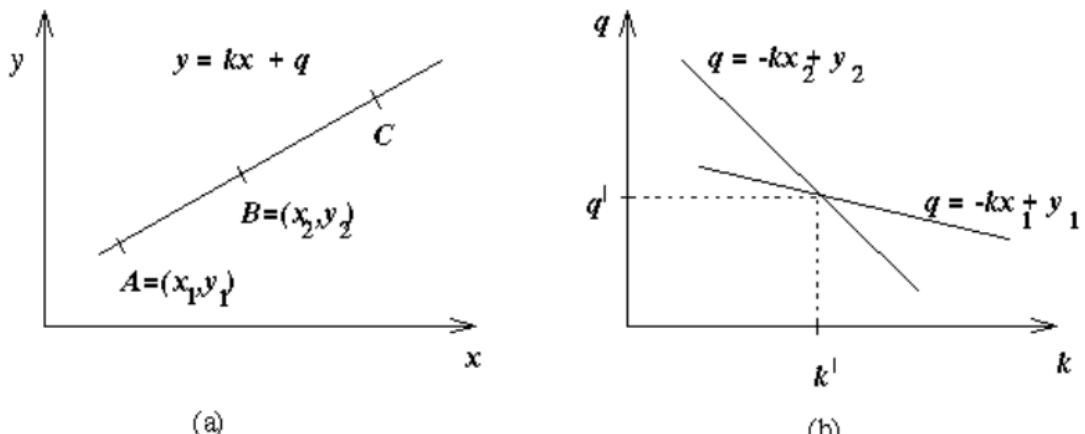


b)



L'obiettivo principale è identificare una forma specifica, come una retta. Per fare ciò, creiamo uno spazio dei parametri (o “accumulator matrix”) in cui possiamo rappresentare tutte le possibili combinazioni di parametri che possono realizzare la nostra forma. Ad esempio, nel caso di una retta, questi parametri sarebbero il coefficiente angolare k e l'intersezione q .

Per ciascun punto potenzialmente “scuro”, consideriamo il fascio di rette che passa per quel punto, quindi un insieme di possibili parametri (k,q) che segniamo nel nostro spazio dei parametri. Poi consideriamo un altro punto dell'immagine. Alla fine, troveremo un punto comune a tutte le rette, che corrisponde ai parametri della retta reale.



L'equazione parametrica della retta $y=kx+q$ è appropriata solo per spiegare i principi della Trasformata di Hough - causa difficoltà nella rilevazione delle linee verticali ($k \rightarrow \text{infinito}$) e nella discretizzazione non lineare del parametro k . Quindi, **di solito si usano le coordinate polari**.

Se stiamo cercando cerchi, l'espressione analitica $f(x,a)$ della curva desiderata è

$$f(x, a) = (x - a)^2 + (y - b)^2 - r^2 = 0$$

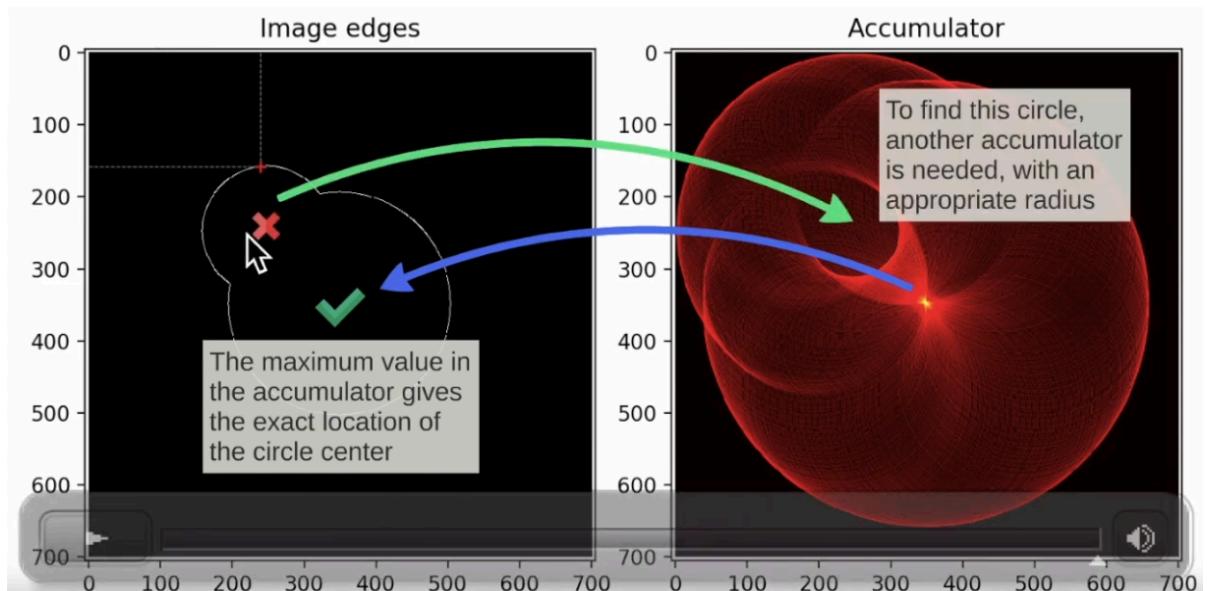
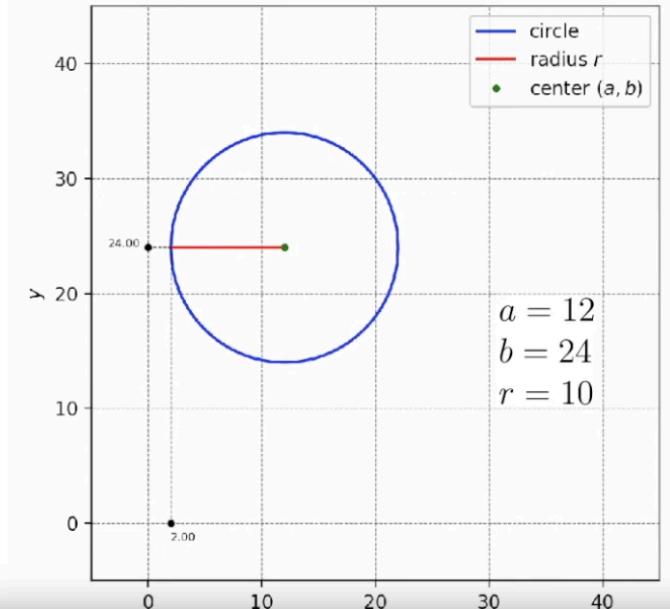
dove (a,b) è il centro del cerchio e r è il raggio. Quindi, lo spazio dei parametri (e la matrice accumulatore) sarà tridimensionale.

In questo approccio, è fondamentale modellare adeguatamente la forma e discretizzare lo spazio dei parametri. Mentre è più semplice individuare il massimo locale, a volte ce ne possono essere più di uno e quindi sarà necessario effettuare un'aggregazione (o sottocampionare lo spazio dei parametri).

Drawing a circle

$$r^2 = (x - a)^2 + (y - b)^2$$

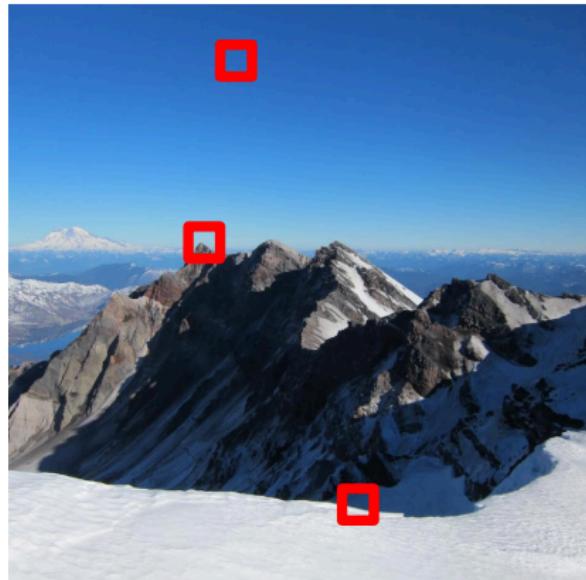
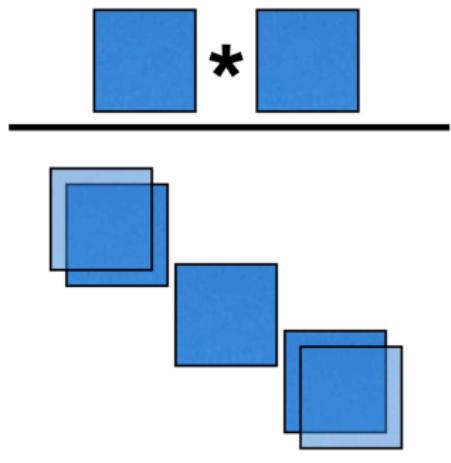
radius circle center



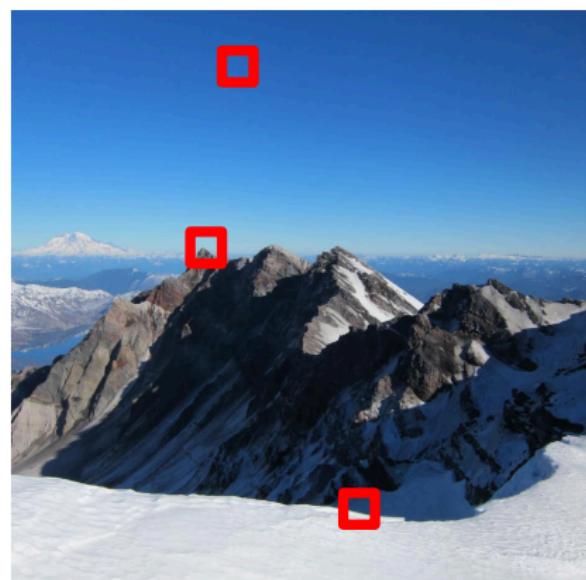
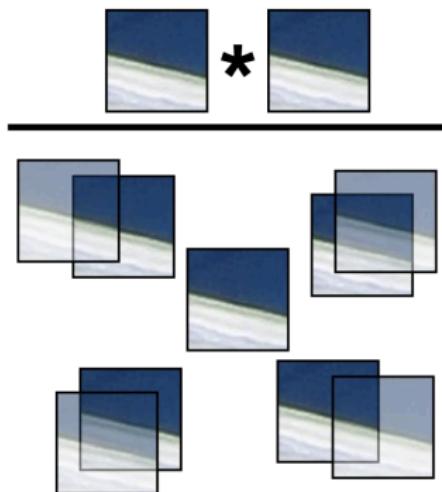
Features

Con il termine **feature** si fa riferimento ad un **processo di estrazione** delle caratteristiche **rilevanti** da un'immagine. Queste sono alla base di diversi algoritmi, come ad esempio quelli di detection e recognition. In questo caso, l'immagine che si vuole analizzare si divide in tante piccole parti e per ognuna si valuta il gradiente, ovvero una misura della variazione nella sottoimmagine che abbiamo ottenuto. Dunque, questo ci permette di rilevare punti facilmente riconoscibili:

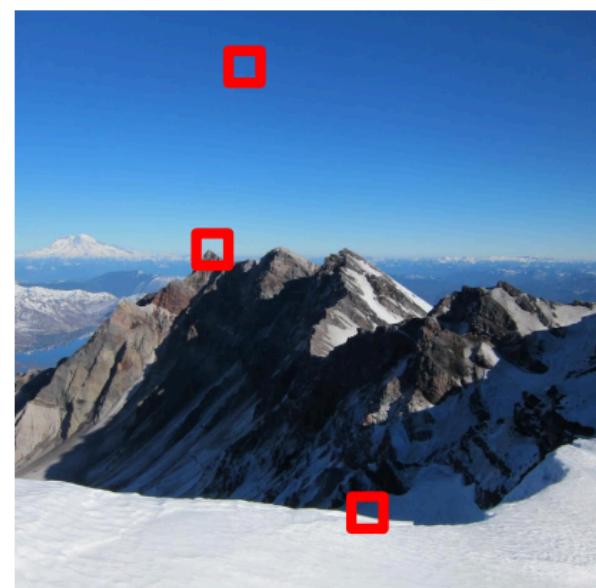
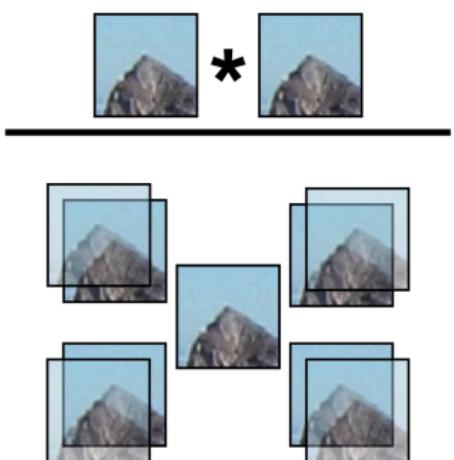
Sky: low everywhere



Edge: low along edge



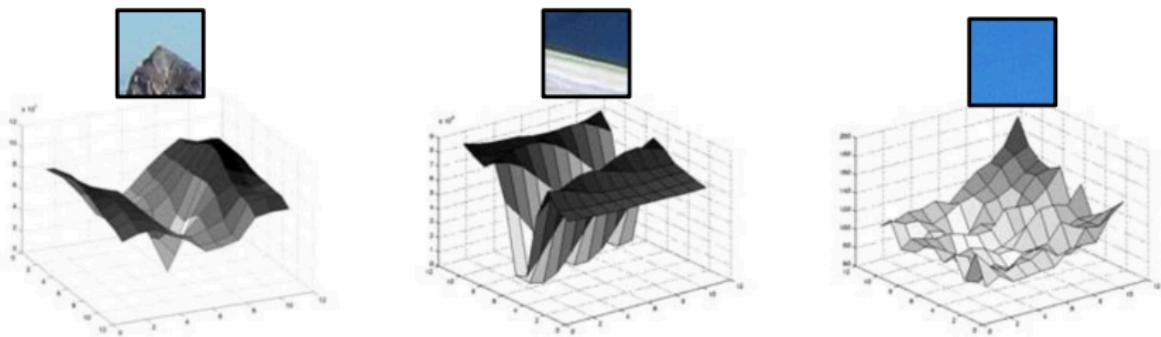
Corner: mostly high



Sky: low everywhere

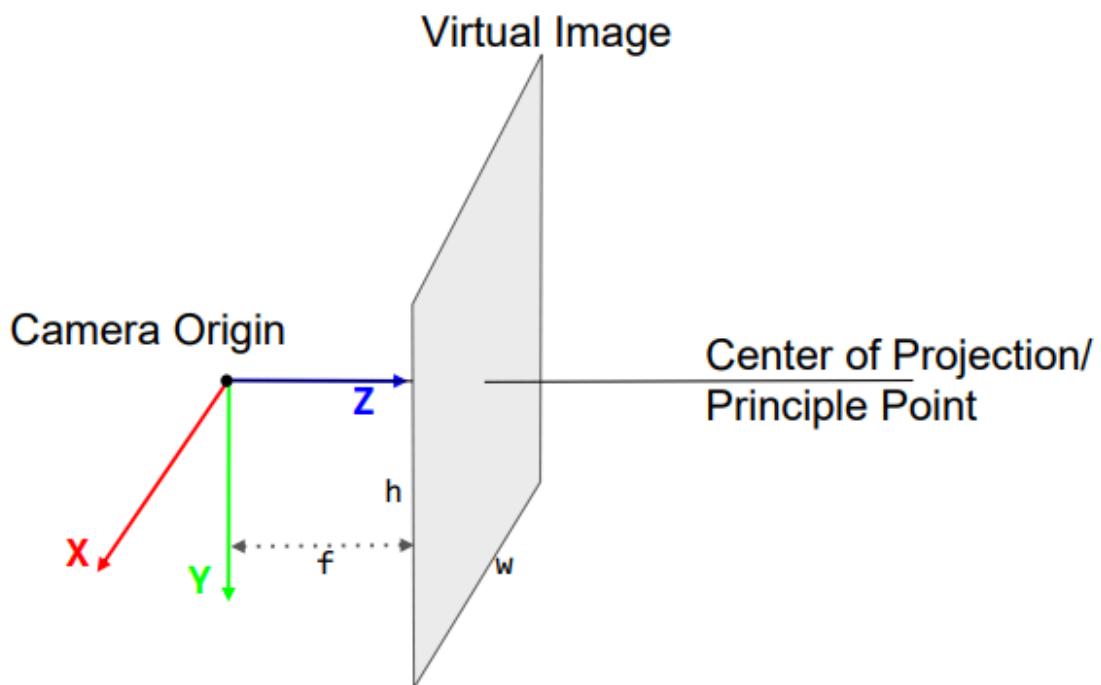
Edge: low along edge

Corner: mostly high



Stereo

Proiezioni Prospettiche

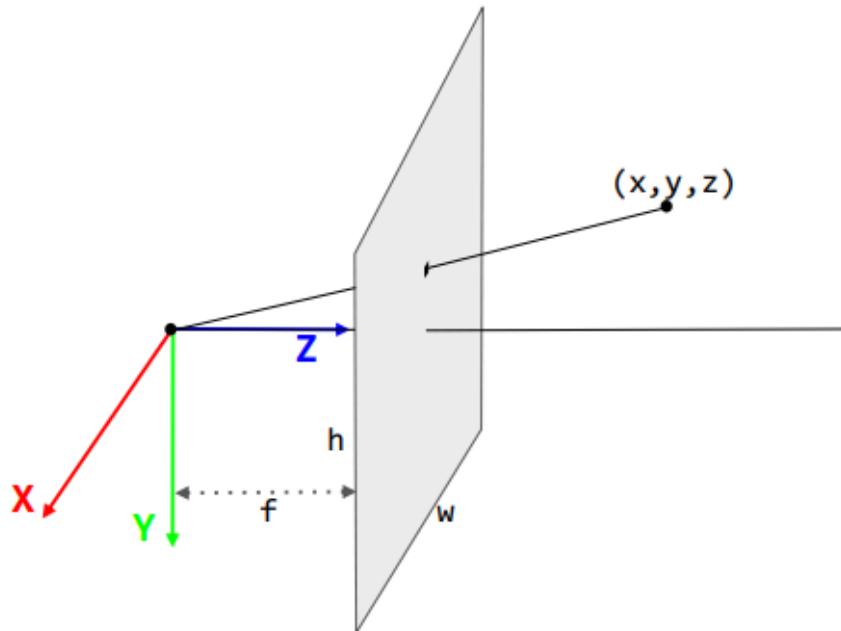


Vogliamo rappresentare qualcosa presente nelle 3 dimensioni all'interno di un piano bidimensionale. Conosciamo x, y, z che rappresentano le coordinate spaziali, e conosciamo anche f, w, h, p , ovvero, rispettivamente:

- **f: distanza focale** della lente, ovvero la distanza dal punto di vista da quello del piano dell'immagine
- **w: larghezza** del piano immagine
- **h: altezza** del piano immagine

- **p**: **pixel pitch**, indica la distanza tra i vari pixel (nel nostro caso di un sensore)

Immaginiamo di voler rappresentare un punto casuale presente nello spazio sul piano dell'immagine

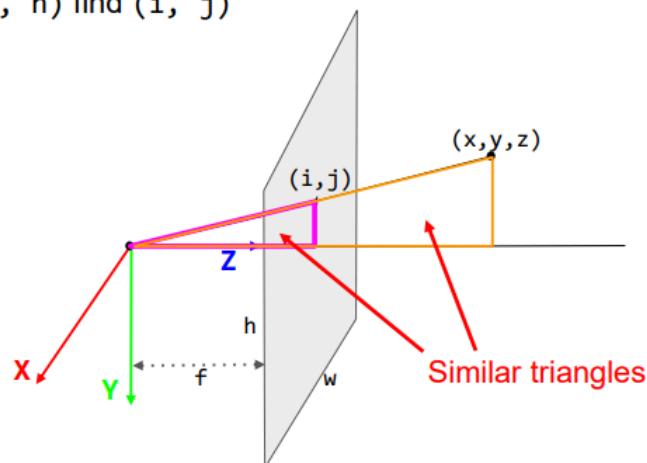


In questo caso conosciamo la sua **posizione** nello **spazio**, quindi le sue coordinate, ma quello che dobbiamo conoscere sono le sue coordinate **relative** al **piano**, che indichiamo con una coppia (i, j)

Given (x, y, z) and (f, w, h) find (i, j)

$$i = ?$$

$$j = ?$$

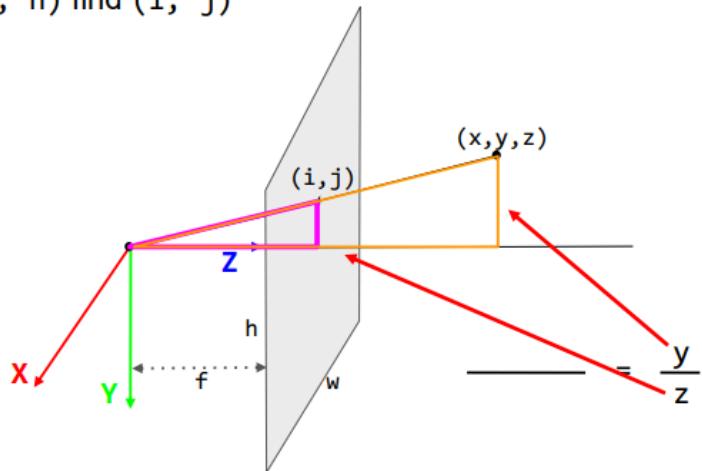


In questo caso, i due **triangoli** che si formano sono **simili**, pertanto

Given (x, y, z) and (f, w, h) find (i, j)

$$i = ?$$

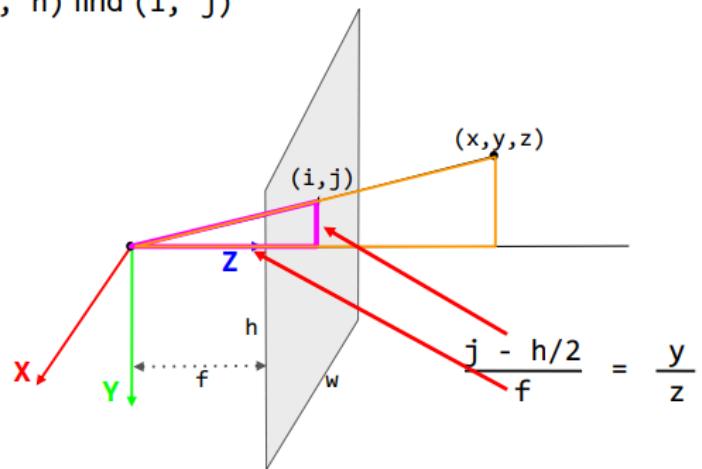
$$j = ?$$



Given (x, y, z) and (f, w, h) find (i, j)

$$i = ?$$

$$j = ?$$



Nel secondo triangolo, andremo ad usare questa relazione in quanto l'altezza sarà proprio calcolabile a partire dal **riferimento** ($h/2$, ovvero il punto centrale in altezza) e f sarà la sua **distanza** rispetto all'**osservatore**. Chiaramente, il numeratore dovrà essere espresso in pixel, mentre il denominatore in metri, pertanto è necessario moltiplicarlo per il pixel pitch, ovvero la distanza tra i pixel, quindi

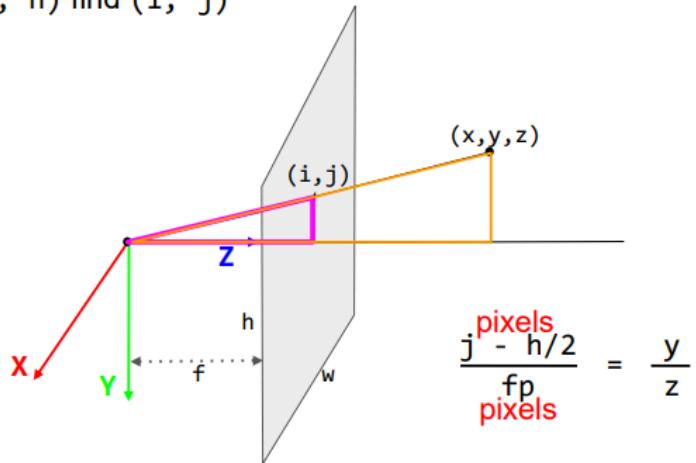
Given (x, y, z) and (f, w, h) find (i, j)

$$i = ?$$

$$j = ?$$

Pixel Pitch:

$$p = \frac{w \text{ (pixels)}}{w \text{ (meters)}}$$

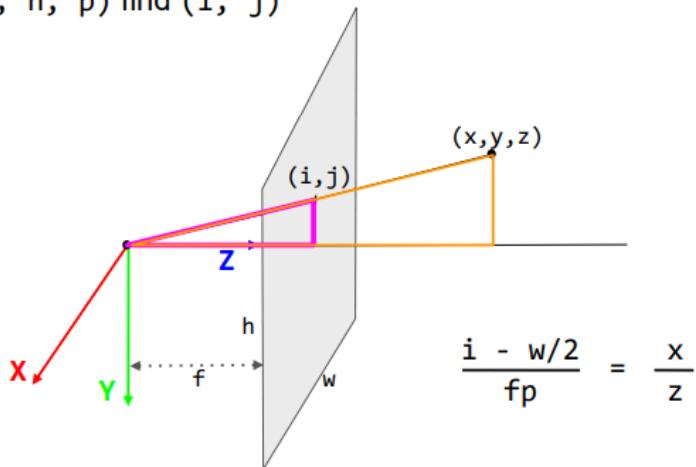


Successivamente andiamo a fare lo stesso calcolo per x e ci calcoliamo il valore esplicito di i e j :

Given (x, y, z) and (f, w, h, p) find (i, j)

$$i = \frac{fp x}{z} + \frac{w}{2}$$

$$j = \frac{fp y}{z} + \frac{h}{2}$$



Bisogna anche ricordare che i **parametri f,w,h,p** sono **intrinseci** alla fotocamera.

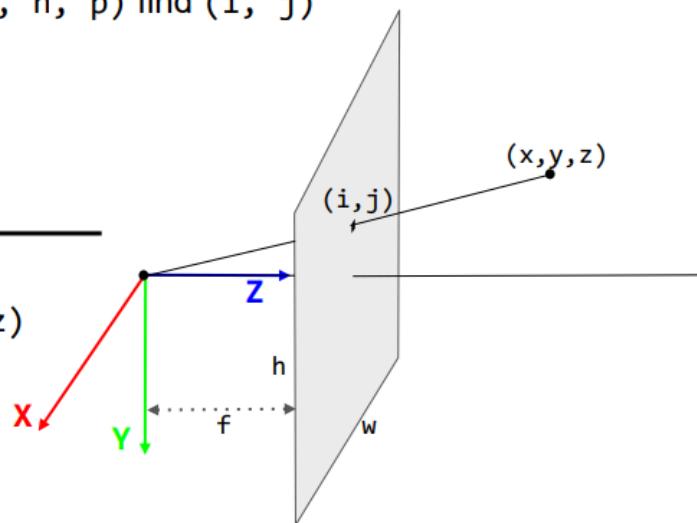
Qualora volessimo fare il calcolo opposto, ovvero conoscere le coordinate spaziali di un punto a partire dalle sue coordinate all'interno del piano dell'immagine, dovremmo usare queste relazioni:

Given (x, y, z) and (f, w, h, p) find (i, j)

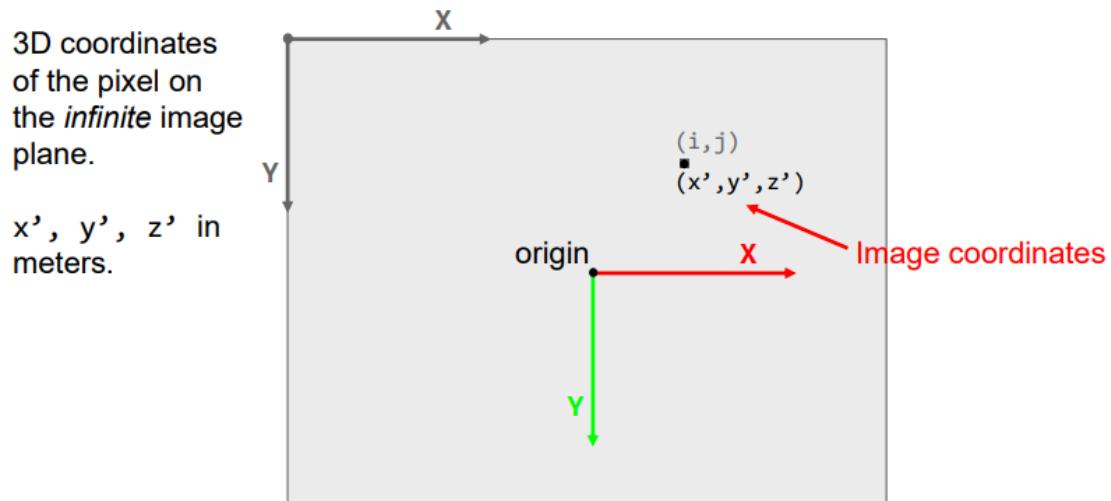
$$\begin{aligned} i &= \frac{fp \cdot x}{z} + \frac{w}{2} \\ j &= \frac{fp \cdot y}{z} + \frac{h}{2} \end{aligned}$$

Given (i, j, d) and (f, w, h, p) find (x, y, z)

$$\begin{aligned} x &= \frac{(i - w/2)d}{fp} \\ y &= \frac{(j - h/2)d}{fp} \\ z &= d \end{aligned}$$

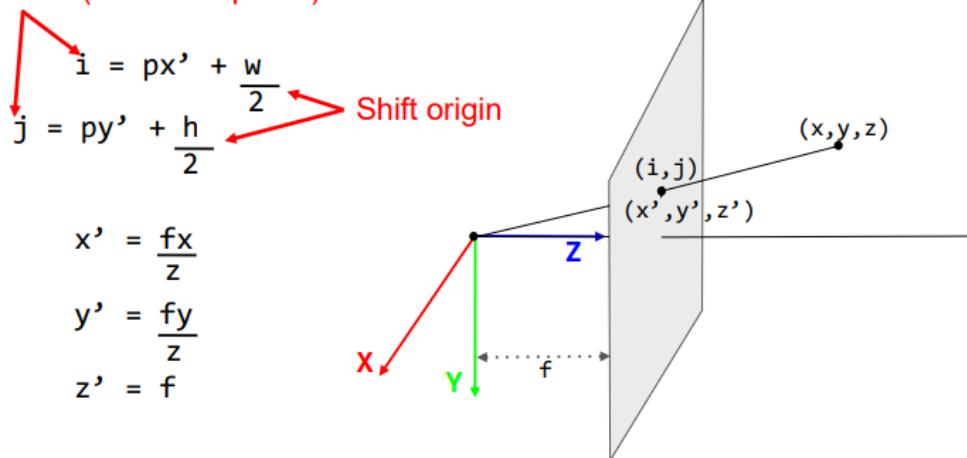


Dove chiaramente il valore della coordinata z è uguale proprio alla distanza che intercorre tra osservatore e il punto. Quello che vedremo sul piano è:



Da questo risultato, è possibile ottenere un risultato normalizzato, assumendo il caso generico in cui z' , ovvero la coordinata z' , quella della proiezione del punto sul piano immagine, nello spazio è pari alla distanza focale.

Re-scale (meters -> pixels)



Questo ci permette di effettuare una normalizzazione, considerando le coordinate ad una distanza =1

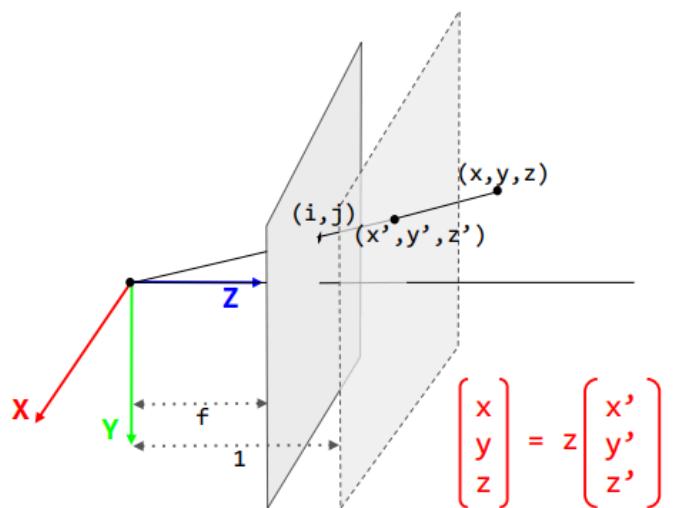
$$i = \cancel{f}px' + \frac{w}{2}$$

$$j = \cancel{f}py' + \frac{h}{2}$$

$$x' = \frac{x}{z}$$

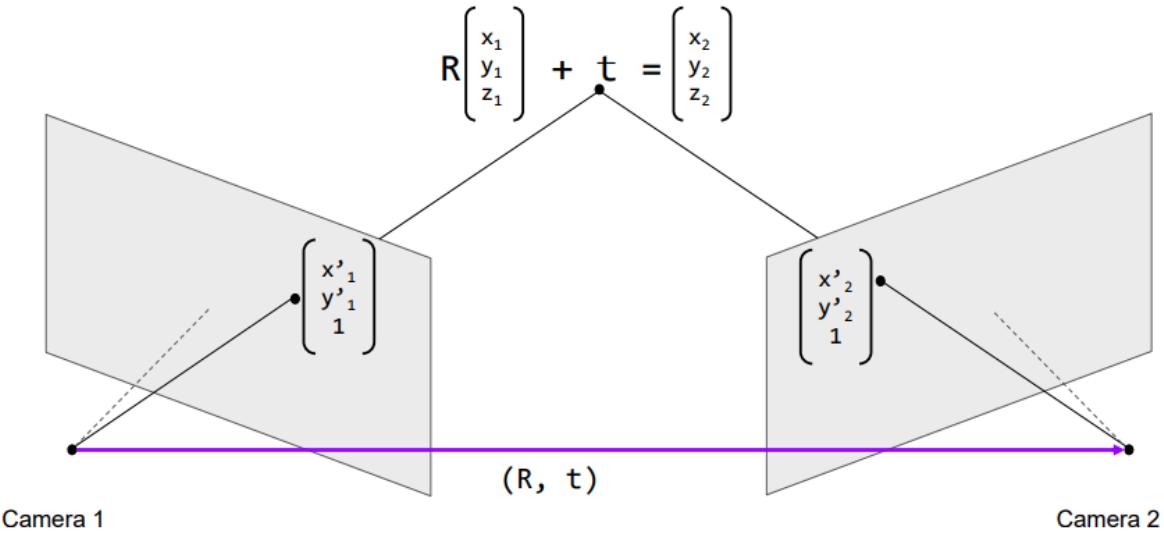
$$y' = \frac{y}{z}$$

$$z' = 1$$



Stereo

I due piani dell'immagine **differiscono** per un certo **theta** (da qui la matrice di rotazione) e per un vettore di traslazione. Infatti, esiste una certa matrice di rotazione e un certo vettore di traslazione per cui viene rispettata la seguente equazione



Utilizziamo un modo standard per rappresentare una retta nello spazio, ovvero:

A line in 3D can be represented as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} + S \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

Point on the line Parametric variable Slope of line

Quindi provvedo a calcolare le due rette rispetto al punto di vista della prima camera

A line in 3D can be represented as:

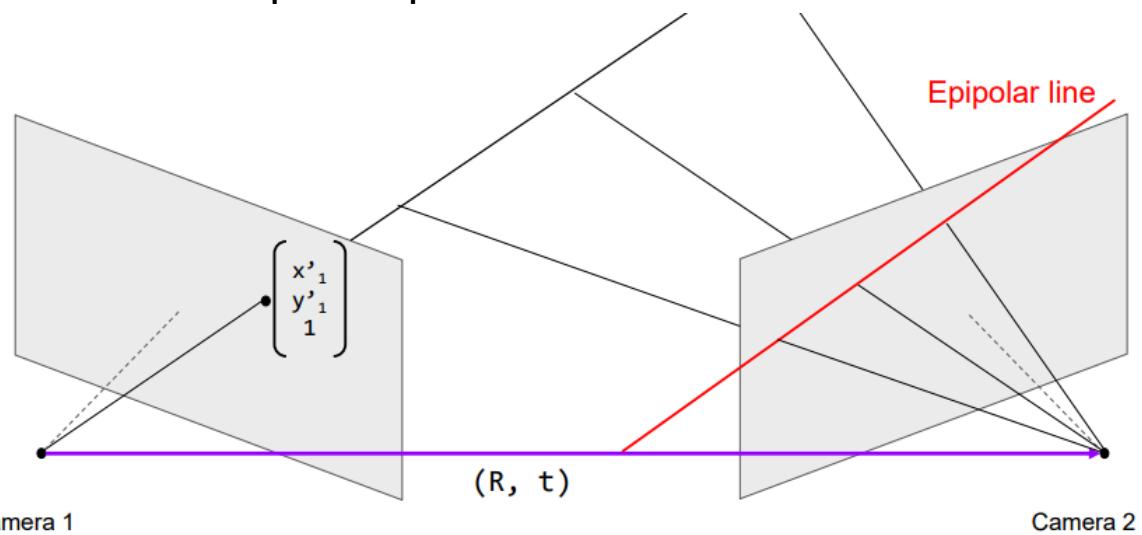
$$\text{Line 1} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + Z_1 \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix}$$

$$\text{Line 2} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + Z_2 \begin{bmatrix} x''_2 \\ y''_2 \\ 1 \end{bmatrix} \xrightarrow{\text{Transform to camera 1 coordinate frame}} -t + R^{-1}Z_2 \begin{bmatrix} x''_2 \\ y''_2 \\ 1 \end{bmatrix}$$

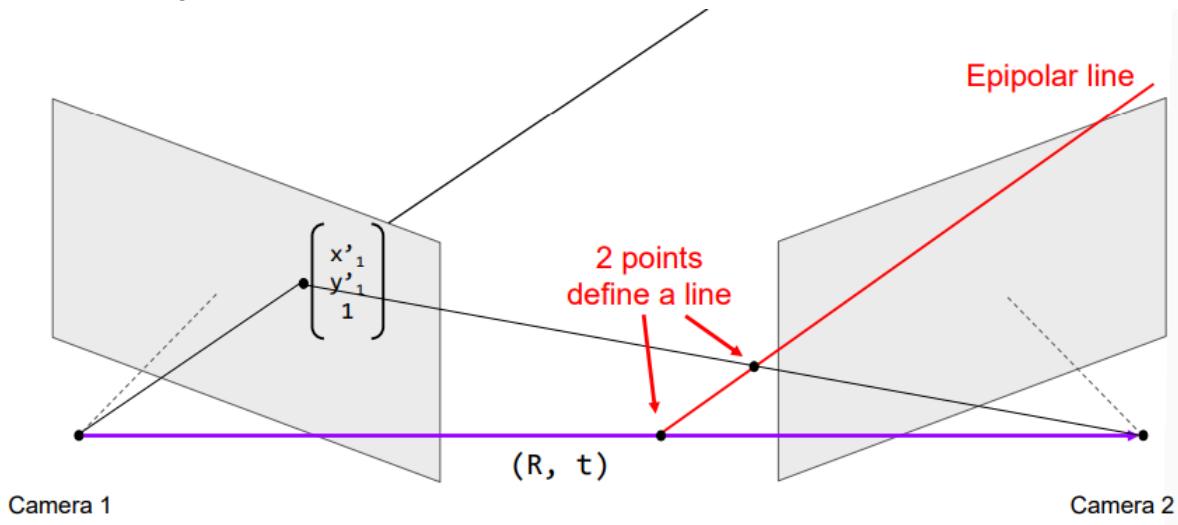
Solve system of linear equations
(Solution: Point P)

$$Z_1 \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} = -t + R^{-1}Z_2 \begin{bmatrix} x''_2 \\ y''_2 \\ 1 \end{bmatrix}$$

Per cercare il punto è inutile utilizzare tutta l'immagine, infatti sappiamo quasi con certezza dove saranno presenti i punti



In questo caso, quel determinato punto sarà presente in quel luogo dei punti nel piano della seconda camera, ovvero la **linea epipolare**, ovvero la proiezione della retta sul piano della seconda immagine. Ci viene in aiuto il fatto che due punti bastino a definire una retta, quindi



Dopo la normalizzazione, rispettivamente per la variabile parametrica $z1'$ per il punto in $(x1', y1', z1')$ e per la variabile parametrica $z1''$ per il punto in $(0, 0, 0)$, ottengo due vettori (vedendoli in coordinate relative alla camera 2, quindi rotati e traslati)

$$\left[t + R \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} \right] \frac{1}{z''_1}$$

$$\left[t + R \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right] \frac{1}{z'_1} \xrightarrow{\text{Simplify}} t - \frac{1}{t_z}$$

Una retta in due dimensioni non posso rappresentarla in funzione della y , ma devo rappresentarla in modo implicito, in quanto da problemi con la rappresentazione delle rette verticali

$$\cancel{y = mx + b}$$

$$ax + by + c = 0$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

Both our projected points have the format
We can pretend they are points in 2D. $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

In questo caso, se ottengo 0 nel prodotto scalare, vuol dire che i due vettori sono **ortogonali** rispetto al vettore (a, b, c) , quindi

$$ax + by + c = 0$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot \left[t + R \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix} \right] \frac{1}{z''_1} = 0$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot t - \frac{1}{t_z} = 0$$

Devo quindi imporre che il vettore (a, b, c) ossia la linea Epipolare sul piano della seconda camera, sia ortogonale ad entrambi i vettori, quindi posso porlo uguale al prodotto vettoriale dei due.

$$t \cdot \frac{1}{t_z} \times \left[t + R \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} \right] \cdot \frac{1}{z''_1} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\frac{1}{z''_1 t_z} t \times \left[t + R \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} \right] = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Nella seconda equazione semplifichiamo il termine scalare, poiché nella retta che si genererà a seguire un fattore di scala (s) non inciderebbe in alcun modo sulla pendenza. Quindi semplifico per $1/tz$ e inoltre, applicando la proprietà distributiva otterrei un primo addendo del tipo $(t \times t) = (0)$, quindi posso rimuovere anche il t nella parentesi quadra.

$$\frac{1}{z''_1 t_z} t \times \left[t + R \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} \right] = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$
~~$$\frac{1}{z''_1 t_z} t \times R \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$~~

$sax + sby + sc = 0$

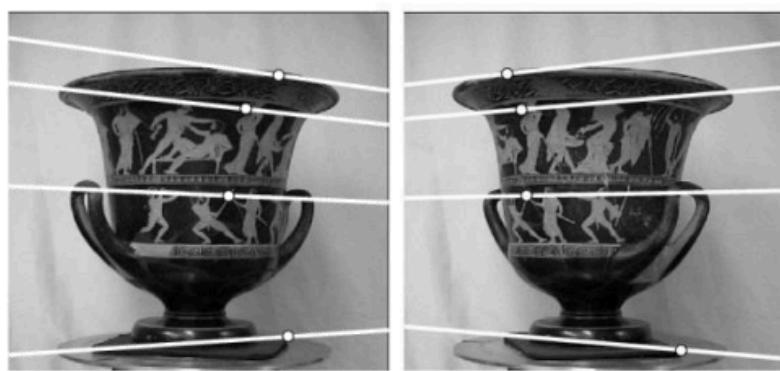
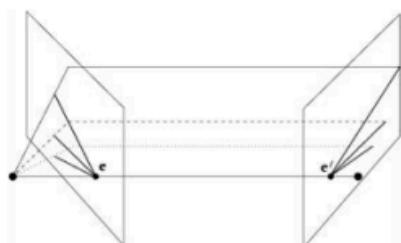
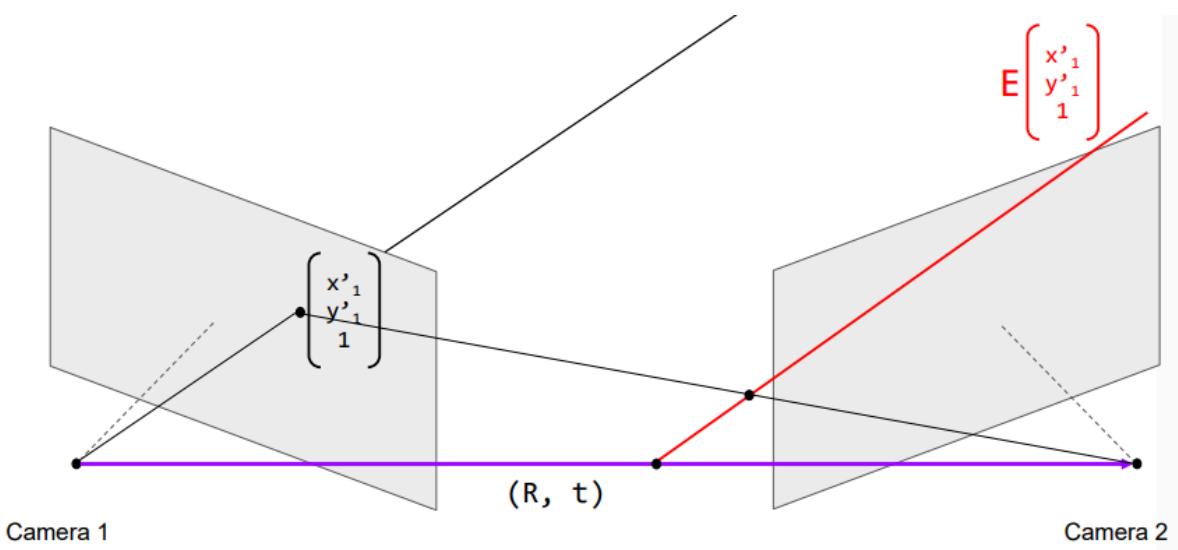
$$s \begin{pmatrix} a \\ b \\ c \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

La **Matrice Essenziale** si definisce come il prodotto vettoriale tra vettore trasposizione e matrice di rotazione, pertanto il risultato complessivo sarà

E → $t \times R \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$

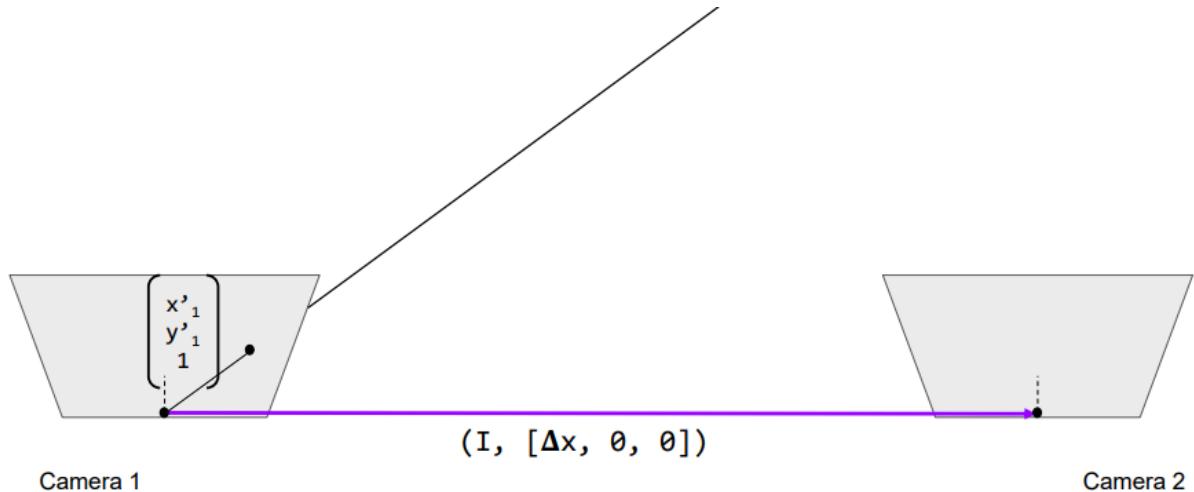
essential matrix
(Longuet-Higgins, 1981)

$$E \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$



Camere parallele

In questo caso, la relazione presente tra le due camere è identica a quella precedente, con la differenza che la matrice di rotazione ora diventa una matrice identità (perché l'angolo vale 0, quindi i valori del seno si annullano e quelli del coseno valgono 1), mentre il vettore di traslazione ha soltanto il parametro dello spostamento lungo l'asse x, com'è facilmente comprensibile



Si procede come nel caso precedente:

$$t \times R \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Applicando questa formula al caso attuale, abbiamo questo calcolo. Possiamo ignorare la matrice identità e successivamente calcolarci la retta

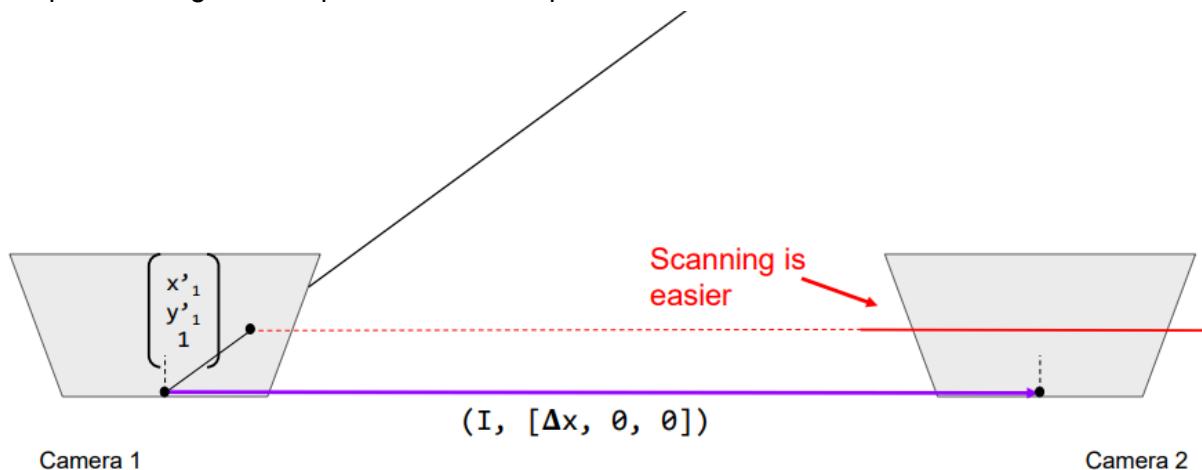
$$\begin{pmatrix} \Delta x \\ 0 \\ 0 \end{pmatrix} \times I \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\begin{pmatrix} \Delta x \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ -\Delta x \\ \Delta x y' \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

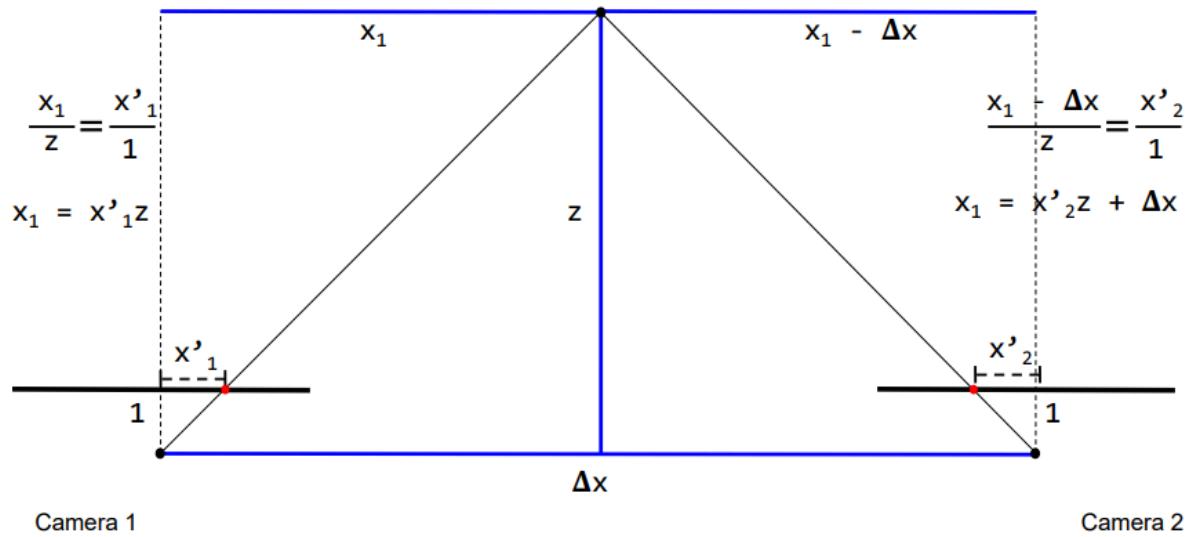
$$(0)x + (-\Delta x)y + (\Delta x y') = 0 \longrightarrow y = y'$$

Il risultato ottenuto ci suggerisce che la linea epipolare è una linea parallela all'asse x, giace sui piani immagine, dunque rende facile il processo.



A questo punto è possibile valutare geometricamente. Immaginando un certo punto, è possibile fare questa costruzione. Partiamo osservando le coppie di triangoli simili, come:

- quello costruito avendo come cateti x_1 e z e l'altro costruito con x'_1 e 1 come cateti, sono simili.
- Vale lo stesso per i triangoli costruiti sulla seconda camera. Questo ci permette di metterli in relazione e calcolarci il valore di x_1



Il prossimo passaggio è quello di eguagliare le due distanze, quindi senza conoscere la reale posizione x_1 dell'oggetto è possibile ottenere così la misura esplicita della profondità del punto (z).

Infatti conosciamo già Δ_x (distanza tra le camere), x'_1 e x'_2 ossia le ascisse del punto sui due piani.

$$x'_{1z} = x'_{2z} + \Delta x$$

$$x'_{1z} - x'_{2z} = \Delta x$$

$$(x'_{1z} - x'_{2z})z = \Delta x$$

$$z = \frac{\Delta x}{x'_{1z} - x'_{2z}}$$

Il fatto che i piani immagine siano paralleli semplifica di molto le cose: infatti mantenendo la stessa ordinata tra i due piani, ciascun punto sarà più facilmente riconoscibile (riconoscimento tramite similitudine grafica)

Algorithm (assuming same intrinsics)

```

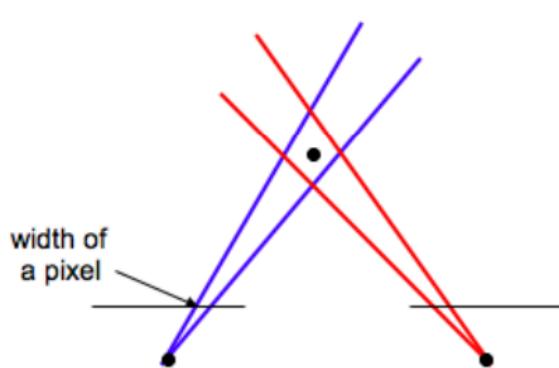
compute_depth(I1, I2, Δx, f, p):
    depth <- new_image_like(I1)
    For every pixel i,j in I1:
        fij = get_feature_descriptor(I1, i, j)
        best_score <- INFINITY
        best_i <- NaN
        For i' in columns(I2):
            fi'j = get_feature_descriptor(I2, i', j)
            score = score_feature_match(fij, fi'j)
            If score < best_score:
                best_score = score
                best_i = i'
        depth[i, j] = fpΔx/(i - best_i)
    Return depth

```

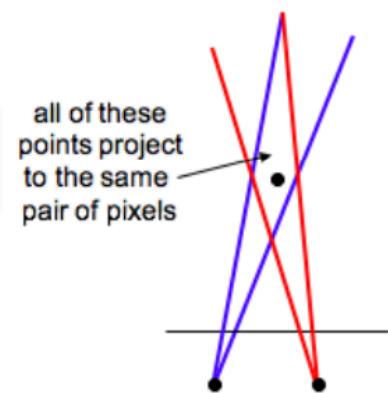
The formula uses the focal length (the previous triangle formed on x'_{1z} was normalized) and the pixel pitch.

Una volta ottenuto z , è facilmente ricavabile anche x_1 , ossia l'ascissa del punto nel sistema di riferimento principale del dispositivo.

Problemi

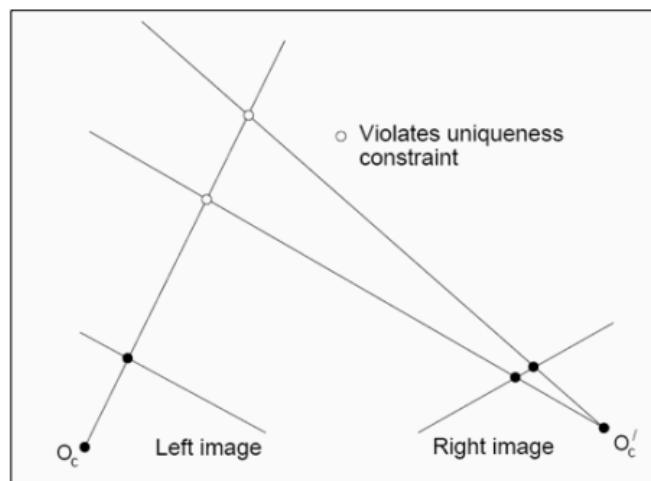


Large Baseline



Small Baseline

- Uniqueness: For any point in one image, there should be at most one matching point in the other image
- Ordering: Corresponding points should be in the same order in both views
- Smoothness: We expect disparity values to change slowly (for the most part)



Esercizi