ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA
Dipartimento di Informatica - Scienza e Ingegneria (DISI)

Tesi di Laurea in
COMPUTER VISION AND IMAGE PROCESSING

STUDIO DI UN SISTEMA DI DIMENSIONAMENTO VOLUMETRICO MEDIANTE SENSORI DI PROFONDITÀ

Candidato:
Maurizio Ingrassia

Relatore:
Chiar.mo Prof.
Luigi Di Stefano

Correlatore:
Ing.
Alessandro Franchi

Anno Accademico 2013/2014
Sessione III

A tutti quelli che ci son sempre stati,
e a chi spero rimarrà sempre.

Indice

Introduzione

Negli ultimi anni si è assistito ad una diffusione sul mercato di diverse camere 3D progettate per ambiti non industriali. Il mercato videoludico si è spinto verso questo settore come offerta per un diverso tipo di intrattenimento, spingendo per una produzione di massa con conseguente abbattimento dei costi di produzione.
Tutto questo ha fatto si che sia possibile trovare camere 3D che offrono discreti risultati in termini di risoluzione e precisione e che siano accessibili economicamente ad una utenza di massa. Ciò porta a chiedere se e in che misura sia possibile utilizzare questi strumenti per delle applicazioni di tipo industriale.

Lo scopo di questa tesi è quello di effettuare una panoramica su modelli di telecamere 3D di tipo industriale e non industriale all'interno di un contesto di Machine Vision e attraverso di esse proporre un sistema di dimensionamento volumetrico attraverso il software Datalogic Impact e fornire una valutazione sperimentale dei risultati.

Nel Capitolo 1 presentiamo una panoramica sullo stato dell'arte inerente alla Computer Vision e sul processo che porta dall'acquisizione di una immagine fino alla sua comprensione, dalle tecniche di denoising all'analisi dei blob, con una sezione . Continueremo quindi dando una definizione di Machine Vision e quali sono le principali problematiche che si focalizzano su questa branchia.

Nel Capitolo 2 daremo una definizione di camere 3D e alle differenti tecnologie che le caratterizzano, a quali sono i metodi di acquisizione e relativi punti di forza di ognuno. Ci soffermeremo in particolar modo sui modelli di camere Kinect e Kinect 2 che sono state utilizzate per la realizzazione del sistema propoto.

Nel Capitolo 3 presenteremo le valutazioni e i processi effettuati per giungere a un sistema di dimensionamento volumetrico, ottenuto mediante l'uso del software di visione industriale della Datalogic Impact.

Nel Capitolo 4 illustreremo i dati sperimentali raccolti attraverso le valutazioni fatte tramite il sistema precedentemente proposto.

Capitolo 1

Stato dell'Arte

La *Computer Vision* rappresenta un ramo dell'Intelligenza Artificiale che punta all'analisi e interpretazione di informazioni visive a partire da una immagine o una sequenza di immagini. [MMK01] L'elaborazione di informazioni semantiche, atte alla "comprensione" dell'immagine stessa passa attraverso varie fasi. Il primo passo riguarda ovviamente l'acquisizione dell'immagine attraverso una camera. Camere più complesse possono essere dotate di più sensori per acquisire informazioni aggiuntive. Nella maggior parte dei casi si tratta di sensori di profondità, ovvero capaci di inferire distanze rispetto agli oggetti inquadrati nella scena; si parla quindi in questo caso di *Camere 3D* (di cui si parlerà in maniera più esaustiva nel capitolo successivo).

1.1 Dall'immagine all'informazione

Una volta che è stata catturata l'immagine, si effettua una fase chiamata di pre-processing, in cui l'immagine viene filtrata tramite algoritmi di *Image Processing*. Lo scopo di queste tecniche è quello di esaltare quel tipo di tratti all'interno dell'immagine che sono rilevanti per l'analisi dei contenuti, o viceversa di escludere quei tratti che possono essere di disturbo per la valutazione dei dati. Normalmente, a meno che il colore non sia una componente che vada esplicitamente analizzata, l'immagine viene trasformata in scala di grigio in quanto:

- la dimensione dell'immagine è notevolmente inferiore, passando da 3 o 4 canali a uno soltanto;
- rende più semplice l'individuazione delle regioni di interesse all'interno dell'immagine;
- spesso diviene utile ricorrere a una binarizzazione;

Dato che, normalmente, si ci trova a dover rimuovere del rumore, ovvero un segnale di disturbo che può dipendere sia da fattori ambientali come illuminazione della scena, qualità del sensore di acquisizione, tempi di esposizione, proprietà intrinseche dei materiali presenti (in riferimento a materiali lucidi, trasparenti, riflettenti o neri che possono in base alla tecnologia di acquisizione del sensore distorcere l'acquisizione dell'oggetto stesso) molti algoritmi di Image Processing riguardano le cosiddette *denoising techniques*.

1.1.1 Tecniche di preprocessing

Le tecniche di preprocessing possono essere suddivise in due grandi categorie, gli operatori puntuali e gli operatori locali. Gli operatori puntuali utilizzano una funzione di trasformazione che viene applicata singolarmente per ogni pixel, del tipo

$$y = T(x)$$

questo comporta una modifica uniforme all'immagine, è quindi una scelta valida quando dobbiamo andare a trattare una caratteristica gloable dell'immagine come ad esempio il contrasto. tra i più comuni operatori troviamo:

- contrast stretching;
- operatore esponenziale;
- equalizzatore dell'istogramma,

Gli operatori locali considerano invece per il generico pixel $P(i, j)$ anche ciò che si trova nell'intorno del pixel stesso, ovvero considerando una finestra di dimensione w centrata nel pixel

$$y = T(P(i, j), w)$$

Lo scopo di questi filtri è migliorare la qualità dell'immagine, tentando però di preservarne le caratteristiche. Vengono definiti filtri a tutti gli effetti perché l'immagine viene trattata come un segnale nel dominio delle frequenze in cui viene filtrato il rumore. Come già detto, questi filtri vengono utilizzati per la rimozione del rumore di acquisizione e/o trasmissione, ma anche come *enhancement* di feature significative e rafforzamento dei contorni. Tra i più famosi filtri locali troviamo:

- filtro mediano;
- filtro di media;

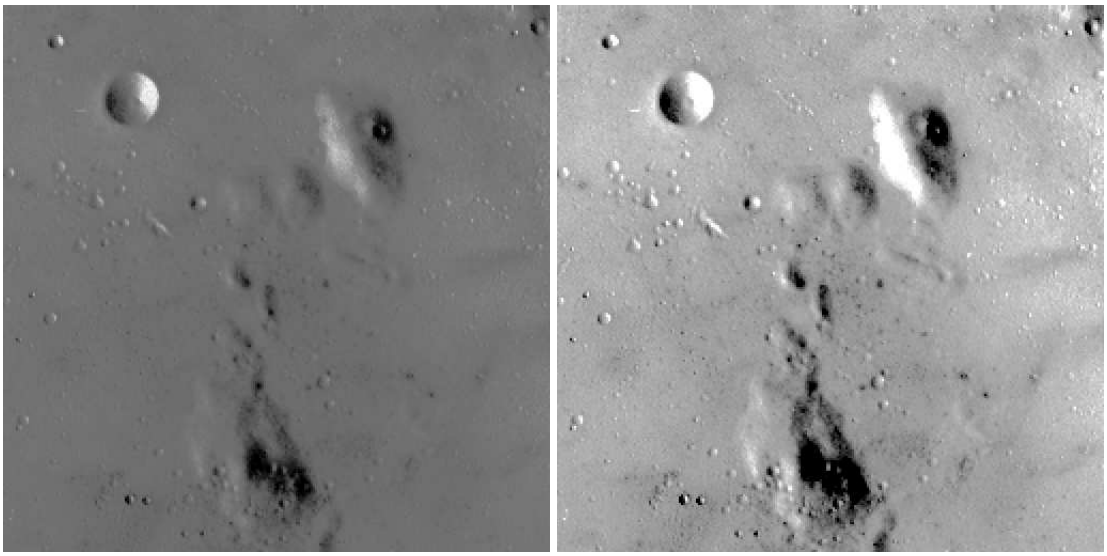


Figura 1.1: Esempio di applicazione di contrast stretching

- filtro Gaussiano;
- filtro di sharpening;
- filtro bilaterale;

Per una trattazione più approfondita riguardo i metodi di denoising si faccia riferimento a [Gri10] [SCTM07].

1.1.2 Blob Analysis

Terminata la fase di esaltazione delle caratteristiche salienti, si passa alla fase di vera e propria analisi. Bisogna adesso individuare e separare gli oggetti in primo piano da quelli di sfondo. Le tecniche che ci vengono in aiuto per questa operazione sono quelle di *blob analysis*.
Un *Blob* (Binary Large OBject) è una regione connessa dell'immagine che siamo risuciti a isolare dal resto, a cui solitamente associamo un oggetto della scena. L'individuazione dei blob nell'immagine ci permette di trovare e isola le componenti che stiamo ricercando per effettuare una ulteriore analisi.
L'analisi dei blob trova largo impiego all'interno del controllo di qualità, ed è quindi molto

[1] http://bigwww.epfl.ch/teaching/projects/abstracts/yuce/

Figura 1.2: Esempio di filtro bilaterale.[1]

utilizzato nei processi di machine vision. Consente di trovare facilmente per esempio scritte e codici su dei pacchi, o di effettuare dei controlli di integrità sugli oggetti. Un filamento all'interno di un'area può essere facilmente identificato come un blob, ma qualora fosse spezzato potrebbero esserne rilevati due anziché uno, questo starebbe ad indicarci che vi è qualcosa di anomalo di cui è immediato potersi accorgere.
L'analisi dei blob inoltre, permette di effettuare delle stime sulla forma che stiamo analizzando. Isolata una forma possiamo calcolarne valori come area, perimetro e baricentro, più altri che risultano essere invarianti alle trasformazioni di traslazione e scala. Le informazioni più rilevanti che possiamo trarre da una blob analysis sono quelle dei *momenti*.

Il momento è definito come

$$M_{m,n} = \sum_{p \in R} u^m v^n$$

Da questo possiamo trarre subito diverse informazioni, come l'area, che corrisponde a $M_{0.0}$ e del centroide

$$\tilde{u} = \frac{M_{0.0}}{M_{1.0}}, \tilde{v} = \frac{M_{0.1}}{M_{0.0}}$$

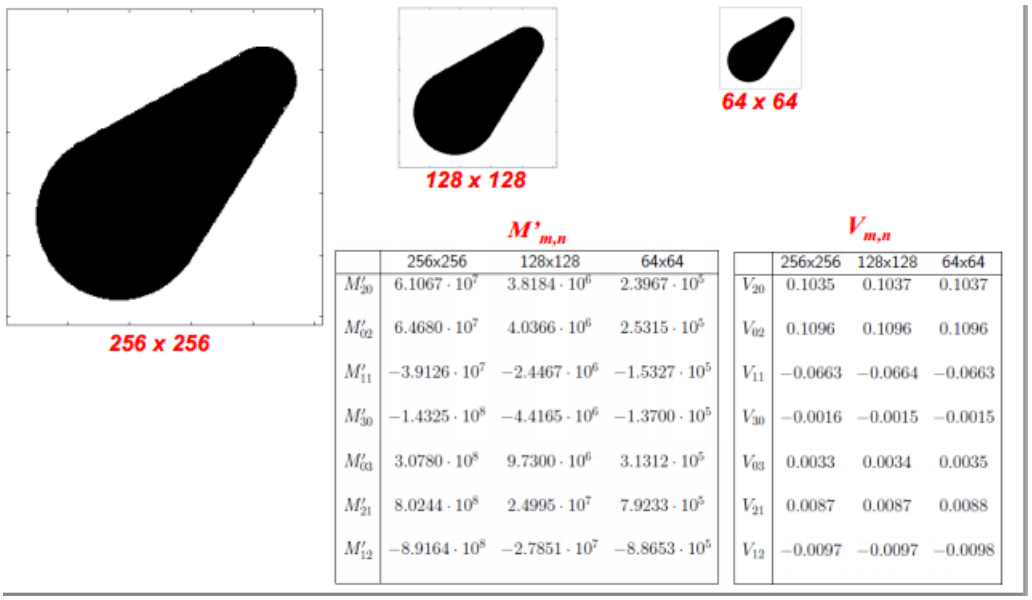
Inoltre il momento può facilmente essere reso invariante alla traslazione, semplicemente calcolando il momento centrale, relativo rispetto alla posizione del baricentro:

$$M'_{m,n} = \sum_{p \in R} (u - \tilde{u})^m (v - \tilde{v})^n$$

per ottenere anche l'invarianza alla scala, è necessario che il momento centrale venga normalizzato

$$V_{m,n} = \frac{M'_{m,n}}{A^a}, a = \frac{m+n}{2}$$

Ottenere dei momenti invarianti a traslazione e scala è molto utile, perché ci consentono di poter trovare uno stesso oggetto anche se inquadrato in posizioni, orientamento e distanza differente. L'immagine in figura 1.3 mostra chiaramente come i momenti risultino invariati al variare della scala.



$M'_{m,n}$

	256x256	128x128	64x64
M'_{20}	$6.1067 \cdot 10^7$	$3.8184 \cdot 10^6$	$2.3967 \cdot 10^5$
M'_{02}	$6.4680 \cdot 10^7$	$4.0366 \cdot 10^6$	$2.5315 \cdot 10^5$
M'_{11}	$-3.9126 \cdot 10^7$	$-2.4467 \cdot 10^6$	$-1.5327 \cdot 10^5$
M'_{30}	$-1.4325 \cdot 10^8$	$-4.4165 \cdot 10^6$	$-1.3700 \cdot 10^5$
M'_{03}	$3.0780 \cdot 10^8$	$9.7300 \cdot 10^6$	$3.1312 \cdot 10^5$
M'_{21}	$8.0244 \cdot 10^8$	$2.4995 \cdot 10^7$	$7.9233 \cdot 10^5$
M'_{12}	$-8.9164 \cdot 10^8$	$-2.7851 \cdot 10^7$	$-8.8653 \cdot 10^5$

$V_{m,n}$

	256x256	128x128	64x64
V_{20}	0.1035	0.1037	0.1037
V_{02}	0.1096	0.1096	0.1096
V_{11}	−0.0663	−0.0664	−0.0663
V_{30}	−0.0016	−0.0015	−0.0015
V_{03}	0.0033	0.0034	0.0035
V_{21}	0.0087	0.0087	0.0088
V_{12}	−0.0097	−0.0097	−0.0098

Figura 1.3: Immagine scalata e rispettivi momenti centrali.[2]

Esiste anche una definizione di momento che possiede anche la proprietà di invarianza alla rotazione, i *momenti di Hu*. Per una maggiore trattazione dell'argomento, fare riferimento a [Li10] [vHR10] [ZXPZ12].

[2] http://didattica.arces.unibo.it/mod/resource/view.php?id=385

1.2 Rappresentazione dei dati 2D e 3D

1.2.1 Rappresentazioni 2D

1.2.1.1 RGB

RGB è la rappresentazione dello spazio di colore più comunemente utilizzata per la rappresentazione di immagini. Questo modello è di tipo additivo, ovvero che la miscelazione delle componenti primarie dà luogo al colore bianco i cui colori primari sono rosso (**R**ed), verde (**G**reen) e blu (**B**lue).
La scelta di questi colori è strettamente collegata alla fisiologia dell'occhio umano, in quanto massimizzano la differenza di risposta delle cellule cono della retina dell'occhio umano alle differenze di lunghezza d'onda della luce. Benché il mix di tre colori primari non sia sufficiente a riprodurre tutti i colori percepibili esso genera un esteso triangolo di colore, che può comunque essere compensato aggiungendo ulteriori componenti di colore stesso. [Hun69]

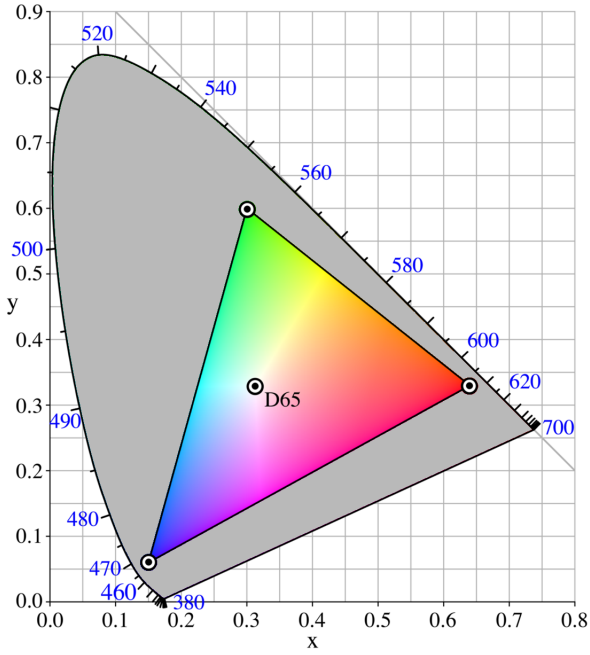


Figura 1.4: Spazio di colore RGB

All'interno della stessa definizione di spazio di colore, vi sono quindi diversi modi per rappresentare i dati. I dati in RGB posseggono 3 canali, uno per colore, in cui vengono rappresentati i singoli valori appartenenti ad ogni pixel. Si ottiene quindi l'immagine rappresentata

come un vettore in cui elencati in ordine per ogni pixel le singole componenti in ordine, come mostrato in Figura 1.5.

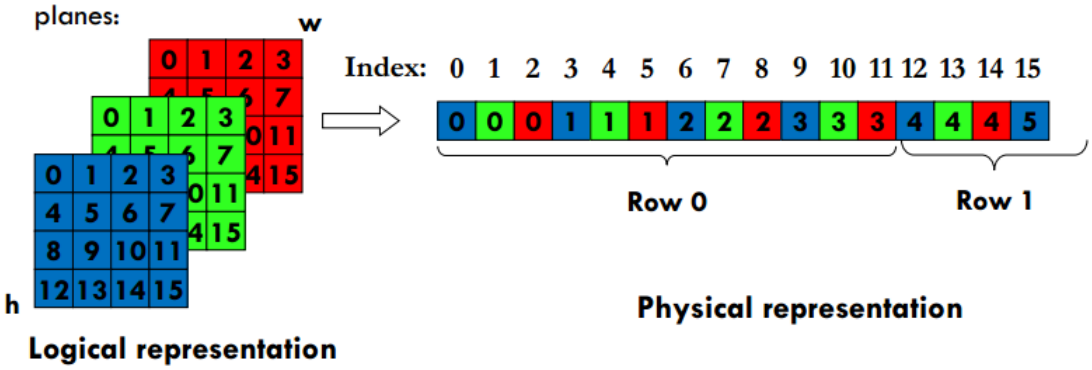


Figura 1.5: Struttura di una immgine RGB [Tomb]

Troviamo poi lra rappresentazione *RGBA*, dotata di 4 canali anziché 3. Il canale aggiuntivo *Alpha* definisce il grado di opacità di ciascun pixel. Questo consente di creare effetti di trasparenza e sovrapposizioni tra oggetti che sono definibili come lucidi. Trova quindi largo impiego all'interno del *digital composing*, ovvero quando si rivela necessario sovrapporre diverse immagini o parti di esse all'interno per crearne una unica. Per ulteriori approfondimenti a riguardo si faccia riferimento a [PD84].

1.2.1.2 Grayscale

Come già anticipato precedentemente, l'analisi del colore in una immagine non è sempre necessaria, questo perché generalmente richiede un sovraccarico in termini di computazioni che rendono il tempo di calcolo eccessivo per il lavoro richiesto (spesso infatti per un'analisi robusta del colore è necessario ricorrere ad un training del sistema). Per lo stesso motivo, l'utilizzo di 3 o 4 canali significa trattare una mole di dati 3 o 4 volte superiore al numero di pixel da analizzare.
In definitiva, se non strettamente necessario, non risulta conveniente lavorare su immagini a colori, si preferisce quindi usare la *scala di grigio*.
Per le immagini in scala di grigio è sufficiente usare uno solo canale, al cui interno per ogni pixel vi è rappresentato un valore di *intensità* calcolato a partire dalle informazioni sul colore. Istintivamente si tende a considerare come valore di intensità una media tra i valori di rosso verde e blu appartenenti a un determinato pixel, ed è effettivamente è il metodo più rapido, ma non l'unico. Vediamo qualche esempio di calcolo di intensità possibile e il suo risultato:

- Media: $I(p) = (R(p) + G(p) + B(p))/3$
- Luma: $I(p) = 0.2126R(p) + 0.7152G(p) + 0.0722B(p)$ [AMCS96]
- Decomposizione minima: $I(p) = max\{R(p), G(p), B(p)\}$
- Decomposizione massima: $I(p) = min\{R(p), G(p), B(p)\}$

Figura 1.6: Immagine RGB

Figura 1.7: Immagini Grayscale col metodo Average (sx) e Luma (dx) [Hel]

Figura 1.8: Immagini Grayscale col metodo di decomposizione minima (sx) e massima (dx) [Hel]

1.2.2 Rappresentazioni 3D

Le rappresentazioni dei dati in tre dimensioni vengono utilizzate per la riproduzione di superfici di oggetti in 3D. Attraverso una collezione di punti appartenente in uno spazio 3D, opportunamente connesse tramite entità geometriche come linee, triangoli e curve consentono di modellare forme e oggetti.
I dati 3D possono essere raggruppati in due categorie, quelle di dati organizzati e non organizzati. I tipi di dati non organizzati si limitano a elencare i dati ed eventuali connessioni senza una rappresentazione topologica, risulta quindi spesso costoso per questi tipi calcolare i propri "vicini". I tipi organizzati invece, hanno una qualche struttura a supporto (come una matrice ad esempio). A seguire ci limiteremo a esporre i tipi di rappresentazioni attinenti a questo lavoro.

1.2.2.1 Point Cloud

La più comune rappresentazione in 3D, è una semplice collezione di dati 3D di tipo (x, y, z). Essendo di base non organizzata, anche qui si ha il problema del calcolo dei vicini di un punto. La struttura può essere organizzata con l'inserimento in una matrice in cui punti punti adiacenti nella matrice coincidono in pixel adiacenti dell'immagine, rendendo così le operazioni di ricerca più efficienti. A partire da una point cloud inoltre si può facilmente passare a una rappresentazione sia a *voxel* che a *mesh*.

Figura 1.9: Stanford Bunny e una sua PointCoud.[3]

1.2.2.2 Range Image

La range image (depth map o depth image) è un'immagine in cui per ogni pixel viene mostrata la distanza rispetto a un punto di vista (generalmente il centro dell'obiettivo della telecamera). La differenza di profondità viene generalmente mostrata tramite una variazione di gradazione in scala di grigio, dove i vari valori vengono quantizzati per rientrare in un range tra 0 e 255, anche se nulla comunque vieta una rappresentazione a colori tramite gradiente.
Per ottenere questo tipo di immagine la soluzione più usata è la *visione stereo*, ovvero l'acquisizione di un scena da due (o più) telecamera poste a distanza nota tra loro. Questa tecnica prende spunto dalla stessa visione umana, dove un oggetto viene osservato da due punti di vista differenti (occhio sinistro e destro) che da la percezione della profondità. Allo stesso modo, eseguita l'acquisizione possiamo cercare i pixel che appaiono nella scena sinistra e ricercarli all'interno di quella destra, ottenendo delle distanze che andranno a rappresentare appunto la mappa delle disparità.
Una alternativa all'uso della visione stereo è quello di utilizzare specifici sensori detti *di profondità*, questi usano pattern di luce strutturata o sfruttano il tempo di volo per calcolare una distanza con una singola inquadratura, ma di questo si parlerà specificatamente nel prossimo capitolo. L'utilizzo delle depth images aiuta la comprensione della scena, degli elementi che la compongono, e la disposizione di quest'ultimi all'interno della scena stessa. Ad esempio,

[3] http://graphics.stanford.edu/data - http://www.ccs.neu.edu/home/gaob

rende molto più semplice la separazione tra oggetti in primo piano e oggetti di sfondo, che se anche presentano un colore o una intensità molto simili tra loro, possono essere facilmente essere riconosciuti e catalogati come parti del primo o secondo insieme.

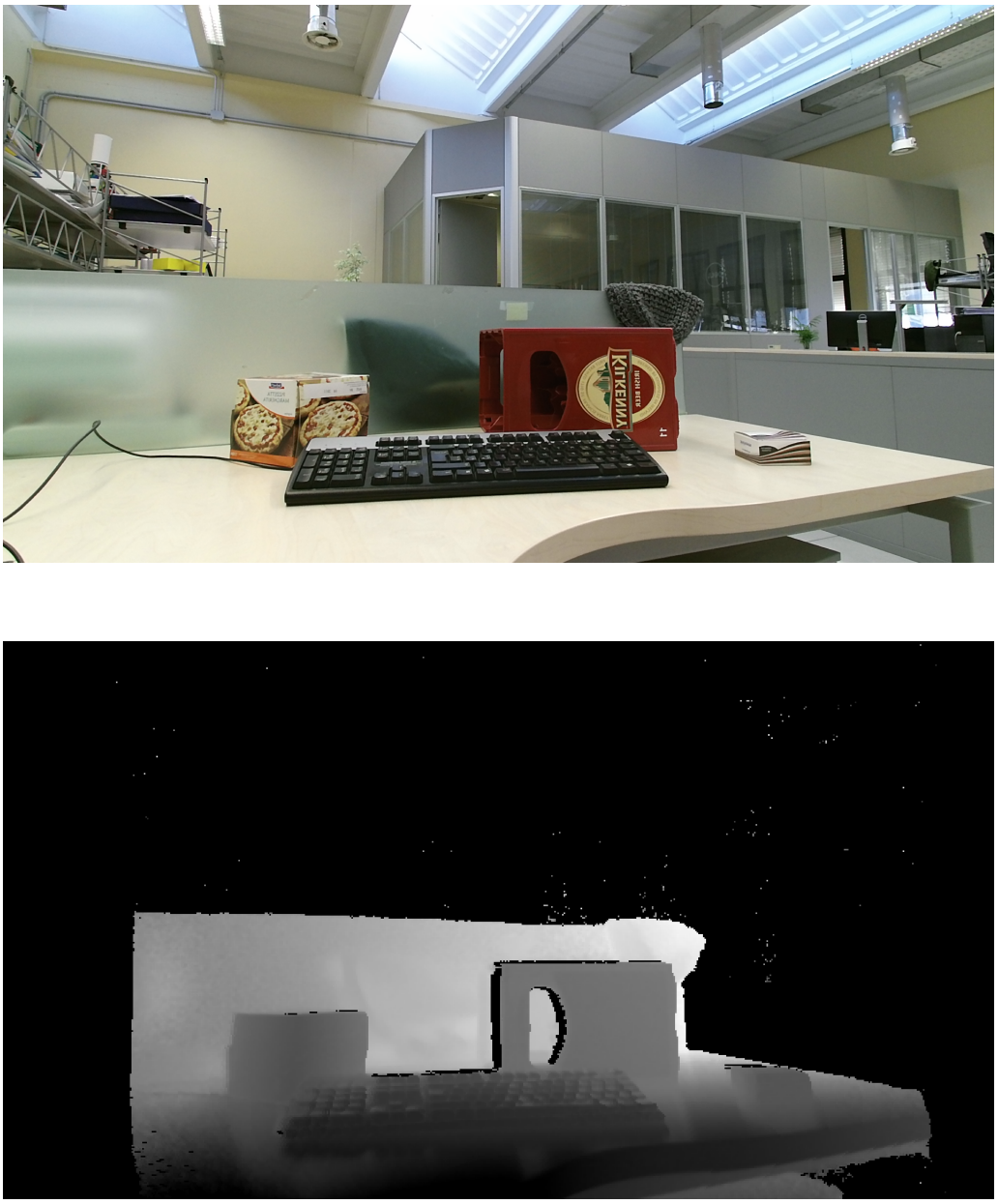


Figura 1.10: Una scena a colori e corrispettiva mappa di disparità

L'esempio in figura 1.10 mostra una scena acquisita tramite una camera a colori e un sensore di profondità ad una risoluzione di 1920x1080. La mappa delle disparità generata prende in considerazione e quantizza solo i punti rilevati tra 500mm e 1400mm.
Da notare come sia netta la separazione tra cassa e scatola col piano di sfondo e i tasti della tastiera risultino ben evidenti, nonostante siano molto vicini, di colore uniforme e soggetta ad una illuminazione differente. Di contro però notare come il berretto grigio sulla destra sia praticamente indistinguibile dal pannello sullo sfondo, e la superficie lucida del tavolo dia luogo a rumore proiettando falsi valori di fronte la tastiera (dovuta al riflesso) e renda difficile la distinzione della piccola scatola sulla destra rispetto al piano.

Capitolo 2

Camere 3D

Con *camera 3D* si fa riferimento a una camera che permetta di acquisire informazioni riguardo l'effettiva distanza tra l'obiettivo e un punto della scena.
L'obiettivo di una camera 3D è quello di generare una point cloud di un oggetto o una scena per permetterne la rappresentazione.
In un contesto di visione idustriale questo tipo di tecnologia trova un largo impiego in quanto permette di effettuare delle ispezioni efficaci riguardo l'integrità di un oggetto. Supponiamo di trovarci in uno scenario dove un oggetto prodotto su scala industriale che viene normalmente ispezionato per un controllo di qualità. Un eventuale difetto di fabbricazione interno all'oggetto come una crepa, una erosione o una scanalatura non sufficientemente profonda sono controlli molto complessi con l'uso di camere normali, ma che diventano molto più semplici con l'uso di camere 3D.
Un altro comune impiego di queste camere è per la guida di parti robotiche, come nel caso di un braccio robotico che deve agire su una specifica parte di un oggetto (che sia per muoverlo, afferrarlo o lavorarlo) in cui è fondamentale avere un esatto riferimento delle coordinate spaziali per svolgere il lavoro correttamente.

Le camere 3D possono essere suddivise in due principali categorie, attive e passive. Le camere *attive* oltre alla telecamera possiedono un ulteriore componente che proietta generalmente un fascio di luce (laser, luce strutturata, ecc.) che un sensore appartenente alla camera sia in grado di rilevare e calcolare in base ad esso le distanze.
Le camere *passive* invece, non utilizzano sensori aggiuntivi durante le riprese, ma fanno uso di una o più camere per riprendere la scena da più punti di vista differenti e generano per essa una *mappa di disparità*.

2.1 Metodi di acquisizione 3D

2.1.1 Stereo Camera

Le camere stereo prendono le loro fondamenta da uno dei sistemi più sofisticati di visione in tre dimensioni presenti in natura, ovvero l'occhio umano. Gli occhi sono separati l'uno dall'altro e consentono quindi di osservare l'ambiente circostante da due prospettive leggermente diverse. Grazie a questo il cervello è capace di percepire in maniera corretta le distanze di ciò che vede [Dep]. Si tratta quindi a conti fatti di una triangolazione di punti su due viste differenti. Ovviamente è possibile utilizzare più di due camere per ottenere una rappresentazione ancora più precisa, ma si farà riferimento solo al caso a due per una trattazione più facilmente comprensibile.

L'esempio in figura 2.1 mostra un tipico sistema di ripresa stereo. In questo caso ci tro-

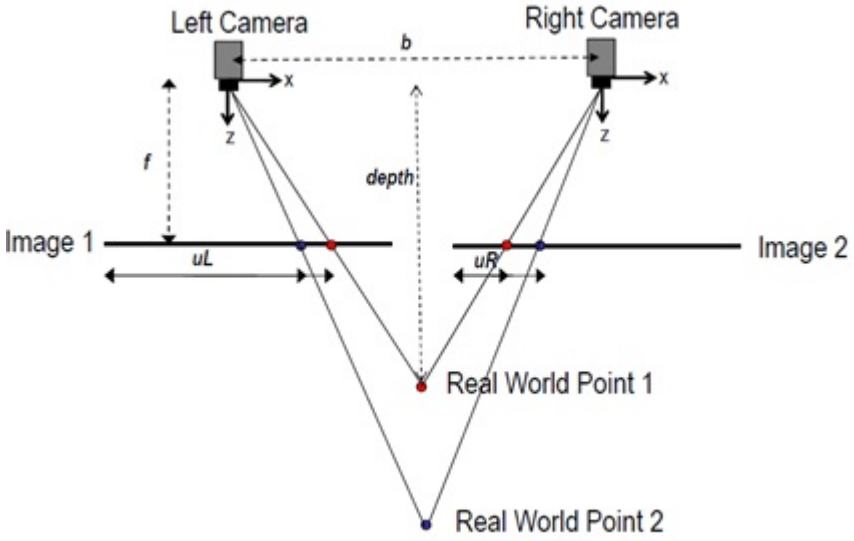


Figura 2.1: Visione Stereoscopica

viamo con due telecamere, poste sullo stesso piano orizzontale e ad una distanza fissata b. I punti 1 e 2 in figura risulteranno leggermente scostati tra l'immagine acquisita con la telecamera sinistra e quella della telecamera destra. Ricercando ogni punto della scena della telecamera sinistra in quello della telecamera destra (cosa non sempre banale) e calcolando per ognuno di essi la differenza di posizione, ci consente di ottenere una *mappa di disparità* dei punti della

Figura 2.2: Esempio di mappa di disparità - Dataset Tsukuba

scena, come mostrato in figura 2.2. Una volta ottenuta la mappa delle disparità, il calcolo delle profondità di ogni punto della scena diventa piuttosto semplice. Con riferimento al sistema in figura 2.1 e chiamando d il valore di disparità ottenuto per ogni punto, si ha che la profondità può essere calcolate come

$$depth = f\frac{b}{d}$$

La visione stereo permette quindi di risolvere problemi di prospettiva e sovrapposizione degli oggetti. Con riferimento alla figura 2.3, un qualunque punto della scena x verrà proiettato sul piano immagine sinistro nel punto x_L, rendendo quindi impossibile sia distinguerli nel caso ve ne fosse più di uno, che ovviamente distinguerne la distanza. Confrontando però con l'immagine destra diventa immediato intendere come tramite la camera destra sia possibile distinguere tutti i punti e calcolare la loro distanza tramite la mappa di disparità, a sinistra han tutti lo stesso punto di riferimento, ma a destra la loro posizione risulta via via spostata e che può essere calcolata tramite mappa di disparità.

Per ottenere delle configurazioni iniziali ideali del sistema, si dovrebbero avere quindi:

- gli assi ottici siano paralleli;
- le camere abbiano la stessa lunghezza focale;
- i piani immagine siano coplanari;
- i due frame di riferimento abbiano assi paralleli;

Sotto queste condizioni, i due frame differiscono solo per una traslazione sull'asse delle x pari a b [Toma].
Per poter inferire la distanza tramite un sistema stereo, un punto di una immagine deve essere

associato a un suo omologo nell'altra immagine, si viene a trovare quello che viene definito problema della *corrispondenza stereo*.
Il problema della corrispondenza stereo però può essere molto semplificato ricorrendo alla *geometria epiolare*.
Gli epipoli sono determinati dall'intersezione del segmento che congiunge i due centri ottici proiettati sul piano immagine 2.3. Le linee epipolari quindi, ruotano attorno all'epipolo per intersecare il punto a cui vogliam far riferimento proiettato nel piano immagine. Quindi, per cercare il punto corrispondente nell'immagine sinistra in quella destra, risulta sufficiente cerare solo all'interno dei punti della linea epipolare, che riduce di molto lo spazio di ricerca rendendolo monodimensionale.

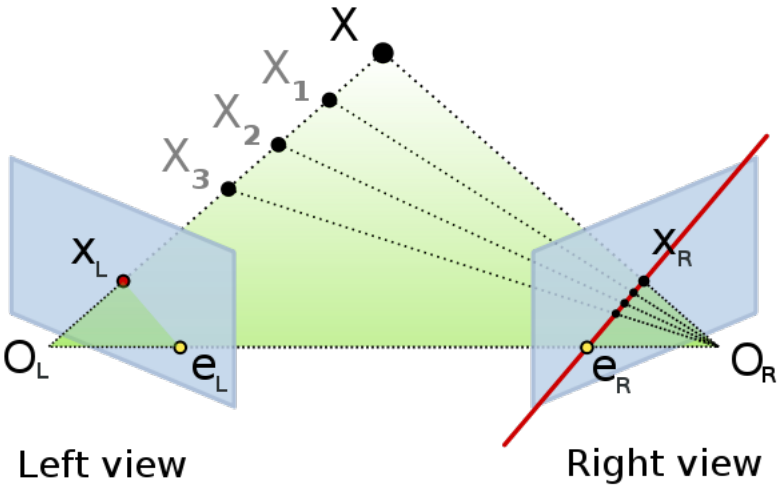


Figura 2.3: Linee epipolari - Wiki Commons

2.1.2 LIDAR

LIDAR (Light Detection and Ranging) è una tipologia di remote sensing che usa una luce laser per illuminare da ampie distanze una zona e misurarne le distanze sotto forma di range [Nat]. Con il termine remote sensing s'intende un processo di acquisizione delle informazioni riguardo un oggetto o ambiente o fenomeno senza entrare fisicamente in contatto con esso, senza quindi alterare in alcun modo l'ambiente circostante durante l'acquisizione. [Rob12].
Dato che LIDAR viene usato su e da grandi distanze (nell'ordine di kilometri), trova impiego comunemente nell'analisi di parte della superficie terrestre, come topologie di ambienti

terrestri e marine o ambiti di archeologia o geologia, dove insomma è necessario avere delle informazioni ad ampissimo raggio per poter effettuare misurazioni o valutazioni. Data l'ampia scala, per la proiezione e/o l'analisi dei fasci laser vengono usati elicotteri, satelliti e radar. Anche se trova il suo maggior impiego in grandi spazi aperti, può essere utilizzata altrettanto efficacemente in ambienti chiusi. [Nat]

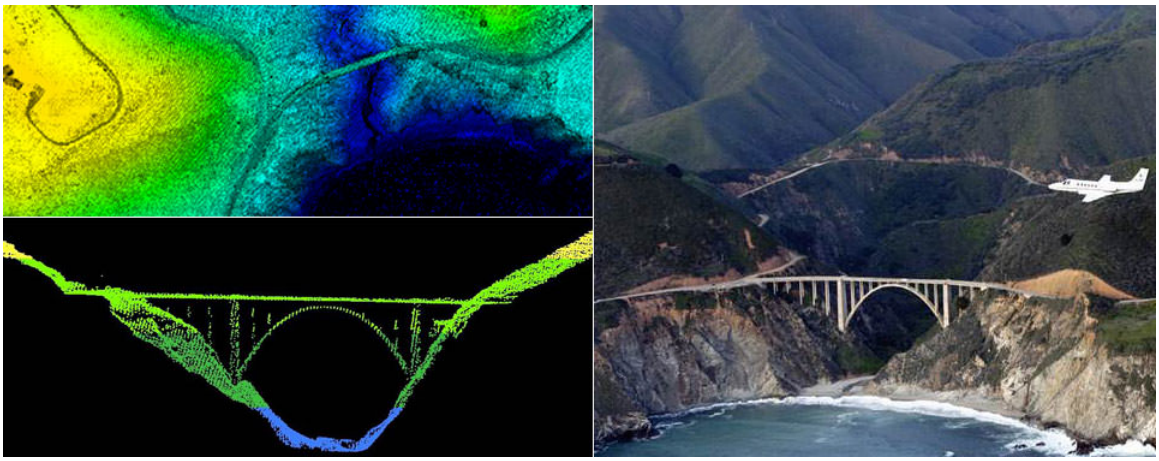


Figura 2.4: Analisi di un ponte tramite LIDAR

La point cloud che viene generata tramite questo procedimento non è molto precisa, in quanto un'acquisizione ottenuta da così ampia distanza e su larga scala non lo consente, ma non risulta troppo rilevante in base al tipo di contesto in cui viene usato, in quanto siamo più interessati in questo caso a misurare l'andamento di un terreno anziché misurarlo al millimetro. Il colore inoltre non è generalmente un'informazione che viene acquisita, dato che viene usato in un contesto di acquisizione di superfici.
Nella figura 2.4 vediamo un esempio pratico di un'acquisizione di una topologia nei dintorni di un ponte, visto dall'alto e da un lato. Questo tipo di rilevazioni vengono molto usate anche in ambito militare ad esempio durante voli di ricognizione.

2.1.3 Time of Flight

Le telecamere a tempo di volo è una classe di scansione LIDAR, dove tramite impulso luminoso viene acquisita l'intera scena stavolta punto per punto. La distanza è inferita tramite la misurazione del tempo che intercorre l'invio del segnale luminoso e il ritorno dello stesso dovuto alla rifrazione con un oggetto della scena (esempio mostrato in figura 2.5). Ogni pixel della camera qui ha ciascuno il proprio temporizzatore il cui valore di profondità viene calcolato indipendentemente per ciascun pixel [Dep] e la cui risoluzione dipende dalla dimensione

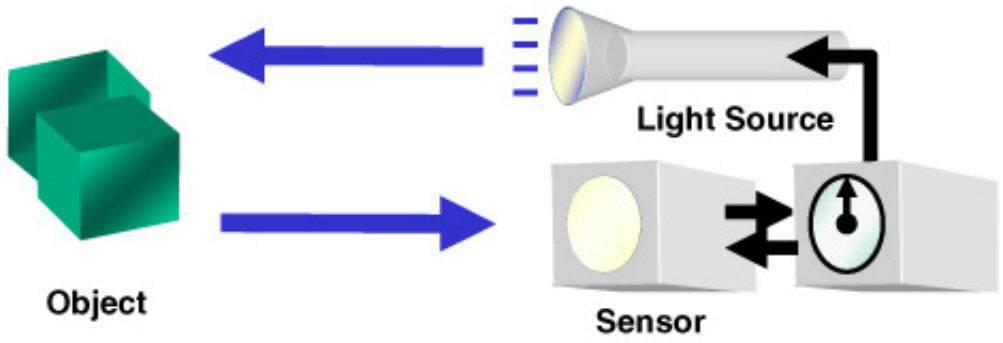


Figura 2.5: Schema di acquisizione TOF [Shi]

del chip. Data la sua struttura questo tipo di camera è in grado di acquisire in real time e presenta una buon rapporto qualità/prezzo, risulta però inadatta agli ambienti esterni a causa delle interferenze che possono essere causate dalla luce solare.
I principali limiti di questa tecnologia sono principalmente dovuti ad una alterazione della rifrazione della luce, questo fenomeno è conosciuto come *scattering*.
Poniamoci nel caso di inquadrare una zona vicina a due pareti, una orizzontale e uan verticale. Il sensore suppone che il tempo che la luce riflessa impieghi a tornare sia lo stesso che abbia impiegato ad arrivare all'oggetto, ma in un caso come questo non è così in quanto la luce a infrarosso subisce una riflessione multipla prima di tornare al sensore, che genera una sovrastima della distanza per quei punti.
Osservando in figura 2.7 si può notare il fenomeno di scattering sul piano d'appoggio attorno

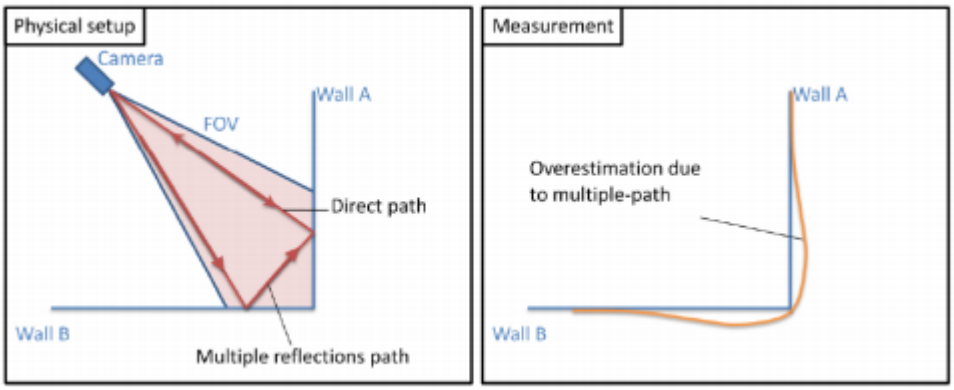


Figura 2.6: Fenomeno di Scattering su TOF

alla tastiera. In più materiali che tendono ad assorbire la maggior parte del colore (oggetti neri)

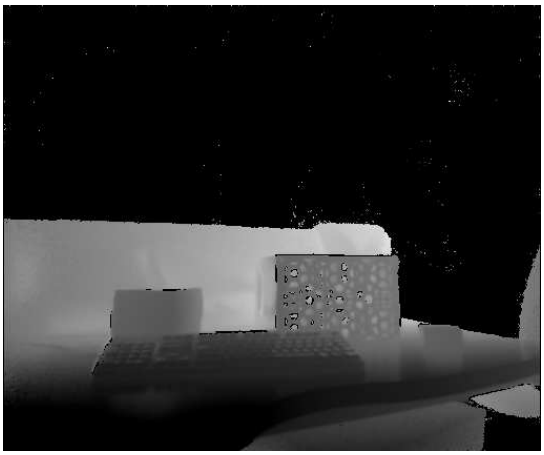


Figura 2.7: Depth Image tramite TOF

attenuano ulteriormente il segnale falsando la misurazione.
Un altro problema dovuto alla rifrazione si ha nella stima dei bordi di un oggetto, in quanto se si ci trova ad inquadrare bordi che presentino una curvatura elevata, buona parte del segnale viene disperso e la parte riflessa è notelmente inferiore, causando in questo caso una sovrastima della distanza.

2.1.4 Structured Light

I sistemi a proiezione di luce strutturata lavorano proiettando un pattern di luce infrarossa noto all'interno di una scena 3D e conseguentemente catturandone l'immagine all'interno di un sensore apposito della camera.
Il pattern di luce proiettato è generalmente composto da strisce verticali che verrà distorto dagli oggetti della scena in base alla loro posizione, pertanto il pattern osservato dalla telecamera risulterà differente da quello originale proiettato sulla scena. Determinando lo scostamento del pattern proiettato per ogni pixel la profondità di ogni punto inquadrato dalla camera può essere determinata [Dep].
Il principio per il calcolo della profondità che sta dietro ai sistemi a luce strutturata è molto simile a quelli delle camere stereo, anche qui viene calcolata una mappa delle disparità basata invece sulla distanza rispetto a dove ci aspetteremmo il pattern in condizioni standard, usando però una singola acquisizione della scena.
Questo tipo di tecnologie è generalmente molto precisa e si attesta sull'ordine dei millimetri e micrometri in termini di accuratezza.
Le camere che fanno utilizzo di un pattern di luce strutturato sono generalmente composte da:

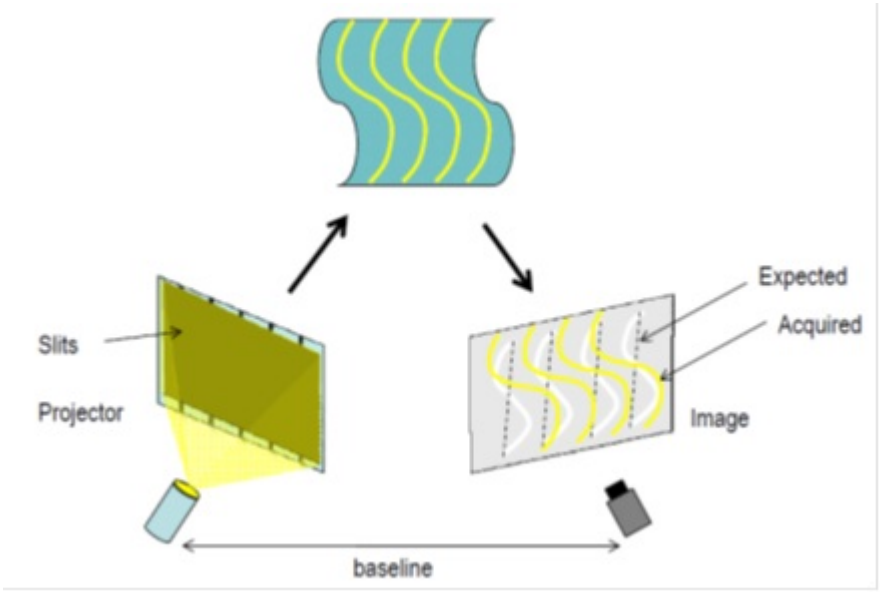


Figura 2.8: Luce strutturata

- un proiettore a infrarossi;
- una camera a infrarossi;
- una camera a colori;

questo consente di rilevare in un determinato istante una rappresentazione accurata sia del colore che dell'informazione di profondità (rappresentati in dati del tipo *RGB-D*).
Di contro l'utilizzo dei pattern rende molto difficile l'utilizzo di questi sistemi in ambienti esterni e lo rende suscettibile alla presenza di altre fonti luminose, comprese eventuali altre camere che proiettano il proprio pattern in una zona comune (si creerebbero intereferenze a vicenda). Inoltre, le proprietà dei materiali quali la lucidità e semitrasparenza possono portare il sensore ad ampi errori di rilevazione o a non rilevare affatto le profondità delle parti iin questione.

2.1.5 Line Scan

Le camere di tipo line scan utilizzano un approccio diverso da tutte le altre, anziché acquisire un'area si limitano a processare una linea di pixel per volta.
Gli oggetti scorrono attraverso un nastro sotto un fascio di luce laser che illumina una striscia

alla volta. Osservando la luce riflessa utilizzando una camera (generalmente ad alta risoluzione e conoscendo posizione e orientazione sia della camera che ella fonte di luce, è possibile determinare le distanze tre i punti riflessi e la fonte di luce o la telecamera [Dep]. Una volta che l'oggetto è stato interamente scandito si ottiene una sequenza di righe di profondità che possono essere assemblate per ottenere un'immagine di disparità.
Dato che è intrinsecamente previsto che o gli oggetti o la fonte di luce si muovano, l'acquisizione è strettamente legata alla velocità con cui scorrono gli oggetti, una variazione sulla velocità determinerebbe una elongazione o restringimento dell'immagine acquisita.

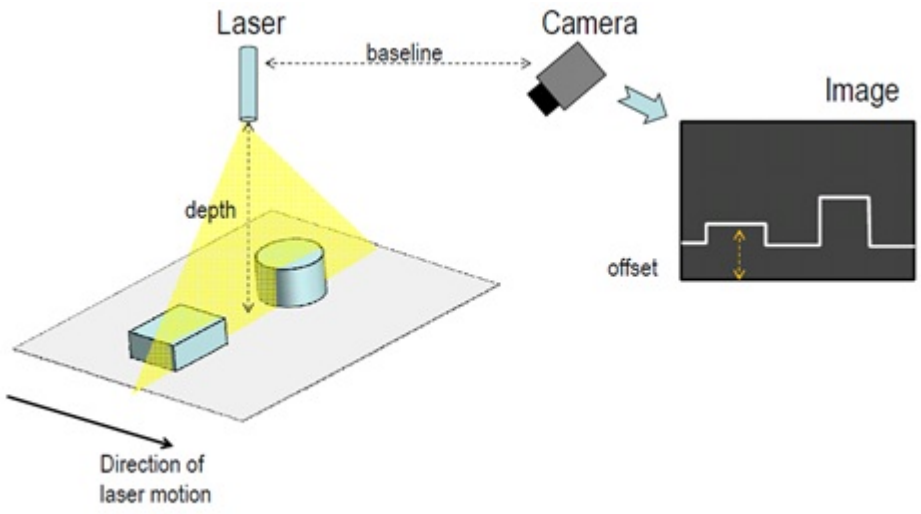


Figura 2.9: Laser Scanning

2.2 Modelli di Telecamere 3D

2.2.1 Kinect

Figura 2.10: Microsoft Kinect

La camera Kinect è la prima camera che verrà utilizzata come strumento di lavoro per questa tesi. Sviluppata da Microsoft inizialmente come supporto ludico per la Xbox 360, si è successivamente rivelato un ottimo strumento per gli sviluppatori di visione che si sono ritrovati un apparecchio dalle buone prestazioni e contemporaneamente dal bassissimo costo rispetto a camere specializzate o industriali (si parla di un paragone nell'ordine tra un centinaio di euro e le migliaia o decine di migliaia). Con le sue oltre 20 mlioni di unità vendute, spetta quindi la definizione di prima camera 3D per il consumo di massa della storia.
Il successo ottenuto ha spinto la Microsoft a rilasciare in un secondo tempo una versione del Kinect specifica per Windows (nonostante sia utilizzabile tramite SDK anche la versione per Xbox) con una modalità aggiuntiva chiamata *near mode* che abbassa la profondità minima di detection fino a 40 centimetri.

Vediamo ora quali sono el componenti della telecamera nello specifico:

Sensore a colori: permette di acquisire una immagine RGB conservata in tre canali, con una risoluzione massima di 1280x960 e una acquisizione fino a 30 frame per secondo.

Emettitore a infrarossi: proietta un pattern di luce strutturata consentendo al sensore di profondità di inferire la distanza.

Sensore di profondità: permette di inferire le distanza ad una risoluzione fino a 640x480 anch'essa a 30 frame per secondo a partire da 80cm fino a 8 metri, anche se effettivamente utilizzata fino a 4 dove risulta affidabile consentendo anche il tracking delle persone.

Tilt: Permette di variare l'angolo di ripresa della camera con un angolo di $\pm 27°$ ed è modificabile via software.

Array di microfoni: Quattro microfoni distribuiti sulla lunghezza del Kinect consentono oltre la registrazione dell'audio la triangolazione delle posizioni delle voci e l'esclusione degli altri rumori nella stanza e la soppressione dell'eco.

Accelerometro: un accelerometro a tre assi, consente di determinare la corrente orientazione del Kinect e viene usata durante le ricostruzioni 3D del Kinect Studio.

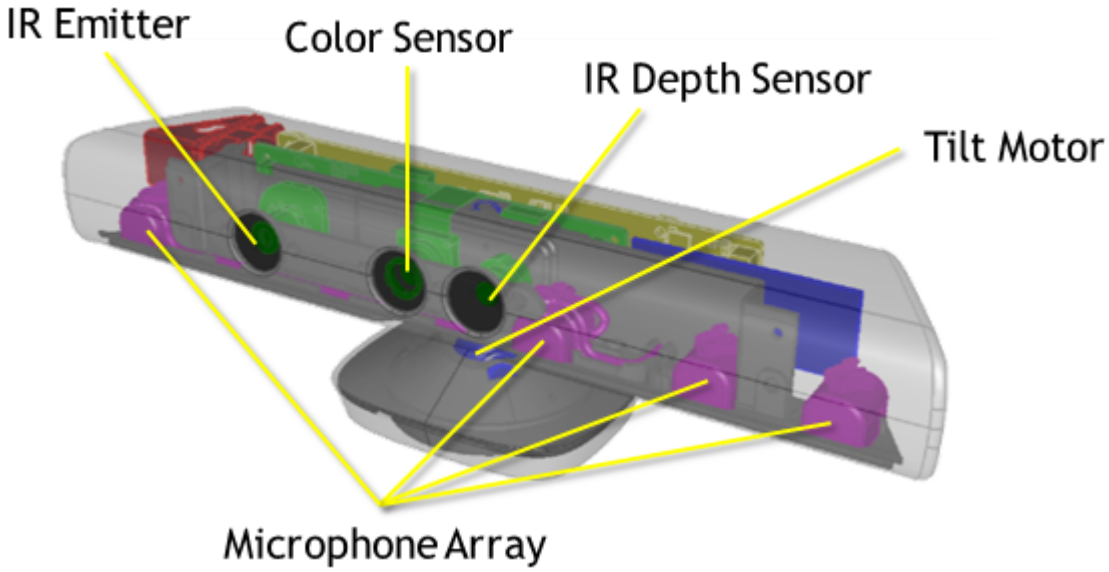


Figura 2.11: Sensori del Kinect - MSDN Microsoft

2.2.1.1 Il sensore a colori

La camera a colori permette due risoluzioni principali, 640x480 e 1280x960. L'uso di uno o dell'altro formato impatta notevolmente sull'acquisizione, in quanto con la prima si arrivano ad acquirsire 30 frame per secondo mentre con quella ad alta risoluzione il massimo a cui si

arriva è 15.
I formati di output disponibili sono 3 e usabili liberamente tramite l'uso di appositi flag via codice. Questi formati sono:

- RGB, bitmap a 32 bit standard X8R8G8B8;
- YUV, bitmap gamma-corrette in forma UYVY;
- Bayer a 32 anch'esso su standard X8R8G8B8;

Oltre il formato di output, è prevista la possibilità di modifica dei parametri della camera via software, questi parametri sono:

- esposizione;
- contrasto;
- saturazione;
- guadagno;
- gamma;
- saturazione;
- bilanciamento del bianco;

2.2.1.2 Il sensore di profondità

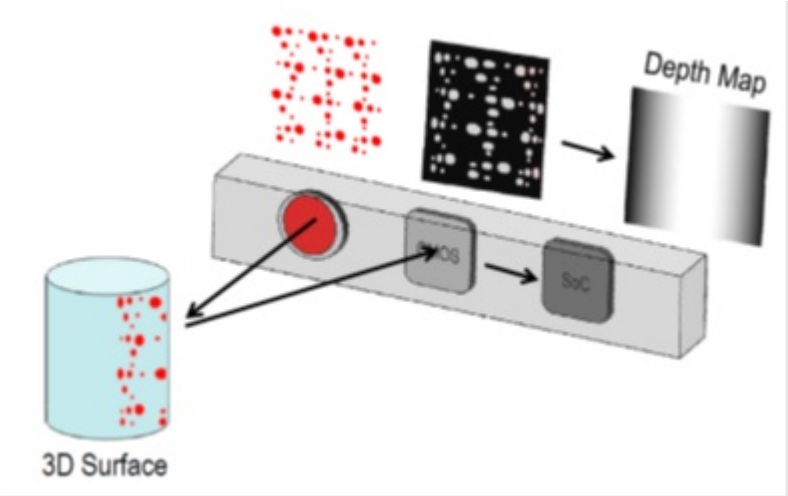


Figura 2.12: Processo di acquisizione del sensore di profondità

Il sensore di profondità del Kinect, sviluppato dalla PrimeSense, funziona secondo il principio del pattern di luce strutturata (spiegato al paragrafo 2.1.4).
Come si può notare dalla figura 2.13 il pattern si contraddistingue in 9 blocchi principali in cui per ognuno vi è illuminato il punto centrale. Il pattern è generato da una serie di reticoli di diffrazione, con una attenzione particolare per la diminuzione dell'effetto di propagazione di ordine zero del punto luminoso centrale [RW].

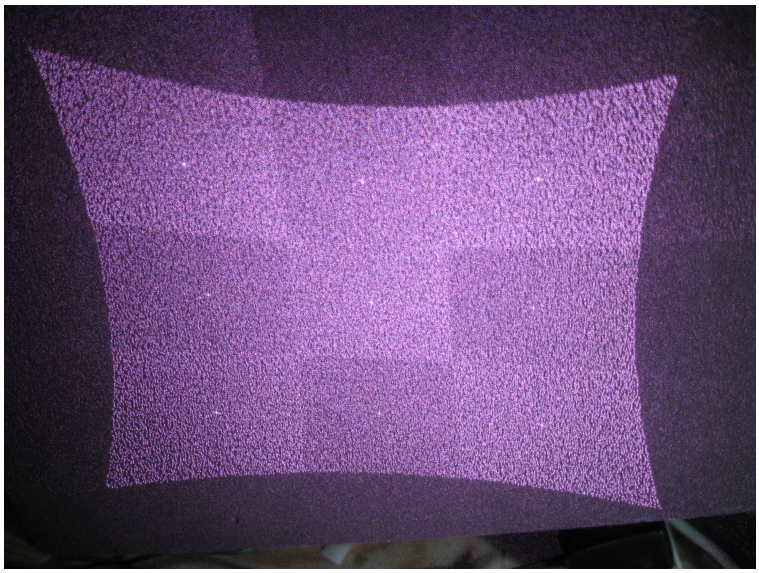


Figura 2.13: Pattern di luce strutturata del Kinect

Il flusso di acquisizione è mostrato in figura 2.12. Per ogni pixel appartenente all'immagine a infrarossi è calcolata una finestra di correlazione (9x9 o 9x7) che viene usata per comparare il pattern locale proiettato e i suoi 64 pixel vicini su una finestra orizzontale. Il miglior matching all'interno di questa finestra viene considerato come la disparità da considerare in termini di pixel. Il sensore quindi calcola un'ulteriore interpolazione del best match per ottenere una precisione di subpixel di $\frac{1}{8}$ di pixel. Ottenuta la disparità, la profondità effettiva viene stimata pixel per pixel tramite triangolazione [RW].

Tutte le valutazioni sulla profondità vengono effettuate in tempo reale, garantendo una velocità di 30 frame al secondo con una risoluzione fino a 640x480. Questo consente al Kinect per ogni momento di ottenere due immagini, una a colori e una di profondità, con la stessa risoluzione e lo stesso frame rate.
Le distanze di detection (figura 2.14) vanno normalmente dagli 80 centimetri ai 4 metri. Il sensore comunque è tale da poter essere forzato per valutare distanze fino a 8 metri, ma oltre i 4 non sono garantite le funzioni di body tracking e inoltre, dato che le prestazioni decadono all'aumentare della distanza (maggiori presenze di rumore etc.) per misurazioni di precisione è meglio sempre attenersi al primo range.
Come già accennato la versione di Kinect per Windows possiede una funzionalità aggiuntiva chiamata *near mode* che permette di abbassare il range da 40 centimetri a 3 metri.

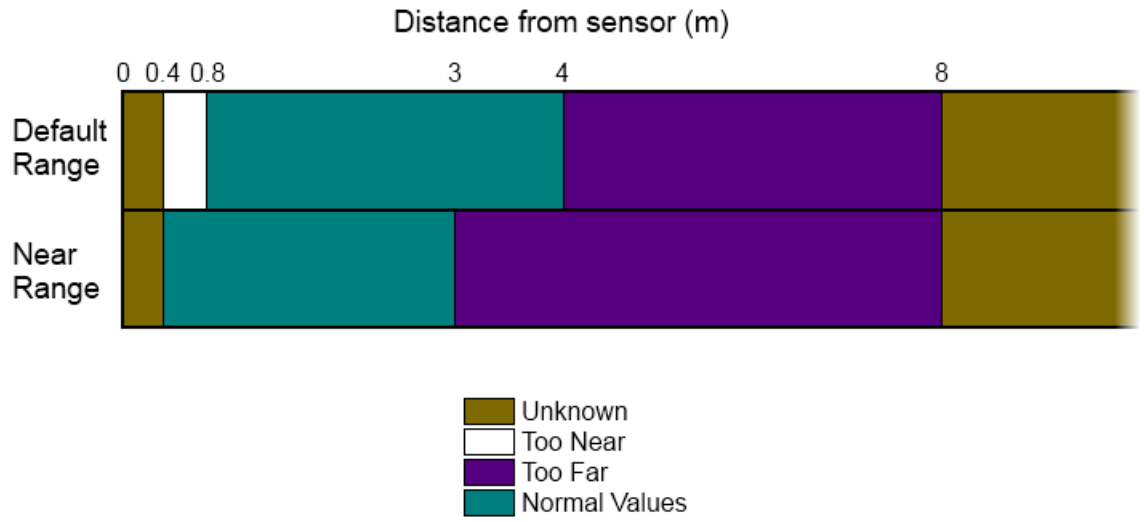


Figura 2.14: Distanze risolvibili - MSDN Microsoft

2.2.1.3 Architettura

I flussi di dati provenienti dal sensore sono 4:

- Image Stream;
- Depth Stream;
- Infrared Stream;
- Audio Stream;

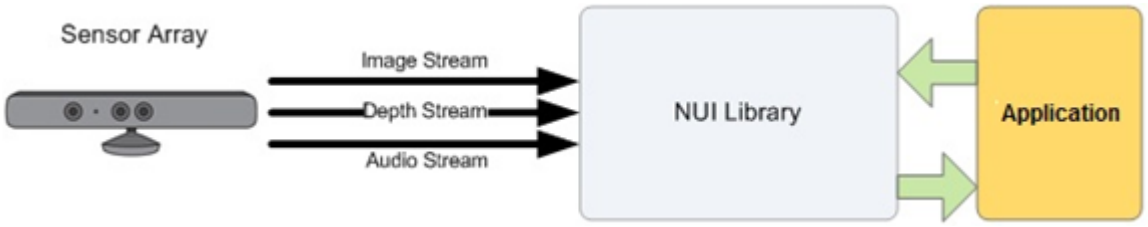


Figura 2.15: Stream e NUI [Mic]

che corrispondo ad uno per ogni sensore. Dato che vengono usati sensori diversi, più stream possono essere avviati in parallelo per una sincronizzazione tra le informazioni. Solo lo stream

infrarossi non usa un proprio canale ma usa la camera RGB, pertanto quello e lo stream a colori sono gli unici che non posso essere usati nello stesso momento. Questi stream sono catturati tramite la libreria *NUI* (Natural User Interface) con cui è possibile interfacciarsi con le applicazioni (in figura 2.15).

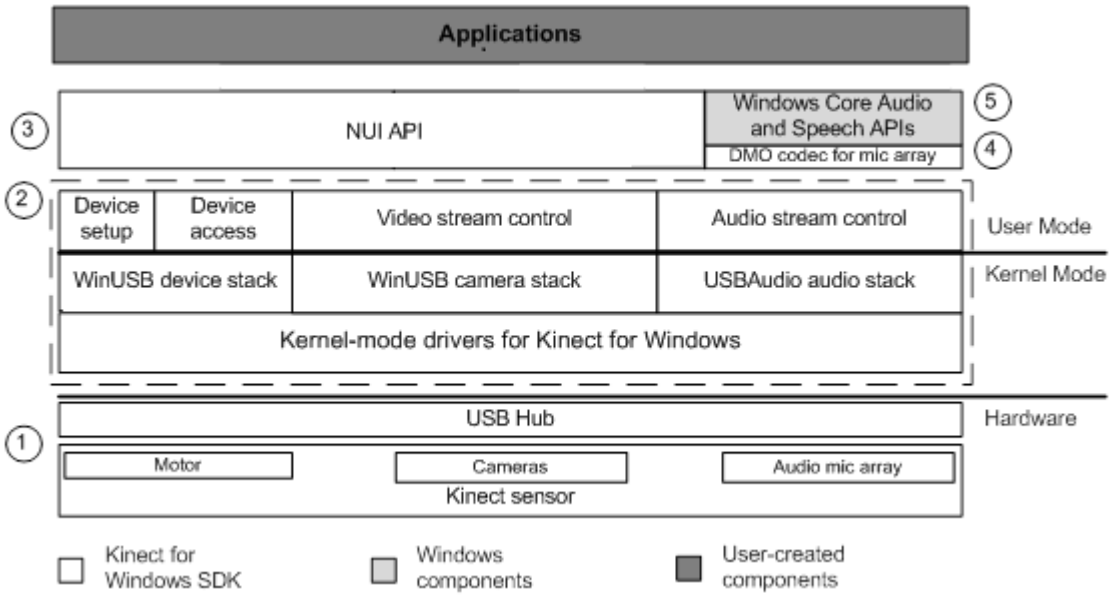


Figura 2.16: Struttura interna [Mic]

Vediamo in figura 2.16 l'architettura che va dal sensore fino al lato applicazione [Mic]:

1. le componenti hardware, inclusi i sensori e gli hub usb attraverso i quali i sensori vengono connessi al computer.

2. lo strato driver, installati come parte delle SDK per la gestione dei sensori e corrispettivi stream.

3. le componenti audo e video del NUI, incluse le feature aggiuntive della libreria riguardanti il tracking.

4. le DirectX Media Object per la gestione dell'array di microfoni e la triangolazione del suono.

5. le API per la comunicazione col sistema operativo.

2.2.1.4 Software

Un breve elenco delle librerie esistenti per lo sviluppo su Kinect:

Microsoft Kinect SDK: La release ufficiale della Microsoft, fortina tramite package di installazione sia a 32 che a 64 bit, utilizzabili per la distribuzione commerciale. Fornisce ampie API e sample delle principali funzionalità dei vari stream e alcune applicazioni più complesse come face tracking e background removing.

OpenKinect: Alternativa free e senza possibilità di redistribzione commerciale, risulta comunque di meno immediato utilizzo rispetto alle release ufficiali.

OpenNI: Sono risorse direttamente rilasciate dalla PrimeSense, che ha sviluppato il sensore della camera, aprendo una comunità a supporto e di libero sviluppo.

2.2.2 Kinect 2

Figura 2.17: Microsoft Kinect 2

Seconda camera presa in esame per questa tesi, è stata lanciata a fine 2013 da Microsoft per la versione console e nell'estate 2014 nella sua controparte per Windows. Data la giovane età della versione PC, le SDK sono ancora in corso d'opera e non è ancora possibile sfruttare

al meglio tutte le funzionalità della camera come ad esempio manca ancora la possibilità di settare i parametri della camera a colori.
Il Kinect 2 va visto come una evoluzione del sistema Kinect, che mostra le stesse funzionalità ma con hardware potenziate e librerie riadattate più da un punto di vista logico. L'unica differenza sostanziale dal punto di vista tecnologico è l'acquisizione delle informazioni di profondità che non sfrutta più un riconoscimento a luce strutturata ma un sistema a tempo di volo e la resa del sensore a infrarossi attivo, quindi capace di elaborare un suo stream dedicato (nella versione 1, come detto al paragrato 2.2.1.3 utilizzava il color stream e pertanto i due non potevano essere utilizzati insieme).

Vediamo ora cosa cambia per sensore rispetto alla prima versione:

Sensore a colori: la risoluzione passa a 1920x1080 garantendo per questa risoluzione un frame rate di 30 fps. il field of view aumenta passando da $57°$ orizzontali e $43°$ verticali a $70°$ orizzontali e $60°$ verticali.

Sensore di profondità: passa a una tecnologia time of flight con un sensore a infrarossi attivo e una risoluzione di 512x424, comunque rimappabile a 1920x1080 automaticamente via SDK.

Tilt: rimosso.

Array di microfoni: l'array è potenziato con i quattro microfoni che passano da 16 kHz a 48kHz.

Accelerometro: non presente.

Sul lato software i potenziamenti sono soprattutto sul tracking, permettendo adesso il finger tracking, il tracking facciale e riconoscimento espressivo in alta definizione e migliorato, una maggiore segmentazione degli arti nel body tracking e tracking e riconoscimento facciale in contemporanea.

2.2.2.1 Il sensore di profondità

Il sensore di profondità rappresenta il cambiamento più radicale all'interno della camera, passando da un sistema a luce strutturata a un modello a tempo di volo, di cui sono state illustrate le caratteristiche generali nel paragrafo 2.1.3.

I vantaggi principali del passaggio a una TOF sono [SO14]:

- un depth sample per pixel: la risoluzione X-Y è determinata dalle dimensioni del chip;

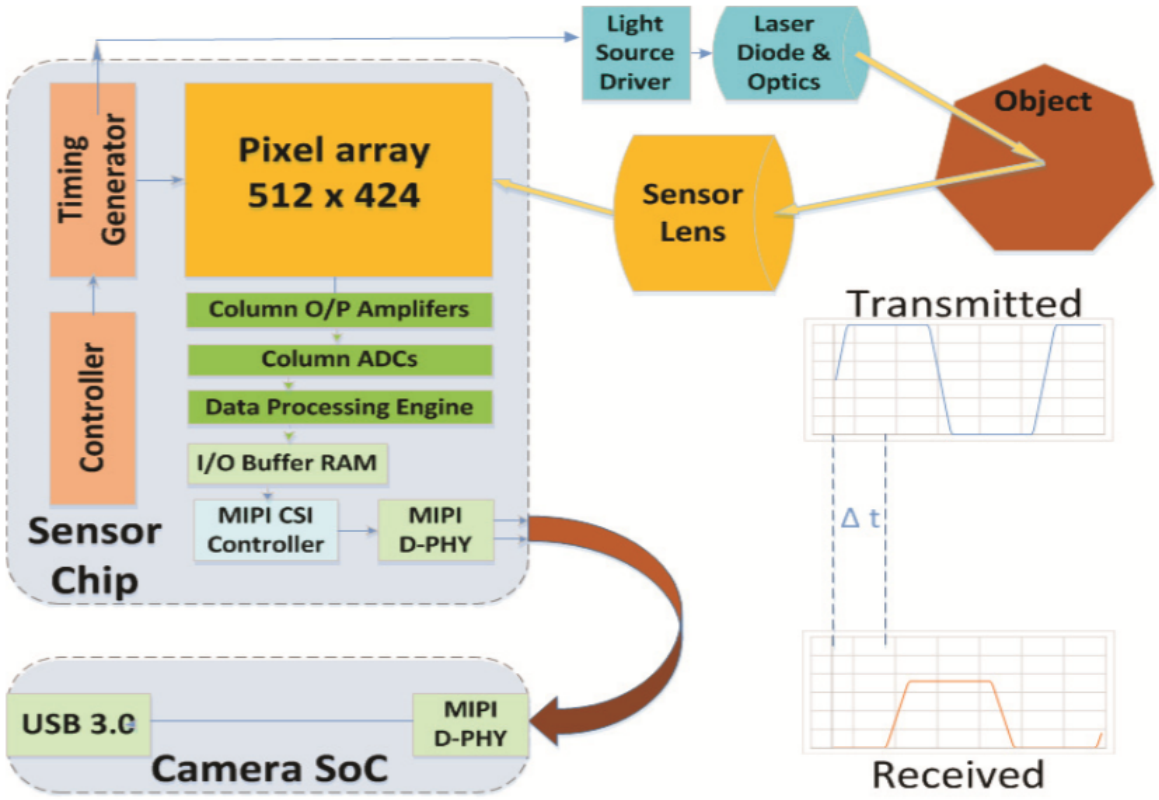


Figura 2.18: Three-dimensional Image Sensor System [SO14]

- la risoluzione della depth è una funzione data dal rapporto tra il segnale di rumore e la frequenza di modulazione;
- frequenza più alta: il rapporto fase/distanza scala direttamente con la modulazione di frequenza risultando meglio raffinato;
- il sensore consente in output tre immagini possibili riferenti agli stessi dati pixel (colore, profondità, infrarosso);
- il sensore attivo lo rende indipendente da fattori di illuminazione ambientali esterni;
- risulta più robusto rispetto al colore in confronto col Kinect 1.

Il Kinect 2 possiede a differenza dell'uno una sola modalità di range della distanza, che va dai 50 centimetri ai 4.5 metri, come mostrato nell'immagine in figura 2.19.

Le SDK ufficiali e non non gestiscono attualmente altri range di distanza, ma [SO14] mostra come sia possibile arrivare lavorando sulle frequenze del sensore a una distanza di 16 metri, come mostrato in figura 2.20. Le differenze di prestazione sulla mappa di profondità è eviden-

Figura 2.19: Kinect 2 Depth Image [SO14]

te soprattutto in termine di attenuazione del rumore in prossimità dei bordi come è possibile da vedere chiaramente in figura 2.21 dove vengono mostrate due depth image con entrambi i Kinect riferiti alla stessa scena. Sempre in riferimento all'immagine si può notare subito la differenza di accuratezza nella rappresentazione del telefono, dove nel Kinect 2 è possibile vedere i tasti mentre nel primo no.

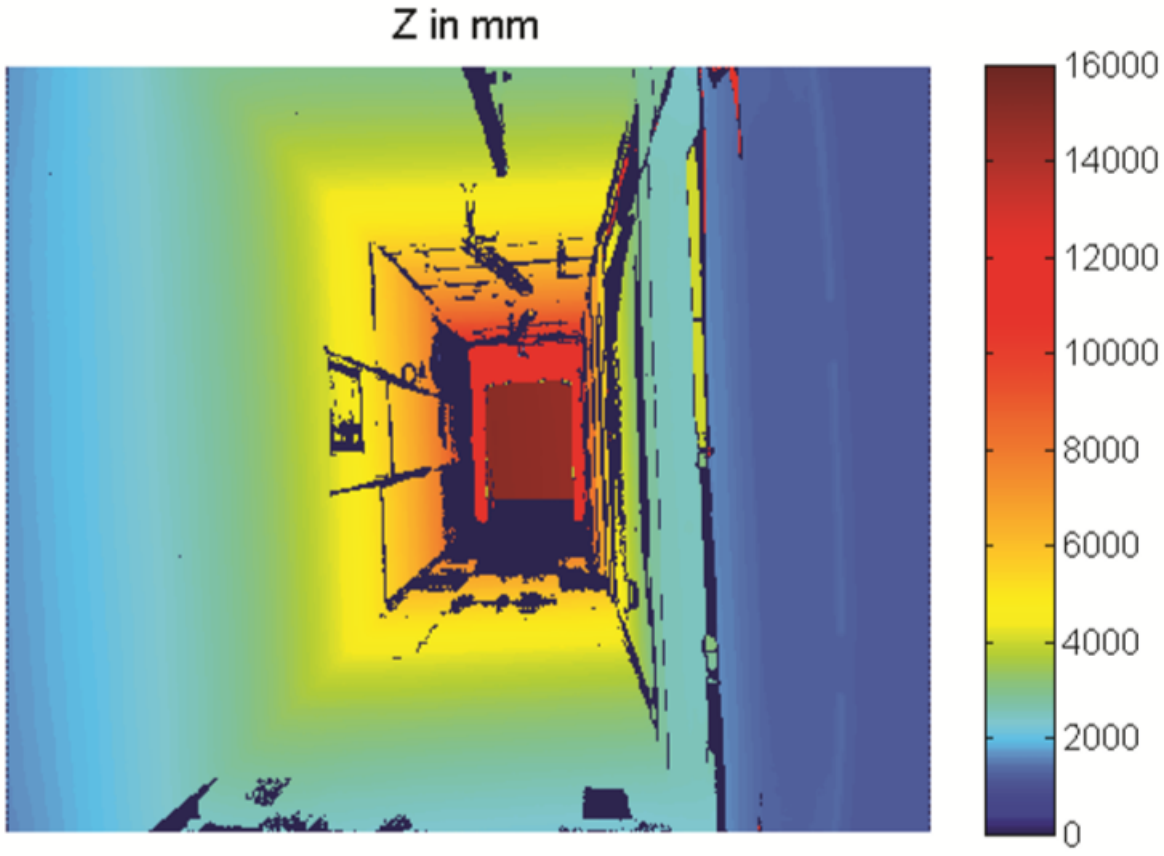


Figura 2.20: Kinect 2 Full Depth Range [SO14]

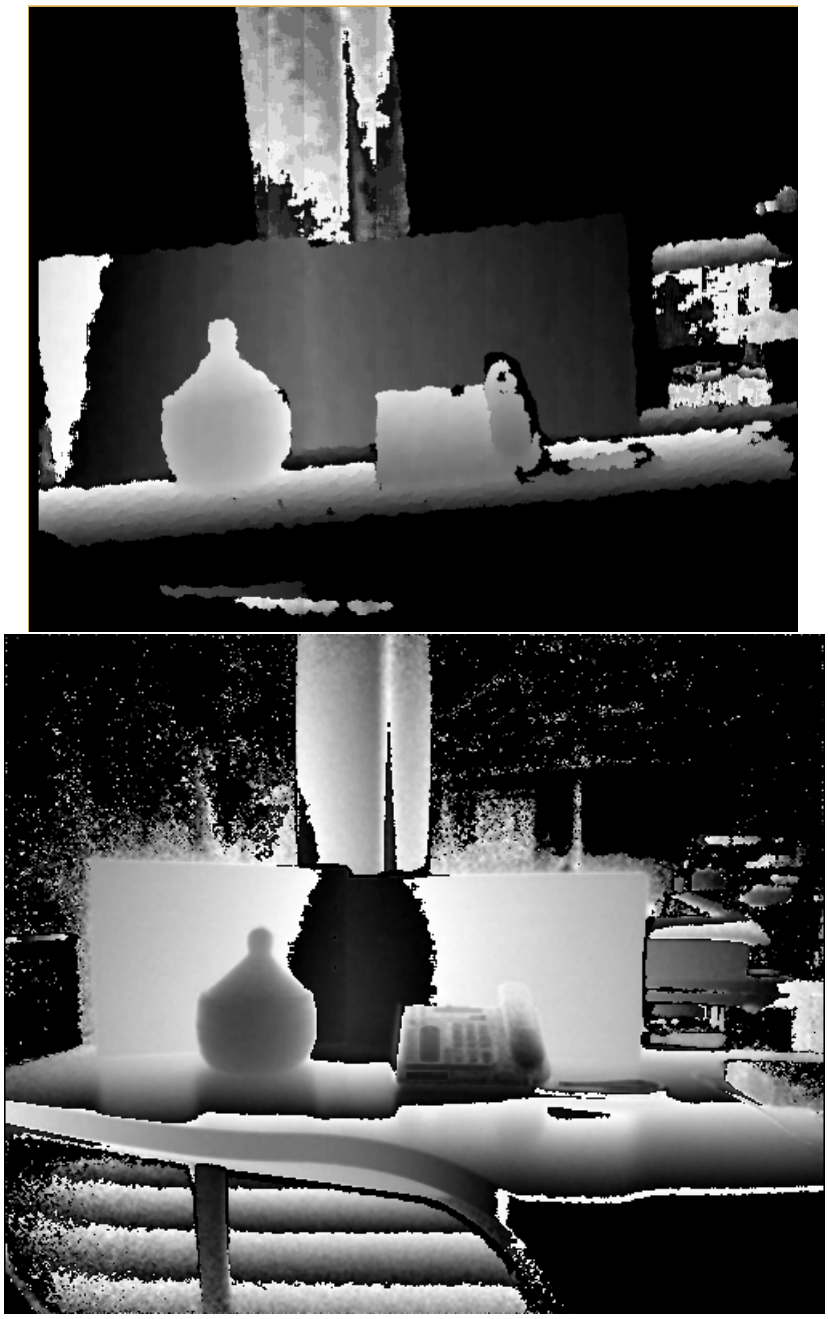


Figura 2.21: Depth Image di Kinect 1 e 2

Capitolo 3

Sviluppo di un sistema di dimensionamento volumetrico

Il sistema che verrà presentato in questo capitolo si inserisce in un contesto di *Machine Vision*. La Machine Vision è quella branca della Computer Vision che si occupa dell'analisi di immagini per ispezioni automatiche in ambito industriale, principalmente per ambiti di controllo qualità, controllo del processo produttivo e guida per robot industriali. Va ad inserirsi quindi all'interno di quei contesti automatizzati dove è indispensabile eseguire controlli ad alta precisione ed in tempo reale tipici di una produzione in serie o catena di montaggio.

3.1 Lo scopo

La finalità di questo sistema è quello di strumento di controllo e misura all'interno di un sistema di inscatolamento, impacchettamento o imballaggio. Si procederà quindi per calcolare la *oriented bounding box 3D* di oggetti che scorrono sopra un nastro trasportatore.
Questo apre possibili scenari applicativi:

1. una stima delle misure e quindi del volume di pacchi, utilizzabile ad esempio in un contesto di ditte di spedizioni come corrieri o uffici postali, in cui dimensioni diverse danno luogo a spese di spedizione diverse.

2. un sistema di inscatolamento per oggetti di dimensione variabile e non nota a priori.

3. un sistema di guida robot basato sulla *collision detection*, ovvero quando due bounding box non si intersecano, allora gli oggetti ivi contenuti non possono collidere a loro volta.

Inoltre, l'utilizzo di questo sistema permetterà di testare le due telecamere Kinect in un ambito prettamente industriale, anche se come già esposto nel 2, non siano state realizzate a questo scopo. Possono essere utilizzate efficacemente però per eseguire test su algoritmi e sistemi software che poi saranno utilizzati da camere ben più potenti nei sistemi finali, portando un notevole risparmio sui costi di ricerca e sviluppo.

3.2 Il contesto di utilizzo

Questo sistema è stato pensato e progettato all'interno di un contesto industriale. Questo porta a fare delle considerazioni sul contesto di utilizzo che impattano sul sistema stesso e la sua realizzazione.
L'applicazione è pensata per essere usata all'interno di un nastro trasportatore dove scorrono dei pacchi di dimensione non nota a priori, con una telecamera posta in alto ad inquadrare in tempo reale la scena, e dove la velocità del conveyor è anch'essa non nota a priori. Per

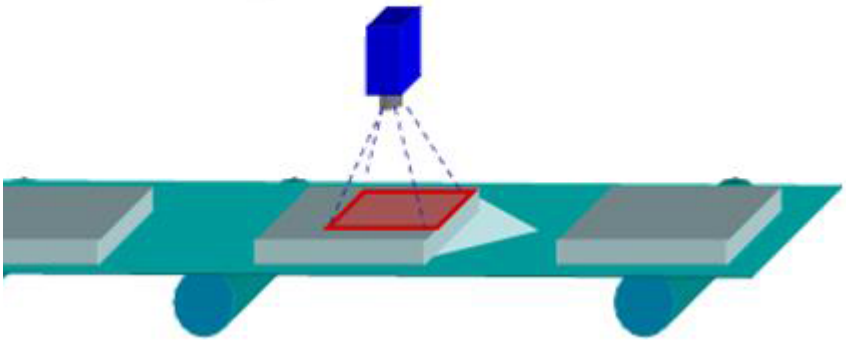


Figura 3.1: Acquisizione da conveyor

l'acquisizione ed il raffinamento dei dati si utilizzerà il software di machine vision della Datalogic *Impact*. Questo fornisce strumenti di ispezione industriale che facilitano le analisi e che permettono tramite tool ad hoc di ispezione di processare le informazioni necessarie all'analisi dall'immagine.

3.3 Acquisizione delle immagini

Il primo passo consiste nell'integrazione delle camere Kinect all'interno del framework di Impact, in modo tale da poter acquisire le immagini grezze dalle camere e processarle direttamente tramite quest'ultimo. In questo modi si potranno ottenere dei dati opportunamente processati e utilizzare gli strumenti di ispezione del software per raffinare i dati e procedere al calcolo bounding box 3D orientata in maniera precisa.
Il flusso è mostrato in figura 3.2. Si è proceduto ad integrare le SDK dei Kinect all'interno di Impact, in modo tale da poterle richiamare permettendone l'acquisizione su Impact stesso. In questo modo, è possibile richiamare il color stream e il depth stream (paragrafo 2.2.1.3) e caricare nei buffer interni di Impact dati ottenuti.

Prima di poter lavorare con le immagini, esse vanno preprocessate in modo tale che, per ogni pixel dell'immagine a colori e quella di profondità si abbia un'informazione appartenente allo stesso punto della scena. Vanno quindi fatte due considerazioni, una inerente la risoluzione delle due camere e una riguardante la loro posizione fisica.
Riguardo la grandezza delle immagini, come visto nel paragrafo 2.2.1 e nel paragrafo 2.2.2, i due Kinect lavorano a differenti risoluzioni e posseggono differenti risoluzioni anche tra i diversi stream. Nel caso del Kinect 1 entrambi i sensori possono dare in output immagini a

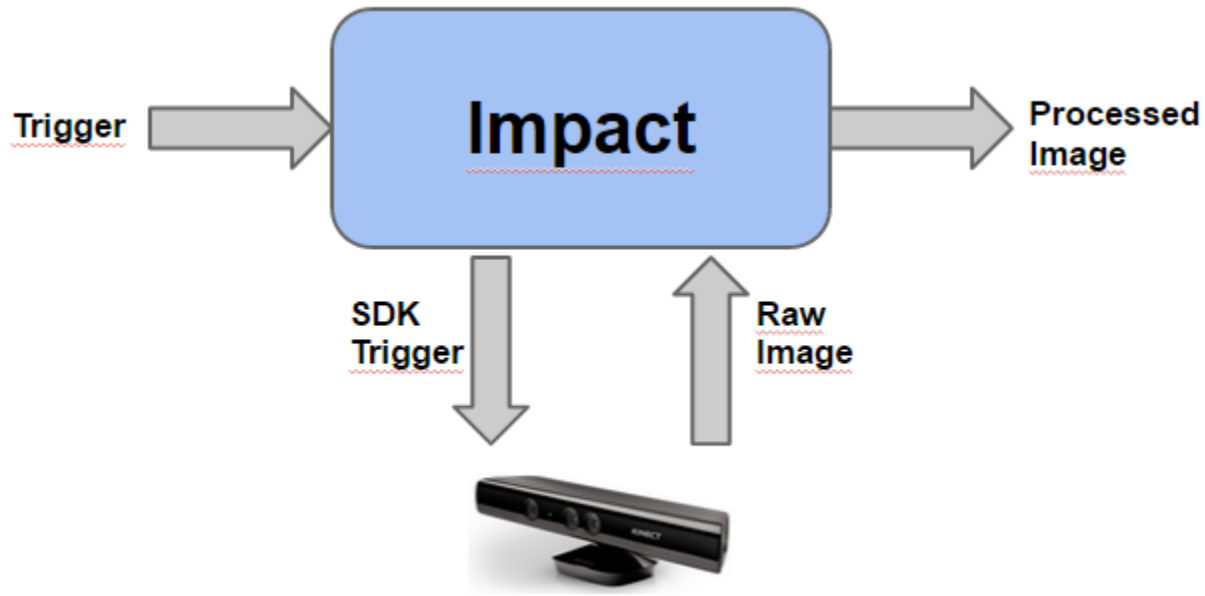


Figura 3.2: Flusso di integrazione tra Kinect e Impact

una risoluzione di 640x480, limitando quindi la risoluzione dello stream a colori non è necessario rimappare le due immagini. Per il Kinect 2 invece, dato che attualmente ritorna immagini esclusivamente a 1920x1080 per l'immagine a colori e 512x424 una forma di mapping risulta necessaria. Le Kinect SDK 2.0 forniscono delle funzioni di mapping tra i frame ottenuti dai due sensori, ed è attualmente più conveniente usare quelli dato che allo stato attuale non sono stati forniti dalla Microsoft i parametri degli intrinseci delle telecamere.
Il mapping della della depth con l'immagine a colori non è una semplice scala, perché va anche considerato che le due camere, in entrambi i Kinect, sono disposti a una distanza nota su un piano orizzontale. L'effetto è quindi lo stesso di una visione stereo in cui in un momento si hanno due viste leggermente diverse.
A mapping completato, sovrapponento l'immagine di profondità a quella a colori l'effetto è simile a quello mostrato in figura 3.3. L'effetto di distorsione è dovuto alla differenza delle distanze focali dei due sensori.
Le SDK del Kinect permettono di ritornare i valori per ogni pixel anche nello spazio della

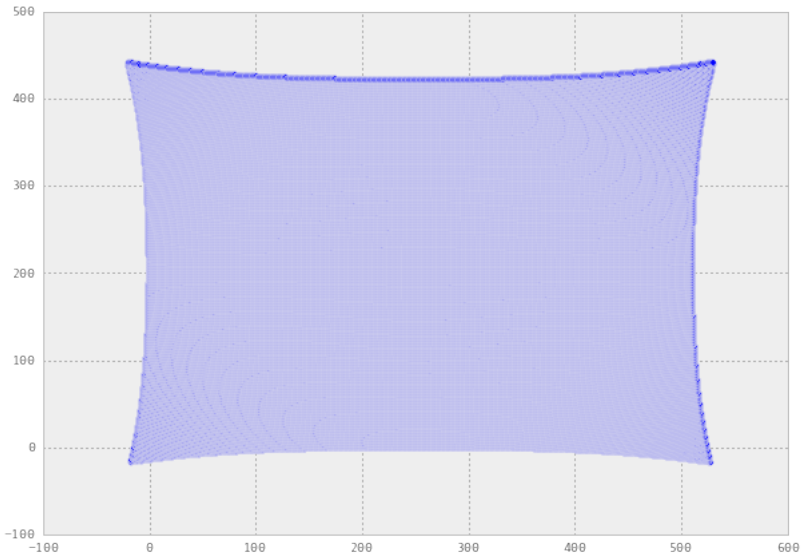


Figura 3.3: Schema della depth image sovrapposta a quella a colori

camera, in coordinate omogenee per Kinect 1 e in coordinate mondo per il Kinect 2. Al momento della generazione della mappa delle disparità verrà creata una pointcloud che conterrà i dati 3D di ogni punto valido della mappa delle disparità corrispondente. Alla fine dell'acquisizione quindi, si otterranno 3 strutture dati, una contenente l'immagine a colori, una quella di profondità e una la point cloud associata.

3.4 Inferimento del piano

Dato che stiamo supponendo di lavorare su un conveyor, dobbiamo calcolarne la distanza dalla telecamera per poi poter così misurare l'altezza degli oggetti che passeranno in un secondo momento. Questa si può considerare come una fase di setup in quanto è sufficiente calcolarla una sola volta all'inizio dell'ispezione (le camere sono generalmente in una posizione fissa rispetto al nastro trasportatore e non si muovono durante le ispezioni, può quindi considerarsi una variazione di questi parametri come una variazione dell'ambiente stesso).
A questo punto, viene selezionata una regione appartenente al piano da identificare, ed estratti i punti dalla point cloud associati ad esso. Questi punti verranno usati per calcolare l'equazione del piano media ed una volta ottenuta, verranno scartati gli *outliers*. Una volta scartati, si riprocede al calcolo del piano ottenendo così una misurazione più accurata.
L'equazione del piano può anche essere calcolata durante l'esecuzione andando a selezionare un'area che non presenta oggetti nell'immagine, ma dato che come già spiegato prima l'ambiente non cambia durante l'esecuzione, risulta più conveniente usare un'unica immagine di setup ed effettuare il calcolo in assenza di elementi che possono provocare rumore nella scena.

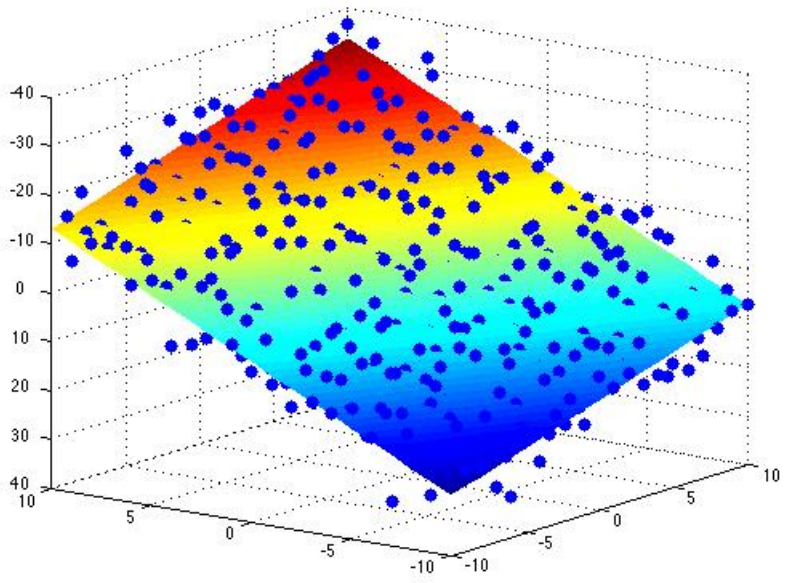


Figura 3.4: Calcolo del piano da una point cloud

3.5 Object Detection

Arrivati a questo punto, bisogna procedere all'identificazione degli oggetti o dei pacchi che scorrono sopra il conveyor per procedere alel misurazioni. Per far ciò verrà fatto con l'ausilio del *Blob Tool* di *Impact*.
Dato che abbiamo precedentemente calcolato il piano su cui giace il conveyor e quindi la sua distanza dalla telecamera, possiamo effettuare un primo filtraggio sulla depth image andando a scartare tutti quei punti che giacciono a una distanza uguale o superiore a quella del piano di riferimento. Questo è utile perché ci consente di filtrare la maggior parte dell'immagine e lasciare visibili soltanto gli elementi che scorrono sul nastro.
Inoltre, dato che buona parte del rumore delle camere Kinect è dato dallo scattering, escludere anche punti poco al di sopra del piano di riferimento permette di tagliare anche parte del rumore dovuto all'errore di misurazione dei punti prossimi al piano che sono soggetti a più errore, consentendo l'analisi di un blob più continuo.

Vediamo adesso il caso in figura 3.5. L'immagine è stata acquisita tramite Impact mediante l'uso del Kinect 2. In figura 3.6 si può vedere la corrispondente depth image senza alcun

Figura 3.5: Immagine a colori acquisita tramite Kinect 2

filtro applicato. È immediato notare una selezione del blob basata sul livello di grigio in questo caso possa risultare difficoltosa da effettuare in modo preciso sui pixel che risultano sul bordo

della ruota.

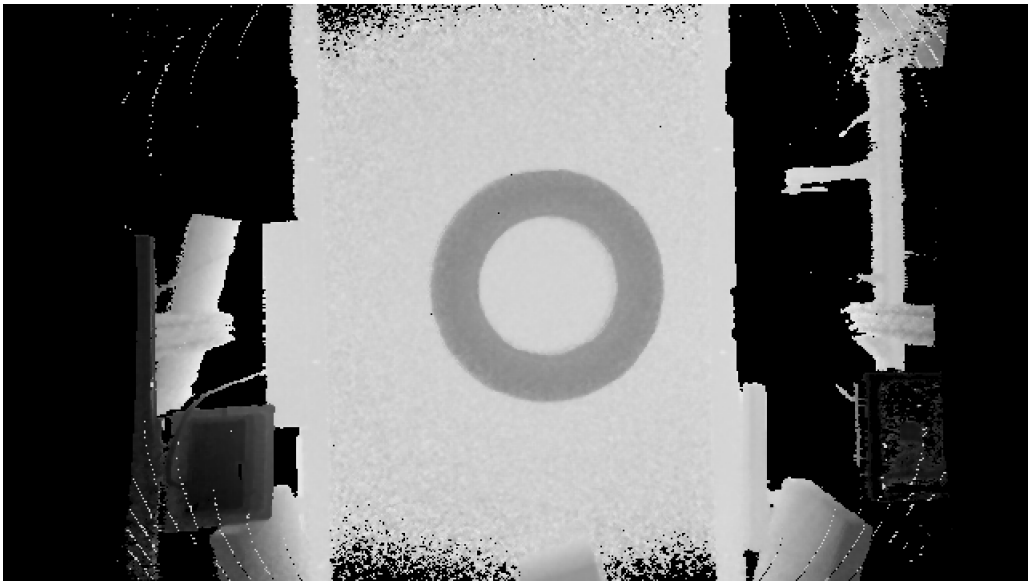


Figura 3.6: Depth Image corrispettiva

Vediamo adesso, con riferimento alla figura 3.7, la stessa depth image con i valori di profondità filtrata risulti molto più semplice da visualizzare e racchiudere all'interno di un blob.
L'immagine 3.7 è creata a partire dagli stessi dati di profondità della 3.6, l'unica differenza è che al momento dell'assegnamento dell'intensità del pixel si vanno a scartare i pixel che hanno un valore di profondità al di fuori del range delle soglie T_{min} e T_{max}, mentre agli altri verrà assegnato un valore di profondità pesato rispetto a queste soglie.
I due valori di soglia vengono scelti T_{max} come la distanza del conveyor dalla camera calcolata nella fase di setup meno un piccolo valore di scarto, nei casi in esame pari a 2 millimetri e T_{min} come la distanza minima che si vuole visualizzare, e comunque non minore ai valori minimi di detection delle due camere Kinect (quindi 40 o 80 millimetri per Kinect 1 e 50 millimetri per Kinect 2).

Il valore di intensità del generico pixel è così calcolato:

$$intensity = \begin{cases} 255\frac{depth - T_{min}}{T_{max} - T_{min}}, & \text{se } T_{min} < depth < T_{max} \\ 0, & \text{altrimenti} \end{cases}$$

Anche se si filtra la profondità non si è esenti da errori, ma questo aiuta anche a tagliare

ulteriori errori di scattering quando più oggetti passano a distanza ravvicinata, rendendo anche più semplice la separazione dei blob. Nell'immagine filtrata ovviamente non si riesce ad escludere tutta la parte del conveyor non necessaria all'ispezione, basti guardare al lato le componenti del macchinario e alcuni punti di rumore agli estremi del conveyor. Questo comunque non presenta un problema in quanto la posizione del conveyor è nota a priori, possiamo quindi selezionare un'area specifica all'interno dell'immagine in cui andare a ricercare i blob di interesse.
A questo punto, si può procedere con il calcolo del blob. L'operazione è abbastanza semplice in quanto sappiamo in che regione trovare il blob e in quell'area l'immagine risulta praticamente binaria.

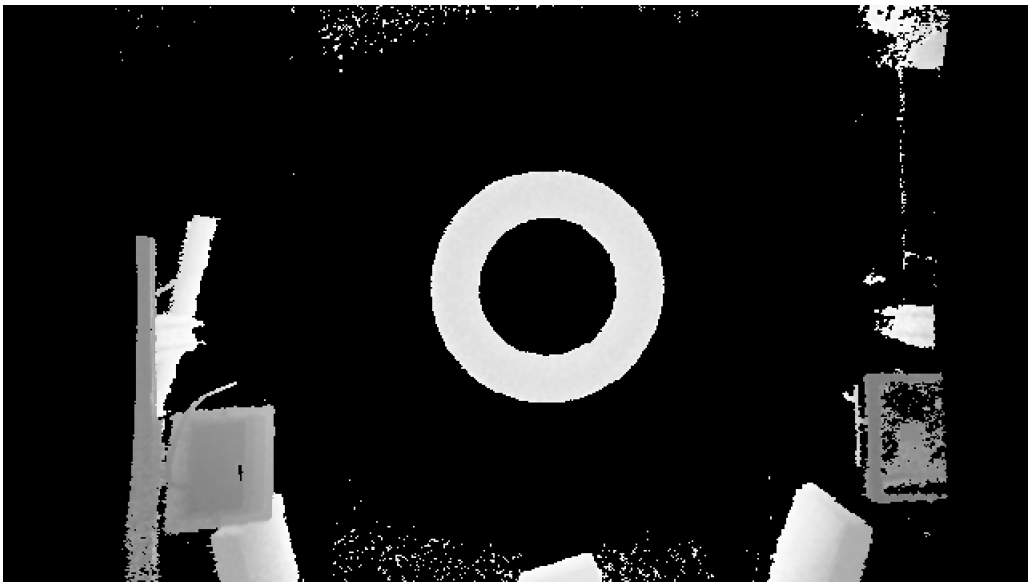


Figura 3.7: Depth Image filtrata

3.6 Raffinamento del Blob

Come già spiegato nel paragrafo precedente, non è sufficiente filtrare la depth per ottenere un blob accurato dell'oggetto inquadrato, questo perché le camere Kinect presentano un rumore laterale e assiale che aumenta in modo proporzionale alla distanza.
In figura 3.8 si mostra un grafico dell'andamento dei due tipi di rumore al variare della distanza. In [MDM14] viene presentato uno studio approfondito in merito alla struttura del rumore e alla sua possibile correzione.

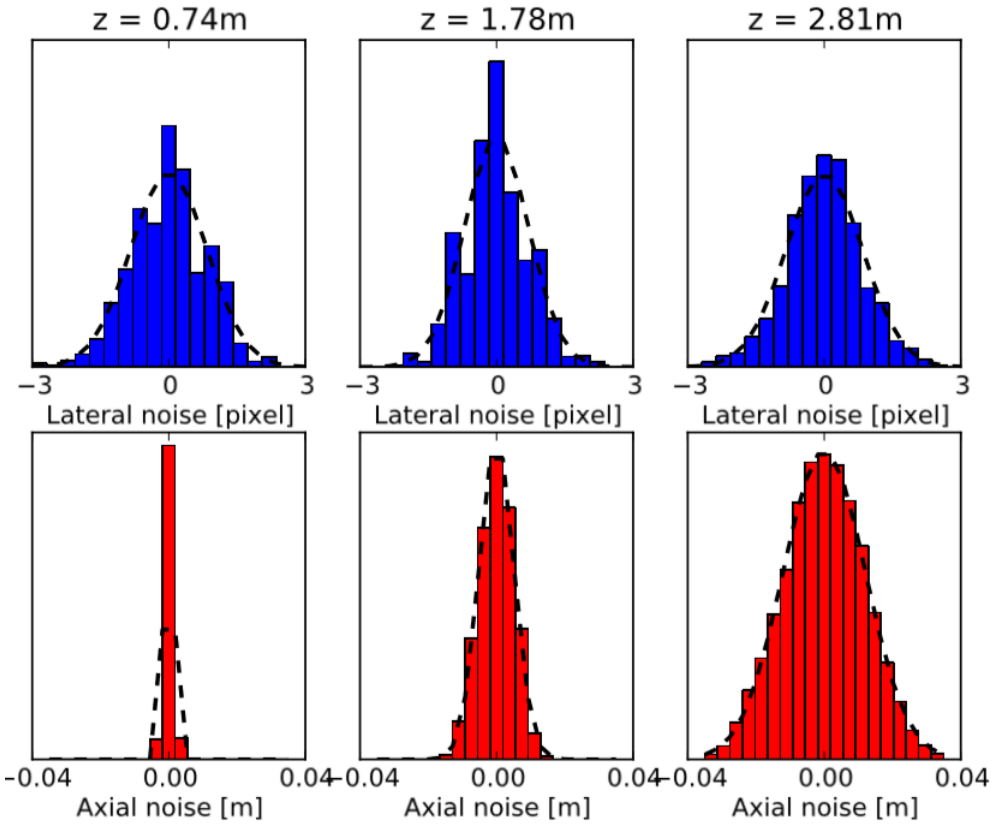


Figura 3.8: Modello del rumore del Kinect. [MDM14]

È quindi necessario ricorrere a delle tecniche di preprocessing per affinare i risultati delle misurazioni. Per questa operazione verranno utilizzati due tool di Impact, chiamati *Erode Tool* e *Polygon Smoothing*.
L' *Erode Tool* esegue, come già suggerisce il suo nome, l'operazione morfologica di erosione in un'area specificata. Seguiamo il processo tramite le immagini, utilizzando in tutti gli esempi il Kinect 2.
In figura 3.9 vediamo la scena in esame ripresa dalla telecamera e la sua corrispettiva immagine di profondità. Si può vedere un particolare in figura 3.10 in cui è possibile vedere del rumore impulsivo sul bordo del pacco, oltre che per la forma anche per il valore di profondità rilevato, molto diverso dal resto dell'oggetto.
Applicando l'erosione, buona parte del rumore sui bordi è stato tagliato, come si vede nella figura 3.11. Comunque, questo non risulta sempre efficace riguardo l'attenuazione del rumore negli angoli al che si ricorre al secondo tool elencato, il *Polygon Smoothing Tool*, per correg-

gere questo aspetto.

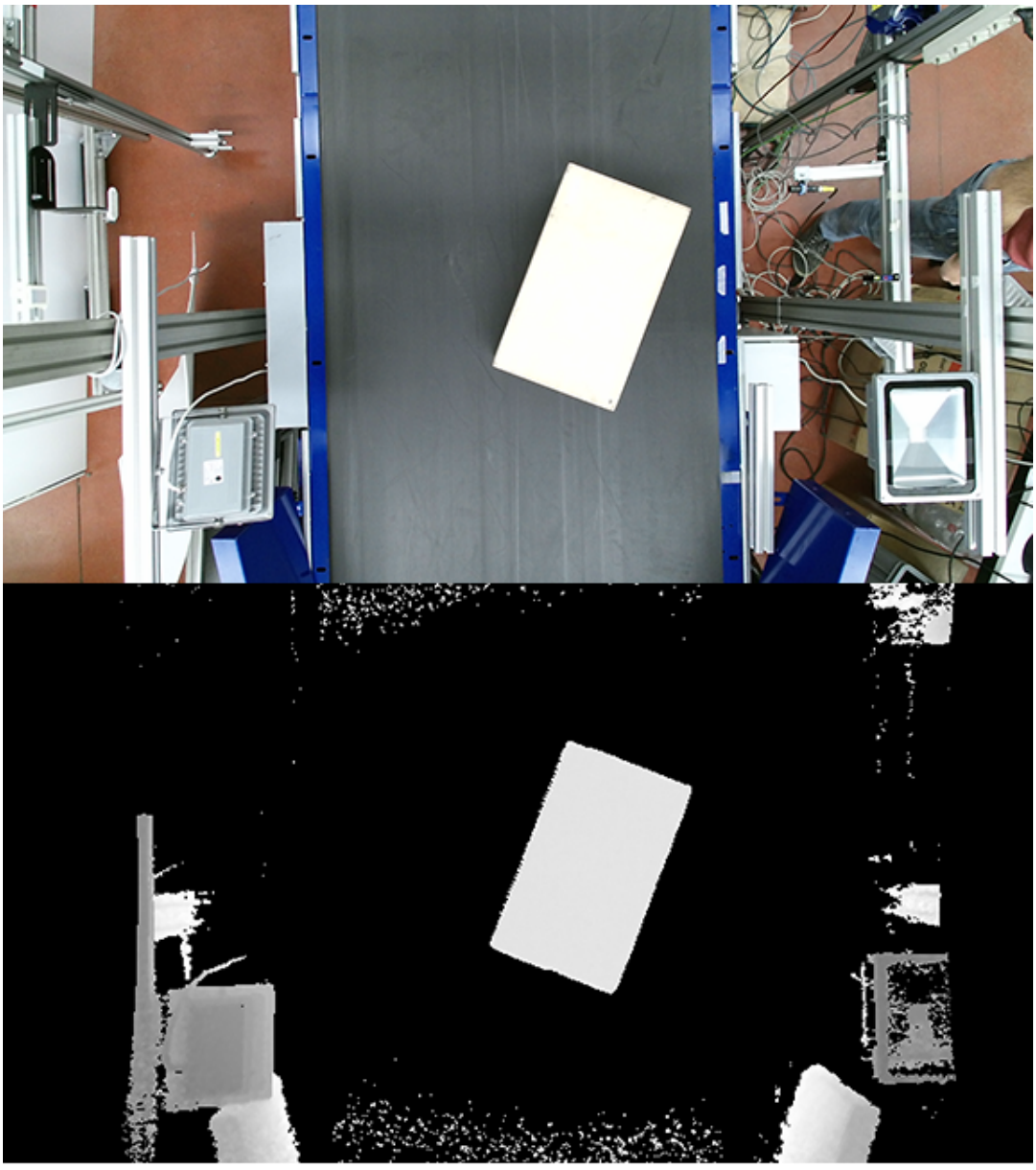


Figura 3.9: Scena a colori e depth image corrispettiva

Figura 3.10: Particolare della depth image

Il Polygon Smoothing Tool esegue un'operazione di smoothing gaussiano tramite convoluzione, ma esso non agisce sull'immagine ma bensì su una lista di poligoni passata in ingresso. Quindi, una volta erosa l'immagine al primo passo, si costruisce il blob sulla base di questo e poi si passa direttamente a fare smoothing sulla forma del blob anziché nuovamente sull'immagine. Questo consente di attenuare meglio il rumore s ugli spigoli che incide maggiormente sulla successiva costruzione del rettangolo orientato.
In figura 3.12 si può vedere un dettaglio dello smoothing applicato e dell'attenuazione della forma del blob risultante. L'applicazione migliora il risultato perché addolcire uno spigolo idealmente di 90° non varia la dimensione del rettangolo minimo che lo contiene, mentre un rumore in quel punto è più soggetto a cambiarne dimensione e variarne l'orientazione.
Il risultato finale del preprocessing si può osservare in figura 3.13, che mostra la bounding box calcolata mostrata sull'immagine prima della fase di preprocessing, dove si può osservare come la quasi totalità del rumore sia stata esclusa.

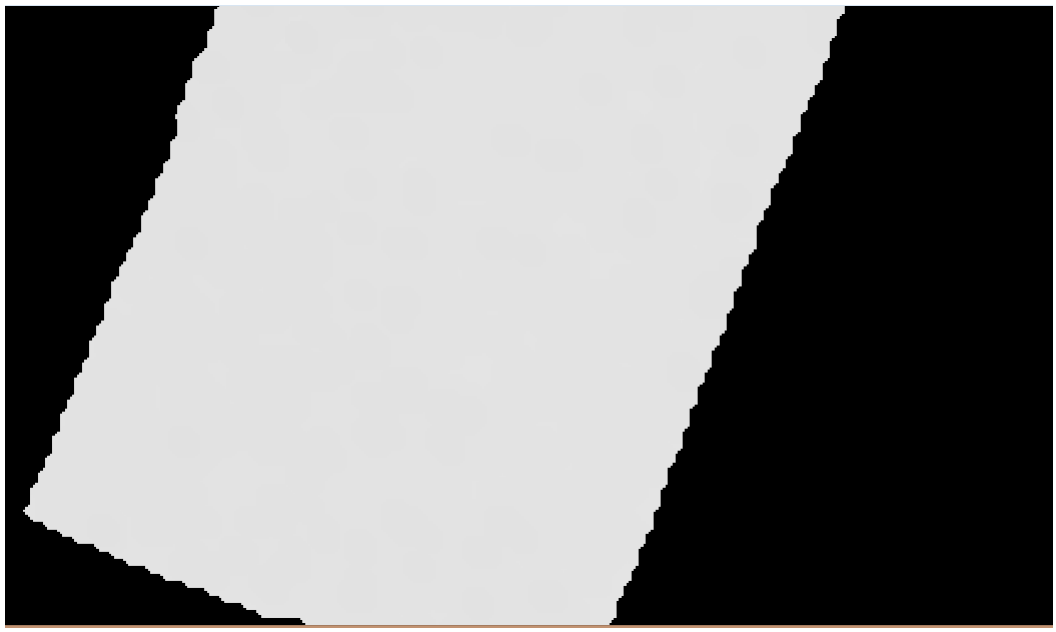


Figura 3.11: Particolare della depth image dopo l'erosione

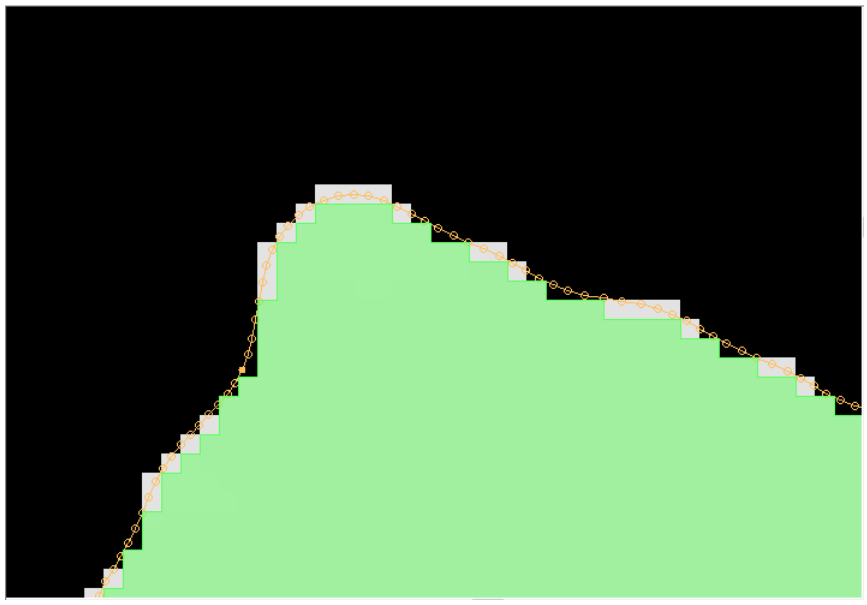


Figura 3.12: Polygon smoothing

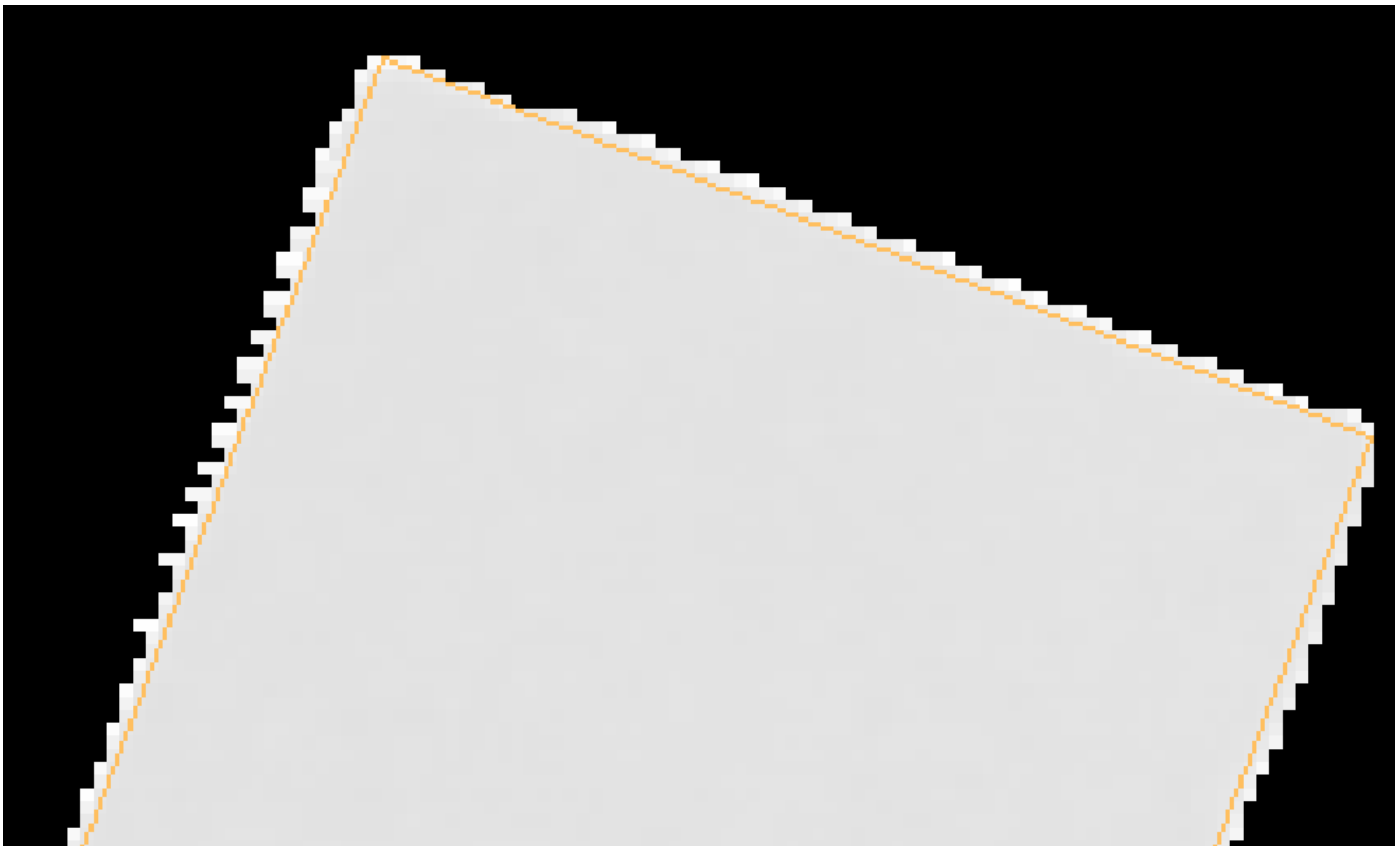


Figura 3.13: Il rettangolo orientato calcolato rispetto all'immagine originale.

3.7 Calcolo della Bounding Box 3D Orientata

Una volta ottenuto processato il blob in maniera opportuna, si può procedere alla misurazione della bounding box 3d. Il blob risulta necessario per due motivi:

1. Calcolare l'orientazione della bounding box.
2. Ffiltrare i punti della point cloud.

Riguardo il primo punto, ricavando dal blob il *major angle* e orientando la bounding box secondo di esso, ci garantisce che la dimensione dell'area stessa sia minima. L'immagine in figura 3.14 mostra in maniera chiara questo concetto.

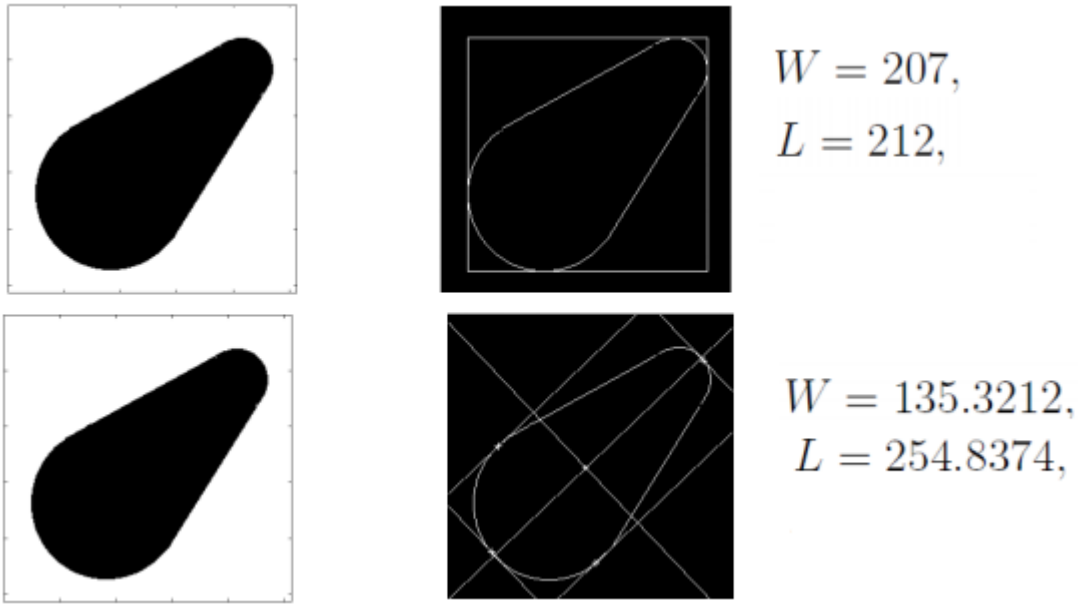


Figura 3.14: Bounding Rectangle non orientato e orientato.[1]

Il major angle è ricavato mediante l'uso dei *momenti centrali* (1.1.2) con la seguente formula:

$$\text{major angle} = \frac{1}{2}\arctan\left(\frac{2M'_{1,1}}{M'_{2,0} - M'_{0,2}}\right)$$

La figura 3.16 mostra un esempio di applicazione di quanto spiegato fin ora. L'immagine mostra la depth image generata tramite uno scatto effettuato durante il pasaggio di due pacchi all'interno del conveyor tramite ill Kinect 1 e con la profondità filtrata. Il rettangolo racchiude l'area di ricerca dei blob nella scena, in modo da escludere a priori le parti della scena non necessarie. Nel momento in cui viene trovato il blob riferito al pacco viene calcolato il suo major angle per valutarne l'orientazione, come si può vedere dalle linee che lo attraversano. Il secondo punto riguarda la selezione di quei punti all'interno della point cloud che si andranno ad analizzare per stimare la bounding box. Si andrà quindi a cercare nei punti appartenenti alla scena 2D della depth image i corrispettivi punti della pointcloud in 3D. Fatto questo, bisognerà calcolare quali sono i punti di massimo e minimo lungo e componenti x, y e z.
Per quanto riguarda le z, abbiamo il piano inferito inizialmente come punto di riferimento che corrisponde allo 0 e chiaramente gli oggetti passano appoggiati ad esso pertanto il minimo sarà quello ed il massimo corrisponderà al punto della pointcloud con distanza maggiore tra quelli

[1] http://bit.ly/1AHlvnX

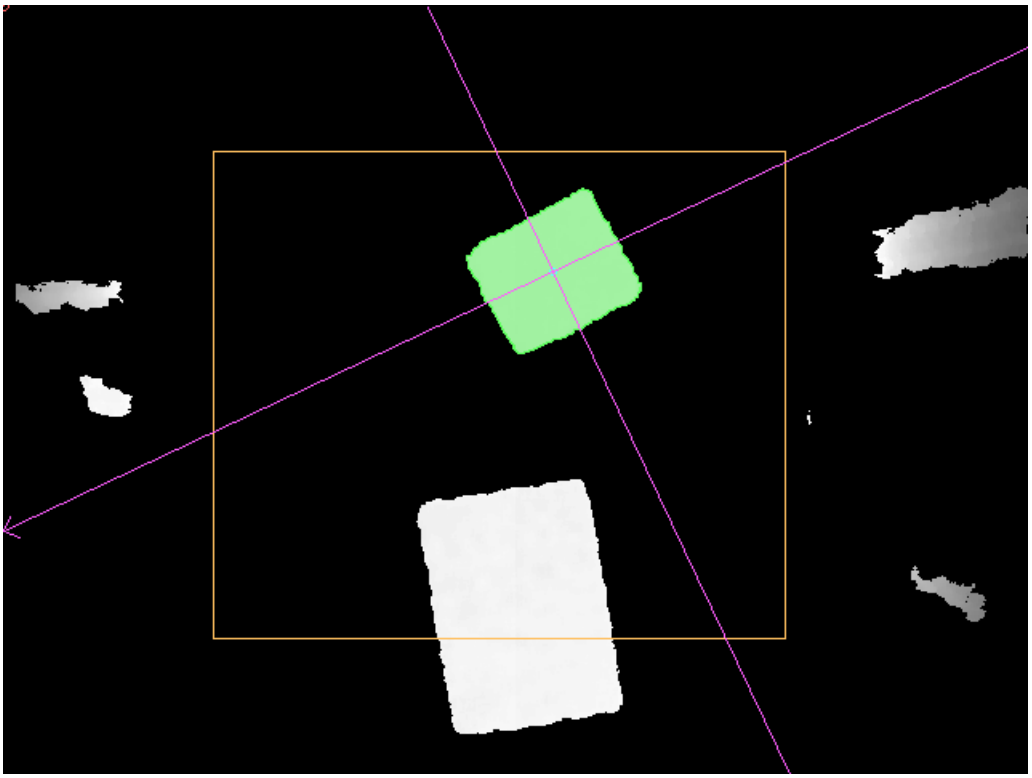


Figura 3.15: Blob identificato con orientazione dentro un'area di ricerca.

che sono stati estratti.
Riguardo x e y i punti vengono momentaneamente ruotati in base al major angle, in modo tale da orientare i punti con il sistema di riferimento. Detto questo è sufficiente cercare nella point cloud i valori minimi e massimi sia per x che per y per ottenere i dati necessari a definire gli otto vertici della bounding box 3D.

I vertici saranno i seguenti:

$$P_1 = (x_{min}, y_{min}, 0) \qquad P_5 = (x_{min}, y_{min}, z_{max})$$
$$P_2 = (x_{max}, y_{min}, 0) \qquad P_6 = (x_{max}, y_{min}, z_{max})$$
$$P_3 = (x_{max}, y_{max}, 0) \qquad P_7 = (x_{max}, y_{max}, z_{max})$$
$$P_4 = (x_{min}, y_{max}, 0) \qquad P_8 = (x_{min}, y_{max}, z_{max})$$

Misurando le distanze euclidee tra i punti, possiamo ottenere le misure della bounding box 3D. Le misure sono reali perché i dati fanno riferimento alla point cloud, ottenendo quindi

altezza, lunghezza e larghezza in millimetri.

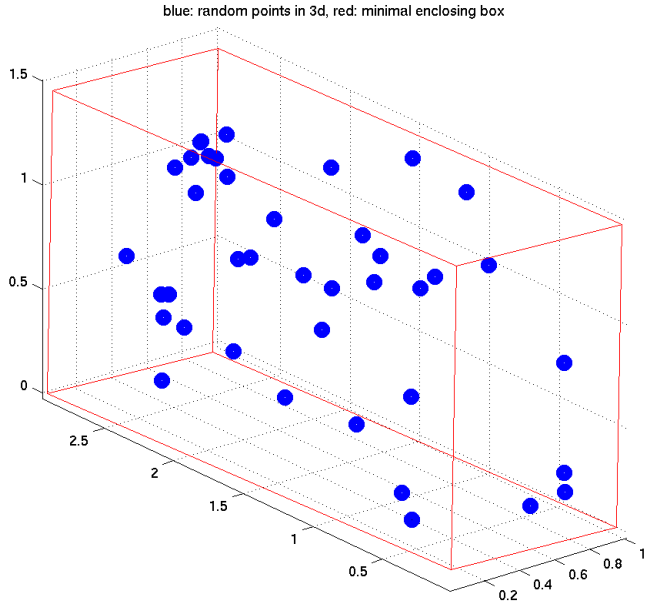


Figura 3.16: Punti 3D racchiusi dalla bounding box minima e orientata.

Capitolo 4

Risultati sperimentali

In questo capitolo verranno esposti i risultati sperimentali ottenuti tramite una serie di test realizzati appositamente per poter valutare la qualità e la robustezza dei dati ottenuti tramite il sistema proposto nel precedente capitolo.
Verranno utilizzati oggetti via via dalle forme più regolari a quelle più irregolari e con materiali differenti per poter valutare la risposta del sistema anche alle variazioni di colore e materiale. Ogni dataset è acquisito con gli oggetti posti staticamente nella scena in posizioni e orientazioni diverse e mentre scorrono all'interno del nastro.
Si analizzerà quindi lo stesso dataset ma acquisito due volte con i due differenti Kinect e si valuteranno le prestazioni in base ai due tipi di camere.

Gli oggetti dei dataset sono posti su un conveyor di larghezza un metro, con le telecamere poste in alto ad una distanza di 150 cm perpendicolari al piano del conveyor.
I tipi di dataset creati ed utilizzati per questi esperimenti sono:

Wood box: Una scatola di legno, dalle dimensioni di 500x300x150 millimetri, usata principalmente a come oggetto di test dalla Datalogic durante le verifiche sui software di dimensionamento. L'oggetto è perfettamente regolare e non presenta materiali o segni nel suo rivestimento che rendono difficoltoso la detection di questo oggetto.

Polystyrene: Una scatola di polistirolo bianca, dalle dimensioni di 545x322x161 millimetri, presenta alcune ammaccature e difetti, inoltre non è perfettamente rettangolare.

Wheel: Un pneumatico da auto dalle dimensione di 540x540x165 millimetri, usato come campione per test sul colore nero.

Tricycle: Un triciclo per bambini delle dimensioni stimate di 670x435x490 millimetri, altamente irregolare nella forma e materiale in plastica lucida.

4.1 Wood Box

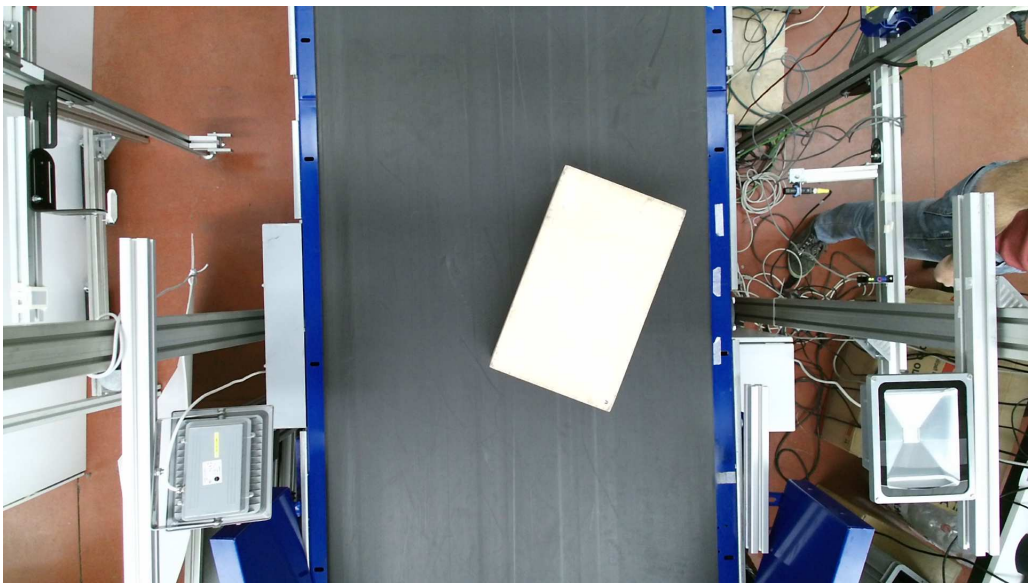


Figura 4.1: Dataset Wood Box

Questo rappresenta il dataset più semplice tra quelli presi in considerazione, perché presenta sia una forma molto regolare senza smussature, perfettamente rettangolare ed il materiale stesso ben si presta all'ispezione, in quanto non possiede caratteristiche fisiche che alterano la riflessione della luce sulla sua superficie.
I grafici mostrati di seguito indicano il grado di errore della misurazione di ogni lato dell'oggetto rispetto alla misura reale per ogni frame acquisito, l'errore medio e la rispettiva deviazione standard (punti tratteggiati).

4.1.1 Kinect 1

Come si può vedere dai grafici in figura 4.2 e 4.3, gli errori medi rispetto ai valori reali si attestano sui 3 e 1.5 millimetri. Il problema principale è dato chiaramente dall'errore di misurazione sui bordi, che porta a degli errori che possono arrivare fino a una distanza di 6 millimetri. Riguardo la profondità (figura 4.4 il il risultato è molto stabile, attestandosi ad una sovrastima costante di 2 millimetri.

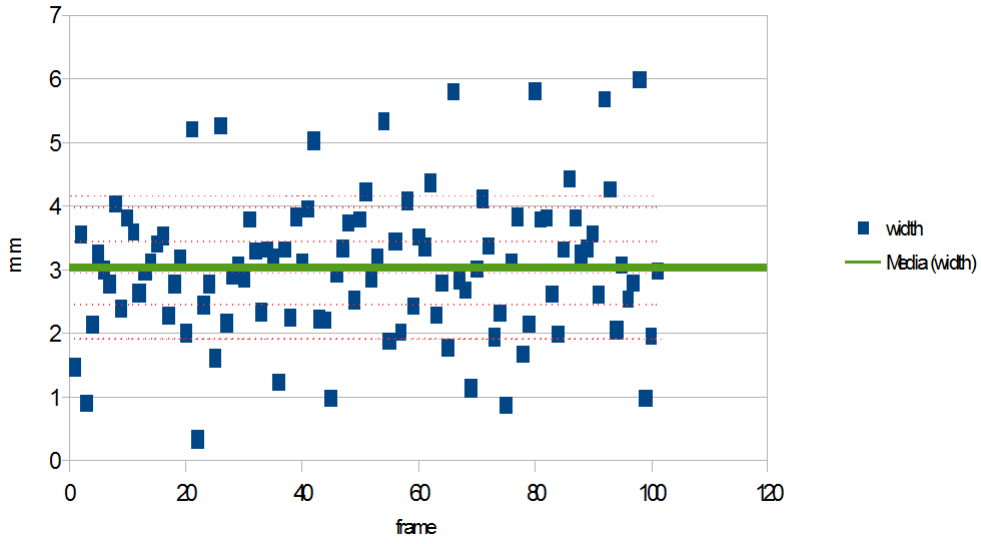


Figura 4.2: Kinect 1 - width

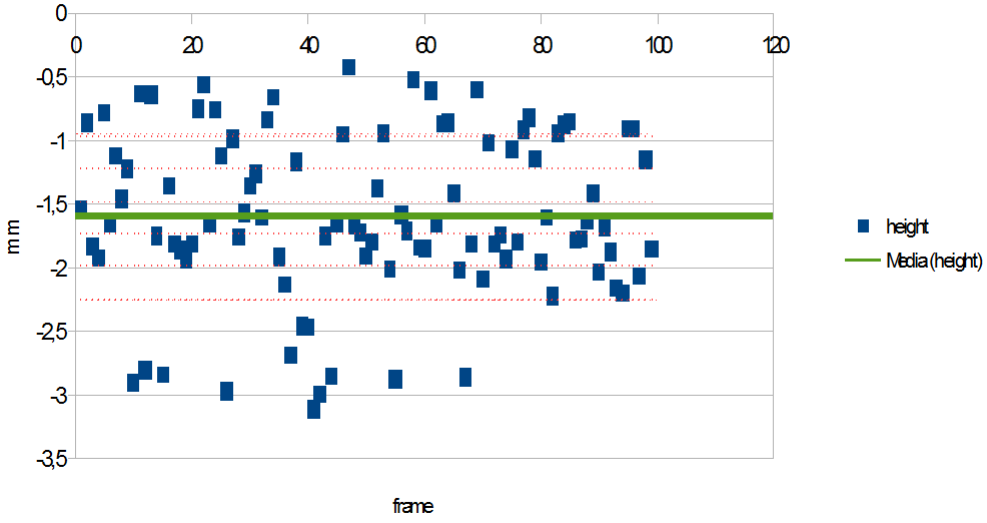


Figura 4.3: Kinect 1 - height

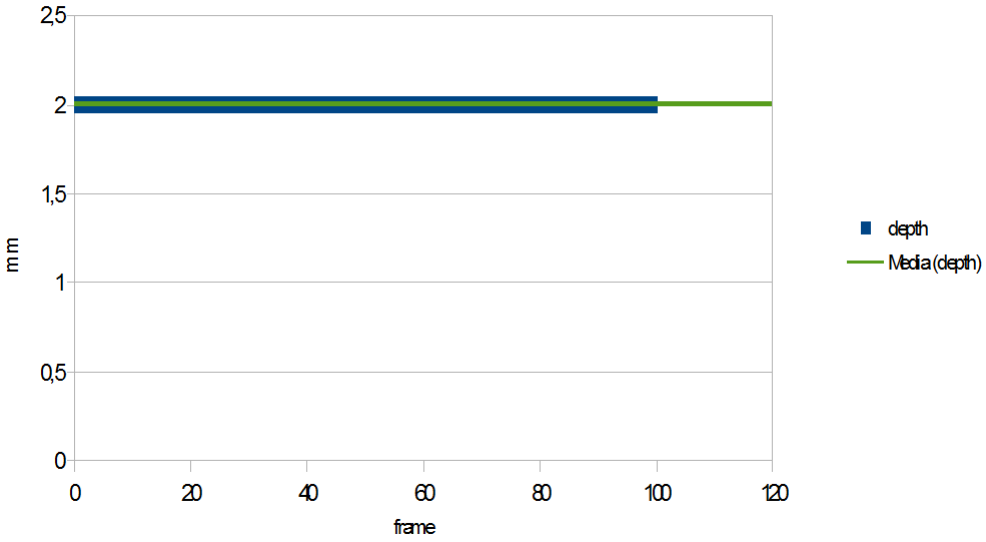


Figura 4.4: Kinect 1 - depth

Dato	Width	Height	Depth
min	0.337	-3.109	2
max	5.988	-0.418	2
media	3,035	-1,593	2
mediana	3,008	-1,658	2
deviazione standard	1,128	0,653	0

Tabella 4.1: Misure degli errori su Wood Box con Kinect 1 mm

4.1.2 Kinect 2

Le misurazioni effettuate col Kinect 2 visibili in figura 4.5 e 4.6 presenta un miglioramento della misurazione media dell'oggetto, con un margine di errore attorno al millimetro. Anche qui la misurazione presenta una soglia di varianza di 2 millimetri, ma risulta più attenuato rispetto alla controparte del Kinect 1. La misurazione della profondità (figura 4.7) non è costante come si è assistito per il Kinect 1, ma soffre di una variazione come quella della lunghezza e della larghezza.

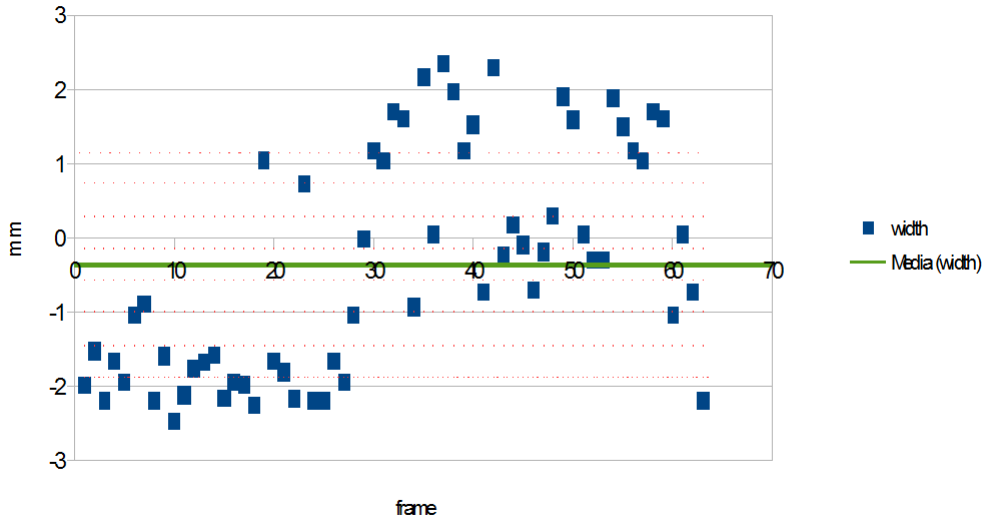


Figura 4.5: Kinect 2 - width

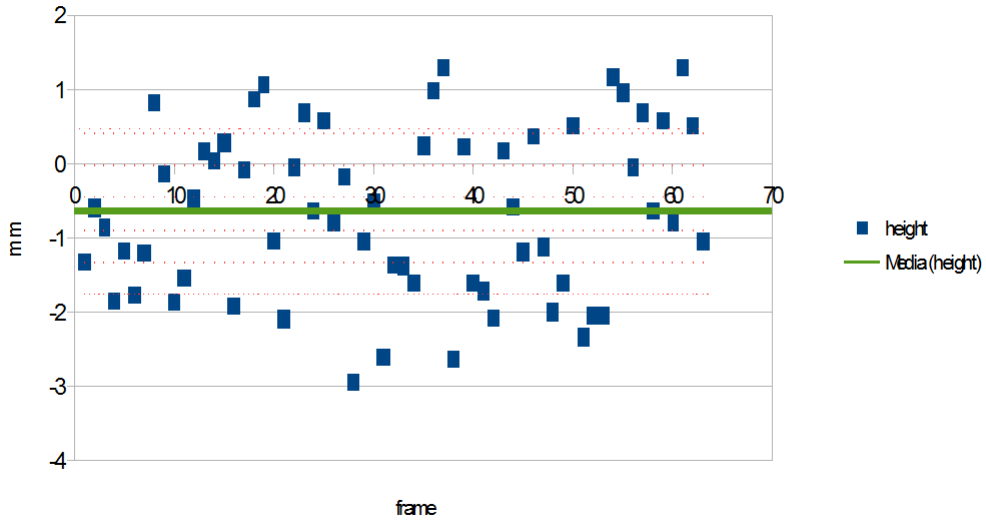


Figura 4.6: Kinect 2 - height

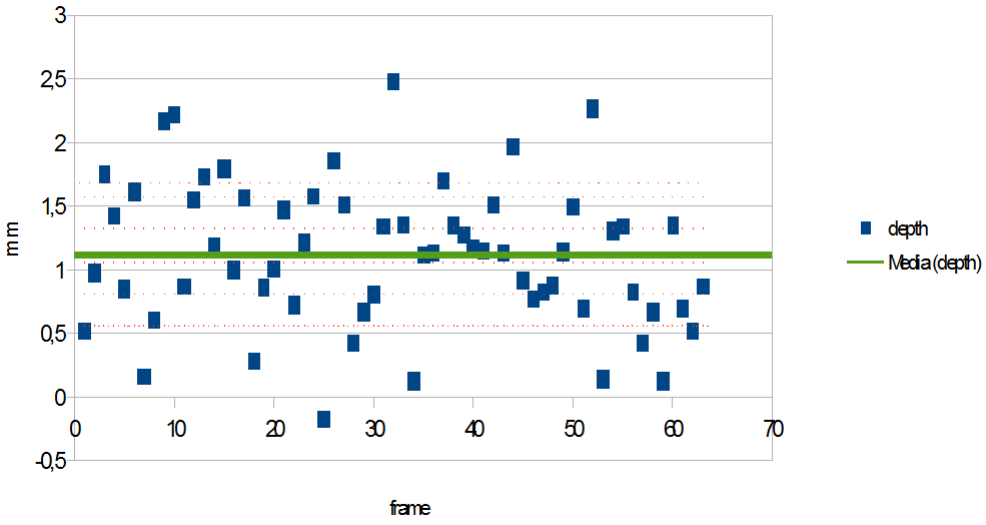


Figura 4.7: Kinect 2 - depth

Dato	Width	Height	Depth
min	-2,461	-3,913	-0,170
max	3,554	1,305	4,135
media	-0,283	-0,689	1,164
mediana	-0,701	-0,631	1,143
deviazione standard	1,671	1,212	0,684

Tabella 4.2: Misure degli errori su Wood Box con Kinect 2 in mm

4.1.3 Comparazione

In figura 4.8 viene mostrato il grafico riferito alle variazione dell'errore del volume stimato della scatola rispetto a quello reale, in particolare, appare evidente il miglioramento dei risultati ottenuti utilizzando a parità di algoritmo il Kinect 2. Questo si assesta con un errore medio di $2,49mm^3$ mentre il primo modello sui $4,04mm^3$.
Per quanto riguarda questo dataset, le prestazioni fornite dal Kinect 2 possono ritenersi buone, sia per quanto riguarda l'andamento medio sia per i picchi, che non arrivano mai a 5 millimetri.

Il Kinect 1 invece presenta raramente misurazioni migliori rispetto al secondo modello, con un errore superiore di circa un millimetro e mezzo. Le prestazioni di per se' possono ritenersi anch'esse discrete, per lo più al di sotto del mezzo centimetro, ma non sono paragonabili a quelle della sua controparte.

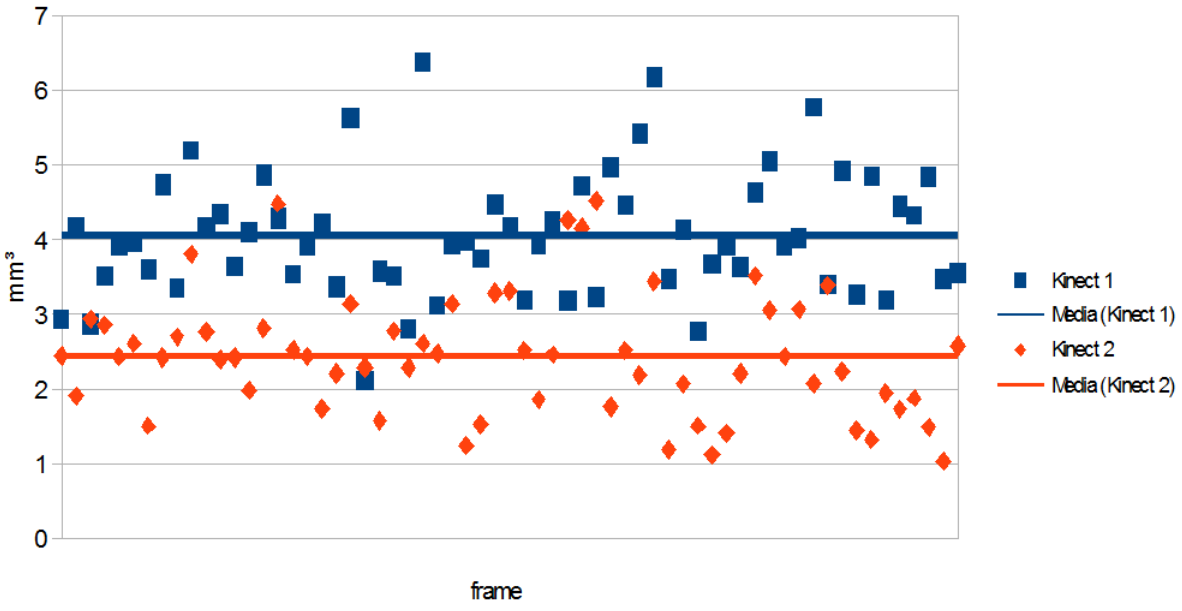


Figura 4.8: Wood Box: comparazione tra Kinect 1 e Kinect 2

4.2 Polystyrene

Figura 4.9: Dataset Polystyrene

Questo dataset presenta delle difficoltà di detection leggermente maggiori di quello precedente. Come prima cosa a scatola non si presenta in maniera uniforme, dato che contiene alcuni difetti sui bordi e non è perfettamente rettangolare. Inoltre il polistirolo può dare errori di rifrazione dovuto al tipo di materiale che possono alterare le misurazioni.

4.2.1 Kinect 1

Si può facilmente osservare come, a differenza del dataset precedente, come i valori siano sempre sottostimati a prescindere da quanto oscilli il valore misurato (figure 4.10 e 4.11).
Riguardo la profondità (figura 4.12), anche qui si osserva una misurazione costante, ma che

si assesta su ben 13 millimetri al di sopra del valore reale, ben di più del precedente che si fermava a soli 2.
È lecito supporre quindi che il tipo di materiale coinvolto alteri la proiezione del fascio di luce strutturata, che come illustrano i grafici, risulta costantemente sottostimata a i bordi e sovrastimata sulla profondità.

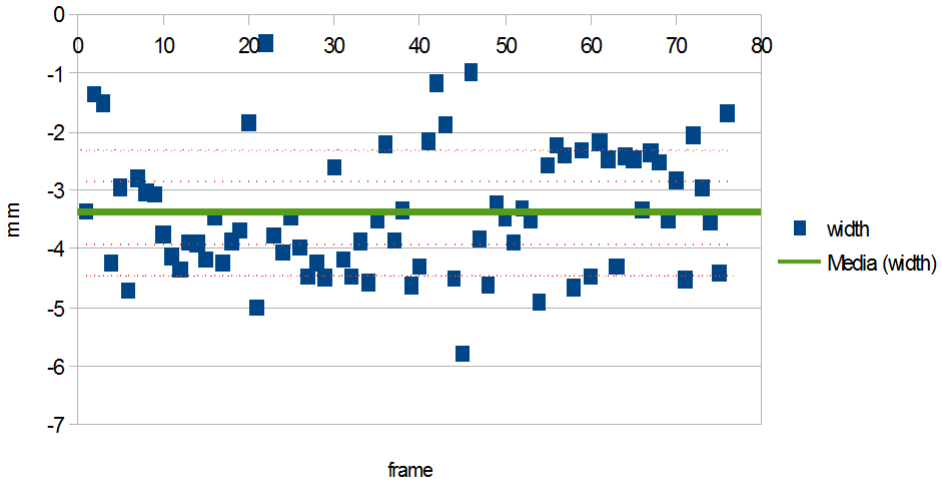


Figura 4.10: Kinect 1 - width

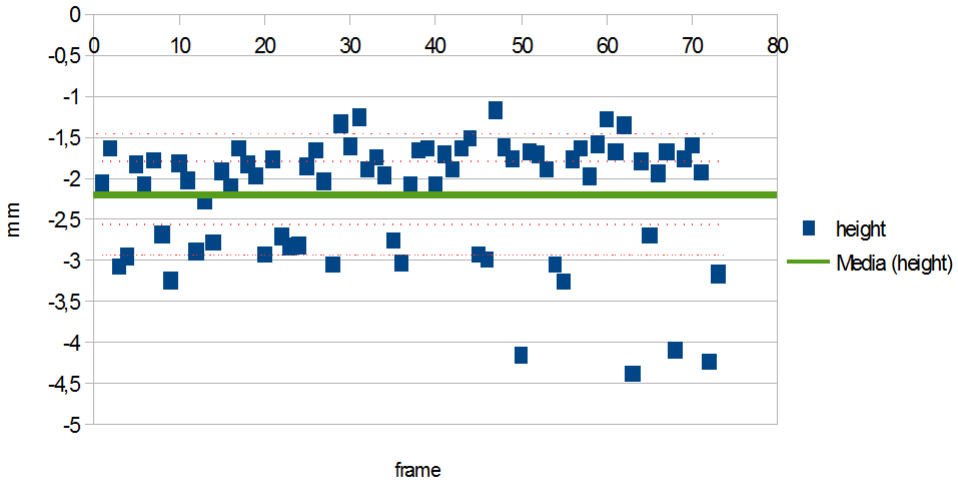


Figura 4.11: Kinect 1 - height

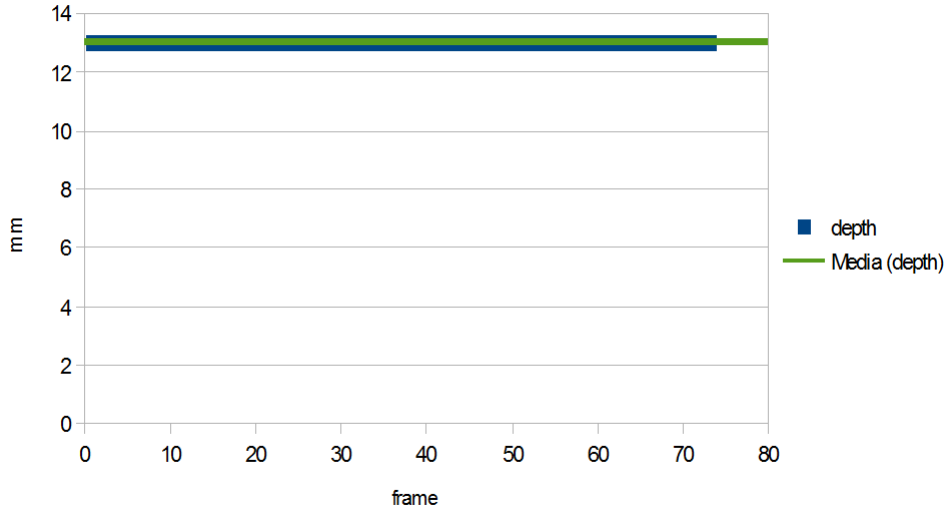


Figura 4.12: Kinect 1 - depth

Dato	Width	Height	Depth
min	-5,787	-4,453	13
max	-0,495	-1,165	17
media	-3,393	-2,230	13,432
mediana	-3,509	-1,919	13
deviazione standard	1,068	0,783	1,251

Tabella 4.3: Misure degli errori su Polystyrene con Kinect 1 in mm

4.2.2 Kinect 2

I valori qui seguono l'andamento mostrato anche per il dataset Wood Box sia in termini di oscillazione che di misurazioni, risultando comunque più robusto al tipo di materiale rispetto al Kinect 1 (figure 4.13 , 4.14,4.15).
I grafici sembrano mostrare non molte differenze rispetto alle misurazioni del dataset precedente, che lasciano supporre una maggiore robustezza della camera rispetto al tipo di materiale utilizzato.

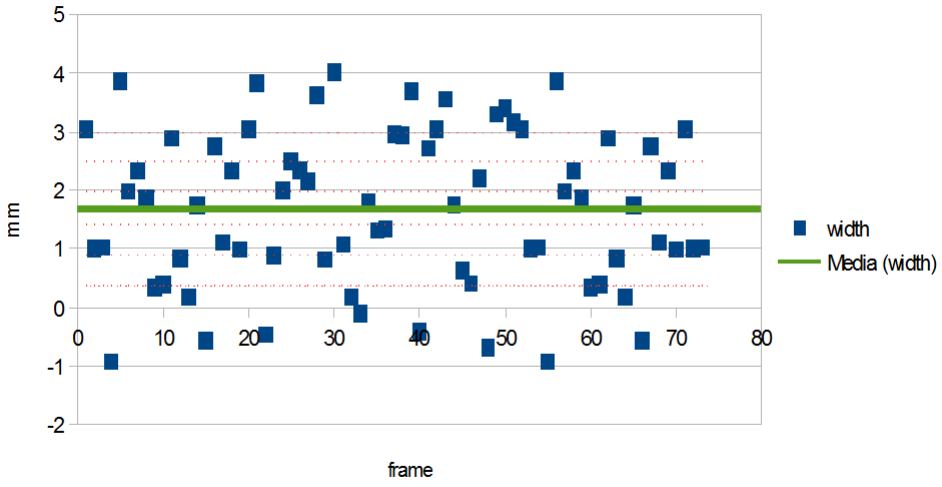


Figura 4.13: Kinect 2 - width

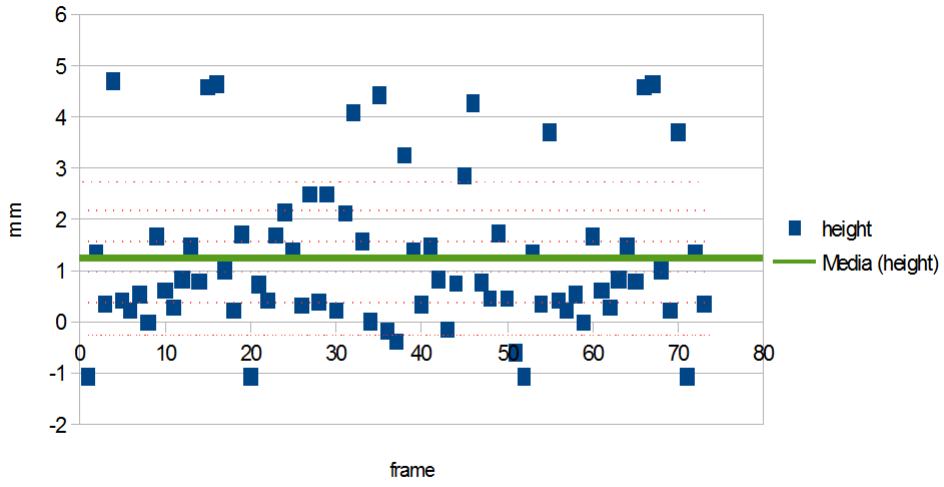


Figura 4.14: Kinect 2 - height

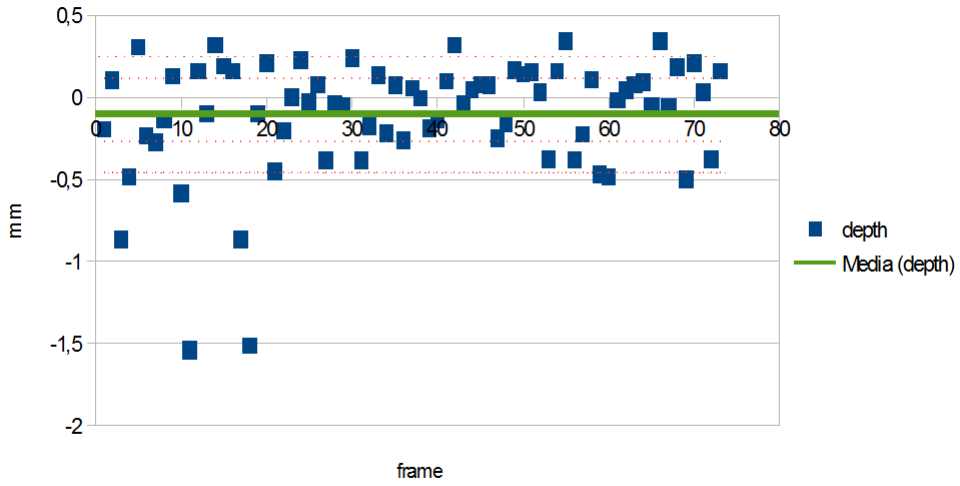


Figura 4.15: Kinect 2 - depth

Dato	Width	Height	Depth
min	-0,929	-1,057	-1,323
max	4,009	4,705	0,352
media	1,680	1,250	-0,130
mediana	1,749	0,796	-0,080
deviazione standard	1,327	1,509	0,378

Tabella 4.4: Misure degli errori su Polystyrene con Kinect 2 in mm

4.2.3 Comparazione

Anche qui, la differenza di prestazioni tra le due camere è evidente (figura 4.16). La componente maggiore di errore del Kinect 1 è data dalla profondità, che come visto nei grafici precedenti mostra sempre un grado di errore molto più alto, che portano l'errore medio sul volume su $14,17$ e una deviazione standard di $1,32$.
Il Kinect 2 risulta più efficace nelle misurazioni, attestandosi con un errore medio di $2,65$ con una deviazione standard di $1,15$, con risultati paragonabili a quelli del dataset Wood Box, che lo mostra robusto al cambio del materiale utilizzato.

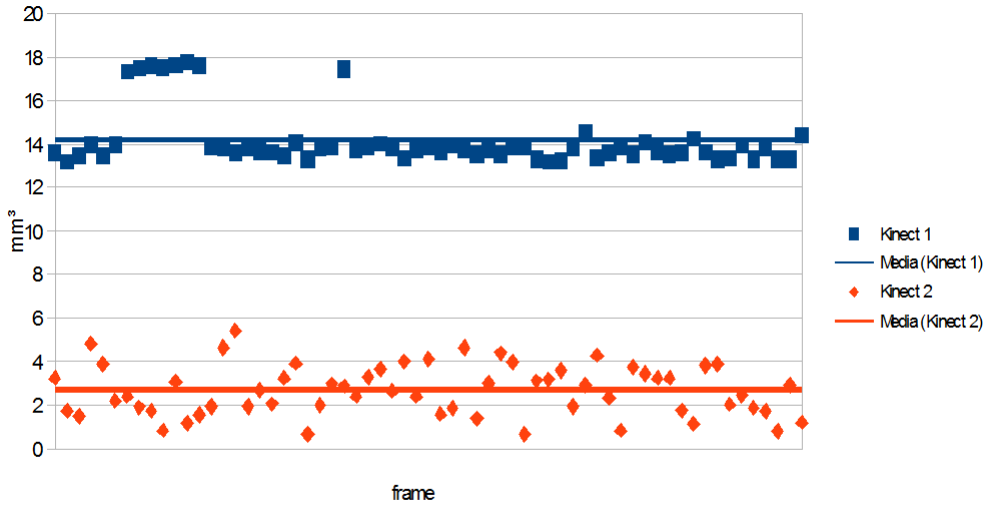


Figura 4.16: Polystyrene: comparazione tra Kinect 1 e Kinect 2

4.3 Wheel

Figura 4.17: Dataset Wheel

Cambiando genere di dataset, si vuole verificare la reazione del sistema e delle camere su oggetti di circolari. In aggiunta questo dataset permette di testare le due camere per quanto riguarda la robustezza al colore nero, su cui molti tipi di camere 3D presentano problemi.

4.3.1 Kinect 1

Nonostante il colore, il Kinect 1 riesce ad effettuare delle misurazioni sulla larghezza e lunghezza abbastanza coerenti coi risultati ottenuti nei dataset precedenti (figure 4.18, 4.19).
Ritroviamo invece un errore sull'acquisizione della profondità legato al colore dello pneumatico (figura 4.20, che a causa del colore ne altera il risultato. Comunque sia, il risultato di profondità risulta accettabile rispetto a quello di Polystyrene.

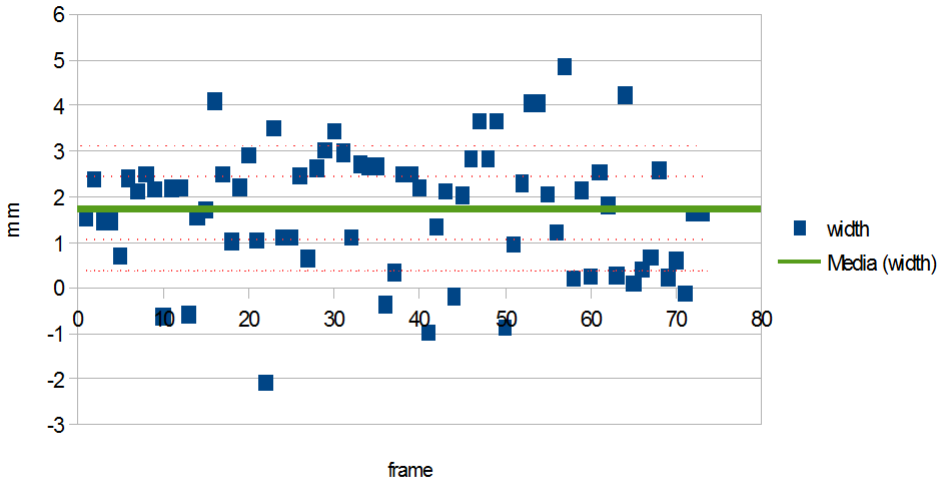


Figura 4.18: Kinect 1 - width

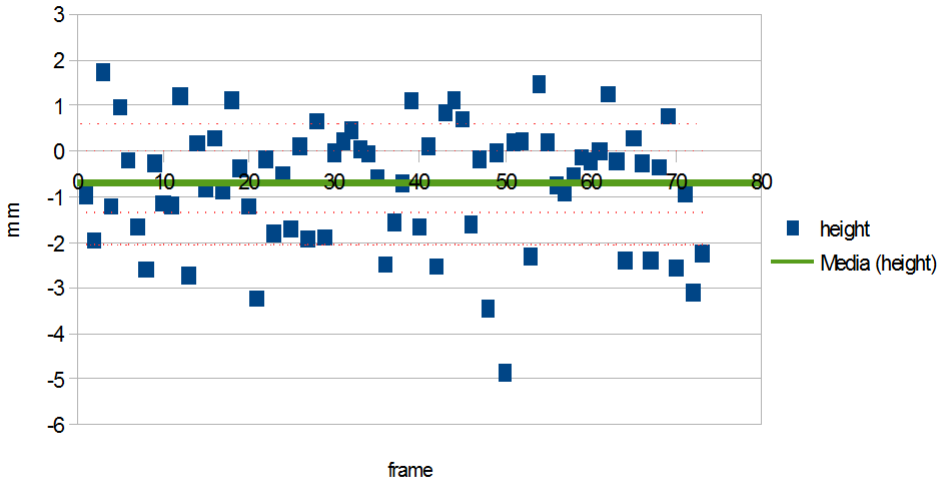


Figura 4.19: Kinect 1 - height

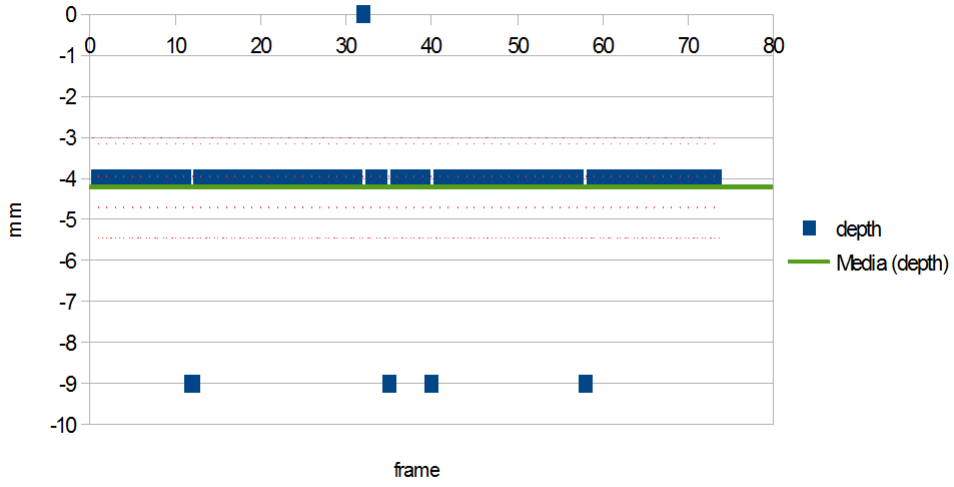


Figura 4.20: Kinect 1 - depth

Dato	Width	Height	Depth
min	-2,083	-4,872	-9
max	5,240 1,737 0		
media	1,724	-0,689	-4,216
mediana	2,056	-0,361	-4
deviazione standard	1,442	1,350	1,242

Tabella 4.5: Misure degli errori su Wheel con Kinect 2 in mm

4.3.2 Kinect 2

I risultati con questa camera risultano come già visto negli esempi precedenti più coerenti, ma si possono osservare delle variazioni più ampie su x e y che sono imputabili alla misurazione della forma non più poligonale come prima.
Riguardo la profondità, anche qui si nota un leggero deterioramento dei risultati che invece sono imputabili al colore stesso, che lo rendono un po' meno preciso di prima.

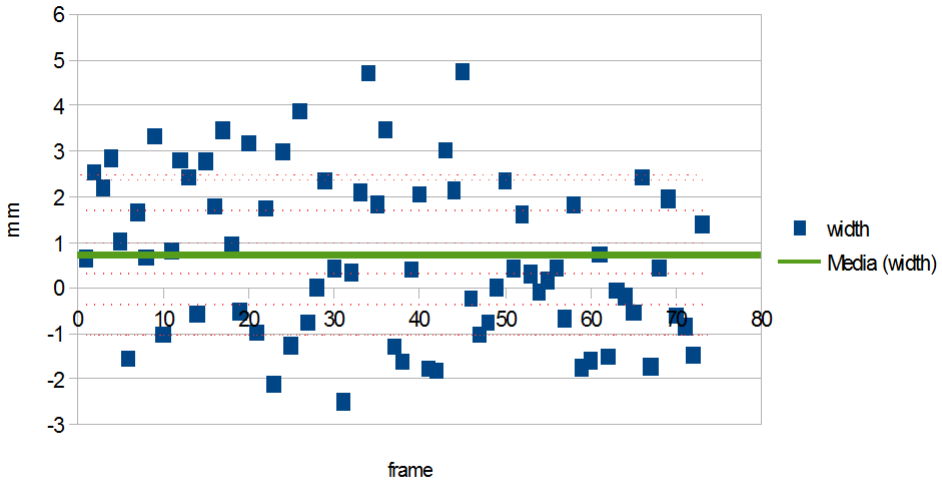


Figura 4.21: Kinect 2 - width

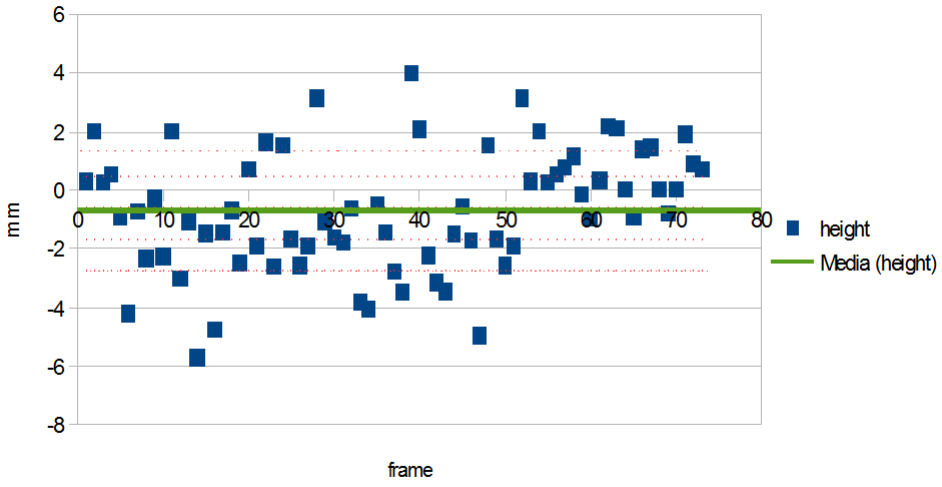


Figura 4.22: Kinect 2 -height

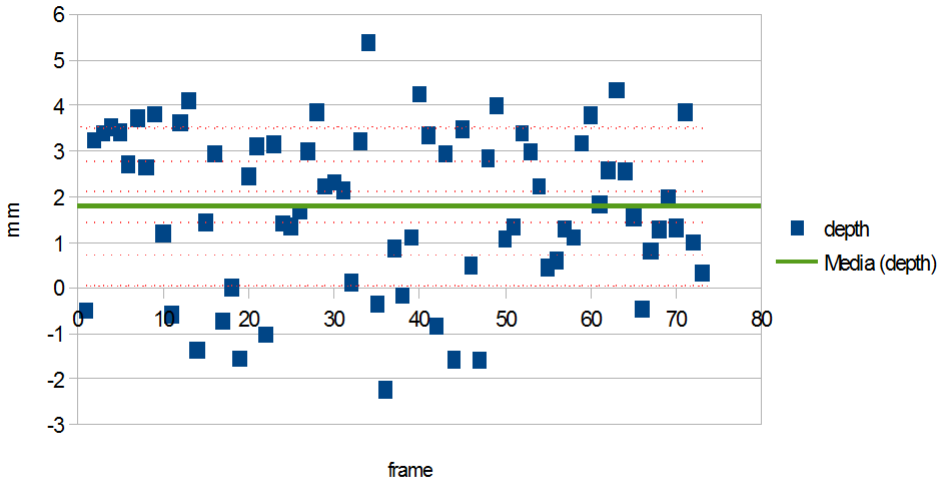


Figura 4.23: Kinect 2 - depth

Dato	Width	Height	Depth
min	-2,509	-5,716	-2,242
max	4,741	3,994	5,382
media	0,824	-0,734	1,967
mediana	0,654	-0,749	2,275
deviazione standard	1,744	2,045	1,821

Tabella 4.6: Misure degli errori su Wheel con Kinect 2 in mm

4.3.3 Comparazione

Osservando in figura 4.24 come le differenze in termini di risultati dei due Kinect sia minore rispetto a quelle viste fino ad ora, ma con una maggiore varianza rispetto ai precedenti.
I dati complessivamente risultano mediamente accetabili, ma la maggiore presenza di valutazioni molto fuori la media rende il Kinect 1 meno affidabile rispetto a una misurazione singola. Kinect 1 qui presenta un errore medio di 5 millimetri con una deviazione standard di 1.293, mentre per il Kinect 2 la media si abbassa a 3.757 mentre la deviazione standard è di 1.419.

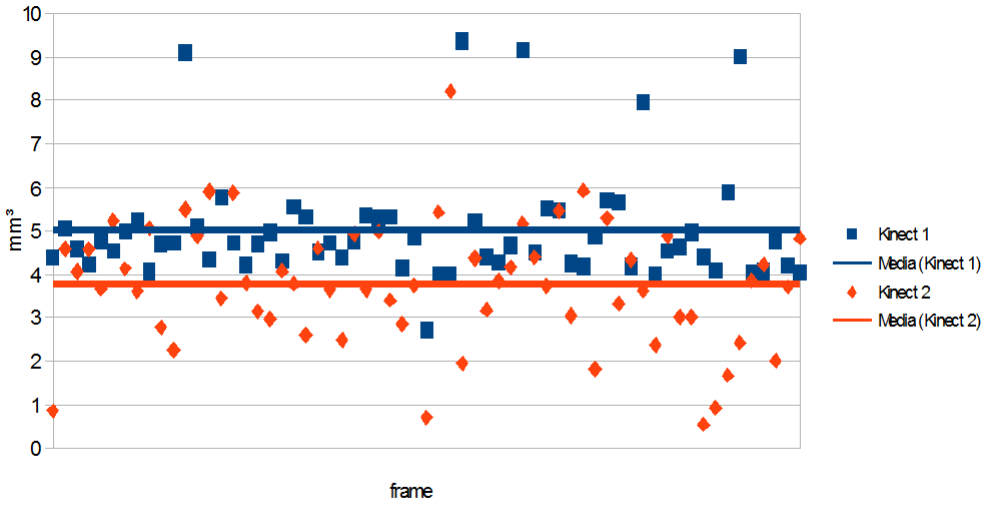


Figura 4.24: Wheel: comparazione tra Kinect 1 e Kinect 2

4.4 Tricycle

Figura 4.25: Dataset Tricycle

Questo dataset è senza dubbio il più complesso tra quelli esposti fino ad ora. Forma altamente irregolare, diversi materiali, componenti in plastica e numerosi spigoli.
Questo dataset presenta numerose difficoltà di misurazione della bounding box in quanto, a causa della irregolarità, è molto semplice che una variazione della misura incida pesantemente sul corrispettivo lato da calcolare e ancora di più sull'orientazione della bounding box stessa, che porta così a netti peggioramenti delle misurazioni. Il materiale plastico poi lo fa risultare leggermente lucido, che potrebbe portare ad errori di rifrazione in differenti condizioni di illuminazione.

4.4.1 Kinect 1

La camera presenta delle evidenti difficoltà con questo dataset. A partire già dalla creazione della depth image, alcune parti sottili e distanti vengono visualizzate con difficoltà o addirittura non riescono ad apparire, che come conseguenza portano il blob tool di Impact a riconoscere parti dell'oggetto come blob separati, potendo comportare grossi errori di valutazione.
Le valutazioni su larghezza e larghezza mostrano errori che arrivano oltre il centimetro, con rumori che oscillano di molto in modo tale da compromettere la misurazione.

Sulla profondità, nonostante la detection sia più stabile si trova comunque un errore maggiore di un centimetro, così come era già successo per Polystyrene, dovuto presumibilmente alla presenza del materiale plastico lucido.

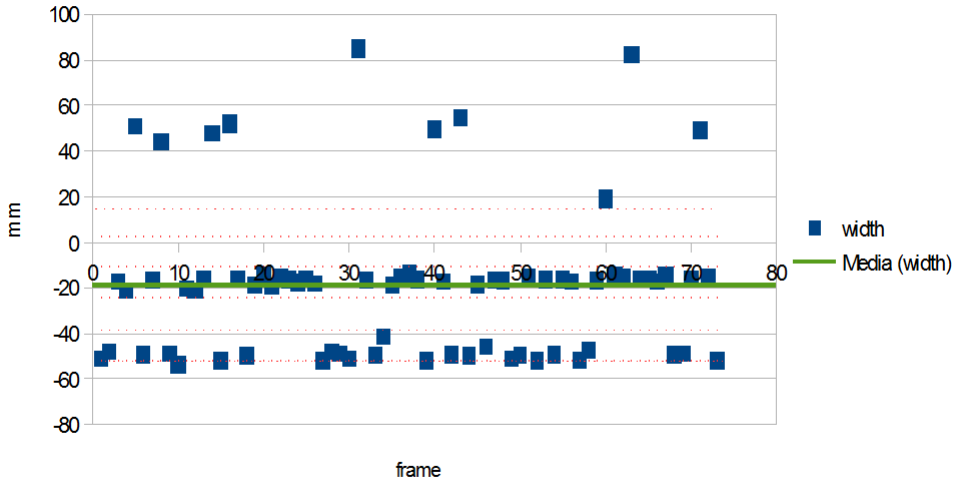


Figura 4.26: Kinect 1 - width

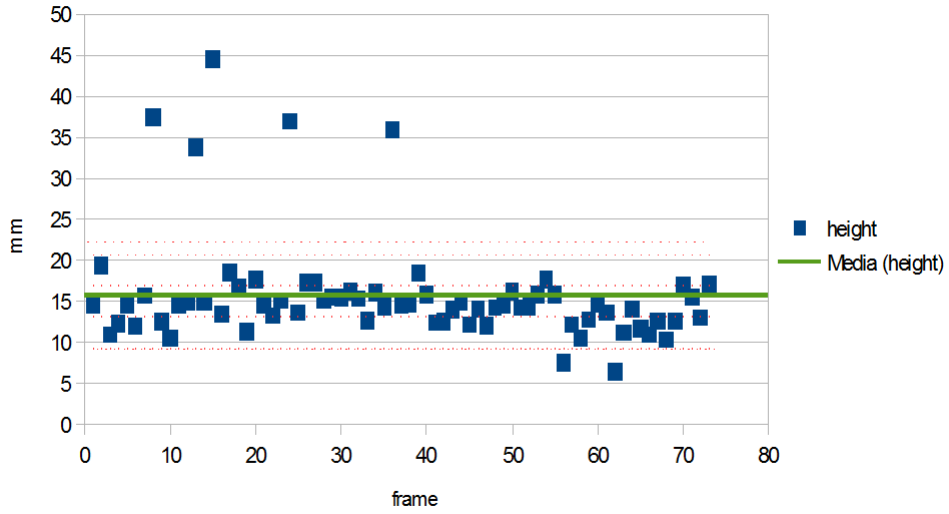


Figura 4.27: Kinect 1 - height

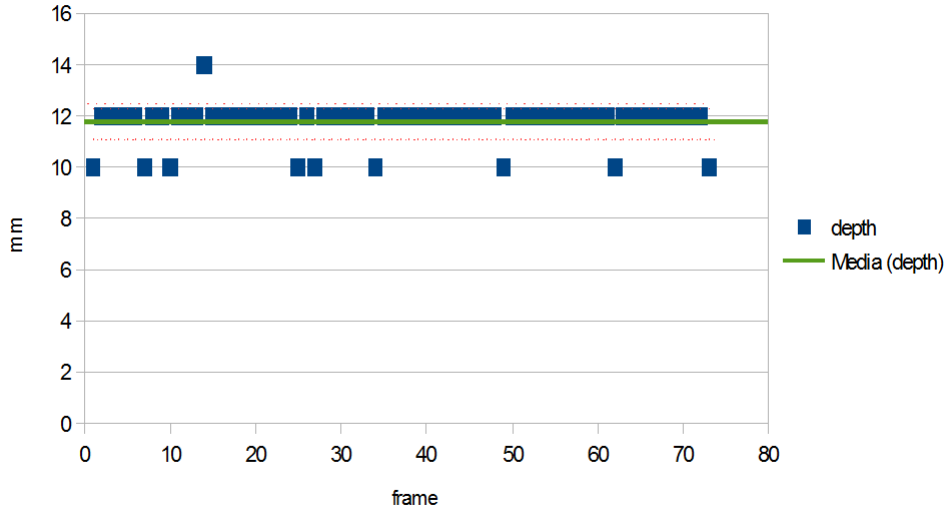


Figura 4.28: Kinect 1 - depth

Dato	Width	Height	Depth
min	-53,633	6,450	10
max	85,106	44,588	14
media	-18,753	15,707	11,784
mediana	-17,350	14,564 6 12	
deviazione standard	33,257	6,489	0,707

Tabella 4.7: Misure degli errori su Tricycle con Kinect 2 in mm

4.4.2 Kinect 2

Questo è il primo dataset dove il Kinect 2 presenta errori di misurazione superiori al centimetro. Come già spiegato inizialmente, le misurazione risultano più difficoltose e risulta facile ottenere una 'orientazione della bounding box non ottimale.
La profondità è la prima volta che non riesce ad essere molto affidabile, con oscillazioni tra i -5 e i +8 millimetri che non la rendono affidabile come negli esempi precedenti.

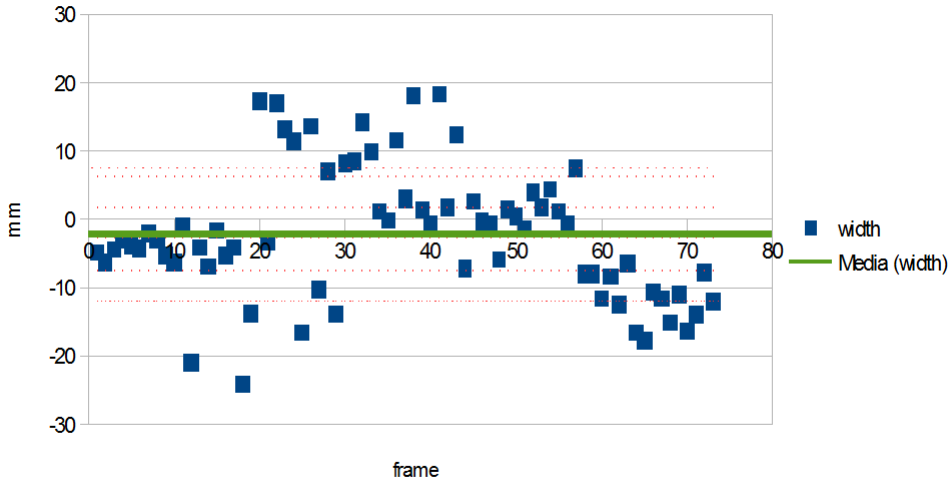


Figura 4.29: Kinect 2 - width

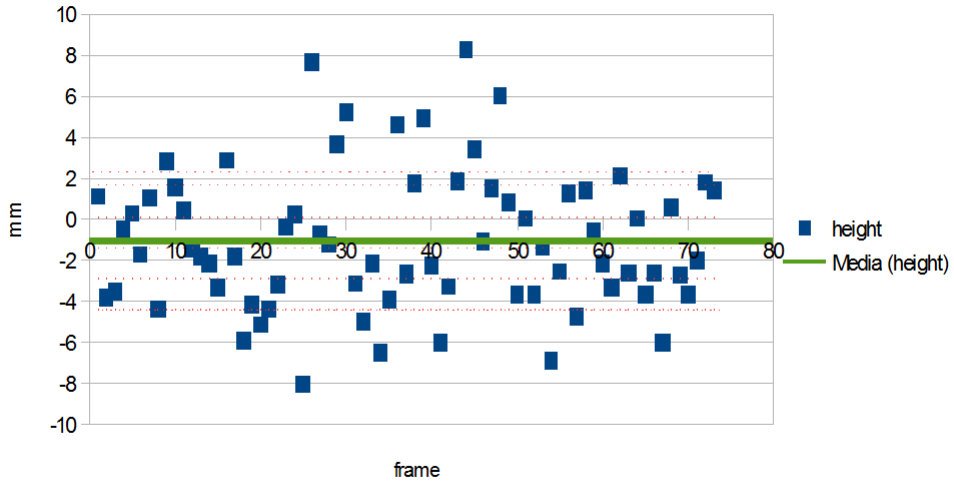


Figura 4.30: Kinect 2 - height

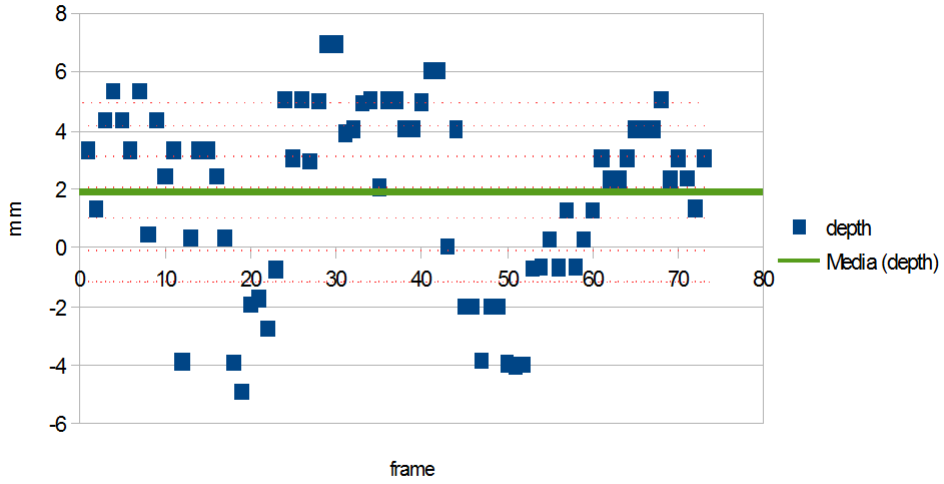


Figura 4.31: Kinect 2 - depth

Dato	Width	Height	Depth
min	-24,154	-8,026	-4,924
max	18,440	8,285	6,950
media	-2,264	-1,035	1,886
mediana	-3,182	-1,554	2,694
deviazione standard	9,726	3,395	3,062

Tabella 4.8: Misure degli errori su Tricycle con Kinect 2 in mm

4.4.3 Comparazione

Osservando in figura 4.32 si vedono chiaramente le conseguenze di quanto detto su questo dataset e i problemi di misurazione delle singole camere. Non si riesce più ad ottenere la precisione vista fino ad ora, e gli errori sul volume arrivano fino a $10mm^3$ per il Kinect 2 con un errore medio di 9,195 mm e una deviazione standard di 5,24 e addirittura oltre i $60mm^3$ per il Kinect 1 con un errore medio di 41,08 e una deviazione standard di 16,5.
Alla luce di questo, i risultati del Kinect 1 non sono da ritenersi affidabili per questo tipo di oggetti. Riguardo al Kinect 2 l'errore è più contenuto, ma sempre di gran lunga superiore a quanto visto, che rende comunque i dati misurati difficilmente accettabili in un contesto pratico.

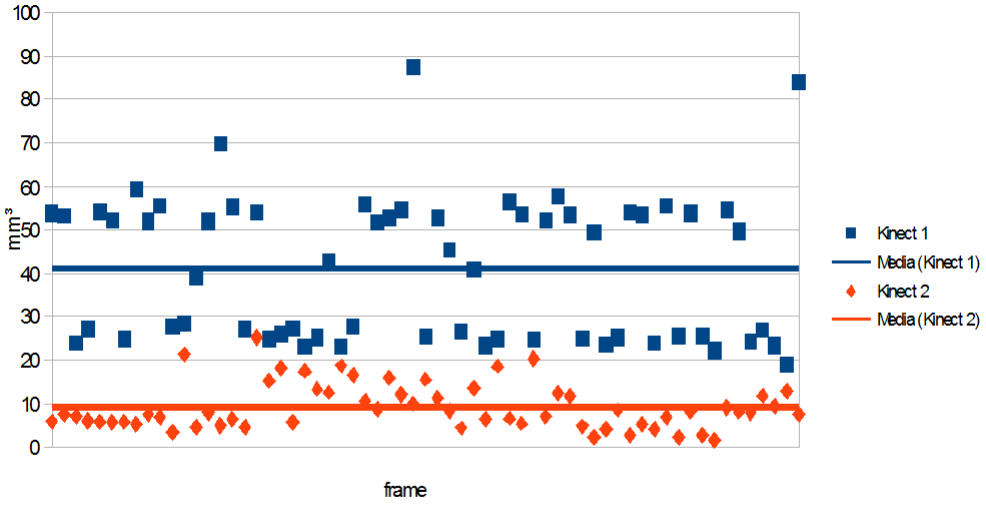


Figura 4.32: Tricycle: comparazione tra Kinect 1 e Kinect 2

Conclusioni e sviluppi futuri

Il sistema proposto mostra come le due Kinect camera mostrino le potenzialità per poter essere inserite all'interno di un contesto industriale, in cui sistemi di dimensionamento attualmente in commercio si attestano di una precisione entro i 5 millimetri per lato. Con opportune modifiche e ulteriori studi è possibile affinare i risultati in modo tale da ottenere valutazioni molto più robuste, in modo da rendere il rapporto tra qualità e costo ancora sia competitivo rispetto alle tecnologie di senisng 3D consolidate in ambito industriale.

Il confronto tra le due camere prese in esame mostra come benché usino tecnologie differenti di acquisizione (Pattern strutturato e TOF), più che essere considerate due modelli differenti, vadano visti come una versione potenziata dell'altro. La maggiore risoluzione e sensibilità del sensore del Kinect 2, lo porta ad essere più adatto ad applicazioni industriali rispetto alla prima versione, che vede il primo modello meno sensibile alle variazione di materiale e colore.
Si può affermare che mentre il primo modello rimanga orientato a un contesto ludico che debba riconoscere delle entità ben definite senza preoccuparsi troppo della precisione, il secondo modello sembra pensato più per essere un sistema general purpose che possa meglio approcciarsi a situazioni più specifiche rispondendo in maniera più precisa. Lo dimostra in particolar modo la maggiore robustezza ai colori e ai materiali.
Il Kinect 1 nasce per il gaming quindi il contesto per cui fu creato non si preoccupava di effettuare le distinzioni appena esposte, in quanto doveva limitarsi a riconoscere quanti giocatori ci fossero sullo schermo. Il Kinect 2 porta con sé quel bagaglio di esperienza accumulata dal primo modello e le risposte ai bisogni degli utenti che ne mostravano un interesse non più per il suo scopo originale, ma anche per scopi applicativi o di ricerca.

Riguardo i possibili sviluppi futuri, per aumentare la precisione del sistema sarebbe sufficiente, quando gli oggetti scorrono sul conveyor, acquisire più immagini riferite allo stesso oggetto e mediarne le misurazioni, in questo modo le oscillazioni risulterebbero attenuate e più coerenti col loro valor medio.
Un altro tipo di accorgimento che può portare al miglioramento della precisione è quello di, in

sede di acquisizione dell'immagine di profondità, applicare un processo come quello descritto in [MFI13] e [CLL12], che tramite l'utilizzo di filtri bilaterali migliorano le rilevazioni sui bordi e assegnano dei valori di profondità più uniformi. Queste zone rappresentano come visto un punto cruciale per le rilevazioni e la loro accuratezza.
Un'altra modifica che può essere applicata al sistema al fine di migliorarne le prestazioni è quella di prevedere l'inserimento di almeno un'altra camera da porre lateralmente, per delegare parti delle misurazioni alle singole camere e meglio correggere gli errori di misura fatta da ciascuna. Dato il basso costo, l'aggiunta di ulteriori Kinect può potenziare l'accuratezza e precisione delle informazioni, mantenendo comunque il costo di gran lunga inferiore rispetto a camere industriali.

Bibliografia

[AMCS96] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a Standard Default Color Space for the Internet. In *IS&T/SID Fourth Color Imaging Conference: Color Science, Systems and Applications*, pages 238–246, 1996.

[CLL12] Li Chen, H Lin, and Shutao Li. Depth image enhancement for kinect using region growing and bilateral filter. *Pattern Recognition (ICPR), 2012 21st ...*, (Icpr):3070–3073, 2012. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6460813.

[Dep] Depth Biomechanics. Science Behind 3D Vision. URL: http://www.depthbiomechanics.co.uk/?p=102.

[Gri10] Daniel A Griffith. Spatial Filtering. *Handbook of Applied Spatial Analysis*, pages 301–318, 2010. URL: http://dx.doi.org/10.1007/978-3-642-03647-7_16, doi:10.1007/978-3-642-03647-7_16.

[Hel] Tanner Helland. Seven Grayscale Algorithm. URL: http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/.

[Hun69] R. W. G. Hunt. The Reproduction of Colour, 1969.

[Li10] Dongguang Li. Analysis of moment invariants on image scaling and rotation. In *Innovations in Computing Sciences and Software Engineering*, pages 415–419, 2010. doi:10.1007/978-90-481-9112-3-70.

[MDM14] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. Characterizations of Noise in Kinect Depth Images: A Review. *IEEE Sensors Journal*, 14(6):1731–1740, 2014. URL: http://ieeexplore.ieee.

org/lpdocs/epic03/wrapper.htm?arnumber=6756961, doi:10.1109/JSEN.2014.2309987.

[MFI13] Takuya Matsuo, Norishige Fukushima, and Yutaka Ishibashi. Weighted Joint Bilateral Filter with Slope Depth Compensation Filter for Depth Map Refinement. *International Conference on Computer Vision Theory and Applications*, pages 300–309, 2013. URL: http://dblp.uni-trier.de/db/conf/visapp/visapp2013-2.html#MatsuoFI13$\delimiter"026E30F$nhttp://vc.cs.nthu.edu.tw/home/paper/codfiles/jrxu/201306110722/WeightedJointBilateralFilterwithSlopeDepthCompensationFilte pdf$\delimiter"026E30F$nhttp://nma.web.nitech.ac.jp/fukushi.

[Mic] Microsoft. Kinect for Windows Architecture. URL: https://msdn.microsoft.com/en-us/library/jj131023.aspx.

[MMK01] M Matthies, H Malchow, and J Kriz. Vision as computation, or: Does a computer vision system really assign meaning to images? *Informatik.Uni-Leipzig.De*, pages 1–12, 2001. URL: http://www.informatik.uni-leipzig.de/~schierwa/vision.pdf.

[Nat] National Ocean Service. LIDAR. URL: http://oceanservice.noaa.gov/facts/lidar.html.

[PD84] Thomas Porter and Tom Duff. Compositing digital images, 1984.

[Rob12] Robert A. Schowengerdt. *Remote Sensing: Models and Methods for Image Processing*, volume 95. 2012. URL: http://www.cabdirect.org/abstracts/19790652875.html$\delimiter"026E30F$nhttp://www.ncbi.nlm.nih.gov/pubmed/22115514, doi:10.1016/j.jenvman.2011.10.007.

[RW] ROS-Wiki. Kinect Calibration. URL: http://wiki.ros.org/kinect_calibration/technical.

[SCTM07] Hae Jong Seo, Priyam Chatterjee, Hiroyuki Takeda, and Peyman Milanfar. A comparison of some state of the art image denoising methods. *Conference Record - Asilomar Conference on Signals, Systems and Computers*, (4):518–522, 2007. doi:10.1109/ACSSC.2007.4487266.

[Shi] Shizuoka. TOF. URL: http://www.idl.rie.shizuoka.ac.jp/study/project/tof/index_e.html.

[SO14] John Sell and Patrick O'Connor. The xbox one system on a chip and kinect sensor. *IEEE Micro*, 34:44–53, 2014. doi:10.1109/MM.2014.9.

[Toma] Ing Federico Tombari. 3D Computer Vision. URL: http://didattica.arces.unibo.it/file.php/59/Elaborazione_dellImmagine_L-S/CVIP/12_3D_computer_vision.pdf.

[Tomb] Tombari Federico; Salti Samuele. Introduction to OpenCV. URL: http://didattica.arces.unibo.it/file.php/59/Elaborazione_dellImmagine_L-S/CVIP/Lab_Session_1_-_Opencv.pdf, doi:10.1111/0023-8333.50.s1.10.

[vHR10] Joviša Žunić, Kaoru Hirota, and Paul L. Rosin. A Hu moment invariant as a shape circularity measure. *Pattern Recognition*, 43:47–57, 2010. doi:10.1016/j.patcog.2009.06.017.

[ZXPZ12] Lei Zhang, Fei Xiang, Jiexin Pu, and Zhiyong Zhang. Application of improved HU moments in object recognition. In *IEEE International Conference on Automation and Logistics, ICAL*, pages 554–558, 2012. doi:10.1109/ICAL.2012.6308139.

Ringraziamenti

Questo lavoro arriva alla fine non solo di un percorso accademico, ma di un lungo viaggio. Un viaggio iniziato ormai parecchi anni fa, molto prima che arrivassi a Bologna, che è proseguito verso diverse tappe, fatte da molte persone, molti luoghi e molte sensazioni e pensieri. Il primo ringraziamento va sicuramente ai miei genitori, che mi hanno sempre sostenuto e che anche ma soprattutto mi hanno permesso di arrivare fin qui,ma anche di svolgere tutte le tappe che mi sono creato prima. Un grazie va sicuramente a Bologna e a quello che mi ha offerto, come persone, conoscenze e luoghi; un posto sereno dove poter stare. Agli amici e colleghi che ho trovato venendo qui, Orfeo (anche se si è dato alla fuga all'ultimo), Nino, Barbara, Ivan, Balta, Asja e le disavventure con Alitalia Laura e Peppe, poi quelli che come me han intrapreso il viaggio da Palermo come Vincenzo, Fulvio, Alberto (il collega più collegosissimo di tutti), Roberto, Gabriella che han contribuito a creare una bella colonia qui e che han condiviso il viaggio da ancora prima. Un altro grazie va agli amici fuori dal contesto universitario, che mi hanno ascoltato, sostenuto e a cui a random ho rotto le scatole con pensieri e pippe mentali, quindi grazie a Guido, Susanna, Arianna, Stefania, Cristobal e a quello che abbiamo condiviso insieme. E grazie (il bello si tiene sempre per il fiinale), a Claudia, che mi ha fatto ricordare e che ogni giorno mi ricorda quanto sia dolce oltre il pensiero di andare la voglia di restare.