

**Alma Mater Studiorum · Università di Bologna**

---

---

**Scuola di Ingegneria e Architettura**

**Dipartimento di Informatica - Scienza e Ingegneria -**

**Corso di Laurea Magistrale in Ingegneria Informatica**

**ANNO ACCADEMICO 2021/2022**

**- RELAZIONE FINALE -  
ANALISI DI BATTISTRADA  
TRAMITE L'UTILIZZO DEL  
PROFILOMETRO LASER GOCATOR**

**ESAME DI COMPUTER VISION AND IMAGE PROCESSING M**

**AUTORE:  
Gabriele Tornatore**



# Indice

<b>Indice</b>	<b>ii</b>
<b>Elenco delle figure</b>	<b>v</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Gocator</b>	<b>3</b>
2.1 Settaggi . . . . .	5
2.1.1 Area Attiva . . . . .	5
2.1.2 Trasformazioni . . . . .	6
2.1.3 Triggers . . . . .	6
2.1.4 Esposizione . . . . .	7
2.1.5 Filtri . . . . .	7
2.2 Metodi di scansione . . . . .	8
2.2.1 Acquisizione superficie . . . . .	8
2.2.2 Acquisizione intensità . . . . .	9
2.2.3 Metodo scelto . . . . .	9
<b>3 Analisi</b>	<b>10</b>
3.1 Prodotti . . . . .	10
3.2 Librerie di supporto . . . . .	12
3.3 Algoritmi di analisi . . . . .	12
3.3.1 Battistrada di tipo 1A/1B . . . . .	12
3.3.2 Battistrada di tipo 2 . . . . .	22
3.3.3 Performance . . . . .	29
<b>4 Applicazione desktop</b>	<b>30</b>
4.1 Sviluppo . . . . .	30
4.2 Librerie di supporto . . . . .	31

4.3	Interfaccia grafica . . . . .	31
<b>5</b>	<b>Conclusioni</b>	<b>36</b>
5.1	Risultati ottenuti . . . . .	36
5.2	Conclusione e lavori futuri . . . . .	37
	<b>Bibliografia</b>	<b>38</b>

# Elenco delle figure

2.1	Funzionamento di un profilometro laser.	3
2.2	Caratteristiche visive del Gocator.	4
2.3	Il box di colore giallo è l'area attiva. La linea rossa indica l'inclinazione del laser sul conveyor.	5
2.4	Confronto con e senza spaziatura uniforme.	9
3.1	Battistrada di tipo 1A	10
3.2	Battistrada di tipo 1B	11
3.3	Battistrada di tipo 2	11
3.4	Contorni dei MacroBlob, in questo ne sono presenti 6.	13
3.5	Contorni dei Blob, in questo caso ne sono presenti 16.	13
3.6	Contorni dei MacroBlob, in questo caso ne sono presenti 9.	13
3.7	Point cloud non elaborata del battistrada di tipo 1A.	14
3.8	Point cloud del battistrada di tipo 1A dopo aver applicato il filtro statistico.	15
3.9	Point cloud del battistrada di tipo 1A dopo aver applicato la segmentazione planare.	16
3.10	Point cloud del battistrada di tipo 1A dopo aver applicato l'operazione di proiezione.	16
3.11	Point cloud del battistrada di tipo 1A dopo aver applicato l'operazione di traslazione.	17
3.12	Immagine planare del battistrada di tipo 1A.	17
3.13	Immagine del battistrada di tipo 1A dopo aver applicato la prima trasformazione morfologica (Closing).	18
3.14	Immagine del battistrada di tipo 1A dopo aver applicato la seconda trasformazione morfologica (Closing).	18
3.15	Immagine del battistrada di tipo 1A con i contorni dei MacroBlob evidenziati.	19

3.16 Immagine del battistrada di tipo 1A con le bounding box dei MacroBlob evidenziate.	19
3.17 Immagine del battistrada di tipo 1A con i contorni dei Blob evidenziati.	19
3.18 Immagine del battistrada di tipo 1A con le bounding box dei Blob evidenziate.	19
3.19 Point cloud non elaborata del battistrada di tipo 1B.	20
3.20 Point cloud del battistrada di tipo 1B dopo aver applicato il filtro statistico.	20
3.21 Point cloud del battistrada di tipo 1B dopo aver applicato la segmentazione planare.	20
3.22 Point cloud del battistrada di tipo 1B dopo aver applicato l'operazione di proiezione.	20
3.23 Point cloud del battistrada di tipo 1B dopo aver applicato l'operazione di traslazione.	21
3.24 Immagine planare del battistrada di tipo 1B.	21
3.25 Immagine del battistrada di tipo 1B dopo aver applicato la trasformazione morfologica (Closing).	21
3.26 Immagine del battistrada di tipo 1B con le bounding box dei MacroBlob evidenziate.	22
3.27 Point cloud non elaborata del battistrada di tipo 2.	22
3.28 Funzione originale.	23
3.29 Funzione smussata.	23
3.30 Massimi e minimi della funzione smussata.	24
3.31 Massimi (da destra) della funzione originale.	25
3.32 Minimi della funzione originale.	25
3.33 Esempio del calcolo di uno dei massimi (da sinistra) della funzione smussata.	26
3.34 Massimi (da sinistra) della funzione originale.	27
3.35 Confronto tra la funzione originale e la funzione smussata. I punti critici segnati sono il risultato di un filtro sulla distanza maggiore di 0.5 millimetri.	27
3.36 Point cloud del battistrada di tipo 2, con vista sul piano <i>XY</i> .	28
3.37 (a) Point cloud contenente i punti di massimo da sinistra e di massimo da destra. (b) Point cloud contenente i punti di minimo, di massimo da sinistra e di massimo da destra.	28

4.1	Interfaccia grafica con la descrizione dei pulsanti. . . . .	31
4.2	Interfaccia grafica con la point cloud filtrata visualizzata a video. . .	33
4.3	Interfaccia grafica con, visualizzata a video, la point cloud del bat- tistrada di tipo 1A e il risultato dell'analisi. . . . .	34
4.4	Interfaccia grafica con, visualizzata a video, la point cloud del bat- tistrada di tipo 2. . . . .	34
4.5	Interfaccia grafica con, visualizzata a video, la point cloud con i punti di profondità delle scanalature minima e massima. . . . .	35
4.6	Interfaccia grafica con, visualizzata a video, la point cloud filtrata. .	35
5.1	Il Gocator in azione durante una scansione. . . . .	37

# Capitolo 1

## Introduzione

Viviamo in un mondo 3D con processi di produzione che richiedono un'ispezione delle parti ad alta velocità e altamente accurata. Le soluzioni di visione artificiale progettate per supportare la produzione automatizzata possono essere complicate da progettare, configurare e installare, e spesso richiedono personale specializzato con un background ingegneristico.

Le soluzioni 2D progettate per risolvere i problemi di misurazione 3D sono complicate, costose e spesso non hanno successo. Il 2D semplicemente non può compensare la variazione di altezza senza una sorta di compromesso in termini di costi, qualità o affidabilità.

Sebbene le soluzioni 3D siano tradizionalmente considerate costose o complicate da configurare, la rivoluzionaria famiglia di prodotti Gocator di LMI semplifica la misurazione 3D per l'automazione in fabbrica.

LMI Technologies Inc. (LMI) è leader mondiale nella tecnologia dei sensori smart 3D per applicazioni di ispezione e di misura in ambito industriale ma non solo. LMI sviluppa costantemente nuove soluzioni 3D che consentano agli utenti finali di sfruttare gli enormi vantaggi della tecnologia laser applicata al controllo di qualità, all'ottimizzazione dei materiali per ridurre gli sprechi oppure all'automazione nelle moderne realtà industriali, al fine di migliorarne la produttività.

Il progetto ha l'obiettivo di interfacciarsi con il Gocator e di creare una demo per *PC*, comprensiva di *GUI*, che permetta l'elaborazione della *point cloud*, effettuando misurazioni sui prodotti presi in esame.

La relazione è articolata come segue:

- nel **capitolo 2** viene presentato il gocator e il settaggio utilizzato;
- nel **capitolo 3** vengono descritti i prodotti presi in analisi e gli algoritmi utilizzati;
- nel **capitolo 4** viene presentato lo sviluppo dell'applicazione desktop e le sue funzionalità;
- il **capitolo 5** conclude la relazione presentando i risultati ottenuti, gli obiettivi raggiunti ed eventuali problematiche che potrebbero essere risolte in futuro;

Il codice e l'applicazione di tutto il progetto possono essere reperiti nella seguente repository git: <https://github.com/it9tst/computer-vision>

# Capitolo 2

## Gocator

Il Gocator è un sistema di visione 3D che cattura la forma e la posizione di un oggetto, indipendentemente dal colore della superficie e senza opportune calibrazioni.

Il design semplice e flessibile del Gocator consente alle fabbriche di ridurre i costi e massimizzare la redditività migliorando l'efficienza nella convalida dei prodotti. Esso riesce a supportare le richieste di misure 3D accurate, dalle aziende, in settori quali: fusione di alluminio, gomma e produzione di pneumatici, lavorazione del legno, assemblaggio di automobili, food & packaging, medicale ecc..

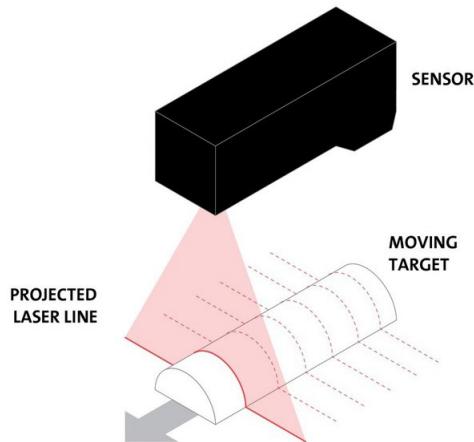


Figura 2.1: Funzionamento di un profilometro laser.

Il modello che è stato utilizzato per l'esecuzione del progetto è il Gocator 2350, facente parte della serie 2300, disponibile in cinque modelli standard, che hanno come caratteristiche comuni:

- **Risoluzione megapixel:** 1280 punti per risoluzione del profilo;
- **Campo visivo:** fino a 1260 mm;
- **Campo di misura:** fino a 800 mm;

In particolare, il modello 2350 presenta le seguenti caratteristiche:

- **Risoluzione dell'asse X:** 0.150 mm;
- **Risoluzione dell'asse Z:** 0.019 mm;
- **Campo visivo (FOV):** 158 mm - 365 mm;
- **Distanza minima (CD):** 300 mm;
- **Campo di misura (MR):** 400 mm;

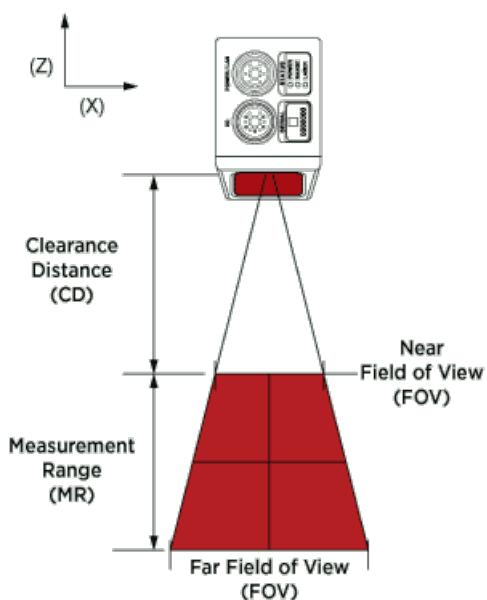


Figura 2.2: Caratteristiche visive del Gocator.

## 2.1 Settaggi

Il collegamento tra il sensore e il PC avviene tramite cavo Ethernet, quindi è importante conoscere l'indirizzo IP del Gocator. Una volta identificato l'indirizzo IP, basta inserirlo nel browser web per interagire con l'interfaccia web che LMI mette a disposizione dell'utente.

Per i nostri propositi sono stati modificati alcuni settaggi che rimarranno immutati anche durante le scansioni, mentre i settaggi che potrebbe essere utile modificare in *runtime*, si troveranno anche nella nostra applicazione.

### 2.1.1 Area Attiva

L'area attiva si riferisce alla regione all'interno del massimo campo visivo del sensore, utilizzata per la profilatura laser (figura 2.3). Riducendo l'area attiva, il sensore può funzionare a velocità più elevate.

Per i nostri scopi, è stata scelta una dimensione di area attiva di:

- **X Field of View:** 300 mm;
- **Measurement Range:** 250 mm;

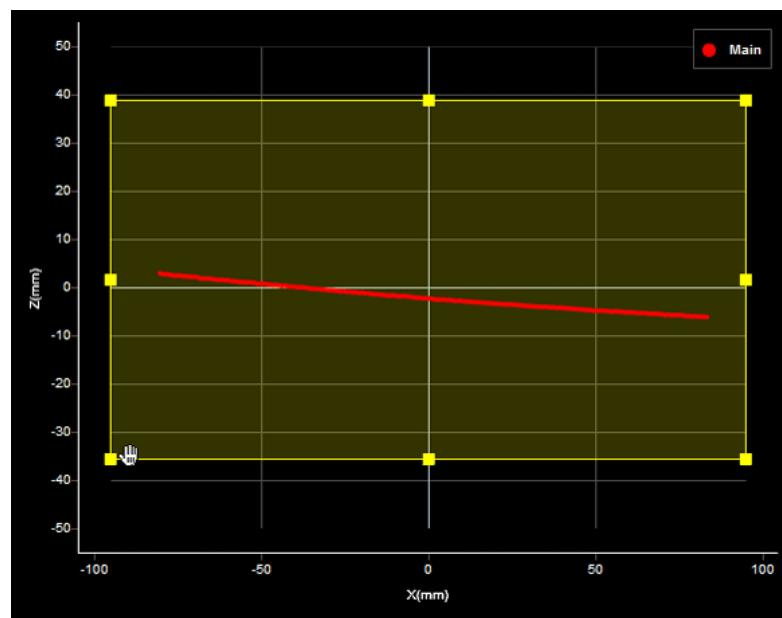


Figura 2.3: Il box di colore giallo è l'area attiva. La linea rossa indica l'inclinazione del laser sul conveyor.

## 2.1.2 Trasformazioni

Le impostazioni di trasformazione determinano come i profili vengono convertiti dalle coordinate del sensore alle coordinate del sistema.

In pratica, i seguenti settaggi servono per allineare il sensore:

- **X Offset:** Specifica lo spostamento lungo l'asse X.
- **Y Offset:** Specifica lo spostamento lungo l'asse Y.
- **Z Offset:** Specifica lo spostamento lungo l'asse Z.
- **Angle X:** Specifica l'inclinazione intorno l'asse X.
- **Angle Y:** Specifica l'inclinazione intorno l'asse Y.
- **Angle Z:** Specifica l'inclinazione intorno l'asse Z.

## 2.1.3 Triggers

Il trigger è un evento che fa sì che il sensore acquisisca una singola immagine.

Esso può essere attivato da una delle sorgenti seguenti:

- **Time:** Il sensore ha un clock interno che può essere utilizzato per generare trigger di frequenza fissa.
- **Encoder:** Il trigger viene fornito da un encoder in risposta al movimento.
- **External Input:** Un ingresso digitale può fornire trigger di risposta a eventi esterni.
- **Software** Il trigger è inviato da un comando di rete.

Non avendo a disposizione un encoder, è stata utilizzata la sorgente *time* con il massimo *frame rate* disponibile (183.413 Hz) che è stato sincronizzato con la velocità del conveyor.

## 2.1.4 Esposizione

L'esposizione determina la durata del tempo della fotocamera e del laser. Esposizioni più lunghe possono essere utili per rilevare i segnali laser su superfici scure o distanti, ma aumentando il tempo di esposizione si riduce la velocità massima.

Il Gocator fornisce tre modalità di esposizione per scansionare diversi tipi di superfici: *Single*, *Dynamic*, *Multiple*.

Per i battistrada è stata utilizzata l'esposizione singola con un valore di  $2000 \mu\text{s}$ , perchè la superficie è uniforme ed è la stessa per tutte le parti.

Inoltre, la corretta regolazione dell'esposizione dipende dalle proprietà riflettenti del materiale in esame, per questo è stato scelto di rendere questo parametro editabile direttamente nel nostro programma.

## 2.1.5 Filtri

Il software di LMI permette di utilizzare dei filtri per post-elaborare i dati di scansione, in modo da pulirli e rimuovere il rumore.

La scelta è ricaduta nel disattivarli tutti per avere delle *point cloud* quanto più "pulite" possibili.

## 2.2 Metodi di scansione

Il Gocator supporta i seguenti metodi di scansione:

- Acquisizione superficie;
- Acquisizione intensità;

### 2.2.1 Acquisizione superficie

Con questo metodo, il sensore genera una superficie (*point cloud*) combinando una serie di profili raccolti lungo il conveyor.

La generazione delle *point cloud* può avvenire applicando diverse modalità:

- **Continuous:** il sensore genera superfici solo quando vengono rilevate;
- **Fixed length:** il sensore genera superfici di lunghezza fissa (in mm) utilizzando un valore settato da noi;
- **Variable length:** il sensore genera superfici di lunghezza variabile;
- **Rotational:** usata per eseguire scansioni di oggetti circolari;

Per i nostri scopi, è stata scelta la modalità **continuous**.

Per quanto riguarda la generazione dei dati delle *point cloud*, il Gocator permette di produrle in due formati:

- Con dati **grezzi** (non elaborati);
- Con **spaziatura uniforme** (ricampionati);

#### Spaziatura uniforme

Con questo formato, gli intervalli vengono ricampionati con una spaziatura uniforme lungo l'asse *X*, che quindi viene divisa in "contenitori" di dimensioni fisse.

I punti del profilo che cadono nello stesso "contenitore", vengono combinati in un unico valore (*Z*).

Un intervallo più ampio crea profili con una risoluzione  $X$  inferiore, riduce l'utilizzo della CPU e potenzialmente aumenta la frequenza dei fotogrammi. Di contro, riduce la velocità di output dei dati.

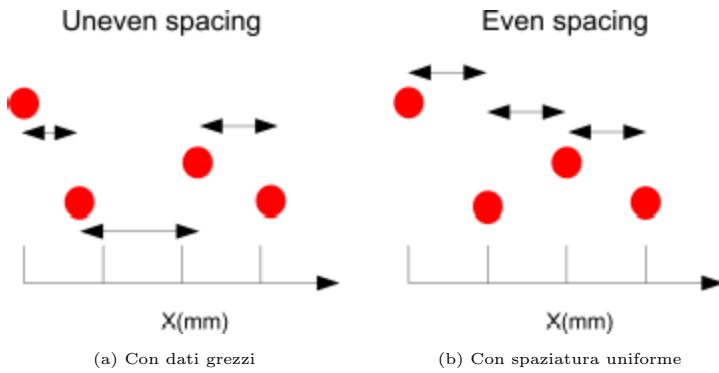


Figura 2.4: Confronto con e senza spaziatura uniforme.

## 2.2.2 Acquisizione intensità

Con questo metodo, verrà prodotto un valore di intensità (0-255) per ciascun punto scansionato dal laser. In pratica viene misurata la quantità di luce riflessa dall'oggetto.

## 2.2.3 Metodo scelto

Per i nostri scopi, ovviamente, è stato scelto il metodo dell'acquisizione della superficie con una spaziatura uniforme di 0.15 mm.

# Capitolo 3

## Analisi

Il capitolo seguente sarà incentrato sulla presentazione dei prodotti, seguita da una descrizione generale degli algoritmi utilizzati per effettuare le rispettive analisi.

### 3.1 Prodotti

Per gli esperimenti sono stati usati tre tipi di battistrada diversi. I primi due (figura 3.1 e figura 3.2) sono molto simili ma differenziano per complessità e numero di blocchi (*blob*), mentre il terzo (figura 3.3) è molto differente perché presenta delle concavità irregolari e delle parti consumate.

Da questo momento in poi, per facilitarne la comprensione, ogni battistrada avrà un suo nome specifico.



Figura 3.1: Battistrada di tipo 1A



Figura 3.2: Battistrada di tipo 1B



Figura 3.3: Battistrada di tipo 2

## 3.2 Librerie di supporto

Gli algoritmi sono stati sviluppati in *C++*, utilizzando *Visual Studio*, un ambiente di sviluppo integrato sviluppato da *Microsoft*.

Le librerie utilizzate sono le seguenti:

- *Gocator Software Development Kit (GoSDK)*, versione 6.1.32.12, è una libreria open source che può essere utilizzata per accedere e controllare a livello di codice il Gocator;
- *OpenCV*, versione 4.5.4, è una libreria software multipiattaforma nell'ambito della visione artificiale in tempo reale;
- *Point Cloud Library*, versione 1.12.1, è una libreria open source di algoritmi per le attività di elaborazione delle nuvole di punti e l'elaborazione della geometria 3D;
- *GNU Scientific Library*, versione 2.7, è una libreria software per calcoli numerici in matematica e scienze applicate;

## 3.3 Algoritmi di analisi

### 3.3.1 Battistrada di tipo 1A/1B

Dall'analisi di questo tipo di battistrada si è deciso di estrarre le seguenti features:

- Distanza minima tra ogni *MacroBlob* adiacente (**1A/1B**).
- Area, perimetro e dimensione della *bounding box* di ogni *MacroBlob* (**1A/1B**).
- Area, perimetro e dimensione della *bounding box* di ogni *Blob* (**1A**).
- Per ogni *MacroBlob*, la distanza minima tra ogni *Blob* adiacente (**1A**).

Successivamente, per fini statistici, si è calcolato il minimo, il massimo e la media dei valori trovati in elenco.

Prendendo come esempio il battistrada di tipo 1A, nella figura 3.4 sono evidenziati i *MacroBlob*, mentre nella figura 3.5 i *Blob*.

Nel caso del battistrada di tipo 1B (figura 3.6), parliamo solo di *MacroBlob* perché non sono presenti blocchi interni.

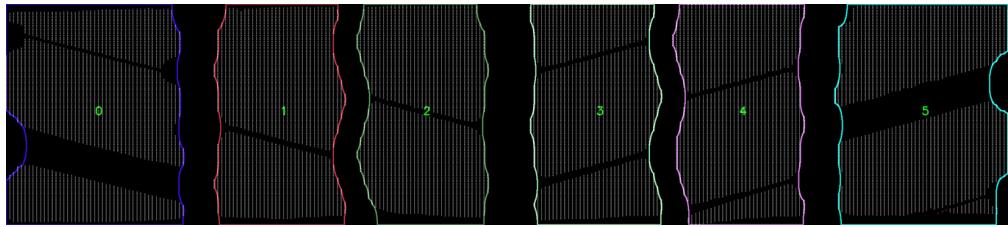


Figura 3.4: Contorni dei MacroBlob, in questo ne sono presenti 6.

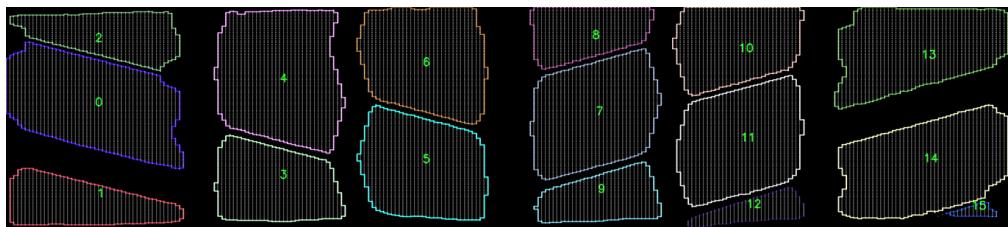


Figura 3.5: Contorni dei Blob, in questo caso ne sono presenti 16.

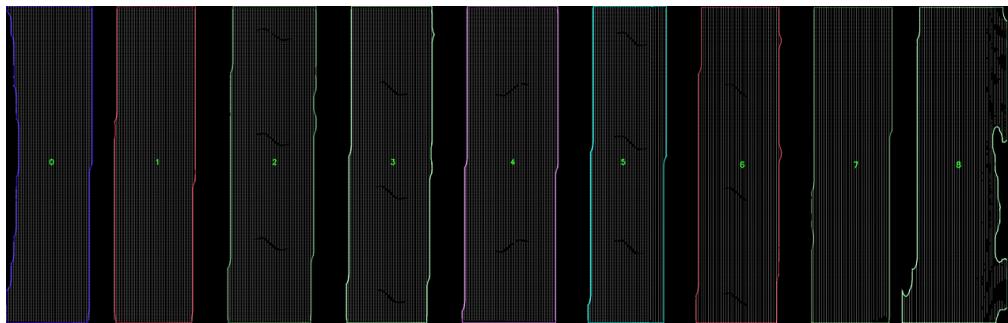


Figura 3.6: Contorni dei MacroBlob, in questo caso ne sono presenti 9.

In seguito alla scansione dell'oggetto, da parte del profilometro laser, ciò che abbiamo non sono altro che delle coordinate  $x$ ,  $y$  e  $z$  dove sono definiti tutti i punti presenti nella *point cloud*, dove il punto  $(0,0,0)$  coincide con il centro del piano della stessa (figura 3.7).

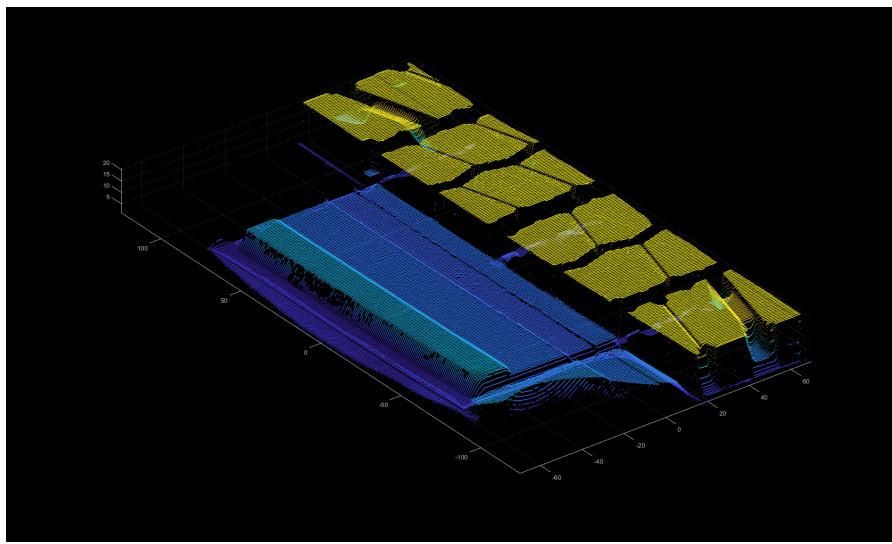


Figura 3.7: Point cloud non elaborata del battistrada di tipo 1A.

La prima operazione effettuata è stata quella di applicare un filtro per rimuovere eventuali misurazioni rumorose, come ad esempio valori anomali. Infatti, le scansioni laser, generando *point cloud* con una densità di punti variabile, effettuano spesso errori di misurazione che riportano valori anomali.

Per evitare il più possibile questi errori, è stata applicata un'analisi statistica sull'intorno di ciascun punto, filtrando quelli che non soddisfano un determinato criterio. Assumendo che la distribuzione della distanza tra un punto ed un altro (per tutti i punti della *point cloud*) risulta approssimabile con una gaussiana, verifico che la distanza media tra un determinato punto e tutti i suoi vicini sia al di fuori dell'intervallo definito dalla media delle distanze globali e dalla deviazione standard. Questi ultimi possono essere considerati valori anomali e quindi rimossi.

La figura 3.8 mostra gli effetti dell'analisi e della rimozione dei valori anomali.

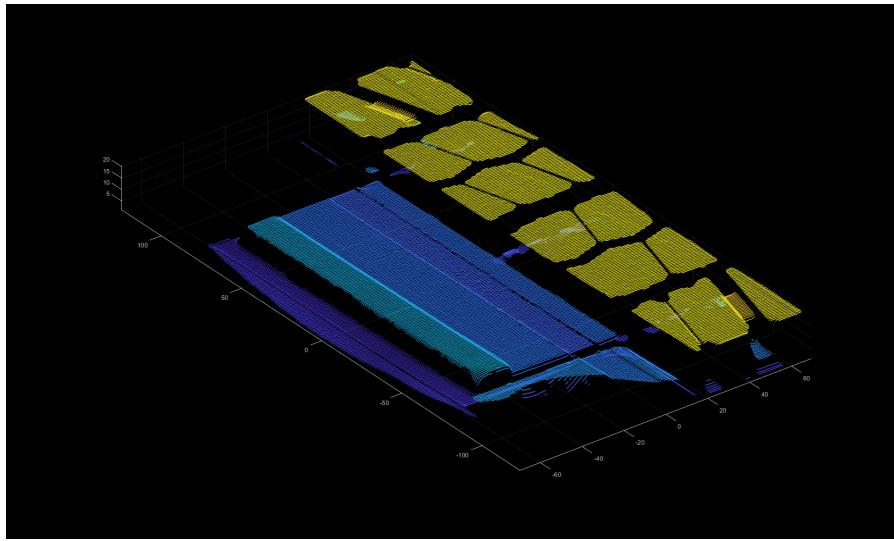


Figura 3.8: Point cloud del battistrada di tipo 1A dopo aver applicato il filtro statistico.

Il nuovo obiettivo è quello di mascherare solo la parte dove si trovano i blocchi e i macro blocchi (punti di colore giallo della figura 3.8). Per fare ciò, è stata eseguita una semplice segmentazione piana di un insieme di punti, ovvero sono stati trovati tutti i punti all'interno della nostra *point cloud* che supportano un modello piano (figura 3.9).

Dopo aver impostato il modello (*pcl::SACMODEL\_PLANE*) e il tipo di metodo (*pcl::SAC\_RANSAC*), è stata specificata una distanza di soglia che determina quanto un punto deve essere vicino al modello piano per essere considerato facente parte del modello stesso.

In questo caso è stata effettuata una prima distinzione tra il battistrada di tipo 1A e quello di tipo 1B, perchè sono state impostate diverse distanze di soglia per ognuno.

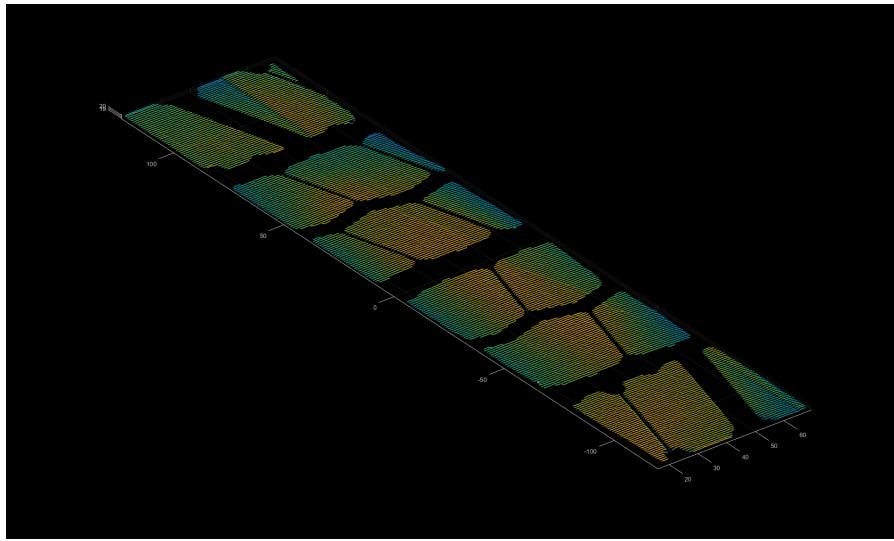


Figura 3.9: Point cloud del battistrada di tipo 1A dopo aver applicato la segmentazione planare.

Le ultime due operazioni sulla *point cloud* sono le più semplici. Prima, tutti i punti sono stati proiettati sul piano dato dalla sua equazione  $z = 0$  (figura 3.10), poi è stata effettuata un'operazione di traslazione per tutti i punti, portando l'origine degli assi nella stessa della notazione matriciale, in alto a sinistra (figura 3.11).

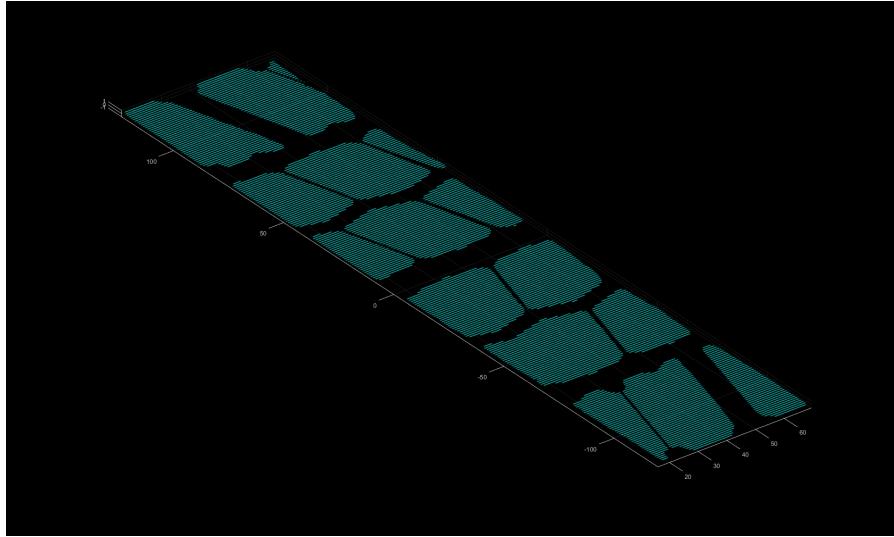


Figura 3.10: Point cloud del battistrada di tipo 1A dopo aver applicato l'operazione di proiezione.

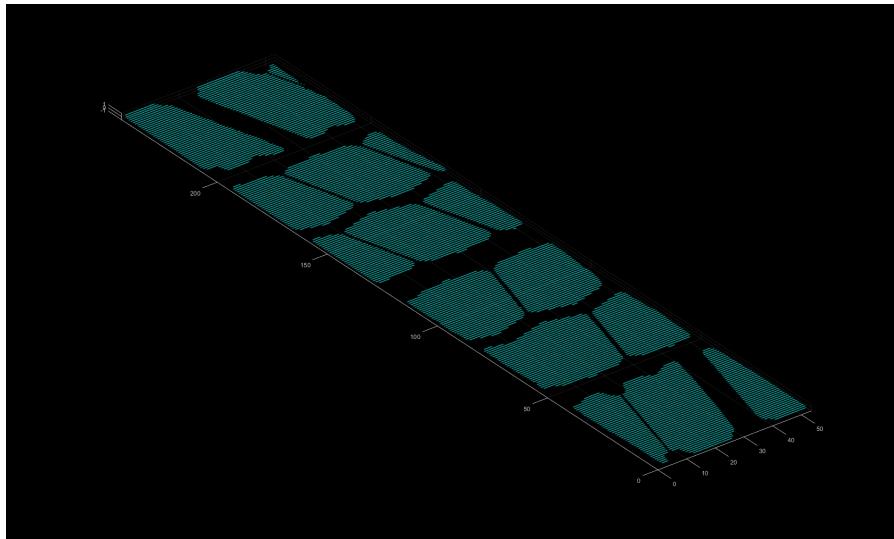


Figura 3.11: Point cloud del battistrada di tipo 1A dopo aver applicato l'operazione di traslazione.

A questo punto, per trovare i contorni e le misure dei vari blocchi, è stata utilizzata la libreria *opencv* e quindi è stato necessario creare un immagine dove i punti della *point cloud* sono stati colorati di bianco e tutto il resto di nero.

Inoltre è stato applicato un moltiplicatore x10 per fare in modo che 1 pixel dell'immagine corrispondesse, nel mondo reale, ad 1  $\mu\text{m}$ .

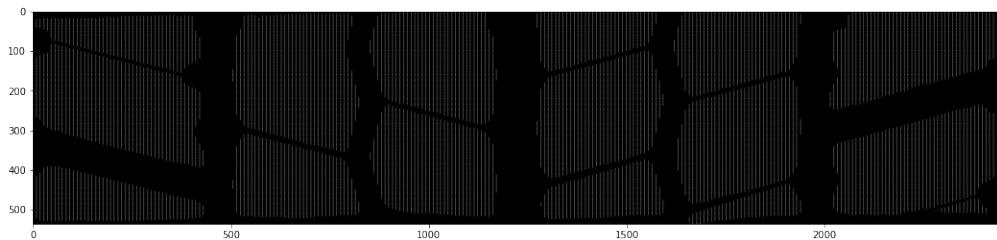


Figura 3.12: Immagine planare del battistrada di tipo 1A.

Per chiudere i piccoli fori presenti nei *Blob* della figura 3.12, sono state usate delle trasformazioni morfologiche, in particolare di tipo *closing*.

Le trasformazioni morfologiche sono delle operazioni che si basano sulla forma dell'immagine. L'operazione di tipo *closing* non è altro che il susseguirsi di due operatori: prima viene applicata una *dilatazione* e poi un *erosione*.

La figura 3.13 è il risultato della prima *closing* applicata per trovare i *Blob*, mentre la figura 3.14 è il risultato della *closing* applicata all'immagine precedente per

trovare i *MacroBlob*.

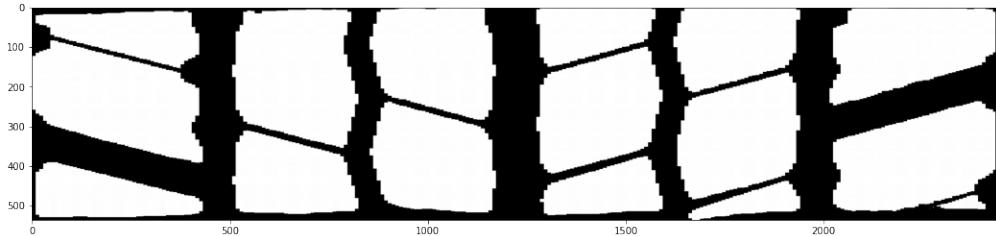


Figura 3.13: Immagine del battistrada di tipo 1A dopo aver applicato la prima trasformazione morfologica (Closing).

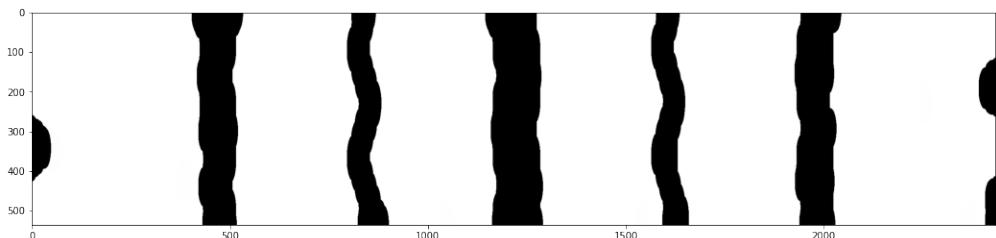


Figura 3.14: Immagine del battistrada di tipo 1A dopo aver applicato la seconda trasformazione morfologica (Closing).

Quindi, per trovare i contorni, è stata utilizzata la funzione *findContours()* di *opencv* che accetta come argomenti: l'immagine sorgente in scala di grigio, la modalità di recupero (*cv::RETR\_EXTERNAL*) e il metodo di approssimazione (*cv::CHAIN\_APPROX\_NONE*) del contorno (figura 3.15 e figura 3.17).

Una volta trovate tutte le coordinate dei punti dei contorni, è risultato immediato calcolare area (*cv::contourArea()*) e perimetro (*cv::arcLength()*). Per quanto riguarda le altre misure, si è preferito ricavarle dalle *bounding box* di ogni blocco (figura 3.16 e figura 3.18).

Per distinguere il battistrada di tipo 1A da quello di tipo 1B, è stata inserita, nel codice, la condizione che se il numero di *Blob* e *MacroBlob* trovati è uguale, allora non si prosegue nel trovare la distanza minima tra *Blob* in quanto sarebbe uguale a quella tra *MacroBlob*.

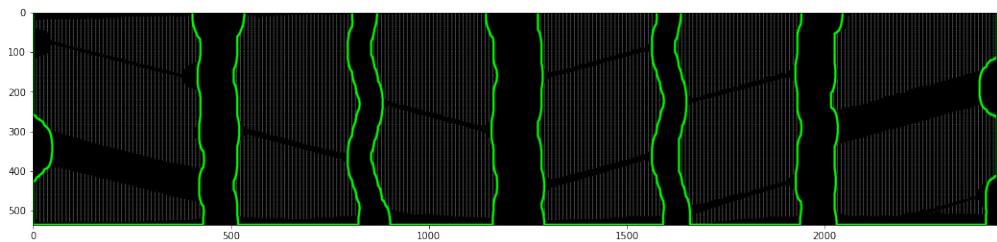


Figura 3.15: Immagine del battistrada di tipo 1A con i contorni dei MacroBlob evidenziati.

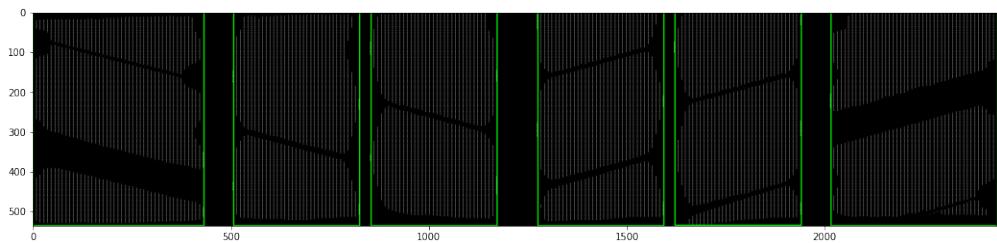


Figura 3.16: Immagine del battistrada di tipo 1A con le bounding box dei MacroBlob evidenziate.

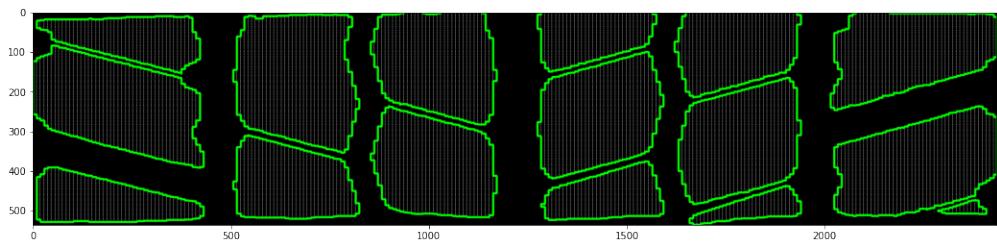


Figura 3.17: Immagine del battistrada di tipo 1A con i contorni dei Blob evidenziati.

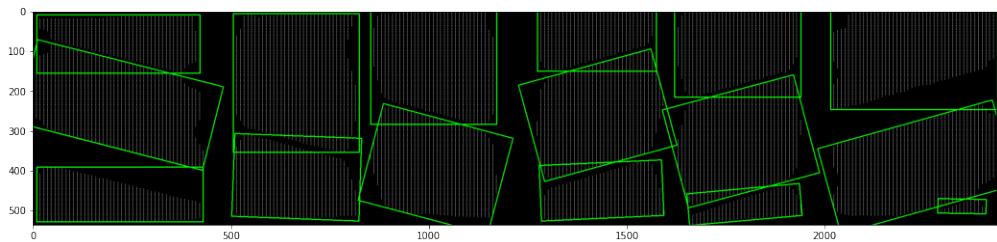


Figura 3.18: Immagine del battistrada di tipo 1A con le bounding box dei Blob evidenziate.

Per trovare la distanza minima tra *Blob* adiacenti presenti in ogni *MacroBlob* è stata creata una funzione specifica che confronta le varie distanze tra i centri dei *Blob*.

Di seguito vengono illustrati i passaggi seguiti per il battistrada di tipo 1B.

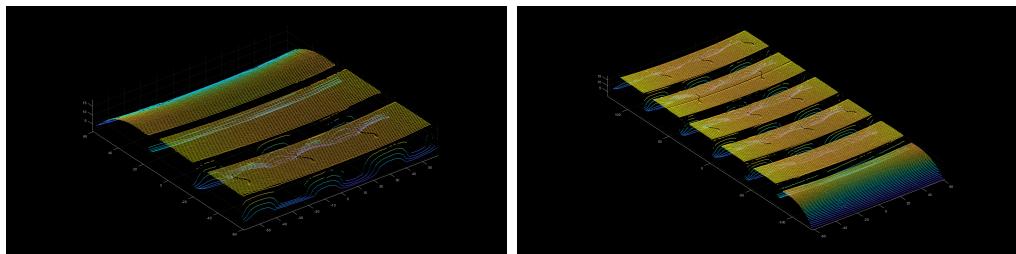


Figura 3.19: Point cloud non elaborata del battistrada di tipo 1B.

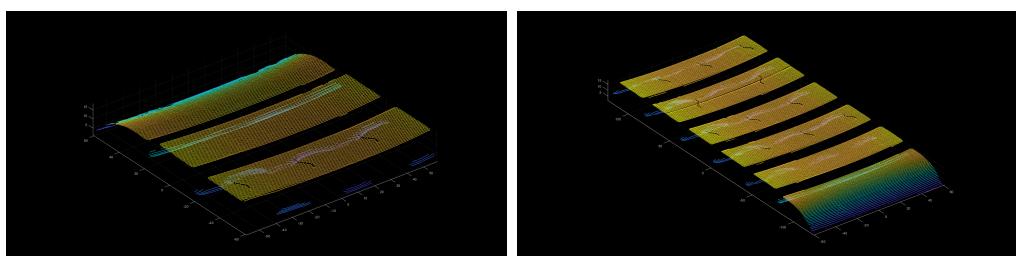


Figura 3.20: Point cloud del battistrada di tipo 1B dopo aver applicato il filtro statistico.

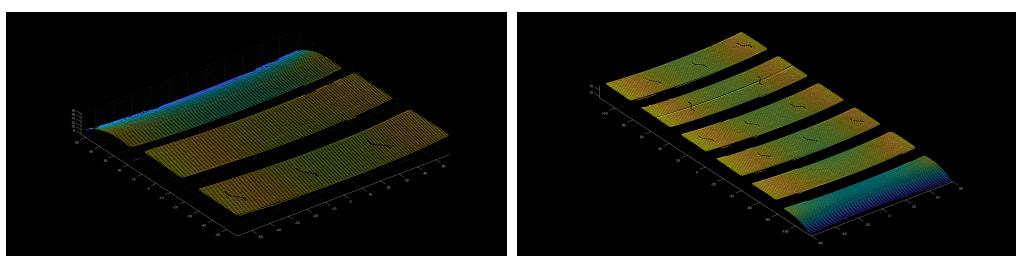


Figura 3.21: Point cloud del battistrada di tipo 1B dopo aver applicato la segmentazione planare.

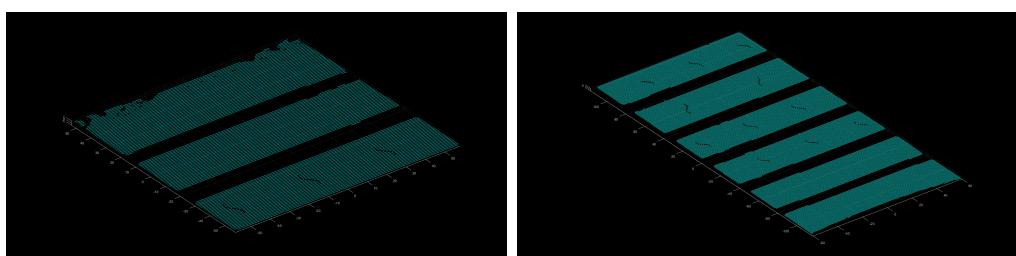


Figura 3.22: Point cloud del battistrada di tipo 1B dopo aver applicato l'operazione di proiezione.

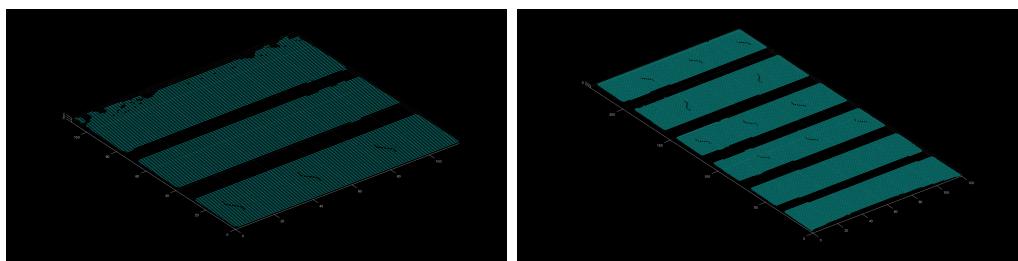


Figura 3.23: Point cloud del battistrada di tipo 1B dopo aver applicato l'operazione di traslazione.

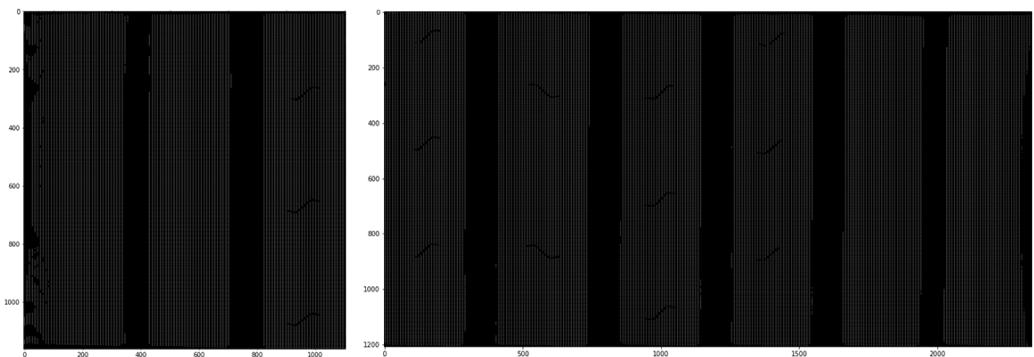


Figura 3.24: Immagine planare del battistrada di tipo 1B.

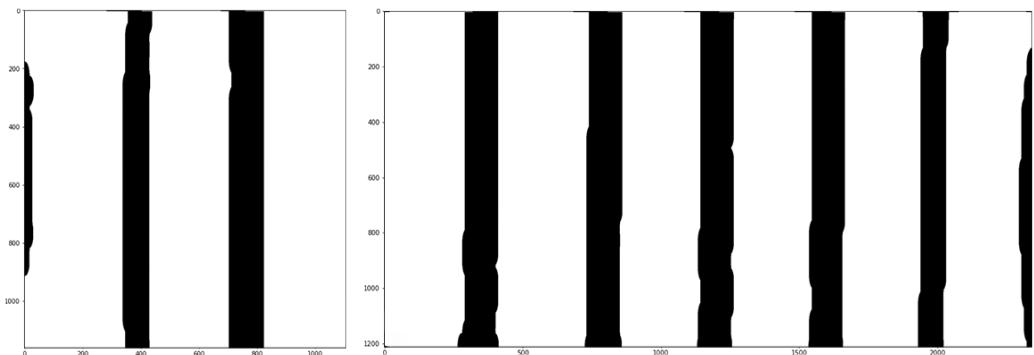


Figura 3.25: Immagine del battistrada di tipo 1B dopo aver applicato la trasformazione morfologica (Closing).

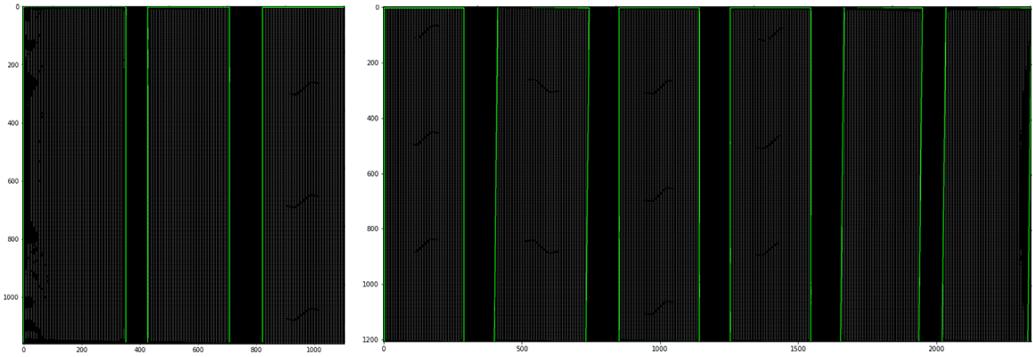


Figura 3.26: Immagine del battistrada di tipo 1B con le bounding box dei MacroBlob evidenziate.

### 3.3.2 Battistrada di tipo 2

L'analisi di questo tipo di battistrada ha l'obiettivo di estrarre le seguenti features:

- Misura della profondità delle scalanature presenti.
- Valore minimo, massimo e medio delle profondità.

Come per i battistrada precedenti, in seguito alla scansione dell'oggetto, anche in questo caso, viene generata una *point cloud* (figura 3.27).

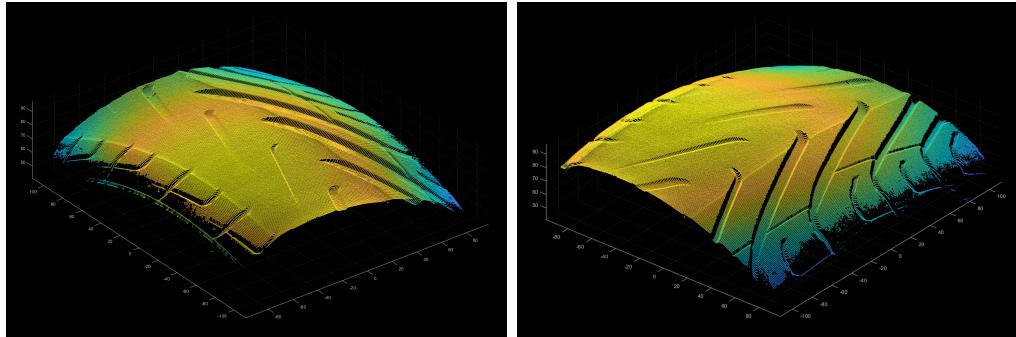


Figura 3.27: Point cloud non elaborata del battistrada di tipo 2.

Per determinare la profondità del battistrada del pneumatico in modo accurato e automatico, è stato sviluppato un metodo di misurazione che prima identifica la posizione delle scalanature e poi calcola, in successione, le profondità di ciascun solco.

L'approccio che è stato seguito è quello di sezionare l'intera *point cloud* lungo

l'asse  $y$ . Ogni sezione è composta da una linea e l'analisi è stata effettuata linea per linea, in modo da avere  $n$  funzioni che, concatenate, ricreino l'intera *point cloud*.

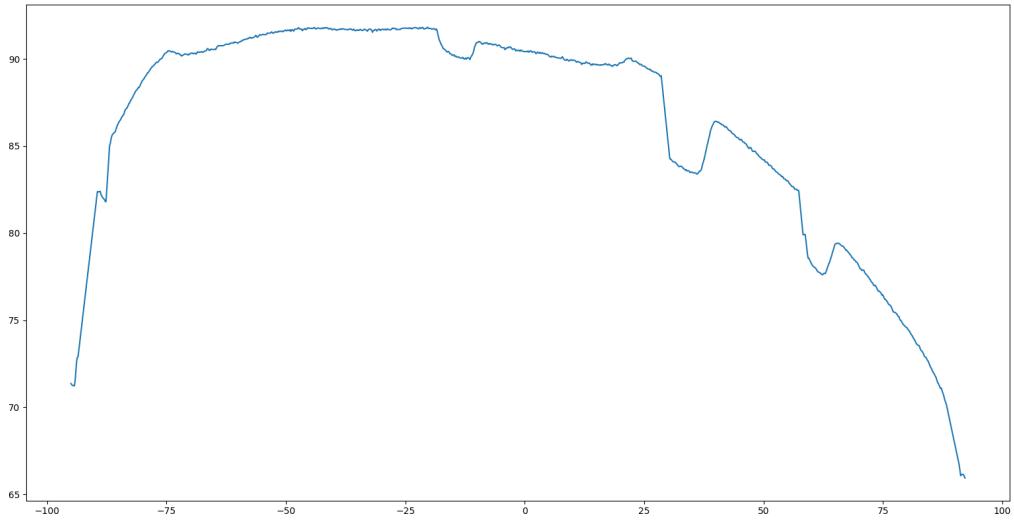


Figura 3.28: Funzione originale.

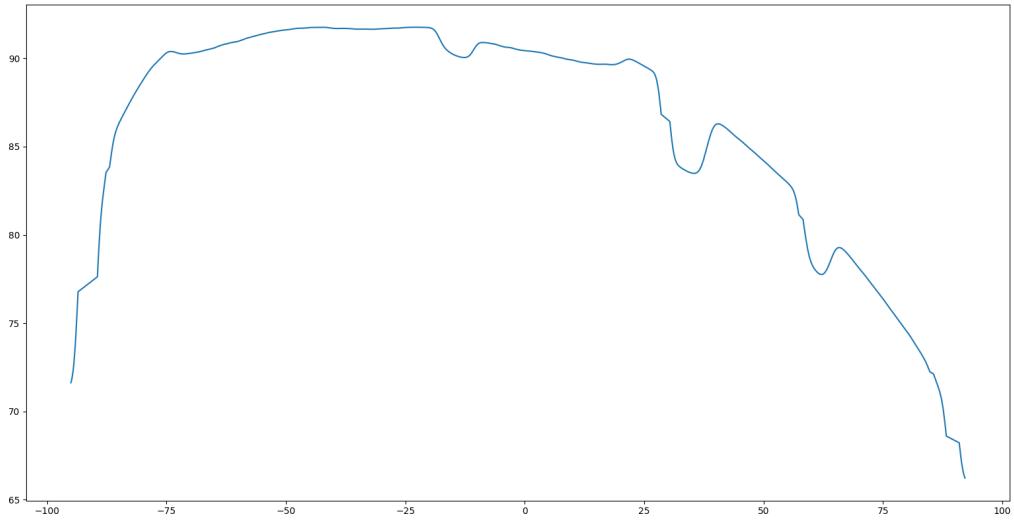


Figura 3.29: Funzione smussata.

Per spiegare l'algoritmo prendiamo come esempio una delle  $n$  funzioni processate. Come si può notare dalla figura 3.28, oltre alle scanalature di grandi dimensioni presenti sulla superficie del pneumatico, sono presenti scanalature molto piccole

(dovute alla rumorosità) che potrebbero trarre in inganno l'algoritmo di ricerca.

Le scanalature più piccole possono essere rimosse dalla curva del profilo applicando un filtro gaussiano, in modo da rendere la funzione più smussata (figura 3.29). Dopo una serie di prove, come parametri del filtro, è stata scelta una finestra  $K = 51$  (dimensione del kernel) e un  $\alpha = 5$  che specifica il numero di deviazioni standard  $\sigma$  desiderate nel kernel.

Ora che la funzione originale è stata smussata, per trovare la posizione e la misura della profondità delle scanalature, ci viene in soccorso lo studio della deriva e dei punti di flesso.

Infatti, tra due punti di massimo della funzione smussata (figura 3.30), si trova obbligatoriamente una scanalatura del battistrada, e la profondità di ciascuna scanalatura è data dalla minima distanza che va dal fondo concavo, alla retta che congiunge i due punti di massimo.

Una volta trovati i punti critici nella funzione smussata, sono state salvate le rispettive coordinate  $x$  di ogni punto e riportate nella funzione originale (figura 3.31 e figura 3.32).

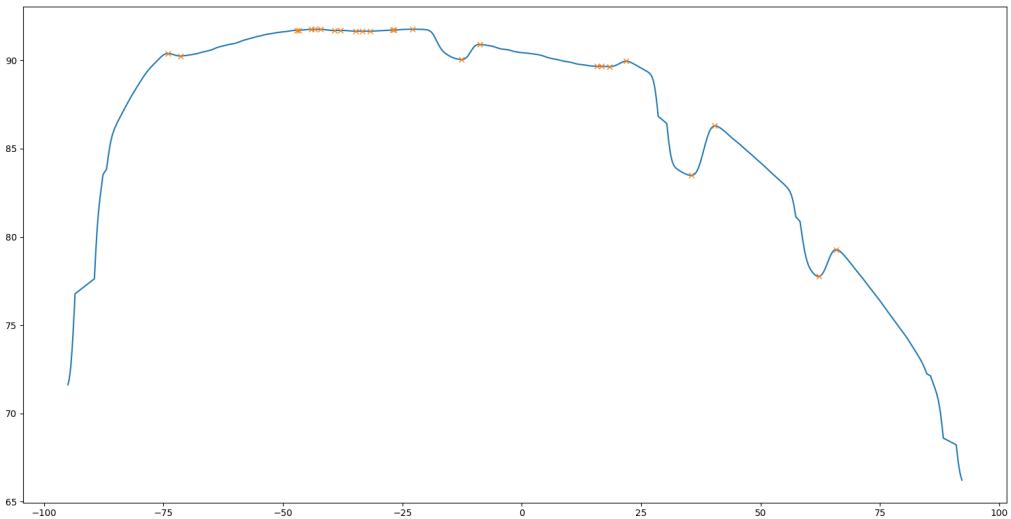


Figura 3.30: Massimi e minimi della funzione smussata.

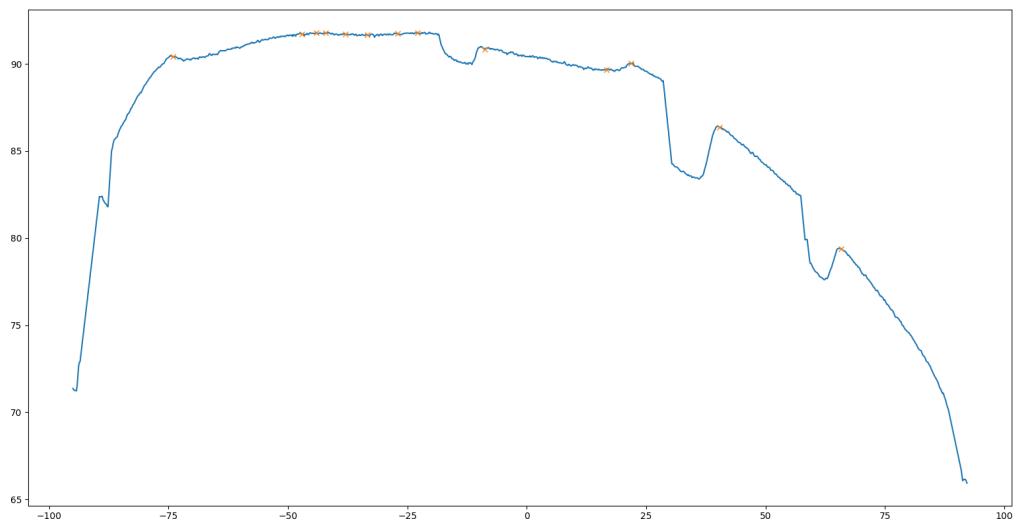


Figura 3.31: Massimi (da destra) della funzione originale.

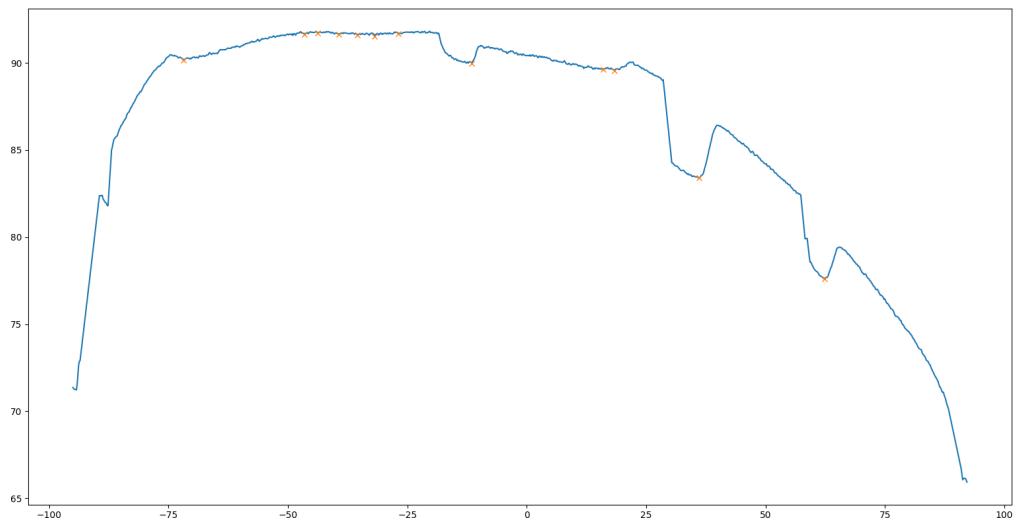


Figura 3.32: Minimi della funzione originale.

Seguendo il percorso del profilo del battistrada, a partire dall'estrema destra, ciò che abbiamo trovato fino ad ora sono: il punto dove inizia la scanalatura, il punto più profondo di essa e il punto dove inizia l'eventuale scanalatura successiva.

Quindi, per migliorare la precisione della posizione delle scanalature, dobbiamo trovare la posizione del punto di chiusura di ogni scanalatura.

Per trovare questi punti, che chiameremo impropriamente punti di massimo da sinistra, è stato seguito il seguente algoritmo:

- Isolare la parte di funzione che si trova tra un minimo e il massimo successivo;
- Ruotarla, facendo in modo che la retta passante tra il minimo e il massimo sia parallela all'asse delle  $x$ ;
- Individuare il punto di massimo della porzione di funzione ruotata, che in definitiva sarà il nostro massimo da sinistra;

Così facendo si trova un buon punto di massimo approssimato, dato che la rotazione non intacca la posizione dei punti.

Nella figura 3.33 è illustrato un esempio di questo calcolo, con evidenziate di verde, sia la funzione smussata sia quella ruotata, da cui si è individuato il nostro punto di massimo.

Una volta ripetuto l'algoritmo per ogni tratto della funzione smussata, la posizione di ogni punto di massimo trovata è stata riportata nella funzione originale (figura 3.34).

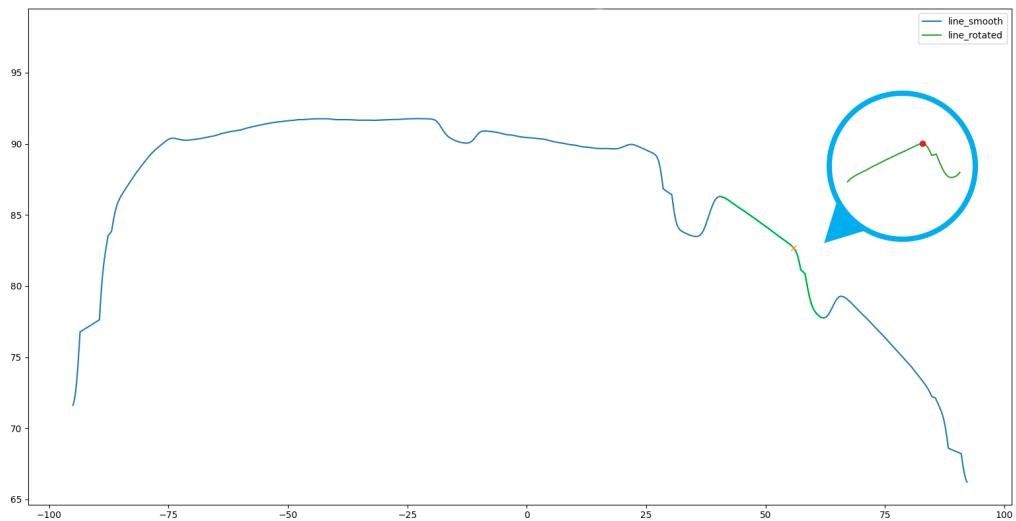


Figura 3.33: Esempio del calcolo di uno dei massimi (da sinistra) della funzione smussata.

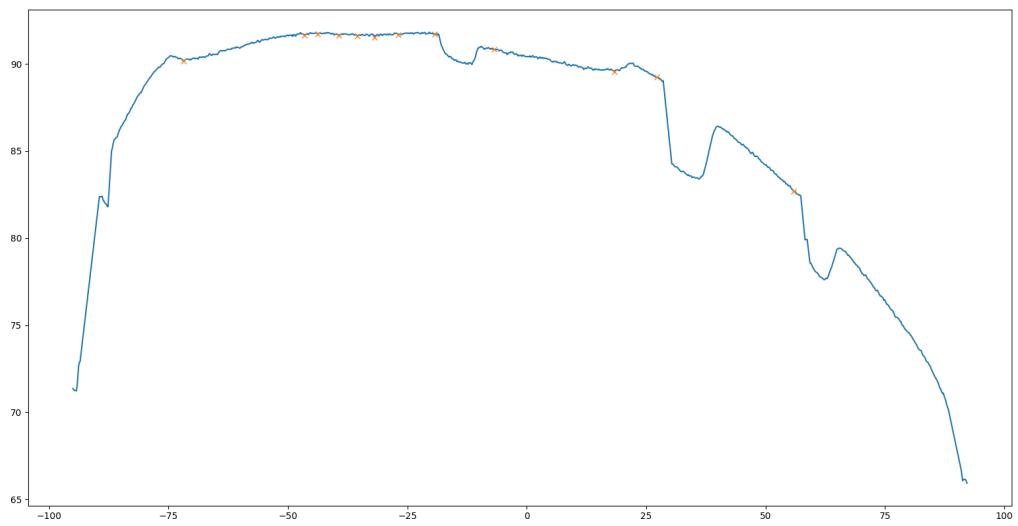


Figura 3.34: Massimi (da sinistra) della funzione originale.

Dopo aver chiarito definitivamente la posizione e il numero delle scanalature del battistrada, è possibile misurare la profondità di ciascuna scanalatura, calcolando la distanza tra la retta che congiunge un massimo da destra e un massimo da sinistra, e il punto di minimo corrispondente.

In fine è stato applicato un filtro, in modo tale da prendere solo le distanze che siano maggiori di 0.5 millimetri (figura 3.35).

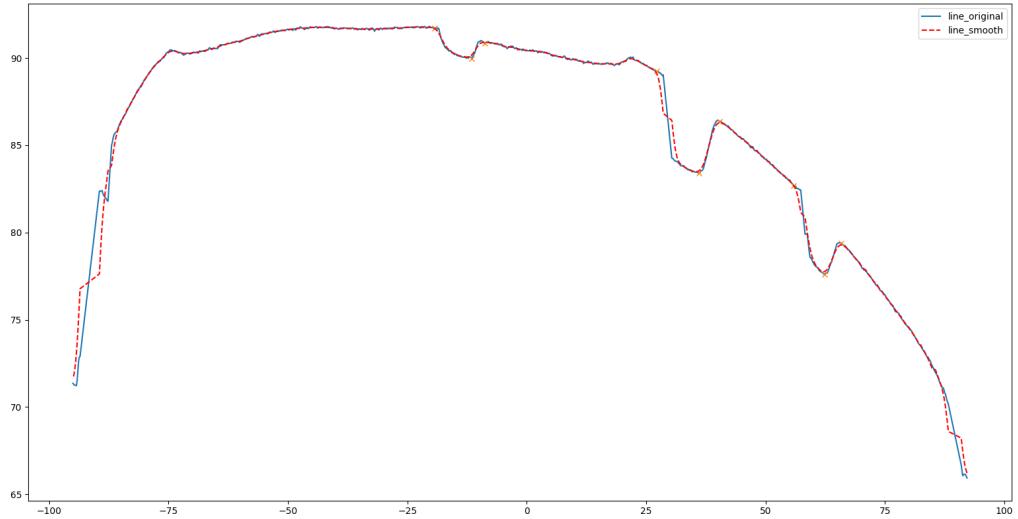


Figura 3.35: Confronto tra la funzione originale e la funzione smussata. I punti critici segnati sono il risultato di un filtro sulla distanza maggiore di 0.5 millimetri.

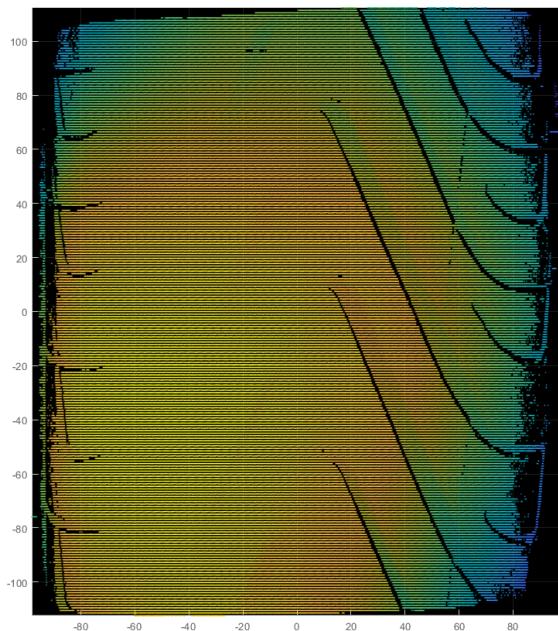


Figura 3.36: Point cloud del battistrada di tipo 2, con vista sul piano  $XY$ .

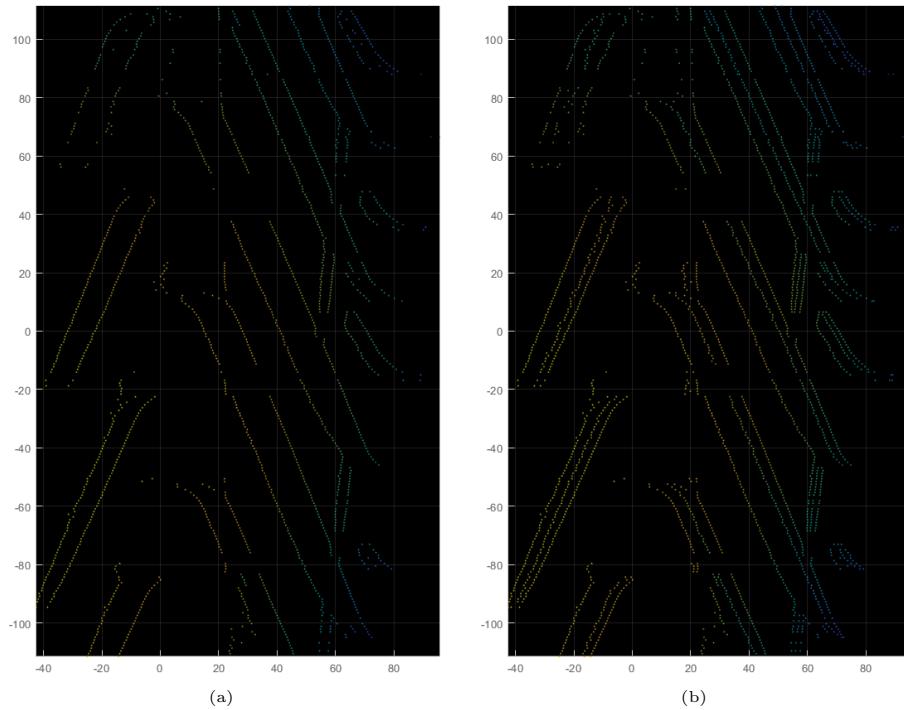


Figura 3.37: (a) Point cloud contenente i punti di massimo da sinistra e di massimo da destra. (b) Point cloud contenente i punti di minimo, di massimo da sinistra e di massimo da destra.

### 3.3.3 Performance

Gli algoritmi sono stati testati su un PC con le seguenti caratteristiche:

- **CPU:** Intel Core i7-10870H;
- **GPU:** Nvidia RTX 3060 6GB GDDR6;
- **RAM:** 16GB DDR4;

Il test è stato svolto scansionando ogni battistrada, consecutivamente, per 10 volte.

Come si nota dai tempi di esecuzione mostrati nella tabella 3.1, gli algoritmi *1A* e *1B* sono abbastanza costosi, in termini di performance, perchè, tra una trasformazione e l'altra, si lavora molto sulle *point cloud*.

L'algoritmo *2*, invece, mostra dei tempi di esecuzione molto buoni.

N	Algoritmo 1A (ms)	Algoritmo 1B (ms)	Algoritmo 2 (ms)
1	2511	3640	181
2	2508	3638	170
3	2505	3650	172
4	2521	3632	165
5	2519	3642	171
6	2502	3651	172
7	2525	3655	176
8	2510	3637	180
9	2530	3653	165
10	2514	3640	175
MEDIA	2514.5	3643.8	172.7

Tabella 3.1: Tempi di esecuzione degli algoritmi.

# Capitolo 4

## Applicazione desktop

Il capitolo seguente sarà incentrato sulla presentazione dell'applicazione desktop e delle sue interfacce e funzionalità.

### 4.1 Sviluppo

L'applicazione è stata creata su *Visual Studio*, utilizzando il framework *WPF* (Windows Presentation Foundation) che fornisce agli sviluppatori un modello di programmazione unificato per la creazione di applicazioni client desktop.

La piattaforma di sviluppo *WPF* supporta un'ampia serie di funzionalità di sviluppo di applicazioni, inclusi un modello applicativo, risorse, controlli, elementi grafici, layout, data binding, documenti e sicurezza.

La parte di *setup* con il Gocator e l'implementazione degli algoritmi di analisi sono stati esportati in una *DLL* (dynamic-link library) che è una libreria a collegamento dinamico che contiene codice e dati che possono essere usati da più programmi contemporaneamente.

In seguito, il collegamento tra la libreria *DLL* e il codice dell'interfaccia grafica, è stato effettuato eseguendo il wrapping di una classe *C++*, contenente le funzioni da esportare, in modo tale da essere usate dal codice creato in *C#*.

Sono state utilizzate anche le *named pipe*, che forniscono la comunicazione interprocesso tra un server pipe e uno o più client pipe.

## 4.2 Librerie di supporto

Le librerie utilizzate per realizzare l'applicazione grafica sono:

- *JSON for Modern C++, versione 3.10.5*, è una libreria per elaborare dati in formato *JSON*;
- *Helix Toolkit, versione 2.20.2*, è una raccolta di componenti 3D per *.NET*;
- *Json.NET - Newtonsoft, versione 13.0.1*, è un popolare framework JSON ad alte prestazioni per *.NET*;
- *Ookii.Dialogs.Wpf, versione 5.0.1*, è una libreria per applicazioni *WPF*.

## 4.3 Interfaccia grafica

L'interfaccia grafica è stata pensata per essere semplice e intuitiva. È presente un pannello di configurazione, uno dove vengono stampati a video i dati derivati dall'analisi e una sezione dove si può visualizzare e interagire con la *point cloud*.

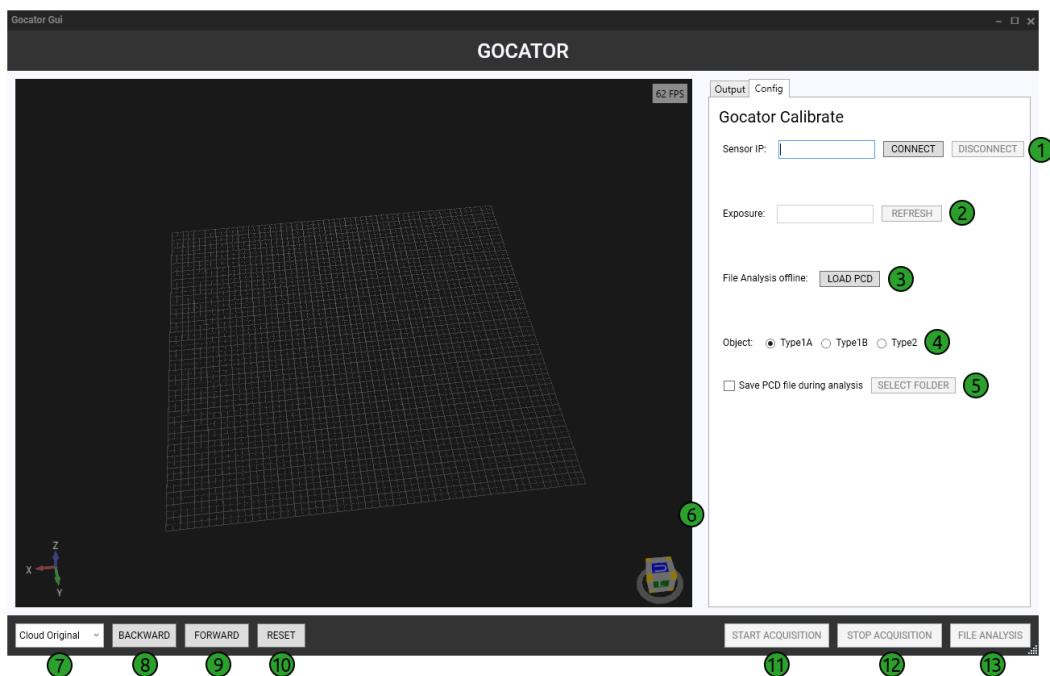


Figura 4.1: Interfaccia grafica con la descrizione dei pulsanti.

Di seguito la descrizione dei pulsanti:

- **1:** Connette e disconnette il sensore, tramite indirizzo IP;
- **2:** Aggiorna il valore dell'esposizione;
- **3:** Carica il file in formato *.pcd* della *point cloud* del battistrada;
- **4:** Seleziona il tipo di battistrada da analizzare;
- **5:** Seleziona la cartella dove salvare le *point cloud*, create durante l'analisi;
- **6:** Visualizzatore 3D delle *point cloud*;
- **7:** Seleziona le *point cloud* da visualizzare per oggetto scansionato;
- **8:** Scorre indietro le *point cloud*;
- **9:** Scorre avanti le *point cloud*;
- **10:** Resetta la visualizzazione e rimuove le *point cloud* caricate;
- **11:** Avvia l'acquisizione dal Gocator;
- **12:** Interrompe l'acquisizione dal Gocator;
- **13:** Avvia l'analisi offline del file caricato;

Quindi, il programma può essere utilizzato sia in modalità online (connesso direttamente al Gocator) sia in modalità offline (caricando e analizzando il file della *point cloud* del battistrada).

La modalità online permette di:

- Inserire l'indirizzo IP del sensore, connettersi e disconnettersi;
- Modificare il valore dell'esposizione;
- Selezionare il tipo di battistrada da analizzare;
- Salvare i file in formato *.pcd*, durante l'analisi, delle *point cloud* step by step;
- Avviare e fermare la scansione, con la visualizzazione del risultato dell'analisi in real time;

- Visualizzare a video le *point cloud* dei battistrada scansionati e analizzati;

La modalità offline permette di:

- Caricare un file contenente la *point cloud* del battistrada;
- Selezionare il tipo di battistrada da analizzare;
- Salvare, durante l'analisi, le *point cloud* step by step;
- Avviare l'analisi del file;
- Visualizzare a video le *point cloud* caricate;

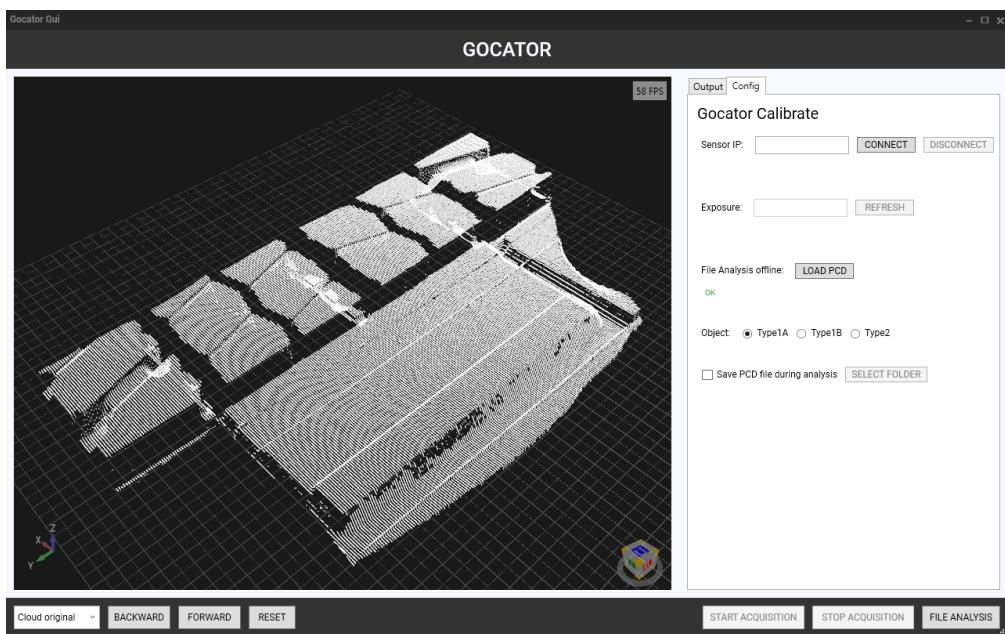


Figura 4.2: Interfaccia grafica con la point cloud filtrata visualizzata a video.

In particolare, nel caso del battistrada di tipo 2, il risultato della scansione riporta a video, oltre alla *point cloud* del battistrada, anche la stessa con evidenziati i punti di altezza delle scanalature minima e massima, e la *point cloud* con solo i punti di massimo da sinistra e destra (figura 4.4, figura 4.5 e figura 4.6).

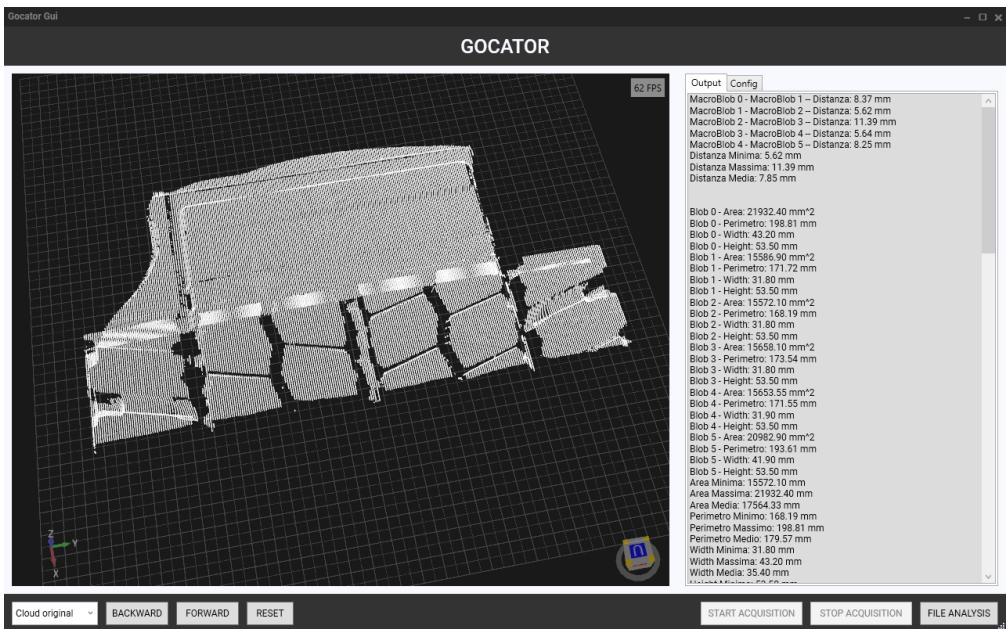


Figura 4.3: Interfaccia grafica con, visualizzata a video, la point cloud del battistrada di tipo 1A e il risultato dell'analisi.

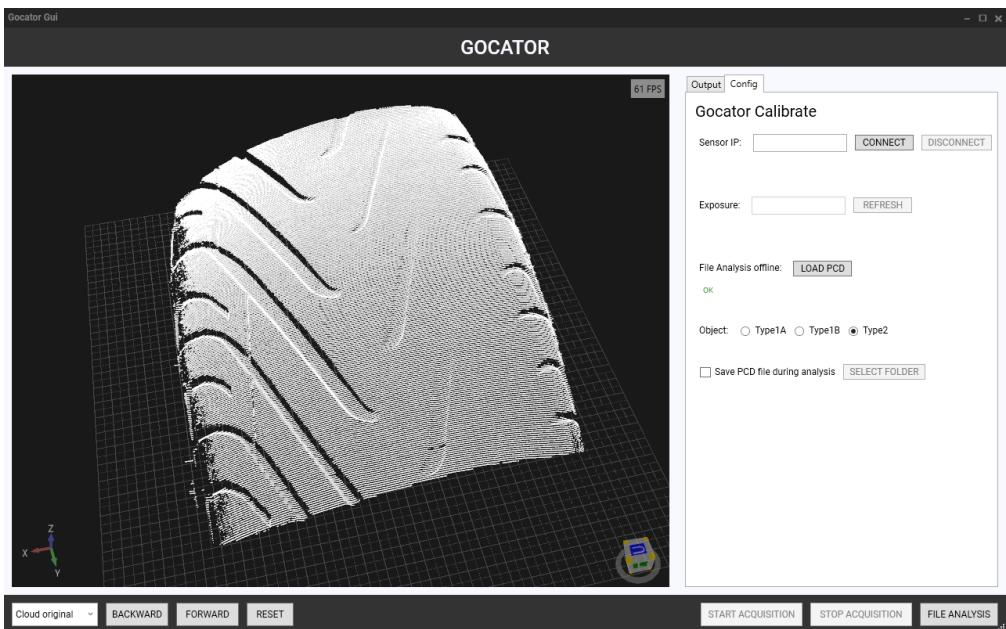


Figura 4.4: Interfaccia grafica con, visualizzata a video, la point cloud del battistrada di tipo 2.

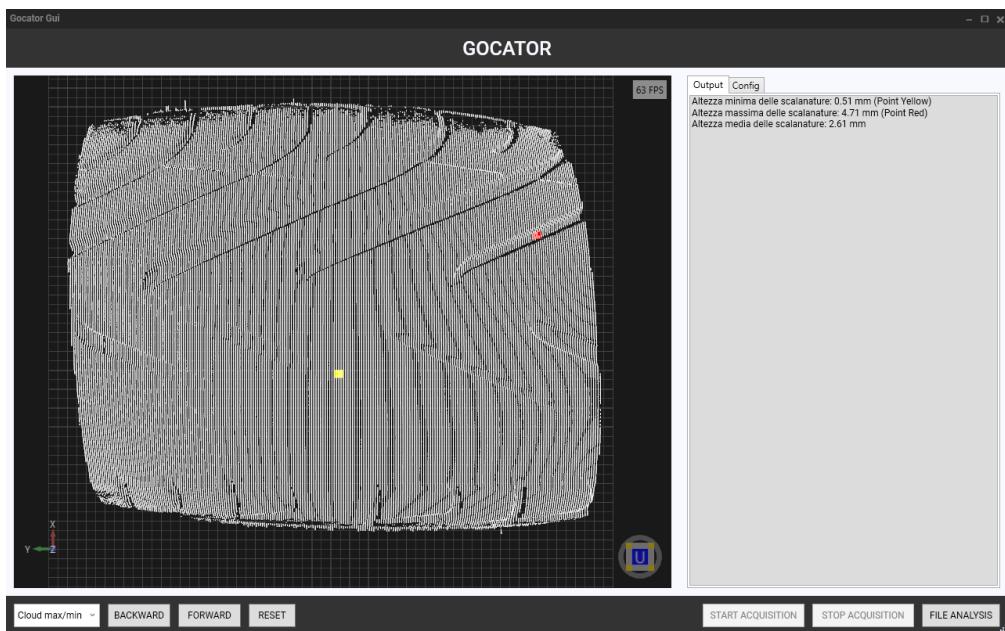


Figura 4.5: Interfaccia grafica con, visualizzata a video, la point cloud con i punti di profondità delle scanalature minima e massima.

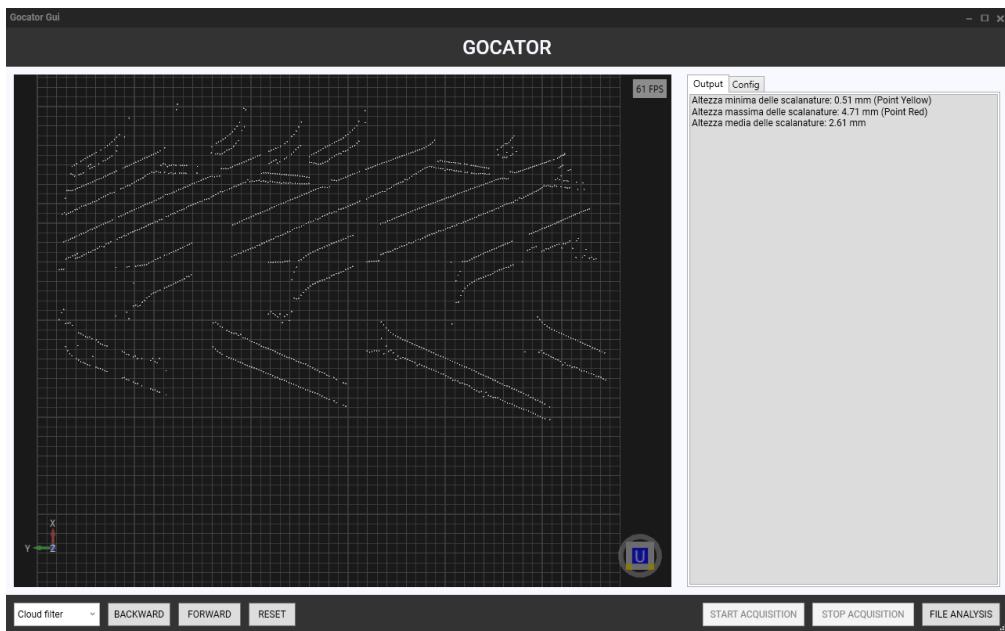


Figura 4.6: Interfaccia grafica con, visualizzata a video, la point cloud filtrata.

# Capitolo 5

## Conclusioni

L'ultimo capitolo fornisce una sintesi dei risultati ottenuti e alcuni suggerimenti su possibili problematiche che potrebbero essere risolte in futuro.

### 5.1 Risultati ottenuti

I risultati sperimentali hanno rilevato che il sistema è in grado di acquisire correttamente le *point cloud* dei battistrada e che gli algoritmi possono effettuare misurazioni, identificare le scanalature e misurarne la profondità.

Riprendendo i test effettuati nel paragrafo 3.3.3, riportiamo i risultati delle scansioni riguardante il battistrada di tipo 2:

N	min (mm)	max (mm)	avg (mm)
1	0.50	4.85	2.68
2	0.50	4.73	2.62
3	0.51	4.76	2.64
4	0.51	4.80	2.66
5	0.50	4.85	2.68
6	0.50	4.77	2.64
7	0.51	4.82	2.67
8	0.50	4.82	2.66
9	0.50	4.79	2.65
10	0.51	4.80	2.66

Tabella 5.1: Valori calcolati su una serie di misurazioni del battistrada di tipo 2.

Ogni misura calcolata dal programma è stata verificata direttamente sugli pneumatici utilizzando un calibro. Nonostante il punto di profondità massima cambiasse leggermente posizione, le misurazioni sono risultate abbastanza accurate e i test hanno sancito il raggiungimento dell'obiettivo che ci si era posti per il progetto.

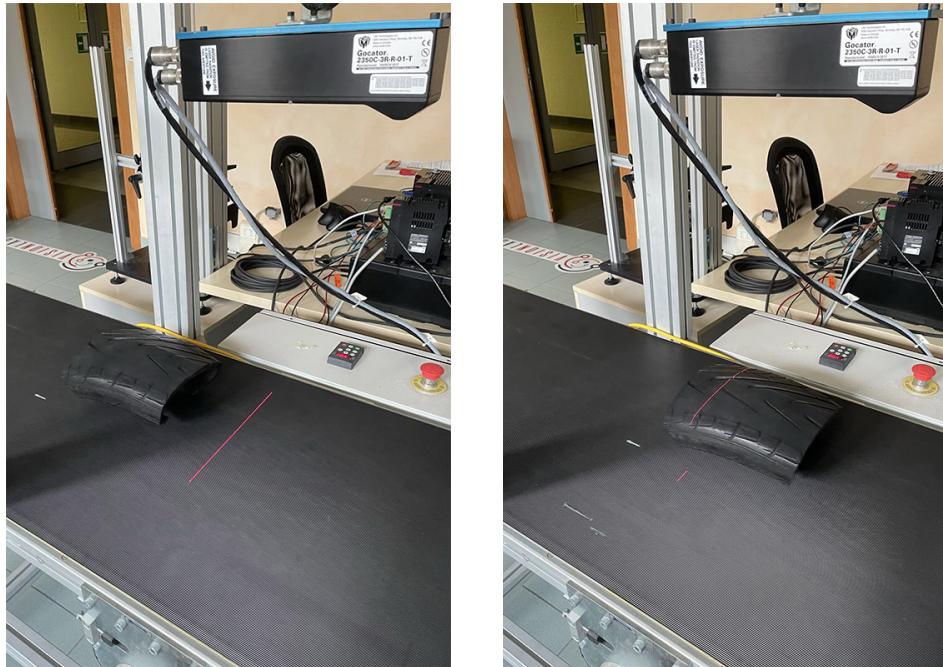


Figura 5.1: Il Gocator in azione durante una scansione.

## 5.2 Conclusione e lavori futuri

Grazie allo studio di questa materia e al rispettivo progetto d'esame, è stata esplorata una branca dell'intelligenza artificiale molto importante per la moltitudine di benefici che apporta in tutti i settori per migliorare l'esperienza del consumatore, ridurre i costi e aumentare la sicurezza.

Seppure i risultati ottenuti sono risultati molto buoni, il margine di miglioramento è ampio, ed una futura ottimizzazione degli algoritmi e di altri particolari del Gocator, potrebbe portare a risultati ancora migliori.

# Bibliografia

- [1] Image S Blog. *LMI Gocator 2490*. URL: <https://www.imagesspa.it/2020/02/26/lmi-gocator-2490/>.
- [2] G. Bradski. «The OpenCV Library». In: *Dr. Dobb's Journal of Software Tools* (2000).
- [3] Helix Toolkit Contributors. *Welcome to Helix Toolkit's documentation!* URL: <http://docs.helix-toolkit.org/en/latest/>.
- [4] David A. Forsyth e Jean Ponce. *Computer Vision - A Modern Approach, Second Edition*. Pitman, 2012, pp. 1–791. ISBN: 978-0-273-76414-4.
- [5] Redazione Geomedia. *LMI Technologies : il sensore Gocator, distribuito in Italia da iMAGE S S.p.a.* URL: <https://rivistageomedia.it/2019072316530/Rilievo-e-localizzazione-lmi-technologies-il-sensore-gocator-distribuito-in-italia-da-image-s-s-p-a>.
- [6] Rafael C. Gonzalez e Richard E. Woods. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN: 9780131687288 013168728X 9780135052679 013505267X. URL: <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>.
- [7] LMI Technologies Inc. *Gocator 2300*. URL: <https://lmi3d.com/series/gocator-2300-series/>.
- [8] Direct Industry. *Sensore di visione intelligente Gocator Series*. URL: <https://www.directindustry.it/prod/lmi-technologies/product-7226-703191.html>.
- [9] Point Cloud Library. *Plane model segmentation*. URL: [https://pcl.readthedocs.io/en/latest/planar\\_segmentation.html](https://pcl.readthedocs.io/en/latest/planar_segmentation.html).
- [10] Point Cloud Library. *Projecting points using a parametric model*. URL: [https://pcl.readthedocs.io/projects/tutorials/en/latest/project\\_inliers.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/project_inliers.html).

- [11] Point Cloud Library. *Removing outliers using a StatisticalOutlierRemoval filter*. URL: [https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical\\_outlier.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical_outlier.html).
- [12] Point Cloud Library. *Using a matrix to transform a point cloud*. URL: [https://pcl.readthedocs.io/projects/tutorials/en/latest/matrix\\_transform.html](https://pcl.readthedocs.io/projects/tutorials/en/latest/matrix_transform.html).
- [13] Microsoft. *Creare DLL C/C++ in Visual Studio*. URL: <https://docs.microsoft.com/it-it/dotnet/standard/io/how-to-use-named-pipes-for-network-interprocess-communication>.
- [14] Microsoft. *Informazioni su Windows Presentation Foundation (WPF)?* URL: <https://docs.microsoft.com/it-it/visualstudio/designers/getting-started-with-wpf?view=vs-2022>.
- [15] Microsoft. *Procedura: eseguire il wrapping di una classe nativa affinché possa essere utilizzata in C#*. URL: <https://docs.microsoft.com/it-it/cpp/dotnet/how-to-wrap-native-class-for-use-by-csharp?view=msvc-170>.
- [16] Microsoft. *Procedura: Usare le named pipe per la comunicazione interprocesso in rete*. URL: <https://docs.microsoft.com/it-it/cpp/build/dlls-in-visual-cpp?view=msvc-170>.
- [17] The GSL Team. *Digital Filtering*. URL: <https://www.gnu.org/software/gsl/doc/html/filter.html>.
- [18] Xi-Bo Wang, Ai-Juan Li, Qin-Peng Ci, Meng Shi, Tian-Long Jing e Wan-Zhong Zhao. «The study on tire tread depth measurement method based on machine vision». In: *Advances in Mechanical Engineering* 11.4 (2019), p. 1687814019837828. DOI: 10.1177/1687814019837828. URL: <https://doi.org/10.1177/1687814019837828>.