



UNIVERSITÀ  
di **VERONA**

Dipartimento  
di **INFORMATICA**



# **Ricostruzione 3D di un ambiente anatomico da immagini endoscopiche**

Tesi di laurea triennale in Informatica  
Università degli studi di Verona  
Laboratorio Altair

Alessandro Riboni

# Indice

- 3. Introduzione
- 6. Algoritmi di segmentazione
- 9. Problematiche incontrate
- 10. Studio ed analisi delle possibili soluzioni
- 12. OpenMVG
- 15. OpenMVS
- 16. Script per la ricostruzione 3D
- 19. Applicazione sul robot Da Vinci
- 22. Utilizzo degli algoritmi
- 25. Risultati ottenuti
- 27. Considerazioni finali
- 29. Software e librerie utilizzati

# Introduzione

**Nel mese di Aprile dell'anno 2017, durante il secondo anno del mio percorso di laurea in Informatica, ho iniziato uno stage presso il Laboratorio Altair.**

**Il progetto a cui avrei lavorato riguardava lo studio e l'applicazione di algoritmi per la ricostruzione, la segmentazione e la classificazione di immagini provenienti dall'endoscopio del robot Da Vinci.**

**Ho lavorato a questo progetto per circa un anno: nella prima parte ho collaborato con un altro studente del corso di laurea in Informatica, Moreno Bragaglio. Abbiamo iniziato lo stage studiando il framework ROS e gli algoritmi della libreria PCL.**

**Dopo aver effettuato diversi test ed aver riscontrato dei problemi riguardanti la ricostruzione 3D, abbiamo valutato alcune alternative e cambiato strategia.**

# Introduzione

Nella seconda parte del progetto, ho utilizzato le librerie OpenMVG e OpenMVS per la ricostruzione di una scena registrata con una stereo camera.

Visti gli ottimi risultati ottenuti, ho deciso di applicare gli algoritmi di queste librerie sulle immagini provenienti dall'endoscopio del robot Da Vinci di un ambiente anatomico.

Dopo alcune difficoltà dovute principalmente alla bassa qualità delle immagini, sono riuscito ad ottenere un'ottima ricostruzione e, grazie al lavoro svolto durante la prima parte dello stage, ho potuto utilizzare gli algoritmi per la segmentazione e la classificazione su di essa.

Utilizzando principalmente algoritmi basati sul colore e sulla distanza euclidea (clustering) sono riuscito ad estrapolare i vari organi presenti nella scena, creandomi un piccolo dataset.

# Introduzione

Utilizzando l'algoritmo 'correspondence\_grouping' della libreria PCL è possibile vedere le corrispondenze dei modelli estrapolati con la scena ricostruita.

Prima di iniziare ad esporre nel dettaglio i vari punti del progetto vorrei ringraziare pubblicamente Nicola Piccinelli per l'aiuto datomi nella scelta delle librerie da utilizzare per la ricostruzione 3D e soprattutto Andrea Roberti, il quale mi ha seguito durante tutto il tirocinio ed è stato fondamentale per la riuscita di questo progetto.

# Algoritmi di segmentazione

Nella prima parte dello stage, mi sono concentrato sullo studio degli algoritmi utili per segmentare una scena 3D.

Per i primi test ho utilizzato il software V-REP, con il quale simulavo una semplice scena formata da due tazze colorate.

Un nodo ROS si sottoscriveva ad un topic, sul quale venivano pubblicati i frame della scena registrata.

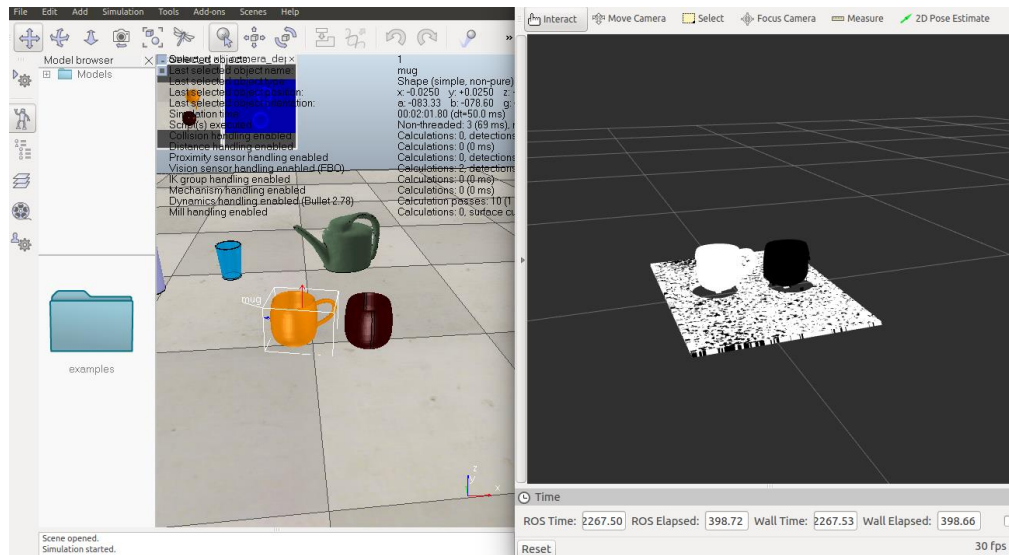
Per visualizzare la point cloud (nuvola di punti 3D) segmentata e/o i cluster estrapolati ho utilizzato Rviz, un plugin presente all'interno di ROS.

Dopo diversi test ed un'attenta analisi ho scelto due diversi approcci, uno basato sul colore, grazie anche alla conversione da RGB a HSV, ed uno basato sul clustering, in particolare sulla distanza euclidea dei punti.

# Algoritmi di segmentazione

- RGB to HSV:

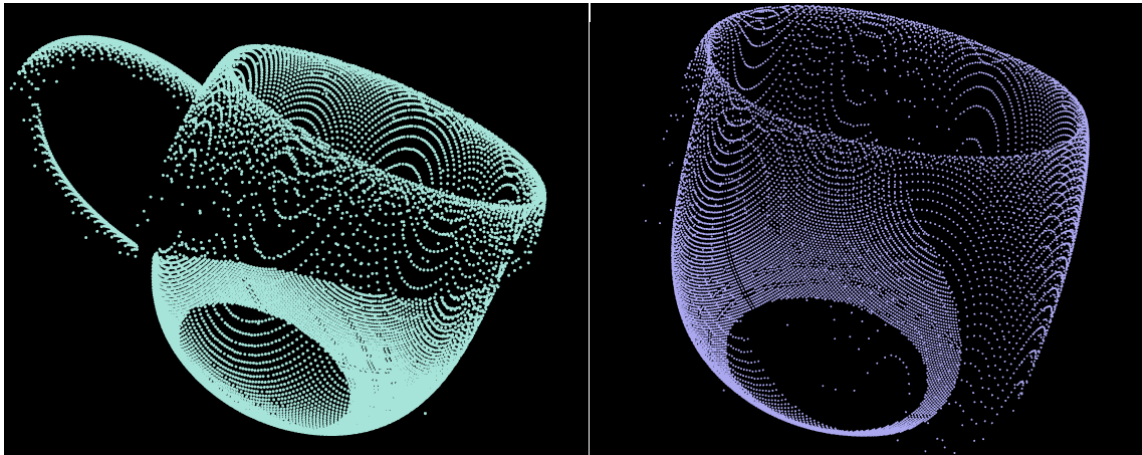
- Questo algoritmo si basa sulla conversione della point cloud dal formato XYZRGB a XYZHSV. La segmentazione è stata fatta utilizzando come parametri la tonalità (hue) e la saturazione (saturation).



# Algoritmi di segmentazione

- **Algoritmo di Clustering :**

- Questo algoritmo non è altro che l'algoritmo di clustering presente nella libreria PCL con le modifiche necessarie per l'utilizzo con ROS.
- La scena viene inizialmente filtrata eliminando 'il pavimento' e successivamente viene divisa in cluster in base alla distanza euclidea dei punti.





# Problematiche incontrate

Dopo aver studiato le varie tecniche di segmentazione, ho provato ad applicarle su una scena registrata da una stereo camera di un ambiente anatomico.

Il tentativo non è andato a buon fine, infatti, non avevo considerato la differenza tra la scena simulata, in cui la camera risultava fissa e inquadrava tutti gli oggetti presenti, e la scena registrata dell'ambiente anatomico, dove la camera è in movimento e spesso inquadra solo una parte degli organi presenti nel manichino, impedendo il riconoscimento di essi dai singoli frame del video.

La soluzione che ho trovato a questo problema è stata quella di creare un ricostruzione 3D in base ai frame provenienti dal video e, successivamente, applicare gli algoritmi di segmentazione per riconoscere i vari organi.

# Studio ed analisi delle possibili soluzioni

- **ORB-SLAM 2:**

- È una libreria che utilizza degli algoritmi di SLAM (Simultaneous Localization and Mapping) per effettuare una ricostruzione real-time della scena.
- Inizialmente sembrava potesse essere la soluzione migliore, considerando soprattutto il vantaggio della ricostruzione in tempo reale, ma a causa della bassa qualità delle immagini della camera e probabilmente anche a dei limiti di questa libreria, le features estrapolate per la ricostruzione della scena erano troppo poche per pensare di applicare alla point cloud ottenuta degli algoritmi di segmentazione.

# Studio ed analisi delle possibili soluzioni

- **OpenMVG e OpenMVS:**
  - OpenMVG è una libreria per la computer-vision e si basa sulla Multiple View Geometry, la quale descrive le relazioni e i vincoli geometrici che legano due immagini 2D della stessa scena 3D catturate da due fotocamere con posizionamento e orientamento distinto.
  - Muovendo la camera sopra l'ambiente anatomico ottenevo dei frame con angolazioni diverse e perciò era possibile applicare tale libreria.
  - Attraverso una pipeline formata da diversi algoritmi è possibile ottenere una ricostruzione 3D partendo da insieme di frame del video registrato.
  - Inoltre, con l'utilizzo della libreria OpenMVS è possibile ottenere una point cloud più densa con superfici più omogenee.
  - Per questi motivi ho scelto di utilizzare queste due librerie per produrre la ricostruzione 3D dell'ambiente anatomico.

# OpenMVG

La tecnica utilizzata dalla libreria OpenMVG è chiamata ‘Structure for Motion’ e permette di ricostruire la forma di oggetti attraverso l’individuazione di punti da un insieme di fotogrammi. Estrapola i punti chiave (features) e incrocia i punti riconoscibili su più foto, trovandone le coordinate spaziali.

Come detto in precedenza la libreria lavora attraverso una pipeline di algoritmi, ognuno dei quali dipende da dei parametri fondamentali per la creazione di un ottima point cloud. La SfM pipeline è formata da:

## 1. ImageListing:

- Creazione di una `sfm_data.json` contenente i frame di input della pipeline. È importante indicare la `k` matrix e/o il punto focale della camera.
- Per ogni immagine crea un oggetto ‘view’ in cui salva le informazioni (nome, dimensione, informazioni riguardanti la calibrazione della camera).

# OpenMVG

## 2. ComputeFeatures:

- Calcola la descrizione dell'immagine di una SfM\_data.json data in input.
- Uno dei parametri di questo algoritmo è il 'descriptorMethod', ovvero il metodo con cui si vuole descrivere l'immagine. Il metodo di default è il SIFT, il quale estrae features, partendo dagli spigoli in modo da riconoscere le varie forme presenti nelle immagini.
- La scelta del metodo dipende soprattutto dal set di immagini di input, non c'è un metodo migliore di altri.

## 3. ComputeMatches:

- Le immagini hanno una visual overlap (sovrapposizione visiva) perciò vengono stabilite delle corrispondenze, le quali successivamente vengono filtrate con filtri geometrici.
- Un parametro a cui è necessario fare attenzione è il 'ratio' che permette di settare la distanza minima dei punti vicini ('nearest neighbor distance ratio').

# OpenMVG

## 4. IncrementalSfM:

- Crescita incrementale della ricostruzione. Il processo parte da due view e in modo iterativo viene esteso aggiungendo nuove views e punti 3D attraverso la stima della posizione e la triangolazione.

## 5. DataColour:

- Calcola il colore della struttura

## 6. ComputeStructureFromKnowPose:

- Calcola le features corrispondenti e una robusta triangolazione in base alla geometria conoscendo i parametri intrinseci e la posizione della camera.

## 7. OpenMVG2openMVS:

- Conversione della point cloud ottenuta come output della pipeline in modo che possa essere elaborata dagli algoritmi della libreria openMVS. Quest'ultimo passaggio rendere la nuvola di punti densa, completa e precisa.

# OpenMVS

OpenMVS è una libreria che permette di colmare le lacune della ricostruzione tramite la pipeline di openMVG, fornendo un set completo di algoritmi che permettono di recuperare l'intera superficie della scena da ricostruire. Data in input un insieme di 'camera poses' e una nuvola di punti sparsa è possibile ottenere come output una mesh strutturata.

L'algoritmo 'DensifyPointCloud' crea una nuvola di punti più densa e completa mentre 'ReconstructionMesh' e 'TextureMesh' permettono di recuperare tutti i dettagli e ottenere una superficie più omogenea della scena ricostruita.

# Script per la ricostruzione 3D

Dopo aver studiato le diverse alternative ed aver deciso di proseguire il progetto utilizzando gli algoritmi delle librerie appena presentate ho deciso di creare uno script in modo da sfruttare le potenzialità della pipeline.

Dato in input un insieme di fotogrammi di una scena registrata, lo script permette di scegliere i parametri fondamentali per ottenere una ricostruzione ottimale.

Lavorando con una stereo camera, l'idea era quella di effettuare due ricostruzioni, una attraverso i frame della camera destra e una con i frame della camera sinistra.

Successivamente unire le due point cloud ottenute attraverso l'algoritmo ICP (iterative closest point) in modo da ottenere un'unica nuvola di punti, su cui sarebbe stato possibile applicare gli algoritmi di segmentazione.



# Script per la ricostruzione 3D

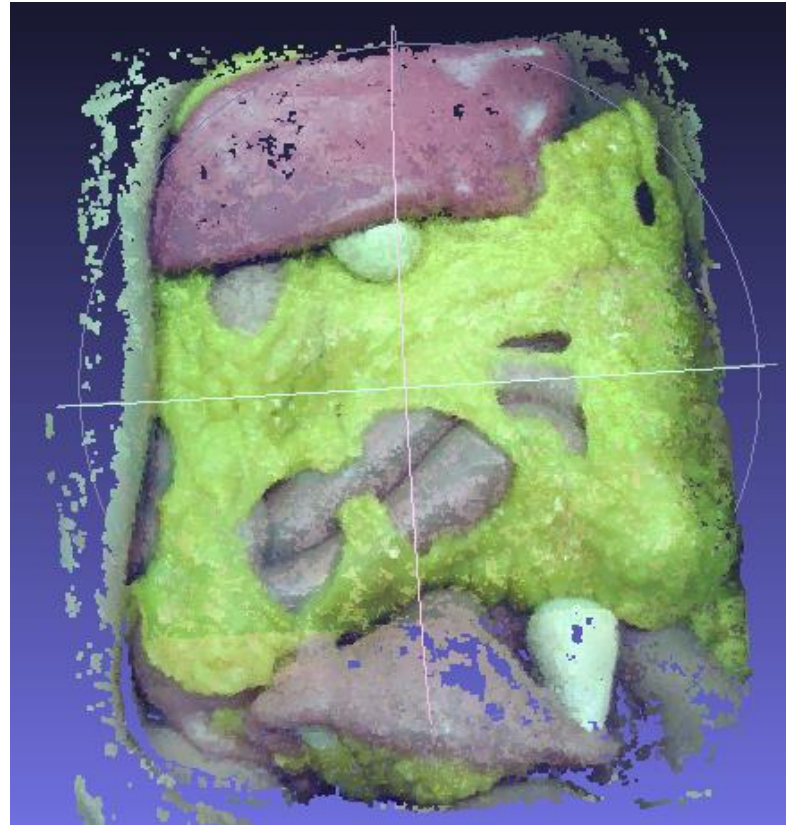
Ho utilizzato un insieme di fotogrammi estrapolati da un bag file di una registrazione di un ambiente anatomico effettuata con la stereo camera e dopo aver testato i vari metodi per descrivere le immagini e adattato i parametri dei vari algoritmi della pipeline sono riuscito ad ottenere un risultato soddisfacente.

In tutte e tre le dimensioni la point cloud ottenuta risultava coerente con le distanze reali del manichino utilizzato.

Inoltre, la tonalità del colore degli organi presenti nella ricostruzione era pressoché identica a quella reale.

Nella slide successiva è presente la ricostruzione ottenuta tramite l'utilizzo dello script con il metodo descrittivo AKAZE\_FLOAT.

# Script per la ricostruzione 3D



# Applicazione sul robot Da Vinci

Visti gli ottimi risultati ottenuti utilizzando lo script per la ricostruzione 3D con una stereo camera ho deciso di provare ad applicarlo su immagini provenienti da un bag file registrato dall'endoscopio del robot Da Vinci sempre sullo stesso ambiente anatomico.

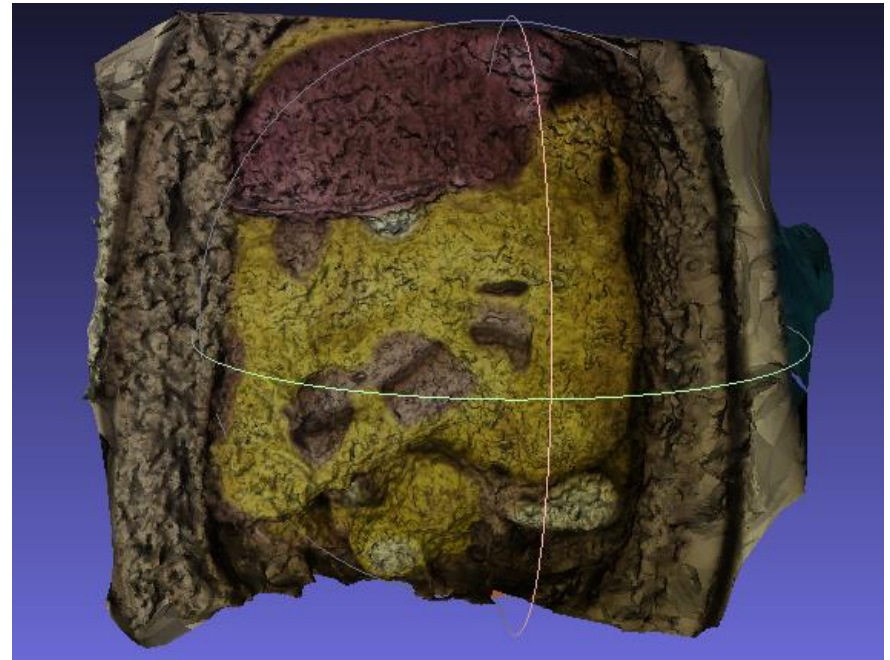
Con le camere analogiche montate sul macchinario lo script non riusciva a trovare le corrispondenze per creare una point cloud a causa della scarsa luminosità e della qualità limitata.

Sostituendo le camere con una digitale e settando lo script con una configurazione diversa da quella utilizza con la stereo camera sono riuscito ad ottenere un ottima ricostruzione, sulla quale avrei potuto applicare gli algoritmi studiati nella prima parte dello stage.

# Applicazione sul robot Da Vinci



Scene\_dense.ply



Scene\_dense\_mesh\_texture.ply

# Applicazione sul robot Da Vinci

Una volta ottenuta la ricostruzione ho convertito la point cloud nel formato .pcd e l'ho filtrata in modo da mantenere solamente i punti dell'ambiente anatomico. Sulla nuvola di punti ottenuta ho applicato gli algoritmi di segmentazione per estrapolare i singoli organi.



# Utilizzo degli algoritmi

Nella fase conclusiva dello stage mi sono concentrato sull'applicazione degli algoritmi studiati sulla ricostruzione 3D ottenuta dalle immagini dell'endoscopio.

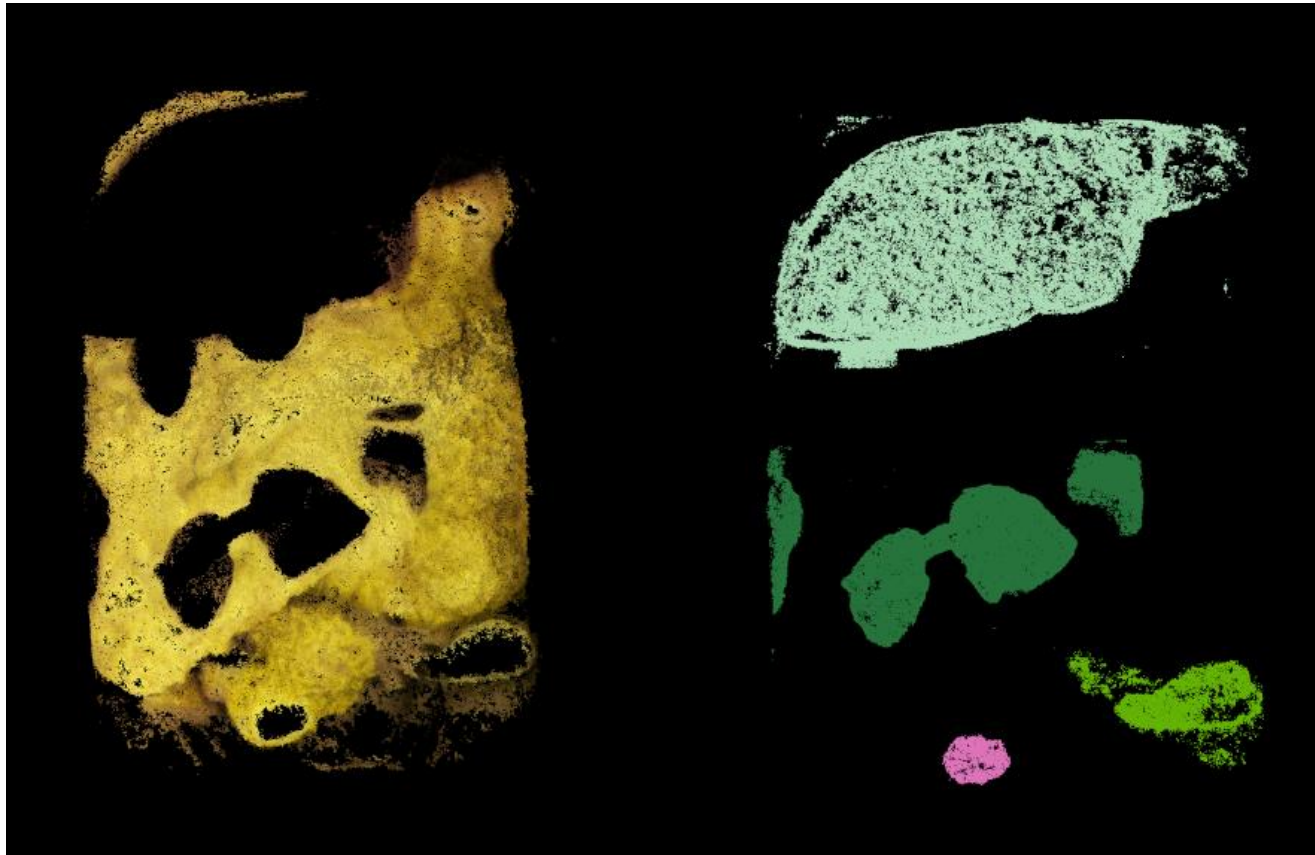
Visto il forte contrasto tra i colori della scena è stato semplice applicare la segmentazione basata sul colore per estrapolare principalmente la parte gialla, ovvero il grasso viscerale.

Sulla point cloud risultante ho applicato l'algoritmo di clustering che mi ha permesso di isolare il fegato e l'intestino in due cluster separati.

La scelta dei parametri riguardanti la tolleranza del cluster e la dimensione minima è stata fondamentale per ottenere i cluster desiderati.



# Utilizzo degli algoritmi



# Utilizzo degli algoritmi

Per poter visualizzare i risultati ottenuti ho deciso di utilizzare l'algoritmo della libreria PCL 'correspondence\_grouping' già studiato e analizzato nella prima parte del progetto.

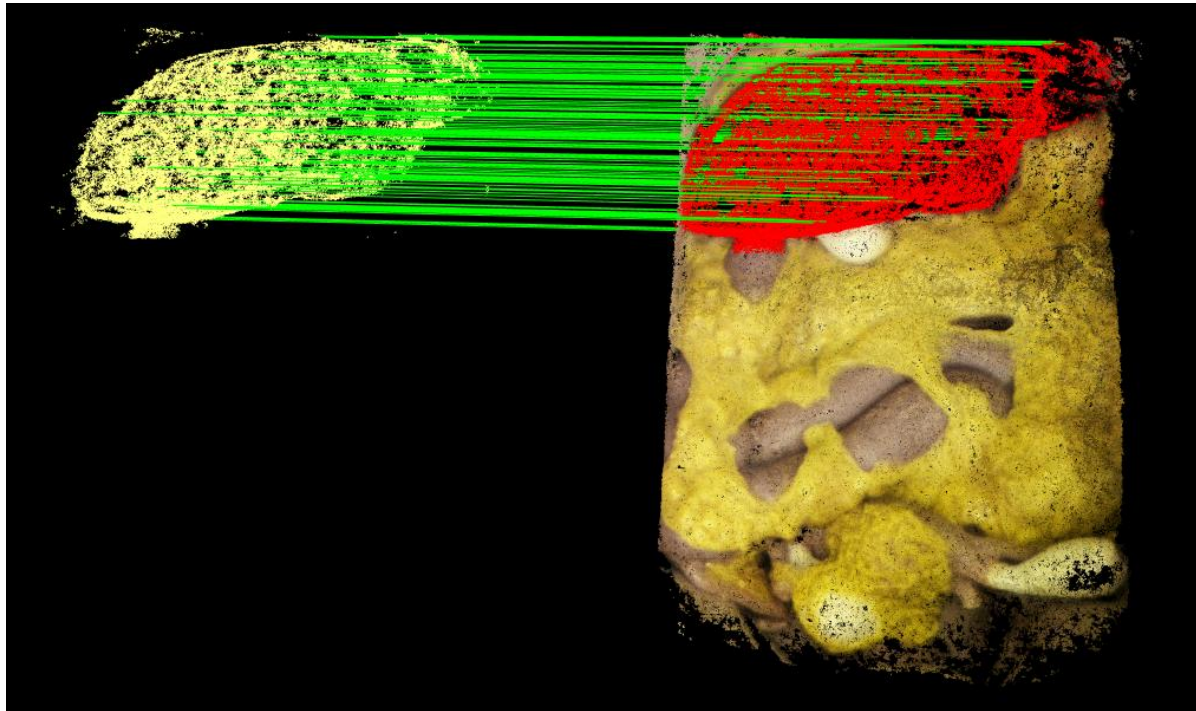
Dato un modello e una scena in input, questo algoritmo è in grado, tramite algoritmi di clustering (Hough 3D), di riconoscere il modello nella scena, nel caso in cui sia presente.

Ho creato un piccolo dataset di modelli formato dagli organi e dai cluster estrapolati precedentemente.

Utilizzando questo algoritmo con la scena ricostruita e i singoli modelli è possibile visualizzare le corrispondenze e i punti chiave.

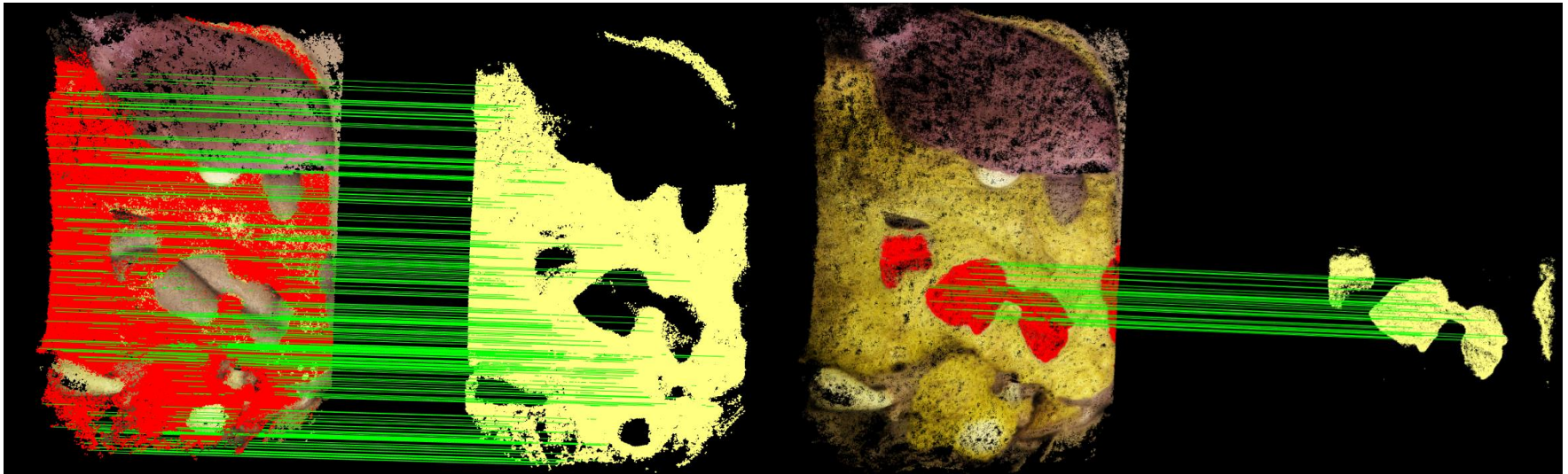


# Risultati ottenuti



Applicazione dell'algoritmo `correspondence_grouping` tra il fegato (modello) e la ricostruzione 3D ottenuta (scena).

# Risultati ottenuti



- Applicazione dell'algoritmo `correspondence_grouping` tra il grasso viscerale (modello) e la ricostruzione 3D ottenuta (scena).
- Applicazione dell'algoritmo `correspondence_grouping` tra l'intestino (modello) e la ricostruzione 3D ottenuta (scena).

# Considerazioni finali

Le librerie utilizzate per la ricostruzione della scena permettono di ottenere buoni risultati anche da un set di immagini con una qualità abbastanza limitata perciò possono essere molto utili per applicazioni di questo tipo.

La problematica principale è il tempo di computazione, infatti, lo script contenente la pipeline Structure for Motion può impiegare addirittura una decina di minuti per ricostruire la scena. Questo fa sì che sia praticamente impossibile pensare di utilizzare questa tecnologia in tempo reale.

Per quanto riguarda gli algoritmi di segmentazione è necessario fare molta attenzione ai parametri. Se ci si basa sul colore bisogna prestare attenzione all'illuminazione presente durante la registrazione della scena, la quale incide sulla tonalità e sulla saturazione del colore della ricostruzione 3D.

# Considerazioni finali

Gli algoritmi di clustering sono sicuramente più robusti rispetto a quelli basati sul colore ma anche per questi è necessario fare attenzione alla tolleranza del cluster e alla dimensione minima e massima del cluster.

Infine, le potenzialità del `correspondence_grouping` sono notevoli, avendo dei modelli salvati su disco raffiguranti gli oggetti che mi aspetto di trovare in un determinato ambiente, il riconoscimento di tali oggetti nella scena 3D risulta veloce e preciso.

# Software e librerie utilizzate

- V-Rep:
  - [www.coppeliarobotics.com](http://www.coppeliarobotics.com)
- Ros indigo:
  - <http://wiki.ros.org/indigo>
- Orb-slam2:
  - [https://github.com/raulmur/ORB\\_SLAM2](https://github.com/raulmur/ORB_SLAM2)
- OpenMVG e OpenMVS:
  - <https://openmvg.readthedocs.io/en/latest/>
- PCL library:
  - <http://pointclouds.org/>
- Meshlab:
  - <http://www.meshlab.net/>