

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA
DIPARTIMENTO DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE
"GUGLIELMO MARCONI" DEI

Corso di Laurea in
Ingegneria Elettronica e Telecomunicazioni

Tesi di Laurea in
Calcolatori Elettronici

Strumento per annotazione semantica 3D con realtà virtuale

Candidato:
Luigi Lella

Relatore:
Ch.mo prof. **Luigi Di Stefano**
Correlatore:
Pierluigi Zama Ramirez

Anno Accademico 2019/2020

Sessione IV

Indice

Introduzione	1
1 Motivazioni alla base del lavoro	3
1.1 Machine Learning	3
1.1.1 Reti Neurali e Deep Learning	4
1.1.2 Il processo di apprendimento	5
1.1.3 Dataset	6
1.2 Segmentazione semantica	7
1.2.1 Fase di annotazione	9
2 Strumenti utilizzati	11
2.1 Realtà virtuale	11
2.1.1 Htc Vive Pro	12
2.2 Unity 3D	14
2.3 Blender	15
2.3.1 Modelli 3D	15
3 Shooting Labels	17
4 Risultati sperimentali	19
Conclusione	21
Bibliografia	24

Elenco delle figure

1.1	Rete Neurale	5
1.2	Processo di apprendimento	6
1.3	Matterport3d	7
1.4	Task principali della Computer Vision	8
1.5	Segmentazione semantica	9
2.1	Htc Vive	13
2.2	Mesh e Pointcloud	15

Introduzione

Negli ultimi anni machine learning e deep learning hanno rivoluzionato il mondo dell'intelligenza artificiale e della computer vision ottenendo risultati impressionanti in molti compiti come classificazione di immagini, segmentazione semantica ecc. superando, in alcuni casi, anche i risultati ottenuti da un utente umano in precisione e accuratezza [1].

Per usare tutti questi strumenti abbiamo bisogno di una grande quantità di dati al fine di supervisionare la parte di training. Per questa ragione sono stati creati innumerevoli dataset contenenti dati grezzi e le rispettive annotazioni come ad esempio KITTI [2], Matterport3D [3] e Replica [4].

La fase di annotazione, come si può facilmente immaginare, è però un processo lento, noioso e molto propenso ad errori. Labellare una singola immagine in 2D può richiedere alcune ore ed è un compito che richiede anche una certa esperienza con l'utilizzo di alcuni software. La difficoltà poi cresce se decido di lavorare con dati 3D come mesh o pointcloud.

Per semplificare e velocizzare questa parte si è deciso di realizzare *Shooting Labels* un tool per la segmentazione semantica 3D basato su realtà virtuale. L'utente viene trasportato all'interno di una scena rappresentata come mesh 3D, dove tutte le superfici possono essere "colorate" semanticamente. L'esperienza immersiva fornita dalle tecnologie VR consente all'utente di muoversi fisicamente all'interno dello scenario etichettare e interagire con gli oggetti in modo naturale e coinvolgente rendendo questo compito, oltre che più veloce, anche divertente come giocare a un video-game [5].

In questo elaborato si parlerà dapprima della segmentazione semantica e dei problemi di ottenere dati con cui allenare una rete, verranno poi presentati gli strumenti utilizzati nel corso della stesura della tesi e tirocinio per poi pas-

sare ad una trattazione dettagliata del software. Infine verranno analizzati e commentati alcuni dati ottenuti dall'utilizzo di questo strumento.

Capitolo 1

Motivazioni alla base del lavoro

1.1 Machine Learning

Negli ultimi decenni il Machine Learning si è imposto diventando uno dei pilastri della tecnologia e ricoprendo un ruolo piuttosto centrale, anche se nascosto, della nostra vita quotidiana. La crescita di questo settore è destinata a non fermarsi presto soprattutto grazie alla sempre maggiore quantità di dati che diventano disponibili.

L'idea di base è quella dell'*apprendimento automatico*, ovvero fornire ai computer l'abilità di imparare alcune operazioni senza essere esplicitamente programmati, sviluppando una propria logica basata solo ed esclusivamente su dei set di dati che vengono elaborati con degli algoritmi.

Le varie tecniche di apprendimento si classificano in tre diversi gruppi [6]:

- **Apprendimento supervisionato** i dati che vengono forniti al modello sono costituiti da una coppia di informazioni: un possibile input e il relativo output. Il sistema in questo caso deve estrarre una regola generale per associare ad ogni input il corrispettivo output corretto.
- **Apprendimento non supervisionato** vengono forniti al modello soltanto gli input senza i relativi output. Il sistema dovrà quindi identificare un pattern per cercare di effettuare ragionamenti e previsioni sugli input successivi. Le classi in questo caso dovranno essere apprese in maniera autonoma.

- **Apprendimento per rinforzo** non ci sono dati. Il modello deve interagire con un ambiente dinamico nel quale cerca di raggiungere un obiettivo (ad esempio il superamento di un livello in un videogioco o la guida di un veicolo) ricevendo soltanto una valutazione sulle sue azioni e non informazioni su quali azioni compiere e quando compierle.

1.1.1 Reti Neurali e Deep Learning

Tra i diversi approcci e tecniche per la progettazione e l'implementazione di sistemi per l'apprendimento automatico troviamo le *reti neurali* e il *deep learning*.

Una *rete neurale* o *neural network* è un sistema di apprendimento che utilizza una rete di funzioni per comprendere e tradurre un input di dati di una forma in un output desiderato, solitamente in un'altra forma. Una rete neurale è composta da elementi chiamati neuroni disposti su vari livelli, questi ricevono degli stimoli e li elaborano attivando o meno le loro uscite. Normalmente gli strati che formano la rete sono tre (vedi figura 1.1), il primo livello detto *input units* è composto dai neuroni che ricevono i segnali dall'ambiente esterno, il secondo strato, *hidden units*, può essere composto anche da più colonne di neuroni e si occupa dell'elaborazione vera e propria, l'ultimo strato invece *output units* fornisce la risposta della rete.

Il concetto di rete neurale artificiale ha preso ispirazione dalla biologia umana in particolare dal modo in cui funzionano e collaborano i neuroni del cervello umano per comprendere gli input sensoriali [7].

Il *Deep Learning*, o apprendimento approfondito, è il ramo più avanzato del Machine Learning. Si tratta di un insieme di tecniche basate su reti neurali artificiali organizzate in diversi strati: ogni strato calcola i valori per quello successivo, in modo da elaborare l'informazione in maniera sempre più completa [8]. Questo approccio trae ispirazione da alcune zone del cervello umano che si occupano della visione e dell'udito cercando di modellarne la struttura. Soprattutto per questo motivo il deep learning ha avuto maggior successo nel campo della *computer vision* e del riconoscimento vocale o *speech recognition*.

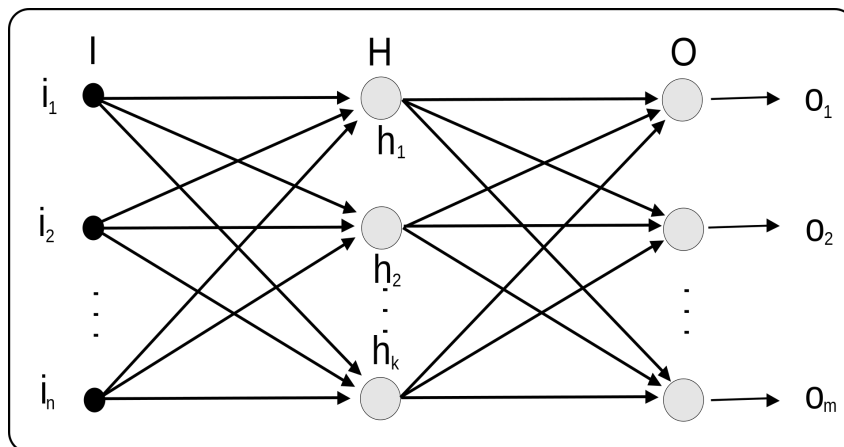


Figura 1.1: Struttura di una rete neurale generica

Se i dati a disposizione sono sufficienti il sistema è in grado di estrarre un modello corretto anche senza un pre-processamento dei dati, come invece avviene per le tecniche tradizionali di Machine Learning. In altre parole, il Deep Learning è una tecnica di apprendimento in cui reti neurali vengono esposte a vaste quantità di dati, in modo che possano imparare a svolgere compiti.

1.1.2 Il processo di apprendimento

Come detto nel primo paragrafo, gli algoritmi di apprendimento automatico che necessitano di dati per essere applicati sono quelli supervisionati e non supervisionati. La fase di apprendimento avviene con i seguenti passaggi [9]:

- **Pre-elaborazione:** i dati, per essere forniti come input al modello, devono essere trasformati pertanto la pre-elaborazione dei dati è uno dei passi cruciali di qualsiasi applicazione di apprendimento automatico. Dai dati grezzi possiamo estrarre alcune caratteristiche significative che possono essere, nel caso in cui i dati siano delle immagini, i colori, la tonalità, l'altezza la larghezza e la profondità. Molti algoritmi di apprendimento automatico richiedono anche che, per ottenere le massime prestazioni, le caratteristiche scelte adottino la stessa scala, il che spesso viene ottenuto trasformando le caratteristiche in un intervallo $[0, 1]$.

Per determinare la bontà del nostro algoritmo spesso si decide di suddividere il dataset in due set distinti, uno di addestramento e uno di test. Useremo il primo per informare e ottimizzare il modello mentre teniamo da parte il secondo per valutare il modello finale.

- **Inizializzazione del modello:** bisogna assegnare dei valori ai parametri del modello, questi possono essere scelti randomicamente oppure ereditati da altri modelli di deep learning.
- **Training:** ogni dato viene fornito in ingresso al modello il quale calcola una *score function* che assegna al dato diversi punteggi (nel caso di classificatore posso essere i valori di appartenenza alle sue categorie) in seguito viene valutata anche la differenza tra valori predetti e quelli reali.
- **Test:** la rete processa i dati di test e viene eseguita anche una valutazione del modello.

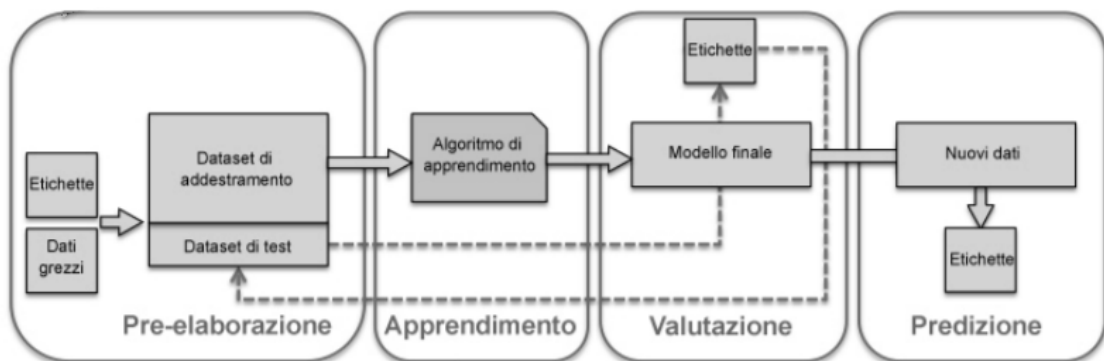


Figura 1.2: Tipico diagramma di flusso per l'impiego dell'apprendimento automatico

La presenza di dati è quindi fondamentale per supervisionare il training e questi devono essere forniti in quantità sufficiente o altrimenti, se il campione di dati è troppo piccolo o la qualità dei dati non rappresenta correttamente la realtà, i modelli di Machine Learning saranno inutilizzabili.

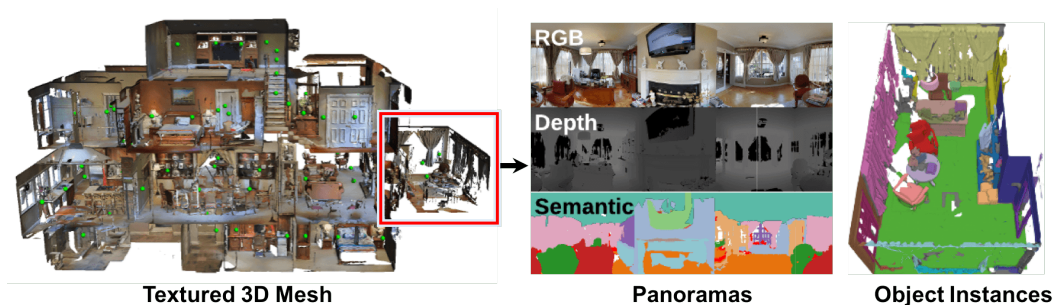


Figura 1.3: Il dataset Matterport

1.1.3 Dataset

Per questa ragione sono stati creati innumerevoli *datasets*, soprattutto negli ultimi anni, che raccolgono enormi quantità di dati specifici per un compito con le relative annotazioni.

Di seguito verranno presentati i dataset utilizzati come reference durante il lavoro.

- Il dataset di **Matterport3D** [3] (Figura 1.2) fornisce dati per il training riguardo scenari indoor. I dati sono stati prelevati da 90 edifici mantenendo anche una certa varietà di ambienti. Il dataset si compone di 194,400 immagini RGB-D ovvero combinazioni di immagini RGB con le loro *depth image* nelle quali ad ogni pixel viene anche associata la distanza tra piano dell'immagine e oggetto corrispondente. Vengono forniti anche i dati annotati semanticamente e per ogni scena anche una *mesh* dell'ambiente.
- **KITTI** fornisce invece un ampio set di dati riguardanti scenari outdoor. Comprende diverse categorie tra cui 'Road', 'City', 'Residential', 'Campus' e 'Person'. Per ogni sequenza vengono forniti dati grezzi, relative annotazioni e un file di calibrazione.

1.2 Segmentazione semantica

La *computer vision*, anche grazie al deep learning, negli ultimi anni ha visto accrescere sempre di più l'interesse nei suoi confronti arrivando al punto in cui

risulta essere molto più precisa dell'uomo in alcuni compiti [10]. I campi in cui questa risulta più usata sono la classificazione di immagini, il rilevamento di oggetti e la segmentazione elencati in ordine di difficoltà.

Se parliamo di *classificazione di immagini* siamo soltanto interessati ad ottenere le etichette di tutti gli oggetti presenti in un immagine. Quando invece abbiamo a che fare con l'*object detection* oltre che alla label vogliamo avere anche informazioni sulla posizione degli oggetti che ci viene fornita con dei bounding boxes ovvero dei riquadri che delimitano il soggetto. Facendo un ulteriore passo avanti in termini di precisione troviamo la *segmentazione semantica* che ha come obiettivo quello di rilevare il confine esatto degli elementi nell'immagine.

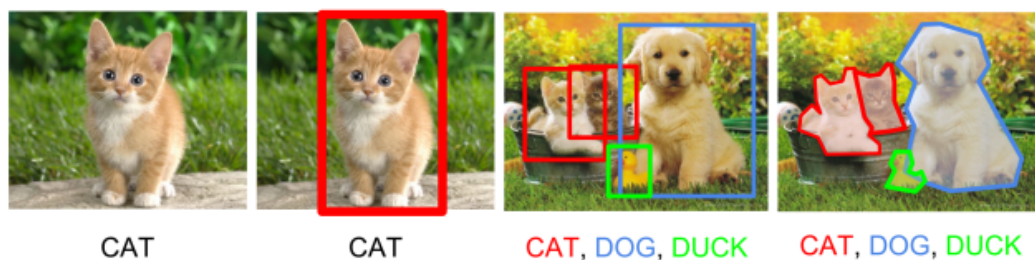


Figura 1.4: Sinistra: esempio di classificazione. Immagini centrali: esempi di object detection. Destra: esempio di segmentazione semantica

Con la *segmentazione semantica* quindi si vuole andare ad associare ad ogni pixel dell'immagine una categoria presa da un set prestabilito, nell'esempio di sinistra nella Figura 1.4 possiamo vedere che tutti i pixel appartenenti al gatto vengono colorati di giallo, tutti quelli appartenenti al terreno vengono colorati di verde, ecc. Ovviamente il numero di categorie da cui andare a pescare è arbitrario, la stessa immagine potrebbe essere segmentata anche in soltanto due categorie (una per il soggetto e una per lo sfondo).

La segmentazione semantica però non fa distinzione tra le varie istanze, nelle immagini di destra abbiamo due mucche, i pixel vengono assegnati indipendentemente e non c'è alcuna distinzione tra le due che però avviene se decidiamo di trattare l'immagine con un approccio di *instance segmentation*.

La definizione data in precedenza vale per immagini o video, questa però può

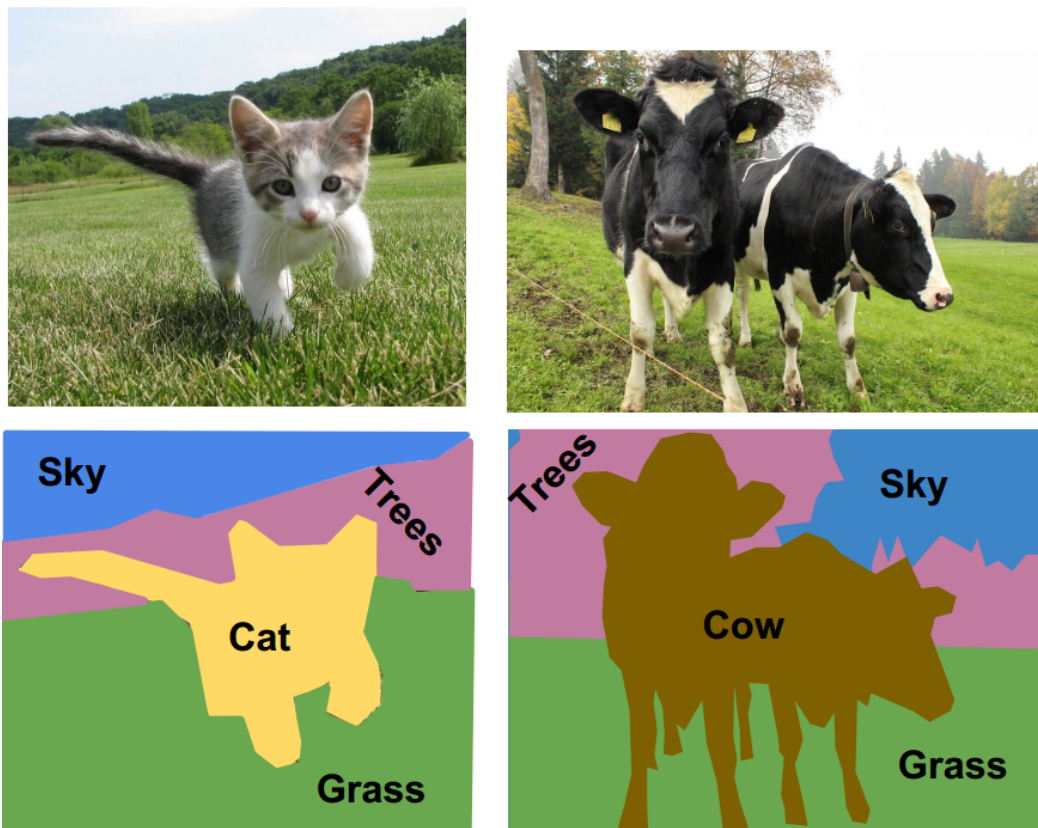


Figura 1.5: Esempio di segmentazione semantica su due immagini

essere estesa facilmente al caso di dati 3D. In caso di *point-cloud* si vuole assegnare una categoria ad ogni suo punto mentre, se i nostri dati sono sotto forma di *mesh*, la categoria va assegnata ad ogni faccia.

Essendo più precisa rispetto alle altre è particolarmente indicata per quelle applicazioni nei settori che richiedono un certo livello di affidabilità come ad esempio guida autonoma, rilevazione di difetti nei materiali e prodotti, imaging medicale e nella visione robotica per identificare e muoversi tra oggetti e terreni.

1.2.1 Fase di annotazione

La fase di annotazione è però un processo noioso, molto lento e inoltre richiede anche una certa dimestichezza con l'utilizzo di alcuni programmi. Questi

problemi si aggravano nel caso in cui decidiamo di lavorare in tre dimensioni dove si aggiunge anche la difficoltà di orientarsi in uno spazio tridimensionale. Al giorno d'oggi esistono due possibili strade da percorrere quando si decide di annotare dati per un *dataset* riguardante la segmentazione semantica 3D ovvero:

- **Utilizzare programmi open source** come ad esempio *Blender* o *MeshLab* che però richiedono una certa esperienza nell'utilizzo di motori grafici 3D.
- **Affidarsi a delle aziende** che mettono a disposizione dei tool per annotare dati o che si offrono di annotarli al tuo posto, come ad esempio *Scale* o *Dataloop*, dietro però compenso.

Per queste ragioni è nato *Shooting Labels* il primo strumento per l'annotazione semantica 3D che sfrutta la realtà virtuale per rendere più semplice e intuitiva questa fase. Utilizzando questo strumento il giocatore non necessita di nessuna competenza specifica ma indossando un headset è subito proiettato all'interno della scena nella quale utilizzando alcune armi "spara" i colori associati alle varie categorie sulle superfici etichettandole. Il lavoro diventa quindi accessibile a tutti e anche più veloce in quanto più immediata l'interazione con l'ambiente circostante rispetto all'utilizzo di altri software che non sfruttano la realtà virtuale.

Capitolo 2

Strumenti utilizzati

2.1 Realtà virtuale

Il concetto di realtà virtuale non è recente come si possa pensare, le prime testimonianze risalgono al 1960 con la *Telesphere Mask*, brevettata da Heiling, che era stata pensata come un apparato da collegare alla tv per simulare la sensazione di realtà muovendosi nelle tre dimensioni. Allora però non ebbe il successo sperato, il boom di questa tecnologia è arrivato solo nel 2012 con l'uscita dell'*Oculus Rift* che la ha resa accessibile a milioni di utenti in tutto il mondo.

Una definizione ufficiale di realtà virtuale non esiste ancora, le più accreditate sono però quella data da Isdale nel 1998 "È la simulazione di un ambiente reale o immaginario che può essere sperimentato visivamente nelle tre dimensioni e che fornisce inoltre un'esperienza interattiva con il suono e possibilmente anche con feedback tattili o di altre forme. La realtà virtuale è un modo per gli esseri umani di visualizzare, manipolare e interagire con computer e dati estremamente complessi" e quella data da Baieier nel 1993 "È un ambiente artificiale creato con hardware e software di computer e presentato a un utente in modo tale che appaia e si senta come un ambiente reale", pertanto la realtà virtuale si riferisce a un mondo 3D completamente immersivo e interattivo creato con l'utilizzo di un calcolatore che permette all'utente, grazie all'uso di un Head Mounted Display, di entrarvi e di diventare parte attiva di esso [11].

L'architettura necessaria per far funzionare il tutto si compone di un viso-

re con un campo visivo dai 100 ai 110 gradi e un frame rate compreso tra 60fps e 120fps dotato anche di sensori che permettono, insieme ad almeno due *base station*, l'*Head Tracking* ovvero lo spostamento dell'immagine in base alla posizione della testa. Per poter interagire effettivamente con il mondo circostante vengono utilizzati due controller, uno per mano, sempre rilevati dalle *base station*.

Spesso il concetto realtà virtuale viene confuso con quello di realtà aumentata; quest'ultima però è un potenziamento della percezione del mondo reale al quale vengono aggiunti dei contenuti digitali e che non necessita di hardware appositi per funzionare ma può essere applicata anche con l'utilizzo di smartphone o tablet. Un classico esempio di realtà aumentata è un famoso gioco Nintendo uscito nel 2016, *Pokemon Go*, nel quale i personaggi venivano proiettati nel mondo reale e il giocatore poteva interagire con loro e catturarli.

Anche se attualmente l'applicazione principale di questa tecnologia è nel campo videoludico ultimamente sta trovando spazio anche in altri campi come ad esempio quello dell'addestramento militare. *DoDAAM*, un'azienda coreana, ha già creato tutta una serie di simulazioni di situazioni di guerra come ad esempio il lancio da un aeroplano, la guida di un carro armato e di un jet riducendo contemporaneamente costi, tempi e rischi. Altre applicazioni si possono trovare nel ambito medico soprattutto nella terapia di disturbi psichiatrici e nella riabilitazione motoria e cognitiva. Le potenzialità della realtà virtuale possono essere usate anche nella formazione del personale medico simulando, ad esempio, un'operazione chirurgica.

2.1.1 Htc Vive Pro

Il sistema di realtà virtuale utilizzato durante la realizzazione del progetto è l'HTC Vive Pro, prodotto da HTC e Valve, (in Figura 2.1) e commercializzato a partire dal 2016. Il visore VIVE presenta due lenti Fresnel non perfettamente circolari ma con un lato piatto che le allinea con il naso permettendo anche di avvicinarle o allontanarle tra loro permettendo di adattarle alla propria distanza interpupillare. L'HMD presenta oltre ai 32 sensori infrarossi per



Figura 2.1: Visore e componenti

il tracking posizionati sulla superficie anche un giroscopio, un accelerometro e un sensore di posizione laser per riprodurre nella maniera più fedele possibile il movimento della testa.

I *controller* servono per interagire con l'ambiente e sono dotati di 24 sensori per il tracking oltre che a una serie di tasti.

Le *Base Station* sono degli emettitori, generano una luce ad infrarossi e due fasci laser che vengono rilevati dai sensori posti su controller e HMD. Basandosi poi sulla posizione di questi sensori e su quanto sono stati irradiati è possibile stimare con un'ottima precisione la posizione e l'orientamento degli oggetti rispetto alle base station.

Per la configurazione di questo strumento ci si avvale di un runtime chiamato *SteamVR* incluso nel client di Steam che consente di usufruire della realtà virtuale. SteamVR viene installato automaticamente quando Steam rileva la connessione di un visore al PC dell'utente, ma può essere installato anche manualmente.

SteamVR consente all'utente la gestione di molti aspetti della sua esperienza di gioco tra cui:

- Configurazione della stanza, con cui definire la propria area di gioco.
- Controllo stato e gestione del dispositivo, con cui aggiornare il firmware,

associare nuovi dispositivi, cambiare le impostazioni audio, impostare la duplicazione e personalizzare funzionalità quali il Motion Smoothing [12].

Attualmente è uno dei migliori dispositivi disponibili sul mercato e come tale per funzionare il pc su cui far girare il tutto deve soddisfare alcuni requisiti. Di seguito vengono riportati quelli consigliati presi direttamente dal sito ufficiale della Vive:

- **Processore** Intel Core i5 o superiore
- **GPU** NVIDIA GeForce GTX 1070/Quadro P5000, AMD Radeon Vega 56 o superiore
- **Memoria** 4Gb di RAM o più
- **Video Output** DisplayPort 1.2 o più recente
- **USB port** 1x USB 3.0 o più recente
- **O.S.** Windows 10

2.2 Unity 3D

Unity è un motore grafico sviluppato da Unity Technologies usato principalmente per sviluppare videogiochi e simulazioni in grafica 2D o 3D tramite scripting in C#. È un ambiente multi-piattaforma che permette la creazione di applicazioni per diversi dispositivi tra cui: Android, iOS, Windows, Playstation e sviluppo di software di realtà virtuale e aumentata. Unity nel suo Asset Store mette a disposizione tutta una serie di librerie, modelli, prefab e script sia gratuiti che a pagamento. Tra questi nel progetto viene utilizzato OpenVR ovvero un API che consente alle applicazioni di interagire con gli hardware di diversi produttori senza necessità di rilevare l'effettivo hardware puntato. Supporta diversi HMD tra cui Valve Index, HTC Vive, Oculus Rift, Windows Mixed Reality e altri.

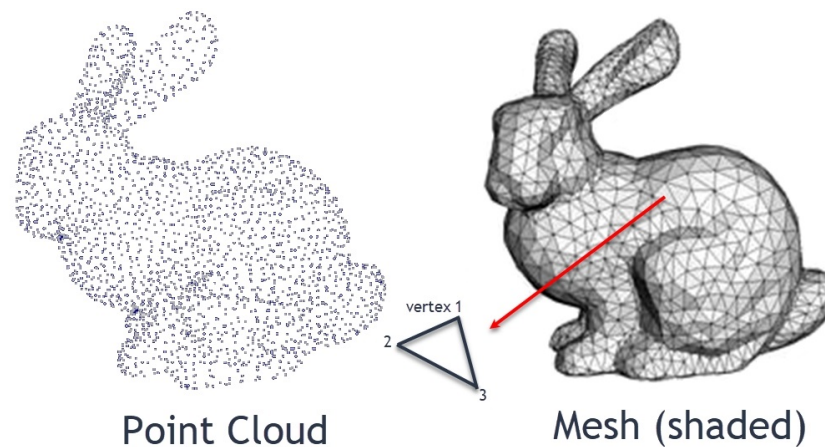


Figura 2.2: Esempio di mesh e pointcloud

2.3 Blender

Blender è una suite di creazione 3D gratuita e open source. Il software mette a disposizione diverse funzionalità tra cui modellazione 3D, texturing, simulazione di elementi come corpi rigidi, fluidi e fumi, animazione 3D, rendering ecc. Nel progetto servirà per la gestione di tutti i dati 3D in particolare per la fase di pre-process per creare la scena e nella fase di post-process per visualizzare e maneggiare i risultati.

2.3.1 Modelli 3D

Tra i vari modelli 3D che si possono trovare nella computer grafica e che durante la realizzazione del progetto sono stati particolarmente ricorrenti troviamo *mesh* e *pointcloud*.

Una mesh poligonale, detta anche maglia poligonale, è una collezione di vertici, spigoli e facce che definiscono la forma di un oggetto poliedrico. . Le facce consistono, di solito, di rettangoli, triangoli, o altri semplici poligoni convessi, dal momento che ci'ò semplifica il rendering.

Una mesh poligonale si compone di tre tipi di elementi:

- Il **vertice** è una posizione nello spazio e ha informazioni relative al colore e al vettore normale.
- Il **lato** è il collegamento tra due vertici.

- La **faccia** un insieme chiuso di lati, ad esempio una faccia triangolare avrà tre lati e una faccia quadrangolare avrà quattro lati. Un poligono è un insieme complanare di facce. Ogni faccia ha inoltre anche un vettore normale alla sua superficie.

Una point cloud, invece, è un insieme di punti nello spazio non dotato di informazioni topologiche (niente lati ne triangoli), questo tipo di dati proviene dall'utilizzo di scanner 3D o sensori come il Kinect.

Capitolo 3

Shooting Labels

Capitolo 4

Risultati sperimentali

Conclusione

Bibliografia

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NIPS*, 2012.
- [2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The international Journal of Robotic Research*, 2013.
- [3] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from rgb-d data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017.
- [4] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, “The replica dataset: A digital replica of indoor spaces,” *arXiv preprint arXiv:1906.05797*, 2019.
- [5] P. Z. Ramirez, C. Paternesi, L. D. Luigi, L. Lella, D. D. Gregorio, and L. D. Stefano, “Shooting labels: 3d semantic labeling by virtual reality,” *IEEE AIVR*, 2020.
- [6] S. Russel and P. Norving, *Artificial Intelligence: a modern approach*. Prentice Hall, 2003.
- [7] “Neural network.” <https://deeppai.org/machine-learning-glossary-and-terms/neural-network>.
- [8] “Alla scoperta del deep learning: significato, esempi e applicazioni.” https://blog.osservatori.net/it_it/deep-learning-significato-esempi-applicazioni.

- [9] S. Raschka, *Machine Learning con python*. Apogeo Editore, 2015.
- [10] R. Geirhos, D. H. J. Janssen, H. H. Schütt, J. Rauber, M. Bethge, and F. A. Wichmann, “Comparing deep neural networks against humans: object recognition when the signal gets weaker,” *arXiv*, 2017.
- [11] M. Onyesolu and U. Eze, *Understanding Virtual Reality Technology: Advances and Applications*. 2011.
- [12] “Stemvr documentazione.” <https://partner.steamgames.com/doc/features/steamvr/info>.