UNIVERSITA' POLITECNICA
DELLE MARCHE

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica e della Automazione

---

Corso di Computer Graphics e Multimedia

Prototipazione Rapida di Digital Twin tramite Mixed Reality e Deep Learning su Meta Quest

Professori:
Zingaretti Primo
Stacchio Lorenzo

Studente:
Marco Di Vita

ANNO ACCADEMICO 2024-2025

Indice

Abstract

Negli ultimi anni il concetto di Digital Twin ha assunto un ruolo centrale nei contesti industriali, ingegneristici e di ricerca, grazie alla possibilità di creare repliche digitali interattive di oggetti e sistemi fisici. Tuttavia, la creazione di modelli tridimensionali rappresenta ancora un processo complesso, spesso basato su modellazione manuale o su tecniche di scansione dedicate.

Il presente lavoro propone una pipeline end-to-end per la generazione automatica di modelli 3D a partire da immagini 2D, integrando tecniche di Deep Learning per la segmentazione e la ricostruzione tridimensionale con un ambiente di visualizzazione e interazione sviluppato in Unity e basato su tecnologie Extended Reality (XR).

La soluzione implementata combina il modello Segment Anything Model (SAM) per l'estrazione automatica dell'oggetto dall'immagine e Stable Fast 3D per la generazione della mesh tridimensionale, orchestrati da un backend Python basato su Flask. Il modello generato viene caricato in Unity e reso interagibile tramite Meta XR SDK.

1. Introduzione

Il concetto di Digital Twin rappresenta una delle evoluzioni più significative nell'ambito della digitalizzazione dei sistemi fisici. Un Digital Twin può essere definito come una replica virtuale dinamica di un oggetto, processo o sistema reale, progettata per simulare, analizzare e interagire con la controparte fisica.

Tradizionalmente, la creazione di modelli tridimensionali richiede competenze avanzate in modellazione 3D o l'impiego di strumenti di scansione specializzati, come scanner LiDAR o sistemi fotogrammetrici. Questi approcci comportano costi elevati, tempi di produzione significativi e limitazioni in termini di scalabilità.

Parallelamente, i recenti progressi nel campo del Deep Learning hanno introdotto modelli in grado di affrontare problemi complessi di computer vision e generazione tridimensionale. In particolare, i foundation models per la segmentazione e le architetture neurali per la generazione 3D hanno aperto nuove prospettive nella ricostruzione automatica di oggetti a partire da immagini bidimensionali.

2. Stato dell'Arte

2.1 Digital Twin

Il termine Digital Twin è stato formalizzato nei primi anni 2000 in ambito industriale e aerospaziale. Un Digital Twin non è semplicemente un modello 3D statico, ma una rappresentazione digitale che può incorporare dati, simulazioni e interazioni dinamiche.

Le principali applicazioni includono:

- manutenzione predittiva;
- simulazione di processi industriali;
- monitoraggio in tempo reale;
- formazione immersiva.

Nonostante i progressi, la creazione di un Digital Twin rimane fortemente dipendente dalla disponibilità di modelli tridimensionali accurati, che costituiscono uno dei principali colli di bottiglia del processo.

2.2 Segmentazione di Immagini

La segmentazione è il processo di suddivisione di un'immagine in regioni significative.

Con l'avvento delle reti neurali convoluzionali (CNN), modelli come U-Net e Mask R-CNN hanno migliorato significativamente la qualità della segmentazione semantica e istanza.

Recentemente, l'introduzione dei Vision Transformer (ViT) ha portato allo sviluppo di modelli foundation capaci di generalizzare su ampie varietà di immagini. Il Segment Anything Model (SAM) rappresenta uno di questi modelli, progettato per segmentare oggetti senza necessità di riaddestramento specifico.

2.3 Generazione 3D da Immagini 2D

La ricostruzione tridimensionale da una singola immagine monoculare costituisce un problema.

Le moderne tecniche basate su Deep Learning utilizzano architetture generative e diffusion models per inferire la geometria tridimensionale plausibile a partire da input bidimensionali.

Stable Fast 3D, il modello usato nel progetto, si inserisce in questo contesto come soluzione orientata alla generazione rapida di mesh tridimensionali coerenti, privilegiando velocità di esecuzione e semplicità di integrazione rispetto a pipeline complesse multi-stage.

2.4 Extended Reality e Interazione

Le tecnologie XR (Extended Reality) comprendono realtà virtuale (VR), realtà aumentata (AR) e mixed reality (MR).

L'integrazione con motori grafici come Unity e con SDK specifici (come Meta XR SDK) permette di rendere i modelli tridimensionali non solo visualizzabili, ma anche interagibili tramite controller o hand tracking.

3. Architettura del Sistema

Il sistema implementato adotta un'architettura client-server con separazione netta tra frontend e backend.

3.1 Frontend

Il frontend è sviluppato in Unity ed è responsabile di:

- interfaccia utente;
- gestione delle richieste verso il backend;
- caricamento dinamico dei modelli GLB;
- interazione XR.

3.2 Backend

Il backend è sviluppato in Python tramite Flask e svolge il ruolo di orchestratore:

- carica il modello SAM all'avvio;
- gestisce l'endpoint di segmentazione;
- esegue Stable Fast 3D tramite subprocess;
- restituisce il percorso del modello generato.

La comunicazione avviene tramite richieste HTTP REST, con scambio di file in formato PNG e GLB.

4. Backend: Segmentazione e Generazione 3D

Il backend costituisce il nucleo computazionale del sistema ed è responsabile dell'esecuzione delle componenti di Deep Learning e della gestione della pipeline automatizzata. È stato sviluppato in Python utilizzando il framework Flask, scelto per la sua leggerezza e flessibilità nella creazione di API REST.

L'architettura del backend è modulare e suddivisa in tre componenti principali:

1. Segmentazione automatica tramite SAM
2. Generazione della mesh tridimensionale tramite Stable Fast 3D
3. Orchestrazione e gestione delle richieste tramite Flask

4.1 Segment Anything Model (SAM)

Il Segment Anything Model è un foundation model basato su architettura Vision Transformer (ViT), progettato per eseguire segmentazione generalizzata su una vasta gamma di immagini.

Nel sistema proposto, SAM viene caricato all'avvio del server per evitare overhead ripetuti di inizializzazione. Il modello utilizza un approccio automatico alla generazione delle maschere tramite SamAutomaticMaskGenerator.

Il flusso operativo della segmentazione è il seguente:

1. Caricamento dell'immagine fornita dall'utente
2. Generazione di un insieme di maschere candidate
3. Selezione della maschera con area maggiore
4. Post-processing morfologico (operazioni di apertura e chiusura)
5. Generazione di un'immagine con sfondo trasparente

La scelta della maschera più grande è motivata dall'assunzione che l'oggetto principale dell'immagine rappresenti l'elemento di interesse per la ricostruzione tridimensionale.

L'output finale è un'immagine PNG con canale alpha, che costituisce l'input per la fase di generazione 3D.

4.2 Stable Fast 3D

Stable Fast 3D è un modello neurale per la generazione di mesh tridimensionali a partire da immagini bidimensionali. Rispetto a pipeline più complesse multi-view, questa soluzione consente una generazione diretta e relativamente veloce di una mesh plausibile.

Nel sistema implementato, Stable Fast 3D viene eseguito tramite chiamata subprocess a partire dal backend Flask. Questo approccio consente:

- isolamento dell'ambiente del modello
- compatibilità con differenti versioni di Python
- separazione tra logica applicativa e generativa

Il flusso di generazione prevede:

1. Creazione di una directory temporanea univoca
2. Esecuzione dello script run.py con parametri:
    - path dell'immagine segmentata
    - directory di output
    - selezione del dispositivo (CPU o GPU)
3. Ricerca del file GLB generato
4. Spostamento del modello nella directory statica servita dal backend

Il formato GLB è stato scelto per la sua compatibilità nativa con Unity e per la possibilità di includere geometria e materiali in un singolo file binario.

4.3 Orchestrazione con Flask

Il backend espone due endpoint principali:

/segment

Riceve un'immagine tramite POST multipart/form-data, esegue la segmentazione con SAM e restituisce i percorsi dei file generati.

/generate3d

Riceve il nome dell'immagine segmentata, avvia Stable Fast 3D e restituisce l'URL del modello GLB generato.

La scelta di un'architettura REST consente:

- disaccoppiamento tra frontend e backend
- possibilità di estensione verso altri client
- scalabilità futura in ambienti distribuiti

5. Frontend: Unity e Interazione XR

Il frontend è stato sviluppato utilizzando Unity, uno dei motori grafici più diffusi in ambito real-time 3D e XR.

5.1 Interfaccia Utente

L'interfaccia consente all'utente di:

- selezionare un'immagine locale
- avviare la segmentazione
- avviare la generazione 3D
- caricare dinamicamente il modello GLB

Le richieste verso il backend vengono effettuate tramite UnityWebRequest, utilizzando chiamate HTTP asincrone per evitare il blocco del thread principale.

5.2 Caricamento Dinamico del Modello

Il caricamento del file GLB avviene tramite la libreria GLTFast. Il modello viene istanziato all'interno di un GameObject predefinito nella gerarchia, evitando la distruzione di prefab XR già configurati.

Questo approccio consente di:

- preservare componenti di interazione
- mantenere configurazioni collider e rigidbody
- sostituire dinamicamente solo la mesh

5.3 Integrazione con Meta XR SDK

L'interazione in ambiente XR è resa possibile tramite Meta XR SDK. Il modello caricato diventa immediatamente manipolabile grazie alla presenza di:

- collider
- rigidbody
- componenti di grab

Particolare attenzione è stata posta alla gestione degli eventi di input XR. È stato implementato un meccanismo di protezione per frame (Frame Guard) per evitare la doppia esecuzione degli eventi dovuta alla sovrapposizione tra input "Submit" e "Click".'

6. Risultati Sperimentali

La pipeline è stata testata su diverse immagini contenenti oggetti singoli ben definiti.

I principali risultati osservati includono:

- corretta segmentazione automatica nella maggior parte dei casi
- generazione di mesh tridimensionali plausibili
- compatibilità immediata con Unity
- interazione XR fluida

7. Limiti

Nonostante i risultati positivi, il sistema presenta alcune limitazioni:

- qualità della mesh dipendente dalla qualità dell'immagine
- assenza di gestione multi-oggetto
- texturing non sempre realistico
- tempi di generazione elevati su hardware non accelerato

Inoltre, la ricostruzione monoculare introduce ambiguità geometriche inevitabili.

8. Conclusioni

Il lavoro ha dimostrato la fattibilità di una pipeline completamente automatizzata per la generazione di Digital Twin 3D a partire da immagini 2D.

L'integrazione di modelli di Deep Learning con un ambiente di Computer Graphics real-time consente di ridurre significativamente la complessità del processo di creazione di asset tridimensionali.

Il sistema sviluppato rappresenta una base solida per future estensioni in ambito industriale, formativo e di ricerca.

G. Bibliografia

[1] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in *Transdisciplinary Perspectives on Complex Systems*, Springer, 2017.

[2] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[3] A. Kirillov et al., "Segment Anything," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[4] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *International Conference on Learning Representations (ICLR)*, 2021.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *MICCAI*, 2015.

[6] T.-Y. Lin et al., "Mask R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[7] B. Mildenhall et al., "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," *ECCV*, 2020.

[8] J. Poole et al., "DreamFusion: Text-to-3D using 2D Diffusion," *arXiv preprint arXiv:220S.14S88*, 2022.

[9] Stability AI, "Stable Fast 3D," GitHub Repository, 2024. [Online]. Available: https://github.com/Stability-AI/stable-fast-3d

[10] Unity Technologies, "Unity Real-Time Development Platform," 2024. [Online]. Available: https://unity.com

[11] Meta Platforms, "Meta XR SDK for Unity," 2024. [Online]. Available: https://developers.meta.com

[12] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.027C7*, 2018.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[14] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2022.