

APPROCCI DEEP LEARNING SU NUVOLE DI PUNTI SINTETICHE PER LA SEGMENTAZIONE SEMANTICA DI EDIFICI MONUMENTALI STORICI

LEARNING FROM SYNTHETIC POINT CLOUD DATA FOR HISTORICAL BUILDINGS SEMANTIC SEGMENTATION

C. Morbidoni^a, R. Pierdicca^b, M. Paolanti^a, R. Quattrini^b, R. Mammoli^b

^a Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60100 Ancona, Italy; c.morbidoni@univpm.it (C.M.); m.paolanti@univpm.it (M.P.)

^b Dipartimento di Ingegneria Civile, Edile e dell'Architettura, Università Politecnica delle Marche, 60100 Ancona, Italy; r.pierdicca@univpm.it (R.P.); r.quattrini@univpm.it (R.Q.); r.mamoli@univpm.it (R.M.)

PAROLE CHIAVE: Segmentazione semantica, nuvole di punti, dataset sintetici, apprendimento profondo

KEY WORDS: Semantic segmentation, points clouds, synthetic dataset, deep learning

RIASSUNTO

Nello scenario del patrimonio storico, la segmentazione semantica delle nuvole di punti 3D sta ottenendo sempre più attenzione, poiché permette di riconoscere automaticamente gli elementi architettonici storici. Tuttavia, la segmentazione semantica è particolarmente interessante nell'architettura storica e classica, a causa della complessità delle forme e della limitata ripetibilità di elementi in edifici diversi, che rende difficile definire modelli comuni all'interno della stessa classe architettonica. Inoltre, poiché i modelli di apprendimento profondo richiedono una quantità considerevolmente grande di dati annotati per essere addestrati al fine di gestire adeguatamente nuove scene, la mancanza di (grandi) nuvole di punti annotate pubblicamente disponibili nel dominio degli edifici storici è un problema enorme, tanto da costituire un problema aperto nella ricerca. D'altra parte, la creazione di una massa critica di nuvole di punti etichettate mediante annotazione manuale è molto dispendiosa in termini di tempo e poco pratica. A tal fine, in questo lavoro viene esplorata l'idea di sfruttare i dati sintetici delle nuvole di punti per addestrare modelli di apprendimento profondo che eseguono la segmentazione semantica delle nuvole di punti. Per raggiungere questo scopo, viene presentata una versione migliorata della Dynamic Graph CNN (DGCNN) denominata RadDGCNN. Negli esperimenti presentati in questo lavoro, i modelli vengono addestrati su dataset sintetici (disponibili pubblicamente) di due diversi edifici storici: il Palazzo Ducale a Urbino e il Palazzo Ferretti ad Ancona, entrambi in Italia, acquisiti tramite Laser Scanner Terrestre. La rete RadDGCNN produce buoni risultati, dimostrando migliori prestazioni di segmentazione sui dataset reali.

ABSTRACT

In the historical heritage scenario, semantic segmentation of 3D point clouds is gaining more and more attention, since it might help to automatically recognize historical architectural elements. However, semantic segmentation is particularly challenging in historical and classical architecture, due to the shapes complexity and the limited repeatability of elements across different buildings, which makes it difficult to define common patterns within the same class. Furthermore, as deep learning models requires a considerably large amount of annotated data to be trained and tuned in order to properly handle unseen scenes, the lack of (big) publicly available annotated point clouds in the historical building domain is a huge problem, blocking the research in this direction. On the other hand, creating a critical mass of annotated point clouds by manual annotation is very time-consuming and impractical.

For this purpose, in this work we explore the idea of leveraging synthetic point cloud data to train deep learning models to perform semantic segmentation of point clouds obtained via Terrestrial Laser Scanning (TLS). To achieve this purpose, we present an improved version of the Dynamic Graph CNN (DGCNN) named RadDGCNN. In our experiments, we evaluate the trained models on synthetic dataset (publicly available) about two different historical buildings: the Ducal Palace in Urbino, Italy, and Palazzo Ferretti in Ancona, Italy. RadDGCNN yields good results, demonstrating improved segmentation performances on the TLS real datasets.

1. INTRODUZIONE

Nel contesto della gestione e conservazione del patrimonio culturale (CH), la creazione di rappresentazioni digitali accurate e ricche di edifici storici è di primaria importanza. Diversi lavori di ricerca studiano come ottenere informazioni accurate e affidabili su un edificio storico attraverso l'acquisizione della realtà e la modellazione 3D per supportare la documentazione dell'architettura ed il progetto di restauro. Il patrimonio storico richiede elaborazioni robuste per ottenere modelli HBIM (Heritage Building Information Modeling) che siano completamente interoperabili e ricchi del loro contenuto

informativo. La disponibilità di flussi di lavoro efficienti potrebbe rappresentare un passo molto importante verso una gestione più efficiente del patrimonio architettonico.

Il processo di documentazione del patrimonio parte solitamente dall'acquisizione di nuvole di punti 3D attraverso tecniche di scansione laser terrestre (TLS) o fotogrammetriche che producono nuvole di punti accurate e precise, fino a milioni di punti. I metodi di acquisizione fotogrammetrica e TLS consentono di generare una grande quantità di coordinate geometriche e 3D di una scena. Tuttavia, mentre le nuvole di punti sono utili per visualizzare una data scena, sono principalmente rappresentazioni non strutturate e carenti di

informazioni sui singoli oggetti, come superfici, dimensioni e semantica. Per questo motivo, le nuvole di punti sono spesso utilizzate, all'interno di ambienti software BIM dedicati, come punto di partenza per costruire rappresentazioni 3D parametriche, che incorporano informazioni semantiche e dove gli elementi architettonici vengono identificati e adeguatamente etichettati. Tale processo è spesso indicato come *Scan-to-BIM*.

La creazione di modelli 3D strutturati da nuvole di punti è un processo manuale che richiede tempo. Richiede agli esperti di dominio di ispezionare visivamente la nuvola di punti identificando diversi tipi di elementi architettonici di interesse, per poi modellare gli oggetti, posizionare correttamente ogni elemento sulla scena e annotarlo secondo i dizionari di dominio e/o le ontologie. Pertanto, vi è una forte necessità di automatizzare parzialmente il processo *Scan-to-BIM*, sviluppando metodologie e strumenti che siano in grado di assistere gli esperti del settore, rendendo l'intero processo più agile e veloce.

Un compito importante in questa direzione è quello di identificare automaticamente i diversi tipi di elementi architettonici nella nuvola di punti. Questa attività viene definita segmentazione semantica e implica la classificazione di ogni punto nella nuvola come appartenente a un particolare tipo di oggetto (ad esempio muro, tetto, colonna, volta, ecc.). Accelerare il processo *Scan-to-BIM*, attualmente eseguito manualmente, è cruciale, ma la segmentazione automatica dei dati 3D è ancora un agli albori per il Digital Cultural Heritage.

Mentre le tecniche di machine e deep learning si sono dimostrate efficaci nel segmentare semanticamente oggetti 2D (cioè immagini), nel caso di nuvole di punti 3D, essendo queste configurazioni geometriche di natura irregolare, caratterizzate dalla mancanza di una griglia ordinata, con un'elevata variabilità di densità, non strutturata e invariante alla trasformazione e alla permutazione, il compito è più difficile.

Recentemente, sono stati raggiunti risultati promettenti mediante l'applicazione di tecniche di deep learning specificamente progettate per gestire le nuvole di punti (Qi et al., 2017a; Qi et al. 2017b; Wang et al., 2019b).

Tali approcci sono stati applicati alla segmentazione di scene di interni (Armeni et al., 2016), cioè uffici, e valutati su benchmark standard, come Stanford 2D-3D-Semantics Dataset¹. Tuttavia, la segmentazione semantica è particolarmente impegnativa nell'architettura storica, a causa della complessità delle forme e della limitata ripetibilità di elementi in edifici diversi, rendendo difficile definire modelli comuni all'interno della stessa classe di elementi (Malinverni et al., 2019; Pierdicca et al., 2020).

Anche se la forma è ripetibile, (e.g. per un dato stile architettonico), gli oggetti sono comunque unici in quanto realizzati artigianalmente e non serializzati.

Inoltre, poiché i modelli di deep learning richiedono una quantità considerevolmente grande di dati annotati per essere addestrati al fine di gestire correttamente le scene non viste, la mancanza di (grandi) nuvole di punti annotate pubblicamente disponibili è un enorme problema, che di fatto rallenta la ricerca in questa direzione. D'altra parte, la creazione di una massa critica di nuvole di punti annotate mediante annotazione manuale è molto dispendiosa in termini di tempo e poco pratica. Per affrontare questo problema, in questo lavoro viene esplorata l'idea di sfruttare i dati sintetici delle nuvole di punti per addestrare modelli di deep learning dello stato dell'arte al fine di eseguire la segmentazione semantica di nuvole di punti TLS reali. Stante la letteratura di riferimento nel settore, questo è il primo tentativo in questa direzione.

Deriviamo nuvole di punti sintetiche annotate da modelli 3D esistenti con l'obiettivo di ottenere una massa critica di dati da cui apprendere, senza la necessità di annotazioni manuali dispendiose in termini di tempo, di enormi quantità di dati TLS. Sebbene le nuvole di punti sintetiche siano relativamente facili da ottenere, sono intrinsecamente diverse dai dati TLS reali, poiché hanno una struttura più regolare, non considerano gli errori accidentali degli strumenti topografici e, in genere, catturano un livello di dettaglio inferiore.

Pertanto, lo scopo della presente ricerca è fornire una prima valutazione dell'uso di dati sintetici per guidare la segmentazione semantica basata sul deep learning nel contesto degli edifici storici.

I principali contributi di questo articolo sono i seguenti:

1. Fornire il primo studio per valutare l'uso di dati sintetici di nuvole di punti nel contesto della segmentazione semantica del patrimonio costruito, valutando modelli DGCNN addestrati su dati TLS reali.
2. Rilasciare un dataset composto da nuvole di punti sintetiche annotate derivate da modelli 3D di 10 diversi edifici storici e da 159 modelli 3D di singoli elementi architettonici raccolti da archivi web aperti. Il dataset² è disponibile su richiesta agli autori e può essere utilizzato per ulteriori ricerche in questo campo.
3. Proporre una variazione dell'architettura DGCNN presentata in (Wang et al., 2019b), basata sull'uso della *radius distance*, che ha dimostrato di aumentare le prestazioni sui nostri dataset di test.

Il contributo è organizzato come segue: la Sezione 2 fornisce una panoramica del lavoro precedente riguardante i recenti sforzi nella segmentazione semantica delle nuvole di punti basata su deep learning e la generazione e l'utilizzo di dati sintetici. Nella Sezione 3 vengono descritti i dati e i metodi usati in questo lavoro. Nelle Sezioni 4 e 5 sono presentati rispettivamente gli esperimenti condotti e i loro risultati. Infine, nella Sezione 6 sono tratte le conclusioni.

2. STATO DELL'ARTE

In letteratura, l'uso di dataset sintetici coinvolge diversi campi di applicazione. Il primo lavoro risale al 1975 quando Miller (Miller and Melton, 1975) generò dati di test appropriati per ottenere il miglioramento della qualità del software a breve termine utilizzando una metodologia che mette in relazione i dati di test del software funzionale con le sue specifiche formali per ottenere una corrispondenza tra il software e le sue specifiche.

Myers (1999) ha mostrato che l'introduzione del dataset simulato porta diversi vantaggi. Ha sviluppato un software per lo studio del genoma umano e un simulatore (Celsim) che permette di descrivere e generare stocasticamente una sequenza di DNA bersaglio con diverse strutture ripetute. La generazione di dati sintetici multidimensionali è stata applicata anche al clustering e all'analisi dei valori anomali in (Pei and Zaïane, 2006).

Le immagini termiche sintetiche vengono utilizzate per aumentare un dataset esistente di immagini visibili nel lavoro di Kniaz et al. (2017). Per aumentare il dataset esistente con immagini termiche sintetiche, hanno sviluppato una rete neurale convoluzionale profonda, denominata Thermalnet, ispirata alla colorizzazione delle reti neurali profonde.

Partendo da una singola immagine bidimensionale, Fan et al. (2017) hanno ricostruito la geometria 3D dell'oggetto completo. Hanno usato un algoritmo di deep learning per generare la

¹ <http://buildingparser.stanford.edu/dataset.html>

² <http://vrai.dii.univpm.it/content/synthetic-dataset-point-cloud-semantic-segmentation>

rappresentazione della nuvola di punti 3D, prevedendo la forma 3D reale dell'oggetto.

Li et al. (2016) hanno proposto un approccio basato sull'apprendimento supervisionato per classificare scene urbane reali 3D acquisite da Terrestrial, Vehicle e Aerial Laser Scanner TLS, VLS e ALS. La fase di addestramento ha utilizzato un dataset di generazione automatica di nuvole di punti 3D. In (Ros et al., 2016) un dataset di immagini sintetiche di scene urbane, estratto automaticamente da scene urbane virtuali, è stato presentato e mostrato per migliorare le prestazioni di segmentazione semantica 2D e aggiunto al dataset di benchmark annotato manualmente durante la fase di addestramento. Nel lavoro di (Wu et al., 2019), nuvole di punti sintetiche realistiche di scene stradali sono derivate da un popolare videogioco (Grand Theft Auto) e utilizzate per aumentare il dataset di riferimento KITTI (Geiger et al., 2012), migliorando le prestazioni della segmentazione semantica basata sulla rete neurale convoluzionale CNN (Convolutional Neural Network).

In Wang et al. (2019a) sensori LiDAR virtuali sono stati utilizzati per acquisire dati sintetici di nuvole di punti da scene di strade, simulando una varietà di strumentazione per l'acquisizione di nuvole di punti. Gli esperimenti sulla segmentazione semantica nel contesto della guida autonoma hanno dimostrato che le prestazioni aumentano grazie all'aumento sintetico dei dati annotati manualmente.

Secondo gli autori (Jaderberg et al., 2014) un dataset sintetico di testo generato usando un motore è una soluzione ottimale per sostituire i dati reali, fornendo un'enorme quantità di dati usati per l'addestramento. Essi hanno usato un dataset sintetico per addestrare una CNN, e al fine di testare la validità dell'approccio, hanno utilizzato un dataset disponibile pubblicamente. Simile è il lavoro di Wang et al. (2012) che ha addestrato una CNN usando esempi di testo generati sinteticamente.

Il lavoro di (Griffiths and Boehm, 2019) ha utilizzato un approccio basato sull'apprendimento profondo per la classificazione automatica dei dati delle nuvole di punti 3D. A causa della necessità di una grande quantità di dati di addestramento, gli autori hanno generato SynthCity, un dataset sintetico di nuvole di punti in un ambiente urbano/suburbano utilizzando il plugin Blensor per Blender. Il loro scopo è dimostrare che una rete addestrata utilizzando dati sintetici è in grado di generalizzare bene.

Nell'ambito del patrimonio culturale i dataset sintetici non sono spesso utilizzati; tuttavia, vengono qui presentati alcuni lavori significativi (Pierdicca et al., 2019).

Martínek et al. (2019) mostra una panoramica degli approcci di addestramento per il riconoscimento ottico dei caratteri nei documenti storici. Poiché il problema principale è la mancanza di dati annotati, riassumono diversi modi per preparare i dati sintetici. Hanno addestrato un classificatore di rete neurale ricorrente convoluzionale utilizzando un dataset sintetico e convalidato il loro approccio con un dataset reale annotato.

In un altro lavoro si è utilizzato un grande dataset renderizzato sinteticamente che presenta scene con forme generate e reali per addestrare Relight-Net, una profonda rete neurale convoluzionale di (Xu et al., 2018).

Un approccio che ha utilizzato un dataset sintetico è quello proposto da (Tran and Khoshelham, 2019), dove viene presentato un confronto tra un modello di informazioni edilizie e una nuvola di punti di un ambiente interno. Usano dataset sintetici e dataset di riferimento ISPRS per valutare le prestazioni nel rilevamento delle differenze tra le due rappresentazioni.

Sebbene lo scopo di questo lavoro non è quello di fornire una revisione della letteratura sugli approcci deep learning per la segmentazione semantica delle nuvole di punti dense, vale la

pena citare alcune architetture proposte di recente. I lettori interessati possono trovare una discussione più ampia sull'argomento in (Griffith and Boehm, 2019).

Le reti neurali profonde dello stato dell'arte sono progettate specificamente per affrontare l'irregolarità delle nuvole di punti, gestendo direttamente i dati grezzi delle nuvole di punti, piuttosto che passare a una rappresentazione regolare intermedia.

Tra gli altri, PointNet (Qi et al., 2017a) è il pioniere di questo approccio. Questa rete ottiene l'invarianza di permutazione dei punti operando indipendentemente su ogni punto, e successivamente applicando una funzione simmetrica per riunire le caratteristiche. L'estensione di PointNet (Qi et al., 2017b) tiene conto dei punti vicini, piuttosto che agire su ciascuno separatamente. Questo consente lo sfruttamento delle caratteristiche locali, migliorando così le prestazioni del modello di base.

PCNN (Atzmon et al., 2018) è un framework di apprendimento profondo per l'applicazione della CNN alle nuvole di punti. La convoluzione della nuvola di punti è definita da un pull-back della convoluzione volumetrica euclidea tramite un meccanismo di restrizione di estensione.

DGCNN, Dynamic Graph Convolutional Neural Networks (Wang et al. 2019b), introduce l'operazione EdgeConv. EdgeConv è un modulo che descrive le relazioni tra un punto e i suoi vicini. Questo modulo è progettato per essere invariante all'ordinamento dei vicini ed alle permutazioni. Nei nostri esperimenti ci basiamo su DGCNN e valutiamo i modelli addestrati su due nuvole di punti TLS acquisite da due diversi edifici storici: il Palazzo Ducale di Urbino e Palazzo Ferretti in Ancona.

3. MATERIALI E METODI

3.1 Classi degli elementi architettonici

Il primo passo fondamentale per la metodologia qui illustrata è identificare le classi di elementi architettonici da considerare per il compito di segmentazione semantica. Le classi selezionate per gli esperimenti sono le seguenti: Colonna, Finestra/Porta, Muro, Pilastro, Pavimento, Modanatura, Volta, altro. Le classi e il loro livello di dettaglio (LOD) sono stati scelti tenendo conto di tre aspetti: coerenza con il dizionario esistente, coerenza con il periodo storico e il suo linguaggio e corrispondenza con i dati acquisiti tramite TLS.

Per quanto riguarda il dettaglio scelto, si è fatto riferimento alle suddivisioni ontologiche proposte da robusti dizionari, disponibili in rete, relativi al campo dell'arte e dell'architettura. Per questo lavoro il riferimento principale è il vocabolario degli elementi architettonici classici e le loro gerarchie sviluppate dal Getty Museum – Research³. Il livello gerarchico della divisione in classi è stabilito tra elementi strutturali e componenti decorativi. Il grado di degrado dell'edificio consente di avere una piena corrispondenza con questo primo test condotto. Infatti, l'accuratezza attesa della nuvola di punti, in accordo con la letteratura di riferimento nonostante sia utilizzata in altri domini, è conforme al livello generale di dettaglio degli elementi architettonici.

Le classi scelte, inoltre, sono coerenti con i principali elementi del linguaggio architettonico classico, massicciamente utilizzato negli edifici rinascimentali (presi come casi studio di riferimento).

Per alcune classi semantiche è stato necessario implementare il dataset disponibile con alcuni modelli 3D online.

³<http://www.getty.edu/research/tools/vocabularies/aat/>.

Le classi architettoniche sono inoltre conformi ai dati acquisiti tramite TLS.

In relazione a questa metodologia di indagine e al fine di mantenere la massima uniformità ai dati acquisiti, alcune classi (come il tetto) non sono state prese in considerazione pur essendo nei vocabolari e nel linguaggio storico architettonico.

Come nel caso di qualsiasi architettura storica di grande valore e complessità, nonostante l'ampiezza delle classi, alcuni elementi non sono stati inclusi in nessuna di queste. Tutti gli elementi architettonici che non corrispondevano a nessuna delle classi selezionate sono stati raggruppati nella classe *Altro*.

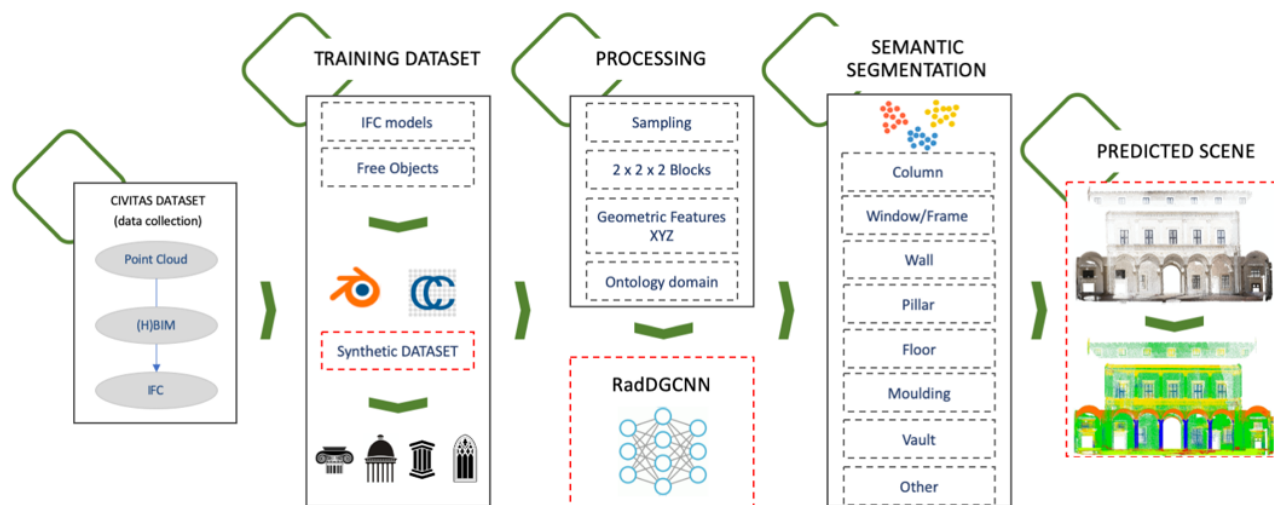


Figura 1. Workflow della segmentazione semantica proposta

3.2 Dataset di addestramento sintetico

In questa sezione viene descritto il dataset utilizzato nei nostri esperimenti per addestrare il modello neurale profondo. Il dataset è costituito da nuvole di punti derivate da modelli 3D esistenti ed è composto da due parti.

Il primo include nuvole di punti estratte da modelli 3D di edifici storici reali. Nella sezione 3.2.1 sono forniti dettagli sul processo di modellazione *Scan-to-BIM* svolto dagli esperti come parte della loro attività di ricerca.

La seconda parte include nuvole di punti derivate da una raccolta di oggetti 3D di singoli elementi architettonici, raccolti da repository online aperti, descritti in dettaglio nella sezione 3.2.2.

In entrambi i casi, è stato utilizzato il software open source CloudCompare⁴ per convertire i modelli 3D, in formato OBJ, in nuvole di punti, campionando casualmente punti da superfici con una densità di 1000 punti per m². Lo stesso sottocampionamento è stato applicato alle nuvole di punti TLS utilizzate per testare le reti neurali.

3.2.1 Modelli 3D dai tradizionali flussi di lavoro Scan-to-BIM: la sfida principale nel campo HBIM è incorporare l'enorme conoscenza dell'architettura storica disponibile da vari domini in modelli affidabili e accurati, sfruttando le piattaforme esistenti e le loro potenzialità in intelligenza semantica, informazioni parametriche e alimentazione di glossari. L'approccio proposto in questo articolo si basa sulla precedente strutturazione semantica e standardizzazione di un insieme significativo di edifici, grazie al solido know-how del gruppo di ricerca sulla digitalizzazione dell'architettura storica e sulla sua rappresentazione (Quattrini et al., 2017).

Un efficace approccio BIM per la documentazione del patrimonio architettonico (Apollonio et al., 2012) è stato applicato negli ultimi anni per lo sviluppo di diversi casi studio (Quattrini et al., 2016), dimostrando chiaramente una

sostanziale coerenza tra il modo classico di costruire e la possibilità di rappresentarlo in modelli semantici (Bianconi et al., 2018; Quattrini et al., 2017). Un interessante approccio metodologico mostra una descrizione semantica degli elementi architettonici basata su riflessioni teoriche ed esperienze di ricerca. Un concetto comune e convalidato è quello di estrarre la conoscenza dell'architettura classica da trattati così come da indagini accurate e, quindi, sviluppare modelli parametrici a diversi livelli di oggetti (Bianchini et al., 2017). Le varianti derivano dai modelli parametrici modificando o raggruppando i parametri in base alle grammatiche o alla consapevolezza degli operatori.

Il primo approccio all'HBIM nel Palazzo Ducale è stato una fase di modellazione di base e già valutata, basata su strumenti di editing di Revit come le famiglie caricabili o locali. Tuttavia, la parametrizzazione e l'adozione di robuste tassonomie sono intese come caratteristiche fondamentali per modelli informati. Al fine di effettuare un'analisi approfondita e sviluppare tutte le parti in modo rigoroso, è stato modellato un primo nucleo del Cortile d'Onore (Figura 2). La scansione ritmica e la modularità dello spazio progettato da Luciano Laurana racchiudono tutti gli elementi in grado di sintetizzare l'intero Palazzo. Gli schemi degli elementi architettonici sono stati realizzati partendo dall'integrazione dei dati di rilievo con la letteratura classica e con riferimenti più recenti, ottenendo anche un riferimento per affrontare il tema dei diversi LOD (level of detail, livello di sviluppo e definizione).

Per quanto riguarda le strutture verticali, l'elemento principale è un ordine corinzio con suddivisione che attualmente non è stato considerato nel raggruppamento successivo. Mentre tutte le aperture erano distribuite semanticamente in "finestra" e "cornici". Per quanto riguarda le famiglie caricabili "colonna", "finestra" e "pilastrini", gli oggetti sono stati modellati impostando vincoli geometrici in grado di sviluppare una parametrizzazione. Gli angoli del Cortile necessitavano di famiglie locali "ad hoc", secondo il disegno innovativo di una "soluzione d'angolo" ben nota in storiografia e letteratura.

⁴<https://www.danielgm.net/cc/>

Un altro passo importante nella modellazione, inoculando funzionalità ricche nel presente framework, è la definizione di LOD, che consente diversi livelli di rappresentazione della realtà, nonché di estrarre nuvole di punti modellate secondo dettagli corretti. Il modello 3D ottenuto viene informato, annotato e dettagliato.

Nel caso di Palazzo Ferretti (Figura 3), la procedura aveva previsto di costruire un modello parametrico dell'edificio in un ambiente Revit integrando più dati: rilievo TLS, rilievi bidimensionali e rilievo fotogrammetrico per informazioni di dettaglio, nonché un confronto continuo con le regole dei trattati. Il lavoro principale riguarda le modanature delle facciate principali con la realizzazione di famiglie caricabili (*.rfa) incorporanti semantiche e vincoli geometrici in coerenza con la pratica dell'architettura classica (Moreira et al., 2018). I risultati più significativi sinora per Palazzo Ferretti riguardano la realizzazione di librerie parametriche per le modanature esterne ed interne degli edifici, ora quelle modanature possono alimentare il nostro dataset sintetico.



Figura 2. Palazzo Ducale di Urbino, il Cortile d'Onore progettato da Luciano Laurana



Figura 3. Palazzo Ferretti, facciata principale

Considerando la necessità di ampliare il dataset semanticamente strutturato, sono stati sfruttati i modelli provenienti dalla Palladio Library (ricerca di rilievo e documentazione digitale delle ville e dei ponti di Andrea Palladio nell'ambito del progetto finanziato da Arcus/CISA Andrea Palladio, per la realizzazione del progetto "Biblioteca Palladio - mediateca e sistema di comunicazione e informazione su Andrea Palladio, le sue opere e il suo territorio", responsabile scientifico dell'Unità UNIVPM: Paolo Clini, responsabile scientifico dell'intero

progetto: Marco Gaiani). Dall'idea di Biblioteca Palladiana, è stato principalmente prodotto un geo-database (Apollonio et al., 2013; Gaiani et al., 2015) che costituisce una sorta di HBIM ante litteram.

Il gruppo di modelli sviluppati per la Biblioteca Palladiana comprendeva 13 modelli, in particolare due ponti e varie ville e/o barchesse. Per il presente lavoro sono state prese in considerazione solo le ville perché presentavano classi coerenti con quelle delle nuvole di punti che dovevano essere segmentate. Una caratteristica principale di questi modelli, alla luce del presente lavoro, è stata la *shape-grammar* adottata nella costruzione dei modelli 3D utilizzando un insieme prestabilito di regole formali a forma di albero che indicano uno scopo chiaro e una struttura evidente. Questa organizzazione è stata estesa, nei nostri casi, su più livelli gerarchici, consentendo ora di scoprire la corrispondenza con il clustering precedente. I modelli della Palladio Library sono stati infatti archiviati in formato file 3DM incorporando una tassonomia grazie alla denominazione dei layer, consentendo l'esportazione in formato OBJ e la successiva creazione di nuvole di punti sintetiche strutturate semanticamente.

Da tutti i modelli sopra menzionati, progettati da esperti di dominio, è stato possibile ottenere il nucleo del dataset sintetico sufficientemente robusto da istanziare il flusso di lavoro successivo.

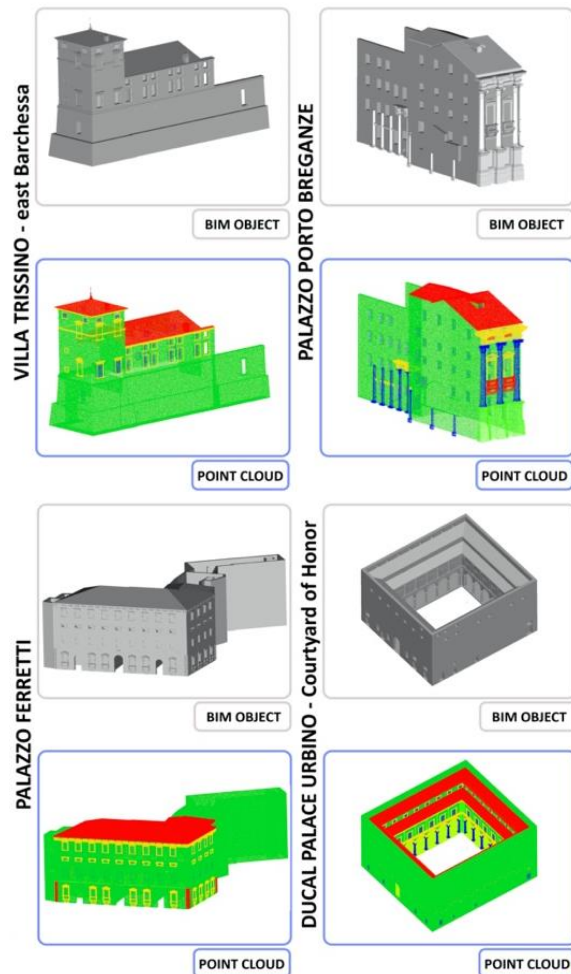


Figura 4. Esempi di modelli 3D di edifici storici e nuvole di punti annotate derivate

3.2.2 Oggetti 3D dagli archivi online: mentre alcune classi, e.g. parete e pavimento, sono ampiamente presenti nei modelli 3D sopra descritti, lo stesso non vale per altre classi di elementi architettonici. Ad esempio, non tutti gli edifici hanno colonne o volte e, quando questi elementi sono presenti, sono solitamente molto simili all'interno dello stesso edificio. Al contrario, per apprendere caratteristiche generalizzabili, dobbiamo migliorare la variabilità della forma all'interno della stessa classe.

Nel tentativo di affrontare tali problemi, abbiamo esteso il nostro dataset sintetico con modelli 3D di singoli elementi architettonici, selezionando oggetti 3D disponibili pubblicamente da una serie di repository online. (<http://www.cgtrader.com>; <http://www.3dcadbrowser.com>; <https://www.turbosquid.com>; <https://free3d.com>; <https://archive3d.net>; <https://3dwarehouse.sketchup.com>).

Nella Tabella 1 riportiamo, per ogni classe, il numero di modelli 3D raccolti, che possono includere uno o più elementi architettonici, e il numero totale di punti ottenuti trasformando tali modelli in nuvole di punti.

Classe	# Modelli 3d	# Punti
Colonne	53	448.297
Finestre/porte	41	1.041.552
Modanature	31	442.809
Pilastrini	24	558.380
Volte	10	3.743.896

Tabella 1. Panoramica degli oggetti 3D raccolti dai repository online

Come mostrato in Figura 5 i tipi di oggetti sono stati scelti come un affinamento delle classi che mancano, dando priorità a modelli di complessità e qualità paragonabili a quelli modellati dagli autori.

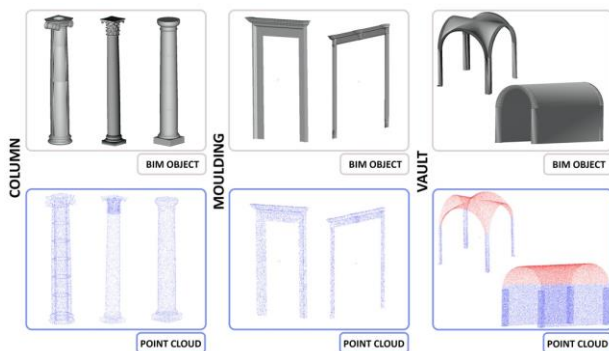


Figura 5. Esempi di modelli 3D di singoli elementi architettonici di classi diverse e corrispondenti a nuvole di punti sintetiche

3.3 Rilevamento delle nuvole di punti

Come spiegato in (Clini et al., 2018, Nespeca, 2019) la digitalizzazione tridimensionale del Palazzo Ducale di Urbino è un primo risultato consentito dal Progetto Strategico CIVITAS. In questo contesto è stata condotta una campagna di indagine completa, basata principalmente su TLS e acquisizione di dati fotografici. Allo stato attuale, l'intero modello numerico del Palazzo è costituito da 1.790 mln di punti, mentre l'esperimento si occupa di una parte della nuvola di punti relativa al Cortile d'Onore (Tabella 2).

La prima nuvola di punti concepita come cornice è un modello completo per questo edificio di grandi dimensioni: un principale dataset di riferimento è stato fornito dall'acquisizione del Leica Backpack. Per il lavoro in corso, invece, si preferisce utilizzare

la nuvola di punti realizzata grazie al TLS (Leica ScanStation C10 e Leica ScanStation P40) montato su treppiede. Questa acquisizione è stata eseguita impostando diversi livelli di risoluzione in base alla complessità degli ambienti e dei loro elementi decorativi, ottimizzando i tempi di scansione. Altre acquisizioni sono attualmente in corso con KAARTA Stencil V2 e UAV per completare il modello numerico.

Palazzo Ducale di Urbino		
	Nuvola intera	Nuvola annotata
Numero di punti	1.224 mln	17,3 mln
Errore max di align.	0,006 m	0,008 m
Errore min di align.	0,001 m	0,003 m
RMS	0,005 m	0,004 m
Palazzo Ferretti		
	Nuvola intera	Nuvola annotata
Numero di punti	1.273 mln	8,4 mln
Errore max di align	0,008	0,005 m
Errore min di align	0,003	0,001m
RMS	0,007	0,007 m

Tabella 2. Numero di punti, errore di allineamento max / min e RMS delle due nuvole di punti TLS utilizzate nei nostri esperimenti

Le operazioni di rilievo TLS del complesso di Palazzo Ferretti risalgono al 2014: sono state effettuate 69 stazioni: 49 stazioni (58 scansioni) per il rilievo esterno dell'edificio e 20 stazioni e scansioni per il rilievo interno, del piano terra e del piano primo. Queste scansioni sono state acquisite con una risoluzione di 1 cm a 100 m, alcune parti in cui erano necessari maggiori dettagli sono state acquisite con una nuvola di punti più densa (0,5 cm a 100 m). La nuvola di punti finale ottenuta dall'allineamento delle nuvole di tutte le 78 scansioni è composta da 1,2 miliardi di punti (vedi Tabella 2). Per questa ricerca una parte della nuvola di punti, riferita all'età rinascimentale, è stata sfruttata per coprire le classi già disponibili nel Palazzo Ducale di Urbino. Solo una parte dell'intera nuvola di punti è stata quindi annotata (Tabella 2).

3.4 Modello di rete neurale profonda

Dynamic Graph Convolutional Neural Networks (DGCNN), recentemente proposta in (Wang et al., 2019b), ha dimostrato di fornire prestazioni di segmentazione semantica su benchmark standard dello stato dell'arte, come il dataset S3DIS (Armeni et al., 2016).

In questo documento abbiamo considerato DGCNN e una sua nuova versione modificata (RadDGCNN) introdotta in questo lavoro; successivamente abbiamo combinato i due approcci per aumentare ulteriormente le prestazioni.

3.4.1 Architettura: il modello DGCNN si basa sull'operazione EdgeConv, introdotta in (Wang et al., 2019b), che apprende le caratteristiche nascoste di un punto in base ai punti vicini. Nonostante il suo nome, EdgeConv è implementato con un Multilayer Perceptron (MLP) alimentato con le cosiddette *edge features* e da una successiva operazione di max-pooling sulle caratteristiche apprese. Nel nostro caso, le caratteristiche del bordo di un punto sono composte dal vettore tridimensionale che rappresenta le coordinate del punto e i vettori di distanza tra il punto e i punti vicini.

L'operazione EdgeConv viene inizialmente eseguita sulla nuvola di punti di input e quindi, ripetutamente, sull'output del livello EdgeConv precedente, apprendendo così le caratteristiche nascoste locali gerarchiche per ogni punto.

Le funzionalità nascoste di output di tutti i punti appresi dai livelli EdgeConv vengono concatenate e inviate a una rete MLP per apprendere le funzionalità globali della nuvola di punti.

Infine, le caratteristiche globali e le caratteristiche locali vengono concatenate e inviate ad un altro MLP che fornisce la categoria prevista per ogni punto.

L'architettura di rete utilizzata nei nostri esperimenti è rappresentata nella Figura 6. In questa figura, N rappresenta il numero di punti per ogni nuvola di punti di input e P è il

numero di classi considerate per la classificazione basata su punti e K è il numero di punti vicini considerati per ogni punto. Tre livelli EdgeConv vengono utilizzati per apprendere le caratteristiche del punto locale. Le funzionalità apprese da ogni livello EdgeConv per ogni punto vengono quindi concatenate e le funzionalità globali vengono apprese con un MLP a tre livelli.

Abbiamo adottato le stesse impostazioni, dimensioni di MLP e numero di livelli, utilizzati in (Wang et al., 2019b).

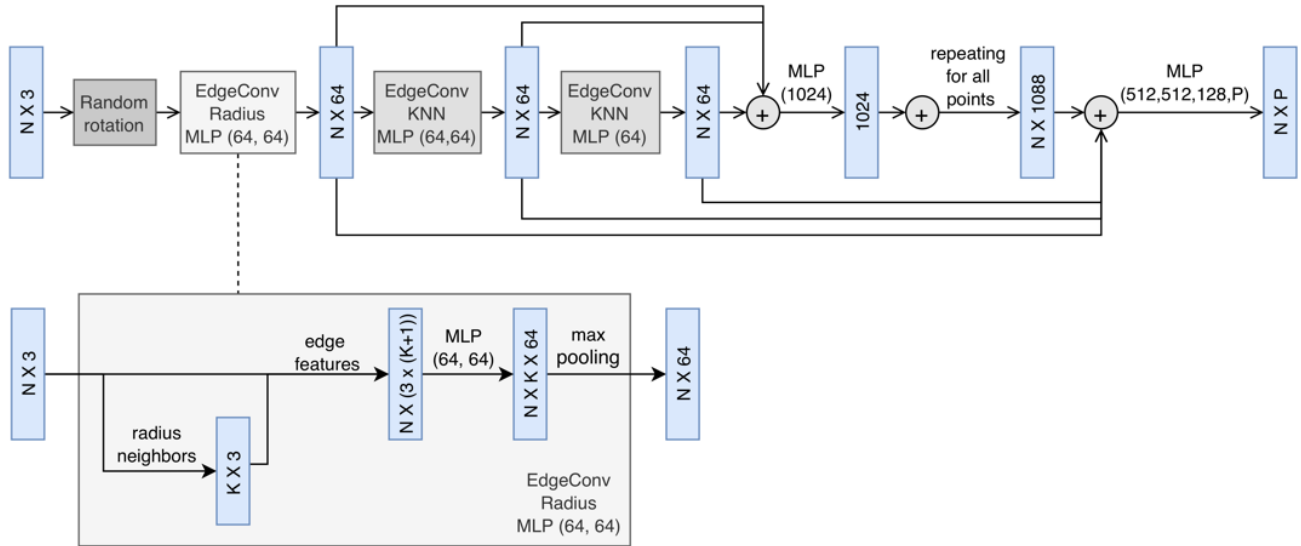


Figura 6. Architettura della rete neurale profonda adottata in fase sperimentale

3.4.2 Vicinanza dei punti: Una decisione importante da prendere nell'attuazione di una DGCNN è come definire la vicinanza di un punto. In (Wang et al., 2019b), i K-Nearest Neighbors (KNN) di un punto sono considerati per calcolare le caratteristiche del bordo ad ogni successiva operazione EdgeConv. Tuttavia, come suggerito da recenti studi (Hermosilla et al., 2018), KNN potrebbe non essere il metodo migliore quando si tratta di nuvole di punti non uniformemente dense. Questo è proprio il nostro caso, poiché l'acquisizione laser produce generalmente nuvole di punti non uniformi. Nel caso dei KNN, l'estensione spaziale dei vicini considerati per i due punti può essere molto diversa, a seconda della densità locale della nuvola di punti. Questo a sua volta, potrebbe portare a caratteristiche di bordo diverse per due punti che appartengono alla stessa classe (finestra). D'altra parte, usando la distanza del raggio, i punti vicini coprono la stessa area.

Spinti da questa considerazione, abbiamo sperimentato una variazione dell'architettura DGCNN (RadDGCNN in questo articolo), dove la distanza del raggio viene utilizzata nella prima operazione EdgeConv al posto del KNN (Figura 4)).

In pratica, per ogni punto, si selezionano prima tutti i punti entro la distanza raggio R dal punto stesso, quindi si campionano casualmente K di questi punti e li forniamo come input all'MLP per estrarre le caratteristiche del bordo. Tale operazione di campionamento ci consente di fornire sempre lo stesso numero di punti vicini (e quindi lo stesso numero di caratteristiche di fronte) come input per il MLP.

4. SETUP SPERIMENTALE

Abbiamo addestrato i modelli DGCNN e RadDGCNN sul nostro dataset sintetico suddividendo ciascuna nuvola di punti in blocchi di dimensioni uguali (2 metri x 2 metri x 2 metri)

ciascuno contenente 4096 punti campionati casualmente. Abbiamo fatto lo stesso per il set di prova.

La segmentazione della nuvola di punti, utilizzata anche in lavori precedenti sulla segmentazione della scena tramite reti neurali, è una fase di preelaborazione obbligatoria poiché fornire le intere nuvole di punti come input sarebbe computazionalmente irrealizzabile. Il campionamento casuale dei punti per ogni blocco ci permette di ottenere input di uguali dimensioni, che è un requisito dell'architettura DGCNN.

Nei nostri esperimenti, ogni punto è rappresentato da un vettore di 6 dimensioni, dove i primi 3 elementi sono le coordinate XYZ assolute del punto nella nuvola di punti originale, mentre gli altri 3 valori rappresentano le coordinate normalizzate all'interno del blocco (nell'intervallo [0-1]).

Per consentire un confronto equo, abbiamo adottato gli stessi iperparametri di rete di (Wang et al, 2019b), con l'unica differenza nella dimensione dei mini-batch, che abbiamo impostato a 16. Più specificamente, il tasso di apprendimento iniziale è stato impostato a 0,001 ed è stato utilizzato un decadimento della velocità di apprendimento di 0,5; lo slancio per la normalizzazione del batch è stato impostato su 0,9; e abbiamo utilizzato l'algoritmo di ottimizzazione di Adam per la retro-propagazione. Infine, è stata applicata una regolarizzazione di abbandono con probabilità di mantenimento di 0,5 negli ultimi due strati completamente connessi.

Per quanto riguarda l'approccio RadDGCNN, la distanza del raggio è stata calcolata sulle coordinate normalizzate, R è stato impostato sperimentalmente a 0,1. In entrambi i modelli il valore di K è stato impostato a 20, seguendo le indicazioni fornite in (Wang et al., 2019b).

A differenza di altri studi, non abbiamo utilizzato le caratteristiche del colore (es. RGB o HSV) poiché il dataset

