UAB
Universitat Autònoma de Barcelona

Towards Handling 3D Shape, Terrain Elevation, and Visual Relocalization with Implicit Neural Representation

Shun Yao

Universitat Autònoma
de Barcelona

Towards Handling 3D Shape, Terrain Elevation, and Visual Relocalization with Implicit Neural Representation

A dissertation submitted by **Shun Yao** to the Universitat Autònoma de Barcelona in fulfilment of the degree of **Doctor of Philosophy** in the Departament de Ciències de la Computació.

Bellaterra, September 02, 2024

Director	**Dr. Mikhail G. Mozerov** Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr. Yongmei Cheng** School of Automation Northwestern Polytechnical University
Thesis committee	**Dr. Zhunga Liu** School of Automation Northwestern Polytechnical University **Dr. Angel D. Sappa** Centre de Visió per Computador Universitat Autònoma de Barcelona **Dr. Vitaly Kober** Department of Computer Center for Scientific Research and Higher Education (CICESE) **Dr. Yongqiang Zhao** School of Automation Northwestern Polytechnical University **Dr. Xialei Liu** College of Computer Science Nankai University

This document was typeset by the author using LaTeX 2_ε.

ISBN: xxx-xx-xxxxxx-x-x

Printed by Ediciones Gráficas Rey, S.L.

Acknowledgements

I must first extend my deepest gratitude to my supervisor Dr. Mikhail G. Mozerov, for his patient guidance and motivating, effective, and professional advice throughout my research lifetime. Without his invaluable help, I couldn't insist continuously on my research interest. He taught me so much about how to clearly express my ideas in writing, how to utilize useful and efficient research tools, how to solve problems with a positive attitude, and so on.

Next, I need to appreciate my family's support. My parents have been devoting themselves to providing me with a carefree life and encouraging me to pursue my academic career. Also thanks to other family members, who cast me quite a lot of concerns, especially in the hardest COVID-19 quarantine time.

I would like to give my special thanks to Dr. Fei Yang, who served as an amazing collaborator and led me to improve my study skills. His logical thinking, rigorous expression, and careful consideration have always inspired me, enabling me to better conduct research at every step.

I still remember my first day arriving in Barcelona, I lost on my way to my apartment because of public transport. Thanks for the help from Xialei and Lu, they came a long way to pick me up and provided me with more help in the following time. Also thanks to my roommates Junda and Zhen, they help me a lot to adapt to my daily life in Barcelona.

I am thankful to my friends from CVC, their company, help, and support mean so much to me: Lei, Yaxing, Yi, Kai, Chenshen, Yixiong, Shiqi, Danna, Hector, Sanket, Albin, and so on. They enriched my experience beyond study life so far.

Finally, thanks to CSC support. They provided me with a scholarship and the opportunity to finish my doctoral studies at CVC, Barcelona.

Abstract

Our real world is located in the physical space field, however, humans need to quantify physical properties in computer vision applications. For example, we represent the visual information as RGB intensity, terrain as elevation values, stereo shapes as surfaces, entities as occupied volumes, *etc.* With advances in machine learning technology, implicit neural representation (INR) models, which parameterize these physical properties using coordinate-based mapping functions, offer promising solutions that are more accurate, higher fidelity, more expressive, faster to implement, and more memory-efficient. This dissertation focuses on developing INR models for 3D shape representation, terrain elevation representation, multi-scale DEM super-resolution, and multi-scene visual relocalization.

In the case of 3D shape representation, we explore the use of hierarchical and topological structures to learn latent representations of 3D geometric data. Noting the limitations of existing graph convolution networks in resolution and structural complexity, we introduce INRs to improve representation granularity and flexibility. Instead of using explicit formats like points, lines, and surfaces, the INR aims to regress the signed distance from any arbitrary 3D point to the shape's surface. The 3D shape is then represented as an iso-surface extracted from the predicted signed distances. However, directly using a single neural network to approximate the entire 3D shape would result in long training times and require numerous network parameters. To improve learning efficiency and reduce training parameters, we propose an INR model that uses multiple latent codes to learn local geometries rather than the entire 3D shape. Additionally, we introduce an auxiliary graph convolution network to transmit these latent codes to specific shape parts and propose a novel geometric loss function to facilitate mutual learning among the latent codes.

For terrain elevation representation, we study the representation precision problem caused by the discretization of existing digital elevation models. Furthermore, different applications require specific discrete representations, necessitating format conversions. However, these conversions inevitably compromise the fidelity of elevation data. To solve the above problems that lead

to inaccurate representation of elevation data, we develop a new continuous representation model (CDEM), an INR model that allows height values to be obtained at any arbitrary query position, aiming to preserve the continuity of topographic elevation data in the real world.

Next, we train an encoder-decoder network to learn CDEM from discrete elevation data for multi-scale DEM super-resolution tasks. To improve model accuracy, we propose predicting the bias of elevation values between the query position and its closest known position. To facilitate the model's ability to predict high-frequency variations, we introduce positional encoding to map query positions into a higher-dimensional space. Our experiments demonstrate that our model can achieve more accurate elevation values and preserve more detailed terrain structures than other methods.

For visual relocalization across multiple scenes, we focus on efficient learning without using prepared scene geometry information and time-consuming pre-built scenario representations. Recently, scene coordinate (SC) regression-based models have demonstrated that accurate visual relocalization can be efficiently achieved using only posed images and camera intrinsic parameters. However, extending SC regression models to multiple scenes typically requires retraining model parameters or using pre-built reference landmarks, both of which are time-consuming. To enhance efficiency and avoid this process, we propose representing multiple scenes within a global reference coordinate and training an SC regression model (*i.e.*, an INR model) using posed images from all scenes simultaneously. To reduce the impact of visual ambiguities, we introduce scene embedding as a prior condition for our model predictions. To enhance our model's generalizability across multiple scenes, we propose the scene-conditional regression-adjust (SCRA) module, which dynamically generates parameters to adapt flexibly to the scene embedding. Additionally, we introduce modulation and complement modules to enhance the model's applicability at both the image sample and scene levels.

Key words: *implicit neural representation, 3D shape representation, continuous DEM model, DEM super-resolution, visual relocalization, scene coordinate prediction, conditional adaption, deep learning*

Resumen

Nuestro mundo real se encuentra en el campo del espacio físico; sin embargo, los seres humanos necesitan cuantificar propiedades físicas en aplicaciones de visión por computadora. Por ejemplo, representamos la información visual como intensidad RGB, el terreno como valores de elevación, las formas estéreo como superficies, y las entidades como volúmenes ocupados, etc. Con los avances en la tecnología de aprendizaje automático, los modelos de representación implícita neuronal (INR), que parametrizan estas propiedades físicas utilizando funciones de mapeo basadas en coordenadas, ofrecen soluciones prometedoras que son más precisas, de mayor fidelidad, más expresivas, más rápidas de implementar y más eficientes en cuanto a memoria. Esta disertación se centra en el desarrollo de modelos INR para la representación de formas 3D, la representación de elevación del terreno, la superresolución multiescala de DEM, y la relocalización visual en múltiples escenas.

En el caso de la representación de formas 3D, exploramos el uso de estructuras jerárquicas y topológicas para aprender representaciones latentes de datos geométricos en 3D. Notando las limitaciones de las redes de convolución de grafos existentes en cuanto a resolución y complejidad estructural, introducimos INRs para mejorar la granularidad y flexibilidad de la representación. En lugar de usar formatos explícitos como puntos, líneas y superficies, el INR tiene como objetivo calcular la distancia firmada desde cualquier punto 3D arbitrario hasta la superficie de la forma. La forma 3D se representa entonces como una isosuperficie extraída de las distancias firmadas predichas. Sin embargo, usar directamente una única red neuronal para aproximar toda la forma 3D resultaría en tiempos de entrenamiento prolongados y requeriría numerosos parámetros de red. Para mejorar la eficiencia del aprendizaje y reducir los parámetros de entrenamiento, proponemos un modelo INR que utiliza múltiples códigos latentes para aprender geometrías locales en lugar de toda la forma 3D. Además, introducimos una red de convolución de grafos auxiliar para transmitir estos códigos latentes a partes específicas de la forma y proponemos una nueva función de pérdida geométrica para facilitar el aprendizaje mutuo entre los

códigos latentes.

Para la representación de la elevación del terreno, estudiamos el problema de precisión de la representación causado por la discretización de los modelos digitales de elevación (DEM) existentes. Además, diferentes aplicaciones requieren representaciones discretas específicas, lo que hace necesario realizar conversiones de formato. Sin embargo, estas conversiones inevitablemente comprometen la fidelidad de los datos de elevación. Para resolver los problemas mencionados que conducen a una representación inexacta de los datos de elevación, desarrollamos un nuevo modelo de representación continua (CDEM), un modelo INR que permite obtener valores de altura en cualquier posición de consulta arbitraria, con el objetivo de preservar la continuidad de los datos de elevación topográfica en el mundo real.

A continuación, entrenamos una red codificador-decodificador para aprender CDEM a partir de datos de elevación discretos para tareas de superresolución DEM multiescala. Para mejorar la precisión del modelo, proponemos predecir el sesgo de los valores de elevación entre la posición de consulta y su posición conocida más cercana. Para facilitar la capacidad del modelo para predecir variaciones de alta frecuencia, introducimos la codificación posicional para mapear las posiciones de consulta en un espacio de mayor dimensión. Nuestros experimentos demuestran que nuestro modelo puede lograr valores de elevación más precisos y preservar estructuras del terreno más detalladas que otros métodos.

Para la relocalización visual en múltiples escenas, nos centramos en el aprendizaje eficiente sin utilizar información de geometría de escena preparada ni representaciones de escenarios preconstruidas que consumen mucho tiempo. Recientemente, los modelos basados en la regresión de coordenadas de escena (SC) han demostrado que la relocalización visual precisa se puede lograr de manera eficiente utilizando solo imágenes posadas y parámetros intrínsecos de la cámara. Sin embargo, extender los modelos de regresión SC a múltiples escenas generalmente requiere volver a entrenar los parámetros del modelo o utilizar puntos de referencia preconstruidos, lo cual es un proceso que consume mucho tiempo. Para mejorar la eficiencia y evitar este proceso, proponemos representar múltiples escenas dentro de un sistema de coordenadas de referencia global y entrenar un modelo de regresión SC (es decir, un modelo INR) utilizando imágenes posadas de todas las escenas simultáneamente. Para reducir el impacto de las ambigüedades visuales, introducimos la incrustación de escena como una condición previa para nuestras predicciones del modelo. Para mejorar la

generalización de nuestro modelo en múltiples escenas, proponemos el módulo de ajuste de regresión condicional a la escena (SCRA), que genera dinámicamente parámetros que se adaptan de manera flexible a la incrustación de la escena. Además, introducimos módulos de modulación y complemento para mejorar la aplicabilidad del modelo tanto a nivel de muestra de imagen como a nivel de escena.

Palabras clave: *representación neuronal implícita, representación de formas 3D, modelo DEM continuo, superresolución de DEM, relocalización visual, predicción de coordenadas de escena, adaptación condicional, aprendizaje profundo*

Resum

El nostre món real es troba en l'espai físic; tanmateix, els humans necessiten quantificar propietats físiques en aplicacions de visió per computador. Per exemple, representem la informació visual com a intensitat RGB, el terreny com a valors d'elevació, les formes estereoscòpiques com a superfícies i les entitats com a volums ocupats, etc. Amb els avenços en la tecnologia d'aprenentatge automàtic, els models de representació neuronal implícita (INR), que parametrizen aquestes propietats físiques mitjançant funcions de mapeig basades en coordenades, ofereixen solucions prometedores que són més precises, de major fidelitat, més expressives, més ràpides d'implementar i més eficients en l'ús de memòria. Aquesta dissertació se centra en el desenvolupament de models INR per a la representació de formes 3D, la representació de l'elevació del terreny, la superresolució multiescala de DEM i la relocalització visual en múltiples escenes.

En el cas de la representació de formes 3D, explorem l'ús d'estructures jeràrquiques i topològiques per a aprendre representacions latents de dades geomètriques en 3D. Observant les limitacions de les xarxes de convolució de grafs existents pel que fa a la resolució i la complexitat estructural, introduïm INR per a millorar la granularitat i la flexibilitat de la representació. En lloc d'utilitzar formats explícits com punts, línies i superfícies, l'objectiu de l'INR és calcular la distància signada des de qualsevol punt 3D arbitrari fins a la superfície de la forma. La forma 3D es representa com una isosuperfície extreta de les distàncies signades predites. No obstant això, utilitzar directament una única xarxa neuronal per a aproximar tota la forma 3D resultaria en temps d'entrenament prolongats i requeriria nombrosos paràmetres de xarxa. Per a millorar l'eficiència de l'aprenentatge i reduir els paràmetres d'entrenament, proposem un model INR que utilitza múltiples codis latents per a aprendre geometries locals en lloc de tota la forma 3D. A més, introduïm una xarxa de convolució de grafs auxiliar per a transmetre aquests codis latents a parts específiques de la forma i proposem una nova funció de pèrdua geomètrica per a facilitar l'aprenentatge mutu entre els codis latents.

Per a la representació de l'elevació del terreny, estudiem el problema de

la precisió de la representació causat per la discretització dels models digitals d'elevació (DEM) existents. A més, diferents aplicacions requereixen representacions discretes específiques, la qual cosa fa necessàries conversions de format. No obstant això, aquestes conversions inevitablement comprometen la fidelitat de les dades d'elevació. Per a solucionar aquests problemes que condueixen a una representació inexacta de les dades d'elevació, desenvolupem un nou model de representació contínua (CDEM), un model INR que permet obtenir valors d'altura en qualsevol posició de consulta arbitrària, amb l'objectiu de preservar la continuïtat de les dades d'elevació topogràfica en el món real.

A continuació, entrenem una xarxa codificador-decodificador per a aprendre CDEM a partir de dades d'elevació discretes per a tasques de superresolució DEM multiescala. Per a millorar la precisió del model, proposem predir el biaix dels valors d'elevació entre la posició de consulta i la seva posició coneguda més propera. Per a facilitar la capacitat del model per a predir variacions d'alta freqüència, introduïm la codificació posicional per a mapar les posicions de consulta en un espai de major dimensió. Els nostres experiments demostren que el nostre model pot aconseguir valors d'elevació més precisos i preservar estructures del terreny més detallades que altres mètodes.

Per a la relocalització visual en múltiples escenes, ens centrem en l'aprenentatge eficient sense utilitzar informació de geometria d'escena preparada ni representacions de escenaris preconstruïdes que consumeixen molt de temps. Recentment, els models basats en la regressió de coordenades d'escena (SC) han demostrat que la relocalització visual precisa es pot aconseguir de manera eficient utilitzant només imatges posades i paràmetres intrínsecs de la càmera. No obstant això, l'extensió dels models de regressió SC a múltiples escenes generalment requereix tornar a entrenar els paràmetres del model o utilitzar punts de referència preconstruïts, la qual cosa és un procés que consumeix molt de temps. Per a millorar l'eficiència i evitar aquest procés, proposem representar múltiples escenes dins d'un sistema de coordenades de referència global i entrenar un model de regressió SC (és a dir, un model INR) utilitzant imatges posades de totes les escenes simultàniament. Per a reduir l'impacte de les ambigüitats visuals, introduïm la incrustació d'escena com a condició prèvia per a les nostres prediccions del model. Per a millorar la generalització del nostre model en múltiples escenes, proposem el mòdul d'ajust de regressió condicional a l'escena (SCRA), que genera dinàmicament paràmetres que s'adapten de manera flexible a la incrustació de l'escena. A més, introduïm mòduls de modulació i complement per a millorar l'aplicabilitat del model tant a nivell de mostra d'imatge com a nivell d'escena.

Paraules clau: *representació neuronal implícita, representació de formes 3D, model DEM continu, superresolució de DEM, predicció de coordenades d'escena, relocalització visual, adaptació condicional, aprenentatge profund*

Contents

List of Figures

List of Tables

1 Introduction

In the era of artificial intelligence (AI), there is a growing demand for data representations that are precise, high-fidelity, compact, convertible, memory-efficient, and easily accessible across various 3D computer vision tasks. These tasks include 3D shape representation, remote sensing, autonomous driving, augmented reality, visual navigation, *etc.* Concurrently, deep learning methods, benefiting from the development of computational hardware and digitization processes, obtain more and more breakthroughs in almost all these computer vision fields. Training these neural models requires an amount of precise 3D data. However, classical 3D representation models, including point clouds, meshes, voxels, *etc.*, use discrete formats to maintain data, leading to some inevitable problems in practice:

(a) Discrete formats demand substantial storage, memory, and computational resources to accurately represent 3D shapes at high resolution or to handle large-scale 3D scenarios. Point cloud models use 3D points to flexibly represent complex shapes and surface details, mesh models use triangles or quadrilaterals to compose the object surface, and voxel models use regular cubes to assemble stereo objects. Achieving high precision requires increasing the density of 3D points, the number of triangles (or quadrilaterals), and the resolution of cubes (similar to pixels). Consequently, this necessitates significant storage and computational resources to save and process these finer 3D data. This issue also extends to dealing with large-scale 3D scenarios.

(b) Different 3D representations require specialized processing methods and neural model architectures. Point cloud models present unordered 3D points without explicit connection relationships, leading to a lack of structural and topological information. In addition, point cloud data may be unevenly distributed, with significant density variations across different regions and object parts, and cannot guarantee uniform coverage of the object surface. Some regions/parts may be distorted and lose geometry features due to lacking sufficient sampling points. On the other hand, point cloud data may contain noise and outliers, which will affect processing and analysis. Prevalent deep learning methods typically use a sub-network [104, 105] to canonicalize point cloud data at the beginning. In contrast to independent points, mesh models have fixed topological structures. Yet different classes of 3D object meshes have

distinct topologies, necessitating the use of template meshes for processing 3D data within the same class. Furthermore, conventional convolutional neural networks (CNNs) are not well-suited for topological structures [132, 26]. One alternative selection is to adopt geometric learning methods, especially using graph convolutional networks (GCNs) [65, 29], with their variants for mesh processing [91, 107, 9, 43, 150]. Voxel models are akin to images but composed of regular cubes. Processing voxels typically costs a lot of computing and memory, requiring specialized architecture design such as the 3D CNN [57, 128] which could capture features along three dimensions (width, height, depth), and the Convolutional Occupancy Network [99].

(c) Converting between different 3D representations often results in a loss of precision. To satisfy applied requirements, it is frequently necessary to convert existing 3D representation models. For example, building meshes from voxel targets to increase rendering efficiency in graphics, represent shape geometry, and run accurate collision detection in physics simulation; sampling point clouds from meshes could facilitate feature extraction, reduce the amount of data, and improve processing efficiency; extracting meshes from point clouds is widely required in CAD modeling, rendering, medical image processing, and geographic information system (GIS).

These problems are common to most 3D computer vision tasks. We list major characteristics of classical 3D representation models in Table 1.1. In this dissertation, we focus on three typical scenarios requiring 3D data: 3D shape learning, terrain elevation modeling, digital elevation model (DEM) super-resolution, and visual relocalization. More depth problems of using classical 3D representations are identified according to these application backgrounds. We pay special attention to efficiently learning a unified representation model, which is memory-saving and precise, to deal with the above applications in the 3D scenario.

Table 1.1: Comparison of major traits between classical 3D models and the implicit neural representation (INR).

Characteristics	Voxels	Mesh	Point cloud	INR
Flexibility	Structure fixed	Limited by topology	Flexible	Flexible
Accessibility	Spatial grid	Topological template	Coordinate	Coordinate
Resolution	Limited by memory	Limited by template	Limtited by density	Unlimited
Memory requirement	Scale with spatial resolution			Scale with the number of network parameters
Fidelity	Limited (discrete format)			High (continuous field)

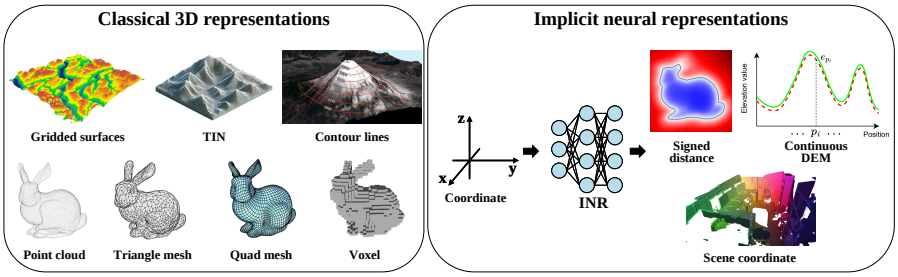

Figure 1.1: Visual examples of classical 3D models and our concerned implicit neural representations.

1.1 Implicit neural representation in 3D scenario

To overcome the deficiencies of using classical 3D representation models, one kind of learning-based representation, known as the implicit neural representation (INR) model, has gained considerable attention in recent years. Unlike explicit and discrete representation formats such as pixels, point clouds, meshes, and voxels, INRs parameterize continuous multi-media data or vector fields with neural networks. The typical pipeline of INR prediction is to input spatial coordinates into a multi-layer perceptron (MLP) neural network to obtain corresponding values. In theory, given sufficient parameters, MLPs can encode continuous signals over arbitrary dimensions at arbitrary resolutions [64]. From this view, INRs can be seen as learning a function that maps spatial coordinates to scalar or vector values. Table 1.1 provides a comparison of the main features of INR models with classical 3D representation models

This dissertation primarily explores INRs in three applications: learning 3D shape representations by regressing a signed distance function, learning terrain elevation representations by regressing a position-to-elevation mapping function, and multi-scene visual relocalization by predicting scene coordinates. Figure 1.1 illustrates examples of classical 3D representations and the INRs of our concerns.

1.1.1 INR for 3D shape learning

Using neural networks to represent 3D shapes implicitly can be traced back to classical mathematical analysis methods of geometric shapes, including level set methods [32, 96], signed distance functions (SDFs) [42], and boundary indicator functions. Almost at the same time, various researchers proposed predicting

the zero-level set [88, 4] (or zero iso-surface [98], decision boundary [27, 87]) of a 3D shape by learning a parameterized neural model to approximate these functions. The key difference between these neural representations (3D shape INRs) and classical 3D shape models is that the neural model can make predictions at arbitrary 3D positions, theoretically resulting in 3D shapes with infinite resolution. In addition, 3D shape INRs are also more memory-efficient than classical models when representing shape surfaces with finer details. Even only giving partial noisy shape observations, 3D shape INRs can recover the entire 3D shape, in contrast to point cloud models which often fail. Unlike the topological restrictions in mesh, 3D shape INRs are flexible in 3D shape transformation and generation.

In this dissertation, we focus on learning 3D shape INRs for the signed distance function. In typical SDF usage, the sign distinguishes between points inside and outside the shape, while the value at any point measures the distance to the shape boundary. Most methods directly learn a global SDF to represent entire 3D shapes, resulting in time-consuming training, numerous model parameters, and coarse reconstructed details. To alleviate this problem, several works proposed to learn a set of local SDFs to represent the 3D shape. These methods decompose space into local regions using voxels [18], grids [58], and ellipses [40, 41] to learn local SDFs. However, due to these SDFs being independently learned within their own local regions, inconsistent surface estimates inevitably occur at region boundaries. Existing methods have to use overlapped region splitting, leading to repetitive learning in the overlapped regions and reducing the efficiency.

Another challenging problem of learning the SDF is obtaining ground truth from raw data samples. Since INR learning relies on a regression-type loss, it requires data samples that clearly distinguish between inside and outside the shape. Unfortunately, raw forms of acquired 3D data, such as point clouds, open meshes, and collections of manifolds, cannot provide ground-truth representations. Current solutions including sign agnostic learning [2, 3], geometric regularization [44], analytical gradients [28], and closest-pulling operation [81], have demonstrated promising performance on raw 3D data, but their application is still limited to learning entire shapes, requiring significant time and numerous model parameters for precise representation. *Thus in this topic of learning 3D shape representation, we will explore how to efficiently learn local SDF representations that compose the entire 3D shape without ground-truth SDF value supervision, while maintaining fine geometry details.*

1.1.2 INR for terrain elevation modeling and multi-scale DEM super-resolution

Obtaining precise elevation data to describe or represent our Earth surface plays an important role in geographical analyses of hydrology [70, 8], ecology [131], geomorphology [134], *etc.*, and applications of terrain visualization [109], urban design [103], terrain-aided navigation [129], *etc.* However, currently available digital elevation data are limited to discrete representations [74, 54] such as digitized points, contour lines, triangulated irregular networks (TINs), and raster elevation values. In geography, points are typically collected by Light Detection and Ranging (LiDAR), which is a sampling tool for capturing highly precise elevation data. Contour lines represent constant elevation by joining points of equal and constant values and show the topography of the landscape. TIN models adapt to the terrain's natural shape, capture the ups and downs of the land, and have the ability to show features like ridges and valleys. By using only ground points, we could construct an elevation raster to represent the bare Earth's surface. This representation is always named the digital elevation model (DEM) and has evenly-spaced grid cells. In practice, we can use DEMs to generate slope and aspect maps, and run hill shade analysis.

Discretization, without taking into account the accuracy of geographical measurements, restricts the precision of these representations to factors such as the sparsity of points, the interval between contour lines, the number of triangles, and the cell size of the grid. In practice, these discrete representations often require transformations to meet the specific requirements of different applications. For instance, converting data to triangular surfaces enables 3D terrain visualization, while converting it to contour lines facilitates terrain topology analysis. However, such conversions inevitably compromise the fidelity of elevation data. To preserve the continuity of topographic elevation data in the real world, we propose to predict elevation values on position coordinates, which are continuous, with an INR model. In contrast to discrete approximations, our INR model can provide an unlimited number of elevation points by regressing the position-to-elevation mapping function. In the meanwhile, this ability to predict an unlimited number of elevation points is very suitable for the valuable application of DEM super-resolution.

In regions with complex topography and areas requiring soil mapping, flood modeling, or landslide hazard assessment, more accurate elevation representation can lead to better predictions and reduced uncertainty. For DEM models, finer terrain details can be retained by increasing resolution. However, collecting high-resolution elevation data is costly and time-consuming. An economical and practical way is to enhance existing DEM datasets with super-resolution

methods. This approach saves resources and accelerates the availability of updated terrain elevation data. It also allows historical data to be upgraded to current standards without extensive field surveys.

Classical DEM super-resolution methods include interpolation and information fusion techniques. However, interpolation methods tend to generate over-smooth high-frequency regions [144], while information fusion methods require complementary datasets such as high-resolution optical imagery, LiDAR data, or SAR (Synthetic Aperture Radar) imagery. In recent decades, learning-based DEM super-resolution methods have demonstrated superior performance by only using DEM data. Since the training procedure requires a fixed size of input/output pairs with a specified super-resolution scale, most existing learning-based methods are limited to increasing DEM resolution by a fixed scale. Extending these methods to applications using multi-scale high-resolution DEMs typically requires retraining model parameters, which is a time-consuming process. *In this dissertation, we aim to propose a precise, flexible, memory-efficient, and high-fidelity representation model of elevation data that allows the generation of high-resolution DEMs at multiple scales through a single learning process.*

1.1.3 INR for visual relocalization

One critical task in remote sensing is determining the position of a video camera relative to the scene depicted in a series of images captured by the camera. Classical approaches often necessitate pre-built scenario representations, typically obtained using structure-from-motion (SfM) techniques [120, 115] or consisting of posed landmark images, and the implementation of complex algorithms such as image retrieving, feature matching, and absolute/relative pose estimation. In practice, using pre-built scenario representations has several drawbacks. First, significant storage is needed to save the pre-built scenario representation. Second, using scenario representations poses a risk of exposing private information [121, 20, 149] present in the scene. Third, constructing an accurate scenario representation for precise relocalization is time-consuming [15, 97].

To avoid using pre-built representations, recent methods utilizing scene coordinate (SC) regression-based models to directly predict 3D points in the scene's world coordinates for 2D pixels in query images. A PnP solver [39] within a RANSAC [37] loop is then used to estimate the final camera pose. Figure 1.2 shows a visual relocalization pipeline by using an SC regression model.Instead of relying on an explicit scenario, SC regression models encode scene information implicitly in model parameters and have demonstrated promising performance in visual relocalization regarding accuracy and efficiency. Even without using

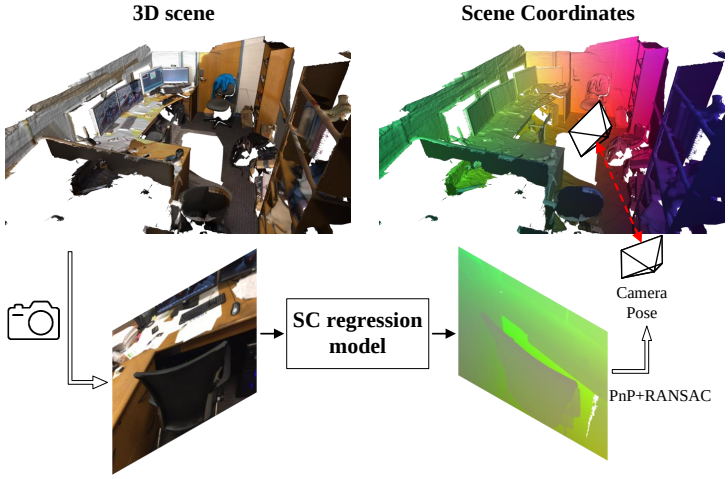

Figure 1.2: A typical pipeline of using an SC regression model for visual relocalization. We visualize 3D scene coordinates by mapping XYZ to the RGB cube.

ground truth 3D scene coordinates for supervision, some advanced SC regression models [13, 84, 85, 16] also can determine the camera position from a query image and simultaneously enable the scene construction of points. However, these SC regression models are limited to the single scene on which they were trained. Applying these models to multiple scenes requires retraining or using pre-built reference landmarks [139, 123, 108], which is time-consuming. *In this dissertation, we regard the SC regression as scene-conditional INR predictions, and aim to extend SC-based visual relocalization methods to multiple scenes with an efficiently trained INR model.*

1.2 Objectives and approach

This thesis aims to develop implicit neural representation (INR) models for various 3D computer vision tasks, including 3D shape learning, terrain elevation modeling, DEM super-resolution, and visual relocalization. We outline the objectives and approach to address the challenges identified in different application contexts.

1.2.1 INR for 3D shape learning

Learning the signed distance function (SDF) for an entire 3D shape using a single neural model requires extensive training time and numerous network parameters, often resulting in coarse reconstructions. Current methods suggest learning multiple SDF-based INRs for 3D shape parts in local space to improve representation accuracy. Since the learned parts compose the whole 3D shape, these methods require overlapped space splitting to keep shape parts consistent in boundaries. However, this leads to redundant learning in overlapping regions, diminishing efficiency. In addition, ground truth SDF values are not available from raw 3D data and its sampling process also costs significant time and computational resources. To solve these problems, we define the following objective:

> **Local geometry codes learning for implicit 3D Shape representation:** We aim to propose an SDF-based INR model that efficiently learns representations of the 3D shape parts in no-overlapped local space, without using ground truth SDF values.

To reduce the complexity of learning a 3D shape, we propose to learn a set of local SDFs to represent the shape parts. Following the DeepSDF [98] setting, we use a shared decoder architecture combined with latent codes to learn these SDFs, where each local SDF is limited to learning in a subspace. To avoid repetitive learning on split boundaries, each subspace does not overlap with the others. To facilitate the smoothing of learned shape parts over the whole space, we introduce a graph neural network (GNN) to exchange messages among these latent codes in learning these local SDFs. Additionally, we also propose a geometric similarity loss function based on the graph structure to promote local SDFs learning from neighbors. In practice, we use meshes to construct the GNN, split space, and distribute latent codes. By using sign agnostic learning [2, 3] and geometric regularization [44], our model only requires the point-to-surface distance as training data.

1.2.2 INR for terrain elevation modeling and multi-scale DEM super-resolution

The surface of the Earth is continuous, and obtaining precise elevation data at arbitrary query positions is essential for many geographical applications and analyses. Existing terrain elevation models suffer from a precision gap caused by discretization. On the other hand, collecting high-resolution elevation data is costly and time-consuming. One economical and practical way is to enhance

existing DEM datasets with super-resolution methods. Most learning-based DEM super-resolution models are limited to increasing DEM resolution by a specified scale. Applications, that need high-resolution DEM at multiple scales, typically require retraining model parameters, resulting in a time-consuming process. To break these practical barriers, we pursue the following objective:

> **A continuous digital elevation representation model for DEM super-resolution:** We aim to develop a new terrain elevation representation model that allows height values to be obtained at arbitrary query position. This model takes advantage of the INR on regressing unlimited resolution signals, achieving multi-scale DEM super-resolution tasks with a single trained neural model.

Unlike discrete approximations, we propose a continuous digital elevation model (CDEM), which uses a parametric neural network to map coordinates to elevation values. These coordinates, along with their corresponding elevation values, are associated with query positions, allowing elevation values to be obtained for any arbitrary position. By introducing a neural encoder, we extend the CDEM application to DEM super-resolution tasks. Since CDEM's predictions are based on query positions, it allows for the utilization of any number of elevation values and their corresponding positions during training. This implies that our model can be trained simultaneously for different super-resolution scales. Moreover, during testing, any query position can be employed to predict elevation values, resulting in a super-resolution DEM with the desired resolution. To amplify relatively small variations in elevation values and modify the data distribution to be more concentrated, We propose the utilization of elevation biases for CDEM prediction. Furthermore, We propose the inclusion of a positional encoding function in the CDEM to improve prediction accuracy.

1.2.3 INR for visual relocalization

Recent advanced neural models can achieve accurate visual relocalization by regressing scene coordinates for query images. Since the model parameters implicitly encode 3D scene representation via gradient descent, leading to scene-specific regression results. These models are typically limited to the scenes in which they were trained. Applying these models to multiple scenes would cost much time and computing resources to retrain model parameters or build reference landmarks in advance. To improve applying efficiency of the SC model on multiple scenes, we define the following objective:

Leaning scene coordinates in a global scenario for multi-scene visual relocalization: We target to design an SC regression-based model for multi-scene visual relocalization. The SCINR can use training data from all scenes in a fast learning schedule.

To avoid separately learning multiple sets of model parameters for applying SC regression-based visual relocalization on different scenes, we propose a scene-conditional implicit neural regression (SCINR) model that can predict multi-scene coordinates within a single global reference coordinate system. We encode scene information in scene embeddings as a prior condition for SCINR predictions, aiming to reduce the impact of visual ambiguities in multi-scene coordinate regression. We design a scene-conditional regression-adjust (SCRA) module to adapt the model to the scene embedding by dynamically generating parameters during inference. Additionally, we employ modulation and complement modules to enhance the model's prediction applicability at both the image sample and scene levels. The modulation module adjusts the amplitude, phase, and frequency of the data flow for each input image, while the complement module derives scene-specific coordinate biases to reduce distribution differences between scenes.

2 3D Shapes Local Geometry Codes Learning With SDF*

2.1 Introduction

Representing 3D data with deep neural networks is widely used in shape surface reconstruction, 3D shape generation, rendering, and compression. As one of the most popular methods, DeepSDF [98] represents the zero-level surface of the whole 3D shape by regressing its continuous signed distance function (SDF). In the model predictions, the sign is used to distinguish between points inside and outside the shape, and the value at any point measures its distance to the shape boundary. However, the effectiveness of such models depends on the complexity of 3D shapes and the capacity of neural networks. With the model capacity decreasing, the accuracy of shape representation would degenerate.

To alleviate this problem, we propose to learn a set of local SDFs to represent the whole surface. Each local SDF is responsible for a part of the reconstructed shape in this case, and learning these local SDFs is much easier. Our motivation is that the complexity of one local part of the 3D shape is much simpler than the whole and usually similar to other local parts. Since the distribution of training data is not uniform, especially when more SDFs are considered in a local region, it can not guarantee enough data are allocated for learning every local SDF. One of the possible solutions is to make these local SDFs learn from both the training data and their neighbors. Here we utilize the reasonable assumption that similarity of the neighbor parts means similar shape geometry in local space.

To prompt local SDFs to learn from each other, we introduce graph neural networks (GNNs) to make connections and propagate information among them. Following the DeepSDF [98] setting, we use a shared decoder architecture combined with latent codes to learn local SDFs. Inspired by the recent researches of 3D reconstruction [107, 9, 150, 49], which demonstrate accurate reconstructed 3D shape geometry with fine details by decoding latent codes with a hierarchical GNN structure, we design a Global-to-Local (G2L) network based on GNN layers as a "scaffold" to generate and distribute latent codes, split local regions,

*This chapter is based on a publication in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (Spotlight), 2021 [140].

and build a communication graph by a template mesh. The G2L network generates local latent codes from one global code that specified to a 3D shape, making the geometrical locality is learnable in latent space as in the original geometric space. In the meantime, the G2L inherits the message-passing mechanism from GNNs to facilitate local SDFs smoothed over the graph. This provides implicit geometrical constraints on learning latent codes, resulting in the similarity of 3D shape parts.

In addition, we also introduce a geometric similarity loss function to enhance the effectiveness of geometrical constraints. Remind our assumption is that the neighboring local regions should have similar latent codes in general. We apply this loss function on one-ring neighbors of each latent code according to their geometric relation instead of space correlation.

Though several related methods also target learning 3D shapes with local SDFs and have shown promising results, they need either the voxel representation to align latent codes or explicit parametric models (*e.g.* , a sphere) to fit. In many usage cases, using this volumetric (or parametric) space is difficult or even insolvable to represent shape parts. For example, thin plates and stick shapes. On the other hand, since the learned parts compose the whole 3D shape, these methods require overlapped space splitting to keep shape parts consistent in boundaries. This leads to repetitive learning in the overlapped regions, reducing learning efficiency. In contrast, our method relies on a more direct and simple representation of local space, *i.e.*, geometrically splitting, and leverages geometric learning techniques to represent 3D shapes with local SDFs. To avoid repetitive learning on split boundaries, we use the G2L network and the geometric similarity loss function to promote local SDFs learning from neighbors. Our experiments on 3D shape reconstruction demonstrate that our method can keep more details with significantly fewer model parameters than the original DeepSDF method.

The contributions of this chapter are as follows:

- We propose the Local Geometry Code Learning method, where we learn the shape as zero-surfaces with local latent codes.
- We use graph neural networks to generate local latent codes and distribute them on the 3D shape, which does not request voxelization of the 3D shape as it is in other local modeling methods.
- We introduce a geometric similarity in the loss function that helps to learn and reduce the fluctuation of the reconstructed surface.
- Our experimental validation shows that the proposed approach could

keep more details of the reconstructed shape in comparison with the original SDF decoder.

2.2 Related work

2.2.1 Learning SDF-based 3D shape representation

Learning a signed distance function to represent the 3D shape as a set of iso-surfaces recently received extensive attention in the field. Chen and Zhang [27] proposed to assign a value to each point in 3D space and use a binary classifier to extract an iso-surface. Mescheder et al. [87] utilized a truncated SDF to decide the continuous boundary of 3D shapes. In contrast with Chen and Zhang [27], they predicted the probability of occupancy in voxels, which could be used in a progressive multi-resolution procedure to get refined output. However, these methods target learning a discrete representation of SDF in the grid coordinate, and can be viewed as a learned shape-conditioned classifier for the decision boundary of the 3D shape surface. Differently, Park et al. [98] proposed to learn the SDF in a continuous field for 3D shape representation. In theory, the shape surface can be extracted with an arbitrary resolution with their model.

Note that obtaining the ground-truth signed distance from sampling points to the surface of the 3D shape typically requires rendering operation, which costs much time and computing power. To address this problem, Atzmon and Lipman [2] proposed a sign agnostic learning (SAL) algorithm to learn from unsigned distance from points to triangle soups directly. Gropp et al. [44] suggested using a geometric regularization paradigm to approximate the signed distance function, which can be achieved without 3D supervision and a direct loss on the surface of the shape. By encouraging the neural network to vanish on the input point cloud and to have a unit norm gradient, they achieved smooth and natural reconstructed 3D shape surfaces avoiding bad zero-loss solutions. Following, inspired by incorporating derivatives in a regression loss leads to a lower sample complexity, Atzmon and Lipman [3] generalized SAL with derivative regularization and showed a significant improvement in the quality of 3D shape reconstruction. However, these methods are limited to representing whole 3D shapes with a learned global SDF, struggling for accurate representation of complex geometry in local regions.

2.2.2 Representing 3D shape with local SDFs

Instead of learning a single SDF for representing a whole shape, Jiang et al. [58] designed an embedding of local crops of 3D shapes during training, and optimized a set of latent codes on a regular grid of overlapping crops with one single shared decoder when run on inference. Inspired by DeepSDF [98], Chabra et al. [18] replaced the dense volumetric SDF representation used in traditional surface reconstruction with a set of locally learned continuous SDFs defined by a single parameterized neural network. In contrast with the voxels (grid)-based representation of SDFs, Genova et al. [40] proposed a network to encode shapes into structured implicit functions (SIF) as a composition of local shape elements. Tretschk et al. [124] designed an explicit parametric surface model, which fits an implicit SDF in each local patch separated within multiple independent spheres. Hao et al. [50] represent 3D shapes as two levels of granularity with SDF, which provides semantic interpretability for the latent space of SDF in local parts of the shape. Moreover, they introduced a novel shape manipulation method by editing the primitives of local SDFs to obtain high-resolution 3D shapes. In the most recent works, Genova et al. [41] developed the SIF to learn a set of local SDFs that are arranged and blended according to an SIF template. They associated a latent vector within each local region, and these latent vectors are used to regress local SDFs for producing finer geometric details.

2.2.3 Geometric learning on 3D shapes

Generalizing neural networks to data with the non-Euclidean structure are known as Graph Neural Networks (GNNs) in the domain of geometric learning. Ranjan et al. [107] proposed to learn a non-linear representation of human faces by spectral convolutions with Chebychev biases [65, 29] as filters. Bouritsas et al. [9] replaced the convolution kernel with operators applied along a spiral path around the graph vertices. Hanocka et al. [49] leveraged the intrinsic geodesic connections of edges to define convolution operators, which inherited the direction invariant property as in 3D points convolution methods [104, 105]. Zhou et al. [150] further improved the reconstruction precision by using locally adaptive convolution operators for registered mesh data.

We incorporate geometric learning in our model, aiming to prompt information exchange of learning local SDFs in latent space. More details are in the next section below.

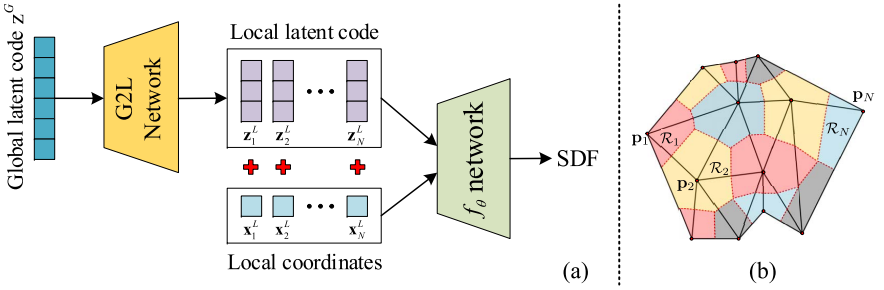

Figure 2.1: (a) Overview of the LGCL model. (b) Local region separation, we use a mesh template to assign sampling points to local regions. For a better illustration, we project these local regions into a 2D plane.

2.3 Local geometry code learning method

2.3.1 Modeling SDF locally

In the field of geometric shapes analysis, classical methods use the level set or a signed distance function to indicate implicitly a nutshell of the 3D object. Since it is difficult to give an explicitly mathematical formulation of the underline shape representation, we use a neural network f_θ (usually as multilayer perceptron network) with learnable parameters θ to represent a shape $\mathcal{S}$ as zero iso-surface:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid f_\theta(\mathbf{x}, \mathbf{z}) = 0\}, \tag{2.1}$$

where $f_\theta(\mathbf{x}, \mathbf{z}) : \mathbb{R}^3 \times \mathbb{R}^m \rightarrow \mathbb{R}$ target to regress a signed surface distance function. The latent code $\mathbf{z}$ decides the target shape by extracting sampling points $\mathbf{x}$ from the whole 3D space.

To improve representation accuracy, we decompose the 3D shape into a set of local parts and separately regress these shape parts with SDFs. Inspired by DeepLS setting [18], we generate a set of latent codes $\{\mathbf{z}_i^L\}$ and assign them to each shape parts. Similarly, we use a neural model f_θ to represent these parts as:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid \bigcup\nolimits_i \mathbb{1}_{\mathbf{x} \in \mathcal{R}_i} f_\theta(T_i(\mathbf{x}), \mathbf{z}_i^L) = 0\}, \tag{2.2}$$

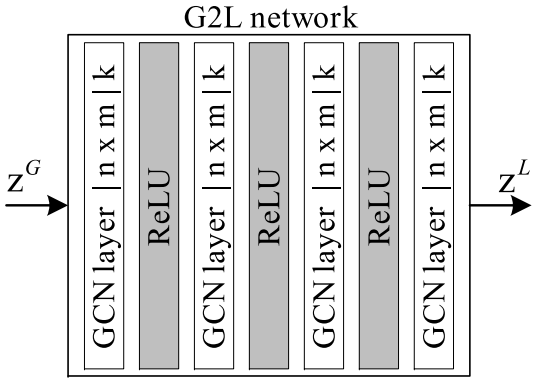

Figure 2.2: Architecture of the G2L network.

where $T_i(\cdot)$ supposes to transfer global location $\mathbf{x}$ to the local coordinate system $\mathbf{x}_i^L$ of a local region $\mathcal{R}_i$, and $\mathbf{z}_i^L$ indicates its related local latent code. This idea is illustrated in Figure 2.1 (a).

Different from splitting the 3D space into voxels [58, 18] or using explicitly parametric surface model [124, 50], we define the local region $\mathcal{R}_i$ according to a key point p_i as:

$$\mathcal{R}_i = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathrm{argmin}_{\mathbf{p}\in\mathcal{P}}\, d(\mathbf{x}, \mathbf{p}) = \mathbf{p}_i\}, \tag{2.3}$$

where $\mathcal{P}$ is a set of key points and $d(\cdot)$ is a distance function (*e.g.* , Euclidean distance). Note that the key points set $\mathcal{P}$ is only used for aligning the sampling points to their corresponding local latent code $\mathbf{z}_i^L$. One simple illustration for our region division method is shown Figure 2.1 (b). One patch with different color from their neighbours indicates a local region, which owns a corresponding latent code.

2.3.2 Geometric leaning on local latent codes

Since we do not use a regular structure to split 3D space as voxels, it is hard to remove invalid local regions by distinguishing the coordinates of sampling points. These invalid regions do not have any assigned sampling points in the learning process, resulting in invalid latent codes used in inference. In the meantime, we do not split the 3D space into overlapped subspaces. Thus these local latent codes are learned independently, leading to inconsistent surface

estimates at the local region boundaries as mentioned in Chabra et al. [18].

To solve these problems, we propose to make local SDFs learn not only from training data but also from each other. This can force local SDFs in invalid regions to learn from their valid neighbors, and make shape parts harmonious at local region boundaries. Inspired by the geometric learning of COMA [107], Neural3DMM [9], and FCM [150], we introduce the mesh structure of 3D shape as a "scaffold" to put key points and propagate information between local latent codes. Then we can get two benefits that it allows each local region to have an intersection with the 3D shape and construct communications among them by representing the mesh as a graph. Specifically, a 3D surface mesh can be defined as a set of vertices $\mathcal{V}$ and edges $\mathcal{E}$. In our situation, we replace $\mathcal{V}$ with $\mathcal{P}$ to define the local regions as shown in Figure 2.1 (b).

Cooperating with several graph convolution layers to construct a graph network $\mathrm{G2L}(\mathcal{E}, \mathbf{z}^G) : \mathbf{z}^G \rightarrow \{\mathbf{z}_i^L\}$, we could predict the local SDF with the local latent codes aligned to these local regions, as shown in Fig 2.1 (a). Such graph neural network provides geometric deformations on each local latent code. Consequently, each local latent code can contribute to the shape representation since each key point is on the shape.

Since our method gets benefits from modeling SDF in **Local** with latent codes and learning these latent codes through **Geometric Learning** with graph neural networks, thus we name it Local Geometry Code Learning (LGCL).

2.3.3 Loss function

Sign agnostic learning. Inspired by advanced works of SAL [2] and IGR [44], we do not request the true distance of sampling points to the surface of shape during training. Instead of getting the true distance in a rendering way, directly calculating the distance from a point to a triangle soup is more convenient and efficient, and also without the requirement of watertight structures. Thus, we construct the basic loss function as:

$$\mathcal{L}_{\text{basic}} = \mathcal{L}_{\text{sal}} + \mathcal{L}_{\text{igr}}, \tag{2.4}$$

where $\mathcal{L}_{\text{sal}}$ just needs the unsigned distance d_{u} from point $\mathbf{x}$ that sampled from whole space Ω to the shape $\mathcal{S}$, and it is defined as:

$$\mathcal{L}_{\text{sal}} = \mathbb{E}_{\mathbf{x}\in\Omega}\Big|\,\big|f_\theta(T_i(\mathbf{x}), \mathbf{z}_i^L)\big| - d_{\text{u}}(\mathbf{x}, \mathcal{S})\Big|. \tag{2.5}$$

For the $\mathcal{L}_{\text{igr}}$, we use its variant type from Sitzmann et al. [119] as:

$$\begin{aligned}\mathcal{L}_{\text{igr}} = \lambda_{\text{grad}}\mathbb{E}_{\mathbf{x}\in\Omega}(\|\nabla_{\mathbf{x}} f_\theta\|_2 - 1)^2 + \\ \mathbb{E}_{\mathbf{x}\in\Omega_0}\|\nabla_{\mathbf{x}} f_\theta - n(\mathbf{x})\|_2,\end{aligned} \tag{2.6}$$

where Ω_0 means the domain of zero-iso surface of the shape, $\|\cdot\|_2$ is the euclidean 2-norm.

Geometric similarity loss. There is a contradiction between the distributions of key points and sampling points. Consider a complex part of the shape, it needs more key points to get more local latent codes for better modeling. However, the more key points are allocated, the harder the optimization of local latent codes is. Since each local region would be smaller and get fewer sampling points for training. Even if increasing the number of sampling points, it is still difficult to guarantee assigning enough sampling points to each local latent code.

To alleviate this problem, we propose a loss $\mathcal{L}_{\text{sim}}$ to make these local latent codes not only learn from the sampling points but also learn from each other. The assumption here is the difference between the adjacent local latent codes is smaller than the ones that are far away from each other. On the other hand, it provides a kind of regularization effect on the local latent codes that cannot get sufficient training, which forces them to be similar to their neighbors.

Again, we use the geometric structure as a graph to calculate $\mathcal{L}_{\text{sim}}$ as:

$$\mathcal{L}_{\text{sim}} = \frac{1}{N_{\text{v}}}\sum_{i}^{N_{\text{v}}}\left|\mathbf{z}_i^L - \sum_{k}^{K} G_{\text{l}}(\mathbf{z}_i^L, \mathcal{N}_k(i))\right| \tag{2.7}$$

where $G(x_i, \mathcal{N}(i)) : x = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}(x_j)$ means to update the value of x_i by the average value of its neighbours. Here $\mathcal{N}_k(i)$ means the neighbours of vertex i in the k layer. And for better learning, we increase the neighbor region of one local latent code by K layers. We use $K = 3$ layers for our experiments.

Total loss Our final loss function consists of above losses and a regular term $\|\text{z}^G\|$ as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{sim}}\mathcal{L}_{\text{sim}} + \mathcal{L}_{\text{basic}} + \lambda_{reg}\|\text{z}^G\|_2 \tag{2.8}$$

In our experiments, we use the setting of $\lambda_{\text{grad}} = 0.1$, $\lambda_{\text{sim}} = 1.0$ and $\lambda_{\text{reg}} = 0.001$ if without extra explanation.

2.4 Experiments

2.4.1 Experimental setup

In our experiments, all of the used models are trained and evaluated mainly on a subset of the D-Faust dataset [7], which is the No.50002 subset of mesh registrations for 14 different actions about the human body, such as leg and arm raises, jumps, etc. Due to the low variation between adjacent scans, we sample the used dataset at a ratio of 1:10 and then split them randomly with 90% for training and 10% for the test. Our data pre-processing method inherits from both IGR [44] and SAL [2], which will generate 600K sampling points from each object, 300K are sampled on the object surface with their normals, and the other 300K are sampled from two Gaussian distributions centered at points on the object surface.

As one of the main baselines, we train the DeepSDF with the same setting in Park et al. [98] on the completed No.50002 sub-dataset of the D-FAUST dataset. In addition, we also train two other different sizes of DeepSDFs with the loss function $\mathcal{L}_{\text{basic}}$: SDF-8, which is similar to the original DeepSDF, but with 8 fully connected layers and 512 neurons in each hidden layer. The dimension of its latent code is 256. One skip connection is also used at the fourth layer with the latent code. SDF-4, has 4 fully connected layers, 128 neurons in each one and no skip connection, and the length of latent code is 64. Each fully connected layer in both SDF-8 and SDF-4 except the last one is followed by Softplus activation and initialized as in Atzmon and Lipman [2].

In our LGCL-based method, we use a 4-layer G2L (shown in Figure 2.2) network followed by an SDF-4 network. The graph convolution kernels in the G2L are from [107] as chebConv or [150] as vcConv, we gave the evaluation of their performance in the following results. we re-implemented the kernels in a more compact form, and the parameters of one graph convolution layer can be represented as $(\mathrm{n}, \mathrm{m}, \mathrm{k})$, where n means the size of input channel and m is the size of the output channel. k stands for the size of the Chebyshev filter when using chebConv and the number of weight basis when using vcConv. More details about the architecture of our models can be viewed in Table 2.1. The local latent code for each vertex of the G2L is set to an 8-length vector.

We train all the models with 300 epochs at a learning rate of 5e-4 for the parameters of neural networks and 1e-3 for latent code optimization. Both learning rates are decayed to half after 200 epochs. We evaluate all the methods on the split test dataset. As same as in DeepSDF [98], the latent code will be estimated with the frozen neural network before the inference.

Table 2.1: Setting of the architecture in the LGCL method. LGCL-VC means using the graph convolution kernels from [150] and LGCL-Cheb is from [107].

	LGCL-VC	LGCL-Cheb
G2L	vcConv(8,8,8) vcConv(8,16,16) vcConv(16,32,32) vcConv(32,64,64)	chebConv(8,8,6) chebConv(8,16,6) chebConv(16,32,6) chebConv(32,64,6)
SDF-4	Linear(67,128) Linear(128,128) Linear(128,128) Linear(128,1)	

2.4.2 Reconstruction results

To evaluate the performance of reconstruction, we measure the Euclidean Distance (ED) from the vertices of ground truth to the surface of reconstruction generated from different methods. We also report our results under the metrics of Chamfer Distance (CD) and Hausdorff Distance (HD).

The HD d_{HD} is calculated as:

$$d_{\mathrm{HD}} = \max(h(\mathcal{P}_{\mathrm{A}}, \mathcal{P}_{\mathrm{B}}), h(\mathcal{P}_{\mathrm{B}}, \mathcal{P}_{\mathrm{A}})), \tag{2.9}$$

where $\mathcal{P}_{\mathrm{A}}$ and $\mathcal{P}_{\mathrm{B}}$ represent two sets of point cloud, and $h(\cdot)$ is defined as:

$$\begin{cases} h(\cdot) = \max\limits_{a \in \mathcal{P}_{\mathrm{A}}} \min\limits_{b \in \mathcal{P}_{\mathrm{B}}} \|a - b\|_2 \\ h(\cdot) = \max\limits_{b \in \mathcal{P}_{\mathrm{B}}} \min\limits_{a \in \mathcal{P}_{\mathrm{A}}} \|b - a\|_2 \end{cases}. \tag{2.10}$$

The CD d_{CD} is calculated as:

$$d_{\mathrm{CD}} = \frac{1}{\mathcal{P}_{\mathrm{A}}} \sum_{a \in \mathcal{P}_{\mathrm{A}}} \min_{b \in \mathcal{P}_{\mathrm{B}}} \|a - b\|_2 + \frac{1}{\mathcal{P}_{\mathrm{B}}} \sum_{b \in \mathcal{P}_{\mathrm{B}}} \min_{a \in \mathcal{P}_{\mathrm{A}}} \|a - b\|_2. \tag{2.11}$$

Due to CD and HD being applied on the point cloud, then we sample 30000 points on both surfaces of ground truths and reconstructions. For a more fair comparison, we also list the size of network parameters and latent codes of different methods, while both are necessary to represent 3D shapes. All of these quantitative results are shown in Table 2.2.

Table 2.2: Quantitative evaluation. CD means Chamfer Distance, HD means Hausdorff Distance and ED stands for Euclidean Distance of the point to surface. All the distances are represented in millimeters. We directly run the DeepSDF code as the baseline.

Model	Net Params	Latent Params	CD	HD	ED	
					Mean	Std
DeepSDF	1.84 M	0.26 K	0.28	68.11	12.51	16.37
SDF-8	1.58 M	0.26 K	0.22	59.86	6.94	10.30
SDF-4	41.86 K	0.06 K	2.20	119.56	24.45	31.19
LGCL-Cheb	58.42 K	55.12 K	2.56	108.58	2.51	1.87
LGCL-VC	0.19 M	55.12 K	1.55	71.67	3.35	2.44

As one can see SDF-only-based methods need more network parameters to achieve comparable performance with our method. We can see that there is a positive correlation between the size of the DeepSDF network and its performance on reconstruction, as all the quantitative results of SDF-4 are worse than SDF-8's.

By introducing local latent codes, our LGCL-based model outperforms SDF-4 by approximately one order of magnitude under the metric of Euclidean distance. Even compared to the DeepSDF-8 which has a huge size of parameters, our results still have competitive advantages and obtain the smallest Euclidean error as 2.51 ± 1.87 mm with chebConv kernels.

Table 2.3: Statistics of reconstruction errors.

Model	Error(mm)		Percentage(%)		
	$< 50\%$	$< 90\%$	> 5 mm	> 10 mm	> 20 mm
DeepSDF	6.19	37.79	57.42	33.27	17.44
SDF-8	3.55	18.59	34.79	14.14	6.39
SDF-4	11.91	73.42	34.80	43.07	25.37
LGCL-Cheb	2.13	5.21	11.44	0.03	0.00
LGCL-VC	2.92	6.85	24.47	0.98	0.00

More details about the Euclidean errors of different methods can be found in Table 2.3. Consequently, LGCL-Vc decreases the errors of CD and HD of

Table 2.4: Influence of geometric similarity loss, all results are shown in millimetres.

Model	λ_{sim}	CD	HD	ED		
				Mean	Std	Median
LGCL-VC	0.1	2.66	105.59	2.45	1.72	2.16
	1.0	1.55	71.67	3.35	2.44	2.92
	10.0	1.45	66.05	4.07	2.97	3.56
LGCL-Cheb	0.1	2.43	108.65	2.72	1.91	2.39
	1.0	2.56	108.58	2.51	1.87	2.13
	10.0	2.74	108.69	4.44	3.11	3.94

SDF-4 by about 30% and 40% respectively.

We visualize two examples of the Euclidean error of each vertex shown in Figure 2.3. It obviously shows that the small size of the DeepSDF network struggles to reconstruct the details, note that it almost loses the whole hands of the human. In contrast, our LGCL model could keep more information in local regions though it causes more fluctuation.

2.4.3 Ablation study

We perform ablative analysis experiments to evaluate the influence of our proposed geometric similarity loss $\mathcal{L}_{\text{sim}}$. It is controlled by adjusting its coefficient λ_{sim} to constrain the similarity between local latent codes with their neighbors. As shown in Table 2.4, the geometric similarity loss takes different influences on LGCL-VC and LGCL-Cheb. Specifically, it tends to get better ED results with less constraint on similarities of local latent codes of LGCL-VC.

In contrast, one needs more similar local latent codes to decrease the errors of CD and HD since the large freedom of graph convolution kernels that used in LGCL-VC. And for LGCL-Cheb, it implicitly has a stronger geometric constraint set by its ChebConv kernels. Thus the extra geometric similar loss has little impact on the errors of CD and HD, but it should be patient to pick the adopted when you consider the ED errors.

We found some interesting explorations on the influence of different methods for getting local latent codes, as shown in Figure 2.4. In our LGCL-based methods, we split the global code $\mathbf{z}^G$ evenly into parts, which is equal to the number of the graph vertices, and then align these parts to different vertices. However, for the result in Figure 2.4 (b), we directly align the same initial

input, which is the global latent code, to each vertex of the G2L. Since each local latent code has the same initial value, it provides a similarity constraint implicitly between them. Then we do not introduce the $\mathcal{L}_{\text{sim}}$. In this case, each local region of reconstruction tends to shrink to the same type of mini polyhedron. We attribute this degeneration of modeling local SDFs to the over-constraint on the similarity among the local latent codes, which introduces a limitation to their geometric deformation.

We also show the results without the usage of $\mathcal{L}_{\text{sim}}$ in our LGCL-based methods in Fig 2.4 (d). It is obvious to see the dramatic vibrations in some local regions. We argue that it is caused by insufficient training of some local latent codes in corresponding local regions.

Furthermore, as shown in Figure 2.4 (c), the result looks like a fall between (b) and (d), and has both similar polyhedrons and vibrations that exist in local regions. We got this result by modifying the G2L network as a pyramid structure as in the COMA [107] decoder. The modification changes the graph structure of each graph convolution layer with the pooling layer. The pyramid structure provides a cluster mapping from top to bottom in this G2L variant. In other words, the mentioned modification adds an extra similarity constraint on local latent codes within a larger geometric range. However, the approach is also limited to deforming the local latent codes and leads to a compromise result compared to (b), (d), and our major results in Figure 2.3.

2.5 Conclusion

In this work, we propose the LGCL method which is based on a new architecture for the local geometry learning handling. The idea is to perform the learning regression process directly in the latent code space. Consequently, our approach makes general GNNs more flexible, compact, and simple in realization. The experimental results show that our method considerably outperforms baseline DeepSDFs both in accuracy and model size. We think that our architecture is novel and promising, and can be further improved in future work.

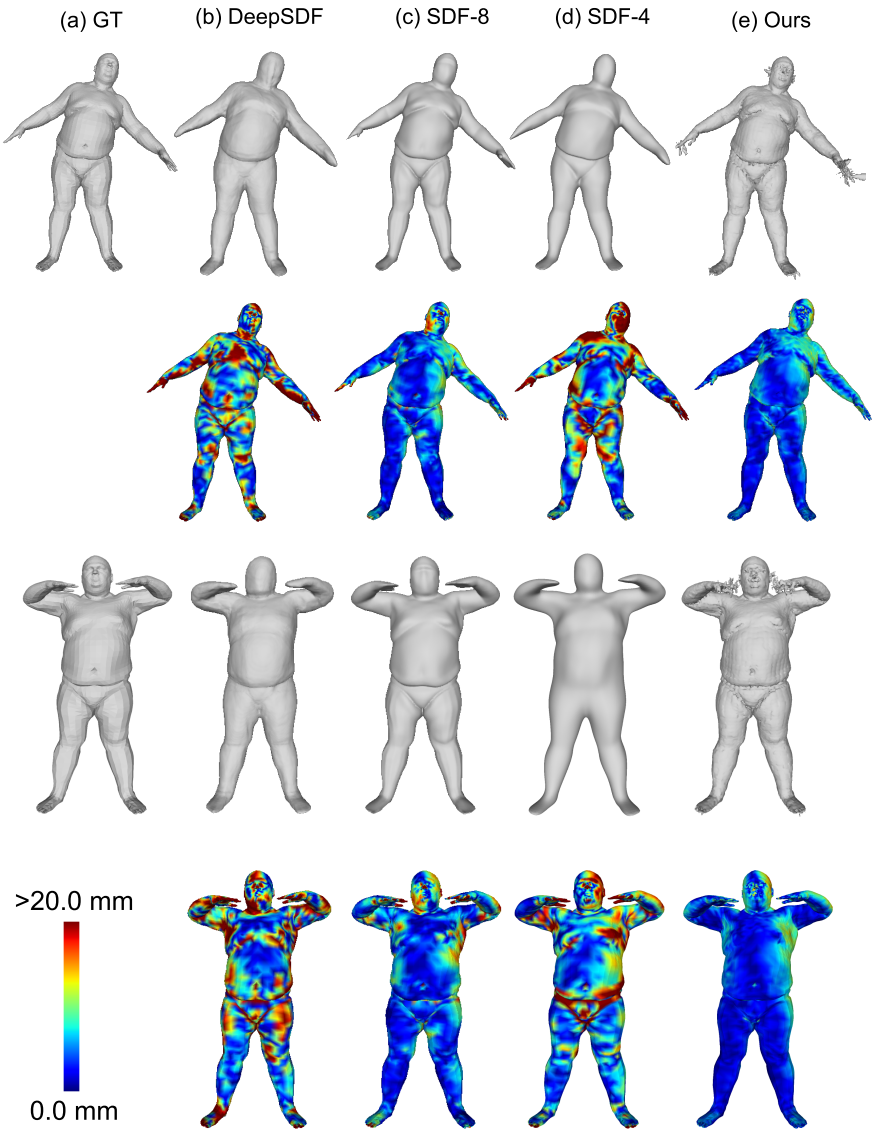

Figure 2.3: Visualization of the per-vertex Euclidean error of the reconstructions. GT means the ground truth shape, the model of Ours here used the LGCL-VC.

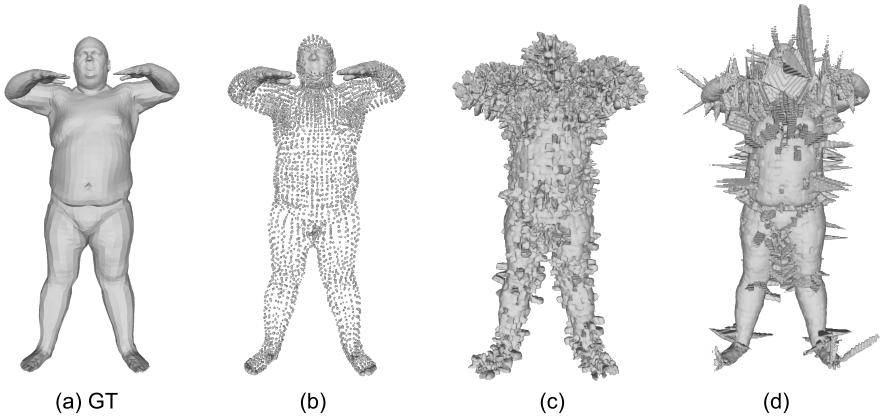

Figure 2.4: Failed results of different methods for getting local latent codes. (a) is the ground truth (GT); (b) uses G2L but with the same initial input of global latent code for each graph vertex; (c) uses a pyramid G2L to get local latent codes; (d) does not include $\mathcal{L}_{\text{sim}}$. Please check more details in Sec 2.4.3.

3 A Continuous Digital Elevation Representation Model for DEM Super-resolution*

3.1 Introduction

Terrain elevation surfaces have significant value for a wide range of analyses, including hydrology [70, 8], ecology [131], geomorphology [134], among others. They also find applications in terrain visualization [109], urban design [103], terrain-aided navigation [129]. Nevertheless, the currently available digital elevation data are limited to discrete representations [74, 54] such as digitized points, contour lines, triangulated irregular networks (TIN), and gridded surfaces. Discretization, without taking into account the accuracy of geographical measurements, restricts the precision of these representations based on factors such as the sparsity of points, the interval between contour lines, the number of triangles, and the cell size of the grid. In practice, these discrete representations often require transformations to meet the specific requirements of different applications. For instance, converting data to triangular surfaces enables 3D terrain visualization, while converting it to contour lines facilitates terrain topology analysis. However, these conversions inevitably compromise the fidelity of elevation data.

In contrast to discrete approximations, our aim is to preserve the continuity of topographic elevation data in the real world. Drawing inspiration from advanced research on implicit neural representation models, we propose a parametric neural network that maps coordinates to elevation values. These coordinates, along with their corresponding elevation values, are associated with query positions, allowing elevation values to be obtained for any arbitrary position. This idea is illustrated in Figure 3.1.

To demonstrate the accuracy and versatility of our continuous digital elevation model (CDEM), we extend its application to super-resolution tasks for raster digital elevation models (DEMs). A neural encoder is introduced to generate latent codes for the target elevation data. These latent codes, along with the coordinates, are then input to the CDEM to derive the desired elevation values. In contrast to many existing learning-based DEM super-

*This chapter is based on a submission under review in ISPRS Journal of Photogrammetry and Remote Sensing (2024) [141].

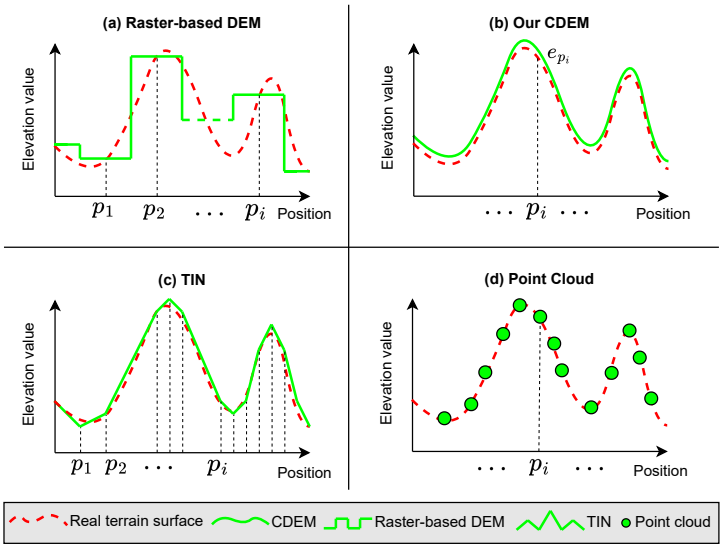

Figure 3.1: Illustration of the proposed CDEM representation vs. other digital elevation models. (a) A raster-based DEM representation. (b) Proposed continuous digital elevation model (CDEM) based on a neural network. Note that we omit the encoder structure here to highlight our core idea. (c) A triangulated irregular network (TIN) representation. (d) A discrete point cloud representation. The red dashed curve shows a real-world continue elevation, and the green ones correspond to different representation curves. Note that the raster data consists of cells, and each cell indicates the elevation value of a local region. Representation accuracy and capacity depend on the cell size. The TIN represents terrain surfaces as triangular facets. Thus, the number and density of triangle vertices are key factors for an accurate representation. The point cloud is composed of numerous discrete spatial points, where the number and density of these points limit the accuracy and range of the representation. All of these digital elevation models are constrained with a fixed number of elevation values. In contrast, our CDEM represents the terrain by network weight, which can be used to obtain elevation values at any arbitrary point of the represented surface. This is the main difference between our representation and standard raster representations.

resolution methods that require a specific super-resolution scale or fixed-size

inputs/outputs, our CDEM-based model, which relies on coordinate inputs, offers a more flexible approach for application, devoid of such limitations. As our model's predictions are based on query positions, it allows for the utilization of any number of elevation values and their corresponding positions during training. This implies that our model can be trained simultaneously for different super-resolution scales. Moreover, during testing, any query position can be employed to predict elevation values, resulting in a super-resolution DEM with the desired resolution.

Due to the analogous grid structure of DEMs and images, several advanced DEM super-resolution techniques have been adapted from the field of image processing. Unlike RGB values (red, green, and blue) in images that are confined to the range of 0–255, elevation values can span a wide and arbitrary range. Furthermore, the diverse landscapes contribute to a complex and varied distribution of elevation values. All these characteristics pose challenges for neural networks in accurately predicting elevation values. Therefore, we suggest predicting the elevation bias instead of the elevation value in our CDEM model. This proposed elevation bias represents the vertical deviation between a query position and its nearest known neighbor. By amplifying relatively small variations compared to the original elevation values, the elevation bias modifies the data distribution to be more concentrated. An example is illustrated in Figure 3.2.

Moreover, real-world terrain surfaces exhibit inherent roughness, with numerous small variations in localized regions that are difficult to predict with precision. We draw inspiration from coordinate-based neural representation models applied to 3D shapes, 3D scenes, and 2D images [135, 147, 119, 92, 90]. These models have demonstrated that mapping coordinates to a higher-dimensional space, or increasing their complexity using positional functions before feeding them into the network, results in improved data fitting, particularly for data containing high-frequency variations. Thus, we suggest integrating a positional encoding function into our CDEM model. Our aim is to enhance prediction accuracy in regions with high-frequency variations and minimize the distortion of terrain features such as ridges, rivers, saddles, cliffs, mountains, and more.

Our experiments provide quantitative and qualitative results based on common terrain metrics, including elevation value, slope, and aspect. We evaluate our approach using four DEM datasets of varying resolutions. In summary, our CDEM demonstrates superior performance in super-resolution tasks, excelling across various super-resolution scales ($\times 2$ to $\times 8$) as well as different DEM resolutions (2m, 10m, and 30m). Additional experiments further validate the generalization capabilities of our CDEM model for practical DEM super-resolution tasks.

The contributions of this chapter are as follows:

- We propose a continuous digital elevation model (CDEM) that represents terrain elevation data in a continuous format, enabling the prediction of elevation values at arbitrary positions using a parametric neural network.
- We propose the utilization of elevation biases for CDEM prediction, which amplify relatively small variations in elevation values and modify the data distribution to be more concentrated.
- We propose the inclusion of a positional encoding function in the CDEM to improve prediction accuracy.
- We finally propose an encoder-based EBCF-CDEM model for DEM super-resolution tasks. Extensive experiments on four DEM datasets validate that our model can achieve more accurate super-resolution results and has better generalization performance than other methods.

3.2 Background

3.2.1 Implicit neural representation models

A 3D shape can be represented using the signed distance function [82, 19], an image can be reconstructed by solving the Poisson equation [102], and a transformation between a 3D scene and 2D views can be achieved by rendering [59]. These applications are based on implicit functions, which have attracted significant attention for deep learning models in recent years. Different from directly representing 3D shapes and images with surfaces (or points) and pixels, implicit neural representation models employ coordinate-based neural networks to predict desired signals continuously [118, 133, 138, 21].

Park et al. [98] proposed to predict the signed distance value for any query position with a neural network, and the value is used in 3D surface reconstruction. Mescheder et al. [87] proposed to predict the occupancy for every voxel with a neural network, and the occupant voxel would be used to represent 3D objects. Mildenhall et al. [90] proposed to predict the color and transmittance at any view-dependent location with a neural network for rendering. Peng et al. [100] proposed to estimate offsets and normals for point clouds with a neural network, and these offsets and normals are used for 3D shape reconstruction. Oechsle et al. [95] proposed to predict color values of spatial points for texture reconstruction of 3D objects. These positions,

voxels, points, and locations are encoded as coordinates for inputting to neural networks.

In the meanwhile, for 2D images, Sitzmann et al. [119] proposed to predict gradients or Laplacian of grid coordinates with a neural network for image reconstruction. Chen et al. [24] proposed to predict RGB values of grid coordinates with a neural network for image super-resolution. Lee and Jin [67] introduced a dominant-frequency estimator based on coordinates to enhance the performance of Chen et al. [24]. Zhang et al. [142] proposed to predict spectral radiance values of grid coordinates with a neural network for generating high-resolution hyperspectral images.

All the above coordinate-based neural networks have been proven for accurate and efficient representation tasks, including 3D shapes, 3D scenes, 2D images, etc. However, as far as we know, there is no work for representing terrain elevation values by a continuous function parameterized by a neural network. Therefore, we propose the CDEM for elevation data representation and hope it could serve as a potent reference for researchers.

3.2.2 DEM super-resolution methods

The topic of DEM super-resolution can be retrospected to Xu et al. [136], which aims to improve the resolution of DEMs without costing extra high-accuracy measurements. Specifically, Xu et al. [136] proposed to recover high-resolution DEMs by the weighted sum of similar patches from learning samples, where the weights are estimated in a searching and optimizing way. Benefiting from the superior performance of CNN applied for image super-resolution [33], Chen et al. [25] introduced D-SRCNN to improve the compatibility and robustness of DEM super-resolution learning. Following, Zhang et al. [143] proposed to recursively deal with DEM cells using a sub-pixel CNN to build high-resolution DEMs. They showed such a recursive operation could speed up the feature extraction process. Another model that focuses on DEM super-resolution efficiency is proposed by Demiray et al. [31]. They increased DEM spatial resolution by introducing and modifying MobileNetV3 [55]. In the meanwhile, generative adversarial networks (GANs) were introduced to DEM super-resolution by Demiray et al. [30] and were further investigated by Zhang and Yu [144]. These studies take advantage of advanced image super-resolution models to achieve superior DEM super-resolution results.

However, DEMs are very different from color images. For example, the range of elevation values is much wider than RGB values, and terrain characteristics (e.g., slope and aspects) are not similar to images. To better leverage the CNN for DEM super-resolution, Xu et al. [137] adapted pre-trained EDSR [75] to

predict gradient maps, and high-resolution DEMs are constructed based on these estimated gradients and low-resolution DEMs. To better extract and fuse terrain features, Zhou et al. [148] proposed a double-filter deep residual block, in which filters have different receptive fields. Furthermore, Zhang et al. [145] proposed a terrain feature-aware super-resolution model (TfaSR), in which a deformable convolution module is integrated into a deep residual structure for extracting terrain features adaptively. They experimentally show the superior performance of TfaSR on terrain feature preservation in DEM super-resolution results.

In summary, the above DEM super-resolution models are designed for a fixed-size output and a specified super-resolution scale, which limits model application for multi-scale DEM super-resolution and prevents model learning from multi-spatial resolution DEM data. Thanks to our continuous DEM representation, the proposed EBCF-CDEM could generate high-resolution DEMs at multiple scales within one model and learn from multi-spatial resolution samples in the training procedure.

3.3 Method

3.3.1 Implicit neural model for elevation representation

In our continuous DEM representation (CDEM), an elevation value e at an arbitrary 2D query position q can be implicitly obtained by a parameterized neural model f_θ as:

$$e = f_\theta(\gamma(q), z(q, \mathrm{C})) \tag{3.1}$$

where γ is a positional encoding function to generate a d-dimension coordinate for q, we will have more discussions in Section 3.3.3. $\mathrm{C} = \{(p_i, \mathrm{c}_i)\}_{i=1}^{N}$ is a set of packs composed of a vector c_i with its 2D position p_i, more details about getting C are in Section 3.4. And $z(\cdot)$ means to retrieve related vectors $\{\mathrm{c}_i\}$ (we call them latent codes from now on) and aggregate them by $(q, \{p_i\})$ as:

$$z(q, \mathrm{C}) = \sum_{p_i \in \mathcal{N}(q)} w_i \cdot \mathrm{c}_i \tag{3.2}$$

where $\mathcal{N}(q)$ represents all retrieved neighbors of q, w_i is the interpolation weight corresponding to c_i and its value depends on the dimension of p_i and q.

3.3.2 Elevation bias prediction

Compared to RGB values (red, green, and blue) in images, which are always in the range of 0–255, elevation values can be arbitrary and present a much wider range. Furthermore, various landscapes make elevation values present a complex and diverse distribution. All of these characteristics are challenging for neural networks to predict elevation values.

To facilitate the model prediction, we propose to use the elevation bias between a query position q to the nearest known position, instead of directly predicting the elevation value. Our motivation comes from a natural phenomenon [136, 145] that elevation values in a local terrain region are highly related.

Formally, assume a predicted elevation value e_q at the query position q and its nearest know position $\tilde{p}$ with its known elevation value $e_{\tilde{p}}$, and based on Equation 3.1, the target of our neural model prediction can be transformed to:

$$f_\theta(\gamma(q), z(q, \mathrm{C})) \rightarrow e_q - e_{\tilde{p}}, \quad \tilde{p} = \underset{p_i \in \mathcal{N}(q)}{\arg\min} |p_i - q| \tag{3.3}$$

We show an example to illustrate the effect of the elevation bias prediction in Figure 3.2. The original high-resolution and low-resolution elevation values are normalized into [0,1] for better visualization. Since there are many relatively small variations in high-resolution elevation data, using neural networks directly to predict elevation values would omit such variations. As mentioned above, the distribution of elevation values is dispersive and would vary drastically across different landscapes. However, if we instead predict the elevation bias based on a known low-resolution to the target high-resolution elevation data, these small variations would be amplified, and the distribution of bias values would be more concentrated (shown in Figure 3.2 (d)).

3.3.3 Positional encoding

Learning to reconstruct high-frequency variation of elevation values is a key factor for obtaining accurate CDEM representations. However, deep neural networks are biased toward learning lower-frequency functions [106]. Though we can train neural networks in a spatially-adaptive progressive encoding way [53], using positional encoding for coordinate-based neural networks is more widely in representing 3D shapes, 3D scenes, and 2D images [135, 147, 119, 92, 90]. In addition, elevation values of a raster DEM are typically considered as 2.5D representations. Thus, to minimize the distortion of terrain features (including ridges, rivers, saddles, cliffs, mountains, and so on) and improve the

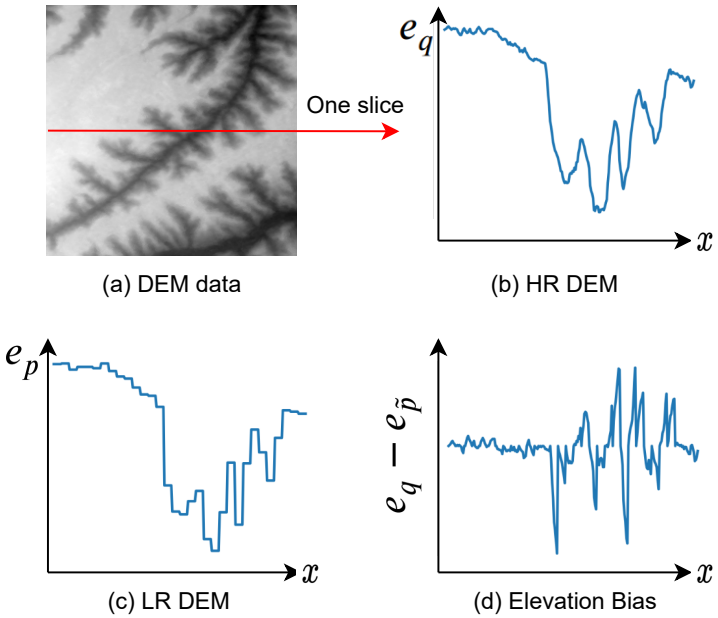

Figure 3.2: Illustration of the proposed elevation bias. (a) Visualization of the demo DEM data. (b) The elevation values e_q along the red arrow slice of the demo high-resolution (HR) DEM. (c) The elevation values e_p along the red arrow slice of the demo low-resolution (LR) DEM. (d) The elevation bias between elevation values of HR and LR DEMs using Equation 3.3.

reconstruction accuracy at high-frequency variation regions, we introduce the positional encoding function γ into our implicit neural model.

Specifically, $\gamma(x)$ maps x into a higher dimensional space and is defined as:

$$\begin{aligned}\gamma(x) = (&\sin(2^0 x), \sin(2^1 x), ..., \sin(2^{L-1} x),\\ &\cos(2^0 x), \cos(2^1 x), ..., \cos(2^{L-1} x))\end{aligned} \tag{3.4}$$

where L is a hyper-parameter to specify the dimension ($\mathbb{R}^{2L}$) of mapping space. Note that $x \in \mathbb{R}$ is a scalar coordinate and is normalized to lie in [-1,1]. It needs to apply γ separately on each coordinate when processing multi-dimensional inputs.

3.4 Learning CDEM for super-resolution

Our CDEM representation independently queries elevation values at different 2D positions, and it is suitable for DEM super-resolution tasks. In this section, we propose a DEM super-resolution model based on our CDEM representation. As shown in Figure 3.3, inspired by some advanced image super-resolution research [67, 24, 69] we construct a training and testing pipeline with an Encoder-Decoder structure.

Firstly, an encoder E_ϕ is introduced to extract latent codes $\{c_i\}$ from a low-resolution DEM. In the meanwhile, each latent code needs to align with the corresponding 2D position p_i and elevation value e_{p_i} of the same low-resolution DEM. Kind of such an encoder could be CNN- and Transformer-based neural networks since we hope to train E_ϕ and f_θ jointly, and hope the E_ϕ is effective in features extraction. In practice, we use the EDSR-baseline encoder [75] as E_ϕ in consideration of computation burden and training time.

In the training procedure, the query position q with its true elevation value e_q is known from the high-resolution DEM. Therefore, the inputs to the decoder f_θ could be calculated based on Equation 3.2 and Equation 3.4. Then, after obtaining the prediction target with Equation 3.3, a training loss function is computed for optimizing neural model parameters ϕ and θ. Referring to the loss function analysis [146, 75] for image super-resolution models, we select L1 loss in our experiments. This setting is also used in some DEM super-resolution methods [47, 77, 148].

Since our CDEM-based super-resolution model can be regarded to compensate for the elevation bias based on known low-resolution DEMs, we use the abbreviation EBCF-CDEM in the following descriptions.

3.5 Experiments

To evaluate the performance of our CDEM in predicting elevation value at any given position, we conduct DEM super-resolution experiments in this section. However, we cannot obtain all the elevation values in practice. Therefore, we simulate the prediction from low-resolution DEM samples and evaluate them with high-resolution DEM samples. As the low-resolution DEM samples are created with varying downsampling factors, it is akin to applying our CDEM model at multiple distinct positions. If the resolution of high-resolution DEM samples is sufficiently precise, and an adequate number of downsampling factors are used, our experiments can be considered as an almost infinite acquisition of elevation values within a real scene. The experiments are organized as follows.

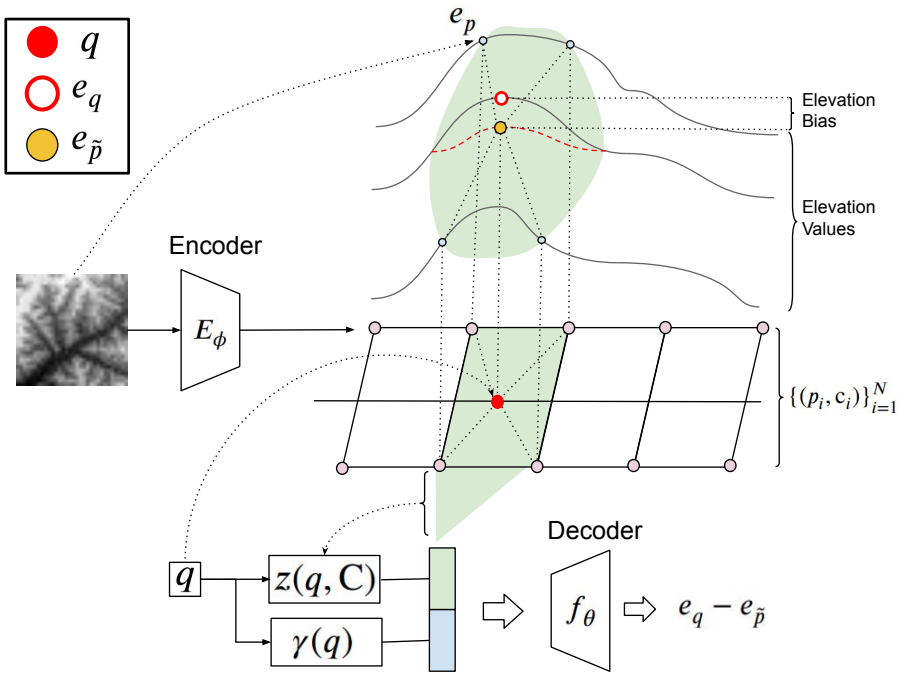

Figure 3.3: The framework of the proposed EBCF-CDEM is based on an Encoder-Decoder structure. The Encoder E_ϕ is used to extract latent codes $\{c_i\}$ from low-resolution DEMs. Giving an arbitrary query position q, a corresponding latent code would be generated by $z(q, C)$ (Equation 3.2), and a positional encoding function γ (Equation 3.4) is introduced to map q to high-dimensional coordinates. Then, the latent code and coordinates are together fetched into the Decoder f_θ for predicting the elevation bias $e_q - e_{\tilde{p}}$ (Equation 3.3).

Firstly, we show accurate predictions of the EBCF-CDEM model with various super-resolution scales $\times 2$, $\times 4$, $\times 6$, and $\times 8$ on two DEM datasets with 30 m and 2 m resolutions. Then we reveal the influence of elevation bias prediction and positional encoding through an ablation study. Finally, we organize two experiments to highlight the generalization ability of our EBCF-CDEM model. One is to directly apply the trained EBCF-CDEM on a different dataset without

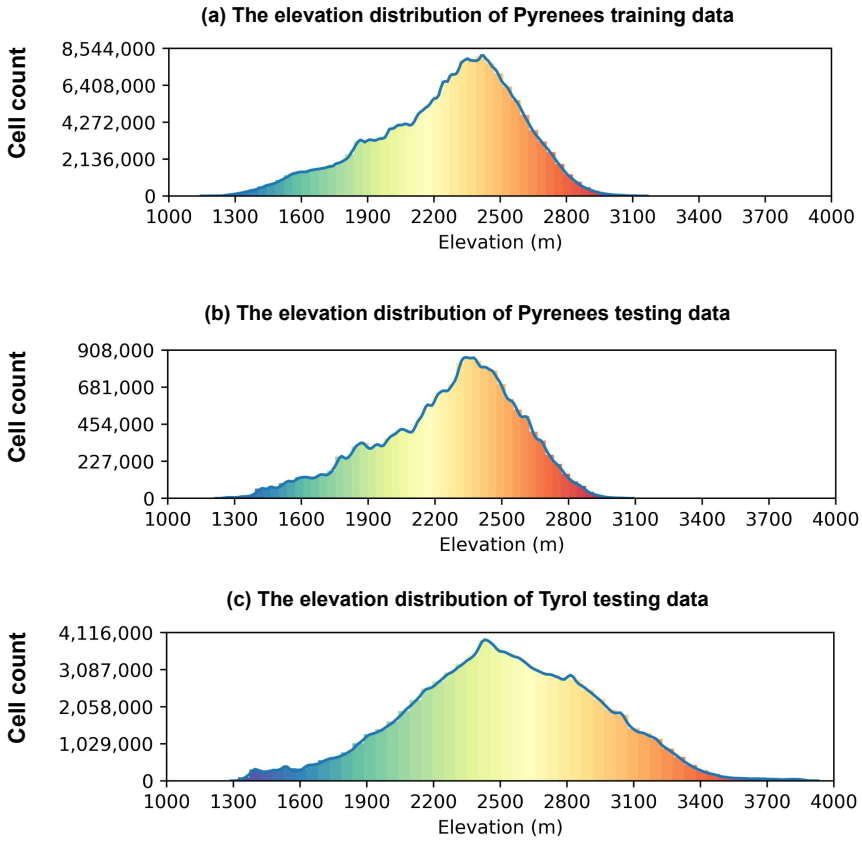

Figure 3.4: The elevation distribution of (a)(b) Pyrenees dataset and (c) Tyrol dataset.

any fine-tuning operations, and the other is to evaluate the super-resolution results using practical paired low- and high-resolution DEMs.

3.5.1 Experimental setup

We choose TFASR30 [145] and Pyrenees [1] datasets in our main experiments. Specifically, the resolution of TFASR30 is 30 m, it includes plateau mountain areas and basin areas, which contain various terrain features, including ridges,

rivers, saddles, cliffs, mountains, and so on. It has been split into 25,088 DEM tiles for training and 6,272 DEM tiles for testing, where each tile is in size of 64×64 cells. More details about TFASR30 terrain characteristics can be found in the original paper [145], so we will not repeat them here. The resolution of the Pyrenees is 2 m, it includes 10 regions with a total area of 643 km^2 in mountain regions. To facilitate super-resolution elevations with multi-scale (i.e., $\times 2$, $\times 4$, $\times 6$, and $\times 8$), we divide these large-region DEMs into a set of non-overlapped subtitles, and the size of each tile is 96×96. We randomly select 90% of DEM tiles for training and the rest for testing. This means we obtain 15,206 DEM tiles as training data and 1,690 DEM tiles as testing data. Tyrol dataset also provides high-resolution (2m) DEMs in mountain regions, and it includes 12 regions with a total area of 304 km^2. In our testing experiment, we cropped these large-region DEMs into tiles (sized of 96×96) with no repetition. A total of 7,930 DEM tiles are obtained. Figure 3.4 presents the elevation distribution of the training data and testing data. Following other DEM super-resolution methods setting [145, 144], we generate low-resolution DEMs by downsampling the high-resolution DEMs with the nearest neighbor method.

All experiments are run on the Pytorch platform. For a fair comparison, we train all networks using Adam optimizer with a batch size of 16 for 300 epochs. The learning rate is set as 0.0001 initially and decays by a factor of 0.1 after the 200 epoch. When training our EBCF-CDEM model, we randomly sample a downsampling factor from $\times 1$ to $\times 4$ (or $\times 1$ to $\times 8$) for generating low-resolution DEMs. But for other models, we specify a single downsampling factor within each training procedure since they are not designed for multiple super-resolution DEM tasks. Our CDEM representation model (i.e., the decoder f_θ) is a 5-layer multi-layer perception (MLP) with ReLU activation, and each hidden layer is 256-dimensional. The hyper-parameter L is set to 16 for mapping query positions. L1 loss is used for optimizing our EBCF-CDEM model in the training procedure, which is also mentioned in Section 3.4.

To evaluate super-resolution results of different methods, we chose three terrain indices including elevation value, slope, and aspect in the calculation form of:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\mathrm{O}_i - \mathrm{S}_i)}{N}} \tag{3.5}$$

where O_i and S_i denote the attribute value of terrain feature (slope, aspect, and elevation) of the i-th cell in the original high-resolution DEM and the predicted HR DEM respectively, and N is the total cell number of DEM. Note that lower

Table 3.1: Evaluation results of different methods for the TFASR30 dataset. The suffix "-×2" and "-×4" represent the super-resolution scale used for training. The suffix "-origin" means that we directly use the network weights from the original paper [145]. The best results are marked in bold.

	RMSE-Elevation (m)		RMSE-Slop (°)		RMSE-Aspect (°)	
	×2	×4	×2	×4	×2	×4
Bicubic	0.70 ± 0.93	2.05 ± 2.59	0.61 ± 0.62	1.48 ± 1.50	48.87 ± 20.70	79.77 ± 22.96
TfaSR-origin	-	1.28 ± 1.72	-	1.04 ± 1.08	-	70.70 ± 26.53
TfaSR-×4	-	1.31 ± 1.78	-	1.07 ± 1.11	-	70.85 ± 26.58
TfaSR-×2	0.57 ± 0.80	-	0.44 ± 0.47	-	49.30 ± 23.94	-
EBCF-CDEM (ours)	**0.45 ± 0.68**	**1.19 ± 1.63**	**0.39 ± 0.43**	**0.99 ± 1.05**	**46.16 ± 26.67**	**68.53 ± 28.20**

RMSE values mean to better super-resolution DEM results. In addition, to decrease the inaccuracy of reconstructed DEM samples at the edge, we drop the edge with a width of two cells for each sample in the evaluation procedure.

3.5.2 Multi-scale DEM super-resolution results

In this section, we evaluate the super-resolution performance of different methods, including the Bicubic interpolation method, TfaSR model, and our EBCF-CDEM on test datasets of TFASR30 and Pyrenees. The statistical results tested on the TFASR30 dataset are listed in Table 3.1. It can be observed that our EBCF-CDEM achieves the best performance in terms of RMSE-Elevation, RMSE-Slope, and RMSE-Aspect at super-resolution scales ×2 and ×4. This suggests that our EBCF-CDEM could not only obtain more accurate elevation values but also recover finer terrain structures. Note that the TfaSR model is tested and trained for the same specified super-resolution scale, which means there are two trained TfaSR models for testing with super-resolution scales ×2 and ×4. But we only use one trained EBCF-CDEM for comparison. This proves the superiority of using continuous representation (i.e., our CDEM) for DEM super-resolution tasks.

In addition, to evaluate in a wider range of super-resolution scales, all methods are tested on the Pyrenees dataset. The statistical results are listed in Table 3.2. Note that we test 4 trained TfaSR models separately for different super-resolution scales ×2, ×4, ×6, and ×8. But we don't mark these scales in the table for a concise display. It can be observed that our EBCF-CDEM achieves the best performance across super-resolution scales ×2, ×4, ×6, and ×8. This suggests that our EBCF-CDEM could learn from elevation data with multi-resolution (the used super-resolution scales correspond to 4 m, 8 m, 16 m, and 32 m) simultaneously, which shows the training efficiency of

Table 3.2: Evaluation results of different methods for the Pyrenees dataset. Note that we train the TfaSR model with different super-resolution scales separately. The best results are marked in bold.

		×2	×4	×6	×8
RMSE-Elevation (m)	Bicubic	0.29 ± 0.228	0.59 ± 0.43	1.02 ± 0.65	1.56 ± 0.91
	TfaSR	0.28 ± 0.21	0.60 ± 0.41	0.94 ± 0.64	1.31 ± 0.89
	EBCF-CDEM	**0.27 ± 0.21**	**0.55 ± 0.40**	**0.84 ± 0.61**	**1.18 ± 0.83**
RMSE-Slop (°)	Bicubic	0.28 ± 0.21	0.58 ± 0.42	0.78 ± 0.53	0.92 ± 0.61
	TfaSR	0.26 ± 0.19	0.55 ± 0.40	0.73 ± 0.52	0.85 ± 0.60
	EBCF-CDEM	**0.26 ± 0.18**	**0.53 ± 0.39**	**0.69 ± 0.50**	**0.79 ± 0.58**
RMSE-Aspect (°)	Bicubic	31.21 ± 23.77	44.95 ± 32.06	52.42 ± 35.19	63.91 ± 35.65
	TfaSR	31.50 ± 24.18	45.66 ± 32.67	53.02 ± 36.85	57.75 ± 39.35
	EBCF-CDEM	**30.67 ± 23.69**	**43.59 ± 31.74**	**50.00 ± 35.61**	**54.20 ± 38.15**

EBCF-CDEM for multi-scale super-resolution tasks. In addition, with the super-resolution scale increasing to ×8, our EBCF-CDEM improves results of TfaSR in terms of RMSE-Elevation by 9.92%, RMSE-Slope by 7.06%, and RMSE-Aspect 6.15%. Therefore, it can be concluded that our EBCF-CDEM is effective in improving elevation accuracy and preserving terrain structures during DEM super-resolution.

To better show the accuracy of super-resolution results, we select two DEM samples from the TFASR30 dataset to illustrate. Figure 3.5 presents the high-resolution DEMs and corresponding super-resolution results generated by different methods. It can be seen that all methods could recover major parts of high-resolution DEM from low-resolution DEM. However, the Bicubic interpolation method and the TfaSR model struggle to recover fine details. In contrast, our EBCF-CDEM model generates more similar terrain surfaces to the original high-resolution DEM. Next, considering the slope and aspect visualizations of the same areas in Figure 3.6 and Figure 3.7, our EBCF-CDEM can preserve a more accurate terrain structure with the original DEM.

To reveal more details of super-resolution results obtained for different super-resolution scales, the elevation error maps and corresponding input low-resolution DEMs are presented in Figure 3.8. By comparing results obtained at different scales, we can find that more errors are generated with the scale increasing. This is because more input information is lost in the large super-resolution scale situation. Meanwhile, all methods tend to generate more errors at the large slope areas when recovering from low-resolution DEM. By analyzing Figure 3.8, we can find our EBCF-CDEM generates more accurate super-resolution results than other methods across different super-resolution scales. In summary, these visualization results support that our EBCF-CDEM

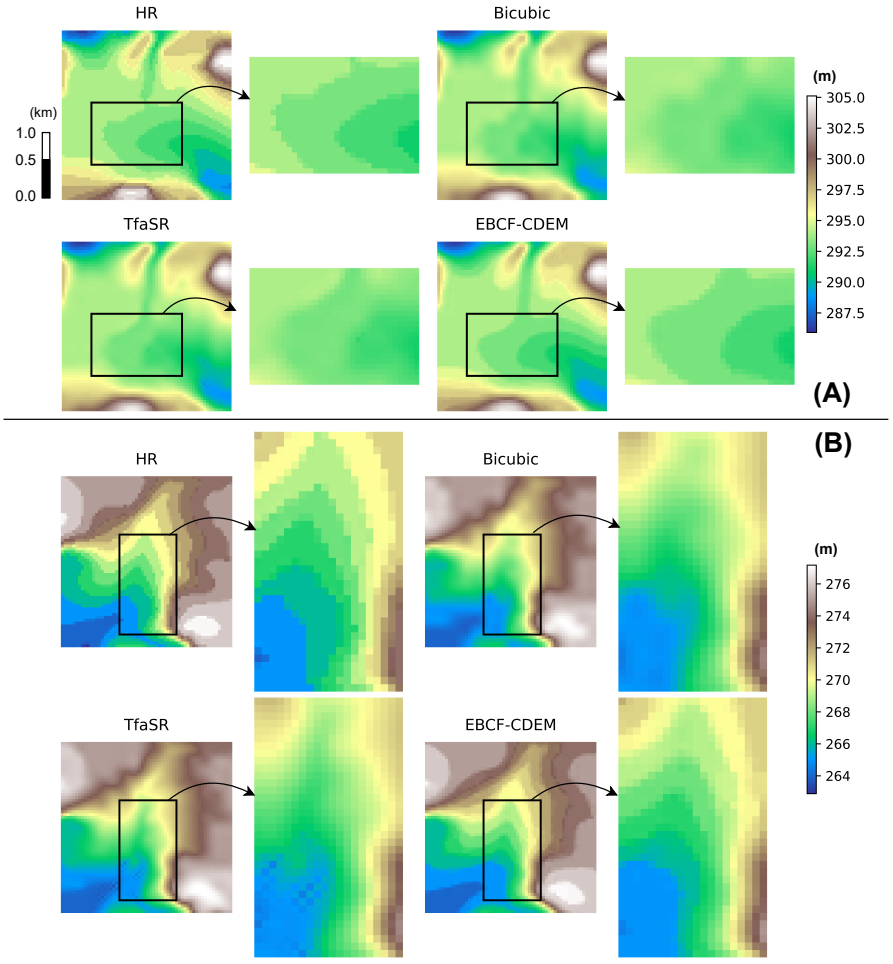

Figure 3.5: The comparison of the super-resolution results by different methods. Samples are from the TFASR30 dataset.

can preserve a more accurate terrain structure and obtain more precise elevation values than other methods.

Another important advantage of our EBCF-CDEM is the continuous representation ability. As shown in Figure 3.9 and Figure 3.10, a more comprehensive

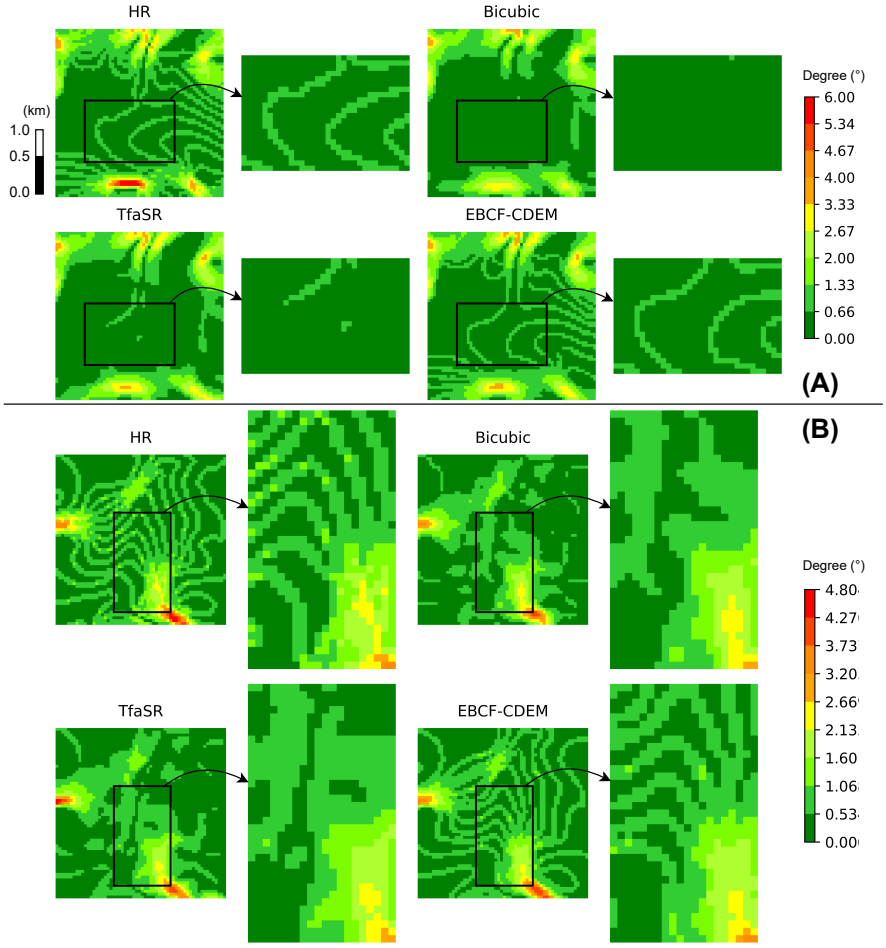

Figure 3.6: The comparison of the slope mapping results by different methods. Samples are from the TFASR30 dataset.

range of super-resolution scales is considered for generating high-resolution DEMs. By comparing the results of TfaSR and EBCF-CDEM across different super-resolution scales, it is easy to find that one trained EBCF-CDEM model generates comparable high-resolution DEMs with four TfaSR models trained independently. Both learning-based models generate more precise elevation val-

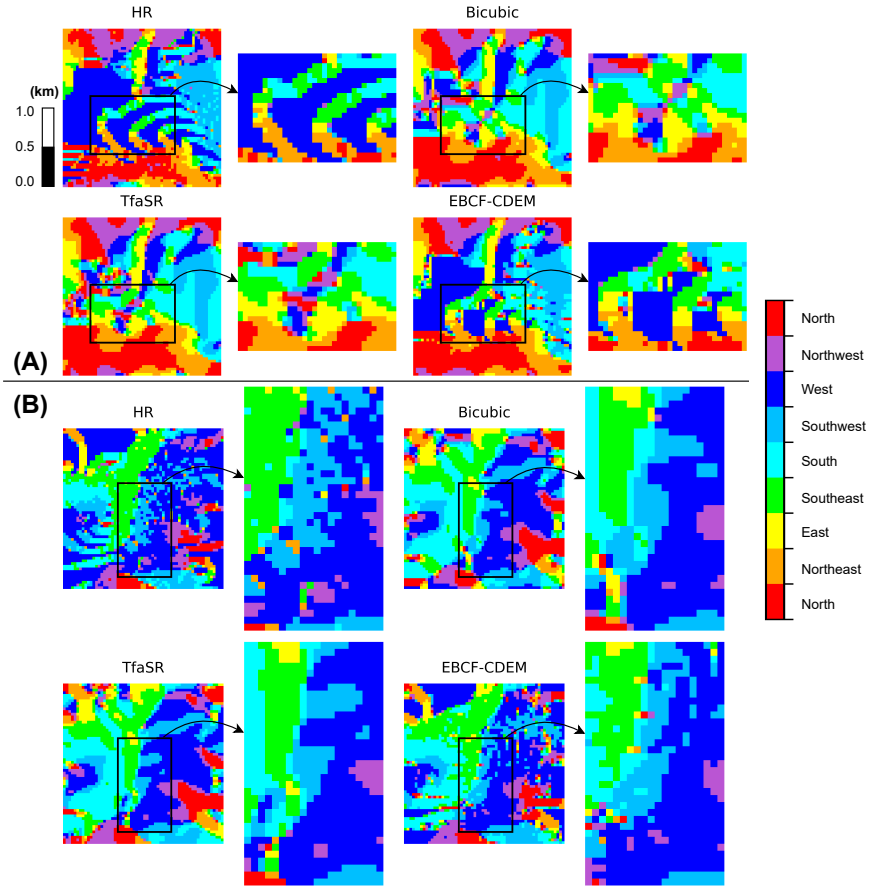

Figure 3.7: The comparison of the aspect mapping results by different methods. Samples are from the TFASR30 dataset.

ues and preserve more structure details than the Bicubic interpolation method. This is consistent with the quantitative results in Table 3.2.

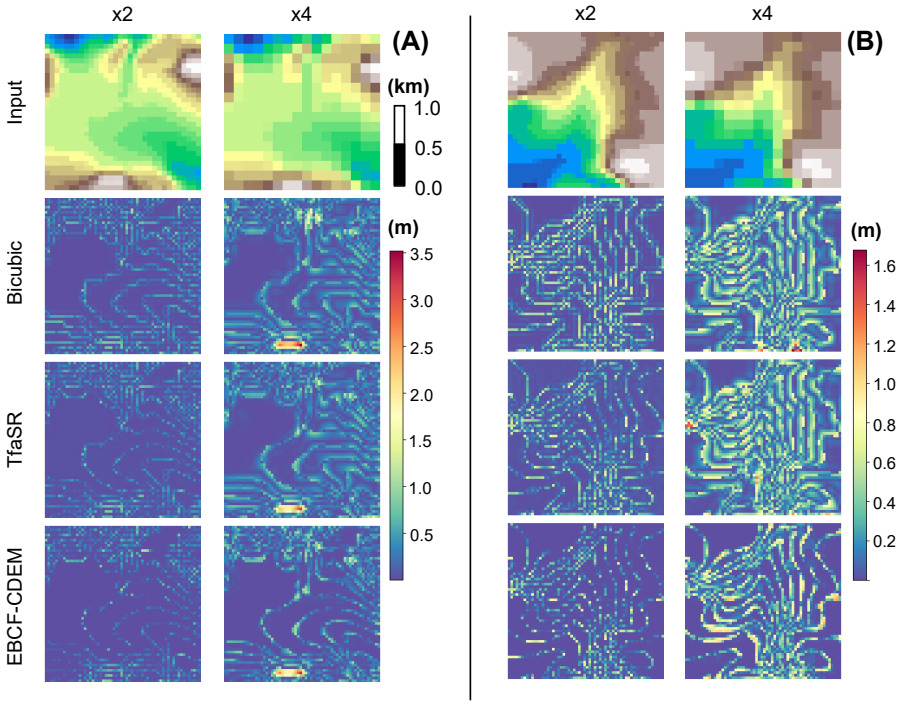

Figure 3.8: The comparison of the error maps generated by different methods across super-resolution scales ×2 and ×4. Samples are from the TFASR30 dataset.

Table 3.3: Evaluation results in terms of different configurations. Elevation-regression means to directly regress the elevation value. Bias-regression means to predict the elevation bias between the query position and the nearest known position. All results are evaluated on TFASR30 dataset.

Elevation prediction	Bias prediction	Positional encoding	RMSE-Elevation (m)		RMSE-Slop (°)		RMSE-Aspect (°)	
			×2	×4	×2	×4	×2	×4
✓			0.48 ± 0.73	1.31 ± 1.75	0.41 ± 0.45	1.08 ± 1.11	48.26 ± 24.80	71.15 ± 26.32
✓		✓	0.46 ± 0.69	1.20 ± 1.64	0.40 ± 0.43	1.00 ± 1.05	47.98 ± 25.02	69.20 ± 26.36
	✓		0.47 ± 0.72	1.29 ± 1.72	0.41 ± 0.44	1.06 ± 1.10	46.36 ± 24.79	69.90 ± 26.89
	✓	✓	0.45 ± 0.68	1.19 ± 1.63	0.39 ± 0.43	0.99 ± 1.05	46.16 ± 26.67	68.53 ± 28.20

3.5.3 The impact of elevation bias prediction and positional encoding

In this section, the impacts of using elevation bias and position encoding are investigated. We design 4 variants of our EBCF-CDEM model to compare. The

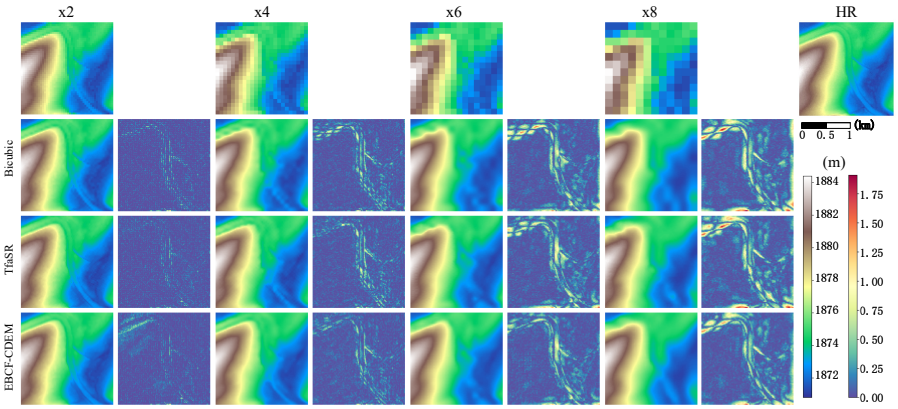

Figure 3.9: The comparison of the super-resolutions and the corresponding error maps across super-resolution scales $\times 2$, $\times 4$, $\times 6$, and $\times 8$. Samples are from the Pyrenees dataset.

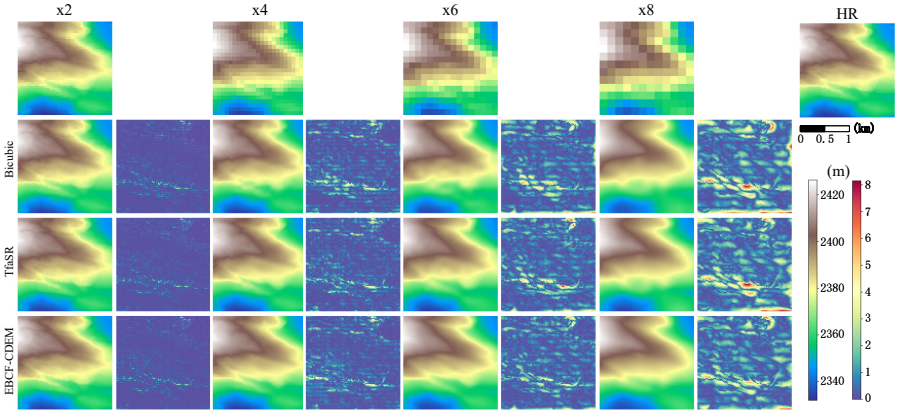

Figure 3.10: The comparison of the super-resolutions and the corresponding error maps across super-resolution scales $\times 2$, $\times 4$, $\times 6$, and $\times 8$. Samples are from the Pyrenees dataset.

comparison results are evaluated on TFASR30 dataset and are listed in Table 3.3. Without using positional encoding, we observe that predicting the elevation

Table 3.4: Evaluation results of different methods trained on the Pyrenees dataset but tested on the Tyrol dataset. The best results are marked in bold.

		×2	×4	×6	×8
RMSE-Elevation (m)	Bicubic	**0.10 ± 0.06**	0.34 ± 0.20	0.82 ± 0.47	1.48 ± 0.88
	TfaSR	0.12 ± 0.06	0.38 ± 0.21	0.72 ± 0.43	1.15 ± 0.78
	EBCF-CDEM	0.11 ± 0.09	**0.29 ± 0.16**	**0.59 ± 0.37**	**0.98 ± 0.68**
RMSE-Slop (°)	Bicubic	0.11 ± 0.06	0.34 ± 0.20	0.58 ± 0.34	0.75 ± 0.47
	TfaSR	0.11 ± 0.06	0.32 ± 0.19	0.50 ± 0.32	0.66 ± 0.45
	EBCF-CDEM	**0.10 ± 0.06**	**0.29 ± 0.17**	**0.45 ± 0.29**	**0.58 ± 0.41**
RMSE-Aspect (°)	Bicubic	19.21 ± 18.83	32.81 ± 28.41	41.46 ± 32.27	55.54 ± 33.67
	TfaSR	19.80 ± 19.21	34.09 ± 29.40	42.30 ± 34.83	47.96 ± 38.05
	EBCF-CDEM	**19.14 ± 18.84**	**31.56 ± 27.94**	**38.65 ± 32.73**	**43.65 ± 35.90**

bias improves the performance of predicting elevation value directly across super-resolution scales ×2 and ×4. This is expected according to our analysis in Section 3.3.2. Then, all results of both models are improved after introducing positional encoding, which proves that accurate reconstruction at high-frequency variation regions is important for our EBCF-CDEM model. Furthermore, by comparing all model variants we observe that the best performance is combining bias prediction and positional encoding. Therefore, it can be concluded that using elevation bias prediction and positional encoding in our EBCF-CDEM model is effective in improving DEM super-resolution performances.

3.5.4 Evaluation across high-resolution DEM datasets

In this section, the generalization ability of our EBCF-CDEM model is investigated on another DEM dataset, i.e., the Tyrol dataset [1], which has a different elevation distribution (shown in Figure 3.4) from the Pyrenees dataset. A total of 7930 DEM tiles (sized of 96×96) are used in this experiment.

To evaluate the generalization ability of different methods, we directly apply the model that was trained on the Pyrenees dataset to the whole Tyrol dataset. Note that we don't optimize any model parameters or execute any fine-tuning operations with the Tyrol dataset. All the experimental results are listed in Table 3.4. It can be observed that our EBCF-CDEM model still outperforms other methods in terms of RMSE-elevation, RMSE-slop, and RMSE-aspect under conditions of super-resolution scales ×4, ×6, and ×8. Moreover, although the Bicubic interpolation method performs better in terms of RMSE-elevation under super-resolution scale ×2, the best results under the other two terrain metrics (i.e., RMSE-slop and RMSE-aspect) are achieved by our EBCF-CDEM model. We argue that this is because of the different

Table 3.5: Evaluation results of different methods to recover the DEM from 30 m to 10 m. The best results are marked in bold.

	RMSE-Elevation (m)	RMSE-Slop (°)	RMSE-Aspect (°)
Bicubic	3.3016 ± 1.4103	1.0651 ± 0.4761	48.7490 ± 29.5496
TfaSR	1.5642 ± 1.3849	0.6998 ± 0.4337	35.8319 ± 26.2609
EBCF-CDEM (ours)	**1.1461 ± 1.0788**	**0.6498 ± 0.4225**	**33.9124 ± 24.9903**

distributions with training and testing data. With the small super-resolution scale ×2, our model struggles to recover the details measured in the difference of elevation values. Nevertheless, the outstanding performance in recovering terrain features proves that our EBCF-CDEM model poses a generalization ability for new data that even have a different distribution.

3.5.5 Evaluation based on practical high-resolution DEMs

In this section, a more challenging experiment is executed on a practical dataset (i,e., TFASR30to10), which has low- and high-resolution DEM pairs. The word "practical" here means that all DEM data are obtained from the real world. Specifically, the high- and low- resolution of DEMs in TFASR30to10 are 30 m and 10 m separately, and the distribution of training data is different from the testing data (more details please refer to Zhang et al. [145]). Note that all training details are the same as the previous setting, and the final results are shown in Table 3.5. Without bells and whistles, our EBCF outperforms previous methods. The results are improved in terms of RMSE-elevation by 26.73%, RMSE-Slop by 7.14%, and RMSE-Aspect by 5.36% when recovering the DEM from 30 m to 10 m in TFASR30to10 dataset. From this view, our EBCF-CDEM is proven to have the potential ability for applying to recover true high-resolution elevation data from DEMs, which further benefits more applications that request high-accuracy terrain analyses.

3.6 Discussion

3.6.1 Advantages of continuous representation

Existing learning-based DEM super-resolution methods require fixed input/output since they rely on the discrete representation, i.e., grid-based format. This limits their applications for generating multi-resolution DEM samples with one trained model. Instead, our CDEM-based model shows superior performance on

multi-scale DEM super-resolution tasks in the above experiments. As shown in Table 3.1 and Table 3.2, only one trained CDEM-based model instead of training with specific scales separately (TfaSR) is used to generate DEM samples with different resolutions. This is one advantage of continuous representation. In addition, our CDEM-based model obtains the most accurate results in terms of RMSE-Elevation, RMSE-Slope, and RMSE-Aspect across all tested scales. Thus, we argue that learning simultaneously with different resolution DEM samples is beneficial for the super-resolution task. And this is also another advantage of continuous representation. Following the experiment from Zhang et al. [145], we evaluate the generalization ability of our CDEM-based model in Section 3.5.4 and Section 3.5.5. It can be observed from Table 3.4 and Table 3.5 that our CDEM-based model significantly improves the accuracy of super-resolution results. This proves that using continuous representation for DEM super-resolution tasks will obtain better generalization performance.

3.6.2 Error analysis of CDEM

In this section, we analyze the errors of our CDEM model for super-resolution from two perspectives: one concerns the relationship between the error value and the query position (i.e., e_q), and the other focuses on the impact of terrain regional complexity on the performance of super-resolution. The detailed contents for each point are presented below:

(1) We calculate the Mean Absolute Error (MAE) of the elevations in the generated super-resolution DEMs along with the absolute distances between the query positions and control points. As depicted in Figure 3.11, it is evident that our CDEM model exhibits decreased accuracy as the distance increases, like the bicubic method. However, our CDEM model outperforms the bicubic method significantly. This finding is consistent with our experimental results presented in Table 3.1 and Table 3.2.

(2) We assess the impact of terrain regional complexity on the accuracy of super-resolution results. Here, terrain regional complexity represents the variation of elevation values within a local region. We posit that a local region exhibiting a substantial standard deviation in elevation values is likely to incur more errors in DEM super-resolution results. Specifically, we randomly sample 6 patches (sized 16×16) from each generated super-resolution DEM across the entire dataset and calculate the standard deviation (STD) for each patch. Then we divide STD values into distinct intervals, and the mean of MAEs is computed for all patches falling within the respective interval. As illustrated in Figure 3.12, both the bicubic method and our CDEM model exhibit an increased error tendency as the complexity of the DEM patch rises. This

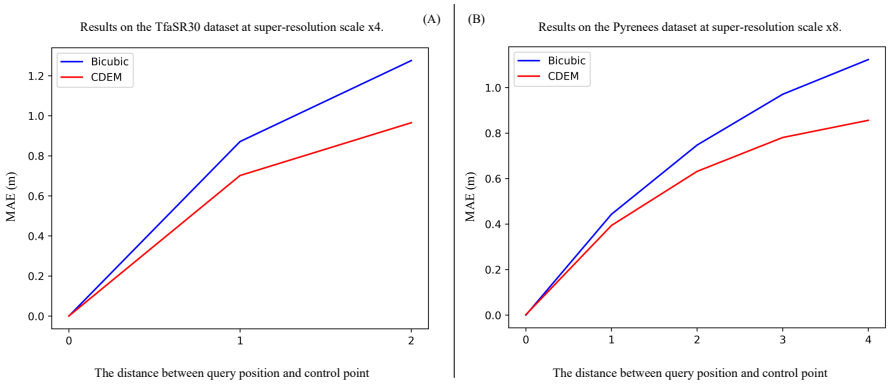

Figure 3.11: The statistic MAE results correspond to the distance between query positions and control points. (A) Calculated on the TfaSR30 dataset at super-resolution scale $\times 4$. (B) Calculated on the Pyrenees dataset at super-resolution scale $\times 8$.

observed pattern persists across both the TfaSR30 and Pyrenees datasets when utilizing our multi-scale super-resolution settings. Additionally, an irregular fluctuation at the 5.6 (STD) position in Figure 3.12 (B) is noted, possibly stemming from an insufficient number of sampled patches within the chosen interval. In conclusion, we assert that the variability in elevation values stands as a pivotal factor influencing the accuracy of our CDEM model in DEM super-resolution tasks.

3.6.3 Comparison with other methods

In this section, we will discuss the main differences between our CDEM-based model and other advanced DEM super-resolution models. Firstly, our model does not need prior or extra information. This information is widely used in other methods, for example, Xu et al. [137] introduced the gradient maps to construct high-resolution DEMs, Argudo et al. [1] introduced high-resolution aerial images to produce highly-detailed DEMs, and Zhang et al. [145] introduced prior near-stream areas and terrain slopes to explicitly optimize their TfaSR model towards preserving local features. Secondly, our model does not rely on complicated optimization. As mentioned in Section 3.4, we only use L1 loss for optimizing model parameters. This is very different from GAN-based methods [30, 144] that require adversarial loss for competitively optimizing

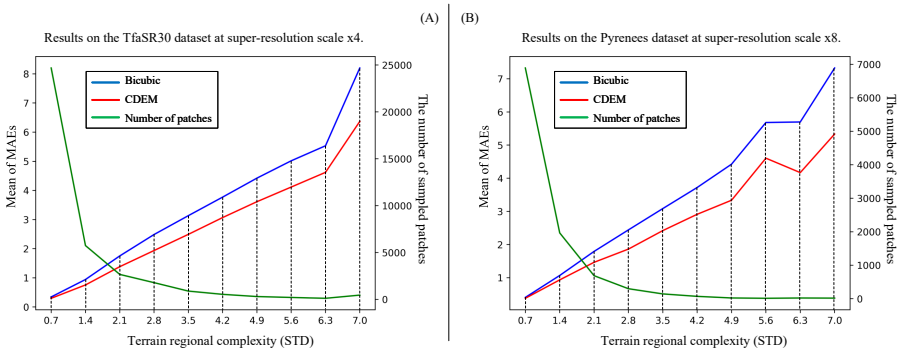

Figure 3.12: The statistic MAE results regarding terrain regional complexity (represented as STD). STD intervals are delineated by dashed lines. (A) Calculated on the TfaSR30 dataset at super-resolution scale $\times 4$. (B) Calculated on the Pyrenees dataset at super-resolution scale $\times 8$.

both the discriminator and generator. In addition, they always need an extra perceptual loss to enhance the high-frequency details. Thirdly, our model is flexible in structure design. Besides the above encoder discussion in Section 3.4, our model is compatible with the structure of some 3D reconstruction models [140, 98]. Specifically, similar to Yao et al. [140], we could introduce a graph neural network to learn latent codes. In this situation, the model input could be an irregular topology (TIN), and the learned latent codes are related to the vertices of the topology. Inspired by Park et al. [98], we even do not need the encoder and directly learn the latent codes with the decoder. In this situation, the model input could be a point cloud. However, the model needs an extra optimization step to obtain latent codes for the test data.

3.6.4 Limitations

In the proposed EBCF-CDEM method, using bias prediction to amplify relatively small variations in elevation values and modify the data distribution to be more concentrated. However, with the resolution of target DEM decreasing, the variation of the bias becomes larger and even similar to the variation of the original elevation value. For example, consider the example in Figure 3.2. If we enlarged the scale between the high-resolution DEM and the low-resolution DEM, the amplified variations would shrink since the max bias value is enlarged. This will compromise the effectiveness of bias as the predicted target.

Another insufficient point is that our CDEM-based representation model lacks global information contained in the terrain. For example, considering spatial autocorrelation in the terrain, if the global Moran's I index shows significant high–high and low-low distribution patterns between elevation values, which indicates that the target region has significant spatial autocorrelation. In this situation, directly using one function to map coordinates to elevation values is reasonable. However, if the target region has a significant spatial negative correlation, using the weighted sum of a set of mapping functions could be more suitable. We will try to introduce this analysis into our CDEM representation to achieve more accurate results.

3.7 Conclusion

This chapter proposes a novel approach, called Continuous DEM (CDEM), for representing digital elevation data in a continuous manner. The CDEM predicts the elevation value of any arbitrary terrain position using a coordinate-based neural network. By introducing a neural encoder, the CDEM is able to represent a wide range of terrain areas in practice. Furthermore, we propose an encoder-based deep learning model, EBCF-CDEM, for raster DEM super-resolution tasks. The EBCF-CDEM combines elevation bias prediction and positional encoding to achieve highly accurate elevation reconstruction of generated high-resolution DEMs. Through extensive experiments, we demonstrate the effectiveness of the proposed CDEM representation on multi-scale DEM super-resolution tasks. We further demonstrate that the proposed model is capable of generating DEMs at different spatial resolutions. Our results show that the EBCF-CDEM outperforms the state-of-the-art TfaSR model and Bicubic interpolation method across different DEM datasets in terms of quantity and quality of results. Ablation experiments have been conducted to validate the effectiveness of the elevation bias prediction and the positional encoding. Finally, we evaluate the generalization ability of the EBCF-CDEM through test experiments on the Tyrol and TFASR30to10 datasets, which demonstrate its potential usage for DEM super-resolution tasks in real-world settings.

Applications utilizing our continuous representation of Digital Elevation Model (CDEM) extend beyond DEM super-resolution. Our primary objective in this chapter is to validate the accuracy, flexibility and generalization of CDEM. Numerous potential works await exploration in the future. We propose two directions for future works: firstly, predicting topographical changes. Our CDEM model can be extended to predict topography changes based on the time and position conditions simultaneously, thus expanding the continuity

of terrain data representation to the time dimension. Secondly, performing geographic analysis in the latent space. By using an encoder to train CDEM on a wide range of terrain areas, we can obtain corresponding latent codes that represent terrain surfaces accurately. These latent codes can be an efficient addition to other learning-based models, such as classification, segmentation and detection.

4 Implicit Neural Representation Model for Camera Relocalization with Learning in Global Multi-Scenes Scenario*

4.1 Introduction

Estimating a camera's position and orientation from a query image taken in a known scene is crucial for computer vision tasks such as autonomous driving [52, 48, 76], simultaneous localization and mapping (SLAM) [114], and augmented reality (AR) [16, 80, 89]. This technique is commonly referred to as visual relocalization.

Existing visual relocalization methods can be categorized depending on whether or not to use a pre-built scenario representation. The pre-built scenario representations, typically obtained using structure-from-motion (SfM) techniques [115, 120] or consisting of posed landmark images, are used to determine image-to-scene [114, 113, 149] or image-to-image [112, 111, 110] correspondences. A PnP solver [39] or relative transform estimator then uses these correspondences to calculate the desired camera pose. However, using pre-built scenario representations has several drawbacks. First, significant storage is needed to save the pre-built scenario representation. Second, using scenario representations poses a risk of exposing private information [149, 20, 121] present in the scene. Third, constructing an accurate scenario representation for precise relocalization is time-consuming [97, 15].

To avoid using pre-built representations, recent neural models perform predictions through a direct feed-forward step. Some researchers [61, 60] use neural models to directly predict the absolute pose from query images. However, this approach often suffers from poor pose accuracy and long training times. To improve visual relocalization accuracy, a scene coordinate (SC) regression method [117] has been proposed. Instead of predicting the absolute camera pose, the SC regression method directly predicts 3D points in the scene's world coordinates (SCs) for 2D pixels in query images. A PnP solver [39] within a RANSAC [37] loop is then used to calculate the final camera pose. Additionally, recent researches [13, 84, 85, 16] show that learning a parametric neural model to regress SC does not require ground truth 3D scene coordinates in both

*This chapter is based on a submission under review in ISPRS Journal of Photogrammetry and Remote Sensing.

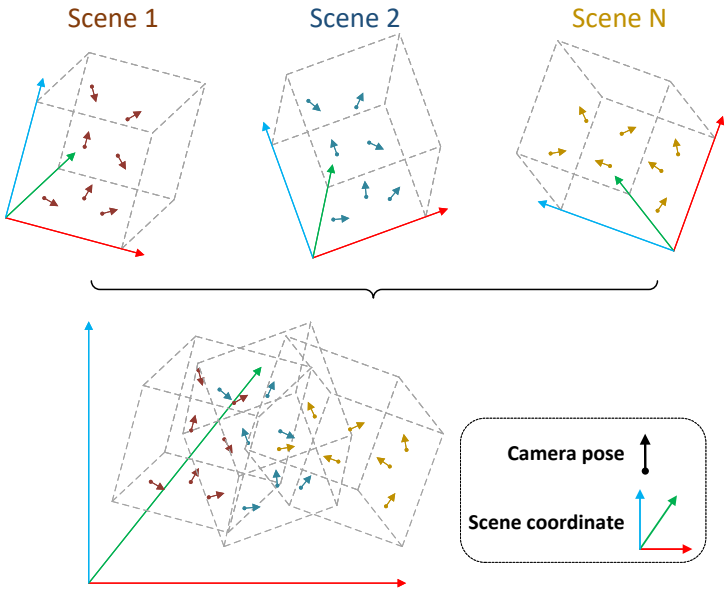

Figure 4.1: In learning a single SC regression model for visual relocalization across multiple scenes, all scenes are represented in a unified global coordinate system. We color-code the scene frame coordinates to illustrate the differences between individual and joint scene representations. When multiple scenes are represented in a global coordinate system, these coordinates may overlap or be widely dispersed. This can cause visual ambiguity; similar image patches from different scenes might correspond to different coordinates, or distinct image patches from overlapping scenes might correspond to the same coordinates. This makes it difficult for a single trained model to accurately predict the desired scene coordinates.

indoor and outdoor scenes. This eliminates the need for prior measurement or preparation of the scene's geometry. In this context, the SC regression model can simultaneously construct a cloud of scene points and determine the camera position from an arbitrary query image. However, these SC regression models are limited to the scenes in which they were trained. Applying these models to multiple scenes requires retraining or using pre-built reference landmarks [139, 123, 108], which is time-consuming.

Inspired by recent absolute pose regression models [6, 116] that use a single neural model for visual relocalization across multiple scenes, we propose an

SC regression model for efficient multi-scene visual relocalization. Instead of separately learning multiple sets of model parameters for different scenes, we use data from all scenes for training. Thus, we represent all scene coordinates within a single global reference coordinate system. Note that the coordinates of different scenes may overlap or be widely distributed. This is illustrated in Figure 4.1.

Our model aims to predict multi-scene coordinates within a single global reference coordinate system. Since multiple scenes may have similar image patches corresponding to different scene coordinates, it is challenging to distinguish which scene the patches belong to and to achieve accurate SC predictions using a single regression model. Additionally, different image patches from overlapping scenes may correspond to the same scene coordinate, further complicating model predictions. To address these issues, we propose encoding scene information as a condition and introducing this condition into our model to resolve visual ambiguities across scenes. A key insight is to assign a specific data embedding for each target scene. These embeddings are input into our model for scene coordinate prediction and are optimized during training. Since the scene representation is implicitly encoded in the model parameters, we name our model SCINR (Scene-Conditional Implicit Neural Regression).

However, because the model parameters implicitly encode 3D scene representation via gradient descent, leading to scene-specific regression results, simply using scene embeddings as an extra input is insufficient for generalizing the regression model across multiple scenes. Furthermore, learned scene embeddings alone are inadequate to simultaneously represent multi-scene information and are inflexible for adjusting model predictions during inference. Inspired by advanced fast parameter generation [46, 38] and meta-learning methods [78, 127], we propose dynamically generating model parameters according to the applied scene to improve generalizability in multi-scene SC regression. Specifically, we propose a scene-conditional regression-adjust module that uses a hypernet conditioned on the scene embeddings to generate module parameters dynamically during inference.

Additionally, we introduce a modulation module and a complement module to enhance SCINR's applicability at the image sample and scene levels. Inspired by neural models representing high-fidelity signals in images, 3D shapes, and videos [83, 63, 35, 119, 66], we use the modulation module [83] to adjust the amplitude, phase, and frequency of the data flow for each input query image. At the scene level, we propose a complement module that utilizes scene embeddings to derive scene-specific coordinate biases. Our idea is motivated by the varying distribution of 3D coordinates across different scenes, which poses a challenge for accurate SC regression by a single neural model. We

combine these predicted biases with the modulation module outputs for the final scene coordinate predictions, aiming to reduce distribution differences between scenes.

Furthermore, our proposed SCINR benefits from a rapid training schedule [16], allowing us to use an aggressive, high learning rate during training. Specifically, SCINR can use training data from all scenes in a single learning schedule. The training pipeline does not propagate pose errors back through a differentiable pose solver [12, 10, 14]; only posed RGB images and camera intrinsic parameters are required. These factors ensure that SCINR achieves lower time consumption in practice. To our knowledge, this is the first work to use an SC regression model with a single set of trained parameters, obtained through a rapid learning process, for accurate visual relocalization across multiple scenes. We evaluated our approach on outdoor and indoor datasets. In summary, SCINR demonstrates superior performance in multi-scene visual relocalization tasks, excelling in both accuracy and efficiency. Additional ablation experiments validate the effectiveness of the hypernet, modulation module, and complement module in improving SCINR's generalization and accuracy.

The contributions of this chapter are as follows:

- We propose SCINR, a scene-conditional implicit neural regression model, enabling multi-scene coordinate regression in a single global reference coordinate for efficient visual relocalization with one parametric neural network.

- We introduce scene embeddings into SCINR for conditional coordinate prediction to reduce the impact of visual ambiguities in multi-scene coordinate regression.

- We propose a scene-conditional regression-adjust module to improve model generalizability across multiple scenes, using a hypernet to dynamically generate module parameters and enable flexible adaptation based on scene embedding.

- We utilize the modulation module and complement module to enhance the model's applicability and accuracy at the image sample and scene levels respectively.

- Our approach employs a fast training schedule, requiring only posed RGB images and camera intrinsic parameters, enabling high-efficiency and lower-cost application in practice.

4.2 Background

4.2.1 Scene coordinate regression-based model for visual relocalization

The use of scene coordinate (SC) regression for camera pose estimation can be traced back to early work [117] that aimed to directly predict 3D points in the scene's world coordinate frame from query 2D image pixels. This approach constructs a dense correspondence between the image and the scene, allowing a robust pose estimator [39] to determine the camera pose using these correspondences. Initially, most SC regression methods relied on regression forests and RGB-D images for prediction [117, 45, 125, 86, 17] but later demonstrated good performance with RGB images [13, 84, 85].

Recently, replacing regression forests with neural networks for SC regression has attracted significant attention. Neural networks encode scene geometric information in their parameters, offering the advantage of privacy preservation. Brachmann et al. [14] proposed a differentiable approximation of RANSAC [37] to train two CNNs for scene coordinate prediction and hypothesis selection. In the following work [10], the authors implemented hypothesis selection with an entropy-controlled soft inlier count and used a single CNN for scene coordinate prediction. They trained the CNN using efficient analytical approximation gradients of PnP solvers. Additionally, this method shows potential for training models solely with RGB images and ground truth poses, without requiring extra scene information such as image depth or a 3D scene model. Further improvements [12] based on [10, 14], including more powerful networks, better initialization, and more efficient loss functions, have led to state-of-the-art pose estimation results. A hierarchical network was later introduced by Li et al. [73], where intermediate layers produce progressively finer discrete location labels to predict scene coordinates in a coarse-to-fine manner. Wang et al. [130] further extended [73] using transformers and angle-based re-projection loss [72]. However, end-to-end training with a differential pose solver [14, 10, 12] can take dozens of hours or even days. To improve the training efficiency, Brachmann et al. [16] proposed a curriculum over a pixel-wise reprojection loss to train a scene-specific regression head, substantially reducing training time to tens of minutes for one scene. Nguyen et al. [94] further integrated a sampling strategy with the ACE [16] to improve pose estimation accuracy.

However, these learning-based SC regression models are often limited to the single scene used for training. To generalize the SC regression model to multiple scenes, Yang et al. [139] proposed to extract a scene representation

from reference images and 3D points and combine this representation with query image features to predict scene coordinates from coarse to fine. This approach allows the network to be applied to different scenes without re-training or adaptation. Tang et al. [123] proposed to construct a cost volume between a query image and reference scene frames (with known 3D coordinates), and a CNN was used to consume this cost volume to predict scene coordinates for the query image. Revaud et al. [108] further introduced the transformer-based encoder-decoder to extract and mix features from the query image and reference scene frames (with sparse correspondences of 2D pixels and 3D coordinates), and these features were used to predict scene coordinates for the query image. These methods require an extra image retrieval step and reference scene frames with 3D geometric information for training and inference. Additionally, Brachmann and Rother [11] proposed using a Mixture of Experts strategy for SC regression across multiple scenes. However, this approach requires training each expert network for the corresponding scene.

Different from these methods, our SCINR aims to estimate camera pose across multiple scenes through direct 3D coordinate regression on a single image. During training, only posed images and camera intrinsic parameters are required. For inference, no additional dataset storage is needed. Additionally, our SCINR can be trained on multiple scenes simultaneously within a very fast learning schedule.

4.2.2 Generalizable implicit neural representations

In recent years, neural networks have been increasingly used to continuously parameterize underlying physical quantities of objects or scenes over space and time, even when these quantities lack an analytic form. Various fields require implicit neural representations (INRs) to be both generalizable and flexible. We will discuss three main types of generalizable INR methods: embedding-based generalization, parameters-based generalization, and signal-based generalization.

Embedding-based generalization: Park et al. [98] proposed to leverage a set of embeddings and Signed Distance Function (SDF)-based INRs to represent 3D shapes. An embedding was assigned to each 3D shape and optimized along with the network parameters during training. Chabra et al. [18] and Jiang et al. [58] followed the work [98], but proposed proposed splitting the 3D space into voxels and assigning an embedding to each voxel. A similar optimization process is applied to both embeddings and network parameters. In the meantime, Genova et al. [41] used an ellipsoid partition of the 3D space for embedding assignment. Further developments Yao et al. [140] and Chen

et al. [22] used graph and Voronoi partitions for space splitting and embedding assignment, respectively, in space-split embedding-based INR for 3D shape representation.

Parameters-based generalization: Ha et al. [46] proposed to use one network, known as a hypernet, to generate weights for another network. The hypernet provides a flexible and efficient way to adapt the neural model based on the input. This concept has promoted the use of INRs in various fields. The hypernet was introduced into INR-based compression by Dupont et al. [35], where hypernet parameters are compressed instead of instance INRs. The hypernet was also used to extract view-dependent global and local information from images within or across scenes [51], generating parameters for NeRF [90] in neural rendering applications. Hu et al. [56] proposed the utilization of neural up-scale filters for image super-resolution representation, where filter weights are dynamically predicted by a neural network based on the super-resolution scale factor. Instead of direct weight prediction, Li et al. [69] used a neural network to produce weights for a linear combination of multiple local image INRs, replacing the up-scale filter. Chen and Wang [23] proposed a transformer-based hypernetwork to directly build the entire set of INR weights using transformers' specialized set-to-set mapping. Furthermore, an effective instance pattern composer for generalizable INRs was proposed by Kim et al. [62], compatible with hypernetworks to predict modulated weights for unseen data during training.

Signal-based generalization: Improving neural model generalization extends beyond learnable parameters. Rahaman et al. [106] indicated that deep networks tend to learn low-frequency functions, inspiring the use of positional encoding for input signals. Tancik et al. [122] proposed a Fourier feature-based mapping for neural models to learn high-frequency functions in low-dimensional problem domains. Mildenhall et al. [90] proposed the positional encoding for neural rendering. Additionally, modulating signal streams within layers has proven to be highly effective. Perez et al. [101] proposed to adaptively influence the output of a neural network by applying an affine transformation to the network's intermediate features. Mehta et al. [83] proposed to use a dual-MLP architecture to encode signals, with one network modulating the amplitude, phase, and frequency of periodic activations in the other. Lee et al. [68] developed the modulation module with coordinates, they suggested to modulate the inter-mediate features using scale and shift parameters extracted from grid-based INRs. They claimed that coordinate-aware modulation maintains the strengths of MLPs while mitigating potential biases, facilitating the rapid learning of high-frequency components.

4.2.3 Visual relocalization across multiple scenes

The concept of extending visual relocalization to learn a single absolute pose regression model for multiple scenes was first introduced by Blanton et al. [6]. They proposed using a classification network to select scene-specific weights from a weights database, after which a regression model would load these weights and use image features to predict the final camera pose. Although they used a shared CNN backbone to extract image features, constructing the weights database still required training multiple models, one for each scene. Their weight selection employed a mixture-of-experts strategy, similar to Brachmann and Rother [11]. Shavit et al. [116] proposed applying Transformers to multi-scene absolute pose regression, using two Transformers with MLP heads to predict camera position and orientation, respectively. As in the previous work [6], a shared CNN backbone was used to first extract image features. The Transformers then processed these features with positional encoding to generate sequences with a latent embedding for each scene. A separate network selected the Transformer outputs based on the scene latent embeddings and passed these selections to respective MLPs for absolute pose regression.

The aforementioned multi-scene visual relocalization methods directly regressed the final pose. To our knowledge, few works have extended the SC regression model on a single image to multi-scene visual relocalization tasks. Furthermore, those methods are time-consuming in training, converse to our approach.

4.3 Method

4.3.1 Camera relocalization in a global multi-scene scenario

Estimating camera pose $T \in \mathrm{SE}(3)$ from a single image I requires finding correspondence pairs $\{(\mathrm{x}_i, \mathrm{y}_i)\}$ between 2D pixel positions $\{\mathrm{x}_i\} \in \mathbb{R}^2$ and 3D scene coordinates $\{\mathrm{y}_i\} \in \mathbb{R}^3$. Here, we represent T as:

$$T = \begin{bmatrix} R & \mathrm{t} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^4 \tag{4.1}$$

where $R \in \mathrm{SO}(3)$ means the rotation matrix and $\mathrm{t} \in \mathbb{R}^3$ means the translation vector. With the known camera calibration matrix $K \in \mathbb{R}^{3\times3}$, we can compute the camera pose using a robust PnP minimal solver [39] within a RANSAC [37] loop followed by a optimization refinement step [12]. Since the mentioned pose

solver is not our focus, we recommend referring to [14, 10, 12] for more details.

Basically, we aim to use a parametric neural network f_θ to construct dense 2D-3D correspondence pairs by directly predicting 3D scene coordinates for 2D-pixel positions in a query image:

$$\mathrm{y}_i = f_\theta(\gamma(\mathrm{x}_i); I), \tag{4.2}$$

where $\gamma(\cdot)$ denotes the positional encoding function, which is defined as:

$$\begin{aligned}\gamma(x) = (&\sin(2^0 x), \sin(2^1 x), ..., \sin(2^{L-1} x),\\ &\cos(2^0 x), \cos(2^1 x), ..., \cos(2^{L-1} x)).\end{aligned} \tag{4.3}$$

In the meanwhile, our model f_θ is designed to be applied to multiple scenes with a single training session. Thus, we represent multiple scenes in a single global coordinate system (as shown in Figure 4.1) and use data from all scenes for training. However, visual ambiguities in image samples from multiple scenes hinder the model's ability to regress scene coordinates. To address visual ambiguities, we introduce a scene embedding set $D \in \mathbb{R}^{n \times d}$ to encode scene information, allowing our model to make predictions based on these embeddings. Each embedding D_n, is assigned to a specific scene. The scene embedding set D is considered part of the model parameters and is optimized throughout the training process. Our model for SC regression across multiple scenes can be represented as follows:

$$\mathrm{y}_i = f_\theta(\gamma(\mathrm{x}_i); I; D). \tag{4.4}$$

Following previous state-of-the-art SC regression methods [94, 16, 34, 12], we use a pre-trained encoder E in f_θ to extract features $\{\mathrm{f}_i\}$ as: $\{\mathrm{f}_i\} = E(I)$. These features are then used to predict 3D scene coordinates. In the following sections, we will describe other main components of f_θ in details, i.e. the regression adjust module f_A with the hypernet H, the modulation module f_M, and the complement module f_C. The overview of our SCINR model is illustrated in Figure 4.2.

4.3.2 Scene-conditional regression-adjust module

Since the SC regression model encodes the scene representation implicitly into the network parameters, it tends to produce scene-specified regression results. To generalize our model across multiple scenes, we propose a scene-conditional regression-adjust (SCRA) module f_A that dynamically generates

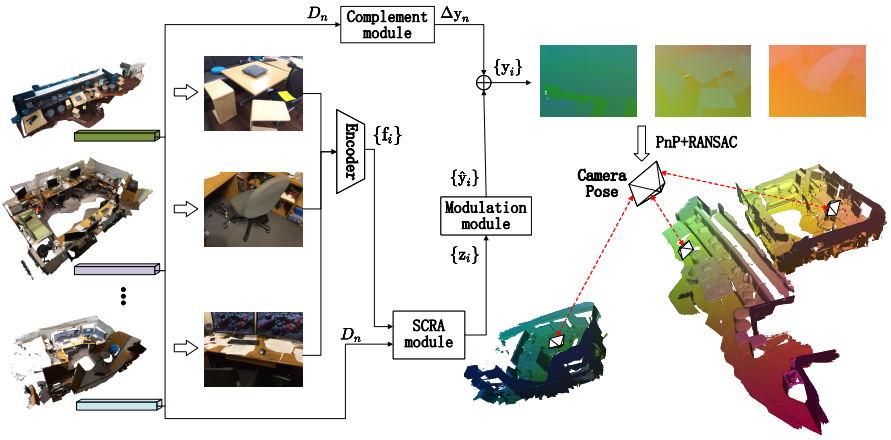

Figure 4.2: Framework Overview: The proposed SCINR comprises a CNN-based encoder, an SCRA module, a modulation module, and a complement module. First, a learned embedding D_n is assigned to each scene, enabling SCINR to predict multi-scene coordinates in a single global reference coordinate. The scene embedding D_n, combined with image features $\{\mathrm{f}_i\}$ extracted by the encoder, is then input to the SCRA module. The SCRA dynamically generates module parameters based on the input scene embedding and processes the input features with these parameters to obtain latent codes $\{\mathrm{z}_i\}$. Next, the modulation module uses $\{\mathrm{z}_i\}$ to adjust the amplitude, phase, and frequency of the data flow when regressing $\{\hat{\mathrm{y}}_i\}$, enhancing model applicability to image samples. Finally, the complement module uses D_n to predict a scene-specific coordinate bias $\Delta \mathrm{y}_n$, which is added to $\{\hat{\mathrm{y}}_i\}$ to obtain the scene coordinates representation. A PnP solver [39] within a RANSAC [37] loop is used to specify the final camera pose. For better illustration, we do not use overlapping scenes.

parameters based on the scene embedding using a hypernet H. Unlike updating network parameters via gradient descent for a single scene, these dynamically generated parameters offer greater flexibility in adjusting model predictions during inference. We show the architecture of the proposed SCRA module in Figure 4.3.

Specifically, we input the embedding D_n as a scene condition to the hypernet H for generating parameters Θ_n. Then the SCRA module f_{A} would load these parameters Θ_n and consume features $\{\mathrm{f}_i\}$ for latent codes $\{\mathrm{z}_i\}$ production.

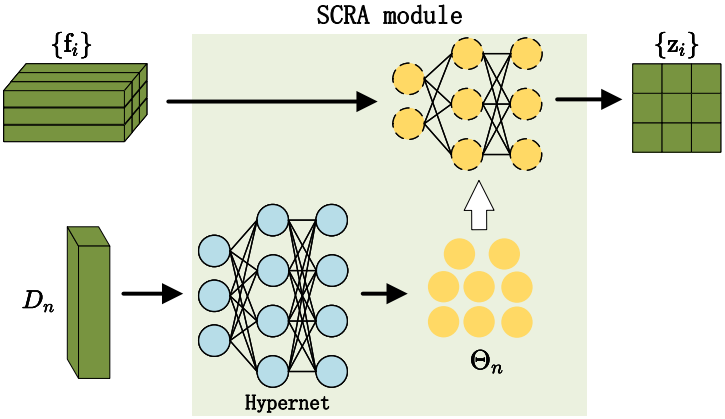

Figure 4.3: The architecture of SCRA module.

This process can be formulated as the following:

$$\begin{cases} \Theta_n = H(D_n) \\ \{\mathrm{z}_i\} = f_{\mathrm{A}}(\{\mathrm{f}_i\}; \Theta_n) \end{cases}. \tag{4.5}$$

In the next section, we will show the usage of latent codes $\{\mathrm{z}_i\}$ in the modulation module f_{M} for better SC predictions on image samples.

4.3.3 Regression applicability enhanced on the image sample and scene levels

To improve the regression model applicability on image samples, we propose to use the modulation module f_{M} to adjust the amplitude, phase and frequency of data flow in mapping x_i to $\hat{\mathrm{y}}_i$. Bring the idea from coordinate-based neural representation models [119, 90, 135, 141], i.e., mapping coordinates to a higher-dimensional space, or increasing their complexity using positional functions before feeding them into the network, results in improved data fitting, particularly for data containing high-frequency variations [106, 122]. As shown in the upper part of Figure 4.4, we input the positional encoding $\gamma(\mathrm{x}_i)$ combined with latent codes $\{\mathrm{z}_i\}$ to f_{M} as:

$$\hat{\mathrm{y}}_i = f_{\mathrm{M}}(\gamma(\mathrm{x}_i), \mathrm{z}_i), \tag{4.6}$$

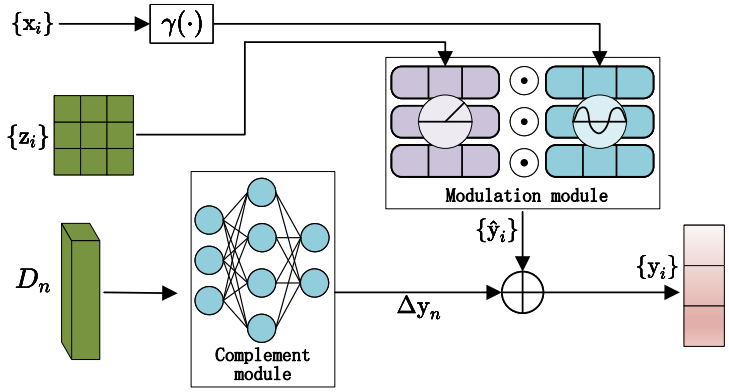

Figure 4.4: The architecture of modulation and complement modules.

where the modulation module f_{M} is composed of a synthesis branch $s(\cdot)$ and a modulator $m(\cdot)$. $s(\cdot)$ maps from positional encoding $\gamma(\mathrm{x}_i)$ to target signals, $m(\cdot)$ consumes latent codes $\{\mathrm{z}_i\}$ to modulate the amplitude, phase and frequency of $s(\cdot)$ output at each layer l. This process can be defined as:

$$\begin{cases} \alpha^0 = m^0(\mathrm{z}_i) \\ h^0 = \alpha^0 \odot s^0(\gamma(\mathrm{x}_i)) \\ \alpha^{l+1} = m^l(\alpha^l, \mathrm{z}_i) \\ h^{l+1} = \alpha^{l+1} \odot s^l(h^l) \end{cases}. \tag{4.7}$$

Different scenes have distinct 3D coordinates distribution, which is a challenge for neural networks in accurately predicting scene coordinates. An example of SC t-SNE results from the 12Scenes dataset is shown in Figure 4.5. To mitigate these differences, we propose a complement module f_{C} to predict a scene-specified coordinate bias $\Delta\mathrm{y}_n$ from a given scene embedding D_n. The architecture of the complement module is shown in the bottom left part of Figure 4.4. Formally, this can be expressed as:

$$\Delta\mathrm{y}_n = f_{\mathrm{C}}(D_n). \tag{4.8}$$

Finally, our final scene coordinate results $\{\mathrm{y}_i\}$ are obtained as: $\mathrm{y}_i = \hat{\mathrm{y}}_i + \Delta\mathrm{y}_n$.

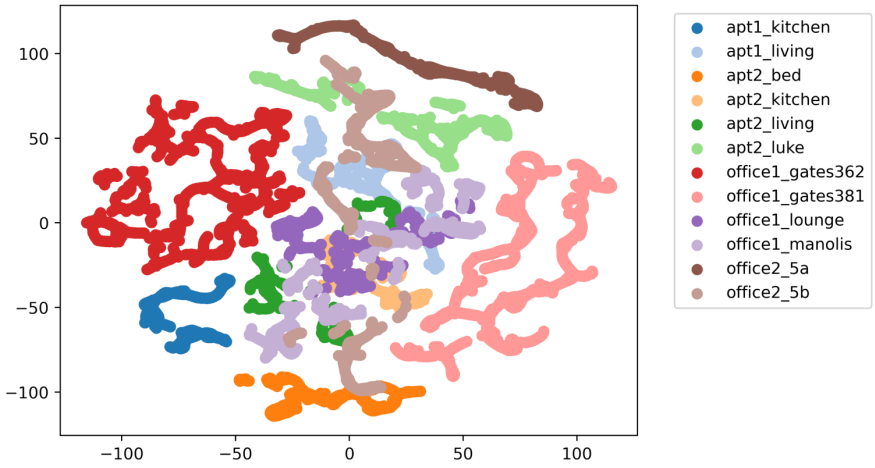

Figure 4.5: The t-SNE results of scene coordinates calculated from the 12Scenes dataset. We can observe a distinct distribution gap between scenes.

4.4 Experiments

4.4.1 Experimental setup

Our implementation is based on the ACE learning pipeline [16], which employs an efficient curriculum training strategy. We adhere to their loss function, based on pixel-wise reprojection loss, and their dynamic rescaling method according to training progress. For comprehensive details, we refer readers to the original paper [16]. Although our experiments involve multiple scenes, we construct a training buffer of 8M, consistent with ACE's original setting, with each sample instance randomly sampled across different scenes. Additionally, we optimize model parameters 16 times over the entire training buffer with a batch size of 5120, and the learning rate of the AdamW optimizer ranges between 5×10^{-4} and 5×10^{-3} with a one-cycle schedule.

All experiments are conducted on the PyTorch platform. To ensure fair comparison and efficient implementation, we utilize the pre-trained ACE encoder to extract image features. To calculate the final camera pose, we use the robust DSAC* [12] pose estimator with the setting of 64 RANSAC hypotheses and a 10-pixel inlier threshold.

We evaluate our SCINR on three standard datasets for camera pose estima-

tion, covering both indoor and outdoor scenarios. Note that we only require the camera calibration matrix, ground-truth poses, and images for training. Specifically, we use the *Cambridge Landmarks* [61] dataset to validate our approach in outdoor scenarios. As the STREET scene in Cambridge Landmarks cannot provide reliable ground truth for interpretation [10], we follow previous works [93, 34, 12] and choose the other five scenes for our experiments. Each scene represents a site at Cambridge University, UK, spanning several hundred to a few thousand square meters. The original authors split the training and test sets based on separate walking trajectories rather than a single trajectory, making the pose estimation task more challenging. Thus, we use this training and test set split in our experiments.

The *7Scenes* [117] and *12Scenes* [126] datasets provide multiple sequences of tracked camera frames in small-scale indoor rooms. The 7Scenes dataset presents several challenges, including motion blur, illumination changes, textureless surfaces, repeated structures (e.g., in the Stairs dataset), reflections (e.g., in the Redkitchen dataset), and sensor noise. The 12Scenes dataset is similar to the 7Scenes but covers slightly larger indoor environments and contains higher-quality images. We follow the ACE configurations for splitting the training and test sets.

Table 4.1: Evaluation results on the Cambridge Landmarks dataset. We report the median errors of position and degree in (m / °). "$*$" means that training separately on different scenes. "-E64" means that we train the model 64 times over the buffer. The results of MS-Transformer* and HyperPose* are from Ferens and Keller [36]. The best results are marked in bold.

	Great Court	King's College	Old Hospital	Shop Facade	St Mary's Church	Average	Training Time
PoseNet(PN)* [61]	-	1.92/5.4	2.31/5.4	1.46/8.1	2.65/8.5	2.08/6.8	Hours
σ^2. PN* [60]	7.00/3.7	0.99/1.1	2.17/2.9	1.05/4.0	1.49/3.4	2.54/3.02	Hours
geo. PN* [60]	6.83/3.5	0.88/1.0	3.20/3.3	0.88/3.8	1.57/3.3	2.67/2.98	Hours
PoseGAN* [79]	-	1.22/4.4	1.52/4.9	0.88/4.8	1.82/5.8	1.36/4.98	Hours
FeatLoc++Au* [5]	-	1.30/3.8	2.05/6.1	0.91/7.5	2.99/10.4	1.81/6.95	Hours
MSPN [6]	-	1.73/3.7	2.55/4.1	2.92/7.5	2.67/6.2	2.47/5.38	Hours
MS-Transformer [116]	-	0.83/1.5	1.81/2.4	0.86/3.1	1.62/4.0	1.28/2.75	Hours
MS-Transformer*	-	0.72/2.6	2.07/3.2	0.68/3.7	1.10/5.3	1.14/3.68	Hours
HyperPose* [36]	-	0.56/2.4	1.14/2.9	**0.54**/3.4	0.98/4.9	0.87/3.43	Hours
ACE	11.86/8.5	0.43/0.6	0.58/1.0	19.07/54.1	16.14/45.5	9.62/21.94	5 Minutes
Ours	5.41/4.8	0.40/0.6	0.53/0.9	2.59/8.2	4.41/10.2	2.67/4.94	13 Minutes
ACE-E64	1.82/1.1	0.37/0.6	0.52/0.9	12.67/41.4	1.37/4.5	3.35/9.70	20 Minutes
Ours-E64	**0.98/0.5**	**0.36/0.5**	**0.40/0.6**	0.59/**2.3**	**0.89/2.7**	**0.64/1.32**	50 Minutes

Table 4.2: Evaluation results on the 7Scenes dataset. We report the median errors of position and degree in (m / °). "$*$" means that training separately on different scenes. The results of MS-Transformer* and HyperPose* are from Ferens and Keller [36]. The best results are marked in bold.

	Chess	Fire	Heads	Office	Pumpkin	RedKitchen	Stairs	Average	Training Time
PoseNet(PN)* [61]	0.32/8.12	0.47/14.4	0.29/12.0	0.48/7.68	0.47/8.42	0.59/8.64	0.47/13.8	0.44/10.4	Hours
σ^2. PN* [60]	0.14/4.50	0.27/11.8	0.18/12.1	0.20/5.77	0.25/4.82	0.24/5.52	0.37/10.6	0.24/7.87	Hours
geo. PN* [60]	0.13/4.48	0.27/11.3	0.17/13.0	0.19/5.55	0.26/4.75	0.23/5.35	0.35/12.4	0.23/8.12	Hours
PoseGAN* [79]	0.09/4.58	0.24/9.46	0.17/13.38	0.19/8.80	0.16/6.28	0.26/8.23	0.28/10.14	0.20/8.70	Hours
FeatLoc++Au* [5]	0.07/3.66	0.17/5.95	0.10/7.57	0.16/5.20	0.11/3.86	0.20/6.43	0.16/8.57	0.14/5.89	Hours
StructureLoc* [71]	0.10/8.44	0.26/11.7	0.16/13.3	0.16/6.63	0.16/5.05	0.20/6.32	0.27/9.65	0.19/8.72	Hours
MSPN [6]	0.09/4.76	0.29/10.5	0.16/13.1	0.16/6.8	0.19/5.5	0.21/6.61	0.31/11.63	0.20/8.41	Hours
MS-Transformer [116]	0.11/4.66	0.24/9.60	0.14/12.2	0.17/5.66	0.18/4.44	0.17/5.94	0.26/8.45	0.18/7.28	Hours
MS-Transformer*	0.10/5.56	0.23/11.0	0.15/12.8	0.17/6.56	0.18/5.32	0.17/6.30	0.26/11.3	0.18/8.41	Hours
HyperPose* [36]	0.08/6.29	0.22/11.2	0.11/12.7	0.17/7.53	0.16/6.66	0.17/8.48	0.26/10.8	0.17/9.08	Hours
ACE	**0.02**/0.8	**0.03**/1.1	2.15/73.9	0.04/1.1	0.05/1.5	0.05/1.6	**0.14/2.8**	0.35/11.8	5 Minutes
Ours	**0.02/0.7**	**0.03/1.0**	**0.03/1.7**	**0.03/0.9**	**0.04/1.2**	**0.04/1.2**	0.31/6.6	**0.07/1.90**	14 Minutes

4.4.2 Main results

This section evaluates the performance of the proposed SCINR on multi-scene visual relocalization tasks. Since we requires the training data do not include information about 3D scene structures and geometries, limiting the applicability of many methods, we primarily compare with other advanced direct regression-based models, namely StructureLoc [71], FeatLoc++Au [5], PoseGAN [79], PoseNet [61, 60], MSPN [6], MS-Transformer [116], and HyperPose [36]. Note that only MS-Transformer and MSPN are trained for relocalization across multiple scenes, while the others are trained separately for different scenes. We focus on efficiently learning an SC regression model for multi-scene relocalization. Thus, we adopt the rapid and aggressive training strategy of ACE. This strategy may not be suitable for other methods, particularly when using all training data from multiple scenes simultaneously. Notably, training our SCINR requires only about 13 minutes. The following results highlight the advantages of our SCINR in multi-scene visual relocalization.

The statistical results for the Cambridge Landmarks dataset are presented in the Table 4.1. It can be observed that our SCINR achieves the best performance on average position and degree median errors across the dataset. Except for the position median error evaluated on the Shop Facade scene, our SCINR achieves the most accurate relocalization results in terms of degree and position median errors for every scene. This suggests that our SCINR performs competitively not only with models trained for multi-scene but also with those trained separately for individual scenes. Conversely, obvious errors are observed in the ACE results for the Great Court, Shop Facade, and St Mary's Church scenes. These error gaps are difficult to reduce, even with extended training time, especially for the Shop Facade scene. In comparison, our SCINR achieves more stable performance than ACE by using all scene data in a single training session. Furthermore, we compare the training times of different models. Our SCINR achieves more accurate results in less time compared to other direct regression-based models. Comparing "Ours" and "ACE-E64" results, our SCINR achieves more accurate relocalization results in less time. This suggests that our SCINR can accurately estimate the camera pose across multiple outdoor scenes with a single efficiently trained model.

Similarly, we report the evaluation results on the 7Scenes and 12Scenes dataset in Table 4.2 and Table 4.3 respectively. We can observe that our SCINR outperforms the other models on average values of position and degree median errors across the 7Scenes and 12Scenes datasets. Except for the Stairs, office1_gates362, and office2_5a scenes, our SCINR achieves the best relocalization results in every indoor scene, demonstrating the lowest position

Table 4.3: Evaluation results on the 12Scenes dataset. We report the median errors of position and degree in (m / °). "$*$" means that training separately on different scenes. The PoseNet and FeatLoc results are from [5]. The best results are marked in bold.

Scenes	PoseNet*	FeatLoc++Au*	StructureLoc* [71]	ACE	Ours
apt1_kitchen	0.62/6.75	0.32/5.19	0.11/6.61	10.10/83.2	**0.14/4.7**
apt1_living	0.61/6.03	0.26/3.89	0.12/7.81	0.07/2.4	**0.02/1.0**
apt2_bed	0.65/5.66	0.37/5.39	0.20/10.4	12.20/98.3	**0.16/5.0**
apt2_kitchen	1.24/6.84	0.73/6.37	0.11/7.88	2.18/32.5	**0.03/1.4**
apt2_living	0.78/7.61	0.40/5.71	0.14/8.08	11.14/101.7	**0.07/2.2**
apt2_luke	0.66/7.10	0.33/4.85	0.20/8.31	5.31/44.9	**0.12/4.0**
office1_gates362	1.05/5.67	0.52/5.22	0.15/8.97	**0.02/0.6**	**0.02**/0.7
office1_gates381	0.70/8.23	0.42/6.23	0.16/9.58	3.24/35.3	**0.12/4.4**
office1_lounge	0.77/7.35	0.39/4.50	0.19/9.44	3.70/43.7	**0.04/1.4**
office1_manolis	0.64/6.56	0.30/4.67	0.18/7.93	8.08/67.7	**0.08/2.4**
office2_5a	0.59/5.65	0.31/4.32	**0.15**/6.40	5.55/47.6	0.20/**4.3**
office2_5b	0.52/4.32	0.23/4.14	0.26/8.21	2.70/24.2	**0.17/3.9**
Average	0.74/6.48	0.38/5.04	0.16/8.3	5.36/48.51	**0.10/2.95**

and degree median errors. Thanks to the fast training procedure, the proposed SCINR can achieve such high relocalization performance on multiple indoor scenes within 20 minutes instead of hours.

Notably, while ACE also learns quickly, it exhibits distinct prediction errors in the Heads scene and most scenes in the 12Scenes dataset. In comparison, our SCINR performs more robustly, avoiding similar distinct relocalization errors. To reveal more detail on the relocalization results of SCINR and ACE, we plot the camera trajectories for several test sequences from the 12Scenes dataset in Figure 4.6 and the cumulative distributions of pose error in Figure 4.7. It can be seen that our SCINR significantly outperforms ACE not only in prediction stability but also in terms of accuracy percentage.

4.4.3 Ablation study

This section examines the impacts of SCINR components, including the SCRA module, modulation module, positional encoding, and complement module. For a fair comparison, we respectively replace these components with the MLP, which poses a similar number of parameters as the replaced network. The comparison results are evaluated on Cambridge, 7Scenes, and 12Scenes datasets and are listed in Table 4.4.

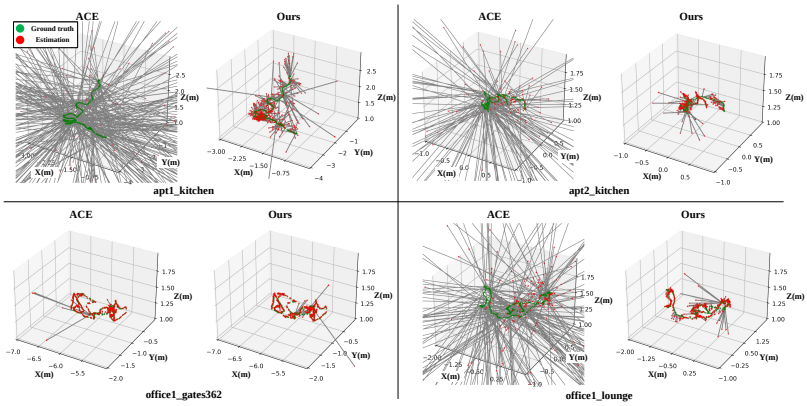

Figure 4.6: Visual relocalization results on 12Scenes dataset. For each subfigure, the 3D plot shows the camera trajectory (green points represent ground truth, red points represent estimations, and gray lines represent correspondences).

Table 4.4: Evaluation results in terms of different configurations. We report the median errors of position and degree in (m / °).

SCRA module	Modulation module	Positional encoding	Complement module	12Scenes	7Scenes	Cambridge
✗				7.18/62.2	0.49/11.69	16.67/37.48
	✗			0.10/3.03	0.08/1.92	2.94/5.60
		✗		0.11/3.12	0.09/2.06	2.71/5.36
			✗	0.27/4.00	0.08/2.01	3.09/5.72
✓	✓	✓	✓	0.10/2.95	0.07/1.90	2.67/4.94

(a)The impact of using scene information:

The SCRA module plays an important role in pose estimation across multiple scenes. Without the SCRA, our SCINR shows a significant performance drop across all datasets. This validates our approach of adapting regression model parameters based on scene information for multi-scene visual relocalization. Additionally, the proposed SCRA module effectively implements conditional parameter generation.

Comparing our complete SCINR model with other variants in the Table 4.4

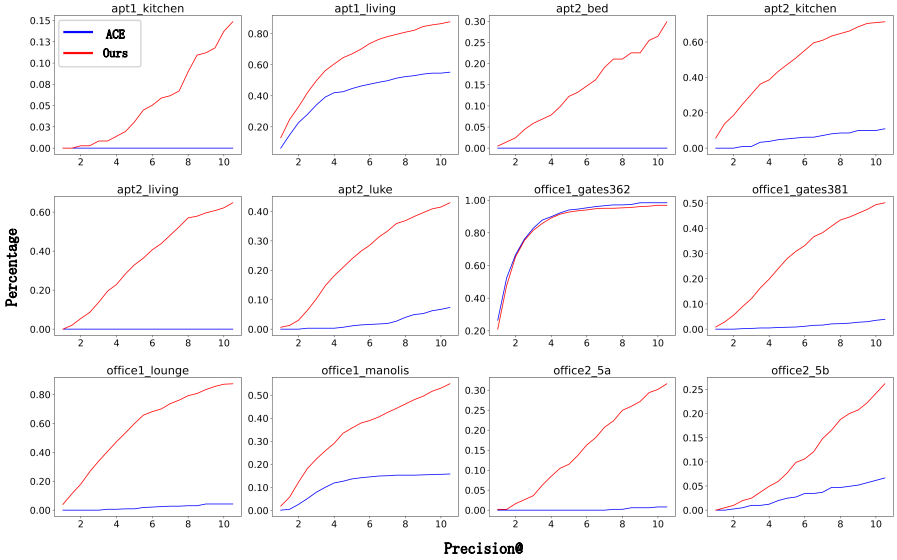

Figure 4.7: Cumulative distributions (Percentage) of pose error (Precision@ means the maximum of position and rotation errors) on 12Scenes dataset.

reveals that the second significant performance improvement comes from the complement module. This is consistent with our analysis of SC distributions across multiple scenes 4.3.3. Like the SCRA module, the proposed complement module effectively predicts scene-specific coordinate bias based on the given scene information.

In our SCINR model, this scene information, termed scene embedding, is input into the SCRA and complement modules as a condition, leading to the above improvements. Thus, it can be concluded that conditional prediction is key to applying visual relocalization across multiple scenes.

(b)The impact of generalizing model representation:

In this section, we investigate the impacts of positional encoding and the modulation module on enhancing the representation ability of SCINR. As shown in Table 4.4, a slight performance drop is observed without positional encoding or the modulation module. In other words, both components further improve scene coordinate regression. Specifically, for indoor scenes (12Scenes and 7Scenes datasets), our SCINR benefits more from positional encoding, while

for outdoor scenes (Cambridge dataset), it benefits more from the modulation module. In summary, using positional encoding and the modulation module enhances the generalization ability of our model in SC regression, leading to more accurate visual relocalization across multiple scenes.

4.5 Discussion

Without storing extra information (e.g., reference scene frame for image retrieval, model parameters, etc.), existing SC regression models for visual relocalization are limited to a single scene. This limitation hinders their application in multi-scene scenarios requiring rapid deployment. In this chapter, we extends the application of SC regression to multiple scenes within a global coordinate system, aiming to address multi-scene visual relocalization with one single trained neural model. With the proposed SCINR, we can simultaneously leverage samples from all scenes to train the model using an aggressive and efficient learning schema. As shown in the Tables 4.1, 4.2, and 4.3, our SCINR achieves the best visual relocalization results across multiple scenes and requires only a dozen minutes for training. This highlights the advantage of learning the SC regression model in a global coordinate for multi-scene visual relocalization. Additionally, our SCINR requires only camera intrinsic parameters, poses, and images during training. This significantly reduces the cost of collecting training data, eliminating the need for sensors to measure scene geometry, ad-hoc data processing (e.g., calibration, filtering), and explicit scene representation construction.

However, the relocalization results of our SCINR are not as precise as those of separately trained ACE models. This is expected, as integrating multiple scenes into a global coordinate complicates SC regression for a single neural model. We believe future research will enhance multi-scene visual relocalization to reduce this accuracy gap. Thus, we consider two directions for future work: first, addressing the imbalance in the number of training samples from different scenes by designing a new loss function or employing a data re-balancing strategy may improve model effectiveness. Second, the calculated loss value of the query image may vary significantly depending on the sampled scene. Training the model with samples from multiple scenes may cause the learning process to focus on scenes with higher loss values. Ensuring the model learns equally from different scene samples may lead to more precise results.

Another limitation is that using reprojection loss makes it difficult to train a valid SC regression model for scenes with a bird's-eye view or a large distance range from the camera to scene surfaces. This presents a more challenging

problem beyond the scope of this chapter.

4.6 Conclusion

This chapter proposes a novel neural model, SCINR, for visual relocalization across multiple scenes. SCINR consumes the query image to predict scene coordinates, which are then used to calculate the desired camera pose using a robust PnP solver. The proposed SCRA module enables SCINR to dynamically generate network parameters according to the scene information. The modulated module and positional encoding enhance SCINR's applicability at image samples for scene coordinate regression. The proposed complement module, which predicts scene-specific coordinate bias, allows SCINR to achieve more accurate predictions across different scenes. Furthermore, SCINR can learn from data of different scenes simultaneously within one model, enabling accurate multi-scene relocalization results with a rapid training schedule. Notably, the training data includes only posed images and camera intrinsic parameters. Extensive experiments on indoor and outdoor datasets demonstrate SCINR's effectiveness and efficiency for visual relocalization across multiple scenes. Our results show that SCINR achieves state-of-the-art multi-scene visual relocalization accuracy in terms of median position and degree errors. Additionally, training SCINR on multiple scenes requires only a fraction of the time of existing methods while outperforming many single-scene trained models. Ablation experiments validate the effectiveness of the SCRA module, the modulated module, the positional encoding, and the complement module.

5 Conclusions and Future Work

5.1 Conclusions

In this thesis, we developed three types of implicit neural representation (INR) models in tasks of 3D shape representation, terrain elevation modeling, multi-scale DEM super-resolution, and multi-scene visual relocalization. Compared to conventional discrete 3D data formats used under different applying backgrounds, we proposed a unified continuous representation based on mapping coordinates to target signals. In experiments, our methods respectively have been proven to improve the accuracy of 3D shape reconstruction with fewer model parameters, maintain more terrain structures and features in DEM super-resolution results, and achieve more efficient implementation in visual relocalization. For more practical problems existing in the applications mentioned above, our corresponding solutions in this thesis are below:

- **Chapter 2: 3D Shapes Local Geometry Codes Learning With SDF.** Previous research attempted to learn one global signed distance function (SDF) with the INR model for representing the whole 3D shape, resulting in a time-consuming training procedure with numerous model parameters and coarse reconstructed details. Some recent methods instead to learn a set of local SDFs for 3D shape representation to alleviate this problem. However, these methods rely on overlapped region splitting, leading to repetitive learning in the overlapped regions and reducing the efficiency. In this chapter, we proposed to use a decoder with latent codes to learn local SDFs of the 3D shape. To avoid repetitive learning, we introduced a graph neural network (GNN) to exchange messages among these local SDFs, which are learned in no-overlapped regions. To enhance these local SDFs learn from neighbors, we proposed a geometric similarity loss function based on the graph structure. Experimental results verified our method considerably outperforms the baseline DeepSDF [98] both in accuracy and model size.

- **Chapter 3: A Continuous Digital Elevation Representation Model for DEM Super-resolution.** Existing terrain elevation models

were limited to discrete representations, and suffered from accuracy gaps caused by discretization and format conversion. In this chapter, we proposed a continuous digital elevation model (CDEM) to achieve elevation value prediction at an arbitrary query position by a coordinate-based parametric neural network. An encoder-decoder structure based on our CDEM representation was developed to achieve multi-scale DEM super-resolution tasks, breaking the constraints of a fixed size of input/output pair with a specified scale in existing DEM super-resolution methods. Furthermore, we introduced elevation bias prediction and positional encoding to improve model accuracy for areas with small elevation changes and high-frequency elevation variations. In our experiments on three DEM datasets, we demonstrated that our model is effective in obtaining more accurate elevation values. The different types of visualized feature maps of DEM super-resolution results proved that our model can preserve more terrain structures and features. Extensive experiments on practical datasets further validated the generalizability of our model.

- **Chapter 4: Multi-scene Visual Relocalization.** Recently advanced methods predicted scene coordinates from a query image to realize visual relocalization in a known scene. Extending these methods to multiple scenes typically requires retraining (or adapting) model parameters, or using pre-built reference landmarks, which is a time-consuming process. This chapter proposed a scene-conditional implicit neural regression (SCINR) model that can achieve accurate visual relocalization results across multiple scenes without the requirements of the aforementioned process. Instead, we encode scene information in latent embeddings and designed a scene-conditional regression-adjust (SCRA) module to dynamically generate parameters during inference according to the scene information. We introduced the modulation module and complement module to enhance the model applicability at the image sample and scene levels respectively. Furthermore, SCINR can learn from data from different scenes simultaneously with one model and using a rapid learning schedule. Notably, the training data includes only posed images and camera intrinsic parameters. Extensive experiments on indoor and outdoor datasets validate our model's efficiency and accuracy in multi-scene visual relocalization.

5.2 Future work

INR models offer a flexible and powerful coordinate-to-signal mapping formulation, utilizing a multi-layer perceptron (MLP) rather than complex neural network architectures, and show potential for continuous prediction across space and time. For the future work, we consider two directions to develop current research: (1) applying INR models to explore the relationships between signals and perceptions across diverse fields, and (2) generalizing INRs to unseen data or unapplied fields.

As shown in this thesis, we demonstrated how to handle different tasks in 3D scenarios with a unified representation (*i.e.*, INRs). Our models not only overcome the limitation of conventional discrete 3D representations in usage but also create connections among 1D, 2D, and 3D physical fields in practice. More signals including non-line-of-sight imaging, non-visible x-rays for computed tomography (CT), magnetic resonance imaging (MRI), pressure waves for audio, time-of-flight imaging, Synthetic aperture sonar (SAS), as well as volumetric light displays are waiting for our exploration with INRs.

Furthermore, exploring the fusion of multiple modalities could be an interesting research topic. For example, incorporating advanced language models (like the GPT) into vision-based INRs could enhance tasks such as 3D shape generation, editing, and transformation, super-resolution DEM generation from aerial images, *etc.* Integrating knowledge from different levels could also paves a promising way to solve practical problems. For example, introducing "high-level" semantic labels would prompt scene coordinate predictions in visual relocalization, using "mid-level" surface reconstruction would improve the accuracy of terrain structures and features in DEM super-resolution tasks, *etc.*

Finally, we will keep pursuing more precise, efficient, and flexible INR models for 3D shape representation, geographic data processing, and visual relocalization tasks.

Publications

1. **Shun Yao**, Fei Yang, Yongmei Cheng and Mikhail G. Mozerov. 3D Shapes Local Geometry Codes Learning with SDF. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (Spotlight), pages 2110-2117, 2021.

2. **Shun Yao**, Yongmei Cheng, Fei Yang, and Mikhail G. Mozerov. A continuous digital elevation representation model for DEM super-resolution. ISPRS Journal of Photogrammetry and Remote Sensing, 208, 1-13, 2024.

3. **Shun Yao**, Yongmei Cheng, Fei Yang, and Mikhail G. Mozerov. Implicit Neural Representation Model for Camera Relocalization with Learning in Global Multi-Scenes Scenario. Under review.

4. Zhaoxu Tian, Yongmei Cheng, **Shun Yao**, and Zhenwei Li. An Adaptive INS/CNS/SMN Integrated Navigation Algorithm in Sea Area. Remote Sensing, 16(4): 612, 2024.

5. Zhaoxu Tian, Yongmei Cheng, and **Shun Yao**. An Adaptive Fast Incremental Smoothing Approach to INS/GPS/VO Factor Graph Inference. Applied Sciences, 14(13): 5691, 2024.

Bibliography

[1] O. Argudo, A. Chica, and C. Andujar. Terrain super-resolution through aerial imagery and fully convolutional networks. *Computer Graphics Forum*, 37(2):101–110, 2018.

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] Matan Atzmon and Yaron Lipman. Sald: Sign agnostic learning with derivatives. In *International Conference on Learning Representations*, 2021.

[4] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Thuan Bui Bach, Tuan Tran Dinh, and Joo-Ho Lee. Featloc: Absolute pose regressor for indoor 2d sparse features with simplistic view synthesizing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:50–62, 2022.

[6] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017.

[8] Guido Borzi, Alejandro Roig, Carolina Tanjal, Lucía Santucci, Macarena Tejada Tejada, and Eleonora Carol. Flood hazard assessment in large plain

basins with a scarce slope in the pampean plain, argentina. *Environmental Monitoring and Assessment*, 193:1–14, 2021.

[9] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[10] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[12] Eric Brachmann and Carsten Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5847–5865, 2022.

[13] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[15] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera relocalisation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6218–6228, October 2021.

[16] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5044–5053, June 2023.

[17] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (10):2465–2477, 2020.

[18] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 608–625, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58526-6.

[19] Tony Chan and Wei Zhu. Level set based shape prior segmentation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 1164–1170. IEEE, 2005.

[20] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15668–15678, June 2021.

[21] Bowei Chen, Tiancheng Zhi, Martial Hebert, and Srinivasa G. Narasimhan. Learning continuous implicit representation for near-periodic patterns. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 529–546, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19784-0.

[22] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Latent partition implicit with surface codes for 3d representation. In *Computer Vision – ECCV 2022*, pages 322–343, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20062-5.

[23] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *Computer Vision – ECCV 2022*, pages 170–187, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19790-1.

[24] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8628–8638, June 2021.

[25] Z. Chen, X. Wang, Z. Xu, and W. Hou. Convolutional neural network based dem super resolution. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3: 247–250, 2016.

[26] Zhiqian Chen, Fanglan Chen, Lei Zhang, Taoran Ji, Kaiqun Fu, Liang Zhao, Feng Chen, Lingfei Wu, Charu Aggarwal, and Chang-Tien Lu. Bridging the gap between spatial and spectral domains: A unified framework for graph neural networks. *ACM Comput. Surv.*, 56(5), dec 2023.

[27] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] Julian Chibane, Mohamad Aymen mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21638–21652. Curran Associates, Inc., 2020.

[29] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[30] Bekir Z Demiray, Muhammed Sit, and Ibrahim Demir. D-srgan: Dem super-resolution with generative adversarial networks. *SN Computer Science*, 2(1):1–11, 2021.

[31] Bekir Z Demiray, Muhammed Sit, and Ibrahim Demir. Dem super-resolution with efficientnetv2. *arXiv preprint arXiv:2109.09661*, 2021.

[32] A. Dervieux and F. Thomasset. A finite element method for the simulation of a rayleigh-taylor instability. In *Approximation Methods for Navier-Stokes Problems*, pages 145–158. Springer, 1980.

[33] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 184–199, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

[34] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *2022 International Conference on 3D Vision (3DV)*, pages 393–402, 2022.

[35] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Golinski, Yee Whye Teh, and Arnaud Doucet. COIN++: Neural compression across modalities. *Transactions on Machine Learning Research*, 2022.

[36] Ron Ferens and Yosi Keller. Hyperpose: Camera pose localization using attention hypernetworks. *arXiv preprint arXiv:2303.02610*, 2023.

[37] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[38] Tomer Galanti and Lior Wolf. On the modularity of hypernetworks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10409–10419. Curran Associates, Inc., 2020.

[39] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8): 930–943, 2003.

[40] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[41] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] José Gomes and Olivier Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, 2000.

[43] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[44] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[45] Abner Guzman-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew Fitzgibbon, and Shahram Izadi. Multi-output learning for camera relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[46] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[47] Xiaoyi Han, Xiaochuan Ma, Houpu Li, and Zhanlong Chen. A global-information-constrained deep learning network for digital elevation model super-resolution. *Remote Sensing*, 15(2), 2023.

[48] Christian Hane, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68: 14–27, 2017. Automotive Vision: Challenges, Trends, Technologies and Systems for Vision-Based Intelligent Vehicles.

[49] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Trans. Graph.*, 38(4), jul 2019.

[50] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Hao He, Yixun Liang, Shishi Xiao, Jierun Chen, and Yingcong Chen. Cp-nerf: Conditionally parameterized neural radiance fields for cross-scene novel view synthesis. *Computer Graphics Forum*, 42(7):e14940, 2023.

[52] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, Gim Hee Lee, Marc Pollefeys, and Torsten Sattler. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4695–4702, 2019.

[53] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-hornung, and Daniel Cohen-or. Sape: Spatially-adaptive progressive encoding for neural optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8820–8832. Curran Associates, Inc., 2021.

[54] Ned Horning. Remote sensing. In Brian Fath, editor, *Encyclopedia of Ecology (Second Edition)*, pages 404–413. Elsevier, Oxford, second edition edition, 2019. ISBN 978-0-444-64130-4.

[55] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[56] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[57] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[58] Chiyu "Max" Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Niessner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[59] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH Comput. Graph.*, 18(3):165–174, jan 1984.

[60] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[61] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[62] Chiheon Kim, Doyup Lee, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Generalizable implicit neural representations via instance pattern composers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11808–11817, June 2023.

[63] Subin Kim, Sihyun Yu, Jaeho Lee, and Jinwoo Shin. Scalable neural video representations with learnable positional features. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 12718–12731. Curran Associates, Inc., 2022.

[64] Taehwan Kim and Tülay Adalı. Approximation by Fully Complex Multilayer Perceptrons. *Neural Computation*, 15(7):1641–1666, 07 2003.

[65] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[66] Doyup Lee, Chiheon Kim, Minsu Cho, and WOOK SHIN HAN. Locality-aware generalizable implicit neural representation. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48363–48381. Curran Associates, Inc., 2023.

[67] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1929–1938, June 2022.

[68] Joo Chan Lee, Daniel Rho, Seungtae Nam, Jong Hwan Ko, and Eunbyung Park. Coordinate-aware modulation for neural fields. In *The Twelfth International Conference on Learning Representations*, 2024.

[69] Hongwei Li, Tao Dai, Yiming Li, Xueyi Zou, and Shu-Tao Xia. Adaptive local implicit image function for arbitrary-scale super-resolution. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 4033–4037, 2022.

[70] Jing Li and David W.S. Wong. Effects of dem sources on hydrologic applications. *Computers, Environment and Urban Systems*, 34(3):251–261, 2010.

[71] Qing Li, Rui Cao, Kanglin Liu, Zongze Li, Jiasong Zhu, Zhenyu Bao, Xu Fang, Qingquan Li, Xianfeng Huang, and Guoping Qiu. Structure-guided camera localization for indoor environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:219–229, 2023.

[72] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[73] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[74] Shunlin Liang and Jindi Wang. Chapter 2 - geometric processing and positioning techniques. In Shunlin Liang and Jindi Wang, editors, *Advanced Remote Sensing (Second Edition)*, pages 59–105. Academic Press, second edition edition, 2020. ISBN 978-0-12-815826-5.

[75] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[76] Hyon Lim, Sudipta N. Sinha, Michael F. Cohen, Matt Uyttendaele, and H. Jin Kim. Real-time monocular image-based 6-dof localization. *The International Journal of Robotics Research*, 34(4-5):476–492, 2015.

[77] Xu Lin, Qingqing Zhang, Hongyue Wang, Chaolong Yao, Changxin Chen, Lin Cheng, and Zhaoxiong Li. A dem super-resolution reconstruction network combining internal and external learning. *Remote Sensing*, 14 (9):2181, 2022.

[78] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[79] Kanglin Liu, Qing Li, and Guoping Qiu. Posegan: A pose-to-image translation framework for camera localization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:308–315, 2020.

[80] Simon Lynen, Torsten Sattler, Michael Bosse, Joel Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, 07 2015.

[81] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7246–7257. PMLR, 18–24 Jul 2021.

[82] Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. *IEEE transactions on pattern analysis and machine intelligence*, 17(2):158–175, 1995.

[83] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14214–14223, October 2021.

[84] Lili Meng, Jianhui Chen, Frederick Tung, James J Little, and Clarence W de Silva. Exploiting random rgb and sparse features for camera pose estimation. In *BMVC*, 2016.

[85] Lili Meng, Jianhui Chen, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Backtracking regression forests for accurate camera relocalization. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6886–6893, 2017.

[86] Lili Meng, Frederick Tung, James J. Little, Julien Valentin, and Clarence W. de Silva. Exploiting points and lines in regression forests for rgb-d camera relocalization. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6827–6834, 2018.

[87] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[88] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface

representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019.

[89] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 268–283, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10605-2.

[90] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021.

[91] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[92] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), jul 2022.

[93] Tayyab Naseer and Wolfram Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530, 2017.

[94] Son Tung Nguyen, Alejandro Fontan, Michael Milford, and Tobias Fischer. Focustune: Tuning visual localization through focus-guided sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3606–3615, January 2024.

[95] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[96] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[97] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Meshloc: Mesh-based visual localization. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 589–609, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20047-2.

[98] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[99] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 523–540, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58580-8.

[100] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13032–13044. Curran Associates, Inc., 2021.

[101] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32 (1), Apr. 2018.

[102] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 313–318. Association for Computing Machinery, New York, NY, USA, 2003. ISBN 1581137095.

[103] G. Priestnall, J. Jaafar, and A. Duncan. Extracting urban features from lidar digital surface models. *Computers, Environment and Urban Systems*, 24(2):65–78, 2000.

[104] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[105] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[106] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019.

[107] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[108] Jerome Revaud, Yohann Cabon, Romain Brégier, JongMin Lee, and Philippe Weinzaepfel. Sacreg: Scene-agnostic coordinate regression for visual localization. *arXiv preprint arXiv:2307.11702*, 2023.

[109] J Saravanavel, SM Ramasamy, K Palanivel, and CJ Kumanan. Gis based 3d visualization of subsurface geology and mapping of probable hydrocarbon locales, part of cauvery basin, india. *Journal of Earth System Science*, 129:1–12, 2020.

[110] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[111] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[112] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3247–3257, June 2021.

[113] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674, 2011.

[114] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

[115] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[116] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2733–2742, October 2021.

[117] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[118] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[119] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc., 2020.

[120] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 835–846, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933646.

[121] Pablo Speciale, Johannes L. Schonberger, Sing Bing Kang, Sudipta N. Sinha, and Marc Pollefeys. Privacy preserving image-based localization.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[122] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[123] Shitao Tang, Chengzhou Tang, Rui Huang, Siyu Zhu, and Ping Tan. Learning camera localization via dense scene matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1831–1841, June 2021.

[124] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 293–309, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58517-4.

[125] Julien Valentin, Matthias Niessner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip H. S. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[126] Julien Valentin, Angela Dai, Matthias Niessner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 323–332, 2016.

[127] Johannes von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2019.

[128] Chuqi Wang. A review on 3d convolutional neural network. In *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 1204–1208, 2023.

[129] Huaxia Wang, Yongmei Cheng, Nan Liu, Yongqiang Zhao, Jonathan Cheung-Wai Chan, and Zhenwei Li. An illumination-invariant shadow-based scene matching navigation approach in low-altitude flight. *Remote Sensing*, 14(16), 2022.

[130] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. *International Journal of Computer Vision*, pages 1–21, 2024.

[131] Huayi Wu, Zhengwei He, and Jianya Gong. A virtual globe-based 3d visualization and interactive framework for public participation in urban planning processes. *Computers, Environment and Urban Systems*, 34(4): 291–298, 2010. Geospatial Cyberinfrastructure.

[132] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24, 2021.

[133] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41(2):641–676, 2022.

[134] Liyang Xiong, Guoan Tang, Xin Yang, and Fayuan Li. Geomorphology-oriented digital terrain analysis: Progress and perspectives. *Journal of Geographical Sciences*, 31:456–476, 2021.

[135] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021.

[136] Zekai Xu, Xuewen Wang, Zixuan Chen, Dongping Xiong, Mingyue Ding, and Wenguang Hou. Nonlocal similarity based dem super resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 110:48–54, 2015.

[137] Zekai Xu, Zixuan Chen, Weiwei Yi, Qiuling Gui, Wenguang Hou, and Mingyue Ding. Deep gradient prior network for dem super-resolution: Transfer learning from image to dem. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:80–90, 2019.

[138] Jingyu Yang, Sheng Shen, Huanjing Yue, and Kun Li. Implicit transformer network for screen content image continuous super-resolution. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13304–13315. Curran Associates, Inc., 2021.

[139] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[140] Shun Yao, Fei Yang, Yongmei Cheng, and Mikhail G. Mozerov. 3d shapes local geometry codes learning with sdf. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2110–2117, October 2021.

[141] Shun Yao, Yongmei Cheng, Fei Yang, and Mikhail G. Mozerov. A continuous digital elevation representation model for dem super-resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208:1–13, 2024.

[142] Kaiwei Zhang, Dandan Zhu, Xiongkuo Min, and Guangtao Zhai. Implicit neural representation learning for hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–12, 2022.

[143] Ruichen Zhang, Shaofeng Bian, and Houpu Li. Rspcn: Super-resolution of digital elevation model based on recursive sub-pixel convolutional neural networks. *ISPRS International Journal of Geo-Information*, 10 (8), 2021.

[144] Yifan Zhang and Wenhao Yu. Comparison of dem super-resolution methods based on interpolation and neural networks. *Sensors*, 22(3), 2022.

[145] Yifan Zhang, Wenhao Yu, and Di Zhu. Terrain feature-aware deep learning network for digital elevation model superresolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:143–162, 2022.

[146] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017.

[147] Jianqiao Zheng, Sameera Ramasinghe, Xueqian Li, and Simon Lucey. Trading positional complexity vs deepness in coordinate networks. In Shai

Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 144–160, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19812-0.

[148] Annan Zhou, Yumin Chen, John P. Wilson, Heng Su, Zhexin Xiong, and Qishan Cheng. An enhanced double-filter deep residual neural network for generating super resolution dems. *Remote Sensing*, 13(16), 2021.

[149] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 407–425, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20080-9.

[150] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9251–9262. Curran Associates, Inc., 2020.