



**POLITECNICO DI MILANO**

SCUOLA DI INGEGNERIA DEI SISTEMI

Corso di Laurea Magistrale in Ingegneria Matematica

**RICOSTRUZIONE ADATTATIVA  
DI DATI SPARSI MEDIANTE  
FUNZIONI A SIMMETRIA  
RADIALE**

Tesi di Laurea Magistrale

Alberto CRIVELLARO  
751750

Relatore:  
Ch.ma Prof.ssa Simona Perotto

Co-relatore  
Ch.mo Prof. Luca Formaggia

Anno Accademico 2010/2011



---

## Abstract

This work deals with the reconstruction of implicit surfaces from 3D scattered data. Our method is based on the radial basis function approach for the multivariate interpolation. Starting from an existing algorithm that establishes a multi-level reconstruction using compactly supported radial basis functions, we develop a new adaptative procedure for building the reconstruction at each scale. The tests we performed have shown that the new procedure appears to be attractive for its ability of computing more accurate reconstructions with less degrees of freedom than the original algorithm, while the computational time is almost the same. Some limitations due to the RBF interpolation method still remain, in particular in the reconstruction of noisy clouds of points of sharp featured clouds. Enhancing performances in such cases will be the object of the future research.



---

## Sommario

Il presente lavoro di tesi tratta della ricostruzione di superfici implicite a partire da dati sparsi tridimensionali. Il nostro metodo è basato sull'interpolazione multivariata mediante funzioni di base a simmetria radiale (RBF). Partendo da un algoritmo esistente che effettua una ricostruzione multi-scala utilizzando RBF a supporto compatto, abbiamo sviluppato una nuova procedura adattativa per effettuare la ricostruzione a ogni scala. I test che abbiamo effettuato hanno mostrato che la nuova procedura permette di calcolare ricostruzioni più accurate e allo stesso tempo più sintetiche di quelle offerte dall'algoritmo originale, mantenendo un tempo di calcolo quasi identico. Restano alcune limitazioni legate alla natura del metodo impiegato, in particolare nell'ambito della ricostruzione di dati affetti da rumore o di superfici con spigoli, angoli e punte. Migliorare le prestazioni del metodo in questi ambiti sarà l'oggetto della ricerca futura.



# Introduzione

Questo lavoro ha per scopo la messa a punto e la sperimentazione di metodi efficienti per la ricostruzione di superfici a partire da dati sparsi 3D mediante tecniche di interpolazione multivariata.

La ricostruzione e la gestione di superfici costituisce un ramo molto attivo della ricerca nelle scienze applicate. Il rilevamento di punti che giacciono sulla superficie di un oggetto e la successiva ricostruzione di un modello virtuale di tale oggetto costituiscono una procedura sempre più diffusa in numerosi settori, dalla medicina alle belle arti. Il modello virtuale, in seguito, può essere impiegato per gli scopi più vari: la semplice visualizzazione (come in alcuni videogiochi di ultima generazione), la creazione di un modello CAD completo (ad esempio, in applicazioni di *reverse engineering*), la creazione di mesh poligonali per simulazioni a parametri distribuiti, e così via.

Il modello virtuale di una superficie deve quindi essere versatile per l'utilizzo con finalità diverse; deve essere in grado di ricostruire insiemi di punti distribuiti in modo non uniforme (ad esempio, superfici con buchi o nuvole di punti provenienti da rilevamenti su sezioni non uniformi dell'oggetto); deve ricostruire fedelmente oggetti con una struttura topologica complessa; deve essere robusto rispetto a dati in ingresso affetti da rumore; deve essere rapido da calcolare ed economico in termini di memoria occupata.

Nuove sfide, infine, sono lanciate dalla continua evoluzione delle tecnologie per l'acquisizione di dati 3D: scanner e altri dispositivi permettono ormai l'acquisizione di decine di milioni di punti sulla superficie di un oggetto, mentre gli algoritmi di ricostruzione oggi disponibili arrivano a gestire al più insiemi di qualche milione di punti.

In questo lavoro di tesi, dopo una breve panoramica sulla ricostruzione di superfici basata su funzioni implicite, ci siamo focalizzati sullo studio di algoritmi che utilizzano la tecnica dell'interpolazione multivariata con funzioni a simmetria radiale (RBF), una famiglia di metodi particolarmente efficace e versatile per la ricostruzione di superfici anche molto complesse.

In particolare, ci siamo concentrati sugli algoritmi proposti in [25] e in [24], che abbiamo implementato in C/C++ e testato su differenti campioni di dati. Gli esperimenti hanno messo in evidenza pregi e difetti di ciascuno di questi due metodi. In particolare, uno dei due algoritmi si è rivelato più efficiente e robusto; a partire da quest'ultimo è stata messa a punto una

nuova procedura per la ricostruzione di superfici, di natura adattativa, che ha portato alla generazione di modelli più precisi e sintetici.

Per quanto riguarda il contenuto della tesi, nel Capitolo 1 viene presentata una contestualizzazione dei diversi campi applicativi dei metodi di ricostruzione basati su superfici implicite, unitamente ad alcuni metodi per la poligonalizzazione di superfici implicite. Nel Capitolo 2 viene presentato il problema dell'interpolazione multivariata RBF, di cui vengono esplorati anche aspetti tecnici e proprietà di buona approssimazione. Nel Capitolo 3 sono descritti i due algoritmi proposti in [25] e in [24] che abbiamo implementato, dapprima, nella loro versione originale. Infine, nel Capitolo 4 sono illustrati i risultati delle prove sperimentali effettuate e viene introdotta la procedura adattativa da noi messa a punto che testiamo e confrontiamo con gli algoritmi originali.



# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Ricostruzione di superfici implicite e interpolazione</b>                                    | <b>1</b>  |
| 1.1      | Contestualizzazione del problema . . . . .  | 2         |
| 1.2      | Interpolazione multivariata per la ricostruzione di superfici . . . . .                         | 2         |
| 1.2.1    | Applicazioni . . . . .  | 2         |
| 1.2.2    | Dispositivi di acquisizione 3D . . . . .  | 3         |
| 1.3      | Interpolazione multivariata: altre applicazioni . . . . .                                       | 5         |
| 1.4      | Descrivere matematicamente una superficie . . . . .   | 7         |
| 1.5      | Generazione di mesh poligonali a partire da superfici implicite . . . . .                       | 9         |
| 1.6      | Ricostruzione di superfici: stato dell'arte . . . . .   | 12        |
| <br>     |   |           |
| <b>2</b> | <b>Ricostruzione con funzioni a simmetria radiale</b>   | <b>15</b> |
| 2.1      | Interpolazione di funzioni multivariate . . . . .   | 15        |
| 2.1.1    | Teorema di Mairhuber–Curtis . . . . .   | 16        |
| 2.2      | Interpolazione mediante funzioni di base radiali . . . . .                                      | 17        |
| 2.2.1    | Dati supplementari . . . . .  | 18        |
| 2.3      | Buona posizione del problema di interpolazione RBF . . . . .                                    | 19        |
| 2.4      | Scelta della funzione radiale . . . . .   | 22        |
| 2.5      | Caratterizzazione di funzioni definite positive e condizionatamente definite positive . . . . . | 24        |
| 2.5.1    | Primo criterio . . . . .  | 24        |
| 2.5.2    | Secondo criterio . . . . .  | 25        |
| 2.6      | Ottimalità dell'interpolazione RBF . . . . .  | 26        |
| 2.6.1    | Spazio nativo . . . . .   | 27        |
| 2.6.2    | Ricostruzione ottimale . . . . .  | 29        |
| 2.7      | Stime dell'errore di interpolazione mediante RBF . . . . .                                      | 30        |
| 2.8      | Principio di indeterminazione . . . . .   | 31        |
| <br>     |   |           |
| <b>3</b> | <b>Due algoritmi efficienti per la ricostruzione di superfici</b>                               | <b>35</b> |
| 3.1      | Un algoritmo multi-livello con RBF a supporto compatto . . . . .                                | 36        |
| 3.2      | Interpolazione multivariata mono-livello . . . . .  | 37        |
| 3.2.1    | Scelta della funzione interpolante . . . . .  | 37        |
| 3.2.2    | Calcolo dell'interpolazione con approccio misto . . . . .                                       | 38        |
| 3.2.3    | Limiti dell'interpolazione multivariata mono-livello . . . . .                                  | 40        |

|          |   |           |
|----------|---|-----------|
| 3.3      | Ricostruzione multi-livello . . . . .   | 42        |
| 3.3.1    | Creazione di livelli gerarchici . . . . .   | 42        |
| 3.3.2    | Ricostruzione multi-livello . . . . .   | 42        |
| 3.4      | Un algoritmo adattativo con RBF a supporto compatto . . . . .   | 44        |
| 3.4.1    | Calcolo dell'approssimante locale . . . . .   | 46        |
| 3.4.2    | Calcolo del parametro $\sigma$ . . . . .  | 46        |
| 3.4.3    | Selezione dei centri di approssimazione . . . . .   | 48        |
| 3.4.4    | Calcolo dei pesi scalari $\lambda_i$ . . . . .  | 48        |
| 3.5      | Conciliare i vantaggi di RBF a supporto globale e a supporto compatto: una sfida possibile? . . . . . | 49        |
| <b>4</b> | <b>Prove sperimentali. Innovazioni. Prospettive</b>   | <b>51</b> |
| 4.1      | Sperimentazioni. Casi test . . . . .  | 52        |
| 4.2      | Test sull'algoritmo adattativo . . . . .  | 52        |
| 4.3      | Test sull'algoritmo multi-scala . . . . .   | 54        |
| 4.3.1    | Proprietà della funzione interpolante . . . . .   | 55        |
| 4.3.2    | Robustezza rispetto alla distribuzione spaziale dei punti   | 56        |
| 4.3.3    | Sensibilità al rumore . . . . .   | 57        |
| 4.4      | Algoritmo multi-scala: dipendenza da parametri; alcune innovazioni. . . . .                           | 58        |
| 4.4.1    | Supporto delle funzioni di base . . . . .   | 59        |
| 4.4.2    | Modalità selezione livelli . . . . .  | 59        |
| 4.4.3    | Offset sui primi livelli . . . . .  | 60        |
| 4.5      | Creazione adattativa dei livelli . . . . .  | 62        |
| 4.5.1    | Selezione adattativa del numero di punti . . . . .  | 65        |
| 4.5.2    | Prove sperimentali sulla selezione adattativa dei punti   | 66        |
| 4.6      | Un'applicazione in campo medico . . . . .   | 67        |
| 4.7      | Conclusioni.Prospettive . . . . .   | 68        |
| <b>A</b> | <b>Appendice A: Strutture dati spaziali</b>   | <b>73</b> |
| A.1      | Kd-tree . . . . .   | 74        |
| A.1.1    | Kd-tree per l'implementazione di algoritmi di ricostruzione di superfici. . . . .                     | 76        |
| A.1.2    | Costruzione dell'albero . . . . .   | 76        |
| A.1.3    | Ricerca spaziale . . . . .  | 78        |
| <b>B</b> | <b>Appendice B: Interpolazione basata su partizione dell'unità</b>                                    | <b>81</b> |
| B.1      | Rappresentazione basata su partizione dell'unità . . . . .  | 81        |
| B.2      | Partizione dell'unità multi-scala . . . . .   | 82        |
| B.3      | Approccio misto: partizione dell'unità e RBF . . . . .  | 84        |
|          | <b>Bibliografia</b>   | <b>85</b> |

# Elenco delle figure

|     |   |    |
|-----|---|----|
| 1.1 | Esempi di modelli virtuali impiegati in ambiti diversi: (a) una biforcazione carotidea; (b) il David di Michelangelo; (c) un circuito automobilistico; (d) un impianto chimico. . . . .   | 4  |
| 1.2 | Principali dispositivi di acquisizione 3D. . . . .  | 5  |
| 1.3 | Descrizione strutturale (a sinistra) e aerodinamica di un aereo (da [34]). . . . .  | 6  |
| 1.4 | Una superficie semi-sferica. . . . .  | 7  |
| 1.5 | Marching cubes: possibili intersezioni fra un cubo e una superficie (da [19]). . . . .  | 10 |
| 1.6 | Passi principali dell'algoritmo di Bloomenthal (da [1]). . . . .  | 12 |
| 3.1 | Approssimazione locale (in nero) e interpolante globale (in rosso) ottenuta da una traslazione delle approssimanti locali. . . . .  | 38 |
| 3.2 | Divisione ricorsiva della bounding box. . . . .   | 40 |
| 3.3 | Stanford Bunny: modello originale (a sinistra) e modello ricostruito mediante interpolazione diretta partendo da una nuvola $X$ di $N = 34833$ punti. . . . .   | 41 |
| 3.4 | Limiti dell'interpolazione diretta: (a) i buchi nella nuvola originale di dati non vengono riempiti dalla ricostruzione; (b) supporto della funzione interpolante per una sezione verticale della bounding box della nuvola $X$ . . . . . | 42 |
| 3.5 | Ricostruzione multi-livello di un arco aortico ascendente utilizzando un approccio a 4 livelli e per $N = 9591$ . . . . .   | 44 |
| 3.6 | Andamento dell'errore per il centro indicato dalla freccia in una ricostruzione dello Stanford Bunny. . . . .   | 47 |
| 4.1 | Passi della ricostruzione multi-scala dello Stanford Bunny su 6 livelli; l'ultimo livello corrisponde a una interpolazione di tutti i punti della nuvola. . . . .   | 56 |
| 4.2 | Insiemi di livello per $F$ derivante dalla ricostruzione riassunta in Tabella 4.3 lungo due sezioni della bounding box. . . . .   | 57 |

|     |  |    |
|-----|--|----|
| 4.3 | Particolare dalla ricostruzione riassunta in Tabella 4.3: la ricostruzione è effettuata anche per regioni con “buchi” nella nuvola di punti. . . . .                                       | 57 |
| 4.4 | Ricostruzione multi-scala di una nuvola con forte discontinuità della distribuzione spaziale dei punti. . . . .  | 58 |
| 4.5 | Errore di ricostruzione al variare del parametro $c$ . . . . .   | 59 |
| 4.6 | Errore di ricostruzione per diverse modalità di selezione dei punti in ogni livello. . . . .   | 62 |
| 4.7 | Ricostruzione di drago ( $N = 437645$ ). . . . .   | 64 |
| 4.8 | Passi della ricostruzione multi-scala di un cervello su 6 livelli ( $N = 46650$ ). Sotto ogni immagine è indicato il numero di punti $N_k$ appartenenti al livello corrispondente. . . . . | 65 |
| 4.9 | Ricostruzione di una nuvola di punti su sezioni parallele. . . . .   | 67 |
| A.1 | Kd-tree su un insieme di dati bidimensionali: elementi memorizzati in tutti i nodi. . . . .  | 75 |
| A.2 | Kd-tree su un insieme di dati bidimensionali: elementi memorizzati esclusivamente sulle foglie. . . . .  | 75 |
| A.3 | Partizione dello spazio e octree. . . . .  | 76 |
| A.4 | 2d-tree costruito con l’algoritmo 4. . . . .   | 78 |
| A.5 | Regione associata a un nodo. . . . .   | 79 |
| A.6 | Ricerca dei punti nella regione $\mathcal{R}$ (a sinistra, in grigio chiaro). . . . .  | 79 |

# Elenco delle tabelle

|     |   |    |
|-----|---|----|
| 2.1 | Alcune RBF a supporto globale. . . . .  | 22 |
| 2.2 | RBF a supporto compatto proposte da H.Wendland. . . . .   | 34 |
| 2.3 | Famiglie di RBF e <i>bounds</i> sull'errore di interpolazione RBF. . . . .  | 34 |
| 4.1 | Campioni utilizzati per i test presentati in questo capitolo. . . . .   | 53 |
| 4.2 | Tre ricostruzioni dello Stanford Bunny con algoritmo adattativo. Sono riportate la taglia massima del supporto $\sigma_{max}$ , la taglia minima $\sigma_{min}$ e il numero di centri selezionati $M$ . . . . . | 54 |
| 4.3 | Punti inclusi in ogni livello $X_n$ per la ricostruzione dello Stanford Bunny ( $N = 34833$ ). . . . .  | 55 |
| 4.4 | Ricostruzioni con e senza offset per il numero di celle dei livelli. Il risultato finale della Ricostruzione 2 è riportato in figura 4.7. . . . .   | 63 |
| 4.5 | Ricostruzioni originale e con selezione adattativa dei punti per modello di cervello (cf. Figura 4.8). . . . .  | 67 |



”Si deve riconoscere che (...) la scienza è tenuta in pregio per i suoi risultati pratici : ma è tenuta in pregio ancor più grande per il suo contenuto informativo e per la sua capacità di liberare la nostra mente dalle credenze vecchie, dai vecchi pregiudizi e dalle vecchie certezze, e di offrirci, al loro posto, nuove congetture e ipotesi ardite. La scienza è tenuta in pregio per la sua influenza liberalizzatrice ; come una delle forze più grandi che operino per la libertà umana.”

*K. R. Popper*





# 1

## Ricostruzione di superfici implicite e interpolazione

Le tecniche presentate in questo lavoro hanno per scopo la ricostruzione di una superficie a partire da una serie di punti rilevati su di essa. L'ambito di riferimento è quello del trattamento dei dati provenienti da dispositivi di acquisizione 3D per la scansione di oggetti. Come osservato nella Sezione 1.1, il problema della ricostruzione può essere formulato matematicamente come un problema di interpolazione multivariata. Nella Sezione 1.2 presentiamo una breve rassegna degli ambiti in cui la ricostruzione di superfici è maggiormente utilizzata e dei tipi principali di dispositivi di acquisizione solitamente utilizzati.

Il problema dell'interpolazione multivariata da noi impiegato per modellizzare la ricostruzione di una superficie trova molte altre applicazioni, di cui forniamo alcuni esempi nella Sezione 1.3. Anticipiamo fin da ora che, sebbene nel seguito si faccia costantemente riferimento alla ricostruzione di superfici, le tecniche presentate nei Capitoli 3 e 4 possono essere facilmente adattate per la risoluzione di problemi di interpolazione multivariata negli ambiti più diversi.

Nella Sezione 1.4 sono descritti i diversi modi in cui una superficie può essere descritta matematicamente. Una delle descrizioni più diffuse è quella *implicita* e nel seguito si farà uso di tale descrizione. Nella Sezione 1.5 descriviamo alcuni metodi utilizzati per la manipolazione di superfici descritte implicitamente, noti come metodi di poligonalizzazione.

Infine nella Sezione 1.6 presentiamo lo stato dell'arte della ricostruzione di superfici nella letteratura scientifica attuale.

## 1.1 Contestualizzazione del problema

Il problema della ricostruzione di superfici può essere espresso in modo conveniente come un problema di *interpolazione multivariata*: data una serie di dati campionati che consistono in un insieme finito  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  di punti nello spazio (*nuvola di punti*) e di valori  $f_j = f(\mathbf{x}_j)$ , con  $j = 1 \dots N$ , assegnati per ogni punto  $\mathbf{x}_j$ , trovare una funzione  $s$  che *interpoli* i dati, ossia una funzione tale che  $s(\mathbf{x}_j) = f_j$ , con  $j = 1 \dots N$ . Visto che spesso i dati di partenza sono ottenuti mediante misurazioni e possono quindi essere affetti da errori, in alcune applicazioni si preferisce optare per un'approssimazione dei dati piuttosto che per un'interpolazione, ossia si richiede che  $s(\mathbf{x}_j) \approx f_j$ , per  $j = 1 \dots N$ .

Il problema dell'interpolazione di funzioni univariate è un problema classico dell'analisi matematica ed esistono molti metodi che permettono di risolverlo efficacemente: funzioni splines, polinomi algebrici, polinomi trigonometrici, etc. (per un approfondimento si veda, ad esempio, [?]). Sorprendentemente, gli stessi strumenti si rivelano spesso insufficienti ad affrontare lo stesso problema per funzioni multivariate, come metteremo in evidenza nel Capitolo 2.

L'interpolazione/approssimazione di funzioni multivariate è un problema totalmente trasversale alla matematica applicata e costituisce un ambito di ricerca molto attivo, i cui risultati trovano applicazione in campi diversissimi come la geologia, la medicina, la biologia, la computer grafica, le belle arti, etc.

È opportuno sottolineare che l'*interpolazione di funzioni* e la *ricostruzione di superfici* sono a priori due concetti ben distinti: il primo indica un metodo matematico astratto, il secondo un problema pratico che può essere risolto in diversi modi. Questo lavoro prende spunto dalle numerose applicazioni (in particolare dalla scansione tridimensionale, cf. paragrafo 1.2) in cui si pone il problema della ricostruzione di superfici e la tecnica dell'interpolazione funzionale viene impiegata come strumento per affrontarlo; per questo motivo, i due termini verranno a volte impiegati con significato equivalente con un piccolo abuso di linguaggio. Inoltre, osserviamo che le tecniche di interpolazione presentate nel seguito hanno una valenza del tutto generale: in particolare, esse possono essere impiegate anche per scopi diversi dalla ricostruzione di superfici, a cui si accenna nella Sezione 1.3.

## 1.2 Interpolazione multivariata per la ricostruzione di superfici

### 1.2.1 Applicazioni

Le tecniche presentate in questo lavoro permettono di ricostruire una superficie continua a partire dal rilevamento di punti che giacciono su di essa.

In particolare, l'ambito a cui facciamo riferimento è il trattamento dei dati provenienti da scanner 3D. Questa definizione racchiude una grande varietà di strumenti con lo scopo di acquisire informazioni sulla forma e sulle dimensioni di un oggetto per poi ricostruirne un modello virtuale (per ulteriori dettagli si veda il Paragrafo 1.2.2). La maggior parte delle apparecchiature rileva (o permette di ricavare) un insieme di punti che giacciono sulla superficie, più alcune informazioni supplementari quali, ad esempio, le normali alla superficie per ogni punto misurato; i dati acquisiti sono poi utilizzati per la ricostruzione. Una volta ricostruito, il modello virtuale dell'oggetto può essere impiegato per gli scopi più diversi.

In Figura 1.1 mostriamo 3 esempi di applicazioni possibili: il modello ricostruito di una biforcazione carotidea, utilizzato per una simulazione di emodinamica (da [18]); la celebre ricostruzione virtuale del David di Michelangelo realizzata grazie al Digital Michelangelo Project (per maggiori informazioni si consulti <http://graphics.stanford.edu/projects/mich/>); un simulatore di guida di Formula 1, le cui ambientazioni sono state ricostruite fedelmente grazie a scansioni tridimensionali; un impianto industriale.

In medicina, modelli di organi e parti anatomiche possono essere ricostruiti grazie alla tomografia e utilizzati per eseguire simulazioni di emodinamica (cf. [18]) o per progettare protesi. In geologia è possibile ricostruire modelli tridimensionali del sottosuolo a partire dai dati rilevati sperimentalmente (cf. [9]). Un'altra applicazione molto utilizzata nell'industria è quella del *reverse engineering*, che permette di ottenere modelli virtuali (CAD) a partire da oggetti reali (pezzi meccanici, modellini, sculture, etc.) per manipolazioni, simulazioni, produzione con utensili automatici.

### 1.2.2 Dispositivi di acquisizione 3D

Esistono numerose tecnologie di scansione tridimensionale, ognuna delle quali comporta vantaggi, costi, limitazioni diverse. Fra le caratteristiche che determinano le performances di un apparecchio per la scansione 3D vi sono: la *portata* (distanza massima dell'oggetto); la *risoluzione*; la *velocità* di scansione; la *portatilità* (esistono in commercio scanner da tenere in mano); la misura di *informazioni supplementari* oltre alla posizione dei punti sulla superficie (ad esempio i vettori normali alla superficie in ogni punto, il colore, il materiale, o addirittura la velocità per oggetti in movimento); la *robustezza* rispetto a svariati fattori che possono rendere problematica la scansione, come la presenza di materiali trasparenti o riflettenti, la presenza di fonti luminose esterne, etc.

È possibile classificare le tecniche di scansione 3D in due grandi categorie: *a contatto* e *non a contatto*. Gli apparecchi basati su tecniche a contatto impiegano sonde che scorrono sulla superficie e rilevano la posizione dei punti; le tecniche non a contatto possono essere ulteriormente suddivise in tecniche

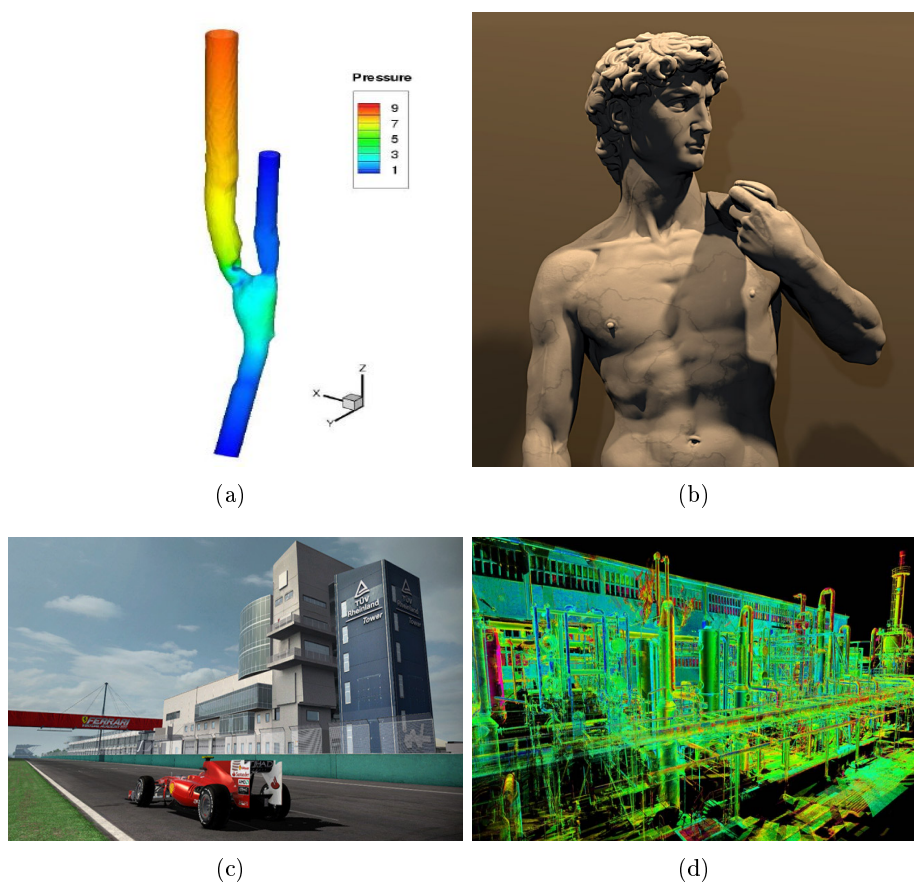


Figura 1.1: Esempi di modelli virtuali impiegati in ambiti diversi: (a) una biforcazione carotidea; (b) il David di Michelangelo; (c) un circuito automobilistico; (d) un impianto chimico.

*attive* e tecniche *passive*. Le tecniche attive prevedono l'invio di segnali (raggi laser, lame di luce, ultrasuoni, etc.) sulla superficie e l'ottenimento di informazioni mediante misure sul segnale di ritorno: ad esempio, gli scanner *a tempo di volo*, al fine di ricavare la distanza di ogni punto, misurano il tempo impiegato dal segnale per arrivare sulla superficie, essere riflesso e tornare alla fonte; altri scanner impiegano metodi di *triangolazione* per misurare la distanza a cui si trova ogni punto rilevato sulla superficie; esistono inoltre dispositivi basati sull'uso di *luce strutturata*, che proiettano sull'oggetto un pattern di luce noto a priori (ad esempio, una griglia o delle righe) e rilevano come il pattern viene deformato dalla superficie dell'oggetto; infine, ricordiamo fra le tecniche attive scanner le tecniche *volumetriche*, molto usate in campo medico, che permettono di rilevare punti di superfici all'interno di altri oggetti: ne sono esempi noti la risonanza magnetica e la tomografia.

Le tecnologie non a contatto passive si basano invece sull'analisi di diverse

immagini di un oggetto acquisite da punti di vista differenti; fra di esse ricordiamo i sistemi stereoscopici (impiegati anche per la visione di immagini tridimensionali) e la fotometria.

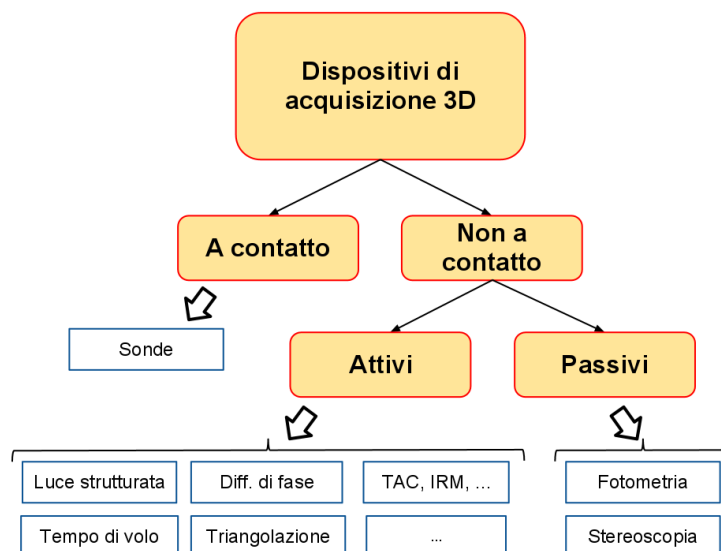


Figura 1.2: Principali dispositivi di acquisizione 3D.

I principali tipi di dispositivi di acquisizione sono riassunti in Figura 1.2. In questo lavoro non sono prese in esame le diverse tecnologie di acquisizione dei dati. Visto che la maggior parte delle tecniche citate permette di ottenere, direttamente o grazie ad una prima elaborazione dei dati, una *nuvola di punti*  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  che giacciono sulla superficie da ricostruire, assumiamo che questo sarà il nostro punto di partenza per applicare i metodi e gli algoritmi presentati nel seguito.

### 1.3 Interpolazione multivariata: altre applicazioni

Come anticipato, la ricostruzione di superfici in senso stretto non è l'unica applicazione possibile delle tecniche presentate in questo lavoro. Un interessante esempio di applicazione è costituito dalla risoluzione di problemi complessi ripartiti in sotto-problemi da esaminare separatamente; questo è il caso, ad esempio, di problemi di interazione fluido-struttura in simulazioni di viscoelasticità.

Assumiamo di essere interessati alla simulazione del comportamento dinamico della struttura di un aereo in volo. Alcuni modelli semplificati sono stati sviluppati per descrivere efficacemente lo scorrimento del fluido sulla superficie rigida dell'aereo (ad esempio, le equazioni di Navier-Stokes); altri permettono di predire le deformazioni di una struttura elastica sottoposta

a uno sforzo (causato, in questo caso, dal campo di pressione provocato dal fluido attorno all'aereo). Il comportamento complessivo dell'aereo è dato, ovviamente, dall'interazione di questi due fenomeni. È possibile dunque procedere in questo modo: dapprima si approssima numericamente il campo di pressione attorno all'aereo, approssimato come una generica struttura rigida; poi si calcolano le deformazioni della struttura sottoposta a tale campo di pressione; a questo punto si itera il procedimento fino a giungere a convergenza, con una procedura di tipo "punto fisso". Ora, mentre la simulazione del fluido richiede una descrizione molto dettagliata della superficie dell'aeromobile con una *mesh* molto fitta soprattutto nei punti di impatto del fluido con la struttura, nel calcolo della deformazione elastica una descrizione troppo fine è inutile e spesso addirittura dannosa ai fini della convergenza del calcolo: sono quindi necessarie modellizzazioni diversificate della struttura dell'aereo (si veda la Figura 1.3).

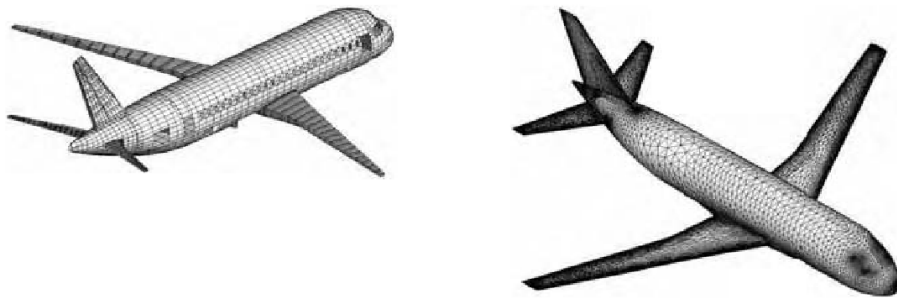


Figura 1.3: Descrizione strutturale (a sinistra) e aereodinamica di un aereo (da [34]).

Tecniche di interpolazione multidimensionale sono a questo punto necessarie per trasferire le informazioni calcolate su un modello sull'altro (ad esempio il valore del campo di pressione calcolato nei nodi della *mesh* associata alla struttura per la simulazione di fluidodinamica). Tali tecniche permettono di trasferire informazioni fra due modelli qualsiasi i cui dati siano specificati su nodi: i modelli possono anche essere dimensionalmente diversi (ad esempio, un modello bidimensionale e uno tridimensionale) e non è richiesta la presenza di una struttura connessa sottostante (ad esempio, una *mesh* poligonale).

Un ulteriore campo di applicazione per tecniche di interpolazione multidimensionale è costituito dai metodi di tipo semi-lagrangiano per l'integrazione numerica di equazioni alle derivate parziali. Tali metodi, molto usati, ad esempio, in meteorologia, si basano sull'idea di seguire la traiettoria delle particelle sottoposte a un campo di forze (approccio *Lagrangiano*) piuttosto che investigare in punti fissi dello spazio le variazioni temporali delle quantità che descrivono il sistema (approccio *Euleriano*). Tali metodi

prevedono inevitabilmente l'uso di tecniche di interpolazione al fine di descrivere la posizione delle particelle ai diversi istanti temporali. Rimandiamo a [34] per ulteriori dettagli.

## 1.4 Descrivere matematicamente una superficie

Come è stato messo in evidenza nella sezione precedente, la ricostruzione di superfici è un problema presente in moltissime applicazioni. Matematicamente, esistono diversi modi per descrivere una superficie nello spazio. Ne citiamo alcuni, riportando come esempio le rispettive rappresentazioni della superficie semi-sferica rappresentata in Figura 1.4.

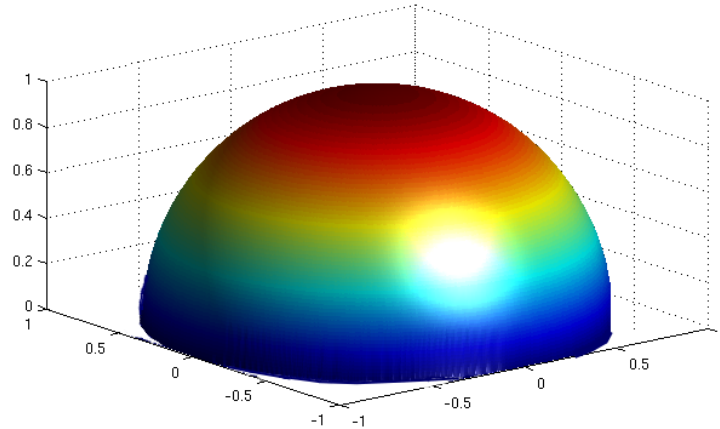


Figura 1.4: Una superficie semi-sferica.

- **Grafico di funzione.** Una prima possibilità è quella di descrivere una superficie attraverso il grafico di una funzione  $f : \Omega \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ . Questo approccio, talvolta utilizzato in geologia, è molto limitante in quanto richiede forti vincoli sulla struttura topologica della superficie che, in teoria, non è nota a priori. Questo tipo di descrizione per la superficie in Figura 1.4 è data da

$$\begin{cases} f : \Omega \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}, & (x, y) \rightarrow \sqrt{1 - x^2 - y^2}, \\ \text{con } \Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}. \end{cases}$$

Si noti che, ad esempio, una sfera intera non può essere rappresentata con questo tipo di descrizione.

- **Descrizione parametrica.** Un altro approccio possibile si basa su una descrizione *parametrica*, secondo cui le coordinate dei punti della superficie sono descritte al variare di uno o più parametri. Ad esempio,

la *parametrizzazione* di una superficie a 2 parametri è rappresentata da un'applicazione

$$\tau : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad (u, v) \rightarrow (x(u, v), y(u, v), z(u, v)). \quad (1.1)$$

Nel caso di superfici complesse possono essere impiegate diverse parametrizzazioni per parti differenti della superficie. La semi-sfera della Figura 1.4 è descritta da:

$$\tau : [0, \pi] \times [0, 1] \rightarrow \mathbb{R}^3 : \quad (u, v) \rightarrow \begin{cases} x(u, v) = \sqrt{1-v^2} \cos(u) \\ y(u, v) = \sqrt{1-v^2} \sin(u) \\ z(u, v) = v \end{cases}$$

- **Superficie implicita.** Un'ulteriore alternativa consiste nel descrivere una superficie come un insieme di livello di una funzione scalare definita nello spazio  $\mathbb{R}^3$ , ovvero, data  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ , la superficie è definita *implicitamente* dall'insieme dei punti  $X = \{\mathbf{x} \in \mathbb{R}^3 | F(\mathbf{x}) = c\}$ , dove  $c$  è un parametro reale (spesso si sceglie  $c = 0$ ). La descrizione della semi-sfera in Figura 1.4 è data in questo caso da

$$X = \{(x, y, z) \in \mathbb{R}^3 | z \geq 0 \text{ e } x^2 + y^2 + z^2 - 1 = 0\}.$$

Questo tipo di rappresentazione presenta innumerevoli vantaggi: permette una descrizione relativamente semplice di superfici anche molto complesse; la funzione  $F$  può essere definita in modo tale che  $F(\mathbf{x})$  fornisca, in ogni punto  $\mathbf{x}$  dello spazio, una stima della distanza dalla superficie; operazioni come la fusione e la deformazione di superfici, l'intersezione, l'unione, la differenza fra oggetti sono facilmente gestibili; la funzione  $F$  può essere valutata in ogni punto dello spazio, permettendo di ingrandire/rimpicciolire dettagli a piacere; infine, osserviamo che in ogni punto della superficie il vettore normale può essere ricavato dalla valutazione del gradiente di  $F$ . Il rovescio della medaglia è costituito dal fatto che una superficie definita implicitamente non è direttamente visualizzabile: una volta ottenuta l'espressione della funzione  $F$ , è necessaria una tappa supplementare per estrapolare una serie di punti che giacciono sulla superficie descritta da  $F$ .

La rappresentazione implicita di superfici è molto diffusa per i vantaggi elencati sopra ed è quella impiegata in questo lavoro. Osserviamo, tuttavia, che non esiste una rappresentazione ideale ma, a seconda dei casi, può essere preferibile impiegare un tipo di rappresentazione piuttosto che un altro; ricordiamo infine che esistono in letteratura numerosi metodi che permettono di passare da una rappresentazione all'altra (si veda, ad esempio, [11] o [12]).



## 1.5 Generazione di mesh poligonali a partire da superfici implicite

Grazie alle tecniche di interpolazione di funzioni multivariate è dunque possibile definire una funzione  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  valutabile in ogni punto dello spazio; l'insieme di livello zero di tale funzione (ossia l'insieme formato dai punti  $\mathbf{x}$  tali che  $F(\mathbf{x}) = 0$ ) definisce una superficie che interpola una nuvola di punti  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ . Di solito, la funzione  $F$  viene scelta in modo da definire una “distanza con segno” dalla superficie: ha modulo crescente a mano a mano che ci si allontana da essa e cambia segno laddove, in presenza di una superficie chiusa, si passi dall'interno all'esterno. Ottenuta  $F$ , nella gran parte dei casi è necessario costruire una mesh poligonale che approssimi la superficie, o almeno identificare una serie di punti che giacciono su di essa per poter effettuare operazioni di calcolo e di visualizzazione. A tal fine esistono diversi algoritmi in letteratura. Presentiamo brevemente due fra gli algoritmi più diffusi e l'algoritmo utilizzato per visualizzare le superfici presentate negli esempi di questa tesi.

- **Marching cubes**: questo metodo, introdotto nel 1987 in [19], permette di ricavare, partendo da  $F$ , una mesh non strutturata che giace sulla superficie  $F(\mathbf{x}) = 0$ . Tale metodo consiste nei seguenti passi:
  1. introduzione di una *bounding box* contenente la nuvola di punti; suddivisione uniforme della bounding box in cubi di taglia identica;
  2. identificazione dei cubi intersecati dalla superficie mediante valutazione di  $F$  in tutti i vertici e verifica di eventuali cambi di segno di  $F$  fra un vertice e l'altro;
  3. ricostruzione del modo in cui la superficie interseca ogni cubo a partire dalla configurazione dei vertici: sono possibili 256 patterns diversi che, per simmetria e rotazione, si riducono a 15, riassunti in Figura 1.5;
  4. calcolo delle normali alla superficie nei punti di intersezione con gli spigoli dei cubi; risoluzione di eventuali ambiguità e generazione della mesh.

I punti utilizzati come vertici della mesh sono quelli di intersezione fra la superficie e lo spigolo di un cubo; essi sono ricavati attraverso un'interpolazione lineare dei vertici del cubo all'estremità dello spigolo.

- **Marching tetrahedra**: questo algoritmo è molto simile al precedente salvo che, al posto di impiegare dei cubi come celle di segmentazione, vengono impiegati dei tetraedri. I passi 1 e 2 di questa procedura sono

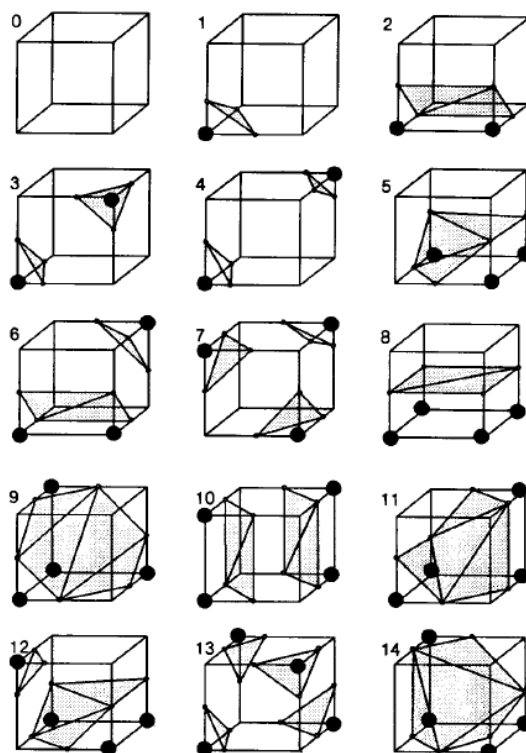


Figura 1.5: Marching cubes: possibili intersezioni fra un cubo e una superficie (da [19]).

identici a quelli del marching cubes; identificati però, i cubi che intersecano la superficie non vengono esaminati direttamente, ma sono dapprima suddivisi in tetraedri; i tetraedri sono poi impiegati per la costruzione della mesh in modo del tutto analogo all'algoritmo precedente. L'uso di tetraedri riduce notevolmente il numero di configurazioni possibili da esaminare (16, ridotte a 8 per simmetria) e semplifica il trattamento delle ambiguità. Ogni punto d'intersezione fra la superficie e un tetraedro è calcolato attraverso un'interpolazione lineare fra i due vertici corrispondenti.

- **Varianti: algoritmo di Bloomenthal:** nella pratica, le tecniche marching cubes e marching tetrahedra sono molto utilizzate e sono state proposte diverse versioni degli algoritmi di base. Alcune varianti di natura adattativa introducono cubi di taglia variabile a seconda del livello di dettaglio richiesto in ogni zona della superficie; altre introducono metodi di *continuazione* per la ricerca dei cubi intersecati dalla superficie in sostituzione di una ricerca esaustiva, che consiste nella suddivisione della bounding box mediante griglie cartesiane e successi-

va valutazione della funzione in tutti i vertici dei cubi così ottenuti; il costo computazionale si riduce così da  $O(n^3)$  a  $O(n^2)$ , dove  $n$  denota la dimensione caratteristica della bounding box della superficie.

L'algoritmo di Bloomenthal, introdotto in [1] e in [2], è un algoritmo basato sul marching tetrahedra; la dimensione dei cubi è fissa, la ricerca dei cubi avviene mediante continuazione e la ricerca dei punti di intersezione con la superficie è effettuata mediante il metodo della bisezione<sup>1</sup> (al posto dell'interpolazione lineare). Questo algoritmo si rivela di semplice implementazione, rapido ed efficace. Esso è stato utilizzato per la visualizzazione di tutte le superfici ricostruite mostrate in questo lavoro. In Figura 1.6 sono rappresentate le tappe principali dell'algoritmo di Bloomenthal:

1. *continuazione*: partendo da un punto iniziale  $\mathbf{x}_0$  che giace nei pressi della superficie si costruisce un cubo che la interseca. La valutazione di  $F$  in tutti i vertici del cubo permette di identificare le facce del cubo che hanno punti in comune con la superficie. Su queste facce vengono costruiti cubi adiacenti al cubo di partenza, che, a loro volta, intersecano la superficie. Il procedimento è ripetuto fino a "ricoprire" tutta la superficie di cubi che la intersecano. Il punto di partenza  $\mathbf{x}_0$  può essere prescelto con tecniche di ricerca casuale se la posizione della superficie non è nota a priori; nel caso di superfici ricostruite per interpolazione esso può essere identificato con un qualsiasi elemento della nuvola di punti  $X$ ;
2. *decomposizione*: i cubi che intersecano la superficie sono suddivisi in tetraedri; i tetraedri che non intersecano la superficie sono scartati;
3. *poligonalizzazione*: per ogni tetraedro che interseca la superficie, si identificano le coppie di vertici attraverso cui avviene un cambio di segno della funzione  $F$ ; questo implica che la superficie intersechi gli spigoli che uniscono tali coppie di vertici, ossia che lungo ogni spigolo esiste un punto  $\mathbf{x}$  per cui  $F(\mathbf{x}) = 0$ . I punti di intersezione fra la superficie e gli spigoli sono ricavati mediante una ricerca binaria. Attenzione particolare viene impiegata per risolvere casi ambigui (ad esempio, i casi in cui una superficie interseca più volte lo stesso tetraedro).
4. *Altri metodi*: uno dei difetti principali degli algoritmi basati su marching cubes e marching tetrahedra è la cattiva ricostituzione

---

<sup>1</sup>Ricordiamo che data l'equazione  $f(x) = 0$  una funzione reale, continua, definita su un intervallo  $[a, b]$  e tale che  $f(a)f(b) < 0$  il **metodo della bisezione** prevede di valutare  $f((a+b)/2)$ ; se  $f((a+b)/2) = 0$  allora  $x = (a+b)/2$  risolve l'equazione; altrimenti si itera il procedimento di ricerca nell'intervallo fra  $[a, (a+b)/2]$  se  $f(a)f((a+b)/2) < 0$ , in  $[(a+b)/2, b]$  altrimenti.

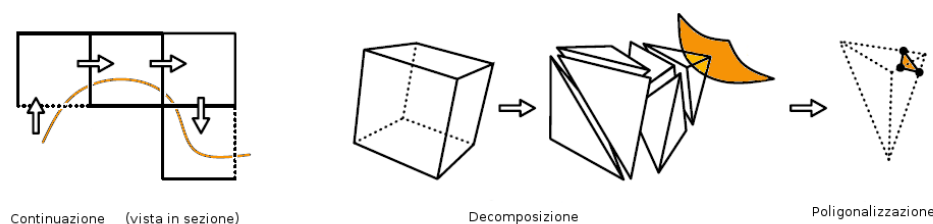


Figura 1.6: Passi principali dell'algoritmo di Bloomenthal (da [1]).

di spigoli, angoli, punte di oggetti (le cosiddette *sharp features*). Per questo motivo sono stati proposti in letteratura soluzioni che prevedono di individuare spigoli e angoli attraverso un'analisi dei vettori normali alla superficie e ricostruirli fedelmente, come l'algoritmo *extended marching cubes* (per ulteriori dettagli si veda [17]) o gli algoritmi basati sul *dual contouring* (per cui rimandiamo a [16]).

Altri algoritmi invece sono specializzati per il *rendering* di superfici con finalità esclusivamente visive. Una delle tecniche più diffuse in questa direzione è il *ray tracing*, che permette di riprodurre fedelmente una scena virtuale tracciando dei cammini dal punto dello spazio in cui si collocherebbe l'osservatore fino agli oggetti presenti nella scena, seguendo il percorso inverso dei raggi di luce che, nella scena reale, partirebbero dagli oggetti per arrivare all'occhio di chi guarda. Tale tecnica permette di ottenere livelli sorprendenti di realismo; alcune implementazioni sono specializzate per il rendering di oggetti descritti attraverso superfici implicite (ad esempio, quella descritta in [10]).

## 1.6 Ricostruzione di superfici: stato dell'arte

La messa a punto di algoritmi per la ricostruzione di superfici a partire da punti campionati costituisce un'area di ricerca molto attiva. In particolare, un impulso in questo senso è fornito dal divario attuale fra le performances dei dispositivi di acquisizione, in grado di fornire insiemi da centinaia di milioni di punti da processare, e l'efficienza degli algoritmi oggi comunemente impiegati, con cui si riescono a ricostruire insiemi fino a qualche milione di punti.

Alcuni autori (in [28] e in [7]) hanno proposto metodi per ricavare la funzione interpolante  $F$  in forma algebrica (ossia  $F$  è una funzione polinomiale delle coordinate dei punti); tuttavia tali metodi richiedono forti vincoli topologici sulla superficie da ricostruire. È possibile descrivere superfici complesse facendo uso di funzioni algebriche definite a pezzi saldate insieme, ma

accorgimenti particolari devono essere adottati per saldare le funzioni al fine di conservare la regolarità globale della funzione.

L'uso di superfici implicite per la gestione di oggetti geometrici si è rivelato uno strumento versatile ed efficace ed è tuttora una delle direzioni di ricerca più attive nella letteratura scientifica. Un'importante classe di funzioni implicite, è rappresentata dalle cosiddette *blobby surfaces* (o *meta-ball surfaces*), ovvero funzioni  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  della forma

$$F(\mathbf{x}) = -t + \sum_{i=1}^n g_i(\mathbf{x}),$$

dove le funzioni  $g_i$  sono le gaussiane  $g_i(\mathbf{x}) = e^{|\mathbf{x}-\mathbf{c}_i|^2/\sigma^2}$ , dove  $\mathbf{c}_i \in \mathbb{R}^3$  sono detti *centri* e  $\sigma$  è un parametro reale che controlla l'ampiezza del profilo gaussiano. Al variare del parametro reale  $t$  la funzione  $F$  descrive una famiglia di superfici nidificate attorno a una sovrapposizione di "blobby spheres". In [22] si propone un modello adattativo basato su tali funzioni per descrivere oggetti geometrici.

Alcuni autori hanno poi introdotto tecniche di tipo Level Set e distanze discretizzate. Per esempio, in [35] per determinare una superficie implicita a partire da una nuvola di punti  $X$  si costruisce dapprima una funzione distanza dai punti della nuvola definendola esclusivamente su punti appartenenti a una griglia cartesiana  $\mathcal{G}$ :

$$\tilde{d} : \mathcal{G} \rightarrow \mathbb{R}^+ \quad : \tilde{d} = \tilde{d}(\mathbf{x}, X);$$

in seguito si partizionano i punti della griglia  $\mathcal{G}$  distinguendo quelli che si trovano "fuori" dalla superficie, "dentro" la superficie e "in prossimità" della superficie. Infine, si ricava una funzione

$$d : \mathbb{R}^d \rightarrow \mathbb{R}^+ \quad : d = d(\mathbf{x})$$

attraverso metodi di tipo Level Set, il cui insieme di livello zero definisce implicitamente la superficie.

Una delle direzioni di ricerca più promettenti è costituita dall'uso di funzioni di base a simmetria radiale (le cosiddette *radial basis functions*, RBF nel seguito), che stanno assumendo un'importanza crescente anche in ambiti diversi dall'interpolazione di funzioni, ad esempio la risoluzione di equazioni differenziali. Le tecniche presentate in questo lavoro fanno uso di questo tipo di funzioni; nel Capitolo 2 è presentata una descrizione più dettagliata delle proprietà di tali funzioni e di come sia possibile impiegarle per ricostruire una superficie mediante interpolazione polinomiale. Lo stato dell'arte dell'interpolazione funzionale mediante RBF è presentato nel Paragrafo ??.

Nell'Appendice B, infine, presentiamo una classe di metodi che fanno uso di una partizione dell'unità: si tratta di tecniche impiegate da tempo nella

risoluzione di equazioni alle derivate parziali che sono state recentemente applicate con buoni risultati alla ricostruzione di superfici (ad esempio, in [23] e [6]). Tecniche basate sulla partizione dell'unità possono anche essere vantaggiosamente accoppiate con tecniche basate su RBF per creare metodi misti (come proposto in [29] e in [30]).

# 2

## Ricostruzione con funzioni a simmetria radiale

In questo capitolo presentiamo l'approccio basato su funzioni a simmetria radiale per l'interpolazione multivariata. Il teorema di Mairhuber–Curtis riportato in Sezione 2.1 mostra che, sorprendentemente, l'interpolazione multivariata è un problema molto più complesso di quello ben noto dell'interpolazione univariata e quindi la sua risoluzione richiede strumenti specifici. L'approccio basato su funzioni a simmetria radiale (RBF) si è rivelato uno strumento potente ed efficace. Questo metodo, di semplice implementazione ma dalle profonde radici teoriche, garantisce la buona posizione del problema di interpolazione multivariata (come illustrato nelle Sezioni 2.3 e 2.5) e l'ottimalità della soluzione trovata, secondo la definizione riportata in Sezione 2.6. La scelta della funzione di base è un aspetto cruciale del metodo RBF: in particolare si distinguono due famiglie di RBF, quelle a supporto globale e quelle a supporto compatto, presentate entrambe in Sezione 2.4. Forniamo in questo capitolo anche un importante risultato, il principio di indeterminazione, che mette in evidenza i limiti dell'approccio RBF e presenta affascinanti analogie con la fisica delle particelle.

### 2.1 Interpolazione di funzioni multivariate

Come anticipato nel Capitolo 1, il problema della ricostruzione di superfici può essere formulato come un problema di interpolazione multivariata:

$$(P) \begin{cases} \text{dato } X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}, \text{ con } \mathbf{x}_j \in \mathbb{R}^d; \\ \text{dati } f_j = f(\mathbf{x}_j), \quad 1 \leq j \leq N; \\ \text{trovare una funzione continua } F : \mathbb{R}^d \rightarrow \mathbb{R} \text{ tale che:} \\ \quad F(x_j) = f_j, \quad 1 \leq j \leq N. \end{cases}$$

Se si suppone che la superficie sia definita implicitamente dall'insieme di livello 0 della funzione  $F$ , affinché i punti rilevati giacciano sulla superficie

si deve imporre  $f_j = 0$ , con  $j = 1 \dots N$ . Un approccio classico prevede di definire uno spazio lineare  $n$ -dimensionale  $\mathcal{V}$  di funzioni continue generato da una base  $\{v_j\}_{1 \leq j \leq n}$  e di trovare una funzione interpolante della forma

$$F(\mathbf{x}) = \sum_{j=1}^n \lambda_j v_j(\mathbf{x}).$$

I vincoli di interpolazione permettono di ricavare immediatamente i coefficienti  $\lambda_j$  incogniti risolvendo il sistema lineare

$$\sum_{j=1}^n \lambda_j v_j(\mathbf{x}_i) = f_i \quad \forall i = 1 \dots N. \quad (2.1)$$

Come vedremo, il risultato riportato nel prossimo paragrafo stabilisce che non è tuttavia possibile trovare uno spazio lineare che contenga funzioni interpolanti per una collezione arbitraria di punti  $\{\mathbf{x}_i\}_{i=1 \dots N}$ .

### 2.1.1 Teorema di Mairhuber–Curtis

Nel Capitolo 1 si è accenato al fatto che gli strumenti abitualmente utilizzati per l'interpolazione di funzioni unidimensionali sono inadatti ad affrontare il caso multidimensionale. Il teorema di Mairhuber–Curtis giustifica questa affermazione. Prima di enunciarlo, introduciamo una definizione e un risultato preliminare.

**Definizione 1.** *Sia  $\Omega \subset \mathbb{R}^d$  un insieme contenente almeno  $N$  punti distinti. Sia  $\mathcal{V} \subset C(\overline{\Omega})$  uno spazio lineare  $N$ -dimensionale. Allora  $\mathcal{V}$  è uno **spazio di Haar** di dimensione  $N$  su  $\Omega$  se e solo se per ogni collezione di punti distinti  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \Omega$  e valori arbitrari  $f_1, \dots, f_N \in \mathbb{R}$  esiste una e una sola funzione  $F \in \mathcal{V}$  tale che  $F(\mathbf{x}_j) = f_j$ ,  $1 \leq j \leq N$ .*

Ad esempio  $\mathbb{P}^{N-1}(\mathbb{R})$ , lo spazio dei polinomi in una variabile reale di grado al più  $N - 1$ , è uno spazio di Haar per ogni  $\Omega \subset \mathbb{R}$  che contenga almeno  $N$  punti. Esistono numerose caratterizzazioni degli spazi di Haar; il seguente teorema ne riporta alcune (per la dimostrazione si veda, ad esempio, [34]).

**Teorema 1.** *Nelle ipotesi della Definizione 1, le seguenti affermazioni sono equivalenti:*

1.  $\mathcal{V}$  è uno spazio di Haar;
2. ogni  $v \in \mathcal{V} \setminus \{0\}$  ha al più  $N - 1$  zeri;
3. per ogni insieme di punti distinti  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \Omega$  e ogni base  $\{v_j\}_{1 \leq j \leq N}$  di  $\mathcal{V}$ , detta  $A \in \mathbb{R}^{N \times N}$  la matrice di coefficienti  $a_{ij} = v_j(\mathbf{x}_i)$  si ha che:

$$\det(A) \neq 0.$$



Il risultato seguente stabilisce che, salvo casi banali, non è possibile costruire uno spazio di Haar quando  $d \geq 2$ .

**Teorema 2.** (*Mairhuber–Curtis*). *Sia  $\Omega \subset \mathbb{R}^d$  con  $d \geq 2$ , contenente almeno un punto interno. Allora non esiste uno spazio di Haar su  $\Omega$  di dimensione  $N \geq 2$ .*

*Dimostrazione.* Sia  $\mathcal{V} = \text{span}\{v_1, \dots, v_N\}$  uno spazio di Haar su  $\Omega$ . Dato che  $\Omega$  contiene un punto interno, esistono  $\mathbf{x}_0 \in \Omega$  e una palla  $B(\mathbf{x}_0, \delta) \subset \Omega$  di raggio  $\delta > 0$ ; è possibile quindi considerare dei punti distinti  $\mathbf{x}_3, \dots, \mathbf{x}_N \in B(\mathbf{x}_0, \delta)$ . Siano ora  $\mathbf{x}_1(t), \mathbf{x}_2(t)$ , per  $t \in [0, 1]$  due curve parametriche continue tali che  $\mathbf{x}_1(0) = \mathbf{x}_2(1)$ ,  $\mathbf{x}_2(0) = \mathbf{x}_1(1)$ ; supponiamo che tali curve non abbiano nè altri punti di intersezione nè alcun punto in comune con  $\{\mathbf{x}_3, \dots, \mathbf{x}_N\}$ . Tali curve esistono dato che  $d \geq 2$ . Allora dato che  $\mathcal{V}$  è uno spazio di Haar, se  $A$  è definita come nel Teorema 1, l'applicazione

$$D(t) = \det(A(t))$$

è continua e non cambia segno. D'altra parte però  $D(0) = -D(1)$  dato che le prime due righe di  $A$  si scambiano al variare del parametro e tutto il resto rimane invariato. Quindi  $D$  deve cambiare segno, il che contraddice quanto osservato poc'anzi sul segno di  $D$ .  $\square$

Il risultato precedente implica, ad esempio, che se  $d \geq 2$  non è possibile interpolare con un polinomio multivariato un insieme arbitrario di punti.

## 2.2 Interpolazione mediante funzioni di base radiali

L'approccio basato su funzioni di base radiali per la risoluzione del problema (P) prevede la ricerca di una funzione interpolante della forma:

$$F(\mathbf{x}) = \sum_{j=1}^N \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) \quad (2.2)$$

dove  $\|\cdot\|$  è la norma euclidea in  $\mathbb{R}^d$ ,  $\phi : \mathbb{R} \rightarrow [0, +\infty)$  è una *funzione radiale* scelta a priori e  $\lambda_j$  sono dei pesi reali. Osserviamo innanzitutto che un'interpolante di questo tipo non rientra nel quadro del teorema di Mairhuber–Curtis, in quanto la base dello spazio delle interpolanti  $\mathcal{V}$  dipende dai punti  $\{\mathbf{x}_j\}$  dove sono assegnati i vincoli di interpolazione per  $F$ . Tali vincoli portano al seguente sistema lineare:

$$\sum_{j=1}^N \lambda_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = f_i \quad \forall i = 1 \dots N. \quad (2.3)$$

La matrice di coefficienti  $A_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ , che indicheremo con  $A_{\phi, X}$  nel seguito, è detta matrice di interpolazione; la scelta della funzione radiale

$\phi$  determina le proprietà della matrice di interpolazione. Inoltre l'esistenza e l'unicità di una soluzione per il problema di interpolazione (2.3) possono essere provate per via algebrica, ovvero dimostrando l'invertibilità della matrice  $A_{\phi, X}$ . Per alcuni tipi di funzioni radiali è possibile garantire esistenza e unicità della soluzione a patto di scegliere la funzione interpolante della forma:

$$F(\mathbf{x}) = \sum_{i=1}^N \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + P(\mathbf{x}), \quad \text{con } P \in \mathbb{P}_m^d, \quad (2.4)$$

dove  $\mathbb{P}_m^d$  è lo spazio (lineare) dei polinomi in  $d$  variabili reali a valori reali di grado al più  $m - 1$ ; <sup>1</sup>  $m$  dipende dalla scelta della RBF  $\phi$  ed è detto *ordine* della funzione radiale  $\phi$ . La buona posizione del problema (P) risolto con interpolanti di tipo RBF verrà ulteriormente approfondito nella Sezione 2.3.

Osserviamo che uno dei vantaggi principali dell'approccio RBF per risolvere il problema (P) consiste nella sua natura di metodo *meshfree*: in particolare, non è richiesto alcun algoritmo di generazione di mesh poligonale attorno ai punti di  $X$  prima di effettuare l'approssimazione della superficie. Per insiemi di punti con una distribuzione spaziale densa in ogni parte della superficie il metodo risulta di immediata implementazione. Accorgimenti particolari sono impiegati per processare grandi insiemi di punti o insiemi di dati con distribuzione non uniforme. Inoltre, è possibile dimostrare che l'interpolazione mediante RBF garantisce che la funzione  $F$ , oltre a rispettare i vincoli di interpolazione, abbia una certa regolarità, soddisfacendo una condizione di *approssimazione ottima*; questo aspetto verrà dettagliato nella Sezione 2.6.

### 2.2.1 Dati supplementari

Le tecniche per l'interpolazione basate su RBF prevedono in generale di utilizzare, oltre alle coordinate di una serie di punti che giacciono sulla superficie incognita, anche alcune informazioni aggiuntive. Da un punto di vista algebrico tali informazioni servono a garantire che il sistema lineare di interpolazione (2.3) ammetta una soluzione non identicamente nulla (ricordiamo che, nell'ambito della ricostruzione di superfici, nel Problema (P) si pone  $f_j = 0$  per ogni  $1 \leq j \leq N$ ). Da un punto di vista geometrico tali condizioni permettono di determinare in modo consistente il segno della funzione interpolante affinché essa cambi di segno passando da una parte all'altra della superficie. Come messo in evidenza in [31], esistono diverse possibilità per scegliere quali dati aggiuntivi fornire.

Consideriamo il caso di una superficie chiusa (come quella che descrive un oggetto) nello spazio tridimensionale  $\mathbb{R}^3$ : vorremmo che la funzione interpolante  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  assumesse valori positivi all'interno della superficie e

---

<sup>1</sup>Per ogni  $\mathbf{x} = \{x_1, \dots, x_d\} \in \mathbb{R}^d$  e per ogni multi-indice  $\alpha = \{\alpha_1 \dots \alpha_d\} \in \mathbb{N}_0^d$ , definiamo  $\mathbf{x}^\alpha = x_1^{\alpha_1} + \dots + x_d^{\alpha_d}$  con  $|\alpha| = \alpha_1 + \dots + \alpha_d$ .

negativi fuori da essa (osserviamo che questa scelta è del tutto arbitraria). Una prima possibilità è quella di aggiungere dei punti addizionali alla nuvola originaria. Ricordiamo che, nell'approccio RBF, essendo la superficie ricostruita come insieme di livello zero di  $F$  si impone che  $F|_X = 0$ . È possibile aggiungere dei vincoli per punti che si trovano all'interno o all'esterno della superficie, imponendo che in questi punti (definiti arbitrariamente)  $F$  assuma un valore rispettivamente positivo o negativo; indicheremo questi vincoli aggiuntivi come *vincoli positivi* e *vincoli negativi*. Come osservato in [31], grazie alla natura variazionale dell'interpolazione RBF, è sufficiente aggiungere vincoli negativi all'esterno della superficie affinché la funzione interpolante assuma automaticamente valori positivi dentro di essa e viceversa. Questa scelta comporta un aumento della taglia del sistema lineare associato ai soli vincoli di interpolazione (2.3); inoltre, non sempre è chiaro dove posizionare i punti senza avere una conoscenza iniziale della forma dell'oggetto da ricostruire.

Un'alternativa è data dall'uso dei vettori normali alla superficie nei punti della nuvola. Il vantaggio principale è che questa informazione è spesso disponibile o facilmente ottenibile. Nel caso di scansioni di oggetti essa viene automaticamente misurata da molti tipi di apparecchi di acquisizione durante la scansione. Nel caso in cui si effettui la ricostruzione di superfici a partire da mesh poligonali, questo dato può essere invece facilmente calcolato (ove non sia già memorizzato fra gli attributi della mesh). Infine, nel caso in cui si abbiano a disposizione esclusivamente i punti della nuvola  $X$  è possibile stimare i vettori normali effettuando, per ogni punto  $\mathbf{x}_i \in X$ , un'analisi spettrale della *matrice di covarianza* associata ai  $K$  punti della nuvola più vicini a  $\mathbf{x}_i$ ; quest'ultima è rappresentata dalla matrice

$$C(\mathbf{x}_i) = \sum_{k=1}^K \begin{bmatrix} (\mathbf{d}_i^k)_x^2 & (\mathbf{d}_i^k)_x(\mathbf{d}_i^k)_y & (\mathbf{d}_i^k)_x(\mathbf{d}_i^k)_z \\ (\mathbf{d}_i^k)_y(\mathbf{d}_i^k)_x & (\mathbf{d}_i^k)_y^2 & (\mathbf{d}_i^k)_y(\mathbf{d}_i^k)_z \\ (\mathbf{d}_i^k)_z(\mathbf{d}_i^k)_x & (\mathbf{d}_i^k)_z(\mathbf{d}_i^k)_y & (\mathbf{d}_i^k)_z^2 \end{bmatrix},$$

dove  $(\mathbf{d}_i^k)_x$ ,  $(\mathbf{d}_i^k)_y$ ,  $(\mathbf{d}_i^k)_z$  rappresentano le 3 componenti del vettore  $(\mathbf{d}_i^k) = \mathbf{x}_k - \mathbf{x}_i$ . Osserviamo che questo procedimento necessita un'analisi successiva per verificare che i vettori normali così ottenuti siano orientati in modo consistente. Per ulteriori dettagli si veda [14].

L'informazione sui vettori normali può essere utilizzata per aggiungere vincoli positivi o negativi, ma offre anche innumerevoli altre possibilità, come vedremo nel Capitolo 3.

## 2.3 Buona posizione del problema di interpolazione RBF

Distinguiamo, nell'analisi della buona posizione del problema dell'interpolazione RBF, il caso in cui si impieghi una funzione di base radiale di ordine

$m = 0$  (ovvero un'approssimante della forma (2.2)) da quello in cui  $m > 0$  (in questo caso  $F(\mathbf{x})$  deve essere scelta della forma (2.4) per ragioni che saranno chiare alla fine di questa sezione).

- **Caso  $m = 0$ .** Introduciamo la seguente definizione.

**Definizione 2.** Una funzione radiale continua  $\phi : [0, \infty) \rightarrow \mathbb{R}$  è detta **definita positiva** su  $\mathbb{R}^d$  se, per ogni collezione finita di punti  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$  distinti a due a due, la matrice

$$A_{\phi, X} \in \mathbb{R}^{N \times N} \text{ con } A_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad 1 \leq i, j \leq N$$

è definita positiva.

Cercando un'interpolante della forma (2.2) e imponendo gli  $N$  vincoli di interpolazione (2.3) si arriva al sistema lineare rappresentato dalla stessa (2.3). La definizione precedente porta quindi direttamente al seguente risultato:

**Teorema 3.** Se la funzione  $\phi(\cdot)$  è definita positiva, il problema (P) ammette una e una sola soluzione della forma (2.2).

- **Caso  $m > 0$ .** La presenza di un polinomio nella definizione della funzione interpolante (2.4) comporta la presenza di  $q$  gradi di libertà aggiuntivi, dove  $q$  è dato da  $q = \binom{m-1+d}{d}$  (per la dimostrazione di questo risultato si veda, ad esempio, [34]). Questi gradi di libertà vengono solitamente determinati richiedendo che siano soddisfatte le seguenti  $q$  condizioni:

$$\sum_{j=1}^N \lambda_j P(\mathbf{x}_j) = 0 \quad \forall P \in \mathbb{P}_m^d. \quad (2.5)$$

L'aggiunta di queste condizioni porta alla risoluzione del nuovo sistema lineare

$$\begin{bmatrix} A_{\phi, X} & P_X \\ P_X^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\Lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \quad (2.6)$$

dove la matrice  $P_X = \{(\mathbf{x}_j)^\alpha\}_{1 \leq j \leq N; |\alpha| < m}$  ha dimensioni  $N \times q$  e il vettore  $\mathbf{f} = \{f_j\}_{1 \leq j \leq N}$  contiene i valori da interpolare; le incognite sono costituite da  $\boldsymbol{\Lambda} = \{\lambda_j\}_{1 \leq j \leq N}$  e da  $\mathbf{c} = \{c_i\}_{1 \leq i \leq q}$ ; quest'ultimo è il vettore contenente i coefficienti del polinomio  $P$ . Per analizzare esistenza e unicità di una soluzione del sistema (2.6), verifichiamo sotto quali ipotesi il corrispondente sistema omogeneo ammette solo la soluzione nulla. Il sistema omogeneo è dato da:

$$A_{\phi, X} \boldsymbol{\Lambda} + P_X \mathbf{c} = \mathbf{0} \quad (2.7)$$

$$P_X \boldsymbol{\Lambda} = \mathbf{0}. \quad (2.8)$$

Moltiplicando l'equazione (2.7) per  $\mathbf{\Lambda}^T$  e utilizzando la (2.8), si ottiene:

$$\mathbf{\Lambda}^T A_{\phi, X} \mathbf{\Lambda} = 0. \quad (2.9)$$

Introduciamo la seguente definizione.

**Definizione 3.** Una funzione radiale continua  $\phi : [0, +\infty) \rightarrow \mathbb{R}$  è detta **condizionatamente definita positiva** rispetto a  $m$  su  $\mathbb{R}^d$  se

$$\mathbf{\Lambda}^T A_{\phi, X} \mathbf{\Lambda} > 0 \quad (2.10)$$

per ogni  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$  insieme di punti distinti e per ogni  $\mathbf{\Lambda} \in \mathbb{R}^N \setminus \{0\}$  tali che  $P_X \mathbf{\Lambda} = 0$ .

Osserviamo che la condizione (2.10) dipende, grazie alla definizione di  $P_X$ , dall'ordine  $m$  dello spazio di polinomi  $\mathbb{P}_m^d$ . Per  $m = 0$  ritroviamo la Definizione 2. Inoltre, per ogni  $m_1, m_2 \in \mathbb{N}_0$  con  $m_1 < m_2$ , se  $\phi$  è condizionatamente definita positiva rispetto a  $m_1$  lo è anche rispetto a  $m_2$ . Per una qualsiasi funzione condizionatamente definita positiva definiamo dunque l'*ordine* di  $\phi$  come il più piccolo intero  $m$  che rende vera la condizione (2.10).

Tornando alla soluzione del sistema omogeneo (2.7)-(2.8), se  $\phi$  (o  $-\phi$ ) è condizionatamente definita positiva di ordine  $m$ , l'equazione (2.9) ha come unica soluzione  $\mathbf{\Lambda} = \mathbf{0}$ . Per concludere che anche  $\mathbf{c} = 0$  basta provare che  $P_X$  è iniettiva. Questa proprietà dipende in realtà dalla disposizione nello spazio dei punti dell'insieme  $X$ . Introduciamo la seguente definizione.

**Definizione 4.** L'insieme  $X = \{\mathbf{x}_1 \dots \mathbf{x}_N\}$ , con  $N \geq q = \dim(\mathbb{P}_m^d)$  è detto  $\mathbb{P}_m^d$ -**unisolvante** se

$$\forall P \in \mathbb{P}_m^d : P(\mathbf{x}_1) = 0, \dots, P(\mathbf{x}_N) = 0 \Rightarrow P \equiv 0.$$

La richiesta di  $\mathbb{P}_m^d$ -unisolvanza è piuttosto debole se  $m$  è piccolo: per  $m = 0$  è impossibile; per  $m = 1$  è triviale; per  $m = 2$  è sufficiente che i punti non giacciono tutti sulla stessa linea. Le osservazioni precedenti portano al risultato seguente.

**Teorema 4.** Se  $\phi$  è condizionatamente definita positiva di ordine  $m$  e l'insieme  $X$  è  $\mathbb{P}_m^d$ -unisolvante allora esiste una e una sola funzione interpolante della forma (2.4) soluzione del problema (P) che soddisfa le condizioni (2.5).

## 2.4 Scelta della funzione radiale

La scelta della funzione radiale è uno dei fattori più importanti per risolvere efficacemente il problema dell'interpolazione multivariata. È possibile classificare le RBF in due famiglie, quelle *a supporto globale* e quelle *a supporto compatto*.

Il vantaggio principale delle RBF a supporto globale è che si rivelano più robuste rispetto a variazioni della distribuzione spaziale dei punti da interpolare; tuttavia questo approccio è poco efficiente quando la dimensione  $N$  dell'insieme di punti da interpolare cresce, per due ragioni: il calcolo dei coefficienti  $\lambda_j$  è laborioso dato che la matrice di interpolazione è piena; la valutazione della funzione  $F$  in un punto generico  $\mathbf{x}$  richiede la somma di  $N$  termini. La Tabella 2.1 (da [15]) riassume alcune famiglie di RBF a supporto globale con i parametri che le caratterizzano e il loro ordine (ricordiamo che per un numero reale  $\nu$  il simbolo  $\lceil \nu \rceil$  rappresenta il più piccolo intero  $n$  tale che  $n \geq \nu$ ).

| RBF                             | $\phi(r)$                                 | Parametri   | Ordine (m)                       |
|---------------------------------|---|---|----------------------------------|
| <b>Splines poliarmoniche</b>    | $r^\nu,$<br>$r^{2k} \log(r)$              | $\nu > 0, \nu \notin 2\mathbb{N}$<br>$k \in \mathbb{N}$ | $\lceil \nu/2 \rceil$<br>$k + 1$ |
| <b>Gaussiane</b>                | $\exp\left(-\frac{r^2}{2\sigma^2}\right)$ | $\sigma > 0$  | 0                                |
| <b>Multiquadratiche</b>         | $(1 + r^2)^\nu$                           | $\nu > 0, \nu \notin \mathbb{N}$                        | $\lceil \nu \rceil$              |
| <b>Multiquadratiche inverse</b> | $(1 + r^2)^\nu$                           | $\nu < 0$   | 0                                |

Tabella 2.1: Alcune RBF a supporto globale.

Sono numerosi in letteratura i metodi proposti che sfruttano RBF a supporto globale. Ad esempio, in [31] si fa uso di una funzione approssimante costruita con RBF a supporto globale, calcolata minimizzando la curvatura della superficie implicita (questa tecnica è chiamata *Thin plate interpolation*); fra l'altro, l'algoritmo proposto permette di ottenere modelli modificabili interattivamente. Gli stessi autori presentano in [32] tecniche ed esempi di manipolazione di superfici implicite ottenute con questo approccio

che ne mostrano la grande versatilità.

Lo stato dell'arte fra questa categoria di metodi sembra, comunque, essere rappresentato dagli algoritmi descritti in [4], in cui si effettua una semplificazione dell'insieme dei centri di interpolazione basata su una strategia *greedy* e si introducono tecniche efficaci per la visualizzazione (*fast multipole methods*). L'idea di questo lavoro è stata sfruttata per la realizzazione del software commerciale FASTRBF © (<http://www.farfieldtechnology.com/>). Altri autori hanno sviluppato metodi ad hoc per rendere più efficiente la risoluzione del sistema lineare (2.3) per il calcolo dei coefficienti  $\lambda_j$  (si veda, ad esempio, [8]).

Un'allettante alternativa alle RBF a supporto globale è rappresentata dall'uso di funzioni radiali a supporto compatto. Esse permettono di ottenere una matrice di interpolazione sparsa e di accelerare notevolmente la valutazione della funzione  $F$  in un punto, dato che, per un punto  $\mathbf{x}$  fissato, solo pochi termini nella sommatoria (2.2) (o (2.4)) risultano non nulli. In particolare faremo riferimento alla famiglia di RBF a supporto compatto proposta da H.Wendland in [33]; si tratta di una famiglia di funzioni della forma

$$\phi_{d,k}(r) = \begin{cases} p_{d,k}(r) & \text{per } 0 \leq r \leq 1 \\ 0 & \text{altrimenti,} \end{cases} \quad (2.11)$$

dove  $p_{d,k}$  è un polinomio univariato di grado al più  $\lfloor d/2 \rfloor + 3k + 1$ . È possibile dimostrare che, per un numero  $d$  di dimensioni assegnato, esiste sempre una funzione radiale definita positiva della forma (2.11). Alcuni esempi di tali funzioni sono riportate in Tabella 2.2; esse hanno la peculiarità di essere di grado minimo fra tutte le funzioni definite positive di classe  $\mathcal{C}^{2k}(\mathbb{R}^d)$  e della forma (2.11). Inoltre, tali funzioni sono uniche (a meno di una costante) rispetto a queste proprietà. (Per la dimostrazione e per la procedura con cui vengono costruite tali funzioni si rimanda a [33].)

Spesso in letteratura quando si utilizzano RBF a supporto compatto come quelle in Tabella 2.2 viene presa una versione scalata della funzione radiale stessa,  $\phi_\sigma = \phi(\cdot/\sigma)$ , al fine di adattare il supporto alle esigenze dell'interpolazione. In questi casi la determinazione del parametro  $\sigma$  diventa uno degli aspetti cruciali del metodo.

Le RBF a supporto compatto godono della proprietà di *località*, la quale implica che una perturbazione locale dei dati non si propaghi a livello globale sulla funzione interpolante. D'altra parte, la loro stessa natura locale le rende poco adatte per la ricostruzione di dati incompleti. Per un'interessante confronto fra le prestazioni di RBF a supporto globale e a supporto compatto e fra le corrispondenti complessità temporali per tutte le fasi della ricostruzione rimandiamo a [21].

Per superare le limitazioni derivanti dall'uso di un tipo particolare di RBF, è possibile mettere a punto algoritmi che ampliano le potenzialità dell'interpolazione RBF "classica". Gli algoritmi presentati nel Capitolo 3,

come vedremo, hanno lo scopo di sfruttare la velocità di calcolo derivante dall'uso di RBF a supporto compatto e allo stesso tempo di superarne le limitazioni in termini di robustezza e ricostruzione di dati incompleti.

## 2.5 Caratterizzazione di funzioni definite positive e condizionatamente definite positive

Le osservazioni effettuate nel Paragrafo 2.3 mettono in evidenza la necessità di criteri sufficienti per garantire che una data funzione radiale sia (condizionatamente) definita positiva. In questa sezione enunciamo due criteri in grado di garantire tale proprietà. L'esposizione segue quella presentata in [15], a cui rimandiamo per ulteriori dettagli. Osserviamo inoltre che tali criteri possono essere impiegati anche per la costruzione di nuove funzioni di base (come descritto, ad esempio, in [33]). Più nello specifico, il primo criterio che enunciamo si basa sulla definizione di funzioni completamente monotone; sfortunatamente questo criterio non è applicabile a qualsiasi funzione radiale: in particolare esso non riguarda quelle a supporto compatto. Il secondo criterio invece fa uso della trasformata di Fourier generalizzata.

### 2.5.1 Primo criterio

Introduciamo la seguente:

**Definizione 5.** *Una funzione reale  $\psi \in C^\infty([0, +\infty))$  si dice **completamente monotona** se e solo se*

$$(-1)^l \psi^{(l)}(r) \geq 0 \quad \forall l \in \mathbb{N} \text{ e } \forall r \in [0, +\infty). \quad (2.12)$$

È possibile dimostrare il seguente risultato, che si applica solo a funzioni radiali.

**Teorema 5.** *Sia  $\phi : [0, +\infty) \rightarrow \mathbb{R}$  una funzione radiale continua. Sia  $\phi_\sqrt{\cdot} \equiv \phi(\sqrt{\cdot})$ . Se  $\phi_m \equiv (-1)^m \phi_\sqrt{\cdot}^{(m)}$  è ben definita e non è costante, allora le seguenti affermazioni sono equivalenti:*

1.  $\phi$  è condizionatamente definita positiva di ordine  $m$  per ogni  $d \geq 1$ .
2.  $\phi_m$  è completamente monotona su  $[0, +\infty)$ .

È possibile applicare il Teorema 5 a ogni funzione riportata in Tabella 2.1 con  $m \geq 1$  e dimostrare così che  $\phi$  (o  $-\phi$ ) è condizionatamente definita positiva. Tuttavia, come anticipato, questo risultato non si applica alle funzioni radiali a supporto compatto. Infatti il teorema di Bernstein-Widder afferma che una funzione  $\psi : [0, +\infty) \rightarrow \mathbb{R}$  è completamente monotona su  $[0, +\infty)$  se e solo se ammette una rappresentazione della forma:

$$\psi(r) = \int_0^{+\infty} \exp(-rs) d\mu(s),$$



con  $\mu$  funzione monotona decrescente tale che  $\int_0^\infty d\mu(s) < \infty$  (rappresentazione di Laplace-Stieltjes). Questo implica che  $\psi$  non può avere zeri e, in particolare, nel Teorema 5  $\phi_m$  non può essere a supporto compatto.

### 2.5.2 Secondo criterio

Un secondo criterio per la caratterizzazione e la costruzione di funzioni condizionatamente definite positive si basa sulla nozione di trasformata generalizzata di Fourier, per la cui definizione rimandiamo a [15]. Ci limitiamo a osservare che, se una funzione radiale  $\phi$  ammette trasformata di Fourier  $\tilde{\phi} \in L^1(\mathbb{R}^d)$  data da

$$\tilde{\phi}(\|\omega\|) = \int_{\mathbb{R}^d} \phi(\|\mathbf{x}\|) \exp(-i\mathbf{x}^T \omega) d\mathbf{x}, \quad (2.13)$$

questa nozione classica di trasformata di Fourier coincide con quella di trasformata di Fourier generalizzata. È il caso, ad esempio, delle funzioni gaussiane, di quelle multiquadratiche inverse in Tabella 2.1 e delle RBF a supporto compatto riportate in Tabella 2.2.

**Ipotesi 1.** *Nel seguito diremo che una funzione radiale continua  $\phi : [0, +\infty) \rightarrow \mathbb{R}$  rispetta le Ipotesi 1 se ammette una trasformata di Fourier generalizzata radiale  $\tilde{\phi} \in \mathcal{C}((0, +\infty))$  tale che:*

1.  $\tilde{\phi}(\|\cdot\|)$  è  $L^1$ -integrabile in un intorno di  $+\infty$ , ossia

$$\int_{\mathbb{R}^d \setminus \mathcal{B}_1(0)} |\tilde{\phi}(\|\omega\|)| d\omega < +\infty;$$

2.  $\tilde{\phi}(\|\cdot\|)$  possiede al più una singolarità di ordine  $s_0 \in \mathbb{N}_0$  nell'origine, ossia è tale che

$$\int_{\mathcal{B}_1(0)} \|\omega\|^s \tilde{\phi}(\|\omega\|) d\omega < +\infty;$$

per ogni  $s \geq s_0$ .

Per funzioni  $\phi$  che rispettano le Ipotesi 1 è possibile dimostrare la seguente identità:

$$\sum_{j,k=1}^N c_j c_k \phi(\|\mathbf{x}_j - \mathbf{x}_k\|) = (2\pi)^{-d} \int_{\mathbb{R}^d} \tilde{\phi}(\|\omega\|) \left| \sum_{j=1}^N c_j \exp(-i\mathbf{x}_j^T \omega) \right|^2 d\omega, \quad (2.14)$$

che può essere dimostrata sotto l'ipotesi che la funzione

$$\sigma_{c,X}(\omega) = \sum_{j=1}^N c_j \exp(-i\mathbf{x}_j^T \omega) \quad (2.15)$$

abbia uno zero nell'origine di ordine  $m \geq \lceil s_0/2 \rceil$ . Quest'ultima può essere garantita, fra l'altro, dalla condizione (2.5).

**Teorema 6.** Una funzione radiale continua  $\phi : [0, +\infty) \rightarrow \mathbb{R}$  è condizionatamente definita positiva su  $\mathbb{R}^d$  se ha una trasformata generalizzata di Fourier  $\tilde{\phi}$  non identicamente nulla, continua, non negativa e che soddisfi le Ipotesi 1. In questo caso, l'ordine di  $\phi$  è  $m = \lceil s_0/2 \rceil$ .

*Dimostrazione.* Supponiamo che  $\tilde{\phi}$  rispetti le Ipotesi 1 e che valgano le condizioni (2.5) con  $m = \lceil s_0/2 \rceil$ , così che valga l'identità (2.14). Grazie al fatto che  $\tilde{\phi}$  è non negativa, la forma quadratica:

$$\mathbf{c}^T \cdot A_{X,\phi} \cdot \mathbf{c} = \sum_{j,k=1}^N c_j c_k \phi(\|\mathbf{x}_j - \mathbf{x}_k\|)$$

è non negativa. Resta da mostrare che essa si annulla se e solo se  $\mathbf{c} = \mathbf{0}$ . Supponiamo che  $\mathbf{c}^T \cdot A_{X,\phi} \cdot \mathbf{c} = 0$ . Allora la funzione  $\sigma_{c,X}(\omega)$  in (2.15) deve annullarsi su un aperto di misura positiva su  $\mathbb{R}^d$ ; ma allora, dato che  $\sigma_{c,X}(\omega)$  è analitica, questo implica che  $\sigma_{c,X}(\omega) \equiv 0$ . Ma ciò può essere verificato se e solo se  $\mathbf{c} = 0$ , visto che i punti  $\mathbf{x}_j$  sono distinti e gli esponenziali  $\exp(-i\mathbf{x}_j^T \omega)$  sono linearmente indipendenti.  $\square$

## 2.6 Ottimalità dell'interpolazione RBF

In questa sezione presentiamo un risultato di ottimalità per l'interpolazione multivariata mediante RBF. Siano  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$  e  $f_1, \dots, f_N$  rispettivamente l'insieme dei punti e i valori da interpolare; cerchiamo una funzione interpolante della forma (2.4). È possibile dimostrare che, se  $\phi$  è condizionatamente definita positiva per qualche intero  $m \geq 0$ , è possibile costruire uno spazio funzionale nativo  $\mathcal{N}_\Phi$  dotato di una semi-norma  $|\cdot|_\phi$  in modo che lo schema di interpolazione (2.4) sia *ottimale* su  $\mathcal{N}_\Phi$ , nel senso specificato nel Paragrafo 2.6.2. In particolare, per qualsiasi funzione  $f \in \mathcal{N}_\Phi$  tale che  $f(\mathbf{x}_j) = f_j$  con  $j = 1, \dots, N$ , l'interpolante  $F = F_{f,X}$  della forma (2.4) (che è unica, come osservato nella Sezione 2.3) appartiene essa stessa allo spazio nativo  $\mathcal{N}_\Phi$  e soddisfa la seguente disuguaglianza:

$$|F_{f,X}|_\phi \leq |f|_\phi \quad \forall f \in \mathcal{N}_\Phi. \quad (2.16)$$

In altre parole, l'interpolante  $F_{f,X}$  minimizza l'energia  $|\cdot|_\phi$  fra tutte le interpolanti  $g \in \mathcal{N}_\Phi$  tali che  $g|_X \equiv f|_X$ . Nella prossima sezione presentiamo la caratterizzazione dello spazio nativo  $\mathcal{N}_\Phi$  per funzioni di base radiali definite positive, che sono quelle che verranno utilizzate negli algoritmi presentati nel Capitolo 3. Una caratterizzazione del tutto analoga può essere effettuata per funzioni condizionatamente definite positive; si rimanda a [34] per ulteriori dettagli.

### 2.6.1 Spazio nativo

Nel seguito consideriamo spazi vettoriali di funzioni  $f : \Omega \rightarrow \mathbb{R}$ , in cui  $\Omega \subseteq \mathbb{R}^d$  è una regione arbitraria di  $\mathbb{R}^d$  non vuota. Introduciamo la definizione di reproducing kernel.

**Definizione 6.** Sia  $\mathcal{F}$  uno spazio di Hilbert di funzioni a valori reali  $f : \Omega \rightarrow \mathbb{R}$  con prodotto scalare  $(\cdot, \cdot)_{\mathcal{F}}$ . Una funzione  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$  è un reproducing kernel per  $\mathcal{F}$  se:

1.  $\Phi(\cdot, \mathbf{y}) \in \mathcal{F}$  per ogni  $\mathbf{y} \in \Omega$ ;
2.  $f(\mathbf{y}) = (f, \Phi(\cdot, \mathbf{y}))_{\mathcal{F}}$  per ogni  $f \in \mathcal{F}$  e per ogni  $\mathbf{y} \in \Omega$ .

È possibile dimostrare che il reproducing kernel di uno spazio di Hilbert è univocamente determinato ([34]). Osserviamo un fatto interessante. Dato un gruppo  $\mathcal{T}$  di trasformazioni  $T : \Omega \rightarrow \Omega$ , lo spazio  $\mathcal{F}$  si dice *invariante* rispetto al gruppo  $\mathcal{T}$  se, per  $g, f \in \mathcal{F}$  e per  $T \in \mathcal{T}$ , si ha che  $f \circ T \in \mathcal{F}$  e  $(f \circ T, g \circ T)_{\mathcal{F}} = (f, g)_{\mathcal{F}}$ . Si può dimostrare che, se  $\Omega = \mathbb{R}^d$  e lo spazio  $\mathcal{F}$  è invariante per traslazioni, allora il suo reproducing kernel è invariante per traslazioni; se  $\mathcal{F}$  è invariante rispetto a traslazioni e a trasformazioni ortogonali allora il suo reproducing kernel è radiale, ossia  $\Phi(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$ . Osserviamo che, in questo contesto, è opportuno interpretare le funzioni di base radiali come funzioni  $\Phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ , mentre altrove esse vengono descritte come funzioni reali di variabile reale  $\phi : [0, \infty) \rightarrow \mathbb{R}$ ; vale la semplice corrispondenza:  $\Phi(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$ .

Fissato uno spazio di Hilbert è possibile determinare il suo reproducing kernel; in realtà, per come abbiamo impostato il problema, in questo contesto è richiesto di operare il processo inverso, ossia, fissata una funzione simmetrica definita positiva  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$ , vorremmo capire se essa sia il reproducing kernel di uno spazio funzionale  $\mathcal{F}$ . A questo scopo, introduciamo il seguente spazio:

$$V_{\Phi}(\Omega) = \text{span}\{\Phi(\cdot, \mathbf{y}), \text{ con } \mathbf{y} \in \Omega\}, \quad (2.17)$$

a cui associamo la forma bilineare

$$\left( \sum_{j=1}^N \alpha_j \Phi(\cdot, \mathbf{x}_j), \sum_{k=1}^M \beta_k \Phi(\cdot, \mathbf{y}_k) \right)_{\Phi} := \sum_{j=1}^N \sum_{k=1}^M \alpha_j \beta_k \Phi(\mathbf{x}_j, \mathbf{y}_k). \quad (2.18)$$

**Teorema 7.** Se  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$  è simmetrica e definita positiva allora  $(\cdot, \cdot)_{\Phi}$  definisce un prodotto interno su  $F_{\Phi}(\Omega)$ ; inoltre,  $F_{\Phi}(\Omega)$  è uno spazio pre-hilbertiano con reproducing kernel  $\Phi$ .

La chiusura  $\mathcal{F}_{\Phi}(\Omega)$  di questo spazio pre-hilbertiano è il candidato ideale per lo spazio che stiamo cercando. È tuttavia necessario un passaggio

supplementare per poter interpretare  $\mathcal{F}_\Phi$  come uno spazio di funzioni. Introduciamo quindi la mappa verso lo spazio delle funzioni continue a valori reali definite su  $\Omega$

$$R : \mathcal{F}_\Phi(\Omega) \rightarrow \mathcal{C}(\Omega); \quad R(f)(\mathbf{x}) := (f, \Phi(\cdot, \mathbf{x}))_\Phi; \quad (2.19)$$

è facile mostrare che  $R(f)$  è una funzione continua, e che la mappa  $R$  è iniettiva da  $\mathcal{F}_\Phi(\Omega)$  a  $\mathcal{C}(\Omega)$ .

A questo punto possiamo introdurre la definizione di spazio nativo associato a  $\Phi$  ed enunciare il risultato che lo lega alle nozioni precedentemente introdotte per l'interpolazione RBF.

**Definizione 7.** *Lo spazio nativo associato a una funzione simmetrica definita positiva  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$  è dato da*

$$\mathcal{N}_\Phi(\Omega) := R(\mathcal{F}_\Phi(\Omega)). \quad (2.20)$$

*Ad esso è associato il prodotto interno:*

$$(f, g)_{\mathcal{N}_\Phi(\Omega)} = (R^{-1}f, R^{-1}g)_\Phi. \quad (2.21)$$

**Teorema 8.** *Lo spazio nativo  $\mathcal{N}_\Phi(\Omega)$  associato a una funzione simmetrica definita positiva  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$  è uno spazio (funzionale) di Hilbert con reproducing kernel  $\Phi$ .*

Osserviamo che la nozione di spazio nativo è generale e può essere associata a una generica funzione (condizionatamente) definita positiva, non necessariamente radiale. Introduciamo ora un risultato legato più strettamente all'ambito dell'interpolazione RBF, che è alla base dei risultati di ottimalità introdotti nella prossima sezione.

**Teorema 9.** *Sia  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$  l'insieme dei nodi di interpolazione con valori associati  $\{f_j\}_{1 \leq j \leq N}$ , e sia  $f \in \mathcal{N}_\Phi$  tale che  $f(\mathbf{x}_j) = f_j$ , per  $j = 1, \dots, N$ . Allora se  $F_{\Phi, X}$  è l'unica interpolante della forma (2.2) con  $\Phi$  simmetrica definita positiva, vale la seguente relazione:*

$$(f - F_{\Phi, X}, s)_{\mathcal{N}_\Phi(\Omega)} = 0 \quad \forall s \in \text{span}\{\Phi(\cdot, \mathbf{x}_j) : \mathbf{x}_j \in X\} \cap F_\Phi(\Omega). \quad (2.22)$$

*In particolare,*

$$(f - F_{\Phi, X}, F_{\Phi, X})_{\mathcal{N}_\Phi(\Omega)} = 0. \quad (2.23)$$

**Corollario 1.** *Nelle stesse ipotesi del Teorema 9, valgono le stime:*

$$|F_{\Phi, X}|_{\mathcal{N}_\Phi(\Omega)} \leq |f|_{\mathcal{N}_\Phi(\Omega)} \quad (2.24)$$

$$|f - F_{\Phi, X}|_{\mathcal{N}_\Phi(\Omega)} \leq |f|_{\mathcal{N}_\Phi(\Omega)}. \quad (2.25)$$

Sul Teorema 9, di cui una versione quasi identica vale per RBF condizionatamente definite positive, si basa la dimostrazione dei risultati di ottimalità riportati nella prossima sezione. A questo punto è legittimo chiedersi quali funzioni appartengono allo spazio nativo associato a una funzione di base radiale, in particolare se esso è abbastanza ricco per dare ai risultati presentati una valenza generale. Esistono diverse caratterizzazioni di questo spazio a seconda del tipo di RBF impiegato. Per le RBF proposte da H. Wendland in [33] vale il sorprendente (e rassicurante) risultato che riportiamo di seguito.

**Teorema 10.** *Siano  $\Phi_{d,k} = \phi_{d,k}(\|\cdot\|)$  le funzioni di base radiali proposte in [33]. Allora*

$$\mathcal{N}_{\Phi_{d,k}}(\mathbb{R}^d) = H^{d/2+k+1/2}(\mathbb{R}^d), \quad (2.26)$$

dove  $H^s(\mathbb{R}^d)$  denota il ben noto spazio di Sobolev:

$$H^s(\mathbb{R}^d) = \{f \in L^2(\mathbb{R}^d) \quad : \quad \tilde{f}(\cdot)(1 + \|\cdot\|^2)^{s/2} \in L^2(\mathbb{R}^d)\}. \quad (2.27)$$

### 2.6.2 Ricostruzione ottimale

Fin qui è stato presentato l'approccio basato su RBF per risolvere il problema di interpolazione multivariata (P). Come osservato precedentemente, uno dei principali vantaggi di questo approccio è costituito dal fatto che esso fornisce una ricostruzione *ottimale*. In questo paragrafo sono presentati due risultati di ottimalità basati sulla nozione di *spazio nativo* introdotta nella Sezione 2.6.1. Sebbene la costruzione di questo spazio sia stata presentata più in dettaglio per funzioni di base radiali definite positive, un procedimento del tutto analogo può essere impiegato per caratterizzare spazi nativi associati a funzioni condizionatamente definite positive (nel senso della definizione 3). I risultati di seguito valgono per entrambi i tipi di funzioni.

Il primo teorema stabilisce la proprietà di migliore approssimazione dell'interpolazione RBF. In particolare, supponiamo che i valori  $\{f_j\}_{1 \leq j \leq N}$  da interpolare siano campionamenti di una funzione incognita  $f$ , che supponiamo appartenere allo spazio nativo associato alla RBF (condizionatamente) definita positiva  $\Phi$ . Si è visto come l'interpolante  $F_{f,X}$ , che esiste ed è unica, utilizza alcune informazioni disponibili su  $f$  per *approssimarla*; tuttavia, è legittimo domandarsi se non sia possibile prendere una qualunque altra funzione dello spazio

$$V_X := \left\{ s(\mathbf{x}) = \sum_{j=1}^N \alpha_j \Phi(\mathbf{x}, \mathbf{x}_j) + P(\mathbf{x}) \quad : \quad P \in \mathbb{P}_m^d \text{ e } \sum_{j=1}^N \alpha_j Q(\mathbf{x}_j) = 0 \quad \forall Q \in \mathbb{P}_m^d \right\}.$$

Possiamo, a tal proposito, enunciare il seguente

**Teorema 11.** *Sia  $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$  continua e condizionatamente definita positiva di grado  $m \geq 0$ . Sia inoltre  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$  un insieme  $\mathbb{P}_m^d$ -unisolvante e  $f \in \mathcal{N}_\Phi(\Omega)$  una funzione incognita di cui si conoscono solo*

i valori su  $X$ . Allora l'interpolante RBF (2.4) (o (2.2) se  $m = 0$ )  $F_{f,X}$  è la migliore approssimazione di  $f$  dello spazio  $V_X$  rispetto alla seminorma  $|\cdot|_{\mathcal{N}_\Phi}$ , ossia vale

$$|f - F_{f,X}|_{\mathcal{N}_\Phi} \leq |f - s|_{\mathcal{N}_\Phi} \text{ per ogni } s \in V_X. \quad (2.28)$$

In particolare,  $F_{f,X}$  è la proiezione ortogonale di  $f$  su  $V_X$ .

*Dimostrazione.* Il risultato deriva direttamente dal Teorema 9, per cui, in particolare, vale la seguente caratterizzazione di migliore approssimazione per gli spazi di Hilbert:  $(f - F_{f,X}, s)_{\mathcal{N}_\Phi(\Omega)} = 0$ , per ogni  $s \in V_X$ .  $\square$

Un'altra interessante proprietà dell'interpolazione RBF è quella della *minimizzazione dell'energia*, stabilita nel prossimo teorema (per la cui dimostrazione si veda [34]). In particolare, esso mette in evidenza come risolvere il problema mal posto (P) mediante funzioni di base radiali (condizionatamente) definite positive rispetti la filosofia di un *approccio regolarizzante*: in particolare, fra tutte le possibili soluzioni di (P) se ne cerca una che minimizzi l'energia, e che sia quindi più regolare possibile. Per un approfondimento sui legami fra RBF e approccio regolarizzante per problemi mal posti si veda [13], dove le RBF vengono introdotte nel contesto delle reti neurali.

**Teorema 12.** *Sotto le stesse ipotesi del Teorema 11, l'interpolante RBF  $F_{f,X}$  ha (semi-)norma  $|\cdot|_{\mathcal{N}_\Phi}$  minima fra tutte le funzioni  $s \in \mathcal{N}_\Phi(\Omega)$  che interpolano i dati  $\{f_j\}$  in  $X$ , ossia:*

$$|F_{f,X}|_{\mathcal{N}_\Phi} = \min\{|s|_{\mathcal{N}_\Phi} : s \in \mathcal{N}_\Phi(\Omega) \text{ tale che } s(\mathbf{x}_j) = f_j, 1 \leq j \leq N\}. \quad (2.29)$$

Questo risultato assume significati particolari a seconda delle RBF utilizzate. Ad esempio, nel caso bivariato l'uso di funzioni di base radiali del tipo  $\phi(r) = r^2 \log(r)$  porta a risolvere il problema (P) minimizzando la *curvatura totale aggregata* (cf. [31])

$$E(f) = \left( \int_{\mathbb{R}^2} \left| \frac{\partial^2 f}{\partial x_1^2} \right| + 2 \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right| + \left| \frac{\partial^2 f}{\partial x_2^2} \right| \right)$$

## 2.7 Stime dell'errore di interpolazione mediante RBF

Presentiamo in questa sezione le stime per l'errore associato all'interpolazione multivariata per il problema (P) mediante RBF. Introduciamo dapprima la seguente

**Definizione 8.** *Un insieme  $\Omega \subseteq \mathbb{R}^d$  soddisfa la condizione ICC (Interior Cone Condition) se esiste un angolo  $\theta \in [0, \pi/2]$  e un raggio  $r > 0$  tali che, per ogni  $\mathbf{x} \in \Omega$ , esiste un vettore di norma unitaria  $\xi(\mathbf{x})$  tale che il cono*

$$C(\mathbf{x}, \xi(\mathbf{x}), \theta, r) := \{\mathbf{x} + \lambda \mathbf{y} : \mathbf{y} \in \mathbb{R}^d, \|\mathbf{y}\|_2 = 1, \mathbf{y}^T \xi(\mathbf{x}) \geq \cos(\theta), \lambda \in [0, r]\}$$

è contenuto in  $\Omega$ .

Supponiamo ora che l'insieme dei centri di interpolazione  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  sia incluso in un aperto limitato  $\Omega \subseteq \mathbb{R}^d$  che rispetta la condizione ICC. Fissata una funzione di base  $\Phi$  fra quelle riportate nelle Tabelle 2.1 e 2.2, supponiamo inoltre che i valori da interpolare siano campionamenti di una funzione (incognita)  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , ossia che  $f(\mathbf{x}_j) = f_j \forall j = 1, \dots, N$ , e che  $f$  appartenga a  $\mathcal{N}_\Phi$ , lo spazio nativo introdotto nel Paragrafo 2.6.1.

Presentiamo alcune stime per l'errore di interpolazione *globale*, che misura lo scarto fra la funzione  $f$  e l'interpolante  $F_{X,f}$  della forma (2.2) (o (2.4) a seconda della scelta di  $\Phi$ ). Osserviamo che tali stime hanno una struttura del tutto analoga a quella che caratterizza un'approssimante basata su elementi finiti, in cui la differenza fra la soluzione esatta di un problema differenziale e la corrispondente approssimante numerica viene valutata in funzione del cosiddetto *passo di griglia*  $h$  che caratterizza una mesh poligonale. In questo caso, visto che il metodo di interpolazione RBF non prevede l'utilizzo di alcuna mesh, il ruolo del passo di griglia viene assunto dalla cosiddetta *fill distance*:

$$h_{\Omega,X} = \sup_{\mathbf{x} \in \Omega} \left( \min_{\mathbf{x}_j \in X} \|\mathbf{x} - \mathbf{x}_j\|_{\mathbb{R}^d} \right). \quad (2.30)$$

In questo paragrafo sono presentati i risultati che coinvolgono le funzioni di base radiali riportate nelle Tabelle 2.1 e 2.2, anche se è possibile estendere questi risultati a casi più generali. Per altri esempi (e per le dimostrazioni delle stime riportate) si rimanda a [34].

Per tutte le funzioni di base radiali riportate nelle Tabelle 2.1-2.2 è possibile dimostrare che per l'errore di interpolazione globale  $\|f - F_{f,X}\|_{L^\infty(\Omega)}$  vale la seguente maggiorazione:

$$\|f - F_{f,X}\|_{L^\infty(\Omega)} \leq C |f|_{\mathcal{N}_\Phi} \mathcal{F}_\Phi^{1/2}(h_{\Omega,X}), \quad (2.31)$$

dove  $C$  è una costante e dove il valore di  $\mathcal{F}_\Phi(h_{\Omega,X})$  per le RBF citate in questo capitolo è riportato nella Tabella 2.7.

## 2.8 Principio di indeterminazione

In [27] Schaback enuncia un risultato fondamentale ai fini della comprensione dei fenomeni che intervengono nell'interpolazione RBF, che impone di fissare un compromesso fra riduzione dell'errore di interpolazione e sensibilità del metodo a possibili variazioni sui dati.

Supponiamo che i dati  $f_j$  nel problema (P) siano i valori di una funzione (incognita)  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  tale che  $f_j = f(\mathbf{x}_j)$  per  $1 \leq j \leq N$ , con  $f \in \mathcal{N}_\Phi$  (dove  $\mathcal{N}_\Phi$  è lo spazio nativo introdotto nel Paragrafo 2.6.1; in particolare, esso è dotato di seminorma  $|\cdot|_{\mathcal{N}}$ ). Allora, se  $F = F_{f,X}$  è la funzione in-

terpolante definita come in (2.2),<sup>2</sup> l'errore di interpolazione puntuale può essere espresso nel seguente modo:

$$|f(\mathbf{x}) - F(\mathbf{x})| \leq \Theta(\mathbf{x})|f|_{\mathcal{N}}, \quad (2.32)$$

dove  $\Theta(\mathbf{x})$  dipende da  $X, \mathbf{x}, \mathcal{N}_{\Phi}, \Phi$  ma non da  $f$ .

D'altra parte, la costante di positiva definizione per la matrice di interpolazione è legata alla sensibilità dell'interpolante. Infatti, come visto nella Sezione 2.3, una condizione necessaria di esistenza per la funzione interpolante è che la matrice  $A \in \mathbb{R}^{N \times N}$  con coefficienti  $A_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$  sia (condizionatamente) definita positiva, ossia che esista  $\lambda > 0$  tale che:

$$\lambda \|\mathbf{x}\|_2^2 \leq \mathbf{x}^T A \mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^d, \text{ con } \mathbf{x} \neq 0. \quad (2.33)$$

Allora,  $\lambda^{-1}$  controlla la sensibilità della funzione interpolante  $F$  rispetto a variazioni dei dati  $\{f_j\}_{1 \leq j \leq N}$ . Ci restringiamo per semplicità al caso  $m = 0$ . Posto

$$\mathbf{f} = [f_1, \dots, f_N]^T, \quad \mathbf{\Lambda} = [\lambda_1, \dots, \lambda_N],$$

vale l'identità

$$\mathbf{\Lambda}^T A \mathbf{\Lambda} = \mathbf{\Lambda}^T \mathbf{f};$$

che implica, se vale la (2.33),

$$\|\mathbf{\Lambda}\|_2 \leq \lambda^{-1} \|\mathbf{f}\|_2.$$

Analogamente, se esiste una costante  $L > 0$  tale che

$$\mathbf{x}^T A \mathbf{x} \leq L \|\mathbf{x}\|_2^2 \quad \forall \mathbf{x} \in \mathbb{R}^d \text{ con } \mathbf{x} \neq 0,$$

possiamo dedurre che

$$L^{-1} \|\mathbf{f}\|_2 \leq \|\mathbf{\Lambda}\|_2.$$

In presenza di una perturbazione sui dati:  $\mathbf{f} + \Delta \mathbf{f}$  che comporta una perturbazione dei valori dei coefficienti dell'interpolante  $\mathbf{\Lambda} + \Delta \mathbf{\Lambda}$ , si può concludere dunque che:

$$\frac{\|\Delta \mathbf{\Lambda}\|_2}{\|\mathbf{\Lambda}\|_2} \leq \frac{L \|\Delta \mathbf{f}\|_2}{\lambda \|\mathbf{f}\|_2} \quad (2.34)$$

Per  $m > 0$ , è possibile osservare che, dal momento che la determinazione dei coefficienti di  $P(\mathbf{x})$  nella (2.6) può essere interpretato come un problema di interpolazione polinomiale, l'accuratezza dell'interpolazione RBF rimane comunque governata dalle costanti  $L$  e  $\lambda$ .

---

<sup>2</sup>Per semplicità trattiamo il caso in cui  $F$  è della forma (2.2), ma tutte le osservazioni riportate valgono anche nel caso di funzioni interpolanti della forma (2.4).



Sotto alcune ipotesi supplementari sullo spazio  $\mathcal{F}$ , è possibile dimostrare il seguente *principio di indeterminazione di Schaback*:

$$\Theta(\mathbf{x})^2 \cdot \lambda^{-1} \geq 1 \tag{2.35}$$

Questo risultato, che presenta un'affascinante analogia con il più celebre principio di indeterminazione che vige nella fisica quantistica, stabilisce quindi l'impossibilità di rimpicciolire a piacere sia i *bounds* per l'errore puntuale che la sensibilità a variazioni dei dati basandosi su un approccio di tipo RBF.

Tabella 2.2: RBF a supporto compatto proposte da H.Wendland.

| Dimensione $d$ | RBF  | Regolarità                       |
|----------------|--|----------------------------------|
| $d = 1$        | $\phi_{1,0} = (1 - r)_+$<br>$\phi_{1,1} = (1 - r)_+^3 (3r + 1)$<br>$\phi_{1,2} = (1 - r)_+^5 (8r^2 + 5r + 1)$  | $C^0$<br>$C^2$<br>$C^4$          |
| $d \leq 3$     | $\phi_{3,0} = (1 - r)_+^2$<br>$\phi_{3,1} = (1 - r)_+^4 (4r + 1)$<br>$\phi_{3,2} = (1 - r)_+^6 (35r^2 + 18r + 3)$<br>$\phi_{3,3} = (1 - r)_+^8 (32r^3 + 25r^2 + 8r + 1)$ | $C^0$<br>$C^2$<br>$C^4$<br>$C^6$ |
| $d \leq 5$     | $\phi_{5,0} = (1 - r)_+^3$<br>$\phi_{5,1} = (1 - r)_+^5 (5r + 1)$<br>$\phi_{5,2} = (1 - r)_+^7 (16r^2 + 7r + 1)$   | $C^0$<br>$C^2$<br>$C^4$          |

| RBF                       | $\phi(r)$                     | $\mathcal{F}_\Phi(h)$ |
|---------------------------|-------------------------------|-----------------------|
| Splines<br>poliarmoniche  | $r^\nu$<br>$r^{2k} \log(r)$   | $h^\nu$<br>$h^{2k}$   |
| Gaussiane                 | $\exp -\frac{r^2}{2\sigma^2}$ | $e^{c \log h /h}$     |
| Multiquadratiche          | $(1 + r^2)^\nu$               | $e^{-c/h}$            |
| Multiquadratiche inverse  | $(1 + r^2)^\nu$               | $e^{-c/h}$            |
| RBF di Wendland (Tab.2.2) | $\phi_{d,k}(r)$               | $h^{2k+1}$            |

Tabella 2.3: Famiglie di RBF e *bounds* sull'errore di interpolazione RBF.

# 3

## Due algoritmi efficienti per la ricostruzione di superfici

Come descritto nel Capitolo 1, il problema della ricostruzione di una superficie a partire da un insieme (nuvola) di punti che giacciono su di essa può essere convenientemente formulato come un problema di interpolazione multivariata. Nel Capitolo 2 sono state illustrate le principali caratteristiche e i vantaggi derivanti dall'uso di funzioni di base radiali (RBF) al fine di risolvere tale problema. In particolare, nella Sezione 2.4 abbiamo illustrato come le RBF possano essere suddivise in due grandi categorie, quelle a supporto globale e quelle a supporto compatto, e come la scelta del tipo di funzione radiale influisca in modo decisivo sulle prestazioni di un algoritmo risolutivo.

In questo capitolo presentiamo due algoritmi suggeriti rispettivamente in [24] e in [25] che tentano di conciliare i vantaggi di entrambe le tipologie di RBF unendo rapidità di calcolo e robustezza della ricostruzione: entrambi i metodi fanno uso di RBF a supporto compatto. Il primo, presentato nelle Sezioni 3.1, 3.2 e 3.3, introduce un procedimento multi-scala per migliorare la ricostruzione in caso di nuvole di punti campionati con densità spaziale variabile; il secondo, descritto nella Sezione 3.4, sfrutta un calcolo adattativo del supporto di ogni funzione di base per migliorare la qualità della ricostruzione.

Entrambi gli algoritmi sono stati implementati in C/C++ e testati su una serie di campioni di dati; come metteremo in evidenza nel Capitolo 4, nei test che abbiamo effettuato uno dei due algoritmi si è dimostrato più efficiente e versatile dell'altro. Nello stesso capitolo inoltre alcune verranno proposte alcune modifiche per uno degli algoritmi al fine di ovviare ad alcuni inconvenienti messi in evidenza dalle prove sperimentali.

Un approccio simile a quello impiegato nel primo algoritmo descritto è stato proposto in [20], in cui dall'insieme dei punti da interpolare  $X$  viene ricavata una sequenza di insiemi di punti con cardinalità crescente

$X_1 \subset X_2 \subset \dots \subset X$  la cui interpolazione permette di ottenere un'approssimazione della superficie a scale diverse, via via più fini. Tuttavia, il processo di costruzione di ogni sottoinsieme  $X_i$  necessita la costruzione di una triangolazione sull'insieme  $X_{i+1}$  e questo rende il processo di calcolo molto meno efficace di quello presentato nel prossimo capitolo.

### 3.1 Un algoritmo multi-livello con RBF a supporto compatto

L'algoritmo RBF presentato in [24] si propone di superare alcuni dei limiti derivanti dall'uso di funzioni a supporto compatto introducendo un opportuno procedimento iterativo. In particolare, viene effettuata una *partizione* della nuvola iniziale di dati in insiemi di punti (*livelli*) di cardinalità via via crescente e per ogni livello viene effettuata un'interpolazione RBF: la funzione interpolante finale è data dalla somma dei contributi calcolati a ogni livello. A seconda del numero di livelli impiegato è possibile ottenere una funzione interpolante o una funzione approssimante.

In particolare, nella Sezione 3.2 presenteremo un metodo per l'interpolazione diretta di un insieme di punti (interpolazione mono-livello) a partire da una nuvola di punti  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . Nel Paragrafo 3.3.1 verrà illustrato come calcolare una serie di nuvole di punti  $X_1, X_2, \dots, X_n \equiv X$  di complessità crescente (*livelli*) a partire dai dati originali collezionati in  $X$ ; nel Paragrafo 3.3.2 sarà descritto come il processo di interpolazione mono-livello possa essere applicato iterativamente a ciascun livello per ottenere una funzione approssimante  $F$  che non soffra degli inconvenienti tipici dell'uso di RBF a supporto locale.

Nel seguito dell'esposizione ci riferiremo sempre alla ricostruzione di superfici in  $\mathbb{R}^3$  anche se i metodi presentati possono essere generalizzati a spazi con un numero arbitrario di dimensioni. Adotteremo inoltre la convenzione per cui, in presenza di superfici chiuse,  $F$  assume valori positivi all'esterno della superficie e valori negativi all'interno di essa.

Per quanto riguarda i dati aggiuntivi richiesti per l'interpolazione RBF a cui abbiamo accennato nel Paragrafo 2.2.1, questo algoritmo fa uso dei vettori normali alla superficie in ogni punto della nuvola. Nel seguito supporremo quindi di avere a disposizione come dati del problema:

- una **nuvola di punti**  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^3$  rilevati sulla superficie compresi in un dominio limitato  $\Omega \subset \mathbb{R}^3$ ;
- l'insieme dei **vettori normali**  $N = \{\mathbf{n}_1, \dots, \mathbf{n}_N\}$  alla superficie da ricostruire in tutti i punti appartenenti alla nuvola.

Per convenzione supporremo che i vettori normali siano rivolti verso la porzione di spazio dove  $F(\mathbf{x}) < 0$ .

## 3.2 Interpolazione multivariata mono-livello

In questa sezione viene descritta una procedura per effettuare l'interpolazione di una generica nuvola di punti  $X$  in modo diretto e ottenere una superficie che passi per tutti i punti della nuvola. Questo metodo si differenzia da quanto previsto dall'approccio RBF classico grazie alla forma assunta dalla funzione interpolante che garantisce una maggiore flessibilità della ricostruzione. Tale procedimento presenta però le limitazioni tipiche legate all'uso di RBF a supporto compatto. Tali limitazioni verranno superate grazie all'approccio multi-scala descritto nella prossima sezione.

### 3.2.1 Scelta della funzione interpolante

Il metodo che presentiamo propone una forma per la funzione interpolante leggermente diversa da (2.2) (o (2.4)). Infatti, dato l'insieme di punti da interpolare  $X$ , cerchiamo la funzione interpolante  $F : \mathbb{R}^3 \rightarrow \mathbb{R}$  della forma:

$$F(\mathbf{x}) = \sum_{i=1}^N (g_i(\mathbf{x}) + \lambda_i) \phi_\sigma(\|\mathbf{x} - \mathbf{x}_i\|), \quad (3.1)$$

dove  $g_i : \mathbb{R}^3 \rightarrow \mathbb{R}$  e  $\lambda_i \in \mathbb{R}$  sono, rispettivamente, delle funzioni e dei coefficienti scalari incogniti,  $\phi$  è una RBF a supporto in  $[0, 1)$  (ad esempio una delle RBF riportate in Tabella 2.2) e  $\phi_\sigma(\cdot) = \phi(\cdot/\sigma)$ .

Osserviamo che rispetto alla forma (2.2) corrispondente all'approccio RBF originale, la funzione interpolante presenta i termini supplementari  $g_i(\mathbf{x})$ . Riscriviamo la (3.1) come

$$F(\mathbf{x}) = \sum_{i=1}^N \left[ g_i(\mathbf{x}) \phi_\sigma(\|\mathbf{x} - \mathbf{x}_i\|) + \lambda_i \phi_\sigma(\|\mathbf{x} - \mathbf{x}_i\|) \right], \quad (3.2)$$

e notiamo che il secondo addendo è identico all'interpolante (2.2) del modello RBF classico. Per quanto riguarda il primo addendo osserviamo che, come descritto in [25], la funzione  $\sum_{i=1}^N g_i(\mathbf{x}) \phi_\sigma(\|\mathbf{x} - \mathbf{x}_i\|)$  ha lo stesso insieme di livello zero (e quindi rappresenta la stessa superficie) dell'interpolante

$$\sum_{i=1}^N g_i(\mathbf{x}) \Phi_{\sigma,i}(\mathbf{x}), \quad (3.3)$$

dove le funzioni

$$\Phi_{\sigma,i}(\mathbf{x}) = \frac{\phi_\sigma(\|\mathbf{x} - \mathbf{x}_i\|)}{\sum_{j=1}^N \phi_\sigma(\|\mathbf{x} - \mathbf{x}_j\|)} \quad (3.4)$$

formano una *partizione dell'unità* su  $\Omega$ , ossia un insieme di funzioni non negative a supporto compatto tali che per ogni  $\mathbf{x} \in \Omega$  si ha che  $\sum_i \Phi_{\sigma,i}(\mathbf{x}) = 1$ .

Nell'Appendice B presentiamo brevemente la ricostruzione di superfici basata sulla partizione dell'unità, un approccio che a volte può essere combinato con l'approccio RBF per ottenere metodi ibridi molto efficaci (si veda, ad esempio, [?]). Esso consiste nel trovare delle funzioni approssimanti locali che vengono poi saldate in modo continuo impiegando una partizione dell'unità per formare l'approssimante globale.

L'interpolazione proposta nel metodo che presentiamo sfrutta l'idea di arricchire l'interpolazione RBF classica con un elemento tipico dell'interpolazione basata su una partizione dell'unità, ossia di introdurre delle approssimanti locali per ogni  $\sigma$ -intorno dei punti della nuvola. Osserviamo che, in ogni caso, si tratta di un metodo essenzialmente basato su RBF, in quanto la partizione dell'unità non viene mai direttamente impiegata nè è necessario calcolarla.

### 3.2.2 Calcolo dell'interpolazione con approccio misto

L'interpolazione, nel caso di una funzione interpolante della forma (3.1), avviene attraverso due passi distinti. Osserviamo innanzitutto che in ogni termine della sommatoria in (3.1) interviene una funzione di base radiale a supporto compatto centrata in un punto  $\mathbf{x}_i$  della nuvola, il cui supporto è un  $\sigma$ -intorno di  $\mathbf{x}_i$ .

Il primo passo consiste nel ricostruire per ogni punto  $\mathbf{x}_i$  un'approssimazione locale  $g_i(\mathbf{x})$  dell'interpolante, valida in un  $\sigma$ -intorno di  $\mathbf{x}_i$ . Il secondo passo invece consiste nel determinare i coefficienti  $\{\lambda_i\}_{1 \leq i \leq N}$  in modo da ottenere una funzione interpolante globale e continua e da soddisfare i vincoli di interpolazione. Questa procedura è semplificata (nel caso bidimensionale) in Figura 3.1. Vediamo come sono effettuati questi due passi.

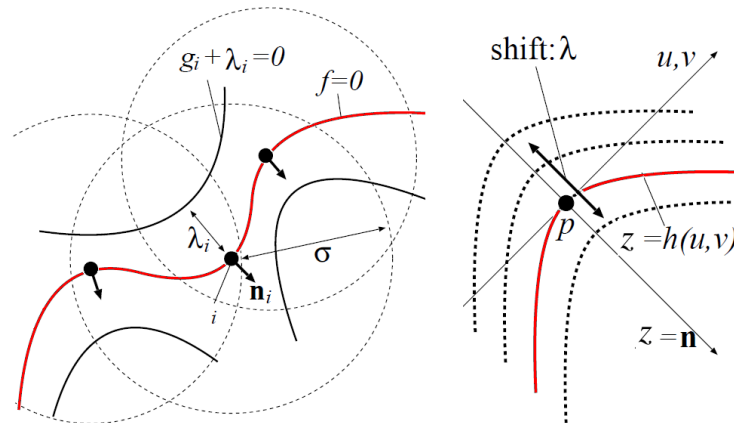


Figura 3.1: Approssimazione locale (in nero) e interpolante globale (in rosso) ottenuta da una traslazione delle approssimanti locali.

1. **Approssimazione locale.** Sia  $\mathcal{B}_\sigma(\mathbf{x}_i)$  il  $\sigma$ -intorno di  $\mathbf{x}_i \in X$  e supponiamo che vi siano  $K$  punti di  $X$  che giacciono in  $\mathcal{B}_\sigma(\mathbf{x}_i)$ . Introduciamo in  $\mathcal{B}_\sigma(\mathbf{x}_i)$  il *sistema di riferimento locale* cartesiano  $\mathcal{R}_i$  di assi  $(u, v, z)$ , in cui  $z$  è orientato come il versore normale  $\mathbf{n}_i$  associato a  $\mathbf{x}_i$ . In seguito, calcoliamo una funzione approssimante locale  $h : (u, v) \rightarrow z = Au^2 + 2Buv + Cv^2$ , dove i coefficienti incogniti  $A, B, C$  sono determinati con una procedura *ai minimi quadrati*, ossia in modo tale che:

$$A, B, C = \arg \min_{A, B, C \in \mathbb{R}} \sum_{\mathbf{x}_j \in \mathcal{B}_\sigma(\mathbf{x}_i) \cap X} \phi_\sigma(\|\mathbf{x}_j - \mathbf{x}_i\|) (z_{\mathbf{x}_j} - h(u_{\mathbf{x}_j}, v_{\mathbf{x}_j}))^2, \quad (3.5)$$

dove  $(u_{\mathbf{x}_j}, v_{\mathbf{x}_j}, z_{\mathbf{x}_j})$  sono le coordinate del punto  $\mathbf{x}_j \in X$  nel sistema di riferimento locale  $\mathcal{R}_i$ . L'approccio ai minimi quadrati formalizzato in (3.5) porta a risolvere il seguente sistema lineare (per ulteriori dettagli rimandiamo a [?]):

$$M^T \Sigma M \mathbf{y} = M^T \Sigma \mathbf{z}, \quad (3.6)$$

dove:

- $\mathbf{y} = [A \ B \ C]^T \in \mathbb{R}^3$  è il vettore dei coefficienti incogniti;
- $M$  è una matrice di dimensioni  $K \times 3$  le cui righe corrispondono a  $M_j = [u_{\mathbf{x}_j}^2 \ 2u_{\mathbf{x}_j} v_{\mathbf{x}_j} \ v_{\mathbf{x}_j}^2]$ ;
- $\Sigma$  è una matrice diagonale quadrata di ordine  $K$  con componenti diagonali  $\Sigma_{jj} = \phi_\sigma(\|\mathbf{x}_j - \mathbf{x}_i\|)$ ;
- $\mathbf{z} \in \mathbb{R}^K$  è il vettore di componenti  $z_{\mathbf{x}_j}$ .

Una volta calcolati i coefficienti  $A, B, C$  della forma quadratica, definiamo per ogni  $\mathbf{x} \in \mathcal{B}_\sigma(\mathbf{x}_i)$  di coordinate locali  $(u_{\mathbf{x}}, v_{\mathbf{x}}, z_{\mathbf{x}})$  la funzione approssimante locale  $g_i$  come

$$g_i(\mathbf{x}) = z_{\mathbf{x}} - h(u_{\mathbf{x}}, v_{\mathbf{x}}). \quad (3.7)$$

2. **Interpolazione globale.** A questo punto, per determinare univocamente la funzione interpolante in (3.1), occorre calcolare i pesi  $\lambda_i$ . Essi sono calcolati imponendo i vincoli di interpolazione  $F(\mathbf{x}_i) = 0$  per ogni  $\mathbf{x}_i \in X$ . Dato che a questo punto le approssimanti locali  $g_i(\mathbf{x})$  sono note, si ottiene un sistema lineare nelle incognite  $\lambda_1, \dots, \lambda_N$  del tutto simile a quello in (2.3) corrispondente alla classica interpolazione RBF:

$$\sum_{j=1}^N \lambda_j \phi_\sigma(\|\mathbf{x}_i - \mathbf{x}_j\|) = - \sum_{j=1}^N g_j(\mathbf{x}_i) \phi_\sigma(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad \forall 1 \leq i \leq N; \quad (3.8)$$

osserviamo che la matrice è esattamente la stessa del sistema (2.3), ciò che cambia è il termine noto. Grazie alle osservazioni effettuate nel Capitolo 2, nel caso in cui  $\phi$  coincida con una delle RBF riportate in Tabella 2.2, sappiamo che il sistema ammette una e una sola soluzione essendo la matrice ad esso associata invertibile (in particolare, essa è simmetrica e definita positiva se i punti di  $X$  sono tutti distinti).

Il sistema lineare (3.8) ha ordine  $N$ , dove  $N$  è il numero di punti che costituiscono la nuvola  $X$ ; esso può essere risolto efficacemente ricorrendo, per esempio, ad opportuni metodi iterativi; nelle prove sperimentali presentate in [24] viene impiegato il metodo del gradiente bi-coniugato stabilizzato. In tutte le prove presentate in questo lavoro tale sistema è stato risolto invece con il metodo del gradiente coniugato, essendo la matrice simmetrica definita positiva, con preconditionamento ILU. Per ulteriori dettagli sui metodi numerici per la risoluzione di sistemi lineari e sulla tecnica del preconditionamento si veda, ad esempio, [?].

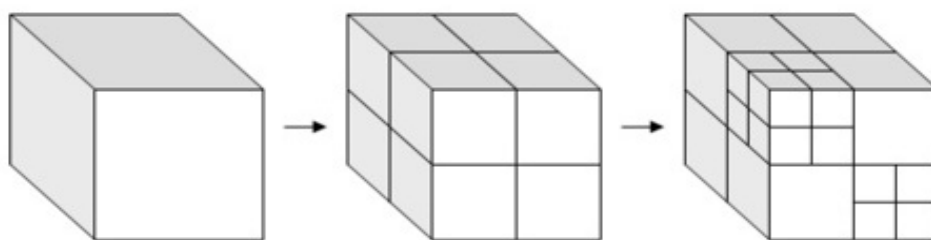


Figura 3.2: Divisione ricorsiva della bounding box.

Il parametro  $\sigma$  è costante e fissato a priori. Qualitativamente, esso dipende dalla densità spaziale dei punti. In [23] è proposta una procedura basata su una divisione ricorsiva dello spazio per determinare  $\sigma$  in modo automatico. Come illustrato in Figura 3.2, partendo da una *bounding box* cubica  $\Omega$  della nuvola di punti  $X$ , la si divide in otto celle di uguali dimensioni il cui spigolo misura metà di quello di  $\Omega$ ; si procede ricorsivamente con le divisioni per ogni cella fino ad arrivare a cubi abbastanza piccoli da contenere ciascuno non più di 8 punti della nuvola; a questo punto  $\sigma$  è calcolata come  $\sigma = cR$ , dove  $R$  è la diagonale delle celle ottenute dopo l'ultima suddivisione e  $c$  è un parametro definito dall'utente. Torneremo sulla questione della scelta di  $\sigma$  nel Capitolo 4.

### 3.2.3 Limiti dell'interpolazione multivariata mono-livello

L'interpolazione multivariata mono-livello può essere utilizzata come metodo a sè per la ricostruzione della superficie desiderata; tuttavia essa presenta alcune limitazioni che rendono necessario un procedimento più raffinato. In Figura 3.3 è mostrata la ricostruzione attraverso interpolazione diretta di un



campione ben noto alla letteratura scientifica, lo Stanford Bunny, partendo da una nuvola di  $N = 34833$  punti<sup>1</sup>. Il processo di interpolazione è accurato e piuttosto rapido (circa 5 secondi per il calcolo della funzione interpolante su un laptop con processore da 2.4GHz); tuttavia esso mostra diversi limiti. Innanzitutto il metodo presenta un'elevata sensibilità al rumore nella mis-

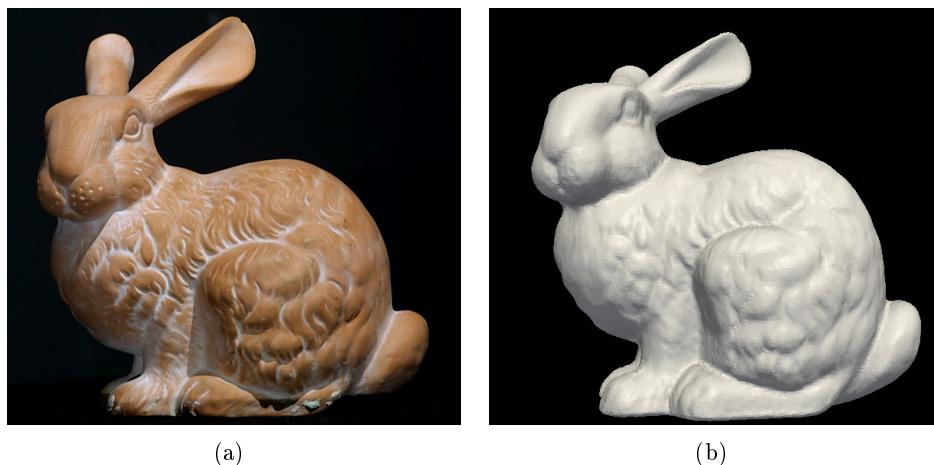


Figura 3.3: Stanford Bunny: modello originale (a sinistra) e modello ricostruito mediante interpolazione diretta partendo da una nuvola  $X$  di  $N = 34833$  punti.

urazione dei dati. Inoltre, come anticipato, se da un lato l'uso di RBF a supporto compatto accelera il calcolo della funzione interpolante, dall'altro esso non permette la ricostruzione di buchi o strappi nella superficie da ricostruire. In Figura 3.4(a) è mostrato il dettaglio della ricostruzione nella parte inferiore del coniglio: in particolare si vedono due buchi, presenti nella nuvola di punti originale, che l'algoritmo di ricostruzione non è riuscito a chiudere. Infine, si riscontra un altro inconveniente tipico dell'uso di RBF a supporto compatto: in Figura 3.4(b) è visibile il supporto della funzione interpolante  $F$  per una sezione verticale della bounding box  $\Omega$ : tale supporto è costituito da una stretta fascia attorno alla superficie ricostruita, al di fuori della quale  $F$  è identicamente nulla. Questo fatto si rivela problematico nella successiva fase di estrazione della mesh tramite poligonalizzazione (come illustrato nella Sezione 1.5), in cui il *passo di poligonalizzazione*  $h$  deve essere obbligatoriamente scelto più piccolo della larghezza di tale fascia.

L'introduzione di un algoritmo multi-livello per l'interpolazione multi-variata permette di ovviare a tutti gli inconvenienti appena messi in evidenza.

<sup>1</sup>Per ulteriori dettagli sulle caratteristiche e le fonti dei campioni utilizzati per i test, si rimanda al capitolo successivo.

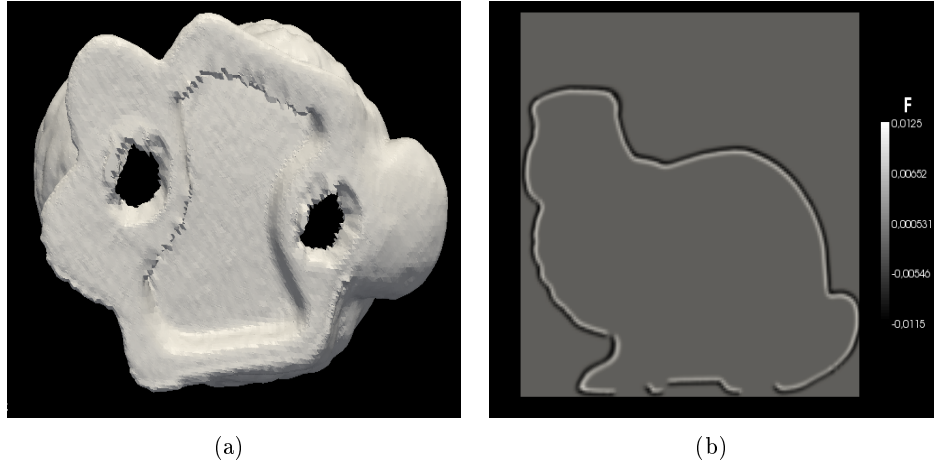


Figura 3.4: Limiti dell'interpolazione diretta: (a) i buchi nella nuvola originale di dati non vengono riempiti dalla ricostruzione; (b) supporto della funzione interpolante per una sezione verticale della bounding box della nuvola  $X$ .

### 3.3 Ricostruzione multi-livello

#### 3.3.1 Creazione di livelli gerarchici

Per ovviare agli inconvenienti che comporta un'interpolazione diretta della nuvola di punti  $X$ , è possibile introdurre un procedimento *iterativo*; l'idea alla base di tale procedimento consiste nel creare una famiglia di nuvole di punti (che indicheremo col nome di livelli nel seguito)  $X_1, X_2, \dots, X_n \equiv X$  di cardinalità crescente:  $\text{card}(X_1) \leq \text{card}(X_2) \leq \dots \leq \text{card}(X_n)$ . In [24] è proposta una procedura per la creazione dei livelli, descritta nell'Algoritmo 1, che permette di creare  $n$  livelli a partire da una nuvola di punti  $X$  e dall'insieme  $N$  dei vettori normali associati a ogni punto in  $X$ .

Osserviamo che i livelli  $X_1, \dots, X_n$  hanno cardinalità crescente e che i punti di ciascun livello non coincidono, salvo casi particolari, con quelli di  $X$ . Nel Paragrafo 4.4.2 discuteremo della possibilità di modificare questa procedura in modo da renderla più efficiente.

#### 3.3.2 Ricostruzione multi-livello

Ottenuti i diversi livelli è possibile effettuare la ricostruzione desiderata definendo ricorsivamente una serie di funzioni approssimanti nel modo seguente:

$$\begin{cases} F_0(\mathbf{x}) \equiv -1; \\ F_k(\mathbf{x}) = F_{k-1}(\mathbf{x}) + o_k(\mathbf{x}) \quad \text{per } k = 1, \dots, n, \end{cases}$$

dove  $o_k(\mathbf{x})$  è una funzione chiamata *offset* da determinarsi in modo tale che  $F_k(\mathbf{x}_i^k) = 0$  per ogni  $\mathbf{x}_i^k \in X_k$ . Per ogni livello, Scegliamo la funzione

---

**Algorithm 1** CREA LIVELLI( $X, N, n$ )

---

1. **Start**
  2. determina un parallelepipedo  $\Omega_0$  contenente tutti i punti di  $X$ ;
  3. **for**  $k = 1, \dots, n$  **do**
  4. per ogni  $i = 1, \dots, 8^{k-1}$  dividi la cella  $\Omega_{k-1}^i$  in 8 celle  $\Omega_k^{8(i-1)+1}, \Omega_k^{8(i-1)+2}, \dots, \Omega_k^{8i}$  spiccando dai punti medi di ciascun lato dei piani paralleli alle facce del parallelepipedo.
  5. per ogni cella del livello  $k$ -esimo  $\{\Omega_k^i\}_{i=1, \dots, 8^k}$ , trova il centro di gravità  $\mathbf{x}_k^i$  dei punti di  $X$  in essa contenuti.
  6. per ogni  $\mathbf{x}_k^i$ , calcola il vettore normale di norma unitaria  $\mathbf{n}_k^i$  corrispondente come media dei vettori normali assegnati ai punti di  $X$  contenuti in  $\Omega_k^i$
  7.  $X_k = \{\mathbf{x}_k^1, \dots, \mathbf{x}_k^{8^k}\}$ ,  $N_k = \{\mathbf{n}_k^1, \dots, \mathbf{n}_k^{8^k}\}$ ;
  8. **end for**
  9. **End**
- 

incognita  $o_k(\mathbf{x})$  della forma

$$o_k(\mathbf{x}) = \sum_{\mathbf{x}_k^i \in X_k} [g_k^i(\mathbf{x}) + \lambda_k^i] \phi_{\sigma_k}(\|\mathbf{x} - \mathbf{x}_k^i\|), \quad (3.9)$$

ovvero della stessa forma della funzione interpolante mono-livello presentata in Sezione 3.2. In particolare, le funzioni  $g_k^i(\mathbf{x})$  sono calcolate come illustrato nel Paragrafo 3.2.2. I coefficienti scalari  $\lambda_k^i$  sono calcolati imponendo le condizioni di interpolazione

$$F_{k-1}(\mathbf{x}_k^i) + o_k(\mathbf{x}_k^i) = 0, \text{ per ogni } \mathbf{x}_k^i \in X_k, \quad (3.10)$$

il che, essendo  $o_k(\mathbf{x})$  della forma (3.9), porta a risolvere un sistema lineare del tutto analogo a quello in (3.8). Tale sistema lineare, analogamente a quanto illustrato nel Paragrafo 3.2.2, può essere risolto con il metodo iterativo del gradiente coniugato. Per quanto riguarda il parametro  $\sigma_k$ , che controlla l'ampiezza del dominio delle RBF a ciascun livello, nell'algoritmo proposto in [24] esso viene determinato ricorsivamente secondo la seguente legge:

$$\begin{cases} \sigma_1 = cL; \\ \sigma_{k+1} = \frac{\sigma_k}{2} \text{ per ogni } k > 1 \end{cases} \quad (3.11)$$

dove  $L$  è la diagonale di una bounding box  $X$  a forma di parallelepipedo e  $c$  è un parametro scelto dall'utente. In [24] si propone di porre  $c = 0.75$ , nel Paragrafo 4.4.1 è presentata una discussione critica di questa scelta.

È possibile applicare questo procedimento multi-livello per un numero di livelli  $n$  sufficiente affinché la funzione  $F$  interpoli tutti i punti della

nuvola  $X$ . Laddove ciò non sia richiesto, invece, è possibile ottenere una ricostruzione approssimata della superficie impiegando un numero di livelli limitato. In [24] si propone di determinare il numero di livelli come  $n = \lceil -\log_2(\sigma_0/(2\sigma_1)) \rceil$ , dove  $\sigma_0$  è la taglia del supporto delle RBF per l'interpolazione mono-livello, e  $\sigma_1 = cL$  è definito come in (3.11).

In Figura 3.5 è mostrato il procedimento iterativo multi-livello di ricostruzione per la superficie di un arco aortico ascendente utilizzando 4 livelli.

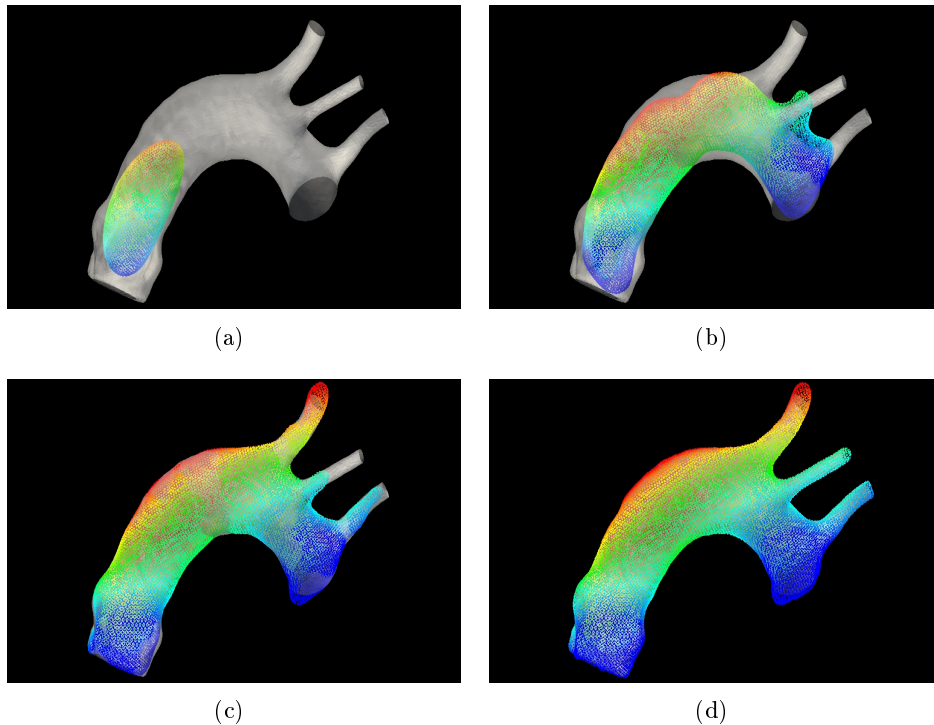


Figura 3.5: Ricostruzione multi-livello di un arco aortico ascendente utilizzando un approccio a 4 livelli e per  $N = 9591$ .

### 3.4 Un algoritmo adattativo con RBF a supporto compatto

L'algoritmo presentato nelle Sezioni 3.1, 3.2 e 3.3 rappresenta un tentativo di sfruttare le potenzialità offerte dalle RBF a supporto compatto ovviando alle limitazioni che il loro uso può comportare. Questa tuttavia non è l'unica possibilità, sono possibili anche altre scelte. Ad esempio, è possibile modificare la taglia del supporto di ogni funzione di base adattandola alle variazioni nella distribuzione spaziale dei punti di  $X$ , in modo tale che il

supporto sia più grande per le funzioni di base centrate in regioni con una bassa frequenza spaziale dei punti (ad esempio, in prossimità di buchi) e più piccola in regioni ricche di punti e di dettagli.

L'algoritmo proposto in [25] fa uso di funzioni di base della forma  $\phi_\sigma(\cdot) = \phi(\cdot/\sigma)$ , dove  $\phi(\cdot)$  è una delle RBF a supporto compatto riportate in Tabella 2.2. Il controllo sulla taglia del supporto avviene attraverso il parametro  $\sigma$ , che viene determinato in modo adattativo per ogni funzione di base. Dati una nuvola di punti  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^3$ , l'insieme delle normali associate  $N = \{\mathbf{n}_1, \dots, \mathbf{n}_N\}$  e un insieme di centri di approssimazione  $C = \{\mathbf{c}_1, \dots, \mathbf{c}_M\}$  con  $M < N$ , cerchiamo una funzione approssimante della forma

$$F(\mathbf{x}) = \sum_{i=1}^M (g_i(\mathbf{x}) + \lambda_i) \phi_{\sigma_i}(\|\mathbf{x} - \mathbf{c}_i\|), \quad (3.12)$$

ovvero definita esattamente come la (3.1), ma con un numero di centri inferiore. Osserviamo che non soltanto le approssimanti locali  $g_i(\mathbf{x})$  e i pesi  $\lambda_i$  sono incogniti, ma ora occorre calcolare anche i parametri  $\sigma_i$ ; inoltre è necessario selezionare i centri  $\mathbf{c}_i$ , che generalmente non sono noti a priori, tra i punti di  $X$ .

Nel seguito esponiamo le procedure proposte in [25] per:

1. il calcolo dell'approssimante locale  $g_i$  associata a un centro  $\mathbf{c}_i$  ;
2. il calcolo del parametro  $\sigma_i$  che controlla il supporto della funzione di base centrata in  $\mathbf{c}_i$ ;
3. la selezione dei centri  $\{\mathbf{c}_i\}_{1 \leq i \leq M}$ ;
4. il calcolo dei pesi scalari  $\{\lambda_i\}_{1 \leq i \leq M}$ .

L'algoritmo proposto in [25] introduce anche la possibilità che, al momento dell'acquisizione dei dati, a ogni punto  $\mathbf{x}_i$  dell'insieme  $X$  venga associato un *parametro di confidenza*  $c_i \in [0, 1]$  che dà un'indicazione di quanto la misura sia affidabile. In caso non si disponga di tale dato si può porre, nella trattazione che segue,  $c_i = 1$  per ogni  $i = 1, \dots, N$ . Nel seguito si fa inoltre uso di un parametro  $d_i$  calcolato per ogni punto della nuvola che ha la funzione di *peso*: i punti della nuvola più isolati dagli altri hanno un peso maggiore; punti con parametro di confidenza maggiore hanno un peso maggiore. I pesi sono infatti calcolati come

$$d_i = c_i \sum_{j=1}^K \|\mathbf{x}_j - \mathbf{x}_i\|^2, \quad (3.13)$$

dove la somma è estesa ai  $K$  punti  $\mathbf{x}_j \in X$  più vicini a  $\mathbf{x}_i$ . Gli autori di [25] propongono di fissare  $K = 20$ .

### 3.4.1 Calcolo dell'approssimante locale

Per un centro  $\mathbf{c}_i$  e per un parametro  $\sigma$  fissato, il calcolo della funzione approssimante locale  $g_i(\mathbf{x})$  viene determinato in modo analogo a quanto descritto nel Paragrafo 3.2.2. In particolare, si definisce in un  $\sigma$ -intorno di  $\mathbf{c}_i$   $\mathcal{B}_\sigma(\mathbf{c}_i)$  un sistema di riferimento ortogonale locale  $\mathcal{R}_i$  di assi  $(u, v, z)$ , in cui  $z$  è orientato come il versore  $\mathbf{n}$  dato dalla media pesata delle normali associate ai punti  $\mathbf{x}_j \in \mathcal{B}_\sigma(\mathbf{c}_i)$

$$\mathbf{n} = \sum_{\mathbf{x}_j \in \mathcal{B}_\sigma(\mathbf{c}_i) \cap X} d_j \phi_\sigma(\|\mathbf{x}_j - \mathbf{c}_i\|) \mathbf{n}_j, \quad (3.14)$$

dove i coefficienti  $d_j$  sono dati dalla (3.13). Si cerca dunque un'approssimazione della superficie in  $\mathcal{B}_\sigma(\mathbf{c}_i)$  della forma

$$z = h_i(u, v) = Au^2 + 2Buv + Cv^2 + Du + Ev + F, \quad (3.15)$$

dove i coefficienti  $A, B, C, D, E, F$  sono determinati tramite un'approssimazione ai minimi quadrati; in particolare, vengono individuati tramite la risoluzione del seguente problema:

$$\{A, \dots, F\} = \arg \min_{A, \dots, F \in \mathbb{R}} \sum_{\mathbf{x}_j \in \mathcal{B}_\sigma(\mathbf{c}_i) \cap X} d_j \phi_\sigma(\|\mathbf{x}_j - \mathbf{c}_i\|) (z_{\mathbf{x}_j} - h_i(u_{\mathbf{x}_j}, v_{\mathbf{x}_j}))^2, \quad (3.16)$$

dove  $(u_{\mathbf{x}_j}, v_{\mathbf{x}_j}, z_{\mathbf{x}_j})$  sono le coordinate locali del punto  $\mathbf{x}_j \in X$  nel sistema di riferimento locale  $\mathcal{R}_i$ . Fissata  $\sigma$ , la soluzione di (3.16) è determinata in modo del tutto analogo a quanto illustrato nel paragrafo 3.2.2. Una volta calcolati i coefficienti della forma quadratica, definiamo per ogni  $\mathbf{x} \in \mathcal{B}_\sigma(\mathbf{c}_i)$  di coordinate locali  $(u_{\mathbf{x}}, v_{\mathbf{x}}, z_{\mathbf{x}})$  la funzione approssimante locale come

$$g_i(\mathbf{x}) = z_{\mathbf{x}} - h_i(u_{\mathbf{x}}, v_{\mathbf{x}}). \quad (3.17)$$

Osserviamo che l'approssimante locale così calcolata dipende dal valore scelto per  $\sigma$ .

### 3.4.2 Calcolo del parametro $\sigma$

Supponiamo ora che sia noto l'insieme dei centri  $C = \{\mathbf{c}_1, \dots, \mathbf{c}_M\}$ . Nel caso delle RBF a supporto compatto in Tabella 2.2, il supporto di ogni funzione di base  $\phi_\sigma(\cdot) = \phi(\cdot/\sigma)$  centrata in  $\mathbf{c}_i$  è costituito da un  $\sigma$ -intorno di  $\mathbf{c}_i$ .

L'idea alla base dell'algoritmo è quella di determinare la taglia del supporto di ogni funzione di base in modo adattativo rispetto alla densità spaziale dei punti della nuvola. Il procedimento proposto in [25] suggerisce di determinare, per ogni centro  $\mathbf{c}_i$ , il parametro  $\sigma_i$  associato minimizzando il funzionale:

$$E(\sigma) = E_{local}^2(\sigma) + E_p(\sigma) = \frac{1}{L^2} \frac{\sum_j d_j \phi_\sigma(\|\mathbf{x}_j - \mathbf{c}_i\|) \left( \frac{g_i(\mathbf{x}_j)}{\|\nabla g_i(\mathbf{x}_j)\|} \right)^2}{\sum_j d_j \phi_\sigma(\|\mathbf{x}_j - \mathbf{c}_i\|)} + \frac{C}{\sigma^2}, \quad (3.18)$$

dove  $C$  è un parametro positivo definito dall'utente e  $L$  è la diagonale della bounding box di  $X$ .

Se supponiamo che in un intorno di  $\mathbf{c}_i$  i punti di  $X$  siano generati da un processo stocastico in modo tale che le loro posizioni medie siano distribuite lungo l'approssimante locale  $g_i(\mathbf{x}) = 0$  e le distanze da tale curva abbiano una distribuzione normale, allora  $E_{local}^2(\sigma)$  è proporzionale al numero di bit necessari a descrivere i punti di  $X$  in prossimità di  $\mathbf{c}_i$ . Osserviamo che  $E_{local}^2(\sigma)$  tende a 0 per  $\sigma \rightarrow 0$ . D'altra parte, il secondo termine aggiunge una penalizzazione per valori di  $\sigma$  troppo piccoli, che comporterebbero un numero di centri  $M$  troppo elevato.

In Figura 3.6 (tratta da [25]) si mostra l'andamento di  $E_{local}(\sigma)$  (a sinistra) e quello di  $E(\sigma)$  per un centro fissato durante la ricostruzione dello Stanford Bunny. Il parametro  $C$ , scelto dall'utente, agisce sul risultato finale della ricostruzione controllando la regolarità della superficie: più grande è  $C$  più la superficie è liscia e senza dettagli. In [25] si propone di scegliere  $C = (T_{SA}L)^2$ , dove  $L$  è definito come in (3.18) mentre  $T_{SA}$  è dell'ordine di  $10^{-5}$ .

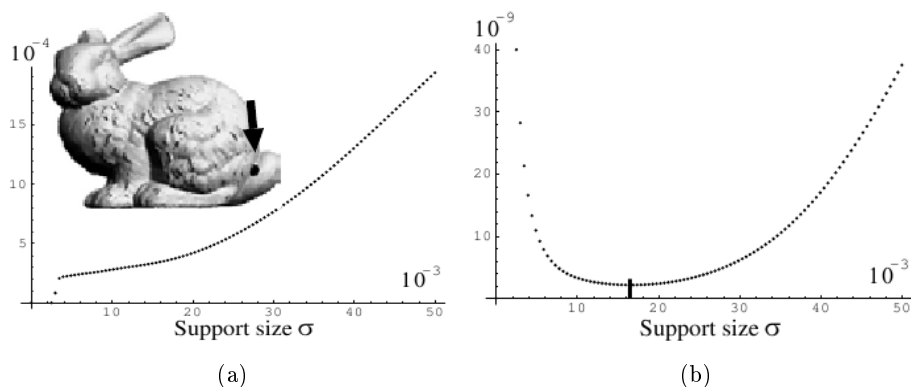


Figura 3.6: Andamento dell'errore per il centro indicato dalla freccia in una ricostruzione dello Stanford Bunny.

Per ogni centro  $\mathbf{c}_i$  si determina  $\sigma_i$  come  $\sigma_i = \arg \min E(\sigma)$ . La minimizzazione dell'errore  $E(\sigma)$  è un problema di minimizzazione monodimensionale che può essere risolto con metodi iterativi di uso comune. Nella nostra implementazione dell'algoritmo ricorriamo al metodo di Brent, un metodo che ha il vantaggio di non richiedere la valutazione delle derivate del funzionale da minimizzare (per ulteriori dettagli su questo metodo si veda [3]). Si noti che per ogni valutazione del funzionale  $E$  per un dato valore di  $\sigma$  è necessario effettuare tutta la procedura descritta nel paragrafo precedente per determinare l'approssimante locale  $g_i(\mathbf{x})$ ; il calcolo del valore ottimale di  $\sigma$  per ogni centro si rivela quindi abbastanza oneroso dal punto di vista computazionale.

### 3.4.3 Selezione dei centri di approssimazione

L'insieme dei centri di approssimazione fa parte delle incognite del problema. In [25] viene proposta una procedura basata su una ricerca randomizzata al fine di selezionare una parte dei punti della nuvola iniziale  $X$  come centri di approssimazione. Ogni volta che un centro  $\mathbf{c}_i$  viene selezionato si calcolano la corrispondente approssimante locale  $g_i(\mathbf{x})$  e  $\sigma_i$  seguendo le procedure descritte sopra.

I centri sono determinati in modo da garantire che tutti i punti della nuvola vengano inclusi in qualche  $\sigma_i$ -intorno di un centro. Per selezionare il minor numero possibile di centri pur garantendo questa proprietà, si introduce un *coefficiente di sovrapposizione* associato a ogni punto della nuvola  $X$  e definito come

$$v(\mathbf{x}_j) = \sum_{i=1}^M \phi_{\sigma_i}(\|\mathbf{x}_j - \mathbf{c}_i\|). \quad (3.19)$$

La procedura per la selezione dei centri si basa su una ricerca randomizzata sui punti di  $X$  con minimo coefficiente di sovrapposizione; essa è riassunta nell'Algoritmo 2, dove  $T_{overlap}$  è una soglia fissata dall'utente che viene fissata a 1.5 in [25].

---

#### Algorithm 2 SELEZIONA CENTRI( $X, N$ )

---

1. **Start**
  2. calcola i pesi  $d_j$  associati a ogni  $\mathbf{x}_j \in X$ ;
  3.  $v(\mathbf{x}_j) = 0$  per ogni  $j = 1, \dots, N$ ;
  4. **while** ( $\exists \mathbf{x}_j \in X : v(\mathbf{x}_j) < T_{overlap}$ ) **do**
  5. scegli in modo casuale  $m$  punti  $\mathbf{x}_j$  tali che  $v(\mathbf{x}_j) < T_{overlap}$  ( $m = 15$  in [25]);
  6. trova il punto  $\mathbf{x}_{min}$  con minimo  $v(\mathbf{x}_j)$  fra quelli selezionati al punto precedente;
  7. seleziona  $\mathbf{x}_{min}$  come centro  $\mathbf{c}_i \in C$ ; poni  $v(\mathbf{x}_{min}) = T_{overlap}$ ;
  8. trova il parametro  $\sigma_i$  e l'approssimazione locale  $g_i(\mathbf{x})$  associate a  $\mathbf{c}_i$  con le procedure descritte nei paragrafi 3.4.1 e 3.4.2;
  9. aggiorna  $v(\mathbf{x}_j)$  per ogni  $\mathbf{x}_j \in X \setminus C$  aggiungendo  $\phi_{\sigma_i}(\|\mathbf{x}_j - \mathbf{c}_i\|)$
  10. **end while**
  11. **End**
- 

### 3.4.4 Calcolo dei pesi scalari $\lambda_i$

Una volta effettuata la selezione di tutti i centri, le uniche quantità incognite nella (3.12) sono i pesi scalari  $\lambda_i$ . Poniamo  $\mathbf{\Lambda} = [\lambda_1, \dots, \lambda_M]$ ; il calcolo di  $\mathbf{\Lambda}$  viene effettuato minimizzando la quantità:

$$[\lambda_1, \dots, \lambda_M] = \arg \min \left( E_{global}(\mathbf{\Lambda})^2 + T_{reg} \|\mathbf{\Lambda}\|^2 \right), \quad (3.20)$$



dove

$$E_{global}(\Lambda) = \frac{\sum_{j=1}^N d_j f(\mathbf{x}_j)^2}{L^2 \sum_{j=1}^N d_j} \quad (3.21)$$

misura l'errore di approssimazione mentre il secondo termine, dove si è posto

$$\|\Lambda\| = \sqrt{\frac{1}{M} \sum_{i=1}^M \left(\frac{\lambda_i}{\sigma_i}\right)^2}, \quad (3.22)$$

introduce una *penalizzazione* per regolarizzare la soluzione in presenza di rumore nei dati.  $T_{reg}$  è un parametro definito dall'utente e posto uguale a  $10^{-5}$  in [25].

Come illustrato in [25], imponendo ora

$$\frac{\partial(E_{global}(\Lambda)^2 + T_{reg}\|\Lambda\|^2)}{\partial\Lambda} = 0$$

nel problema di minimizzazione quadratica (3.20), si ottiene il sistema lineare con  $M$  equazioni in  $M$  incognite

$$(A + T_{reg}D)\Lambda = \mathbf{b}, \quad (3.23)$$

dove i coefficienti della matrice simmetrica  $A \in \mathbb{R}^{M \times M}$ , della matrice diagonale  $D \in \mathbb{R}^{M \times M}$  e del termine noto  $\mathbf{b} \in \mathbb{R}^M$  sono dati rispettivamente da:

$$\begin{cases} A_{ij} = \frac{\sum_{k=1}^N d_k \phi_{\sigma_i}(\|\mathbf{x}_k - \mathbf{c}_i\|) \phi_{\sigma_j}(\|\mathbf{x}_k - \mathbf{c}_j\|)}{L^2 \sum_{k=1}^N d_k} \\ D_{ii} = \frac{1}{M} \left(\frac{1}{\sigma_i}\right)^2 \\ b_i = \frac{\sum_{k=1}^N d_k \phi_{\sigma_i}(\|\mathbf{x}_k - \mathbf{c}_i\|) (-f(\mathbf{x}_k)|_{\Lambda=\mathbf{0}})}{L^2 \sum_{k=1}^N d_k}. \end{cases}$$

Si noti che  $A$  è una matrice sparsa in cui il numero di elementi non nulli per ogni riga cresce per valori crescenti del parametro  $T_{overlap}$ . Per la risoluzione del sistema lineare (3.23) valgono le stesse osservazioni effettuate per il sistema (3.8).

Osserviamo che imponendo la (3.20) stiamo rinunciando alla proprietà di interpolazione richiesta precedentemente per la funzione  $F$  e cerchiamo invece un'approssimazione regolare della superficie.

### 3.5 Conciliare i vantaggi di RBF a supporto globale e a supporto compatto: una sfida possibile?

I due algoritmi presentati in questo capitolo si propongono lo stesso scopo, superare le limitazioni legate al tipo particolare di RBF impiegata, attraverso

due strade opposte: il primo con una procedura di tipo *coarse-to-fine* di analisi multiscala della nuvola di punti; il secondo con un procedimento mono-livello ma adattativo. Nel Capitolo 4 presentiamo una serie di prove sperimentali effettuate allo scopo di indagare il comportamento di entrambi gli algoritmi.

Come vedremo, l'algoritmo adattativo presenta ancora molti dei limiti legati all'uso di RBF compatte (ad esempio, la funzione approssimante  $F$  non ha supporto globale) e richiede un'attenta gestione di un elevato numero di parametri che dipendono dal campione ricostruito; in certe condizioni e per un settaggio dei parametri appropriato esso può rivelarsi più robusto rispetto al rumore nella misurazione dei punti.

D'altro canto, il metodo di ricostruzione multi-livello permette di superare tutte le limitazioni dell'interpolazione mono-livello presentate nel Paragrafo 3.2.3. In particolare, esso ricostruisce superfici con buchi, presenta una migliore robustezza al rumore e fornisce una funzione interpolante il cui supporto ha carattere globale (nel senso che è esteso a tutta la bounding box). Nel prossimo capitolo verranno proposte anche alcune soluzioni per aumentare l'efficienza del metodo; in particolare verrà proposto un metodo per la creazione adattativa dei livelli allo scopo di aggiungere, per ogni livello successivo, punti solo nelle regioni spaziali in cui il livello precedente fornisce una cattiva ricostruzione locale della superficie.

# 4

## Prove sperimentali. Innovazioni. Prospettive

Nel Capitolo 3 abbiamo presentato due algoritmi per la ricostruzione di superfici basati su RBF a supporto compatto proposti in [24] e [25]. Entrambi gli algoritmi sono stati implementati in C++ e testati per la ricostruzione di alcune nuvole di punti. I risultati dei test effettuati sono riportati in questo capitolo.

Nella Sezione 4.2 presentiamo i risultati di alcune sperimentazioni effettuate sull'algoritmo descritto nella Sezione 3.4. I test hanno mostrato che questo algoritmo, pur presentando delle caratteristiche accattivanti, non sembra in grado di superare le limitazioni che l'uso di RBF a supporto compatto comporta accanto ai numerosi vantaggi.

Nella Sezione 4.3 presentiamo le evidenze sperimentali ricavate dall'utilizzo dell'algoritmo multi-scala descritto nelle Sezioni 3.1, 3.2 e 3.3: i test mostrano come la ricostruzione multi-scala permetta di ricostruire efficacemente anche superfici con buchi, di ottenere una funzione approssimante  $F$  che ha le proprietà di una “distanza con segno” dalla superficie e di mantenere i vantaggi provenienti dall'uso di RBF a supporto compatto. Nella Sezione 4.4 è presentato il comportamento dell'algoritmo al variare di alcuni parametri e si mette in evidenza come sia possibile introdurre alcune modifiche per ottimizzare le prestazioni in casi di oggetti dalla forma complessa. Infine, nella Sezione 4.5 proponiamo un nuovo metodo adattativo di costruzione dei livelli che tiene conto della qualità dell'approssimazione fornita dal livello precedente in ogni regione spaziale per creare il livello successivo.

## 4.1 Sperimentazioni. Casi test

Tutti i risultati riportati in questo capitolo sono effettuati su nuvole di punti ri-scalate affinché il più piccolo parallelepipedo contenente tutti i punti di  $X$  abbia lo spigolo più lungo di lunghezza 1 e un vertice nell'origine. Per tutte le prove abbiamo utilizzato come funzione di base, analogamente a quanto riportato in [25] e in [24], la RBF a supporto compatto di Wendland  $\phi(r) = (1 - r)_+^4(4r + 1)$ .

Per quanto riguarda i campioni testati, rappresentati in Tabella 4.1, le nuvole dello Stanford Bunny e del Dragone Cinese sono tratti Large Geometric Models Archive dei Georgia Institute of Technology [http://www.cc.gatech.edu/projects/large\\_models/](http://www.cc.gatech.edu/projects/large_models/); gli altri modelli provengono da immagini acquisite presso l' Ospedale di Borgo Trento di Verona e fornite al laboratorio MOX del Politecnico di Milano dal dott G.B. Luciani e il Dott G. Puppini (Università di Verona) e da immagini acquisite presso il Policlinico di Milano e fornite al MOX dal Dott. M. Domanin, dal Dott. G Biondetti e dal Dott. L Forzenigo (Policlinico di Milano e Università di Milano) all'interno del progetto MACARENA@MOX (principal investigator Christian Vergara, Università degli Studi di Bergamo). Le nuvole di punti sono impiegate per gentile concessione di Elena Faggiano del laboratorio MOX del Politecnico di Milano, che ha effettuato l'elaborazione delle immagini per l'ottenimento di tutti i modelli.

## 4.2 Test sull'algoritmo adattativo

Diversi test sono stati effettuati per sondare il comportamento dell'algoritmo adattativo presentato in Sezione 3.4; i risultati dei nostri esperimenti sembrano indicare che, sebbene l'algoritmo presenti delle caratteristiche interessanti, la selezione adattativa dei supporti per le funzioni di base non sia sufficiente, da sola, a superare le limitazioni caratteristiche della ricostruzione con RBF a supporto compatto. L'implementazione e l'utilizzo di questo algoritmo si rivelano più complessi di quello multi-scala descritto nelle Sezioni 3.1, 3.2 e 3.3; in particolare, l'utente deve settare per ogni ricostruzione un elevato numero di parametri che influiscono notevolmente sulla velocità e sulla qualità della ricostruzione.

Ad esempio, la ricerca della taglia ottimale per il supporto avviene mediante minimizzazione del funzionale (3.18) come descritto nel Paragrafo 3.4.2. Il metodo impiegato per la minimizzazione richiede di impostare gli estremi di un intervallo  $[a, b]$  all'interno del quale viene ricercato il minimo; la taglia ottimale può variare molto a seconda della regione, della densità dei punti, della forma della superficie: determinare  $a, b$  in modo corretto, garantendo la convergenza del metodo verso il valore cercato ma evitando intervalli troppo grandi, richiede un'attenta calibrazione.

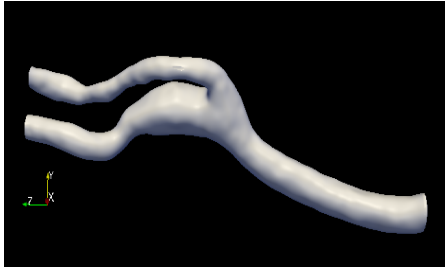
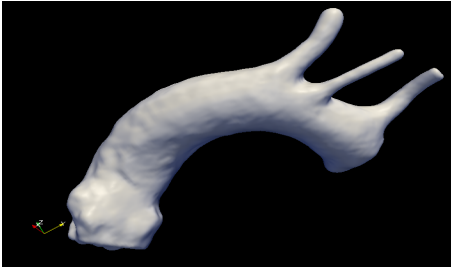
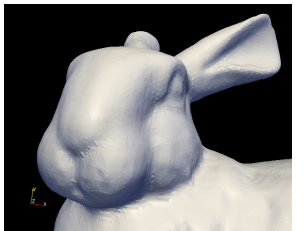
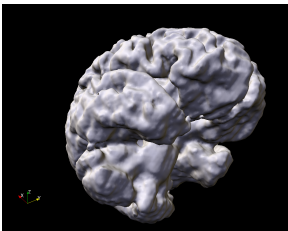

|  |  |   |
|--|--|---|
|   |  |   |
| <p>Biforcazione<br/>Carotidea<br/><math>N = 1251</math></p>                        | <p>Arco<br/>Aortico<br/><math>N = 9591</math></p>                                  |   |
|  |  |  |
| <p>Stanford Bunny<br/><math>N = 34833</math></p>                                   | <p>Cervello<br/><math>N = 46650</math></p>   | <p>Dragone Cinese<br/><math>N = 437645</math></p>                                   |

Tabella 4.1: Campioni utilizzati per i test presentati in questo capitolo.

Un altro svantaggio è costituito dal supporto compatto dell'approssimante globale: analogamente a quanto mostrato in Figura 3.4(b) per l'interpolazione mono-livello con RBF a supporto compatto, anche in questo caso il supporto di  $F$  è costituito da un sottile “strato” attorno alla superficie da approssimare. Inoltre, lo spessore di questo strato varia da punto a punto e la sua larghezza minima non è nota a priori, bensì è determinata dal minimo valore calcolato per  $\sigma_i$ . Questo fatto può rivelarsi problematico durante la successiva fase di poligonalizzazione, che spesso risulta più lenta (e meno precisa) rispetto al caso dell'algoritmo multi-scala.

Un'altra limitazione tipica dell'uso di RBF a supporto compatto che il metodo presenta è data dall'impossibilità di ricostruire parti della superficie in cui siano presenti buchi o strappi. Abbiamo osservato che l'algoritmo adattativo è efficace nel ricostruire fessure e buchi di piccole dimensioni, ma non sempre riesce a ricostruire grandi buchi. Ad esempio, in tutte le ricostruzioni dello Stanford Bunny che abbiamo effettuato i buchi nella parte inferiore del coniglio non sono stati ricuciti.

Test sono stati effettuati anche sulla sensibilità al rumore del metodo e sulla ricostruzione di nuvole con brusche variazioni della densità spaziale

dei punti (simili a quelli presentati nel Paragrafo 4.3.2), ma non sono stati osservati risultati soddisfacenti. Osserviamo infine che la procedura per la selezione dei centri non si presta a parallelizzazione in quanto la selezione dei centri deve avvenire in modo sequenziale.

Un aspetto interessante di questo algoritmo è che attraverso il controllo del parametro  $T_{SA}$  è possibile ottenere un effetto di *smoothing* della superficie ricostruita. Riportiamo a titolo di esempio in Tabella 4.2 tre ricostruzioni diverse dello Stanford Bunny per tre diversi valori del parametro  $T_{SA}$ ; per ciascuna ricostruzione sono riportati la taglia massima e minima dei supporti delle RBF e il numero di centri selezionati.



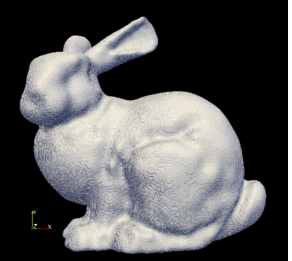
|  |  |   |
|--|--|---|
|  |  |  |
| $T_{SA} = 10^{-6}$   | $T_{SA} = 10^{-5}$   | $T_{SA} = 10^{-4}$  |
| $\sigma_{min} = 0.00945$<br>$\sigma_{max} = 0.19582$<br>$M = 28790$                | $\sigma_{min} = 0.01491$<br>$\sigma_{max} = 0.19498$<br>$M = 6680$                 | $\sigma_{min} = 0.03615$<br>$\sigma_{max} = 0.21098$<br>$M = 1362$                  |

Tabella 4.2: Tre ricostruzioni dello Stanford Bunny con algoritmo adattativo. Sono riportate la taglia massima del supporto  $\sigma_{max}$ , la taglia minima  $\sigma_{min}$  e il numero di centri selezionati  $M$ .

In conclusione, le nostre prove non hanno mostrato evidenze sperimentali che questo algoritmo sia in grado di superare le limitazioni legate all'uso di RBF a supporto compatto. In compenso, aspetti come la selezione adattativa della taglia del supporto delle RBF, l'introduzione di algoritmi di ricerca randomizzata e l'uso di un termine regolarizzante nel sistema lineare (3.23) restano spunti interessanti.

### 4.3 Test sull'algoritmo multi-scala

In questa sezione presentiamo i risultati di prove sperimentali effettuate sull'algoritmo multi-scala descritto nelle Sezioni 3.1, 3.2 e 3.3. Nella sezione 3.5 abbiamo messo in evidenza come la scelta di RBF a supporto compatto o a supporto globale in un algoritmo di ricostruzioni portino a risultati in generale

molto diversi, e come ciascuna scelta presenti dei vantaggi e degli inconvenienti: un'attenzione particolare è stata rivolta, nell'esecuzione delle nostre prove sperimentali, alla messa in luce di eventuali abilità dell'algoritmo di conciliare i vantaggi di entrambi i tipi di RBF.

In Figura 4.1 sono visibili i passi di una ricostruzione dello Stanford Bunny ( $N = 34833$ ). La ricostruzione è stata effettuata con 6 livelli; questo numero è sufficiente affinché l'ultimo livello comprenda tutti i punti della nuvola. Il numero di punti compresi in ciascun livello è riportato in Tabella 4.3.

| RICOSTRUZIONE     |              |
|-------------------|--------------|
| $c = 0.75, n = 6$ |              |
| Livello           | Numero punti |
| 1                 | 8            |
| 2                 | 47           |
| 3                 | 231          |
| 4                 | 1261         |
| 5                 | 8676         |
| 6                 | 34833        |

Tabella 4.3: Punti inclusi in ogni livello  $X_n$  per la ricostruzione dello Stanford Bunny ( $N = 34833$ ).

È possibile effettuare alcune osservazioni che verranno riprese nel seguito. Il numero di punti cresce esponenzialmente con l'aumentare dei livelli. La ricostruzione ottenuta al sesto e ultimo livello corrisponde a un'interpolazione su tutti i punti della nuvola e rappresenta quindi l'approssimazione migliore che può essere ottenuta. Tuttavia già al livello 5 utilizzando solo 8676 punti si ottiene una ricostruzione visivamente accettabile. Questo suggerisce che l'aumento di punti fra il livello 5 e il livello 6 possa essere troppo brusco; torneremo su questo punto nella Sezione 4.5.

### 4.3.1 Proprietà della funzione interpolante

In Figura 4.2 sono visibili i valori della funzione  $F$  per la ricostruzione multi-scala i cui dati sono riassunti nella Tabella 4.3. La superficie corrisponde all'insieme di livello zero della funzione, che assume valori diversi da zero in tutto il resto del dominio. Nonostante la ricostruzione sia effettuata mediante RBF a supporto locale quindi, il supporto della funzione approssimante è esteso a tutta la bounding box della nuvola di punti.

$F$  assume valori positivi all'interno della nuvola e negativi all'esterno. Inoltre, dalla figura si evince che essa rispetta (qualitativamente) una proprietà di *monotonia*: a punti  $\mathbf{x} \in \mathbb{R}^3$  più lontani dalla superficie corrispondono valori  $|F(\mathbf{x})|$  più elevati ( $|\cdot|$  rappresenta il valore assoluto). Questa proprietà,

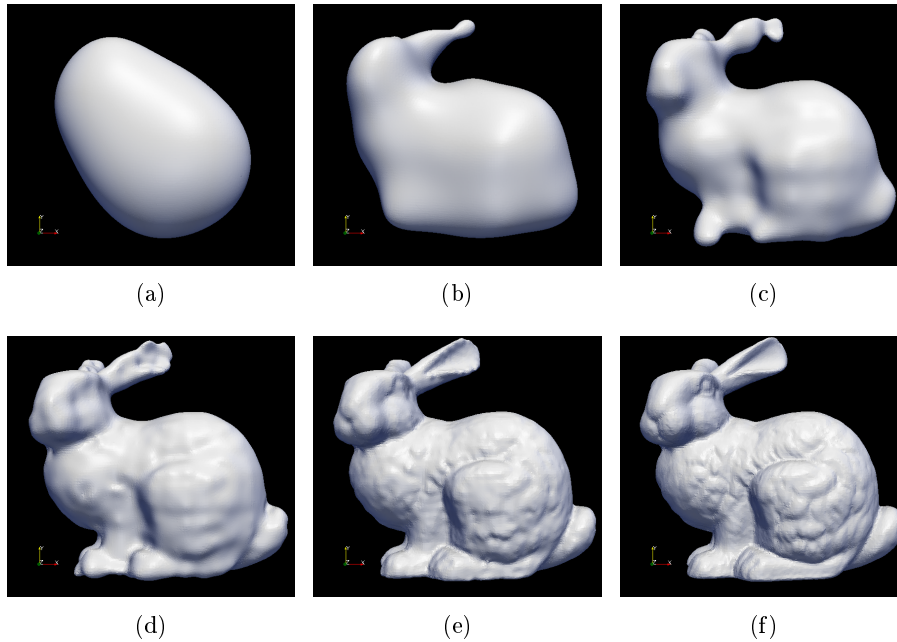


Figura 4.1: Passi della ricostruzione multi-scala dello Stanford Bunny su 6 livelli; l'ultimo livello corrisponde a una interpolazione di tutti i punti della nuvola.

che è stata riscontrata in tutti i test effettuati, si rivela preziosa rendendo più stabili e robuste operazioni di post-processing come la costruzione di mesh poligonali.

Grazie alla proprietà di monotonia di cui sopra, negli esperimenti che abbiamo effettuato abbiamo potuto stimare la qualità della ricostruzione  $F$  valutando l'errore di interpolazione

$$Err = \sum_{\mathbf{x}_j \in X} |F(\mathbf{x}_j)|; \quad (4.1)$$

osserviamo che questa quantità potrebbe non essere significativa per ricostruzioni basate su RBF a supporto compatto con la procedura tradizionale.

### 4.3.2 Robustezza rispetto alla distribuzione spaziale dei punti

In Figura 4.3 è riportato un particolare della ricostruzione multi-scala. Sono visibili i due "buchi" nella nuvola di punti nella parte inferiore del modello che sono stati richiusi grazie all'approccio multi-scala (si confronti questa figura con la 3.4(a)). Un altro test è stato effettuato per verificare il comportamento della ricostruzione in presenza di brusche variazioni della densità spaziale dei punti. La ricostruzione è stata applicata alla nuvola rappresentata in Figura 4.4(a), dove nella parte posteriore la frequenza spaziale dei punti è  $1/30$



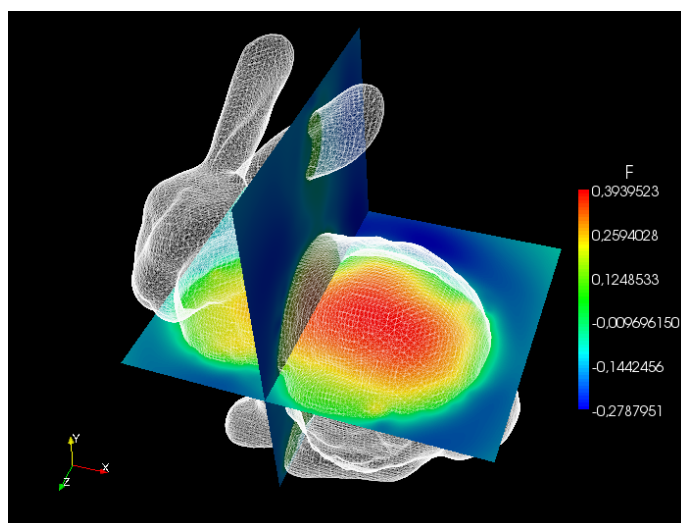


Figura 4.2: Insiemi di livello per  $F$  derivante dalla ricostruzione riassunta in Tabella 4.3 lungo due sezioni della bounding box.

di quella nella parte anteriore. Il risultato è mostrato in Figura 4.4(b). In situazioni di questo genere possono verificarsi delle piccole instabilità locali, che scompaiono aumentando leggermente il parametro  $c$  (fino a  $c = 1$ ).

### 4.3.3 Sensibilità al rumore

Abbiamo testato l'algoritmo multi-scala su nuvole in cui era stato precedentemente aggiunto un rumore bianco con distribuzione gaussiana. Abbiamo constatato che il metodo non è molto robusto rispetto alla presenza di rumore nelle coordinate dei punti: in particolare, basta introdurre un rumore con deviazione standard di  $\sigma = 0,01$  su ogni punto (nella nuvola ri-scalata

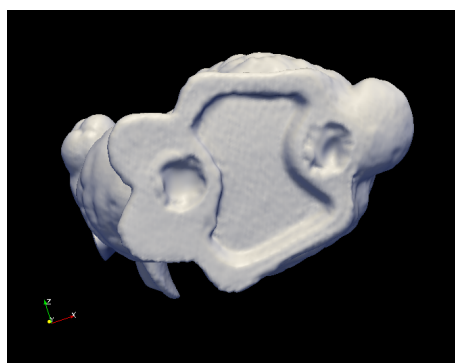


Figura 4.3: Particolare dalla ricostruzione riassunta in Tabella 4.3: la ricostruzione è effettuata anche per regioni con “buchi” nella nuvola di punti.

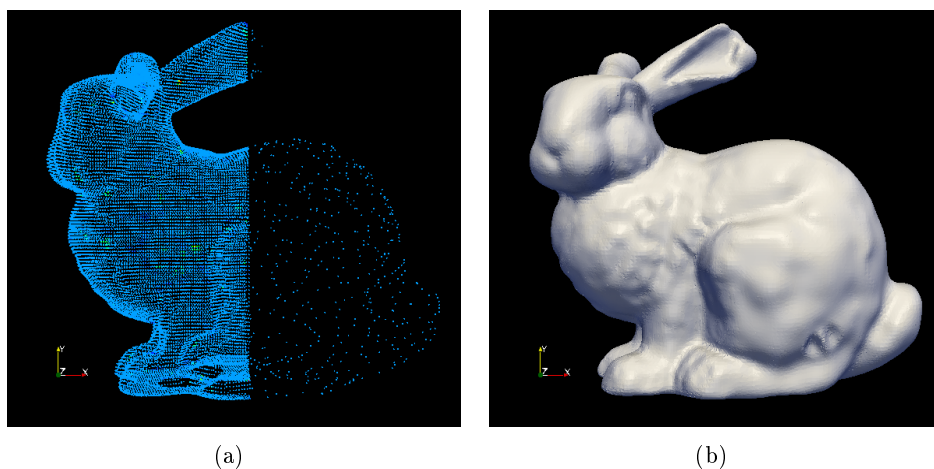


Figura 4.4: Ricostruzione multi-scala di una nuvola con forte discontinuità della distribuzione spaziale dei punti.

come illustrato all'inizio del capitolo) per ottenere risultati visivamente inaccettabili. All'origine di questo fenomeno vi è la natura del metodo, basato su interpolazione diretta dei punti di ogni livello; l'operazione di *clustering* dei punti di  $X$  in celle effettuata a ogni livello non è sufficiente a compensare gli effetti del rumore.

Tutt'altro comportamento è manifestato dal rumore aggiunto nei vettori normali. Anche l'aggiunta di un rumore gaussiano con  $\sigma = 0,1$  permette di ottenere risultati visivamente identici dall'originale. Non solo, ma, come osservato in [24] è anche possibile eliminare una parte delle normali (ponendo ciascuna delle componenti uguale a 0, fino al 2% delle normali) e ottenere un risultato visivamente soddisfacente, benchè non identico alla ricostruzione senza perturbazione.

#### 4.4 Algoritmo multi-scala: dipendenza da parametri; alcune innovazioni.

Durante i nostri test abbiamo riscontrato che non sempre l'algoritmo nella sua versione originale presenta dei risultati soddisfacenti. In questa sezione presentiamo alcuni risultati che hanno messo in luce la dipendenza del comportamento del metodo al variare del parametro  $c$  e altri che hanno permesso l'introduzione di alcune modifiche per migliorare le performances in casi particolari.

#### 4.4.1 Supporto delle funzioni di base

In [24] il raggio del supporto  $\sigma$  delle funzioni di base è costante per ogni livello  $k$  ed è proporzionale alla diagonale  $D_k$  delle celle. In particolare, si suggerisce la scelta di  $\sigma = cD_k$  con  $c = 0.75$ , senza ulteriori giustificazioni. In figura 4.6 è riportato l'andamento dell'errore medio per la ricostruzione di tre diverse nuvole di punti: Stanford Bunny ( $N = 34833$ ), Arco Aortico ( $N = 9591$ ) e Biforcazione Carotidea ( $N = 1251$ ). Abbiamo arrestato ogni ricostruzione al livello precedente quello corrispondente all'interpolazione completa (per cui, ovviamente, si ha  $E_{medio} \approx 0$ ). Per tutte le ricostruzioni l'errore minimo (il cui valore dipende dalla frequenza spaziale dei punti) si ha in un intorno di  $c = 0.7$ ; inoltre, si nota nel grafico un andamento quasi perfettamente lineare dell'errore per valori crescenti di  $c$ . Non disponiamo di giustificazioni teoriche di questo fenomeno che si ripresenta con sorprendente regolarità.

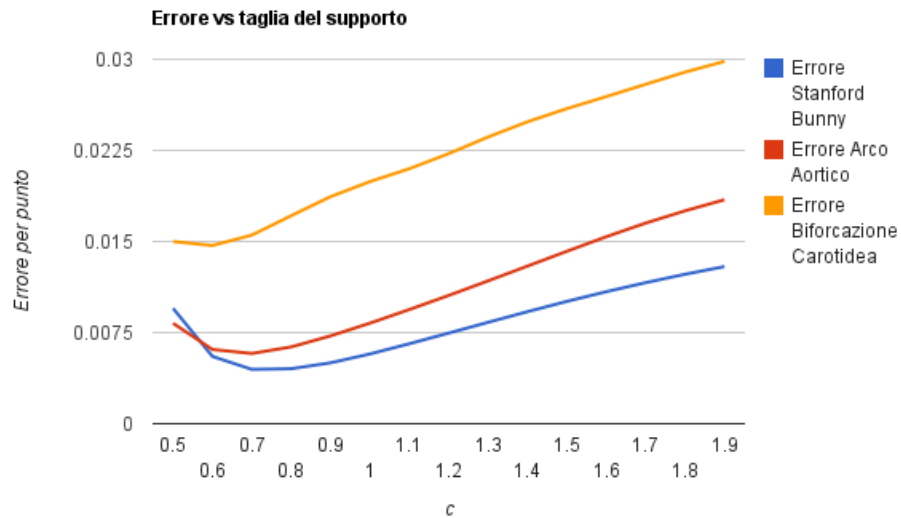


Figura 4.5: Errore di ricostruzione al variare del parametro  $c$ .

Nella maggior parte dei casi quindi abbiamo mantenuto per i nostri test  $c = 0.75$ . In casi in cui la distribuzione dei punti della nuvola  $X$  è molto bassa, aumentare il valore di  $c$  può migliorare la qualità visiva della ricostruzione.

#### 4.4.2 Modalità selezione livelli

L'algoritmo per la creazione di un livello descritto nel Paragrafo 3.3.1 prescrive di suddividere la bounding box della nuvola in una serie di celle effettuando una divisione ricorsiva e poi di aggiungere al livello un punto per ogni cella, dato dal centro geometrico dei punti di  $X$  contenuti nella cella.

Questa operazione comporta che ai livelli più bassi la superficie non solo non approssimi bene i dettagli più fini, ma non rifletta nemmeno quelli macroscopici. Si pensi, ad esempio, alla ricostruzione di un oggetto di forma sferica con dei particolari a piccole scale sulla superficie (ad esempio delle increspature). La modalità di selezione dei punti fa sì che ai livelli più bassi non solo i dettagli fini non siano riprodotti, ma anche che la sfera ricostruita sia più piccola dell'originale, in quanto i punti interpolati non giacciono sulla superficie dell'oggetto ma al suo interno.

Per ovviare a questo inconveniente (soprattutto in vista dell'implementazione di un algoritmo adattativo per la costruzione dei livelli, come specificato nella Sezione 4.5), abbiamo sperimentato altre due modalità di selezione dei punti. La prima prevede che una volta calcolato il centro geometrico  $\mathbf{x}_G$  dei punti di  $X$  inclusi in una cella  $\mathcal{G}$  venga trovato e inserito nel livello il punto di  $X$  più vicino a  $\mathbf{x}_G$ . La seconda modalità prevede semplicemente di selezionare un punto aleatorio fra i punti di  $X$  inclusi in una data cella. L'algoritmo 1 è stato quindi sostituito dall'algoritmo 3. Il parametro  $ms$ , impostato all'inizio della ricostruzione, permette di selezionare la modalità di selezione dei punti ( $ms = 1$  per selezionare il centro di gravità dei punti in ogni cella,  $ms = 2$  per il punto di  $X$  più vicino al centro di gravità,  $ms = 3$  per selezionare un punto casuale di  $X$  appartenente alla cella).

Ovviamente se il numero  $n$  di livelli è abbastanza elevato affinché nell'ultimo livello non vi sia più di un punto di  $X$  per cella, allora la ricostruzione coincide con un'interpolazione dei dati e non vi è differenza fra le modalità di selezione. Al contrario, se la ricostruzione si arresta a un livello intermedio le modalità di selezione 2 e 3 permettono di migliorare notevolmente la qualità della ricostruzione. In Figura 4.6 mostriamo l'andamento dell'errore in funzione del parametro  $c$  per la ricostruzione dell'arco aortico rappresentato in figura 3.5 ( $N = 9591$ ) con  $n = 4$  livelli (l'ultimo livello ha 1291 punti). Si nota come le modalità 2 e 3 comportino una notevole riduzione dell'errore medio per punto. In alcuni esperimenti la modalità 3 ha portato a risultati visivi poco soddisfacenti. Riteniamo quindi che la modalità 2 sia la migliore.

#### 4.4.3 Offset sui primi livelli

Le prove che abbiamo effettuato hanno messo in evidenza che la qualità della ricostruzione ai livelli più bassi può influire notevolmente sulla qualità di quella ottenuta con i livelli finali. Nella ricostruzione di alcuni oggetti dalla struttura complessa abbiamo rilevato come la forma della superficie ricostruita ai primissimi livelli fosse completamente diversa dalla forma dell'oggetto da ricostruire a causa del numero di punti troppo esiguo. In alcuni casi abbiamo osservato come saltare i primissimi livelli permetta di ottenere non solo una ricostruzione più rapida, ma anche più accurata.

La procedura per la creazione dei livelli descritta nel Paragrafo 3.3.1 prevede una divisione della bounding box in  $8^k$  celle per il  $k$ -esimo livello:

---

**Algorithm 3** CREA LIVELLI( $X, N, n, ms$ )

---

1. **Start**
  2. determina un parallelepipedo  $\Omega_0$  contenente tutti i punti di  $X$ ;
  3. **for**  $k = 1, \dots, n$  **do**
  4.   **switch**( $ms$ )
  5.     **case** 1:
  6.       per ogni cella del livello  $k$ -esimo  $\{\Omega_k^i\}_{i=1, \dots, 8^k}$ , trova il centro di gravità  $\mathbf{x}_k^i$  dei punti di  $X$  in essa contenuti;
  7.       per ogni  $\mathbf{x}_k^i$ , calcola il vettore normale di norma unitaria  $\mathbf{n}_k^i$  corrispondente come media dei vettori normali assegnati ai punti di  $X$  contenuti in  $\Omega_k^i$ ;
  8.       **break**;
  9.     **case** 2:
  10.       per ogni cella del livello  $k$ -esimo  $\{\Omega_k^i\}_{i=1, \dots, 8^k}$ , trova il punto  $\mathbf{x}_k^i \in X$  più vicino al centro di gravità dei punti di  $X$  in essa contenuti;
  11.       per ogni  $\mathbf{x}_k^i$ , seleziona il vettore normale di norma unitaria  $\mathbf{n}_k^i \in N$  corrispondente;
  12.       **break**;
  13.     **case** 3:
  14.       per ogni cella del livello  $k$ -esimo  $\{\Omega_k^i\}_{i=1, \dots, 8^k}$ , seleziona un punto aleatorio  $\mathbf{x}_k^i \in X$  fra tutti i punti di  $X$  contenuti nella cella;
  15.       per ogni  $\mathbf{x}_k^i$ , seleziona il vettore normale di norma unitaria  $\mathbf{n}_k^i \in N$ ;
  16.       **break**;
  17.     **end switch**
  18.    $X_k = \{\mathbf{x}_k^1, \dots, \mathbf{x}_k^{8^k}\}$ ,  $N_k = \{\mathbf{n}_k^1, \dots, \mathbf{n}_k^{8^k}\}$ ;
  19. **end for**
  20. **End**
- 

8 celle per il primo livello, 64 per il secondo, 512 per il terzo e così via. Per ricostruire modelli dalla forma complessa a volte è meglio introdurre una partizione in  $8^{k+\text{offset}}$  celle al  $k$ -esimo livello: per esempio, per  $\text{offset}=2$  la bounding box viene divisa in 512 celle per il primo livello, in 4096 celle per il secondo, e così via. In Tabella 4.4 riportiamo il risultato di due ricostruzioni effettuate su un modello dalla forma complessa e dall'elevato numero di punti, il Dragone Cinese ( $N = 437645$ ); la prima è effettuata con  $\text{offset}=0$ , nella seconda (il cui risultato finale è riportato in Figura 4.7) abbiamo posto  $\text{offset}=2$ . A parità di punti impiegati nel livello finale, l'accuratezza è notevolmente migliorata nella seconda ricostruzione.

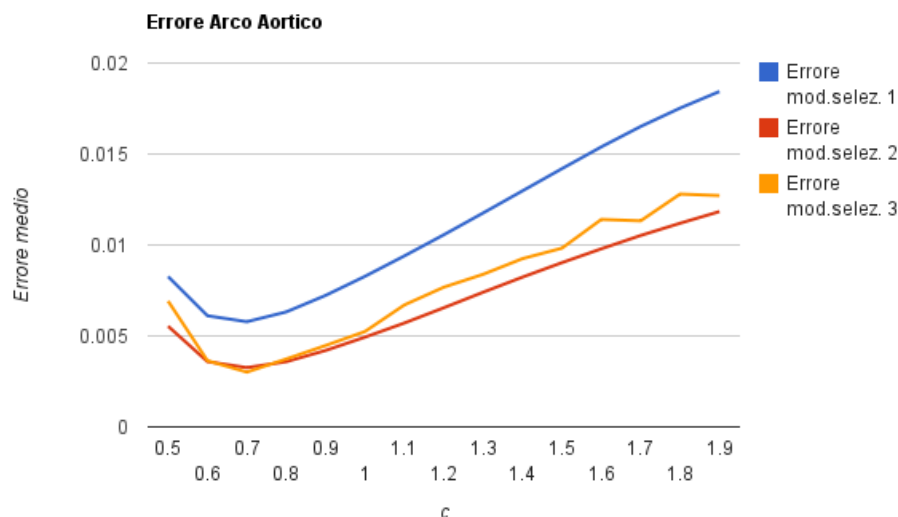


Figura 4.6: Errore di ricostruzione per diverse modalità di selezione dei punti in ogni livello.

## 4.5 Creazione adattativa dei livelli

Entrambi gli algoritmi presentati nel Capitolo 3 hanno la finalità di superare i limiti legati alla ricostruzione RBF tradizionale effettuata con RBF a supporto compatto. L'algoritmo presentato nella Sezione 3.4 introduce una strategia adattativa, quello descritto nelle Sezioni 3.1, 3.2 e 3.3 si basa su un approccio multiscala. I risultati dei test descritti sopra indicano che il secondo algoritmo si rivela più efficace e robusto; in questa sezione presentiamo un algoritmo concepito a partire dall'algoritmo multiscala proposto in [24] che introduce un approccio adattativo per selezionare i punti di ogni livello.

In Figura 4.8 è rappresentata la ricostruzione multi-scala di un cervello umano a partire da una nuvola di  $N = 46650$  punti. Sotto ogni immagine è riportato il numero di punti inclusi nel livello corrispondente. Osserviamo che la ricostruzione effettuata al livello 4 ( $N_4 = 2894$ ) è visivamente insoddisfacente, mentre quella con 5 livelli è accettabile ( $N_5 = 25955$ ). È lecito chiedersi se il risultato sarebbe accettabile anche per un livello "intermedio" senza effettuare un aumento di punti così importante fra il quarto e il quinto livello.

Inoltre, come abbiamo illustrato nella Sezione 3.3, la ricostruzione al  $k$ -esimo livello avviene in due tempi:

- calcolo degli  $N_k$  punti che fanno parte del livello;

|         |          | RICOSTRUZIONE 1   |         |          | RICOSTRUZIONE 2   |         |  |
|---------|----------|---|---------|----------|---|---------|--|
|         |          | Drago, $N = 437645$<br>$\sigma = 0.75$<br>6 livelli<br>offset=0 |         |          | Drago, $N = 437645$<br>$\sigma = 0.75$<br>4 livelli<br>offset=2 |         |  |
| Livello | N. punti | $\sigma$  | Errore  | N. punti | $\sigma$  | Errore  |  |
| 1       | 8        | 0.4885  | 26887.3 | 281      | 0.1221  | 7698.35 |  |
| 2       | 53       | 0.2442  | 23015.2 | 1366     | 0.0610  | 3905.67 |  |
| 3       | 281      | 0.1221  | 12669.3 | 6549     | 0.0305  | 1736.36 |  |
| 4       | 1366     | 0.0610  | 6180.95 | 37645    | 0.0152  | 541.61  |  |
| 5       | 6549     | 0.0305  | 2885.68 | -        | -   | -       |  |
| 6       | 37465    | 0.0152  | 1149.81 | -        | -   | -       |  |

Tabella 4.4: Ricostruzioni con e senza offset per il numero di celle dei livelli. Il risultato finale della Ricostruzione 2 è riportato in figura 4.7.

- calcolo dell'offset  $o_k(\mathbf{x})$  e della funzione approssimante

$$F_k(\mathbf{x}) = F_0 + \sum_{j=1}^k o_j(\mathbf{x}).$$

Osserviamo che l'informazione raccolta durante la costruzione dei livelli precedenti viene impiegata soltanto nella seconda fase, il calcolo dell'offset  $o_k(\mathbf{x})$  attraverso la risoluzione dell'equazione (3.10). È possibile tenere conto dell'informazione dei livelli precedenti anche durante la prima fase, per selezionare i punti in modo più efficiente?

A partire da tali considerazioni abbiamo sperimentato una procedura adattativa per la selezione dei punti dei livelli. Per il  $k$ -esimo livello, si procede in questo modo:

1. vengono calcolati la nuvola di punti  $X_k$  e l'insieme di vettori normali associato  $N_k$  come descritto nell'Algoritmo 3;
2. per ogni punto  $\mathbf{x}_i^k \in X_k$ , si valuta l'errore associato alla ricostruzione fino al livello  $k - 1$ :  $|F_{k-1}(\mathbf{x}_i^k)| = |\sum_{j=0}^{k-1} o_j(\mathbf{x}_i^k)|$ ;
3. vengono conservati in  $X_k$  solo i punti con l'errore più alto.

Osserviamo che con questa procedura affinché i punti del  $k$ -esimo livello siano selezionati in modo ottimale è auspicabile avere un'approssimazione più accurata possibile della superficie anche nei livelli precedenti: per questo,

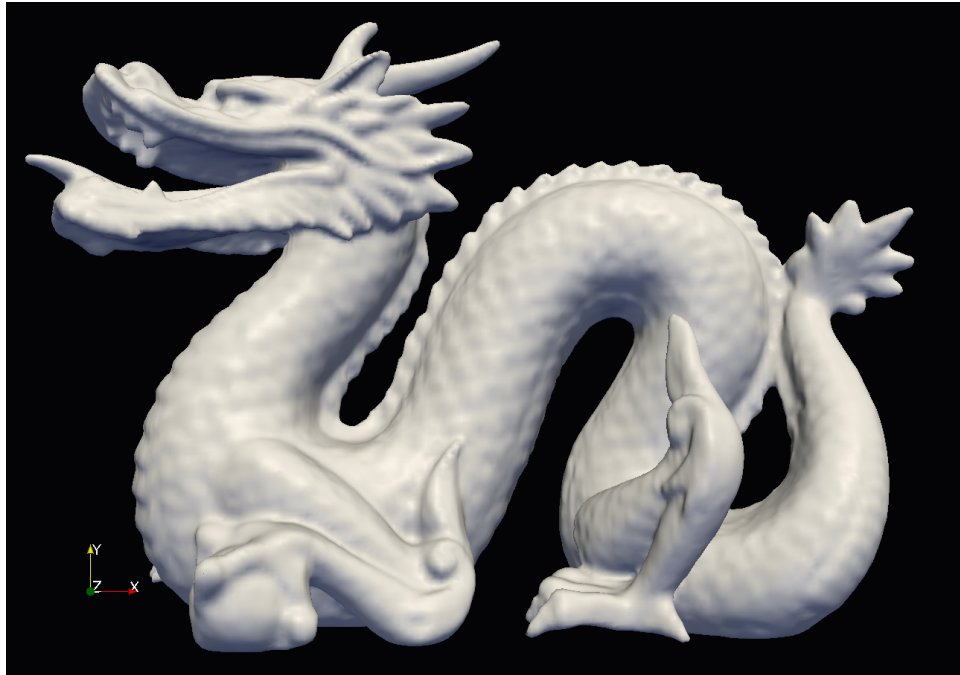


Figura 4.7: Ricostruzione di drago ( $N = 437645$ ).

nell'Algoritmo 3 è preferibile utilizzare la seconda modalità di selezione dei punti (punto della superficie più vicino al centro geometrico della cella).

Con questa modalità di selezione, i punti di ogni livello sono un sottoinsieme dei punti della nuvola originale  $X$ . Potrebbe apparire lecito chiedersi, allora, se per ogni livello non basti valutare  $|F_{k-1}(\mathbf{x}_i)|$  per ogni  $\mathbf{x}_i \in X$  e selezionare i punti con l'errore maggiore per formare  $X_k$ , rendendo così inutile la suddivisione in celle. In questo modo in realtà si rischierebbe di prendere tutti i punti di un livello dalla stessa regione dello spazio, portando a una ricostruzione troppo accurata per quella regione senza apportare miglioramenti nel resto della superficie.

L'algoritmo così concepito permette di:

- costruire livelli in modo adattativo, aggiungendo punti alla ricostruzione solo dove necessario;
- garantire che i punti di ogni livello abbiano una distribuzione spaziale regolare;
- selezionare punti diversi per livelli diversi: infatti, la selezione basata sull'errore garantisce che i punti  $\mathbf{x}_i \in X$  che fanno parte dei livelli dal primo al  $(k-1)$ -esimo, per cui  $F_{k-1}(\mathbf{x}_i) = 0$ , non vengano selezionati per il  $k$ -esimo livello.



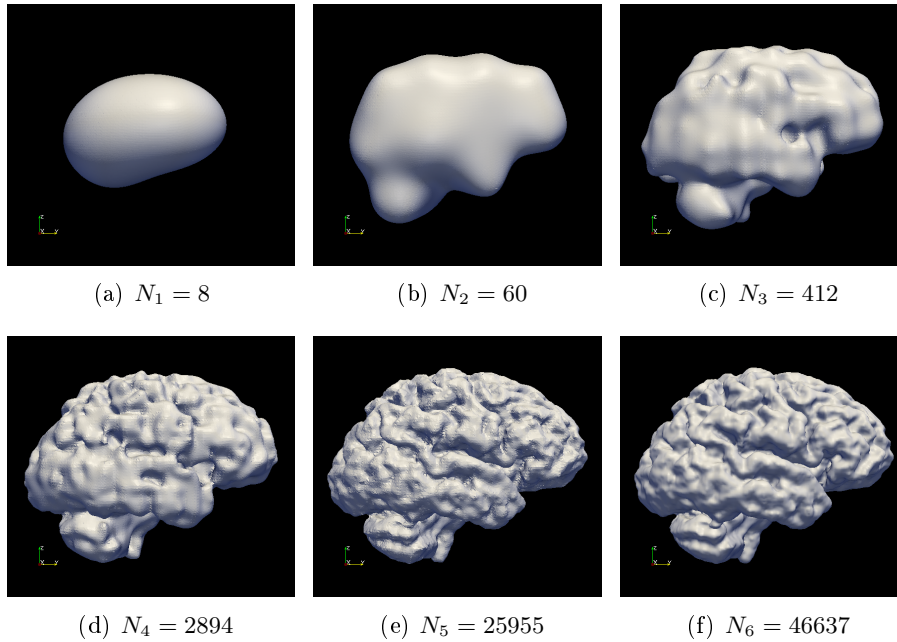


Figura 4.8: Passi della ricostruzione multi-scala di un cervello su 6 livelli ( $N = 46650$ ). Sotto ogni immagine è indicato il numero di punti  $N_k$  appartenenti al livello corrispondente.

#### 4.5.1 Selezione adattativa del numero di punti

La scelta adattativa dei punti del  $k$ -livello richiede che la ricostruzione fino al  $(k - 1)$ -esimo livello sia abbastanza accurata da indicare in quali parti del dominio è prioritario effettuare dei raffinamenti. Per questa ragione è opportuno procedere con una selezione non adattativa dei punti nei primi livelli e introdurre la selezione adattativa solo nei livelli più elevati. Il livello a cui introdurre la selezione adattativa dipende dal numero di punti della nuvola  $X$ ; nella maggior parte delle nostre esperienze introdurla a partire dal quarto o dal quinto livello si è rivelata una scelta opportuna.

Resta da chiarire come avvenga la selezione adattativa dei punti al  $k$ -esimo livello. Supponiamo che dopo la partizione spziale siano presenti  $\tilde{N}_k$  punti in  $X_k$ . Abbiamo sperimentato due modalità diverse per selezionare solo i punti con l'errore più grande.

- **Numero di punti costante:** a ogni livello vengono aggiunti gli  $M$  punti con l'errore più alto, con  $M < \tilde{N}_k$  costante e definito a priori. Con questa scelta viene creato un alto numero di livelli, tutti contenenti  $M$  punti; è quindi opportuno che il raggio  $\sigma$  del supporto delle funzioni di base non diventi troppo piccolo (nella versione originale dell'algoritmo esso è dimezzato ogni volta che si crea un nuovo livello).

In particolare, potrebbe essere necessario mantenere un valore costante di  $\sigma$  partire da un certo livello in poi.

- **Percentuale di punti costante:** con questa modalità di scelta, al  $k$ -esimo livello viene selezionata solo una proporzione fissa degli  $\tilde{N}_k$  punti di  $X_k$ . Con questa modalità l'algoritmo è molto simile alla sua versione originale (in particolare, si ha un aumento esponenziale del numero dei punti all'aumentare dei livelli), salvo che il numero di punti di ogni livello viene ridotto di una percentuale fissa.

La prima modalità permette di ottenere un numero di livelli più elevato ma facilmente calcolabili; inoltre non si assiste più all'aumento esponenziale di punti fra un livello e l'altro per cui la soglia di dettaglio a cui si può arrestare la ricostruzione è più facilmente controllabile. D'altro canto, se un numero troppo elevato di livelli viene aggiunto e c'è l'esigenza di mantenere costante il valore di  $\sigma$ , si assiste spesso a un rallentamento della convergenza dell'errore. La seconda permette di ottenere dei livelli in cui il numero di punti aumenta in modo del tutto analogo alla versione originale, ma a ogni livello sono presenti meno punti.

#### 4.5.2 Prove sperimentali sulla selezione adattativa dei punti

Osserviamo che non è facile confrontare due ricostruzioni il cui numero di livelli e/o il numero di punti per livello differiscono di molto. Meno punti per livello permettono un calcolo più rapido, ma aumentano il numero di livelli necessari; questo a sua volta rischia di allungare i tempi di valutazione della funzione approssimante  $F$ .

Un possibile criterio per effettuare confronti (in modo qualitativo) fra ricostruzioni molto diverse è confrontare gli errori che si ottengono a parità di *punti complessivi*, ossia quando la somma di tutti i punti presenti in tutti i livelli dal primo al  $k$ -esimo è simile per le due ricostruzioni. In Tabella 4.5 è riportato il confronto fra l'andamento dell'errore per la ricostruzione multi-scala originale e quella con selezione adattativa dei punti per i livelli a partire dal quinto livello. Si nota la notevole diminuzione dell'errore a parità di punti totali della ricostruzione  $N_{tot} = \sum_{k=1}^n N_k$  (ad esempio fra l'errore del livello 5 della prima ricostruzione e quello del livello 7 della seconda). In tutte le prove sperimentali effettuate abbiamo rilevato che la ricostruzione adattativa richiede leggermente più tempo di quella non adattativa, ma i tempi di calcolo restano comparabili e il divario diminuisce al crescere di  $N$ . Ad esempio, per lo Stanford Bunny la ricostruzione non adattativa è svolta in 6 secondi, quella adattativa in 9 secondi. Per il Cervello, la ricostruzione non adattativa è eseguita in 21 secondi, quella adattativa in 24.

|         |          | RICOSTRUZIONE 1   |         |          | RICOSTRUZIONE 2  |         |  |
|---------|----------|---|---------|----------|--|---------|--|
|         |          | Cervello( $N = 46650$ )<br>$\sigma = 0.75$<br>6 livelli<br>NO selez. adattativa |         |          | Cervello( $N = 46650$ )<br>$\sigma = 0.75$<br>7 livelli<br>Selez. adattativa da liv 5<br>Punti per livello adatt.=8000 |         |  |
| Livello | N. punti | $\sigma$  | Errore  | N. punti | $\sigma$   | Errore  |  |
| 1       | 8        | 0.5563  | 4626.62 | 8        | 0.5563   | 1981.41 |  |
| 2       | 60       | 0.2781  | 2961.2  | 60       | 0.2781   | 1393.13 |  |
| 3       | 412      | 0.1390  | 1700.13 | 412      | 0.1390   | 937.038 |  |
| 4       | 2894     | 0.0695  | 804.102 | 2894     | 0.0695   | 444.684 |  |
| 5       | 25955    | 0.0347  | 180.089 | 8000     | 0.034770   | 186.2   |  |
| 6       | 46637    | 0.0173  | 0.06046 | 8000     | 0.0173   | 86.3005 |  |
| 7       | -        | -   | -       | 8000     | 0.0086   | 39.7091 |  |

Tabella 4.5: Ricostruzioni originale e con selezione adattativa dei punti per modello di cervello (cf. Figura 4.8).

## 4.6 Un'applicazione in campo medico

Abbiamo testato le capacità di ricostruzione di dati incompleti dell' algoritmo multi-scala con un tipo di nuvole di punti utilizzato di frequente in campo medico. Tali nuvole sono caratterizzate da una distribuzione dei punti su sezioni parallele (*slices*) molto distaccate fra loro su cui è tracciato il contorno di un corpo. Ad esempio, questo tipo di nuvole è il risultato tipico proveniente dalla segmentazione di immagini acquisite con TAC o IRM. Abbiamo ricostruito il modello rappresentato in Figura 4.9(a) con l'algoritmo multi-scala; in particolare, non è stato necessario fare uso della selezione adattativa dei punti in quanto la dimensione della nuvola è piuttosto esigua, e data la bassa distribuzione spaziale dei punti abbiamo posto  $c = 1.75$ ) La

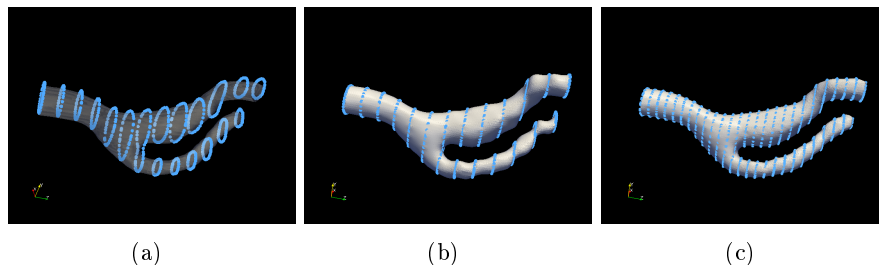


Figura 4.9: Ricostruzione di una nuvola di punti su sezioni parallele.

nuvola è stata rilevata sulla superficie di una biforcazione carotidea (lunghezza tot. 42 mm). Una prima ricostruzione, riportata in Figura 4.9(b), è stata effettuata con punti rilevati a sezioni distanti 3mm fra loro ( $N = 912$ ). Si nota come il ramo più grosso sia ben ricostruito, mentre quello più piccolo, il cui diametro ha una lunghezza analoga a quella della distanza fra le sezioni, presenti delle deformazioni "a bolla" lungo tutta la sua lunghezza. Una seconda ricostruzione, utilizzando una nuvola con dati su *slices* ogni 1.5mm, è raffigurata in Figura 4.9(c) ( $N = 1814$ ): le oscillazioni sono scomparse e la ricostruzione è eseguita fedelmente in tutte le parti della biforcazione.

Questo esempio dimostra la versatilità dell'algoritmo proposto (entro certi limiti sulla distribuzione spaziale dei punti) anche per nuvole di piccole dimensioni e con punti distribuiti in modo non uniforme.

## 4.7 Conclusioni.Prospettive

Lo scopo degli algoritmi che abbiamo descritto nel Capitolo 3 è quello di proporre delle soluzioni originali per la ricostruzione di superfici basata sull'approccio RBF, che permettano di conciliare semplicità di implementazione, rapidità di calcolo e robustezza della ricostruzione. Entrambi gli algoritmi prevedono l'uso di RBF a supporto compatto.

L'implementazione effettuata in C++ dei due metodi e i numerosi test effettuati hanno permesso di mettere in luce vantaggi e svantaggi di ciascuno. L'idea alla base del primo metodo, che prevede un approccio adattativo per scegliere il supporto di ogni funzione di base, si è rivelata insufficiente a superare le limitazioni tipiche della ricostruzione con RBF a supporto compatto tradizionale.

Molto più promettente è apparso l'approccio multi-scala del secondo algoritmo, che si è rivelato versatile e robusto. Con le soluzioni proposte nella sezione 4.5 abbiamo reso più efficiente l'algoritmo introducendo, fra l'altro, una procedura adattativa per il calcolo della funzione approssimante a ogni scala. Tale procedura si è rivelata molto efficace nell'aumentare l'accuratezza del metodo e nella riduzione del numero di punti necessario.

Diverse sono le strade possibili per la ricerca futura. In particolare, uno studio sistematico dei parametri che regolano la procedura adattativa per la selezione dei punti sarebbe opportuno su nuvole di punti di dimensioni molto diverse fra loro, per massimizzare la riduzione dell'errore.

Uno dei limiti più ingombranti della ricostruzione RBF è l'incapacità di ricostruire fedelmente superfici con angoli, spigoli e punte (le cosiddette *sharp features*). Un'interessante prospettiva è data dalla forma particolare di funzione approssimante proposta in [25] e in [24], che comprende un termine approssimante locale  $g_i(\mathbf{x})$ . In tutti gli algoritmi che abbiamo presentato le funzioni approssimanti locali sono costituite da funzioni quadratiche; potrebbe rivelarsi proficuo sperimentare forme diverse di approssimante lo-

cale, analogamente a quanto accade per i metodi basati sulla partizione dell'unità, scegliendo la forma di approssimante locale che più si adatta alle caratteristiche di una data regione spaziale.

Un limite che siamo riusciti a superare solo parzialmente è la sensibilità al rumore dei dati. Migliorare questo aspetto renderebbe il metodo con approccio multi-scala e adattativo davvero completo e utilizzabile per una vastissima categoria di applicazioni.

Un altro spunto interessante è costituito dalla possibilità di introdurre tecniche di calcolo parallelo per accelerare il calcolo; una scelta implementativa che richiederebbe un attento ripensamento di una parte importante degli algoritmi, ma che potrebbe portare grandi benefici. Ad esempio, l'uso di tecniche di calcolo su GPU potrebbe rivelarsi molto efficace nei casi in cui la ricostruzione venga effettuata a fini visivi: in questo caso, infatti, i dati calcolati sarebbero direttamente disponibili sulla scheda grafica per la visualizzazione.



# Conclusioni

La concezione di algoritmi efficienti per la ricostruzione di superfici a partire da dati sparsi è un compito complesso. Tale attività è una delle tappe principali del processo che permette di ottenere il modello virtuale di un oggetto a partire da una sua scansione. Molti metodi attualmente in uso permettono di ottenere buone prestazioni rispetto a un dato criterio (ad esempio, l'accuratezza, la velocità di calcolo, la sensibilità al rumore nei dati trattati o la capacità di ricostruire fedelmente le cosiddette *sharp features*). Il fatto che la ricostruzione di superfici sia sempre più impiegata in ambiti molto diversi e con le finalità più disparate fa sorgere però l'esigenza di algoritmi completi e versatili, che garantiscano una ricostruzione di qualità nel caso più generale possibile. Molto lavoro resta da fare in questo senso.

Gli algoritmi di cui ci siamo occupati in questo lavoro di tesi permettono la ricostruzione di superfici implicite, sfruttando un modello per la descrizione di superfici estremamente versatile ed efficiente. Tali algoritmi fanno uso di tecniche basate su funzioni di base a simmetria radiale (RBF), uno degli strumenti più efficaci introdotti recentemente in letteratura per l'interpolazione multivariata. In particolare, l'introduzione di un approccio multi-scala in uno degli algoritmi proposti permette di sfruttare tutti i vantaggi in termini di efficienza di calcolo derivanti dall'uso di RBF a supporto compatto e, allo stesso tempo, di ottenere ricostruzioni robuste e affidabili tipiche delle ricostruzioni effettuate mediante RBF a supporto globale.

La procedura adattativa che abbiamo introdotto per il calcolo delle approssimazioni a scale differenti ha permesso di ottenere descrizioni più accurate e, allo stesso tempo, più sintetiche della superficie desiderata, con un tempo di calcolo della ricostruzione comparabile a quello dell'algoritmo originale.

Restano migliorabili le prestazioni dell'algoritmo in alcune situazioni particolari, come la ricostruzione di dati affetti da rumore o la ricostruzione di *sharp features*, che saranno oggetto della ricerca futura.

Tale risultato originale è stato ottenuto grazie a un'attività complessa che ha toccato tutte le fasi tipiche della ricerca per le scienze applicate, dall'analisi teorica dei metodi alla loro implementazione per poi concludere con la validazione numerica, per perseguire la nostra missione inconfessabile di Ingegneri Matematici: rubare il fuoco sacro della conoscenza dall'Empireo

dell'analisi matematica e farne dono al mondo imperfetto e mortale delle applicazioni ingegneristiche.





## Appendice A: Strutture dati spaziali

Numerosi algoritmi per il trattamento di un insieme di dati geometrici  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^d$  (così come quelli presentati in questo lavoro) richiedono di processare molte volte richieste come:

- trovare il punto di  $X$  più vicino a un dato punto  $\mathbf{y} \in \mathbb{R}^d$ ;
- trovare i  $K$  punti di  $X$  più vicini a un dato punto  $\mathbf{y} \in \mathbb{R}^d$ ;
- trovare tutti i punti che giacciono in una certa regione dello spazio, come un ipercubo o una sfera.

Poter processare in modo efficiente richieste di questo tipo costituisce un aspetto di importanza fondamentale ai fini dell'implementazione di tali algoritmi. A questo fine è possibile impiegare delle strutture dati appositamente concepite, la maggior parte delle quali si basa sull'organizzazione dei dati dell'insieme  $X$  (o di informazioni su di esso) in una struttura ad albero. In generale, i nodi dell'albero corrispondono a regioni dello spazio e/o ai punti di  $X$  in esse contenuti; la regione corrispondente a ogni nodo interno viene suddivisa in regioni più piccole associate ai figli di quel nodo: in questo modo è possibile, dopo la costruzione della struttura, ottenere rapidamente informazioni sulla localizzazione nello spazio dei punti di  $X$ .

Descriviamo in particolare la struttura utilizzata nell'ambito dell'implementazione degli algoritmi di questo lavoro, il *kd-tree*: essa consiste in un albero binario in cui i punti di  $X$  sono organizzati nei nodi a partire dal valore delle loro coordinate. Non descriviamo in dettaglio i concetti legati agli alberi di ricerca e all'analisi della complessità temporale degli algoritmi, per cui rimandiamo, ad esempio, a [5].

## A.1 Kd-tree

Un Kd-tree è una struttura di dati a forma di albero binario, in cui è memorizzato un insieme di  $N$  punti di uno spazio  $K$ -dimensionale. Per semplicità, supporremo nel seguito che lo spazio sia  $\mathbb{R}^K$  dotato di una base ortogonale  $\{\mathbf{e}_1, \dots, \mathbf{e}_K\}$  (ad esempio la base canonica), anche se non vi sono restrizioni particolari sullo spazio (a patto che esso sia finito-dimensionale) né sulla base. Il concetto alla base dei Kd-tree è una generalizzazione di quello utilizzato per gli alberi binari di ricerca.

I punti dell'insieme sono associati ai nodi dell'albero. Ogni nodo interno  $\nu$  dell'albero introduce una partizione dello spazio rispetto a una coordinata  $\mathbf{e}_i$ ,  $1 \leq i \leq K$ . Il valore della coordinata lungo  $\mathbf{e}_i$  di tutti i punti associati ai nodi del sottoalbero sinistro di  $\nu$  è minore (o uguale) di quello di  $\nu$ , il valore di quelli associati al sottoalbero destro è maggiore.

Se  $K = 1$  la coordinata di ogni punto può essere interpretata come *chiave* associata a un dato: in questo caso, il Kd-tree è strutturato esattamente come un albero di ricerca binario. Se  $K > 1$ , allora si effettua un ciclo sulle  $K$  coordinate per introdurre le partizioni: i nodi alla profondità  $p$  inducono una partizione lungo la coordinata  $\mathbf{e}_{(p \bmod K)+1}$ . Ad esempio, per punti in  $\mathbb{R}^2$  (con un sistema di riferimento cartesiano usuale) nel 2d-tree i nodi con profondità pari inducono partizioni lungo l'asse delle ascisse, quelli situati a profondità dispari partizionano lo spazio lungo l'asse delle ordinate.

Esistono numerose varianti di questa struttura dati; fra le diverse scelte possibili, è possibile associare un punto dell'insieme a ogni punto dell'albero, oppure associare i punti dell'insieme soltanto alle foglie. Nel primo caso la partizione introdotta da un nodo  $\nu$  lungo la coordinata  $\mathbf{e}_i$  è basata sul valore di quella coordinata per il punto associato al nodo. Nel secondo caso il valore discriminante per la coordinata  $\mathbf{e}_i$  può essere scelto in diversi modi, tenendo conto delle coordinate dei punti dell'insieme  $X$  in modo arbitrario.

Riportiamo un esempio con l'organizzazione di una serie di punti bidimensionali (a cui sono associati nomi di città) rappresentati nella tabella A.1; nelle figure A.1 e A.2 sono raffigurati i kd-tree ricavati con i due approcci descritti sopra.

| Punto   | $x$ | $y$ |
|---------|-----|-----|
| Toronto | 60  | 75  |
| Buffalo | 80  | 65  |
| Denver  | 5   | 45  |
| Chicago | 35  | 40  |
| Atlanta | 85  | 15  |
| Omaha   | 25  | 35  |
| Miami   | 90  | 5   |
| Mobile  | 50  | 10  |

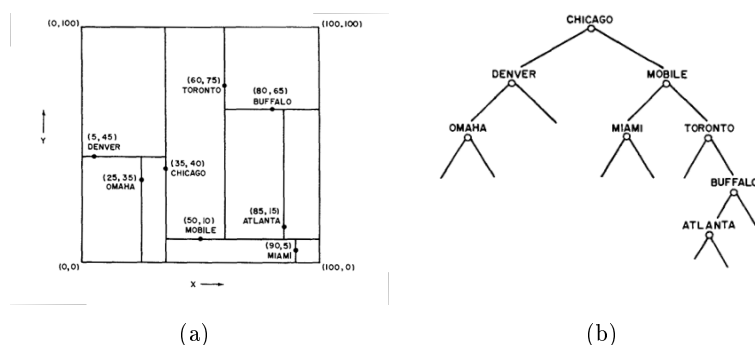


Figura A.1: Kd-tree su un insieme di dati bidimensionali: elementi memorizzati in tutti i nodi.

Geometricamente, la partizione dello spazio in 2 regioni rispetto a una coordinata  $i$  introdotta da ogni nodo corrisponde a un *iperpiano* ( $K - 1$ )-dimensionale di vettore normale  $\mathbf{e}_i$ ; nella versione canonica di Kd-tree gli iperpiani sono allineati con gli elementi di una base ortogonale dello spazio (nell'esempio riportato nelle figure A.1 e A.2 essi sono costituiti da rette parallele agli assi cartesiani); in letteratura sono proposte tipi di partizioni differenti.

Osserviamo che il Kd-tree è una struttura ad albero *binario*, indipendentemente dal numero di dimensioni  $K$ ; altre strutture dati spaziali non hanno questa caratteristica. Ad esempio, i *quadtree*, specializzati per gestione di dati bidimensionali, sono alberi con 4 figli per nodo, gli *octree*, concepiti per la gestione di dati tridimensionali, hanno 8 figli per nodo: a ogni nodo è associato un cubo che viene suddiviso ricorsivamente in 8 cubi dimezzando ogni lato (cf. Figura A.3<sup>1</sup>).

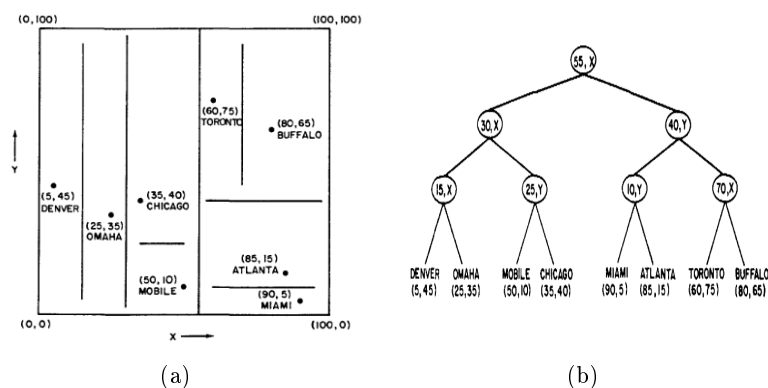


Figura A.2: Kd-tree su un insieme di dati bidimensionali: elementi memorizzati esclusivamente sulle foglie.

<sup>1</sup>Questa figura è tratta da <http://en.wikipedia.org>.

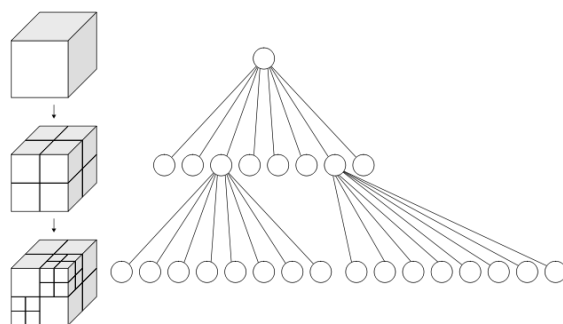


Figura A.3: Partizione dello spazio e octree.

### A.1.1 Kd-tree per l'implementazione di algoritmi di ricostruzione di superfici.

Nell'implementazione degli algoritmi presentati nei Capitoli 3 e 4 si è fatto uso di Kd-tree per aumentare l'efficienza della gestione di dati geometrici. In particolare, fra le differenti versioni quella che abbiamo utilizzato<sup>2</sup> impiega una struttura analoga a quella della figura A.2, in cui i punti dell'insieme  $X$  sono associati solo alle foglie dell'albero. Nel seguito sono descritte più in dettaglio le procedure per la costruzione dell'albero e per la ricerca di punti in una regione spaziale per Kd-tree basati esclusivamente su questa scelta; inoltre, sono effettuate alcune ipotesi verificate nel caso della nostra implementazione ma non necessariamente nel caso generale (ad esempio, quella di conoscere già tutti i punti dell'insieme  $X$  al momento della creazione dell'albero); per un'esposizione più generale sui Kd-tree rimandiamo a [26].

### A.1.2 Costruzione dell'albero

La nozione fondamentale nella costruzione di un kd-tree è quella di *partizione*: con questo termine, come abbiamo visto, si indica la suddivisione ricorsiva dello spazio (o più propriamente di una *bounding box* di  $X$ ) in regioni sempre più piccole mediante iper-piani. A ogni livello dell'albero si introduce una partizione rispetto a una coordinata differente. Nell'esempio bidimensionale delle figure A.1 e A.2, i nodi ai livelli 1,3,5 introducono partizioni rispetto al valore di  $x$ , quelli ai livelli 2,4 introducono partizioni rispetto al valore di  $y$ . In un albero  $k$ -dimensionale, una partizione rispetto a una certa coordinata è introdotta ogni  $k$  livelli.

Supporremo nel seguito di avere tutti i punti dell'insieme  $X$  a disposizione al momento della creazione dell'albero. Allora è possibile effettuare la costruzione dell'albero ricorsivamente, in modo da garantire che esso sia

<sup>2</sup>La libreria, disponibile per Matlab e per C++ è reperibile all'indirizzo <http://sites.google.com/site/andreatagliasacchi/software/matlabkd-treelibrary>

*bilanciato*.<sup>3</sup> La scelta del nodo rispetto a cui effettuare ogni partizione può determinare se l'albero è bilanciato o no. In particolare, al livello corrispondente a una partizione rispetto alla coordinata  $\mathbf{e}_i$ , scegliere l'*elemento mediano* dei punti ancora da inserire ordinati secondo il valore di  $\mathbf{e}_i$  assicura che l'albero finale sia bilanciato.

Dato un insieme di punti  $P$  e fissato  $i \in [1, K]$ , supponiamo che la procedura  $[P_1, P_2, l]=\text{SPLIT}(P, \mathbf{e}_i)$  produca due insiemi di punti  $P_1$  e  $P_2$  tali che

- la cardinalità di  $P_1$  differisca al più di 1 da quella di  $P_2$ ;
- $P_1 \cup P_2 = P$ ;
- per ogni  $\mathbf{x} \in P_1$ ,  $\mathbf{x} = (x_1, \dots, x_K)$  e per ogni  $\mathbf{y} \in P_2$ ,  $\mathbf{y} = (y_1, \dots, y_K)$  allora  $x_i \leq l < y_i$ , dove  $l$  è il valore dell' $i$ -esima coordinata dell'*elemento mediano* dei punti di  $P$  ordinati per valori crescenti della stessa coordinata .

Questa procedura necessita di stimare l'elemento mediano di un insieme di dati rispetto a una chiave, un'operazione che per un insieme di  $N$  elementi richiede una complessità temporale  $O(N)$ ; alternativamente, è possibile estrarre a sorte dall'insieme una collezione di  $J$  elementi, con  $J \ll K$ , e utilizzare l'elemento mediano di questi ultimi; questo metodo in teoria non garantisce di ottenere un albero perfettamente bilanciato, ma si rivela spesso efficace.

---

**Algorithm 4** CREAKDTREE( $P, depth$ )

---

1. **Start**
  2. **if** ( $P$  contiene solo un punto) **then**
  3.     **return** una foglia associata a questo punto;
  4. **else**
  5.     calcola  $i = (depth \bmod K)$ ;
  6.      $[P_1, P_2, l]=\text{SPLIT}(P, \mathbf{e}_i)$ ;
  7.      $\nu_{left}=\text{CREAKDTREE}(P_1, depth + 1)$ ;
  8.      $\nu_{right}=\text{CREAKDTREE}(P_2, depth + 1)$ ;
  9.     Crea un nodo  $\nu$  che memorizza il valore di  $l$ , rendi  $\nu_{left}$  figlio sinistro di  $\nu$ ,  $\nu_{right}$  figlio destro di  $\nu$ ;
  10.    **return**  $\nu$ ;
  11. **end if**
  12. **End**
- 

<sup>3</sup>Ricordiamo che un *nodo* si dice bilanciato se l'altezza del suo sottoalbero sinistro differisce al più di 1 da quella del suo sottoalbero destro; un *albero* è bilanciato se e solo se tutti i suoi nodi sono bilanciati.

L'algoritmo 4 permette di costruire un Kd-tree come quello mostrato in figura A.2(a). Osserviamo che la complessità temporale dell'algoritmo per un insieme contenente  $N$  punti rispetta la seguente equazione per ricorrenza:

$$\begin{cases} T(1) = O(1), \\ T(N) = O(N) + T(N/2). \end{cases}$$

La complessità temporale dell'algoritmo per la creazione di un Kd-tree da un insieme di  $N$  elementi è quindi  $O(N \log(N))$  (per la dimostrazione di questo fatto si veda [5]).

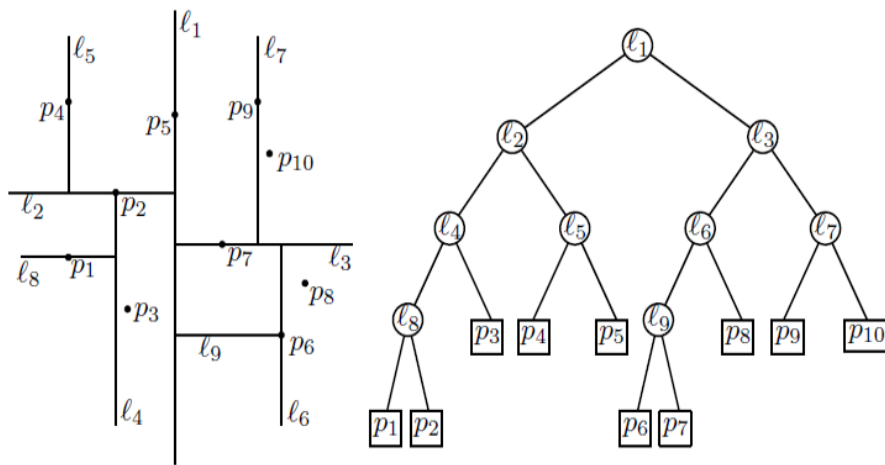


Figura A.4: 2d-tree costruito con l'algoritmo 4.

### A.1.3 Ricerca spaziale

Illustriamo il modo in cui, una volta formato il Kd-tree a partire da un insieme di punti  $X$ , sia possibile effettuare ricerche spaziali (*range queries*):

$$\text{“Trovare tutti i punti di } X \text{ in } \mathcal{R} = [a_1, b_1] \times \dots \times [a_K, b_K].\text{”} \quad (\text{A.1})$$

Per questo introduciamo il concetto di *regione* associata a un nodo  $\nu$ : essa è costituita dalla parte di spazio delimitata da tutti gli iperpiani associati ai nodi del cammino dalla radice al nodo stesso (cf. figura A.5). Osserviamo che è possibile conoscere la regione associata a un nodo  $\nu$  memorizzandola nel nodo al momento della creazione dell'albero, oppure calcolarla scorrendo l'albero dalla radice a  $\nu$ .

Un esempio di ricerca spaziale è riportato in figura A.6 con regione  $\mathcal{R}$  in grigio chiaro (a sinistra) e 3 tipi di nodi evidenziati nell'albero con colori diversi:

1. **grigio scuro**: nodi  $\nu$  tali che  $regione(\nu) \subseteq \mathcal{R}$ ;

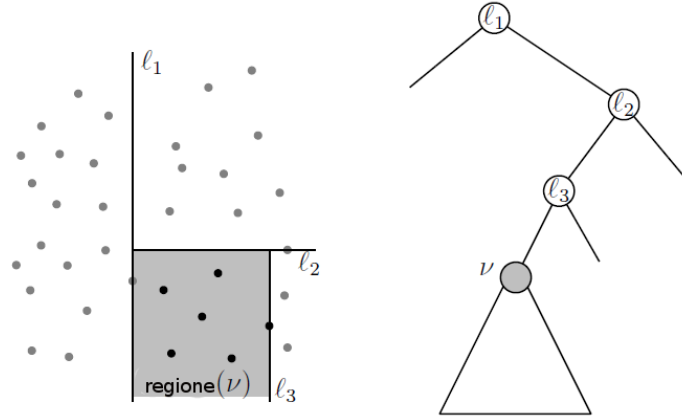


Figura A.5: Regione associata a un nodo.

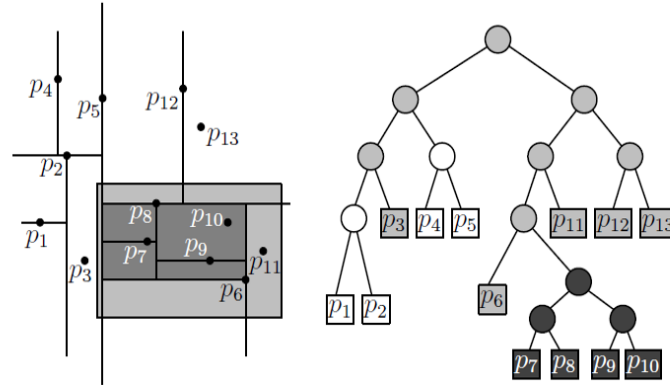


Figura A.6: Ricerca dei punti nella regione  $\mathcal{R}$  (a sinistra, in grigio chiaro).

2. **grigio chiaro**: nodi  $\nu$  tali che  $regione(\nu) \cap \mathcal{R} \neq \emptyset$  ma  $regione(\nu) \not\subseteq \mathcal{R}$ ;
3. **bianco**: nodi  $\nu$  tali che  $regione(\nu) \cap \mathcal{R} = \emptyset$ .

L'algoritmo 5 permette di processare la ricerca (A.1) ( $\nu_{left}$  e  $\nu_{right}$  indicano, rispettivamente, il figlio sinistro e destro del nodo  $\nu$ ).

Per quanto riguarda la complessità temporale dell'algoritmo per un albero costruito su un insieme di  $N$  punti, vale il seguente risultato (che non dimostriamo):

**Teorema 13.** *Un insieme  $X$  di  $N$  punti appartenenti a uno spazio  $K$ -dimensionale può essere organizzato in una struttura ad albero in un tempo  $T = O(N \log(N))$  in modo che ogni ricerca spaziale (A.1) su un'area  $\mathcal{R}$   $K$ -dimensionale può essere effettuata in un tempo  $O(N^{1/K} + m)$ , dove  $m$  è il numero di punti di  $X$  che giacciono in  $\mathcal{R}$ .*

---

**Algorithm 5** RANGEQUERY( $\nu, \mathcal{R}$ )

---

**Inputs:** Radice di un (sotto)albero  $\nu$ , area  $\mathcal{R}$ .

**Outputs:** Tutti i punti memorizzati

1. **Start**
  2. **if** ( $\nu$  è una foglia) **then**
  3.     **if** (il punto memorizzato in  $\nu$  appartiene a  $\mathcal{R}$ ) **then**
  4.         **return** il punto memorizzato in  $\nu$ ;
  5.     **end if**
  6. **else**
  7.     **if** ( $regione(\nu_{left}) \subseteq \mathcal{R}$ ) **then**
  8.         RITORNAPUNTIDELLALBERO( $\nu_{left}$ );
  9.     **else if** ( $regione(\nu_{left}) \cap \mathcal{R} \neq \emptyset$ ) **then**
  10.         **return** RANGEQUERY( $\nu_{left}, \mathcal{R}$ );
  11.     **end if**
  12. **else**
  13.     **if** ( $regione(\nu_{right}) \subseteq \mathcal{R}$ ) **then**
  14.         RITORNAPUNTIDELLALBERO( $\nu_{right}$ );
  15.     **else if** ( $regione(\nu_{right}) \cap \mathcal{R} \neq \emptyset$ ) **then**
  16.         **return** RANGEQUERY( $\nu_{right}, \mathcal{R}$ );
  17.     **end if**
  18. **end if**
  19. **End**
-



# B

## Appendice B: Interpolazione basata su partizione dell'unità

Come abbiamo accennato nel Capitolo 1, è prassi comune ricondurre il problema della ricostruzione di superfici a un problema di interpolazione multivariata; l'approccio RBF tuttavia è solo una delle soluzioni possibili per risolvere tale problema. Diamo qui una breve introduzione a una classe di metodi basati sulla partizione dell'unità, in quanto essi presentano alcune caratteristiche in comune con gli algoritmi presentati nei capitoli 3 e 4. Metodi basati sulla partizione dell'unità sono impiegati da lungo tempo nel campo della risoluzione numerica di equazioni alle derivate parziali; l'applicazione dello stesso approccio alla ricostruzione di superfici mediante interpolazione multivariata è, invece, di recente introduzione.

In particolare, descriviamo di seguito la rappresentazione di superfici basata su partizione dell'unità presentata in [23] e presentiamo brevemente un approccio misto partizione dell'unità-RBF proposto in [30], che mostra come le due tecniche possano essere vantaggiosamente combinate.

### B.1 Rappresentazione basata su partizione dell'unità

Sia  $\Omega$  un dominio limitato in uno spazio euclideo (per comodità supporremo nel seguito  $\Omega \subset \mathbb{R}^3$ ). Partiamo dalla seguente

**Definizione 9.** *Un insieme  $\{\phi_i\}_{1 \leq i \leq n}$  di funzioni  $\phi_i : \mathbb{R}^3 \rightarrow \mathbb{R}$  continue, non negative e a supporto compatto forma una **partizione dell'unità** su  $\Omega$  se:*

- $\Omega \subseteq \bigcup_i \text{supp}(\phi_i)$ ;

- $\sum_i \phi_i(\mathbf{x}) = 1$  per ogni  $\mathbf{x} \in \Omega$ .

Osserviamo che data una famiglia di funzioni  $\{w_i\}_{1 \leq i \leq n}$  di funzioni  $w_i : \mathbb{R}^3 \rightarrow \mathbb{R}$  continue, non negative a supporto compatto e tali che  $\Omega \subseteq \bigcup_i \text{supp}(w_i)$ , si può facilmente ottenere una partizione dell'unità ponendo:

$$\phi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j=1}^n w_j(\mathbf{x})}. \quad (\text{B.1})$$

Spostiamoci ora nell'ambito della ricostruzione di funzioni multivariate e consideriamo il problema (P) enunciato all'inizio del Capitolo 2: vogliamo ricostruire una funzione incognita  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  a partire dai valori campionati su una nuvola di punti  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^3$ ,  $f_j = f(\mathbf{x}_j)$  per ogni  $j = 1, \dots, N$ . Abbiamo già visto come il problema così formulato si applichi alla ricostruzione di una superficie definita implicitamente dall'insieme di livello zero di  $f$ . Nel seguito indicheremo con  $F$  una funzione approssimante di  $f$ , supporremo che i punti di  $X$  giacciono sulla superficie e che siano noti i vettori normali alla superficie in ogni punto della nuvola:  $N = \{\mathbf{n}_1, \dots, \mathbf{n}_N\}$ . L'approccio basato su una partizione dell'unità prevede di trovare una famiglia di approssimazioni locali  $\{Q_i\}_{1 \leq i \leq n}$  per la funzione incognita e una famiglia di pesi  $\{w_i\}_{1 \leq i \leq n}$  per formare una partizione dell'unità come in (B.1). La funzione approssimante  $F$  può allora essere assemblata raccordando le approssimazioni locali in modo continuo grazie alla partizione dell'unità:

$$F(\mathbf{x}) = \sum_{i=1}^n Q_i(\mathbf{x})\phi_i(\mathbf{x}) \quad \text{per ogni } \mathbf{x} \in \Omega. \quad (\text{B.2})$$

## B.2 Partizione dell'unità multi-scala

In [23] si propone una rappresentazione di superfici basata sulla partizione dell'unità in cui è introdotto un'approccio multiscala per scegliere le approssimanti locali. Questa rappresentazione si basa su tre aspetti fondamentali:

- una divisione ricorsiva dello spazio per localizzare le approssimanti locali in modo che siano più ricche di dettagli laddove richiesto;
- funzioni approssimanti locali sotto forma di curve quadratiche;
- funzioni peso per formare una partizione dell'unità e saldare insieme le approssimanti locali.

Data una nuvola di punti campionati sulla superficie  $X$  e un parallelepipedo  $\Omega \subset \mathbb{R}^3$  tale che  $X \subset \Omega$ , quest'ultimo viene ricorsivamente suddiviso in celle di piccole dimensioni; in ogni cella verrà calcolata una funzione approssimante locale: celle di dimensioni minori garantiscono un maggiore livello

di dettaglio ma comportano un aumento del numero di approssimanti. Per questo motivo è importante adattare la dimensione delle celle al grado di dettaglio richiesto in ogni regione. Nel procedimento proposto in [23] il parallelepipedo  $\Omega$  viene diviso in otto celle identiche dividendo a metà tutti gli spigoli e spiccando dai punti medi dei piani paralleli alle facce di  $\Omega$ . Questa divisione può essere ripetuta ricorsivamente su ogni cella per ottenere celle via via più piccole (come in Figura 3.2). Ogni cella non viene ulteriormente suddivisa se contiene meno di 8 punti di  $X$ . E' possibile implementare un controllo per verificare che alla fine della partizione ogni cella contenga un numero minimo di punti.

Una volta effettuata la partizione, in ogni cella viene calcolata un'approssimante locale della superficie a partire dai punti di  $X$  presenti nella cella. Una volta fissata una forma per l'approssimante locale è possibile calcolarla attraverso approssimazioni ai minimi quadrati. Ad esempio, una delle procedure per il calcolo dell'approssimante locale proposte in [23] è molto simile a quella per il calcolo di  $g_i(\mathbf{x})$  presentata nella Sezione 3.2.

1. In una generica cella  $\mathcal{G}_i$ , si calcola il centro di gravità  $\mathbf{x}_{\mathcal{G}_i}$  di tutti i punti  $\mathbf{x}_j \in X \cap \mathcal{G}_i$ , e un vettore normale associato  $\mathbf{n}_{\mathcal{G}_i}$  dato dalla media pesata di tutti i vettori normali;
2. si introduce un sistema di riferimento locale  $\mathcal{R}_{\mathcal{G}_i}$  con assi  $(u, v, z)$ ;
3. si calcola un'approssimante locale della forma  $g_i(\mathbf{x}) = z_{\mathbf{x}} - (Au_{\mathbf{x}}^2 + 2Bu_{\mathbf{x}}v_{\mathbf{x}} + Cv_{\mathbf{x}}^2)$ ; i coefficienti  $A, B, C$  sono determinati grazie a una minimizzazione ai minimi quadrati.

Per quanto riguarda le funzioni peso  $\{w_i\}_{1 \leq i \leq n}$ , in [23] sono proposte due alternative: la prima, da usarsi per l'approssimazione di superfici, impiega la B-spline quadratica  $b(t)$  per generare le funzioni-peso

$$w_i(\mathbf{x}) = \left( \frac{3\|\mathbf{x} - \mathbf{c}_i\|}{2R_i} \right);$$

la seconda, da impiegarsi per effettuare l'interpolazione, è data da

$$w_i(\mathbf{x}) = \left( \frac{(R_i - \|\mathbf{x} - \mathbf{c}_i\|)_+}{R_i - \|\mathbf{x} - \mathbf{c}_i\|} \right)^2,$$

dove

$$(a)_+ = \begin{cases} a & \text{se } a > 0 \\ 0 & \text{altrimenti;} \end{cases}$$

Osserviamo che in entrambi i casi il coefficiente scalare  $R_i$  determina il supporto di ogni funzione-peso, che coincide con il  $R_i$ -intorno di  $\mathbf{c}_i$ . Ai fini della ricostruzione è fondamentale che i supporti delle funzioni peso si intersechino; per questo motivo  $R_i$  deve essere scelto in modo che il  $R_i$ -intorno di  $\mathbf{c}_i$  ricopra interamente la cella  $\mathcal{G}_i$  e una parte delle celle vicine.

Uno dei vantaggi dell'approccio basato su partizione dell'unità consiste nel fatto che le approssimanti locali possono avere una forma arbitraria: in particolare approssimanti corrispondenti a celle diverse possono avere forme diverse. Così ad esempio in [23] si propone di adottare una forma di approssimante specifica in quelle celle in cui venga rilevata la presenza di una *sharp feature* come un angolo, uno spigolo, etc. . Questo rende l'approccio basato sulla partizione dell'unità versatile per la modellizzazione di oggetti complessi con parti dalle forme più varie.

### B.3 Approccio misto: partizione dell'unità e RBF

Gli approcci per la ricostruzione basati su partizione dell'unità e su RBF possono anche essere combinati per creare metodi misti. Abbiamo visto nel Paragrafo 3.2.2 come gli algoritmi presentati in questo lavoro, basati sull'approccio RBF, introducano un elemento tipico dei metodi basati su partizione dell'unità, l'uso di funzioni approssimanti locali. Un'altra possibilità è descritta in [30], in cui si effettua una partizione ricorsiva dello spazio in celle sempre più piccole come descritto nella sezione precedente (anche se in questo caso le celle possono essere sia parallelepipedi sia ellissi). Il processo di divisione ricorsiva di una cella si ferma quando essa contiene meno di  $T_{max}$  punti; se una cella contiene meno di  $T_{min}$  punti essa viene ingrandita fino a contenerne almeno  $T_{min}$ .  $T_{max}$  e  $T_{min}$  sono parametri scelti dall'utente (gli autori di [30] suggeriscono  $T_{max} \in [120, 200]$  e  $T_{min} > 30$ ).

Una volta effettuata la partizione della *bounding box* in celle, in ogni cella l'approssimante locale viene costruita con la tecnica dell'interpolazione RBF classica descritta nel Capitolo 2. Un approccio analogo (suddivisione ricorsiva dello spazio per partizione dell'unità e approssimanti locali calcolate con approccio RBF) è proposto in [29], dove però è descritta una procedura di divisione spaziale differente che permette anche la ricostruzione dei dati a scale differenti.

Tali metodi basati su un approccio misto, sebbene in generale si rivelino più lenti di quelli basati su RBF, permettono di ottenere funzioni approssimanti versatili (in particolare anche angoli, spigoli e punte possono essere fedelmente riprodotti) la cui valutazione può essere resa più rapida dal *principio di località* di esse cui godono; infatti, la ricostruzione di ogni regione spaziale dipende solo dalle regioni più vicine e può essere effettuata localmente. Questo, oltre a offrire interessanti possibilità di parallelizzazione, può essere vantaggioso in quei contesti in cui solo la ricostruzione di una parte di un oggetto sia necessaria (ad esempio, nella riproduzione virtuale di una scena in cui solo alcuni particolari di un oggetto sono visibili).

# Bibliografia

- [1] Jules Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4), 1988.
- [2] Jules Bloomenthal. An implicit surface polygonizer. *Graphics Gems*, 4:324–349, 1994.
- [3] RP Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [4] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, e T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. *Computer Graphics (Proceedings of ACM SIGGRAPH 2001)*, pp. 67–76, August 2001.
- [5] T.H. Cormen, C.E. Leiserson, R.L. Rivest, e C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.
- [6] J.P. Gois, V. Polizelli-Junior, T. Etienne, E. Tejada, A. Castelo, T. Ertl, e L.G. Nonato. Robust and adaptive surface reconstruction using partition of unity implicits. In *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on*, pp. 95–104. IEEE, 2007.
- [7] C. Gotsman e D. Keren. Tight fitting of convex polyhedral shapes. *International Journal of Shape Modeling*, 4(3/4):111–126, 1998.
- [8] Nail A. Gumerov e Ramani Duraiswami. Fast radial basis function interpolation via preconditioned krylov iteration. *SIAM J. Sci. Comput.*, 29:1876–1899, 2007.
- [9] T. Hantschel e A.I. Kauerauf. *Fundamentals of basin and petroleum systems modeling*. Springer Verlag, Berlin, 2009.
- [10] J.C. Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996.

- [11] Erich Hartmann. Numerical implicitization for intersection and gn-continuous blending of surfaces. *Computer Aided Geometric Design*, 15(4):377–397, 1998.
- [12] Erich Hartmann. Numerical parameterization of curves and surfaces. *Computer Aided Geometric Design*, 17:251–266, 2000.
- [13] Simon Haykin. *Neural Networks - A comprehensive foundation*. Prentice Hall International, Inc., second edizione, 1999.
- [14] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, e W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics-New York-Association for Computing Machinery-*, 26:71–71, 1992.
- [15] Armin Iske. *Multiresolution Methods in Scattered Data Modelling*. Springer-Verlag, Berlin, 2004.
- [16] T. Ju, F. Losasso, S. Schaefer, e J. Warren. Dual contouring of hermite Data. *ACM Transactions on Graphics (TOG)*, 21(3):339–346, 2002.
- [17] L.P. Kobbelt, M. Botsch, U. Schwanercke, e H.P. Seidel. Feature sensitive surface extraction from volume data. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 57–66. ACM, 2001.
- [18] Matteo Longoni. Generation of high quality meshes on open surfaces for biomedical CFD. Tesi per Master, Politecnico di Milano - Imperial College London, 2007.
- [19] William E. Lorensen e Harvey E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *SIGGRAPH 87 Proceedings*, pp. 163–170, 1987.
- [20] M. S. Floater e A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *J. Comp. Appl. Math*, 73:64–78, 1996.
- [21] B.S. Morse, T.S. Yoo, P. Rheingans, D.T. Chen, e K.R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pp. 89–98. IEEE, 2001.
- [22] Shigeru Muraki. Volumetric shape description of range data using -blobby model. *Computer Graphics (Proceedings of ACM SIGGRAPH 1991)*, pp. 227–235, 1991.

- [23] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, e H.P. Seidel. Multi-level partition of unity implicits. In *ACM Transactions on Graphics (TOG)*, volume 22, pp. 463–470. ACM, 2003.
- [24] Y. Ohtake, A. Belyaev, e H.P. Seidel. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. 2003.
- [25] Y. Ohtake, A. Belyaev, e H.P. Seidel. 3d scattered data approximation with adaptive compactly supported radial basis functions. 2004.
- [26] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley Reading MA, 1990.
- [27] Robert Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.
- [28] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pp. 658–665. IEEE, 1993.
- [29] I. Tobor, P. Reuter, e C. Schlick. Multi-scale reconstruction of implicit surfaces with attributes from large unorganized point sets. In *Shape Modeling Applications, 2004. Proceedings*, pp. 19–30. IEEE, 2004.
- [30] I. Tobor, P. Reuter, C. Schlick, e altri. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. *Journal of WSCG*, 12(3):467–474, 2004.
- [31] G. Turk e J.F. O'brien. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics (TOG)*, 21(4):855–873, 2002.
- [32] Greg Turk e James F. O'brien. Shape transformation using variational implicit surfaces. *Computer Graphics (Proceedings of ACM SIG-GRAPH 1999)*, pp. 335–342, August 1999.
- [33] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [34] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, 2005.
- [35] Hongkai Zhao e Stanley Osher. Visualization, analysis and shape reconstruction of unorganized data sets. In *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, 2002.