

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Facoltà di Ingegneria

Corso di Laurea Specialistica in Ingegneria Informatica

Classe n. 35/S – Ingegneria Informatica

**SUPERFICI DI SUDDIVISIONE PER LA  
PROGETTAZIONE ASSISTITA AL CALCOLATORE:  
ASPETTI ANALITICI, IMPLEMENTAZIONE E STUDIO  
DI FATTIBILITÀ**

Relatore:

Dott. Christian VERGARA

Correlatore:

Dott.ssa Milvia ROSSINI

Tesi di Laurea Specialistica

Calogero CALÌ

Matricola n. 44976

ANNO ACCADEMICO 2009 / 2010



# Ringraziamenti

Desidero ringraziare tutte le persone che mi sono state vicine durante questo lavoro di tesi.

Ringrazio il dottor **Christian Vergara**, mio relatore, per i suoi consigli, i suoi insegnamenti e la sua disponibilità. Un grazie particolare per la stima dimostrata nei miei confronti.

Ringrazio la dottoressa **Milvia Rossini**, mio correlatore, e la dottoressa **Lucia Romani** per la loro disponibilità e i loro preziosi consigli tecnici.

Ringrazio il **team DDX** per avermi offerto gli spazi, le infrastrutture e l'ambiente di lavoro adatto per svolgere al meglio le mie mansioni. Un grazie particolare va all'ingegner **Dario Sassi**, responsabile dell'area sviluppo, per i suoi consigli e suoi stimoli.

Ringrazio la mia fidanzata **Federica** per avermi sostenuto nei momenti di difficoltà con il calore delle sue parole e la dolcezza del suo sorriso.

Ringrazio mio papà **Giuseppe** e mia mamma **Concetta**, nonché mio fratello **Enzo** e mia sorella **Valentina** per il loro profondo affetto e per aver sempre creduto nelle mie potenzialità.

Ringrazio infine tutte le persone che mi hanno dato una mano o un incoraggiamento e che ho dimenticato di citare.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Nozioni di base sulla suddivisione</b>	<b>3</b>
2.1	Curve B-spline . . . . .	3
2.1.1	Funzioni polinomiali definite a tratti . . . . .	4
2.1.2	Definizione delle B-spline uniformi . . . . .	4
2.1.3	Raffinabilità delle B-spline . . . . .	8
2.1.4	Raffinamento per curve spline . . . . .	9
2.1.5	Algoritmo di Chaikin per la suddivisione di curve . . . . .	12
2.2	Suddivisione come raffinamento ripetuto . . . . .	13
2.2.1	Convoluzione discreta . . . . .	13
2.2.2	Convergenza di uno schema di suddivisione . . . . .	16
2.3	Analisi della matrice di suddivisione . . . . .	19
2.3.1	Intorno invariante . . . . .	19
2.3.2	Analisi degli autovalori . . . . .	22
2.3.3	Invarianza rispetto alle trasformazioni affini . . . . .	24
2.3.4	Comportamento geometrico di suddivisioni ripetute . . . . .	24
<b>3</b>	<b>Superfici di Suddivisione uniformi e non uniformi</b>	<b>27</b>
3.1	Concetti chiave e una breve panoramica . . . . .	28
3.1.1	Classificazione degli schemi di suddivisione . . . . .	28
3.1.2	Terminologia . . . . .	30
3.2	Superfici di suddivisione uniformi . . . . .	32
3.2.1	Schema di Doo-Sabin . . . . .	32
3.2.2	Schema di Catmull-Clark . . . . .	36
3.3	Superfici di suddivisione non uniformi . . . . .	42
3.3.1	Vettore dei nodi uniformi e non uniformi . . . . .	42
3.3.2	Raffinamento di B-spline non uniformi . . . . .	43
3.3.3	Raffinamento delle NURSS . . . . .	48

3.3.4	Discussione . . . . .	51
<b>4</b>	<b>Implementazione</b>	<b>55</b>
4.1	Sviluppo dei prototipi . . . . .	55
4.1.1	Strutture dati . . . . .	56
4.1.2	Soluzioni adottate nei casi limite . . . . .	57
4.2	Implementazione per sistemi CAD . . . . .	58
4.2.1	I sistemi CAD . . . . .	58
4.2.2	Scelte di progetto . . . . .	60
4.2.3	Le classi . . . . .	62
<b>5</b>	<b>Risultati numerici</b>	<b>67</b>
5.1	Confronto numerico fra i vari schemi . . . . .	67
5.1.1	Confronto tra schemi primali e duali . . . . .	68
5.1.2	Confronto tra schemi uniformi e non uniformi . . . . .	71
5.2	Superfici di suddivisione per la progettazione industriale . . . . .	74
<b>6</b>	<b>Conclusioni</b>	<b>81</b>

# Elenco delle figure

2.1	Definizione di B-spline $B_1(t)$ (a destra) come convoluzione di $B_0(t)$ con se stessa (a sinistra). . . . .	6
2.2	Basi di B-spline uniformi dal primo al sesto grado. . . . .	6
2.3	Passi di suddivisione per una spline cubica. . . . .	12
2.4	Suddivisione di una curva mediante l'algoritmo di Chaikin. . . . .	13
2.5	Regole topologiche della suddivisione di Chaikin. . . . .	14
2.6	Maschere della suddivisione di Chaikin. . . . .	14
2.7	Dimensione del vicinato invariante per la suddivisione di B-spline cubiche. . . . .	21
2.8	Invarianza alla traslazione. . . . .	25
3.1	Differenti regole di raffinamento per facce quadrilatere. . . . .	30
3.2	Superfici di suddivisione con sharp features. . . . .	32
3.3	Regole topologiche per la suddivisione di Doo-Sabin. . . . .	33
3.4	Maschere per la suddivisione di Doo-Sabin. . . . .	34
3.5	Pezza standard di una B-spline bi-quadratica. . . . .	34
3.6	Illustrazione di una superficie di suddivisione di Doo-Sabin, tratta da [13]. . . . .	37
3.7	Illustrazione di un modello di tubatura ottenuto con superfici Doo-Sabin, presa da [13]. . . . .	38
3.8	Pezza standard di una B-spline bi-cubica. . . . .	38
3.9	Maschere per la suddivisione Catmull-Clark di mesh quadrilatere. . . . .	41
3.10	Maschere per la suddivisione Catmull-Clark di mesh arbitrarie. . . . .	42
3.11	Maschere di sharp features per la suddivisione di Catmull-Clark. . . . .	43
3.12	Regole topologiche per la suddivisione di Catmull-Clark. . . . .	43
3.13	Illustrazione di una superficie di suddivisione di Catmull-Clark, tratta da [13]. . . . .	44
3.14	Illustrazione del modello di una pistola ottenuta con superfici Catmull-Clark, presa da [13]. . . . .	45

3.15	Funzioni base di una B-spline cubica con due differenti vettori dei nodi.	45
3.16	B-spline cubiche con gli stessi punti di controllo, ma differenti vettori dei nodi. . . . .	46
3.17	Suddivisione di una curva B-spline quadratica non uniforme. . . . .	46
3.18	Suddivisione di una curva B-spline cubica non uniforme. . . . .	47
3.19	Suddivisione di una superficie B-spline quadratica non uniforme. . . . .	47
3.20	Suddivisione di una superficie B-spline cubica non uniforme. . . . .	48
3.21	Regole di raffinamento per NURSS quadratiche. . . . .	49
3.22	Punto di faccia di NURSS cubiche. . . . .	50
3.23	Punto di lato di NURSS cubiche. . . . .	51
3.24	Punto di vertice di NURSS cubiche. . . . .	51
3.25	Spaziatura nodale di NURSS cubiche. . . . .	52
3.26	Effetto di una spaziatura non uniforme nelle NURSS. . . . .	52
3.27	Illustrazione di diverse forme ricavate con superfici Doo-Sabin non uniformi, tratta da [27]. . . . .	53
3.28	Superfici Doo-Sabin non uniforme con diverse spaziature nodali, tratta da [27]. . . . .	53
3.29	Superfici Catmull-Clark non uniforme con diverse spaziature nodali, presa da [27]. . . . .	54
4.1	Esempio di una mesh nel prototipo in Matlab. . . . .	56
4.2	Struttura dati half-edge. . . . .	61
4.3	Attraversamento del vicinato di un vertice mediante la struttura dati half-edge. . . . .	62
4.4	Diagramma delle classi. . . . .	65
5.1	Illustrazione di un cubo ottenuta con i prototipi in Matlab. . . . .	68
5.2	Illustrazione di un anello ottenuta con i prototipi in Matlab. . . . .	70
5.3	Illustrazione di un vaso ottenuta con le classi C++. . . . .	71
5.4	Illustrazione di una croce ottenuta con i prototipi in Matlab. . . . .	76
5.5	Illustrazione di un pezzo meccanico ottenuta con le classi C++. . . . .	77
5.6	Superfici Doo-Sabin non uniformi con differenti spaziature nodali ottenute mediante i prototipi in Matlab. . . . .	78
5.7	Superfici Catmull-Clark non uniformi con differenti spaziature nodali ottenute mediante le classi in C++. . . . .	79

# Elenco delle tabelle

3.1	Classificazione degli schemi di suddivisione. . . . .	30
5.1	Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura	
5.1.	. . . . .	69
5.2	Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura	
5.2.	. . . . .	69
5.3	Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura	
5.3.	. . . . .	72
5.4	Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura	
5.4.	. . . . .	73
5.5	Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura	
5.5.	. . . . .	74



# Capitolo 1

## Introduzione

La progettazione assistita al calcolatore riguarda l'utilizzo delle tecnologie informatiche durante il processo di design di prodotti industriali. I software CAD (Computer-Aided Design) sono strumenti estremamente importanti in diverse applicazioni industriali: dall'automotive alla costruzione di navi, dall'architettura all'elettronica, dalla scultura alla falegnameria. Gli ambienti CAD sono utilizzati per la modellazione geometrica dei beni da produrre poiché permettono di progettare curve e superfici nello spazio 3-D.

Due sono le tecnologie principali disponibili oggi per definire superfici di forma arbitraria: le *Non-Uniform Rational B-Splines* (NURBS) e le *superfici di suddivisione*. Entrambe le rappresentazioni sono basate sulle B-spline, ma le estendono in modi diversi per offrire ai designer differenti libertà di progettazione. Le NURBS, offrendo un controllo puntuale della parametrizzazione e della regolarità della superficie, sono lo standard dominante nella progettazione assistita al calcolatore. Le superfici di suddivisione, invece, sono diffuse nell'animazione e nell'intrattenimento perché permettono un'estrema libertà dai vincoli topologici. Infatti, a differenza delle NURBS, non necessitano di "cucire" (trimmare) tra loro differenti pezzi della superficie, riducendo notevolmente le tempistiche di progettazione. E' quindi naturale chiedersi quando le applicazioni CAD potranno beneficiare appieno dell'utilizzo delle superfici di suddivisione, oggi poco diffuse in tale ambito, dato che negli ultimi anni i prodotti sono sempre più smussati e di forma libera.

In letteratura si può trovare una gran varietà di schemi di suddivisione per lo più utilizzati nelle applicazioni grafiche e nell'animazione [4, 7, 8, 11, 18, 20, 27]. Negli ultimi anni, al fine di poter utilizzare con profitto le superfici di suddivisione anche nell'ambito della progettazione industriale, sono state effettuate continue ricerche in tal senso [3, 16, 17, 24, 25]. In questa direzione si colloca anche questo lavoro. Dapprima sono stati analizzati gli schemi di suddivisione Doo-Sabin e Catmull-

Clark uniformi e non uniformi [4, 7, 27], identificandone le maggiori potenzialità e limiti nell'ambito della progettazione assistita al calcolatore. Si è passati quindi all'implementazione degli schemi di suddivisione focalizzandosi sulla scrittura di algoritmi robusti, efficienti e facilmente integrabili in una comune applicazione CAD. Infine sono stati confrontati i risultati numerici ottenuti per il caso uniforme e non uniforme, e sono stati identificati i principali sviluppi necessari per poter utilizzare diffusamente le superfici di suddivisione in un contesto industriale.

Il sommario della presente tesi è come segue. Il Capitolo 2 introduce le nozioni di base necessarie per comprendere gli schemi di suddivisione; particolare enfasi viene data alle proprietà di raffinamento delle B-spline. Il Capitolo 3 fornisce una breve panoramica degli schemi di suddivisione presenti in letteratura, per poi descrivere in dettaglio gli schemi Doo-Sabin e Catmull-Clark uniformi e non uniformi. Gli aspetti implementativi sono presentati nel Capitolo 4, mentre il Capitolo 5 fornisce un'analisi sperimentale dei risultati numerici ottenuti. Infine, il Capitolo 6 offre delle conclusioni e dei possibili sviluppi futuri.

# Capitolo 2

## Nozioni di base sulla suddivisione

L'obiettivo di questo capitolo è introdurre i concetti base necessari per comprendere i diversi schemi di suddivisione che verranno studiati successivamente. Sono analizzate le curve di suddivisione, e non le superfici, per mantenere semplice la trattazione. D'altra parte, tutte le osservazioni fatte sono direttamente estendibili anche al caso di superfici. In più, sono proposti diversi esempi per motivare le idee e i concetti presentati.

Nella Sezione 2.1 sono descritte brevemente le basi B-spline e le relative proprietà di raffinamento poiché nel Capitolo 3, dedicato agli schemi di suddivisione, viene posta particolare enfasi sugli schemi che generalizzano le superfici B-spline. La Sezione 2.1 introduce anche la notazione basata sulle *matrici di suddivisione*, di fondamentale importanza per lo studio che ci si è preposti, e presenta il primo schema di suddivisione noto come algoritmo di Chaikin [5]. La convergenza di un processo di suddivisione viene, invece, discussa nella Sezione 2.2. In più, Per studiare le proprietà di una curva (o di una superficie), come ad esempio la differenziabilità, è necessario comprendere quali punti di controllo influenzano l'intorno del punto di interesse. Questa nozione è racchiusa nel concetto di *intorno invariante* descritto nella Sezione 2.3 che presenta, inoltre, l'analisi della struttura degli autovalori della matrice di suddivisione locale, fondamentale per studiare le proprietà della superficie limite.

### 2.1 Curve B-spline

Nella progettazione assistita dal calcolatore, le spline sono uno strumento molto popolare in quanto facili da memorizzare, manipolare e calcolare. In particolare la base delle B-spline e le curve B-spline rivestono un ruolo fondamentale in questo ambito.

Il disegnatore produce un insieme di punti  $\{\mathbf{p}_i\}_{i=0}^m$  con una regola di connessione che rappresentano una prima approssimazione della forma da disegnare. Mediante questi punti, detti *punti di controllo*, si vuole scrivere una curva parametrica  $(x(t), y(t))$  come combinazione lineare di tali vertici  $\mathbf{p}_i = [x_i \ y_i]$  e opportune basi  $b_i(t)$

$$\begin{aligned}x(t) &= \sum_{i=0}^m x_i b_i(t) & t \in [a, b] \\y(t) &= \sum_{i=0}^m y_i b_i(t)\end{aligned}$$

In particolare si usa la base delle B-spline associate ad una partizione dell'intervallo  $[a, b]$ .

### 2.1.1 Funzioni polinomiali definite a tratti

Le spline sono curve polinomiali continue di grado arbitrario definite a tratti su ogni intervallo in cui è stato suddiviso il dominio di interesse (per una trattazione esaustiva si rimanda a [9]).

In particolare, si consideri l'intervallo  $I = [a, b]$  e un insieme di punti  $\{t_i\}_{i=0}^m$  detti *nodi*. Per semplificare la trattazione ci si limita al caso di nodi distinti cioè  $a = t_0 < t_1 < \dots < t_m = b$ . Si chiama spline di ordine  $n$ , con nodi  $\{t_i\}_{i=0}^m$ , una funzione  $S(t)$  con le seguenti proprietà:

- $S : [a, b] \rightarrow \mathbb{R}$ ,
- $S(t) = s_i(t) \quad t \in [t_i, t_{i+1})$ , dove  $s_i(t)$  è un polinomio di grado al più  $n - 1$ ,
- $D^j s_{i-1}(t) = D^j s_i(t) \quad j = 0, 1, \dots, k - 2$ .

Una funzione spline così definita è di classe  $C^{k-2}$  su  $[a, b]$ .

Il vettore  $\mathbf{t} = [t_0, \dots, t_m]$  è detto *vettore dei nodi* della spline. Se tali nodi sono equamente distribuiti in  $[a, b]$  la spline è *uniforme*, in caso contrario è *non-uniforme*.

### 2.1.2 Definizione delle B-spline uniformi

Ci sono molti modi per definire le B-spline; nel caso uniforme le B-spline possono essere definite come convoluzione ripetuta di funzioni. In tal modo è possibile verificare direttamente come una curva B-spline possa essere generata mediante suddivisione.

Si indica con

$$B_0(t) = \begin{cases} 1 & \text{se } 0 \leq t < 1 \\ 0 & \text{altrimenti,} \end{cases}$$

la B-spline costante su  $[0, 1)$  di grado<sup>1</sup> 0 e con  $B_{i,0}(t) = B_0(t - i)$  le traslazioni di  $B_0(t)$  che sono non nulle su  $[i, i + 1)$ .

Si mostra ora come una funzione base di una B-spline di grado  $n$  possa essere ottenuta convolvendo la funzione base di grado  $n - 1$  con la funzione box  $B_0(t)$ . In seguito sarà indicata la convoluzione di due funzioni  $f(t)$  e  $g(t)$  con

$$(f \otimes g)(t) = \int_{\mathbb{R}} f(s)g(t - s)ds.$$

Per esempio, la B-spline di grado 1 è definita come la convoluzione di  $B_0(t)$  con se stessa

$$B_1(t) = \int_{\mathbb{R}} B_0(s)B_0(t - s)ds.$$

Graficamente (si veda la Figura 2.1) questa convoluzione può essere valutata facendo “slittare” una funzione box lungo la direzione crescente delle ascisse e mantenendo fissa la seconda funzione box. Il valore della convoluzione, per una data posizione del box in movimento, è l’area sotto il prodotto dei box, che è semplicemente la lunghezza dell’intervallo dove entrambi i box non sono nulli. All’inizio i due box non hanno un supporto comune, ma una volta che il box ha raggiunto lo zero c’è una sovrapposizione crescente tra i supporti delle due funzioni. Il valore della convoluzione cresce con  $t$  fino a  $t = 1$ . Successivamente la sovrapposizione inizia a diminuire, e il valore della convoluzione con essa, fino a raggiungere lo zero in  $t = 2$ . La funzione  $B_1(t)$  è dunque la funzione lineare mostrata nella Figura 2.1.

Continuando, è possibile calcolare la base di una B-spline di ordine 2 convolvendo di nuovo  $B_1(t)$  con la funzione box  $B_0(t)$

$$B_2(t) = \int_{\mathbb{R}} B_1(s)B_0(t - s)ds.$$

In questo caso, la curva risultante consiste di tre segmenti quadratici definiti negli intervalli  $(0,1)$ ,  $(1,2)$  e  $(2,3)$ . In generale, la B-spline di grado  $l$  è

$$B_l(t) = \int_{\mathbb{R}} B_{l-1}(s)B_0(t - s)ds.$$

Definendo le B-spline come convoluzione, seguono immediatamente delle importanti proprietà, la prima delle quali riguarda la continuità delle spline

---

<sup>1</sup>Il *grado* di un polinomio è l’ordine massimo dei suoi esponenti, mentre l’*ordine* conta il numero di coefficienti e quindi è più grande di 1 rispetto al grado. Per esempio, una curva cubica ha grado 3 e ordine 4.

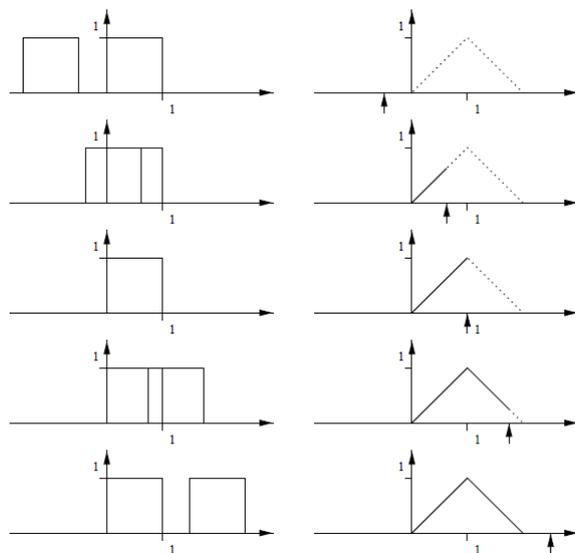


Figura 2.1: Definizione di B-spline  $B_1(t)$  (a destra) come convoluzione di  $B_0(t)$  con se stessa (a sinistra).

**Teorema 1.** *Se  $f(t)$  è  $C^k$ -continua, allora  $(B_0 \otimes f)(t)$  è  $C^{k+1}$ -continua.*

Questa è una conseguenza diretta della convoluzione con una funzione box; da cui segue che le basi di B-spline di grado  $n$  sono  $C^{n-1}$ -continue, essendo le basi di B-spline di grado 1  $C^0$ -continue.

Nella Figura 2.2 sono riportate le funzioni base delle B-spline dal grado 1 fino al grado 6. Come si può vedere, al crescere del grado, le funzioni base diventano più smussate e hanno un supporto maggiore.

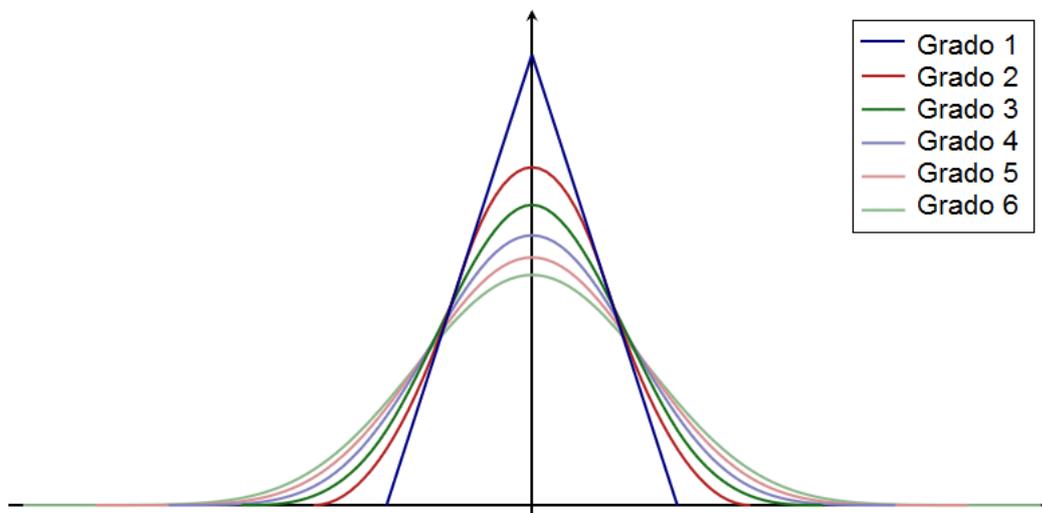


Figura 2.2: Basi di B-spline uniformi dal primo al sesto grado.

Una curva B-spline di grado  $l$ ,  $0 \leq l \leq m - 2$ , è una curva parametrica

$$\mathbf{S} : [t_0, t_{m-1}] \rightarrow \mathbb{R}^2$$

composta da una combinazione lineare di funzioni base B-spline  $B_{i,l}$  di grado  $l$

$$\mathbf{S}(t) = \sum_{i=0}^{m-l-2} \mathbf{p}_i B_{i,l}(t), \quad t \in [t_l, t_{m-l-1}].$$

Ci sono  $m - l - 1$  punti di controllo a cui corrispondono altrettante funzioni base.

Per ogni grado  $l$ , le funzioni base possono essere scritte in forma parametrica attraverso la formula ricorsiva di Cox-de Boor [9]

$$B_{j,0}(t) = \begin{cases} 1 & \text{se } t_j \leq t < t_{j+1}, \quad j = 0, \dots, m-2 \\ 0 & \text{altrimenti} \end{cases}$$

$$B_{j,l}(t) = \frac{t - t_j}{t_{j+l} - t_j} B_{j,l-1}(t) + \frac{t_{j+l-1} - t}{t_{j+l-1} - t_{j+1}} B_{j+1,l-1}(t), \quad j = 0, \dots, m-l-2.$$

Si noti che  $j + l - 1$  non può eccedere  $m - 1$ . Ciò limita sia  $j$  sia  $l$ .

Le B-spline più comuni sono le quadratiche e le cubiche. Le funzioni base di B-spline uniformi quadratiche possono essere facilmente precalcolate e, a meno di una traslazione, sono le stesse per ogni segmento:

$$B_{j,2}(t) = \begin{cases} \frac{1}{2}t^2 \\ -t^2 + t + \frac{1}{2} \\ \frac{1}{2}(1-t)^2 \end{cases}.$$

Riscrivendo la curva  $\mathbf{S}(t)$  in forma matriciale, si ottiene

$$\mathbf{S}_i(t) = \mathbf{t} M P_i \quad t \in [0, 1], \quad i = 1, 2, \dots, m-2 \quad (2.1)$$

$$\text{dove } \mathbf{t} = [t^2 \quad t \quad 1], \quad M = \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{e } P_i = \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \end{bmatrix}.$$

Le funzioni base di B-spline cubiche con vettore dei nodi uniforme sono, invece, le seguenti

$$B_{j,3}(t) = \begin{cases} \frac{1}{6}t^3 \\ \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ \frac{1}{6}(3t^3 - 6t^2 + 4) \\ \frac{1}{6}(1-t)^3 \end{cases}.$$

E' possibile, quindi, riscrivere la curva nella forma

$$\mathbf{S}_i(t) = \mathbf{t} M P_i \quad t \in [0, 1], i = 1, 2, \dots, m-2 \quad (2.2)$$

$$\text{in cui } \mathbf{t} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}, M = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \text{ e } P_i = \begin{bmatrix} \mathbf{p}_{i-1} \\ \mathbf{p}_i \\ \mathbf{p}_{i+1} \\ \mathbf{p}_{i+2} \end{bmatrix}.$$

### 2.1.3 Raffinabilità delle B-spline

Una notevole proprietà delle B-spline è quella di soddisfare un'equazione di raffinamento. Questa è l'osservazione chiave che lega le B-spline e le superfici di suddivisione. L'equazione di raffinamento di una B-spline di grado  $l$  è la seguente

$$B_l(t) = \frac{1}{2^l} \sum_{k=0}^{l+1} \binom{l+1}{k} B_l(2t-k). \quad (2.3)$$

In altre parole, una B-spline di grado  $l$  può essere scritta come combinazione lineare di copie traslate ( $k$ ) e dilatate ( $2t$ ) di se stessa.

Data l'importanza dell'equazione (2.3), si riporta la dimostrazione tratta da [34]. Si osservi, dapprima, che la funzione box, in altre parole una B-spline di grado 0, può essere scritta in termini di traslazione e dilatazione di se stessa

$$B_0(t) = B_0(2t) + B_0(2t-1). \quad (2.4)$$

Ciò può essere verificato immediatamente dalla definizione di  $B_0(t)$ . Si ricorda, inoltre, che è stata definita una B-spline di grado  $l$  come

$$B_l(t) = \bigotimes_{i=0}^l B_0(t) = \bigotimes_{i=0}^l (B_0(2t) + B_0(2t-1)). \quad (2.5)$$

Questa espressione può essere trasformata in una forma più conveniente sfruttando

alcune proprietà della convoluzione, che si riportano qui per comodità

$$\begin{aligned}
 f(t) \otimes (g(t) + h(t)) &= f(t) \otimes g(t) + f(t) \otimes h(t) && \text{linearità} \\
 f(t - i) \otimes g(t - k) &= m(t - i - k) && \text{spostamento nel tempo} \\
 f(2t) \otimes g(2t) &= \frac{1}{2}m(2t) && \text{scalatura nel tempo}
 \end{aligned}$$

dove  $f(t)$ ,  $g(t)$ , e  $h(t)$  sono funzioni date e  $m(t) = f(t) \otimes g(t)$ . Queste proprietà possono essere facilmente verificate sostituendo la definizione di convoluzione e applicando dei semplici cambi di variabile di integrazione. E' possibile, quindi, riscrivere  $B_1(t)$  nel seguente modo

$$\begin{aligned}
 B_1(t) &= B_0(t) \otimes B_0(t) \\
 &= (B_0(2t) + B_0(2t - 1)) \otimes (B_0(2t) + B_0(2t - 1)) \\
 &= B_0(2t) \oplus B_0(2t) + B_0(2t) \oplus B_0(2t - 1) + B_0(2t - 1) \oplus B_0(2t) + B_0(2t - 1) \oplus B_0(2t - 1) \\
 &= \frac{1}{2}B_1(2t) + \frac{1}{2}B_1(2t - 1) + \frac{1}{2}B_1(2t - 1) + \frac{1}{2}B_1(2t - 1 - 1) \\
 &= \frac{1}{2}(B_1(2t) + 2B_1(2t - 1) + B_1(2t - 2)) \\
 &= \frac{1}{2^1} \sum_{k=0}^2 \binom{2}{k} B_1(2t - k).
 \end{aligned}$$

Infine, iterando il procedimento e sfruttando il teorema binomiale,

$$(x + y)^{l+1} = \sum_{k=0}^{l+1} \binom{l+1}{k} x^{l+1-k} y^k,$$

con  $B_0(2t)$  al posto di  $x$  e  $B_0(2t - 1)$  al posto di  $y$ , si ottiene l'equazione (2.3) di raffinamento per B-spline di grado  $l$ .

### 2.1.4 Raffinamento per curve spline

Una volta verificata la raffinabilità delle curve B-spline si analizza il caso generale di una curva spline. Innanzitutto, una curva spline di grado  $l$  è definita come segue

$$\gamma(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_i \mathbf{p}_i B_{i,l}(t)$$

dove  $(x_i, y_i)^T = \mathbf{p}_i \in \mathbb{R}^2$  sono i punti di controllo; in più, non sarà riportato il grado  $l$  da considerarsi fisso per tutta l'analisi che segue. Si fa notare, inoltre, che, a causa della definizione di  $B_{i,l}(t) = B_l(t - i)$ , ogni punto di controllo esercita un'influenza solo sul tratto di curva tale che  $t \in [i, i + l]$ .

Si consideri la matrice  $P$  dei punti di controllo di una curva data:

$$P = \begin{bmatrix} \vdots \\ \mathbf{p}_{-2} \\ \mathbf{p}_{-1} \\ \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \end{bmatrix}$$

e il vettore  $\mathbf{B}(t)$  costituito dalle traslazioni della funzione  $B(t)$

$$\mathbf{B}(t) = [ \dots \ B(t+2) \ B(t+1) \ B(t) \ B(t-1) \ B(t-2) \ \dots ].$$

Con questa notazione è possibile indicare la curva spline  $\gamma(t)$  come  $\mathbf{B}(t)P$ .

Utilizzando la relazione di raffinamento (2.3) derivata precedentemente, possiamo riscrivere ogni elemento di  $\mathbf{B}(t)$  in termini delle sue dilatazioni

$$\mathbf{B}(2t) = [ \dots \ B(2t+2) \ B(2t+1) \ B(2t) \ B(2t-1) \ B(2t-2) \ \dots ].$$

Precisamente

$$\mathbf{B}(t) = \mathbf{B}(2t)S.$$

La matrice  $S$  è detta *matrice di suddivisione* e i suoi elementi non nulli sono dati dall'equazione (2.3)

$$S_{2i+k,i} = s_k = \frac{1}{2^l} \binom{l+1}{k}.$$

Possiamo utilizzare questa relazione per riscrivere  $\gamma(t)$

$$\gamma(t) = \mathbf{B}(t)P = \mathbf{B}(2t)SP.$$

Si tratta ancora della stessa curva, ma descritta a partire da B-spline dilatate. Sono state sostituite le vecchie funzioni base  $\mathbf{B}(t)$  con le nuove  $\mathbf{B}(2t)$  e contemporaneamente sono stati cambiati i vecchi punti di controllo  $P$  con i nuovi punti di controllo  $SP$ : vale a dire che è stato calcolato un nuovo livello di suddivisione. Questo

procedimento può essere ripetuto più e più volte

$$\begin{aligned}
 \gamma(t) &= \mathbf{B}(t)P^0 \\
 &= \mathbf{B}(2t)P^1 = \mathbf{B}(2t)SP^0 \\
 &= \mathbf{B}(2^2t)P^2 = \mathbf{B}(2^2t)SP^1 \\
 &\quad \vdots \\
 &= \mathbf{B}(2^j t)P^j = \mathbf{B}(2^j t)SP^{j-1},
 \end{aligned}$$

la relazione tra punti di controllo, a diversi livelli di suddivisione, risulta essere

$$P^{j+1} = SP^j.$$

Analizzando ora un singolo punto di controllo si può vedere che

$$\mathbf{p}_i^{j+1} = \sum_l S_{il} \mathbf{p}_l^j. \quad (2.6)$$

Per identificare esattamente quali coefficienti  $s_{il}$  agiscono sui  $\mathbf{p}_l^j$ , si scrive l'espressione (2.6) per i vertici di posto dispari e per quelli di posto pari. Per i termini dispari si ottiene

$$\mathbf{p}_{2i+1}^{j+1} = \sum_l S_{2i+1,l} \mathbf{p}_l^j = \sum_l s_{2(i-l)+1} \mathbf{p}_l^j,$$

mentre per i pari

$$\mathbf{p}_{2i}^{j+1} = \sum_l S_{2i,l} \mathbf{p}_l^j = \sum_l s_{2(i-l)+1} \mathbf{p}_l^j,$$

da cui sostanzialmente si ricavano due regole di suddivisione differenti: una per i nuovi punti di controllo della curva con indice *pari* e una per i nuovi punti *dispari*.

Ad esempio, per la suddivisione lineare definita a tratti, i coefficienti dispari sono  $\frac{1}{2}$  e  $\frac{1}{2}$ , mentre per i pari è un solo 1. Per il caso lineare, quindi, un modo per vedere la differenza tra punti dispari e punti pari è di notare che i primi, al livello  $j + 1$ , sono nuovi punti inseriti, mentre i secondi, al medesimo livello  $j + 1$ , corrispondono ai vecchi punti del livello  $j$ . Si vedrà in seguito che schemi di suddivisione che hanno questa proprietà sono detti *interpolanti*, poiché i punti, una volta calcolati, appartengono all'insieme successivo. Al contrario, considerando ad esempio spline cubiche, i punti al livello  $j + 1$  sono medie locali dei punti al livello  $j$ , ottenendo così che  $\mathbf{p}_{2i}^{j+1} \neq \mathbf{p}_i^j$ . Schemi di quest'ultimo tipo sono detti *approssimanti*. Per spline cubiche i coefficienti dispari risultano essere  $\frac{1}{2}$  e  $\frac{1}{2}$ , mentre per i pari  $\frac{1}{8}$ ,  $\frac{6}{8}$  e  $\frac{1}{8}$  (si veda la Figura 2.3).

Concludendo, si è visto come sia possibile raffinare la sequenza dei punti di con-

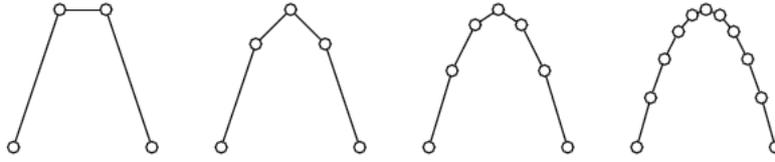


Figura 2.3: Passi di suddivisione per una spline cubica.

trollo che definiscono una data spline moltiplicando il vettore dei punti di controllo  $P$  per la matrice  $S$ , che codifica le equazioni di raffinamento per le B-spline utilizzate nella definizione della curva. Ripetendo questo processo più e più volte ( $j \rightarrow \infty$ ) la spezzata congiungente i punti di controllo  $\mathbf{p}_i^j$  converge alla curva spline. La velocità di convergenza è geometrica, vale a dire che la differenza tra la curva e i suoi punti di controllo decresce di un fattore costante ad ogni passo di suddivisione. Ciò significa che la curva reale e i suoi punti di controllo sono difficilmente distinguibili dopo pochi passi di suddivisione. Quest'ultima osservazione è il cuore del paradigma di suddivisione: invece di disegnare la curva reale si disegnano i suoi punti di controllo e la curva lineare definita a tratti che li attraversa.

### 2.1.5 Algoritmo di Chaikin per la suddivisione di curve

A partire da un iniziale poligono di controllo, l'algoritmo di Chaikin è basato sul taglio degli angoli per la definizione di curve di forme libere mediante raffinamenti ricorsivi [5]. Al limite, l'algoritmo di Chaikin produce curve B-spline quadratiche uniformi.

La Figura 2.4 mostra una curva chiusa suddivisa attraverso l'algoritmo di Chaikin. Ogni vertice del poligono di controllo raffinato è calcolato come una combinazione affine dei vecchi vertici dell'intorno.

L'algoritmo di Chaikin ha due elementi comuni a tutti gli schemi di suddivisione: le regole topologiche e le regole geometriche<sup>2</sup>. Le regole topologiche della suddivisione di Chaikin sono mostrate nella Figura 2.5, spesso sono chiamate regole del taglio degli angoli. Per ogni vertice  $\mathbf{p}_i^j$ , viene tagliato l'angolo corrispondente inserendo due nuovi punti  $\mathbf{p}_{2i}^{j+1}$  e  $\mathbf{p}_{2i+1}^{j+1}$  collegati dal nuovo lato  $\mathbf{p}_{2i}^{j+1}\mathbf{p}_{2i+1}^{j+1}$ . In questo modo la lunghezza dei lati preesistenti viene ridotta (il vecchio lato  $\mathbf{p}_i^j\mathbf{p}_{i+1}^j$  diventa  $\mathbf{p}_{2i+1}^{j+1}\mathbf{p}_{2i+2}^{j+1}$ ).

<sup>2</sup>Chi non è familiare con i termini utilizzati in questo contesto può anticipare la lettura della Sezione 3.1 che chiarisce il significato della terminologia adottata.

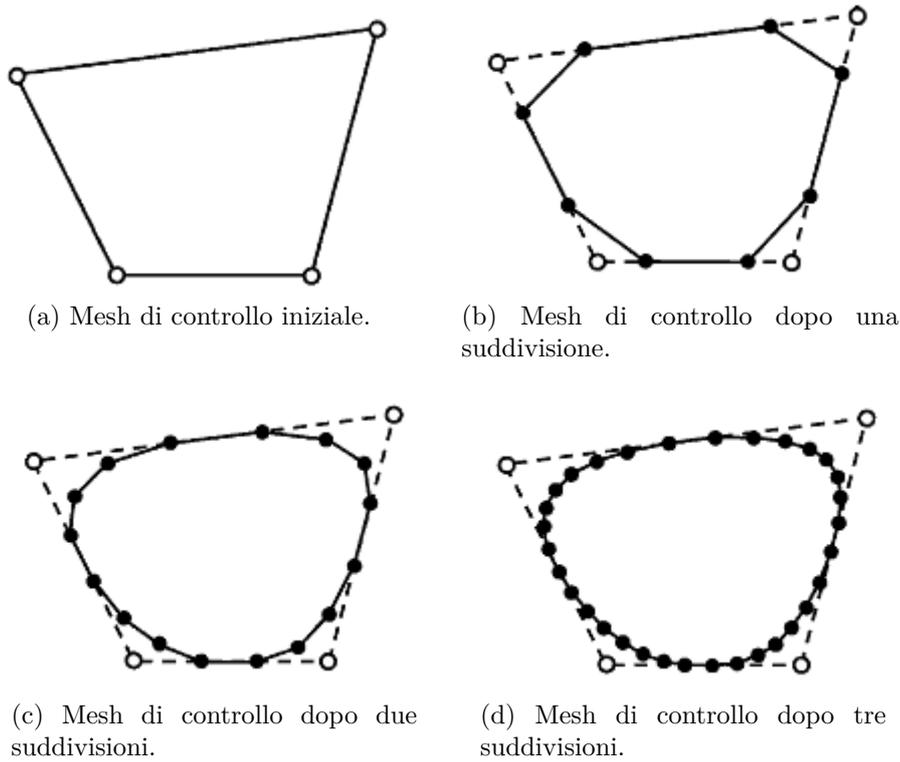


Figura 2.4: Suddivisione di una curva mediante l'algoritmo di Chaikin.

Le regole geometriche dell'algoritmo di Chaikin sono

$$\begin{aligned}
 \mathbf{p}_{2i}^{j+1} &= \frac{1}{4}\mathbf{p}_{i-1}^j + \frac{3}{4}\mathbf{p}_i^j, \\
 \mathbf{p}_{2i+1}^{j+1} &= \frac{3}{4}\mathbf{p}_{i-1}^j + \frac{1}{4}\mathbf{p}_i^j.
 \end{aligned} \tag{2.7}$$

Per chiarezza e facilità di implementazione, le equazioni (2.7) sono spesso rappresentate attraverso una maschera di suddivisione, come mostrato nella Figura 2.6. Un nuovo vertice inserito (punto nero) è calcolato come una combinazione lineare dei vertici originali (cerchi bianchi). I coefficienti sono segnati sopra i vertici corrispondenti. Si noti che i coefficienti sono quelli corrispondenti al raffinamento di B-spline quadratiche.

## 2.2 Suddivisione come raffinamento ripetuto

### 2.2.1 Convoluzione discreta

Si introduce ora la convoluzione discreta per poter dimostrare la convergenza del processo di suddivisione. Per tale scopo si utilizzeranno strumenti per l'analisi di serie quali la trasformata Z e la trasformata di Fourier. Infatti, i coefficienti  $s_k$  del-

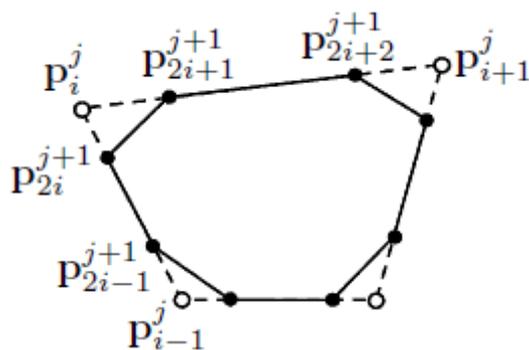


Figura 2.5: Regole topologiche della suddivisione di Chaikin.



Figura 2.6: Maschere della suddivisione di Chaikin.

l'equazione di raffinamento di una B-spline possono essere derivati anche da un'altra prospettiva, chiamata appunto convoluzione discreta. Questo approccio segue da vicino la definizione di B-spline come convoluzione continua, e permette di derivare e verificare molte proprietà utili sulla suddivisione analizzando dei semplici polinomi.

Si ricorda che la funzione di generazione di una sequenza  $a_k$  è definita come

$$A(z) = \sum_{k \in \mathbb{Z}} a_k z^k,$$

$A(z)$  è la trasformata Z di una sequenza  $a_k$ . Questa rappresentazione è fortemente correlata alla trasformata discreta di Fourier di  $a = \{a_k\}$ . E' sufficiente, infatti, restringere l'argomento  $z$  al cerchio di raggio unitario  $z = \exp(i\theta)$ .

La convoluzione di due vettori  $\mathbf{a}$  e  $\mathbf{b}$  è il vettore  $\mathbf{c}$  le cui componenti sono

$$c_k = (a \otimes b)_k = \sum_n a_{k-n} b_n.$$

E' noto che la funzione generatrice di  $\mathbf{c}$  è

$$C(z) = A(z)B(z).$$

Il principale vantaggio delle funzioni generatrici, e il motivo per cui sono utilizzate

in questo contesto, riguarda la possibilità di manipolare le sequenze mediante semplici operazioni sulle funzioni generatrici stesse. Un esempio molto utile di quanto detto si riscontra con la prossima osservazione.

Siano date due funzioni che soddisfano l'equazione di raffinamento

$$\begin{aligned} f(t) &= \sum_k a_k f(2t - k), \\ g(t) &= \sum_k b_k g(2t - k). \end{aligned}$$

In questo caso la convoluzione  $h = f \otimes g$  di  $f$  e  $g$  soddisfa anch'essa un'equazione di raffinamento

$$h(t) = \sum_k c_k h(2t - k),$$

i cui coefficienti  $c_k$  sono dati dalla convoluzione dei coefficienti delle singole equazione di raffinamento

$$c_k = \frac{1}{2} \sum_i a_{k-i} b_i.$$

Si vedrà a breve che con questa piccola osservazione si possono velocemente ricavare l'equazione di raffinamento, e con essa i coefficienti della matrice di suddivisione  $S$ , mediante moltiplicazione ripetute delle funzioni generatrici.

La funzione box  $B_o(t)$  soddisfa l'equazione di raffinamento (2.4) ed inoltre la funzione di generazione di tale espressione è  $A(z) = (1 + z)$ , dato che gli unici termini non nulli dell'equazione di raffinamento sono quelli con indici 0 e 1, che risultano uguali a 1.

Ricordando che la B-spline di grado  $l$  è

$$B_l(t) = \bigotimes_{k=0}^l B_o(t),$$

si ottiene immediatamente che la funzione generativa associata è

$$S(z) = \frac{1}{2^l} (1 + z)^{l+1}.$$

I valori  $s_k$  utilizzati per la definizione della matrice di suddivisione sono semplicemente i coefficienti delle varie potenze di  $z$  nel polinomio  $S(z)$

$$S(z) = \frac{1}{2^l} \sum_{k=0}^{l+1} \binom{l+1}{k} z^k,$$

dove è stato utilizzato il teorema binomiale per espandere  $S(z)$ .

Con il Teorema 1 è stato dimostrato che una B-spline di grado  $n$  è  $C^{n-1}$ -continua. Lo stesso teorema può essere riscritto in termini di funzioni generatrici come segue.

**Teorema 2.** *Se  $S(z)$  definisce uno schema di suddivisione convergente ad una funzione limite  $C^k$ -continua, allora  $\frac{1}{2}(1+z)S(z)$  definisce uno schema di suddivisione convergente in una funzione limite  $C^{k+1}$ -continua.*

Restano ancora da definire i criteri di convergenza ad una funzione continua per uno schema di suddivisione; criteri che saranno oggetto della prossima sezione.

### 2.2.2 Convergenza di uno schema di suddivisione

Esistono diverse nozioni di convergenza, in questo contesto ci si riferisce alla convergenza uniforme.

Si dice che una sequenza di funzioni  $f_i$ , definite in un intervallo  $[a, b] \subset \mathbb{R}$ , converge uniformemente ad una funzione limite  $f$  se

$$\forall \varepsilon > 0, \exists n_0 > 0 : \forall n > n_0 \max_{t \in [a, b]} |f(t) - f_n(t)| < \varepsilon.$$

Inoltre, se una sequenza di funzioni continue converge uniformemente ad una certa funzione limite  $f$ , allora tale funzione limite è essa stessa continua.

E' chiaro che la sequenza di funzioni che si intende analizzare è quella dei poligoni di controllo che vengono raffinati applicando le regole di suddivisione  $S$ . Indipendentemente da quest'ultime, è possibile utilizzare le B-spline lineari per ricavare la curva lineare definita a tratti che passa per i punti di controllo come

$$H^j(t) = \mathbf{B}_1(2^j t) P^j,$$

al livello di suddivisione  $j$ .

Un modo per mostrare che un dato schema di suddivisione  $S$  converge ad una funzione limite continua è dimostrare che per ogni  $t$  esiste il limite

$$H^\infty(t) = \lim_{j \rightarrow \infty} H^j(t)$$

e la successione  $H^j(t)$  converge uniformemente. Al fine di dimostrare tale proprietà, si assume che tutte le righe della matrice  $S$  sommino 1, in altre parole i coefficienti dispari e pari della relazione di raffinamento sommano separatamente 1. Questa è una richiesta ragionevole dato che è necessaria per garantire l'invarianza affine del processo di suddivisione, come si vedrà più avanti. Utilizzando una notazione matriciale questo significa  $S\mathbf{1} = \mathbf{1}$ , in altre parole, il vettore con soli 1 è un autovettore

della matrice di suddivisione con autovalore 1. In termini di funzione di generazione equivale a  $S(-1) = 0$ .

Una funzione  $f(t)$  è continua in  $t_0$  se per ogni  $\varepsilon > 0$  esiste  $\delta > 0$  tale che se  $|t_0 - t| < \delta$  allora  $|f(t_0) - f(t)| < \varepsilon$ . Quanto appena enunciato, in termini di suddivisione, si traduce nella necessità di verificare che la differenza tra due punti di controllo adiacenti  $\mathbf{p}_{i+1}^j - \mathbf{p}_i^j = (\Delta \mathbf{p}^j)_i$  si riduca velocemente in modo che esista una curva limite e che sia continua. Più precisamente, si ha la seguente

**Lemma 1.** *Se  $\|\Delta P^j\| < c\gamma^j$  per una certa costante  $c > 0$  e un fattore di restringimento  $0 < \gamma < 1$  per ogni  $j > j_0 \geq 0$ , allora  $H^j(t)$  converge ad una funzione limite continua  $H^\infty(t)$ <sup>3</sup>.*

*Dimostrazione.* Come riportato in [34], sia  $S$  la matrice di suddivisione,  $P^1 = SP^0$  e  $S_1$  la regola di suddivisione di una B-spline di grado 1. Si noti che le righe di  $S - S_1$  sommano 0

$$(S - S_1)\mathbf{1} = S\mathbf{1} - S_1\mathbf{1} = \mathbf{1} - \mathbf{1} = \mathbf{0}.$$

Ciò implica l'esistenza di una matrice  $D$  tale che  $S - S_1 = D\Delta$ , dove  $\Delta$  calcola la differenza di elementi adiacenti ( $(\Delta)_{ii} = -1$ ,  $(\Delta)_{i,i+1} = 1$  e zero altrove). Gli elementi di  $D$  sono definiti come  $D_{ij} = -\sum_{k=i}^j (S - S_1)_{ik}$ . Ora si consideri la differenza tra due approssimazioni lineari successive dei punti di controllo

$$\begin{aligned} \|H^{j+1}(t) - H^j(t)\| &= \|\mathbf{B}_1(2^{j+1}t)P^{j+1} - \mathbf{B}_1(2^j t)P^j\| \\ &= \|\mathbf{B}_1(2^{j+1}t)SP^j - \mathbf{B}_1(2^{j+1}t)S_1P^j\| \\ &= \|\mathbf{B}_1(2^{j+1}t)(S - S_1)P^j\| \\ &\leq \|\mathbf{B}_1(2^{j+1}t)\| \|D\Delta P^j\| \\ &\leq \|D\| \|\Delta P^j\| \\ &\leq \|D\| c\gamma^j. \end{aligned}$$

Questo comporta che la somma  $H^0(t) + \sum_{k=0}^j (H^{k+1} - H^k)(t)$  converge ad una funzione limite ben definita dato che la norma di ogni somma è limitata da una costante.

---

<sup>3</sup>Si introducono i simboli di norma

$$\begin{aligned} \|f(t)\| &= \sup_t |f(t)| \\ \|\mathbf{p}\| &= \sup_i |p_i| \\ \|S\| &= \sup_i \sum_k |S_{ik}| \end{aligned}$$

Infine, sfruttando la coda delle serie geometriche, per  $j \rightarrow \infty$  vale

$$\|H^\infty(t) - H^j(t)\| < \frac{\|D\|c}{1-\gamma}\gamma^j$$

che implica la convergenza uniforme e dunque la continuità di  $H^\infty(t)$  come si voleva dimostrare.  $\square$

Per verificare se uno schema di suddivisione converge si analizza lo schema di *suddivisione alle differenze*

$$\Delta P^{j+1} = D\Delta P^j,$$

definito come lo schema che soddisfa

$$\Delta S = D\Delta.$$

Dato lo schema  $D$ , bisogna solo dimostrare che esiste una potenza  $m > 0$  di  $D$  che ha una norma inferiore a 1,  $\|D^m\| = \gamma < 1$ . In questo modo  $\|\Delta P^j\| < c(\gamma^{1/m})^j$ .

Come esempio si riporta la verifica delle condizioni appena enunciate per una B-spline cubica. La funzione generatrice è  $B_3(z) = \frac{1}{8}(1+z)^4$  e quindi le equazioni di raffinamento sono

$$\begin{aligned} \mathbf{p}_{2i+i}^{j+1} &= \frac{1}{8}(4\mathbf{p}_i^j + 4\mathbf{p}_{i+1}^j) \\ \mathbf{p}_{2i}^{j+1} &= \frac{1}{8}(\mathbf{p}_{i-1}^j + 6\mathbf{p}_{i+1}^j + \mathbf{p}_{i+1}^j). \end{aligned}$$

Prendendo le differenze si ottiene per i punti pari

$$\begin{aligned} (\Delta \mathbf{p}^{j+1})_{2i} &= \mathbf{p}_{2i+1}^{j+1} - \mathbf{p}_{2i}^{j+1} \\ &= \frac{1}{8}(-\mathbf{p}_{i-1}^j - 2\mathbf{p}_i^j + 3\mathbf{p}_{i+1}^j) \\ &= \frac{1}{8}(3(\mathbf{p}_{i+1}^j - \mathbf{p}_i^j) + 1(\mathbf{p}_i^j - \mathbf{p}_{i-1}^j)) \\ &= \frac{1}{8}(3(\Delta \mathbf{p}^j)_i + 1(\Delta \mathbf{p}^j)_{i-1}), \end{aligned}$$

e un'espressione analoga per i punti dispari. Possiamo ricavare, quindi, che  $D(z) = \frac{1}{8}(1+z)^3$  e concludere che  $\|D\| = \frac{1}{2}$ . Dunque lo schema di suddivisione di una B-spline cubica converge uniformemente ad una funzione limite continua, vale a dire la B-spline stessa.

In generale, la parte difficile è trovare un insieme di coefficienti per cui la suddivisione converge. Non esiste un metodo generale per raggiungere tale obiettivo.

Una volta identificato uno schema di suddivisione convergente, è sempre possibile ottenere l'ordine di continuità desiderato convolvendo con la funzione box.

Fin qui è stata considerata solo la suddivisione di spline dove la regola di suddivisione, vale a dire i coefficienti utilizzati per calcolare un insieme di punti di controllo più raffinato, è fissa e la medesima in ogni punto. In realtà non c'è alcun motivo per cui debba essere così, è possibile creare una grande varietà di curve differenti manipolando i coefficienti della matrice di suddivisione. Questo può essere fatto globalmente o localmente, si possono cambiare i coefficienti in uno stesso livello di suddivisione oppure tra differenti livelli (in seguito parleremo di schemi di suddivisione *stazionari* e *non stazionari*).

A questo proposito, le spline possono essere viste come un caso particolare di una famiglia di curve più generale: le curve di suddivisione.

## 2.3 Analisi della matrice di suddivisione

### 2.3.1 Intorno invariante

Si supponga di voler studiare la curva limite di un dato schema di suddivisione nelle vicinanze di un particolare punto di controllo (che d'ora in avanti assumeremo essere sempre l'origine, condizione sempre verificabile rinumerando appropriatamente i punti di controllo). Per determinare le proprietà *locali* di una curva di suddivisione non serve l'intero vettore dei punti di controllo e nemmeno l'intera matrice di suddivisione. La differenziabilità, ad esempio, è una proprietà locale della curva e per studiarla è necessaria solo una piccola parte della curva vicina all'origine. Questo porta alla questione di quali punti di controllo influenzano la curva nel vicinato dell'origine.

Per rispondere alla domanda di cui sopra, si riscriva la curva del poligono di controllo come segue

$$\begin{aligned}
 H^j(t) &= \mathbf{B}_1(2^j t) S^j P^0 \\
 &= \mathbf{B}_1(2^j t) S^j \left( \sum_i \mathbf{p}_i^0 \mathbf{e}_i \right) \\
 &= \sum_i \mathbf{p}_i^0 \mathbf{B}_1(2^j t) S^j \mathbf{e}_i \\
 &= \sum_i \mathbf{p}_i^0 \varphi_i^j(t)
 \end{aligned}$$

dove si è posto  $\varphi_i^j(t) := \mathbf{B}_1(2^j t) S^j \mathbf{e}_i$  e in cui si è sfruttata la linearità dell'operatore di suddivisione e dove  $\mathbf{e}_i$  rappresenta il vettore riga in cui tutti gli elementi sono nulli

tranne quello nella posizione  $i$  che vale 1. In altre parole, la curva finale è sempre una combinazione lineare con pesi  $\mathbf{p}_i^0$  della cosiddetta *soluzione fondamentale*

$$\lim_{j \rightarrow \infty} \varphi_i^j(t) = \varphi_i(t).$$

Utilizzando gli stessi pesi di suddivisione sull'intero dominio è facile vedere che  $\varphi_i(t) = \varphi(t - i)$ , in altre parole esiste una sola funzione  $\varphi(t)$ , detta soluzione fondamentale dello schema di suddivisione, tale per cui tutte le curve ottenute mediante suddivisione di una iniziale sequenza di punti  $P^0$  sono combinazioni lineari di traslazioni di  $\varphi(t)$  stessa.

Questioni quali la differenziabilità della curva limite possono essere ora studiate esaminando la soluzione fondamentale  $\varphi(t)$  definita come

$$\varphi(t) = \lim_{j \rightarrow \infty} S^j \mathbf{e}_0.$$

Per esempio, è possibile determinare, analizzando il supporto di questa funzione, fino a che distanza si sente l'influenza di un punto di controllo. In modo analogo, la forma di tale funzione ci dà delle informazioni su come la curva (o la superficie) cambierà inserendo un nuovo punto di controllo. Si noti, inoltre, che nel caso di superfici le regole che si applicano dipendono dalla valenza<sup>4</sup> del vertice in questione, ottenendo così diverse soluzioni fondamentali per ciascuna valenza.

E' possibile ora rispondere alla domanda di quanto sia grande l'intorno invariante. Come esempio, si consideri la suddivisione di una B-spline cubica le cui funzioni base hanno un supporto di 4 intervalli. Se siamo interessati ad un piccolo intervallo aperto centrato nell'origine, ad esempio nell'intervallo  $[-1, 1]$ , possiamo notare che 5 funzioni base si sovrappongono a tale intervallo. Si dice appunto che il vicinato invariante ha dimensione 5 perché 5 punti di controllo (associati alle 5 funzioni base) determinino il comportamento della curva limite in prossimità dell'origine indipendentemente dal livello di suddivisione. Con i 5 punti di controllo centrali al livello  $j$  è possibile calcolare i 5 punti di controllo centrali al livello  $j + 1$ . Quanto detto ci permette di studiare il comportamento della curva nell'origine analizzando solo una piccola sotto-matrice  $5 \times 5$  (detta *matrice di suddivisione locale*) della

---

<sup>4</sup>La valenza di un vertice è pari al numero di lati in esso incidenti.

matrice di suddivisione

$$\begin{pmatrix} \mathbf{p}_{-2}^{j+1} \\ \mathbf{p}_{-1}^{j+1} \\ \mathbf{p}_0^{j+1} \\ \mathbf{p}_1^{j+1} \\ \mathbf{p}_2^{j+1} \end{pmatrix} = \frac{1}{8} \begin{pmatrix} 1 & 6 & 1 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & 1 & 6 & 1 & 0 \\ 0 & 0 & 4 & 4 & 0 \\ 0 & 0 & 1 & 6 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_{-2}^j \\ \mathbf{p}_{-1}^j \\ \mathbf{p}_0^j \\ \mathbf{p}_1^j \\ \mathbf{p}_2^j \end{pmatrix}.$$

La dimensione dell'intorno invariante dipende dal numero di elementi non nulli di ciascuna riga della matrice di suddivisione, che per una cubica sono 2 per i punti dispari e 3 per i pari. A causa di questi ultimi punti è necessario un punto extra a sinistra del punto di controllo -1 e uno a destra dell'1 (si ricorda che il segmento cubico sulla sinistra dell'origine ha parametri  $t \in [-1, 0]$  e il segmento sulla destra nel range  $t \in [0, 1]$ ). Nella Figura 2.7 è riassunto schematicamente quanto appena enunciato.

La Figura 2.7 illustra che sono necessari 5 punti di controllo al livello iniziale per determinare il comportamento della curva limite nei due tratti adiacenti l'origine. Ad ogni livello è necessario un altro punto di controllo esterno all'intervallo  $t \in [-1, 1]$  per continuare la suddivisione.

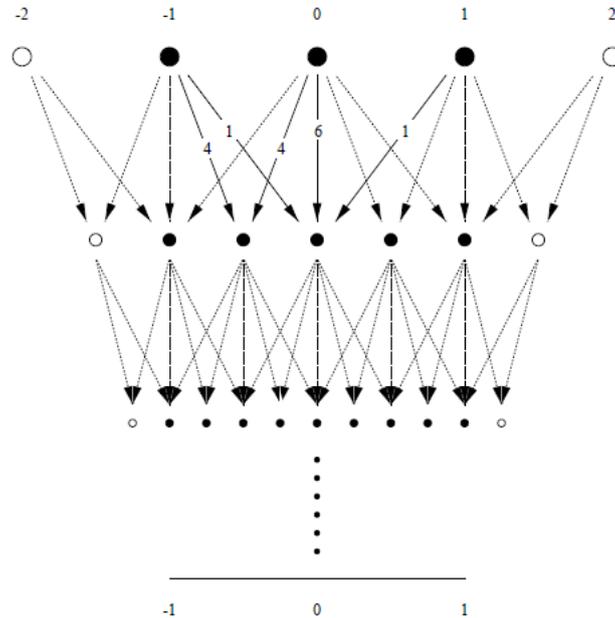


Figura 2.7: Dimensione del vicinato invariante per la suddivisione di B-spline cubiche.

In sintesi, la matrice di suddivisione locale controlla il comportamento della curva nell'intorno dell'origine, non sorprenderà dunque che molte proprietà delle curve generate dalla suddivisione possono essere dedotte da proprietà di tale matrice. In

particolare, le proprietà di differenziabilità della curva sono legate alla struttura degli autovalori della matrice di suddivisione locale. Da ora in avanti il simbolo  $S$  si riferirà alla matrice di suddivisione *locale*.

### 2.3.2 Analisi degli autovalori

Si ricorda dall'algebra lineare che un *autovettore*  $\mathbf{x}$ , di una matrice  $M$ , è un vettore non nullo tale per cui  $M\mathbf{x} = \lambda\mathbf{x}$ , dove  $\lambda$  è uno scalare detto *autovalore* associato al corrispondente autovettore  $\mathbf{x}$ .

Si assuma che la matrice di suddivisione locale  $S$  abbia dimensione  $n \times n$  e autovettori reali  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$  che formano una base e corrispondono agli autovalori reali  $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{n-1}$ . Per esempio, nel caso di una spline cubica,  $n = 5$  e

$$\begin{aligned} (\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4) &= \left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) \\ (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) &= \begin{pmatrix} 1 & -1 & 1 & 1 & 0 \\ 1 & -\frac{1}{2} & \frac{2}{11} & 0 & 0 \\ 1 & 0 & -\frac{1}{11} & 0 & 0 \\ 1 & \frac{1}{2} & \frac{2}{11} & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Dati questi autovettori si ottiene

$$\begin{aligned} S(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) &= (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) \begin{pmatrix} \lambda_0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & 0 & \lambda_4 \end{pmatrix} \\ SX &= XD \\ X^{-1}SX &= D. \end{aligned}$$

Le righe  $\tilde{\mathbf{x}}_i$  di  $X^{-1}$  sono dette autovettori sinistri dato che soddisfano  $\tilde{\mathbf{x}}_i S = \lambda_i \tilde{\mathbf{x}}_i$ , ottenuta moltiplicando a destra l'ultima uguaglianza per  $X^{-1}$ .

In realtà non tutti gli schemi di suddivisione hanno autovalori reali e un insieme completo di autovettori. Queste degenerazioni sono il motivo principale della difficoltà della teoria della suddivisione. Per mantenere questa esposizione semplice e comunicare solo le idee essenziali saranno ignorati questi casi e si assumerà sempre che sia disponibile un insieme completo di autovettori.

Con questa impostazione è possibile scrivere un qualsiasi vettore  $\mathbf{p}$  di lunghezza  $n$  come una combinazione lineare di autovettori:

$$\mathbf{p} = \sum_{i=0}^{n-1} a_i \mathbf{x}_i,$$

dove gli  $a_i$  sono dati dal prodotto interno  $a_i = \tilde{\mathbf{x}}_i \cdot \mathbf{p}$ . Questa decomposizione è valida anche quando gli elementi di  $\mathbf{p}$  sono  $n$  punti 2-D (o punti 3-D se si tratta di superfici) invece di scalari. In questo caso ciascun coefficiente  $a_i$  sarà anch'esso un punto 2-D (o 3-D), mentre gli autovettori  $\mathbf{x}_0, \dots, \mathbf{x}_{n-1}$  sono semplicemente vettori di  $n$  numeri reali.

E' possibile scrivere il risultato dell'applicazione della matrice di suddivisione ad un vettore di punti di controllo (cioè i punti di controllo al livello successivo) rispetto alla base degli autovettori in modo molto semplice

$$\begin{aligned} SP^0 &= S \sum_{i=0}^{n-1} a_i \mathbf{x}_i \\ &= \sum_{i=0}^{n-1} a_i S \mathbf{x}_i \\ &= \sum_{i=0}^{n-1} a_i \lambda_i \mathbf{x}_i, \end{aligned}$$

dove ancora una volta è stata sfruttata la linearità di  $S$ .

A questo punto applicando  $j$  volte l'operatore di suddivisione  $S$  si ottiene

$$P^j = S^j P^0 = \sum_{i=0}^{n-1} a_i \lambda_i^j \mathbf{x}_i.$$

Se  $\lambda_0 > 1$  allora  $S^j P^0$  cresce illimitatamente al crescere di  $j$  e dunque lo schema di suddivisione non è convergente. Perciò, affinché la sequenza  $S^j P^0$  sia convergente è necessario che tutti gli autovalori siano al più unitari. Inoltre, è possibile dimostrare che solo un unico autovalore può avere modulo unitario per garantire la convergenza [22].

Una semplice conseguenza di questa analisi è la possibilità di calcolare direttamente la posizione limite rispetto alla base degli autovettori

$$P^\infty(0) = \lim_{j \rightarrow \infty} S^j P^0 = \lim_{j \rightarrow \infty} \sum_{i=0}^{n-1} a_i \lambda_i^j \mathbf{x}_i = a_0,$$

dato che tutti gli autovalori con modulo  $< 1$  decadono a zero per  $j \rightarrow \infty$ . Per esempio, nel caso di suddivisione di B-spline cubiche, si può calcolare la posizione

limite di  $\mathbf{p}_i^j$  come  $a_0 = \tilde{\mathbf{x}}_0 \cdot \mathbf{p}^j$ , che corrisponde a

$$\mathbf{p}_i^\infty = a_0 = \frac{1}{6}(\mathbf{p}_{i-1}^j + 4\mathbf{p}_i^j + \mathbf{p}_{i+1}^j).$$

Si noti che questa espressione è completamente indipendente dal livello  $j$  in cui viene calcolata.

### 2.3.3 Invarianza rispetto alle trasformazioni affini

Se si spostano contemporaneamente tutti i punti di controllo di una certa quantità, la curva definita da tali punti di controllo deve muoversi come se fosse un oggetto rigido. In altre parole, la curva dovrebbe essere *invariante alle trasformazioni che preservano le distanze*, come la traslazione e la rotazione. Si verifica di seguito tale proprietà.

Innanzitutto, segue dalla linearità della suddivisione che se la suddivisione è invariante ad una trasformazione che preserva le distanze sarà anche *invariante ad una trasformazione affine*. Sia  $\mathbf{1}$  un vettore  $n$ -dimensionale e  $\mathbf{a} \in \mathbb{R}^2$  uno spostamento nel piano (si veda la Figura 2.8). Allora  $\mathbf{1} \cdot \mathbf{a}$  rappresenta uno spostamento di tutti i punti pari al vettore  $\mathbf{a}$ . Applicando la suddivisione ai vettori trasformati, si ottiene

$$\begin{aligned} S(P^j + \mathbf{1} \cdot \mathbf{a}) &= SP^j + S(\mathbf{1} \cdot \mathbf{a}) \\ &= P^{j+1} + S(\mathbf{1} \cdot \mathbf{a}). \end{aligned}$$

Da questa espressione è evidente che per avere l'invarianza alla traslazione è necessario che

$$S(\mathbf{1} \cdot \mathbf{a}) = \mathbf{1} \cdot \mathbf{a},$$

quindi  $\mathbf{1}$  dovrebbe essere un autovettore di  $S$  con autovalore associato  $\lambda_0 = 1$ .

Si ricorda, infine, che quando è stata provata la convergenza della suddivisione è stato assunto che  $\mathbf{1}$  fosse un autovettore con autovalore 1 della matrice di suddivisione. Ora si è visto che tale assunzione dev'essere soddisfatta da qualunque schema di suddivisione ragionevole.

### 2.3.4 Comportamento geometrico di suddivisioni ripetute

Se si assume  $\lambda_0 = 1$ , e tutti gli altri autovalori inferiori a 1, è possibile scegliere le coordinate del sistema in modo che  $a_0$  sia l'origine in  $\mathbb{R}^2$ . In questo modo si ottiene

$$\mathbf{p}^j = \sum_{i=1}^{n-1} a_i \lambda_i^j \mathbf{x}_i.$$

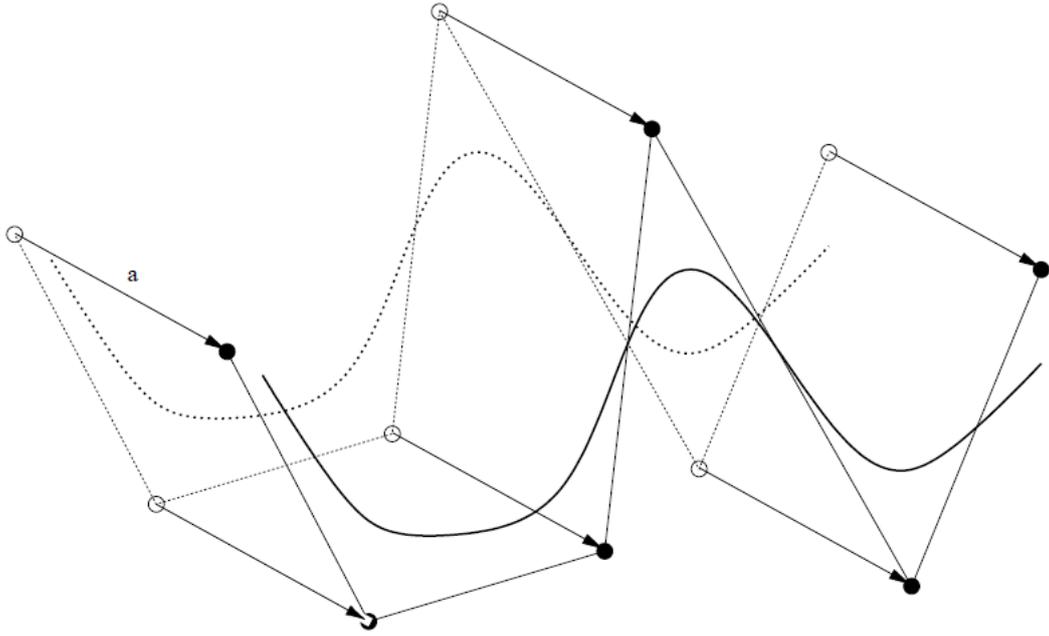


Figura 2.8: Invarianza alla traslazione.

Dividendo entrambi i termini per  $\lambda_1^j$ , si ricava

$$\frac{1}{\lambda_1^j} \mathbf{p}^j = a_1 \mathbf{x}_1 + \sum_{i=2}^{n-1} a_i \left( \frac{\lambda_i}{\lambda_1} \right)^j \mathbf{x}_i.$$

Se si assume che  $|\lambda_2|, \dots, |\lambda_{n-1}| < |\lambda_1|$  la somma sulla destra tende a 0 per  $j \rightarrow \infty$ . In altre parole, il termine corrispondente a  $\lambda_1$  dominerà il comportamento del vettore dei punti di controllo. Al limite, si ottengono un insieme di  $n$  punti posizionati lungo il vettore  $a_1$ . Geometricamente questo è il vettore tangente alla curva nel punto centrale.

Analogamente al calcolo del punto limite della suddivisione di una B-spline cubica mediante il calcolo di  $a_0$ , è possibile calcolare il vettore tangente nel punto  $\mathbf{p}_i^j$  calcolando  $a_1 = \tilde{\mathbf{x}}_1 \cdot \mathbf{p}^j$

$$\mathbf{t}_i^\infty = a_1 = \mathbf{p}_{i+1}^j - \mathbf{p}_{i-1}^j.$$

Nel caso in cui ci siano due autovalori uguali, diciamo  $\lambda_1 = \lambda_2$ , al crescere di  $j$  i punti sulla configurazione limite saranno combinazioni lineari dei due vettori  $a_1$  e  $a_2$ , ed in generale non apparterranno alla stessa retta. Ciò indica che non esiste un vettore tangente nel punto centrale. Possiamo quindi riportare la seguente

**Proposizione 1.** *Condizione necessaria per l'esistenza del vettore tangente è che tutti gli autovalori di  $S$ , eccetto  $\lambda_0 = 1$ , siano in modulo minori di  $\lambda_1$ .*

In sintesi, per una matrice di suddivisione  $S$  sono desiderabili le seguenti carat-

teristiche

- gli autovettori dovrebbero formare una base;
- il primo autovalore  $\lambda_0$  dovrebbe essere 1;
- il secondo autovalore  $\lambda_1$  dovrebbe essere inferiore a 1;
- tutti gli altri autovalori dovrebbero essere inferiori a  $\lambda_1$ .

## Capitolo 3

# Superfici di Suddivisione uniformi e non uniformi

Questo capitolo è dedicato allo studio degli schemi di suddivisione che generalizzano le superfici B-spline prodotto tensoriale; particolare rilievo viene dato alla differenza tra schemi uniformi e non uniformi. Doo e Sabin [7], Catmull e Clark [4] hanno proposto generalizzazioni delle superfici uniformi permettendo a ciascun vertice del poligono di controllo di avere una valenza diversa da 4 e di gestire anche facce non quadrilatere. Le loro regole di suddivisione sono state progettate in modo tale che in presenza di una mesh con topologia a griglia rettangolare siano equivalenti al raffinamento di una superficie B-spline uniforme. Le superfici di suddivisione sono definite come il limite della mesh di controllo quando le regole di suddivisione sono applicate un numero infinito di volte. Di contro, le *Non-Uniform Recursive Subdivision Surfaces* (NURSS) [27], vale a dire superfici Doo-Sabin e Catmull-Clark non uniformi, generalizzano le superfici B-spline prodotto tensoriale non uniformi con topologia arbitraria. Questa flessibilità aggiunta permette, tra l'altro, la gestione naturale di features quali cuspidi, creste e punte, mantenendo lo stesso ordine di continuità dei rispettivi algoritmi uniformi.

La Sezione 3.1 presenta i concetti chiave degli schemi di suddivisione e fornisce una classificazione degli stessi. Nella Sezione 3.2 sono descritti le superfici di suddivisione uniformi, vale a dire gli schemi di Doo-Sabin e di Catmull-Clark, e come tali schemi possano essere derivati direttamente dalle superfici B-spline uniformi in forma parametrica. Infine, la Sezione 3.3 introduce il raffinamento delle curve B-spline non uniformi, propone un approccio molto semplice per l'etichettatura degli intervalli nodali, un'idea cruciale per l'estensione alle superfici di suddivisione, e discute ampiamente le NURSS.

## 3.1 Concetti chiave e una breve panoramica

In letteratura si possono trovare ricche famiglie di schemi di suddivisione. Quelli proposti in questo lavoro sono generalizzazioni di un sottoinsieme di spline [4, 7, 20], altri schemi sono invece estensioni di box-spline [18, 20] e non saranno analizzati. Tuttavia, prima di addentrarsi nella trattazione di specifici schemi di suddivisione, è bene presentare una classificazione di più ampio respiro e chiarire la terminologia adottata.

### 3.1.1 Classificazione degli schemi di suddivisione

Una prima fondamentale caratterizzazione degli schemi di suddivisione riguarda gli schemi *approssimanti* e quelli *interpolanti*. Se la superficie limite dello schema di suddivisione non attraversa i punti di controllo iniziali il metodo è detto approssimante. Esempi di schemi di suddivisione di questo tipo sono Catmull-Clark [4], Doo-Sabin [7] e Loop [20]. Al contrario, lo schema di suddivisione è interpolante se i vertici della mesh iniziale sono anche vertici della superficie limite. Tipici esempi di schemi interpolanti sono la suddivisione Butterfly [8], gli schemi  $\sqrt{2}$  [18] e  $\sqrt{3}$  [11].

L'interpolazione è una caratteristica attraente sotto diversi aspetti: i punti di controllo iniziali sono anche punti della superficie limite, ciò consente al progettista di controllare la superficie in una maniera più intuitiva; inoltre, molti calcoli possono essere eseguiti a livello locale. Sfortunatamente la qualità di queste superfici non è così elevata come quella delle superfici ottenute con schemi approssimanti. In più, gli schemi interpolanti non convergono altrettanto velocemente alla superficie limite come gli schemi approssimanti. Gli schemi interpolanti sono oggetto di un'intensa ricerca, ma attualmente non esistono schemi soddisfacenti e largamente utilizzati. Per tale motivo in questo lavoro sono trattati solo schemi approssimanti.

Molti schemi di suddivisione esistenti sono detti *uniformi*, questo perché la mesh viene raffinata uniformemente mediante l'inserimento di punti intermedi ai vertici originali lungo l'intera superficie e per tutti i livelli di suddivisione. Tuttavia, schemi di suddivisione non uniformi sono oggi molto studiati (si vedano [3, 16, 25, 28]); un esempio è lo schema di suddivisione NURSS [27] che svolge una suddivisione parametrizzata e non-uniforme. L'intera Sezione 3.3 è dedicata all'analisi delle NURSS.

Se le regole di suddivisione non cambiano durante il processo di raffinamento, lo schema è detto *stazionario*, altrimenti *non stazionario*. La maggior parte degli schemi di suddivisione esistenti sono stazionari. D'altra parte, per ottenere certe

famiglie di forme, come cerchi perfetti, potrebbe essere utilizzato uno schema di suddivisione non stazionario [14, 24].

La maggior parte degli schemi di suddivisione sono progettati per eseguire suddivisioni *globali*, cioè sull'intera mesh. In alcune situazioni, una suddivisione *locale* e adattativa potrebbe essere desiderabile [15, 29]. Tuttavia, non esistono schemi di suddivisione che possono garantire una suddivisione adattativa senza aver effetti sulla superficie limite.

E' naturale gestire mesh con facce dello stesso tipo; in particolare si cerca di utilizzare poligoni regolari: quadrati, triangoli equilateri ed esagoni. Nella pratica, mesh costituite da esagoni non sono molto comuni. Questo porta all'esistenza di soli due tipi di schemi di suddivisione: quelli definiti per tassellazioni *quadrilatere*, e quelli definiti per tassellazioni *triangolari* (sono comunque stati proposti schemi che gestiscono contemporaneamente sia facce quadrilatere sia triangolari [32]). Nelle prossime sezioni saranno analizzati solo i principali schemi di suddivisione con mesh a facce quadrilatere perché di maggior interesse per lo scopo di questo lavoro.

Una volta scelto il tipo di tassellazione, è necessario definire come sono legate le facce raffinate con quelle originali. Esistono due approcci principali che sono utilizzati per generare una tassellazione raffinata: una è la divisione di facce e l'altra la divisione di vertici (si veda la Figura 3.1). Gli schemi che utilizzano il primo metodo sono detti *primali*, mentre gli schemi che utilizzano il secondo sono chiamati *duali*. Negli schemi primali di mesh quadrilatere ogni faccia è divisa in altre quattro. I vecchi vertici sono mantenuti, nuovi vertici non inseriti sui lati, un vertice aggiuntivo è inserito per ogni faccia. Negli schemi duali, per ogni vecchio vertice, sono inseriti tanti nuovi vertici quante sono le facce ad esso adiacenti. Viene aggiunta una nuova faccia per ciascuna faccia, lato e vertice del livello di suddivisione precedente. Gli schemi duali sono supportati solo da mesh con facce quadrilatere.

La varietà di schemi esistenti può apparire caotica. Tuttavia, è possibile classificare la maggior parte degli schemi di suddivisione basandosi su quattro criteri:

1. il tipo di regole di raffinamento (divisione delle facce oppure divisione dei vertici),
2. il tipo di mesh generata (triangolare o quadrilatera),
3. lo schema è approssimante o interpolante,
4. la regolarità della superficie limite ( $C^1$ ,  $C^2$ , etc.).

La tabella 3.1 mostra alcuni schemi di suddivisione classificati sulla base dei criteri appena enunciati.

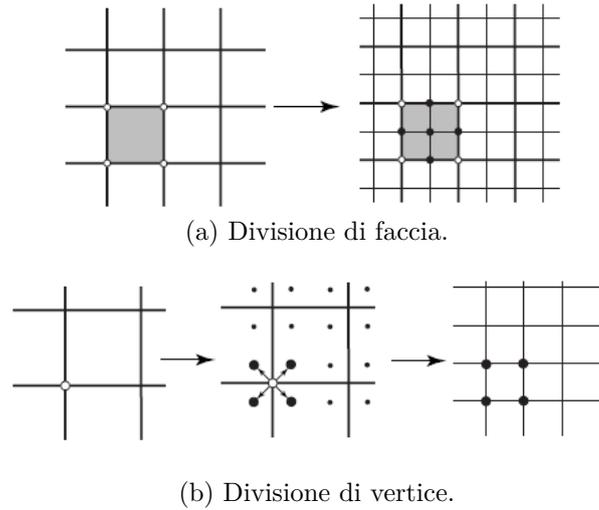


Figura 3.1: Differenti regole di raffinamento per facce quadrilatere.

Suddivisione di faccia		
	Mesh triangolari	Mesh quadrilatere
Approssimanti	Loop ( $C^2$ )	Catmull-Clark ( $C^2$ )
Interpolanti	Butterfly ( $C^1$ )	Kobbelt ( $C^1$ )

(a) Schemi primali.

Suddivisione di vertice	
	Mesh quadrilatere
Approssimanti	Doo-Sabin ( $C^1$ )
Interpolanti	$\sqrt{2}$ ( $C^1$ )

(b) Schemi duali

Tabella 3.1: Classificazione degli schemi di suddivisione.

### 3.1.2 Terminologia

Come già anticipato, uno schema di suddivisione è definito da un insieme di *regole topologiche* e di *regole geometriche* per il raffinamento della mesh. Le regole topologiche definiscono come debba essere spezzata la mesh di controllo per ottenere la mesh raffinata. A seconda del tipo di schema di suddivisione, operazioni usuali di regole topologiche riguardano l’inserimento di nuovi vertici nei lati e nelle facce, l’aggiornamento dei vertici già esistenti, la connessione dei nuovi vertici inseriti e dei vertici aggiornati, l’eliminazione di alcuni vertici, lati e facce. Le regole geometriche, invece, sono utilizzate per calcolare le coordinate esatte dei nuovi vertici di controllo. Quando si progettano le regole geometriche per la suddivisione di mesh,

alcune proprietà chiave devono essere prese in considerazione: l'invarianza affine, il supporto finito con piccole maschere di suddivisione, la simmetria e il comportamento della superficie limite. Si è discusso nel capitolo 2 come tecniche, quali l'analisi della struttura degli autovalori, la trasformata  $Z$  e la trasformata di Fourier, sono spesso utilizzate per selezionare appropriate regole di suddivisione.

Nelle mesh quadrilatere sia la suddivisione di faccia sia quella di vertice creano solo vertici con valenza 4 (se il vertice è sul bordo avrà valenza 3). Dopo alcune suddivisioni, la maggior parte dei vertici avrà valenza 4 (3 se sul bordo). Tali vertici sono detti *regolari*, mentre tutti i vertici con valenza diversa da 4 sono chiamati vertici *straordinari*. L'analisi delle superfici di suddivisione in corrispondenza di vertici straordinari differisce fortemente da quella delle regioni regolari. Uno studio completo e dettagliato è realizzato in [22].

Negli schemi primali, i vertici della mesh grezza sono anche vertici della mesh raffinata. Per ogni livello di suddivisione, tutti i nuovi vertici inseriti sono chiamati *vertici dispari*. Questo termine deriva dal caso monodimensionale, dove i vertici del poligono di controllo possono essere numerati sequenzialmente e ad ogni livello i nuovi vertici inseriti hanno degli indici dispari. Mentre i vertici ereditati dal livello precedente sono detti *pari* (si veda il capitolo 2). Negli schemi quadrilateri, alcuni vertici sono inseriti in corrispondenza della divisione di lati, altri al posto di facce. Questi due tipi di vertici dispari sono chiamati rispettivamente vertici di lato (*edge*) e di faccia (*face*), mentre i vertici pari sono detti punti di vertice (*vertex*).

Sovente si specifica una regola di suddivisione fornendo una *maschera*, ovvero un insieme di coefficienti che definiscono lo schema. L'azione della maschera si può visualizzare con una figura che mostra i punti di controllo utilizzati per calcolare i nuovi vertici (indicati con punti neri). I numeri adiacenti ai vertici sono i coefficienti della regola di suddivisione; un esempio è riportato nella Figura 2.6.

Infine, molti schemi di suddivisione possono gestire sagome affilate (*sharp features*), come creste e cuspidi. Tali forme possono essere caratterizzate a seconda del numero e del tipo di lati e vertici che si incontrano. I lati possono essere classificati in lati interni smussati, dove la superficie limite è almeno  $C^1$ , pieghe, in cui la superficie limite è  $C^0$ , e lati di confine, dove la superficie finisce. Per la classificazione dei vertici, si indichi con  $s$  il numero di pieghe che si incontrano nel vertice in considerazione. Se  $s = 0$  e la superficie limite in prossimità del vertice è per lo meno  $C^1$ , il vertice è detto smussato. Mentre, se la superficie è  $C^0$ , si distingue il caso in cui non esiste il vettore tangente nel vertice, quest'ultimo è allora una cuspide, da quello in cui esiste la direzione tangente, dove si parla di vertice di tipo conico. Un vertice è, invece, chiamato dardo se  $s = 1$ , piega se  $s = 2$ , spigolo se  $s \geq 3$ . Nella

Figura 3.2 sono riportati alcuni esempi di sharp features.

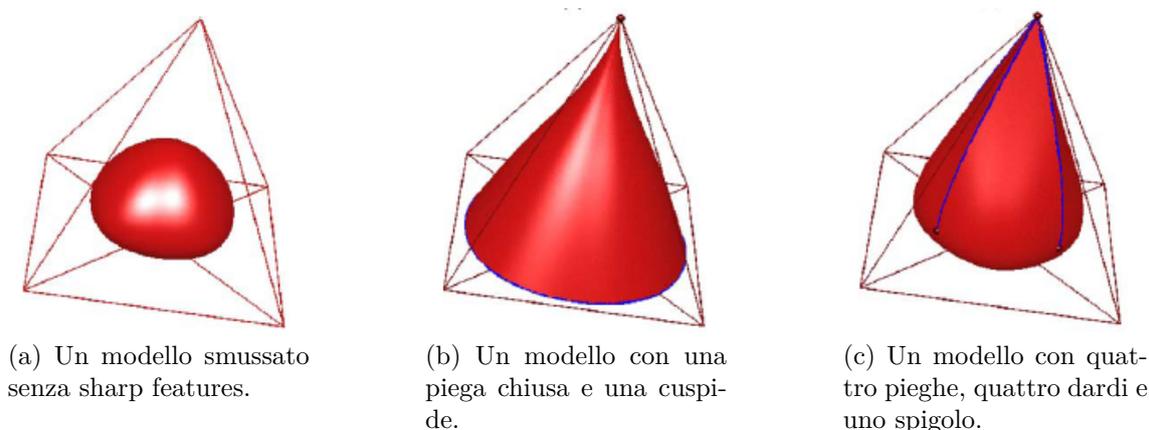


Figura 3.2: Superfici di suddivisione con sharp features.

## 3.2 Superfici di suddivisione uniformi

### 3.2.1 Schema di Doo-Sabin

Una generalizzazione della versione prodotto tensoriale della suddivisione di Chaikin (descritta nella sezione 2.1.5) è conosciuta col nome di superfici di suddivisione Doo-Sabin [7], largamente discussa in letteratura. A partire da una mesh di controllo rettangolare, la suddivisione di Doo-Sabin produce al limite *superfici B-spline bi-quadratiche uniformi*. Per ogni mesh di controllo arbitraria, la superficie limite ottenuta è globalmente continua di classe  $C^1$ .

Le regole topologiche delle superfici Doo-Sabin sono mostrate nella Figura 3.3. Per ogni faccia con  $n$  vertici, vengono inseriti  $n$  nuovi vertici calcolati mediante le regole geometriche descritte più avanti. La nuova mesh raffinata è costruita collegando in modo appropriato i nuovi vertici, creando così le facce di tipo F (faccia), E (lato) e V (vertice). In particolare, per ogni vecchia faccia, viene costruita una nuova faccia F collegando i nuovi vertici inseriti corrispondenti ai vertici della faccia originale. Mentre, per ogni vecchio lato, viene aggiunta una nuova faccia E unendo i nuovi quattro vertici adiacenti al lato stesso. Infine, per ogni vecchio vertice, viene costruita una nuova faccia V collegando tutti i nuovi vertici ottenuti dal vertice originale.

La suddivisione di Doo-Sabin è piuttosto semplice: non c'è distinzione tra vertici dispari e pari, ed una singola regola di suddivisione è sufficiente per definire lo schema. La regola geometrica è ancora una combinazione lineare dei vecchi vertici

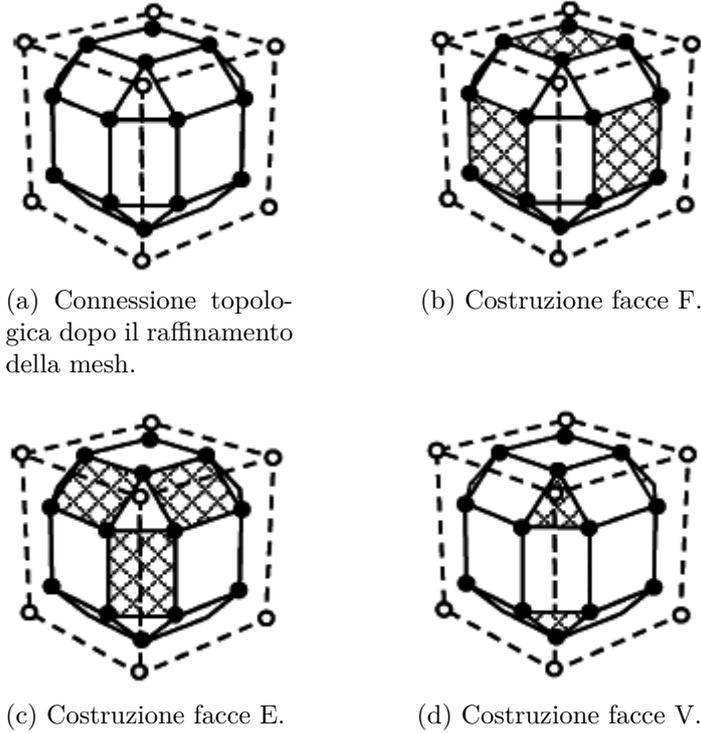


Figura 3.3: Regole topologiche per la suddivisione di Doo-Sabin.

del vicinato; la Figura 3.4 mostra le maschere per il caso regolare (facce con 4 lati) e irregolare (facce con  $n \neq 4$ ). Per il caso di facce regolari con quattro lati, un nuovo vertice può essere calcolato come

$$\mathbf{p}_0^{j+1} = \frac{9}{16}\mathbf{p}_0^j + \frac{3}{16}\mathbf{p}_1^j + \frac{1}{16}\mathbf{p}_2^j + \frac{3}{16}\mathbf{p}_3^j. \quad (3.1)$$

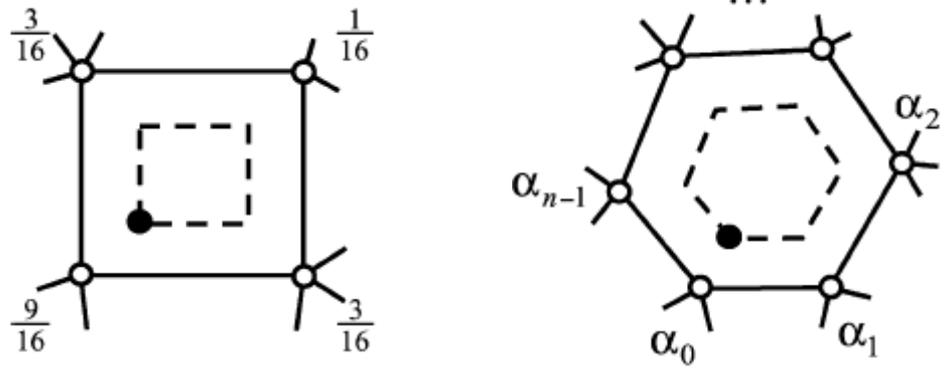
Mentre, per il caso irregolare, Doo e Sabin propongono

$$\mathbf{p}_0^{j+1} = \sum_{i=0}^{n-1} \alpha_i \mathbf{p}_i^j, \quad (3.2)$$

dove i coefficienti  $\alpha_i$  possono essere calcolati così

$$\alpha_i = \begin{cases} \frac{n+5}{4n} & i = 0 \\ \frac{3+2\cos(2\pi i/n)}{4n} & i = 1, 2, \dots, n-1 \end{cases}.$$

E' possibile ricavare l'espressione (3.1) a partire dalla forma parametrica delle superfici B-spline bi-quadratiche uniformi prodotto tensoriale. In particolare, l'equazione (2.1), vale a dire l'equazione di una curva B-spline quadratica uniforme, può essere facilmente estesa al caso di superfici bi-quadratiche prodotto tensoriale



(a) Maschera per facce regolari con quattro lati.

(b) Maschera per facce irregolari con  $n$  lati.

Figura 3.4: Maschere per la suddivisione di Doo-Sabin.

con parametrizzazione in  $(u, v)$  nel modo seguente (si veda la Figura 3.5)

$$S(u, v) = \mathbf{u} M P^0 M^T \mathbf{v}^T \quad u, v \in [0, 1] \quad (3.3)$$

dove  $\mathbf{u} = [u^2 \quad u \quad 1]$ ,  $M = \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}$ ,  $\mathbf{v} = [v^2 \quad v \quad 1]$  e

$$P^0 = \begin{bmatrix} \mathbf{p}_{11}^0 & \mathbf{p}_{12}^0 & \mathbf{p}_{13}^0 \\ \mathbf{p}_{21}^0 & \mathbf{p}_{22}^0 & \mathbf{p}_{23}^0 \\ \mathbf{p}_{31}^0 & \mathbf{p}_{32}^0 & \mathbf{p}_{33}^0 \end{bmatrix}.$$

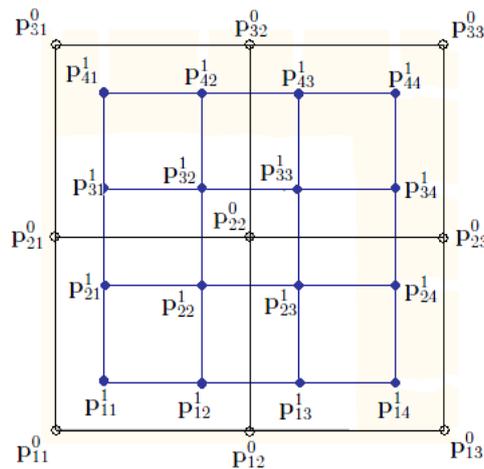


Figura 3.5: Pezza standard di una B-spline bi-quadratica.

Si consideri ora la prima sotto-patch corrispondente a  $0 < u, v < \frac{1}{2}$  (per ragioni di simmetria non è necessario considerare le altre pezze) che corrisponde a  $S(u_1, v_1)$ ,

dove  $u_1 = \frac{u}{2}$  e  $v_1 = \frac{v}{2}$ . Sostituendo queste due espressioni nella (3.3) si ottiene:

$$S(u_1, v_1) = \mathbf{u} H M P^0 M^T H^T \mathbf{v}^T \quad (3.4)$$

$$\text{con } H = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

La patch trovata dev'essere ancora una B-spline bi-quadratica definita mediante la nuova matrice dei punti di controllo  $P^1$ , dunque deve soddisfare la relazione:

$$S(u, v) = \mathbf{u} M P^1 M^T \mathbf{v}^T.$$

Richiedendo che quest'ultima espressione sia uguale alla (3.4), è necessario imporre

$$M P^1 M^T = H M P^0 M^T H^T$$

da cui si ricava, per la non singolarità di  $M$ , che

$$\begin{aligned} P^1 &= M^{-1} H M P^0 M^T H^{-T} M^{-T} \\ &= S P^0 S^T, \end{aligned}$$

dove  $S = M^{-1} H M$  è la *matrice di suddivisione*.

Svolgendo tutti i conti è immediato verificare che

$$S = \frac{1}{4} \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

.

In questo modo si è mostrata la relazione che lega i nuovi punti di controllo con

quelli originali, relazione che in forma esplicita risulta

$$\begin{aligned}
 P^1 &= \begin{bmatrix} \mathbf{p}_{11}^1 & \mathbf{p}_{12}^1 & \mathbf{p}_{13}^1 \\ \mathbf{p}_{21}^1 & \mathbf{p}_{22}^1 & \mathbf{p}_{23}^1 \\ \mathbf{p}_{31}^1 & \mathbf{p}_{32}^1 & \mathbf{p}_{33}^1 \end{bmatrix} \\
 &= S P^0 S^T \\
 &= \frac{1}{16} \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{11}^0 & \mathbf{p}_{12}^0 & \mathbf{p}_{13}^0 \\ \mathbf{p}_{21}^0 & \mathbf{p}_{22}^0 & \mathbf{p}_{23}^0 \\ \mathbf{p}_{31}^0 & \mathbf{p}_{32}^0 & \mathbf{p}_{33}^0 \end{bmatrix} \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 3 \\ 0 & 0 & 1 \end{bmatrix} \\
 &= \frac{1}{16} \begin{bmatrix} 9\mathbf{p}_{11}^0 + 3\mathbf{p}_{21}^0 + 3\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0 & \cdots & \cdots \\ 3\mathbf{p}_{11}^0 + 9\mathbf{p}_{21}^0 + \mathbf{p}_{12}^0 + 3\mathbf{p}_{22}^0 & \cdots & \cdots \\ 9\mathbf{p}_{21}^0 + 3\mathbf{p}_{31}^0 + 3\mathbf{p}_{22}^0 + \mathbf{p}_{32}^0 & \cdots & \cdots \end{bmatrix}.
 \end{aligned}$$

Pertanto, come già anticipato, un nuovo vertice può essere calcolato come combinazione lineare dei vecchi vertici. Ad esempio,  $\mathbf{p}_{11}^1$  è definito come

$$\mathbf{p}_{11}^1 = \frac{9}{16}\mathbf{p}_{11}^0 + \frac{3}{16}\mathbf{p}_{21}^0 + \frac{3}{16}\mathbf{p}_{12}^0 + \frac{1}{16}\mathbf{p}_{22}^0. \quad (3.5)$$

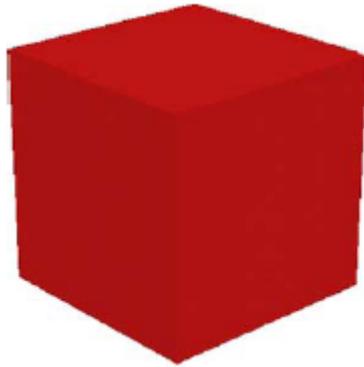
L'espressione (3.5) è la regola geometrica dello schema di suddivisione di Doo-Sabin per il caso di facce regolari con quattro lati, e coincide con l'equazione (3.1).

Due esempi di superficie di suddivisione di Doo-Sabin sono mostrati nelle Figure 3.6 e 3.7. Si fa notare che dopo il primo passo di suddivisione, il numero di facce irregolari rimane costante e vengono sempre più isolate l'una dall'altra.

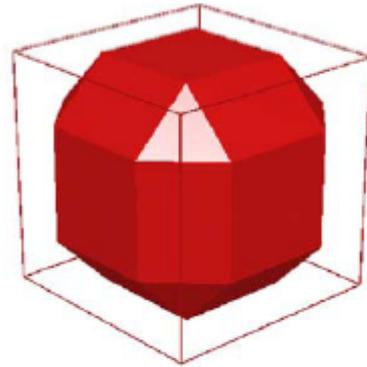
### 3.2.2 Schema di Catmull-Clark

Si è visto nella sezione precedente che le superfici prodotte dallo schema Doo-Sabin sono generalizzazioni di superfici B-spline bi-quadratiche uniformi. Le superfici ottenute con lo schema Catmull-Clark [4] sono, invece, generalizzazioni di *superfici B-spline bi-cubiche uniformi*, globalmente continue di classe  $C^2$ .

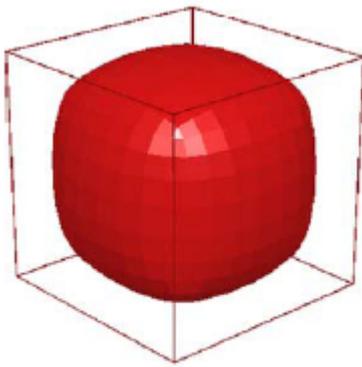
La base del metodo Catmull-Clark nasce dal considerare una pezza B-spline bi-cubica standard. La forma di tale patch è data da 16 punti di controllo, come mostrato nella Figura 3.8. L'espressione parametrica di una pezza B-spline bi-cubica uniforme deriva direttamente da quella di una B-spline cubica, presentata



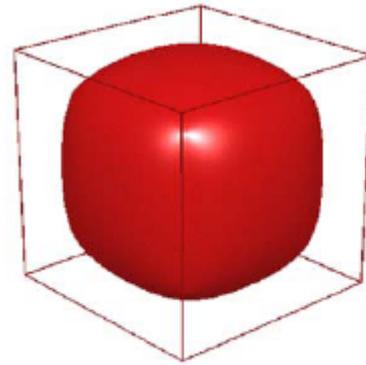
(a) Mesh di controllo iniziale.



(b) Mesh di controllo dopo un livello di raffinamento.



(c) Mesh di controllo dopo tre livelli di raffinamento.



(d) Superficie limite.

Figura 3.6: Illustrazione di una superficie di suddivisione di Doo-Sabin, tratta da [13].

nell'equazione (2.2), e può essere sintetizzata come

$$S(u, v) = \mathbf{u} M P^0 M^T \mathbf{v}^T \quad u, v \in [0, 1] \quad (3.6)$$

$$\text{dove } \mathbf{u} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}, M = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \text{ e}$$

$$P^0 = \begin{bmatrix} \mathbf{p}_{11}^0 & \mathbf{p}_{12}^0 & \mathbf{p}_{13}^0 & \mathbf{p}_{14}^0 \\ \mathbf{p}_{21}^0 & \mathbf{p}_{22}^0 & \mathbf{p}_{23}^0 & \mathbf{p}_{24}^0 \\ \mathbf{p}_{31}^0 & \mathbf{p}_{32}^0 & \mathbf{p}_{33}^0 & \mathbf{p}_{34}^0 \\ \mathbf{p}_{41}^0 & \mathbf{p}_{42}^0 & \mathbf{p}_{43}^0 & \mathbf{p}_{44}^0 \end{bmatrix}.$$

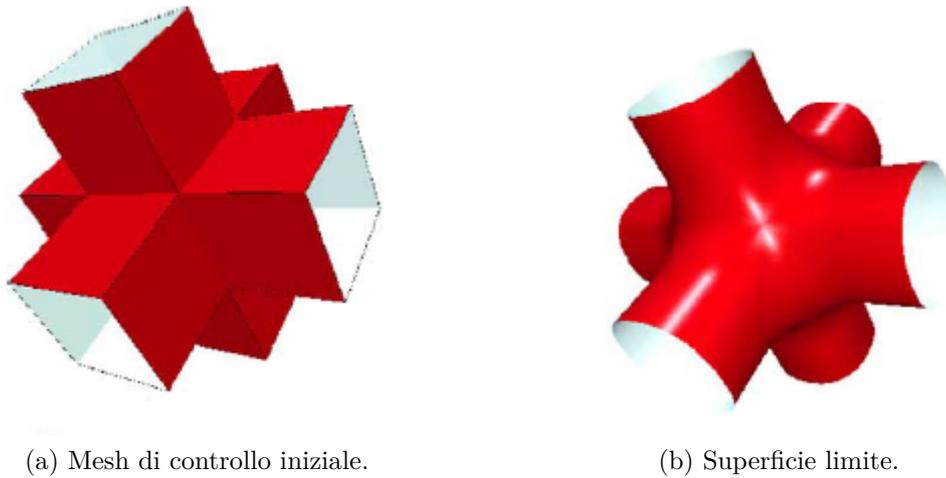


Figura 3.7: Illustrazione di un modello di tubatura ottenuto con superfici Doo-Sabin, presa da [13].

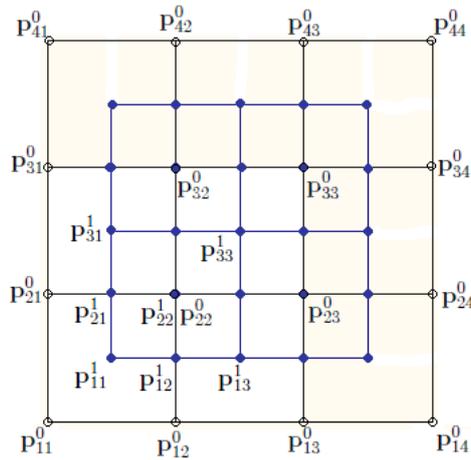


Figura 3.8: Pezza standard di una B-spline bi-cubica.

Come fatto per le superfici Doo-Sabin, si consideri ora la prima sotto-pezza corrispondente ai parametri  $0 < u, v < \frac{1}{2}$ ; non serve considerare le altre sotto-pezze per la simmetria delle funzioni base B-spline. Si denoti con  $S(u_1, v_1)$  la sotto-patch in questione, dove  $u_1 = \frac{u}{2}$  e  $v_1 = \frac{v}{2}$ . Sostituendo queste due espressioni nella (3.6) si ottiene

$$S(u_1, v_1) = \mathbf{u} H M P^0 M^T H^T \mathbf{v}^T \quad (3.7)$$

$$\text{dove } H = \begin{bmatrix} \frac{1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Ovviamente la pezza deve rimanere una B-spline bi-cubica con la propria mesh di controllo  $P^1$  che soddisfa la relazione

$$S(u, v) = \mathbf{u} M P^1 M^T \mathbf{v}^T.$$

Richiedendo che quest'ultima espressione sia uguale alla (3.7), bisogna imporre

$$M P^1 M^T = H M P^0 M^T H^T.$$

Dato che la matrice base  $M$  è invertibile, si ricava

$$\begin{aligned} P^1 &= M^{-1} H M P^0 M^T H^{-T} M^{-T} \\ &= S P^0 S^T, \end{aligned} \tag{3.8}$$

dove  $S = M^{-1} H M$  è la *matrice di suddivisione*. Eseguendo tutte le moltiplicazioni tra matrici si ottiene

$$S = \frac{1}{8} \begin{bmatrix} 4 & 4 & 0 & 0 \\ 1 & 6 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 6 & 1 \end{bmatrix}.$$

Facendo riferimento alla Figura 3.8, si nota che alcuni punti della mesh raffinata si trovano nel centro dei rettangoli della mesh originale; questi nuovi vertici sono chiamati *punti di faccia*. Allo stesso modo, alcuni nuovi punti giacciono sui lati che connettono due punti originali, questi sono chiamati *punti di lato*. Infine, altri punti si trovano in prossimità dei vecchi punti di controllo e sono chiamati *punti di vertice*. Esplicitando l'equazione (3.8), è facile verificare che ogni nuovo punto di controllo di un dato tipo (faccia, lato o vertice) è calcolato a partire dai vertici dell'intorno mediante la stessa espressione algebrica.

Un punto di faccia viene calcolato come la media dei vertici originali della faccia in questione; ad esempio il nuovo punto di controllo  $\mathbf{p}_{11}^1$  è ottenuto come

$$\mathbf{p}_{11}^1 = \frac{\mathbf{p}_{11}^0 + \mathbf{p}_{21}^0 + \mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{4}. \tag{3.9}$$

Analogamente, un punto di lato è esprimibile come la media tra il punto centrale del lato originale e la media dei nuovi punti di faccia corrispondenti alle facce che

condivido il lato. Ad esempio, il nuovo punto di lato  $\mathbf{p}_{12}^1$  è calcolato come

$$\begin{aligned}
\mathbf{p}_{12}^1 &= \frac{\mathbf{p}_{11}^0 + 6\mathbf{p}_{12}^0 + \mathbf{p}_{13}^0 + \mathbf{p}_{21}^0 + 6\mathbf{p}_{22}^0 + \mathbf{p}_{23}^0}{16} & (3.10) \\
&= \frac{\mathbf{p}_{11}^0 + \mathbf{p}_{12}^0 + \mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{16} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0 + \mathbf{p}_{13}^0 + \mathbf{p}_{33}^0}{16} + \frac{4\mathbf{p}_{12}^0 + 4\mathbf{p}_{22}^0}{16} \\
&= \frac{\mathbf{p}_{11}^1}{4} + \frac{\mathbf{p}_{13}^1}{4} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2} \\
&= \frac{\frac{\mathbf{p}_{11}^1 + \mathbf{p}_{13}^1}{2} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2}}{2}.
\end{aligned}$$

L'espressione di un nuovo punto di vertice è leggermente più articolata. Si indichi con  $Q$  la media dei punti di faccia di tutte le facce adiacenti al vertice originale, con  $R$  la media dei punti medi di tutti i lati incidenti nel vecchio vertice e con  $S$  il punto di controllo stesso. Un nuovo punto di vertice è dunque definito come

$$\frac{Q}{4} + \frac{R}{2} + \frac{S}{4}. \quad (3.11)$$

Seguono i passaggi per ricavare l'espressione (3.11) per il punto di vertice  $\mathbf{p}_{22}^1$

$$\begin{aligned}
\mathbf{p}_{22}^1 &= \frac{\mathbf{p}_{11}^0 + 6\mathbf{p}_{21}^0 + \mathbf{p}_{31}^0 + 6\mathbf{p}_{12}^0 + 36\mathbf{p}_{22}^0 + 6\mathbf{p}_{32}^0 + \mathbf{p}_{13}^0 + 6\mathbf{p}_{23}^0 + \mathbf{p}_{33}^0}{64} & (3.12) \\
&= \frac{1}{4} \frac{\mathbf{p}_{11}^1 + \mathbf{p}_{13}^1 + \mathbf{p}_{31}^1 + \mathbf{p}_{33}^1}{4} + \frac{1}{2} \frac{\frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2} + \frac{\mathbf{p}_{12}^0 + \mathbf{p}_{22}^0}{2}}{4} + \frac{\mathbf{p}_{22}^0}{4} \\
&= \frac{Q}{4} + \frac{R}{2} + \frac{S}{4}.
\end{aligned}$$

Le maschere di suddivisione corrispondenti alle equazioni (3.9), (3.10) e (3.12) sono riportate nella Figura 3.9.

Le equazioni viste finora possono essere facilmente estese per gestire mesh a topologia arbitraria. Nella Figura 3.10 sono illustrate le maschere da utilizzare nel caso generale. Con  $n_f$  è stato indicato il numero di lati di una faccia, con  $n$  la valenza di un vertice,  $\beta = \gamma = \frac{1}{n}$  e  $\alpha = 1 - \beta - \gamma$ . I coefficienti  $\beta$  e  $\gamma$ , in corrispondenza di vertici straordinari, possono essere selezionati da un range di valori molto vario e possono essere ottimizzati per ottenere forme ben definite in prossimità di tali punti straordinari. Per informazioni più dettagliate si veda [26].

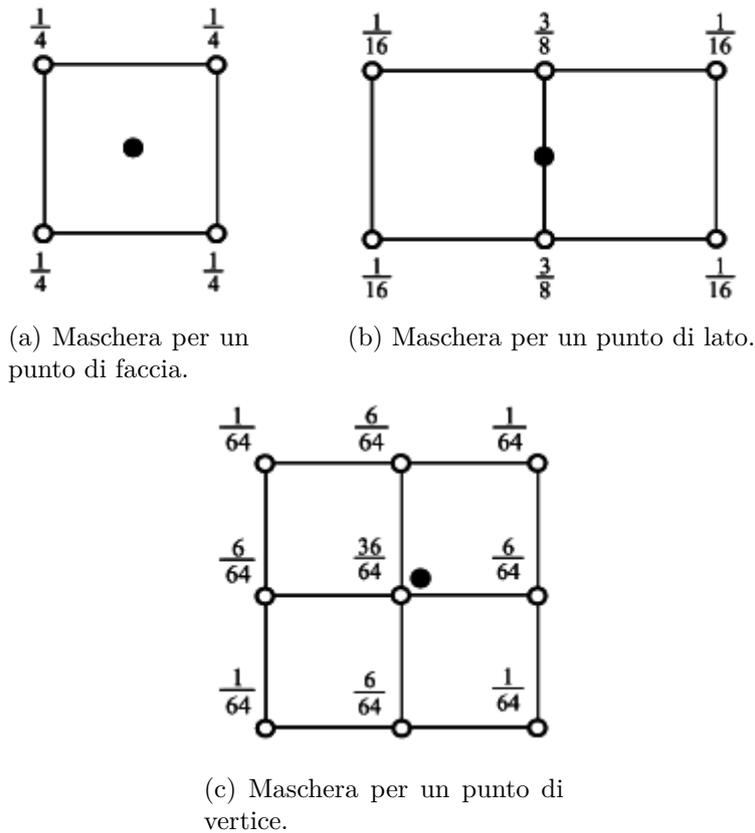


Figura 3.9: Maschere per la suddivisione Catmull-Clark di mesh quadrilatere.

In aggiunta, sono state definite diverse regole di suddivisione per forme affilate come pieghe, angoli e dardi. Alcune di queste regole sono descritte nelle maschere mostrate nella Figura 3.11.

Le regole topologiche per la suddivisione di Catmull-Clark sono riportate nella Figura 3.12 che mostra come viene costruita esattamente la mesh di controllo raffinata. Si può osservare che dopo un passo di suddivisione tutte le facce hanno quattro lati e il numero di vertici straordinari rimane costante.

Per concludere, le Figure 3.13 e 3.14 riportano due esempi di superfici di suddivisione di Catmull-Clark.

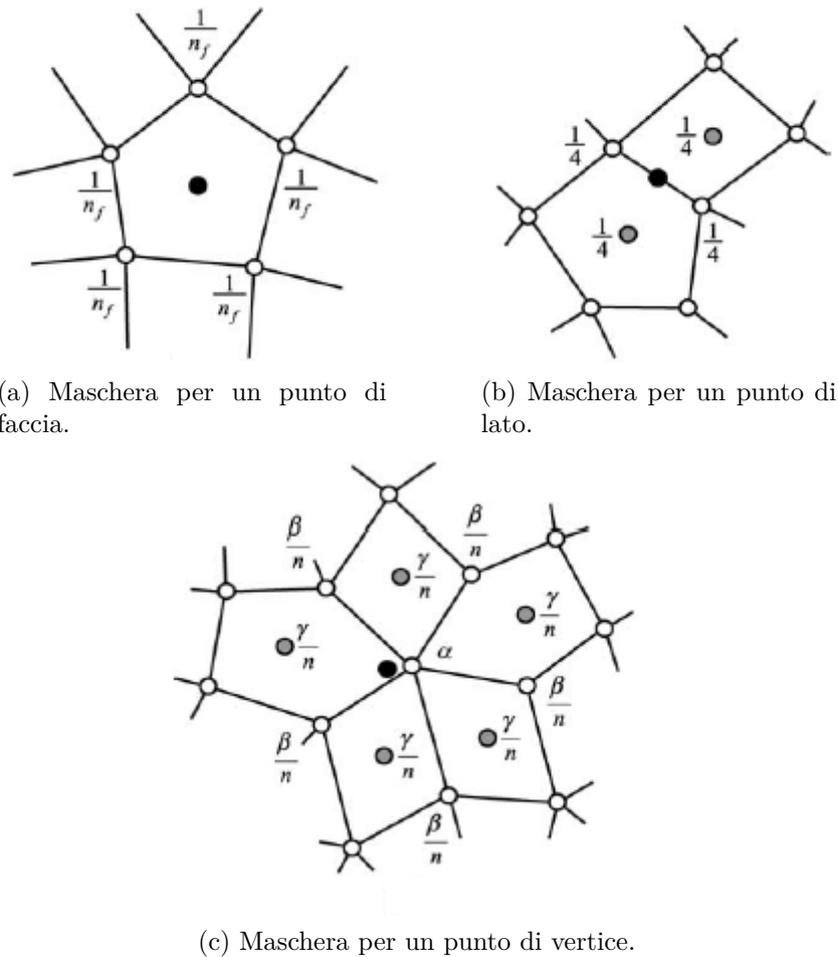


Figura 3.10: Maschere per la suddivisione Catmull-Clark di mesh arbitrarie.

### 3.3 Superfici di suddivisione non uniformi

#### 3.3.1 Vettore dei nodi uniformi e non uniformi

In una B-spline uniforme tutti i nodi devono essere equispaziati, mentre in una B-spline non-uniforme possono essere posizionati arbitrariamente. La Figura 3.15 mostra un esempio di funzioni base di B-spline uniforme e uno di funzioni base non uniforme. Come visto in precedenza, tutti i vettore dei nodi con spaziatura uniforme si ottengono come traslazione e scalatura della stessa funzione base, mentre nel caso non uniforme le funzioni base possono differire tra loro. La Figura 3.16 mostra due esempi di curve B-spline ottenibili con le basi definite nella Figura 3.15.

Vettori dei nodi non-uniformi sono utili per diverse applicazioni pratiche. Per esempio, nella Figura 3.16, un vettore dei nodi non-uniforme è stato utilizzato per interpolare i punti di controllo della curva ai suoi estremi. Questo setup è molto comune e permette ad un progettista di posizionare accuratamente il bordo del-

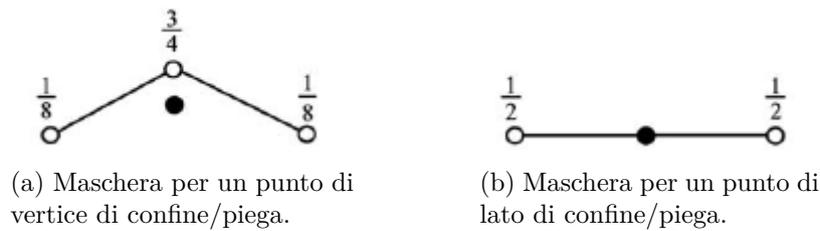


Figura 3.11: Maschere di sharp features per la suddivisione di Catmull-Clark.

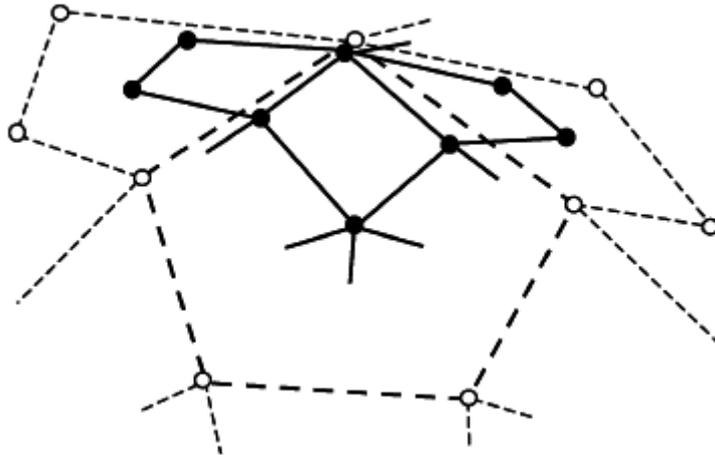


Figura 3.12: Regole topologiche per la suddivisione di Catmull-Clark.

la B-spline. Tuttavia, per ottenere tale risultato, il vettore dei nodi necessita di nodi multipli (cioè sovrapposti), il che è possibile solo con un vettore dei nodi non-uniforme. Nodi multipli possono essere utilizzati anche all'interno della curva (o della superficie), riducendo così la continuità in un certo nodo (ricavando degli spigoli). In più, vettori dei nodi non uniformi permettono di allocare uno spazio parametrico maggiore ad una sezione più lunga della curva. Questo può essere importante quando si adatta la B-spline a dei valori misurati che non sono equispaziati. Inoltre, la rappresentazione di forme ondulate è molto più efficiente e gestibile se i punti di controllo sono più densi dove la variazione di curvatura è maggiore. In questo caso, si ottengono risultati di elevata qualità utilizzando vettori dei nodi non uniformi che rispecchiano la spaziatura dei punti di controllo.

### 3.3.2 Raffinamento di B-spline non uniformi

Nelle curve B-spline non uniformi quadratiche ogni vertice del poligono di controllo corrisponde ad un singolo tratto della curva quadratica. Risulta, quindi, conveniente esprimere il vettore dei nodi scrivendo l'intervallo nodale  $d_i$ , di ogni segmento di curva, vicino al corrispondente punto di controllo  $\mathbf{p}_i^j$ , come illustrato nella Figura

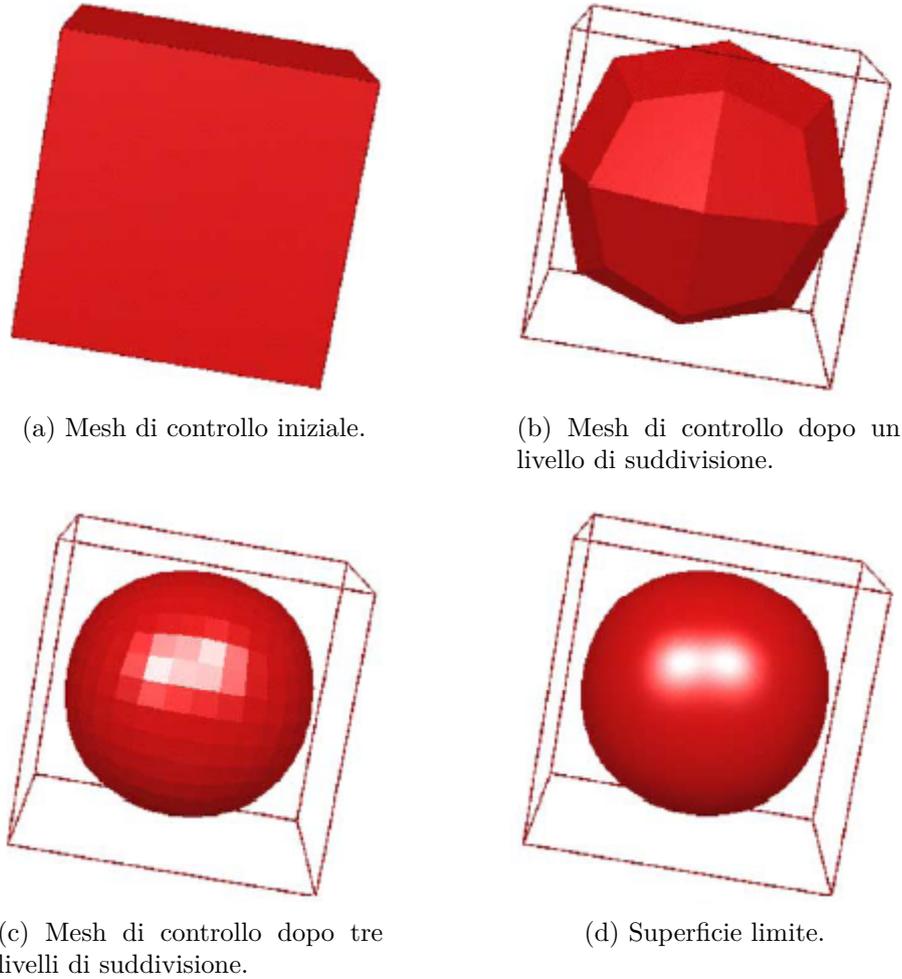


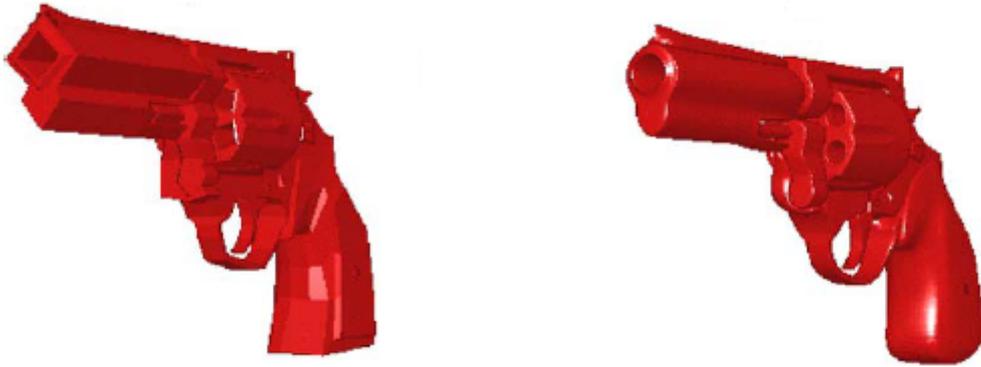
Figura 3.13: Illustrazione di una superficie di suddivisione di Catmull-Clark, tratta da [13].

3.17. Inserendo un nuovo nodo nel mezzo di ogni intervallo, si ottiene un poligono di controllo con un numero di punti raddoppiato, tali punti presentano coordinate  $\mathbf{p}_i^{j+1}$  calcolabili come:

$$\mathbf{p}_{2i}^{j+1} = \frac{(d_i + 2d_{i+1})\mathbf{p}_i + d_i\mathbf{p}_{i+1}}{2(d_i + d_{i+1})},$$

$$\mathbf{p}_{2i+1}^{j+1} = \frac{d_{i+1}\mathbf{p}_i + (2d_i + d_{i+1})\mathbf{p}_{i+1}}{2(d_i + d_{i+1})}.$$

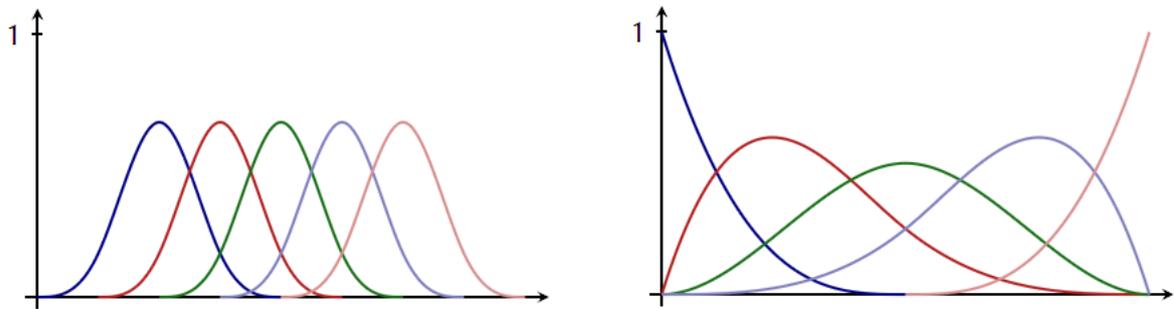
Nel raffinamento di curve cubiche non uniformi, invece, ogni lato del poligono di controllo corrisponde ad un singolo segmento cubico della curva, dunque è possibile scrivere l'intervallo nodale vicino ad ogni lato del poligono di controllo, si veda la Figura 3.18. Le equazioni dei nuovi punti di controllo  $\mathbf{p}_i^{j+1}$ , generati con l'inserimento di un nodo nel mezzo dei ogni intervallo nodale, sono le seguenti:



(a) Mesh di controllo iniziale.

(b) Superficie limite.

Figura 3.14: Illustrazione del modello di una pistola ottenuta con superfici Catmull-Clark, presa da [13].



(a) Caso uniforme. Ogni funzione base è una copia traslata di tutte le altre.

(b) Caso non uniforme. In questo esempio tutte le funzioni base sono uniche.

Figura 3.15: Funzioni base di una B-spline cubica con due differenti vettori dei nodi.

$$\mathbf{p}_{2i+1}^{j+1} = \frac{(d_i + 2d_{i+1})\mathbf{p}_i^j + (d_i + 2d_{i-1})\mathbf{p}_{i+1}^j}{2(d_{i-1} + d_i + d_{i+1})},$$

$$\mathbf{p}_{2i}^{j+1} = \frac{d_i\mathbf{p}_{2i-1}^{j+1} + (d_{i-1} + d_i)\mathbf{p}_i^j + d_{i-1}\mathbf{p}_{2i+1}^{j+1}}{2(d_{i-1} + d_i)}.$$

E' possibile ora estendere facilmente le formule di raffinamento di B-spline non uniformi anche alle superfici. Quest'ultime sono definite mediante una rete di controllo che topologicamente è una griglia rettangolare, per cui tutti i vettori di nodi orizzontali sono l'uno la scalatura dell'altro. Lo stesso vale per i vettori dei nodi verticali.

Per il caso di superfici quadratiche si veda la Figura 3.19, dove viene posto in

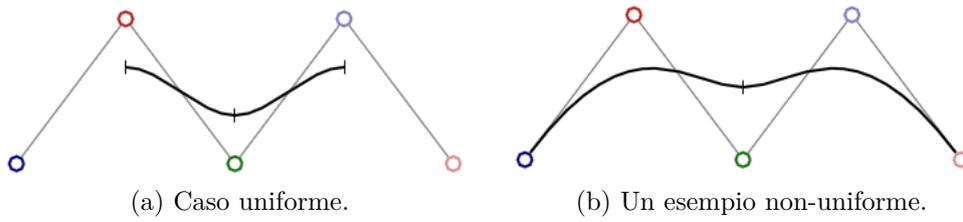


Figura 3.16: B-spline cubiche con gli stessi punti di controllo, ma differenti vettori dei nodi.

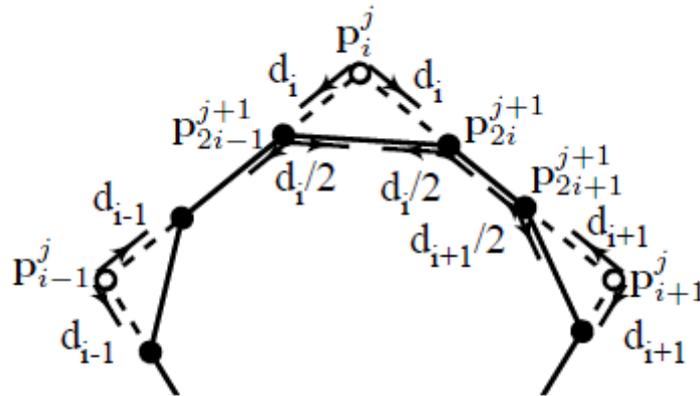


Figura 3.17: Suddivisione di una curva B-spline quadratica non uniforme.

evidenza il nuovo punto di controllo  $\mathbf{F}_A$  calcolabile come

$$\mathbf{F}_A = \frac{\mathbf{V} + \mathbf{A}}{2} + \frac{ac(\mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D})}{4(ad + ac + bc + bd)},$$

dove

$$\mathbf{V} = \frac{bd\mathbf{A} + ad\mathbf{B} + bc\mathbf{C} + ac\mathbf{D}}{ad + ac + bc + bd}.$$

La formula per il nuovo punto di controllo  $\mathbf{F}_A$  è scritta nella forma di Doo-Sabin, ciò è significativo perché in questo modo i nuovi punti di controllo sono visti come se fossero in gruppo, creando una nuova faccia a partire da ciascuna faccia del livello di suddivisione precedente. In più, i vertici delle nuove facce sono in corrispondenza 1 a 1 con quelli vecchi, mentre con la forma prodotto tensoriale ciò non emerge.

Si analizzano ora le regole di raffinamento di superfici B-spline cubiche non uniformi, a tale scopo si veda la Figura 3.20. Innanzitutto, ogni faccia è sostituita con nuovo vertice  $\mathbf{F}_i$ . Per esempio

$$\mathbf{F}_1 = [(e_3 + 2e_4)(d_2 + 2d_1)\mathbf{p}_0 + (e_3 + 2e_4)(d_2 + 2d_3)\mathbf{p}_1 + (e_3 + 2e_2)(d_2 + 2d_3)\mathbf{p}_5 + (e_3 + 2e_2)(d_2 + 2d_1)\mathbf{p}_2] / [4(e_2 + e_3 + e_4)(d_1 + d_2 + d_3)].$$

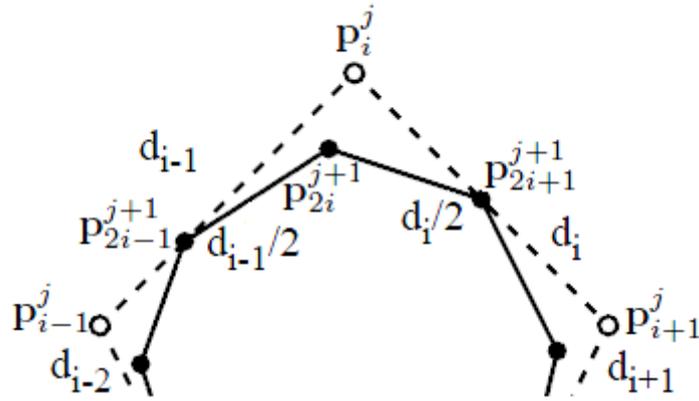


Figura 3.18: Suddivisione di una curva B-spline cubica non uniforme.

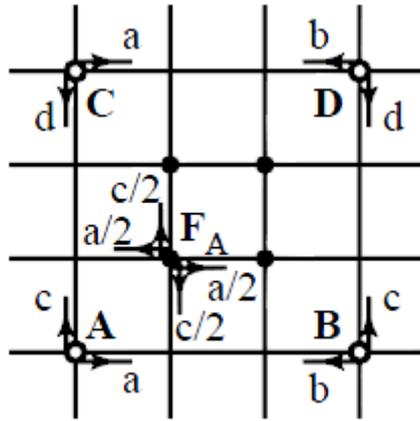


Figura 3.19: Suddivisione di una superficie B-spline quadratica non uniforme.

Ogni lato, invece, è spezzato con un vertice di edge  $\mathbf{E}_i$

$$\mathbf{E}_1 = \frac{e_2 \mathbf{F}_1 + e_3 \mathbf{F}_4 + (e_2 + e_3) \mathbf{M}_1}{2(e_2 + e_3)},$$

dove

$$\mathbf{M}_1 = \frac{(2d_1 + d_2) \mathbf{p}_0 + (d_2 + 2d_3) \mathbf{p}_1}{2(d_1 + d_2 + d_3)}.$$

Infine, ogni punto di controllo originale è sostituito dal punto di vertice  $\mathbf{V}$

$$\begin{aligned} \mathbf{V} = & \frac{\mathbf{p}_0}{4} + \frac{d_3 e_2 \mathbf{F}_1 + d_2 e_2 \mathbf{F}_2 + d_2 e_3 \mathbf{F}_3 + d_3 e_3 \mathbf{F}_4}{4(d_2 + d_3)(e_2 + e_3)} + \\ & + \frac{d_3(e_2 + e_3) \mathbf{M}_1 + e_2(d_2 + d_3) \mathbf{M}_2 + d_2(e_2 + e_3) \mathbf{M}_3 + e_3(d_2 + d_3) \mathbf{M}_4}{4(d_2 + d_3)(e_2 + e_3)}. \end{aligned}$$

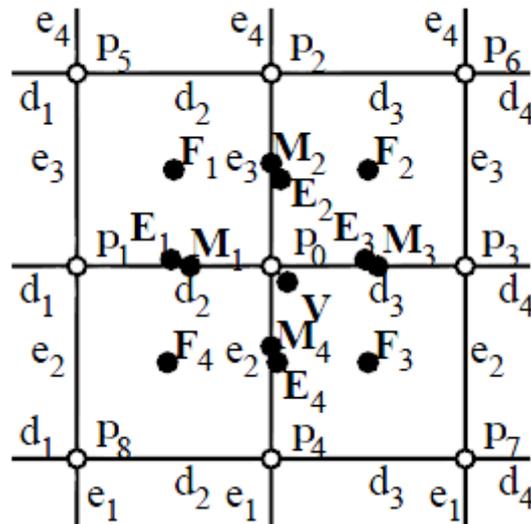


Figura 3.20: Suddivisione di una superficie B-spline cubica non uniforme.

### 3.3.3 Raffinamento delle NURSS

In questa sezione si presentano le regole di suddivisione per le NURSS quadratiche e cubiche. Oltre alle regole per il calcolo nei nuovi punti di controllo, sarà necessario introdurre le regole per determinare i nuovi intervalli nodali. In una superficie Doo-Sabin non uniforme, ad ogni vertice è assegnato un intervallo nodale per ogni lato adiacente ad esso. Mentre per superfici Catmull-Clark non uniformi ad ogni lato del poligono di controllo è assegnato un solo intervallo nodale.

Si guardi la Figura 3.22,  $d_{ij}^0$  è la spaziatura nodale che si riferisce al lato con vertici  $\mathbf{p}_i$  e  $\mathbf{p}_j$ . Ruotando in senso antiorario attorno a  $\mathbf{p}_i$ ,  $d_{ij}^1$  denota la spaziatura nodale il primo lato incontrato,  $d_{ij}^2$  per il secondo, etc. Nelle superfici cubiche ogni lato ha una singola spaziatura nodale, ne segue che  $d_{ij}^0 = d_{ji}^0$ . Per non creare confusione con la notazione utilizzata, in questa sezione gli apici sono da intendersi come rotazioni attorno ad un vertice, si indicherà con  $\mathbf{p}_i$  un vertice originale e con  $\bar{\mathbf{p}}_i$  il corrispondente vertice al livello di suddivisione successivo.

Nel caso quadratico, il raffinamento procede in maniera identica alla suddivisione di Doo-Sabin. A partire da un poliedro, se ne calcola uno nuovo in cui sono create nuove facce per ogni faccia, lato e vertice della mesh iniziale. Durante un passo di suddivisione, ogni faccia viene sostituita da una nuova faccia collegata attraverso i vecchi lati e i vecchi vertici ad altre nuove facce. In questo schema di raffinamento, i punti straordinari sono punti situati al centro di facce con  $n$  lati, dove  $n \neq 4$ . Dopo un'iterazione, ogni vertice del nuovo poliedro ha valenza 4 e il numero di facce non quadrilatero resta costante.

Guardando la Figura 3.21 per le etichette, il nuovo vertice  $\bar{\mathbf{p}}_i$  di una NURSS

quadratica può essere calcolato come:

$$\bar{\mathbf{p}}_i = \frac{\mathbf{V} + \mathbf{p}_i}{2} + (d_{i+1,i+2}^0 d_{i+3,i+2}^0 + d_{i-1,i-2}^0 d_{i-3,i-2}^0) \times \frac{-n\mathbf{p}_i + \sum_{j=1}^n (1 + 2 \cos(\frac{2\pi|i-j|}{n})) \mathbf{p}_j}{8 \sum_{k=1}^n d_{k-1,k}^0 d_{k+1,k}^0}, \quad (3.13)$$

dove

$$\mathbf{V} = \frac{\sum_{k=1}^n d_{k-1,k}^0 d_{k+1,k}^0 \mathbf{p}_k}{\sum_{k=1}^n d_{k-1,k}^0 d_{k+1,k}^0}. \quad (3.14)$$

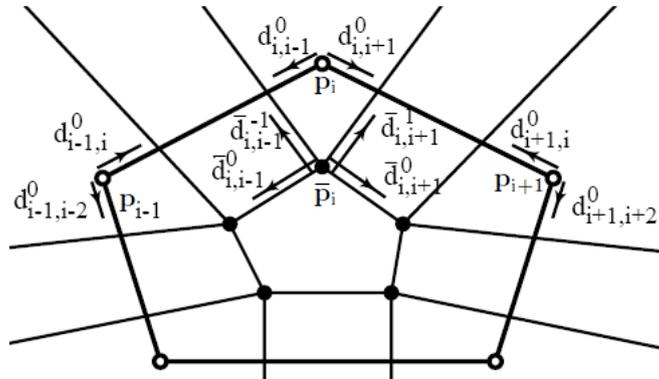


Figura 3.21: Regole di raffinamento per NURSS quadratiche.

Le nuove spaziature nodali  $\bar{d}_{ij}^k$  possono essere specificate in vari modi. Sederberg in [27] propone i seguenti:

$$\begin{aligned} \bar{d}_{i,i+1}^0 &= \bar{d}_{i,i-1}^{-1} = d_{i,i+1}^0/2 \\ \bar{d}_{i,i-1}^0 &= \bar{d}_{i,i+1}^1 = d_{i,i-1}^0/2 \end{aligned} \quad (3.15)$$

oppure

$$\begin{aligned} \bar{d}_{i,i+1}^0 &= d_{i,i+1}^0/2 \\ \bar{d}_{i,i-1}^{-1} &= (d_{i,i+1}^0 + d_{i,i-1}^{-1})/4 \\ \bar{d}_{i,i-1}^0 &= d_{i,i-1}^0/2 \\ \bar{d}_{i,i+1}^1 &= (d_{i,i-1}^0 + d_{i,i+1}^1)/4 \end{aligned} \quad (3.16)$$

Le prime espressioni permettono di ottenere una matrice di suddivisione costante dopo poche iterazioni. Le seconde sembrano produrre forme più soddisfacenti.

Si presenta ora il caso cubico facendo riferimento alla Figura 3.22. Il punto di faccia, per una faccia con  $n$  lati, viene calcolato come

$$\mathbf{F} = \frac{\sum_{i=0}^{n-1} w_i \mathbf{P}_i}{\sum_{i=0}^{n-1} w_i} \quad (3.17)$$

dove

$$\begin{aligned} w_i &= (d_{i+1,i}^0 + d_{i+1,i}^2 + d_{i+1,i}^{-2} + d_{i-2,i-1}^0 + d_{i-2,i-1}^2 + d_{i-2,i-1}^{-2}) \\ &\times (d_{i-1,i}^0 + d_{i-1,i}^2 + d_{i-1,i}^{-2} + d_{i+2,i+1}^0 + d_{i+2,i+1}^2 + d_{i+2,i+1}^{-2}) \end{aligned} \quad (3.18)$$

L'espressione di un vertice di lato, invece, è la seguente (si veda la Figura 3.23):

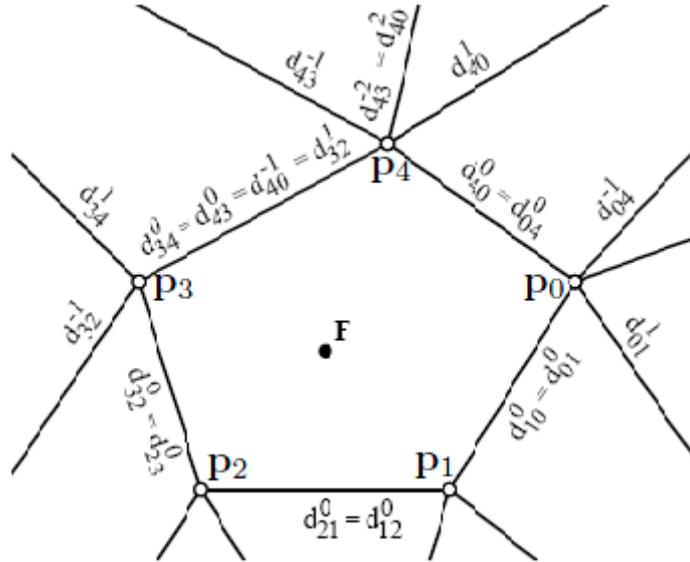


Figura 3.22: Punto di faccia di NURSS cubiche.

$$\mathbf{E} = (1 - \alpha_{ij} - \alpha_{ji})\mathbf{M} + \alpha_{ij}\mathbf{F}_{ij} + \alpha_{ji}\mathbf{F}_{ji}, \quad (3.19)$$

in cui

$$\alpha_{ij} = \frac{d_{ji}^1 + d_{ij}^{-1}}{2(d_{ji}^1 + d_{ij}^{-1} + d_{ji}^{-1} + d_{ij}^1)} \quad (3.20)$$

e quindi

$$\mathbf{M} = \frac{(d_{ji}^0 + d_{ji}^2 + d_{ji}^{-2})\mathbf{P}_i + (d_{ij}^0 + d_{ij}^2 + d_{ij}^{-2})\mathbf{P}_j}{d_{ji}^0 + d_{ji}^2 + d_{ji}^{-2} + d_{ij}^0 + d_{ij}^2 + d_{ij}^{-2}}. \quad (3.21)$$

La (3.20) è valida solo se  $d_{ji}^1 + d_{ij}^{-1} + d_{ji}^{-1} + d_{ij}^1 \neq 0$ , altrimenti  $\alpha_{ij} = \alpha_{ji} = 0$ . Inoltre, i punti  $\mathbf{F}_{ij}$ , calcolati utilizzando l'equazione (3.17), sono i vertici di faccia

corrispondenti alle facce adiacenti al lato.

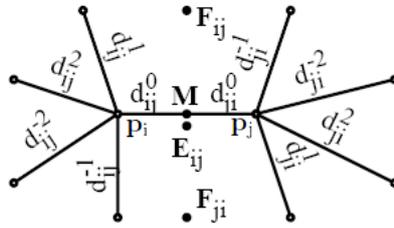


Figura 3.23: Punto di lato di NURSS cubiche.

Il punto di vertice, per un punto con valenza  $n$ , è dato dalla seguente (si veda la Figura 3.24):

$$\mathbf{V} = c\mathbf{p}_0 + \frac{3 \sum_{i=1}^n (m_i \mathbf{M}_i + f_{i,i+1} \mathbf{F}_{i,i+1})}{n \sum_{i=1}^n (m_i + f_{i,i+1})}, \quad (3.22)$$

dove gli  $\mathbf{M}_i$  sono definiti come nell'espressione (3.21), gli  $\mathbf{F}_{i,i+1}$  come nella (3.17), e

$$\begin{aligned} m_i &= (d_{0i}^1 + d_{0i}^{-1})(d_{0i}^2 + d_{0i}^{-2})/2, \\ f_{ij} &= d_{0i}^1 d_{0j}^{-1}, \\ c &= \frac{n-3}{n}. \end{aligned}$$

Tuttavia, se  $\sum_{i=1}^n (m_i + f_{i,i+1}) = 0$  si assume  $c = 1$ .

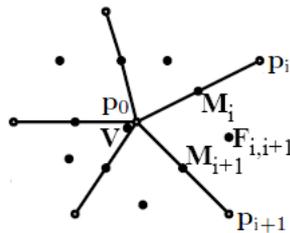


Figura 3.24: Punto di vertice di NURSS cubiche.

Infine, la Figura 3.25 mostra come calcolare la nuova spaziatura nodale.

### 3.3.4 Discussione

La Figura 3.26 mostra l'effetto che la spaziatura nodale può causare in una superficie. Nella griglia sulla sinistra, a tutti i lati è stata assegnata una spaziatura pari a 1, ad eccezione dei quattro lati etichettati con 0. Le immagini di destra mostrano due passi di suddivisione con Catmull-Clark non uniforme.

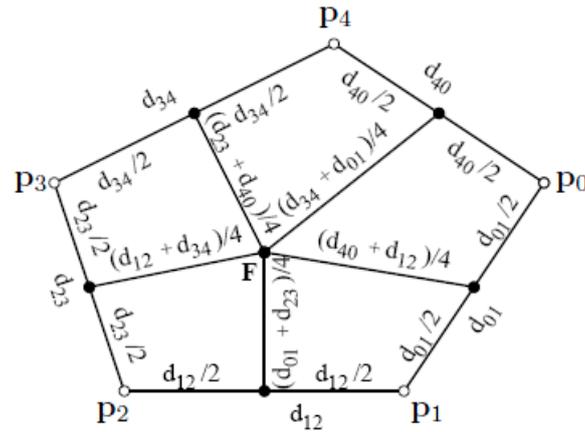


Figura 3.25: Spaziatura nodale di NURSS cubiche.

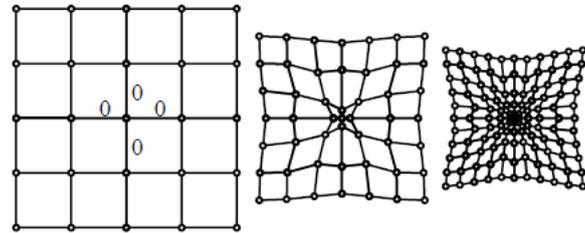


Figura 3.26: Effetto di una spaziatura non uniforme nelle NURSS.

In generale, le NURSS permettono di rappresentare cuspidi, creste e faccette piane senza la necessità di maschere particolari. Per esempio le Figure 3.27 e 3.29 mostrano una varietà di forme che possono essere ottenute con le NURSS, ma non sono realizzabili utilizzando superfici uniformi. I poliedri iniziali sono rappresentati in wireframe. Diverse forme spigolose possono essere realizzate impostando a zero la spaziatura nodale di lati appropriati. Un altro uso della spaziatura nodale è la modifica della forma di una superficie. La Figura 3.28 mostra l'effetto di differenti spaziature nodali per una superficie Doo-Sabin non uniforme.

In sintesi, le NURSS estendono i metodi a topologia generale permettendo intervalli dei nodi non equispaziati. Ciò conferisce ad una singola descrizione di superficie l'espressività sia delle superfici prodotte tensorialmente non uniformi sia delle superfici di suddivisione uniformi. Inoltre, questo schema permette una grande libertà nel modificare la forma delle superfici. In particolare, possono essere modellate figure affilate impostando a zero appropriate spaziature nodali.

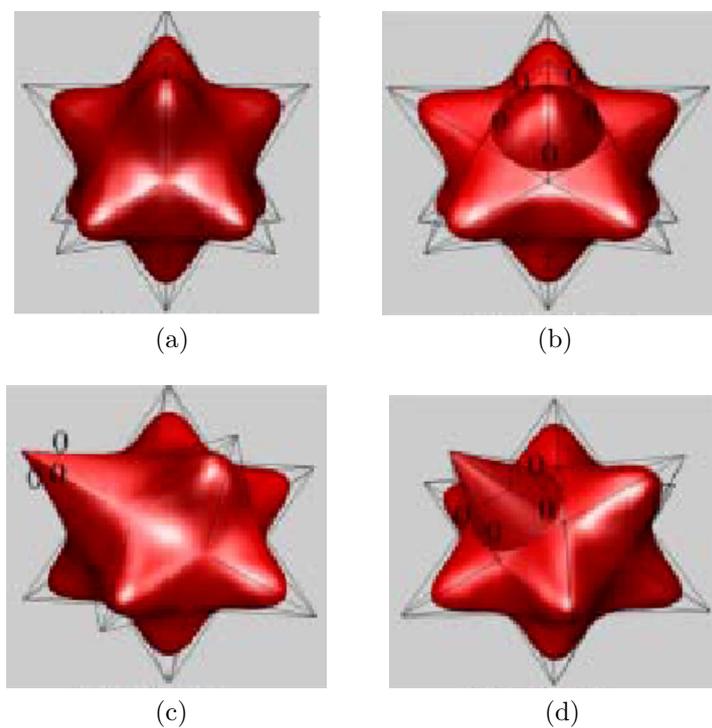


Figura 3.27: Illustrazione di diverse forme ricavate con superfici Doo-Sabin non uniformi, tratta da [27].

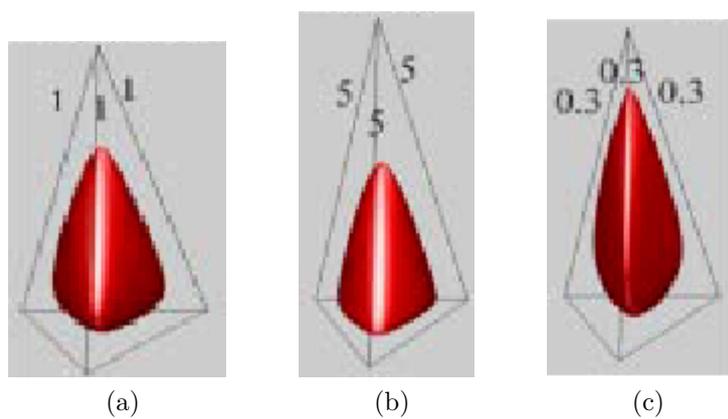
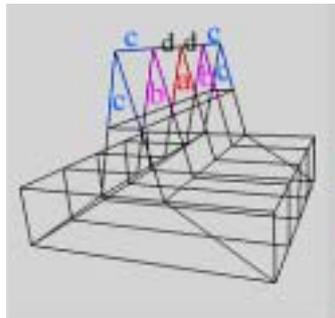


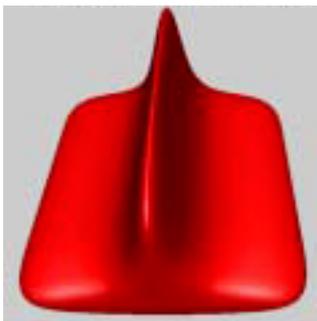
Figura 3.28: Superficie Doo-Sabin non uniforme con diverse spazature nodali, tratta da [27].



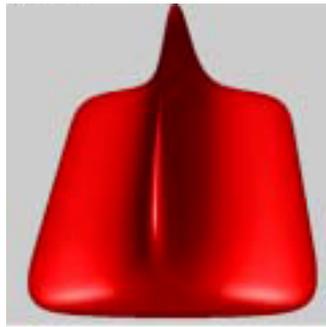
(a) Mesh di controllo iniziale con le etichette delle spaziature nodali



(b) Superficie Catmull-Clark uniforme



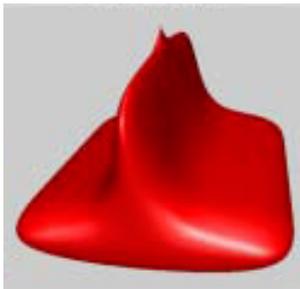
(c)  $a = b = 0.2, c = d = 1$



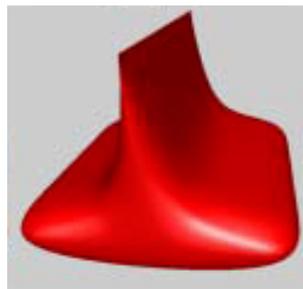
(d)  $a = b = 0, c = d = 1$



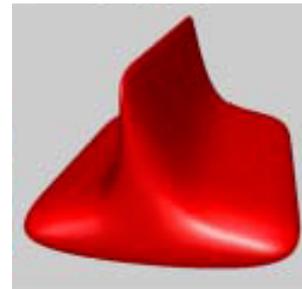
(e)  $a = b = 20, c = d = 1$



(f)  $a = d = 0, b = c = 1$



(g)  $a = b = c = 0, d = 1$



(h)  $a = b = c = 0.2, d = 1$

Figura 3.29: Superficie Catmull-Clark non uniforme con diverse spaziature nodali, presa da [27].

# Capitolo 4

## Implementazione

L'implementazione degli algoritmi di suddivisione presentati nel Capitolo 3 è articolata in due fasi. Nella prima fase è stato sviluppato un prototipo per ciascun schema di suddivisione analizzato, vale a dire Doo-Sabin, Catmull-Clark, NURSS quadratiche e NURSS cubiche. Lo sviluppo di tali prototipi ha permesso di evidenziare le principali problematiche di ciascun algoritmo di suddivisione, specialmente per gli schemi non uniformi. La seconda fase è stata, invece, focalizzata sulla realizzazione di algoritmi robusti, efficienti e facilmente integrabili in sistemi CAD (Computer-Aided Design) di uso comune.

La Sezione 4.1 presenta le principali problematiche implementative emerse durante lo sviluppo dei prototipi; in particolare sono analizzate le strutture dati utilizzate e le soluzioni adottate nei casi non trattati esplicitamente in letteratura. La Sezione 4.2, oltre a presentare brevemente i sistemi CAD, articola gli aspetti di maggior interesse per l'implementazione delle superfici di suddivisione in tali software.

### 4.1 Sviluppo dei prototipi

Lo sviluppo dei prototipi è stato realizzato in Matlab. Quest'ultimo è un linguaggio di calcolo tecnico di alto livello e un ambiente interattivo per lo sviluppo di algoritmi, visualizzazione e analisi di dati, e computazioni numeriche. Matlab supporta operazioni su vettori e matrici che sono fondamentali in ambito grafico. Permette una veloce implementazione rispetto ai tradizionali linguaggi di programmazione perché nasconde la gestione di compiti di basso livello, come definire variabili e allocare la memoria. Allo stesso tempo, Matlab permette il controllo di flusso, la definizione di strutture dati, il debugging e l'esecuzione di istruzioni senza la necessità di compilare e linkare il codice. In più, Matlab fornisce funzioni per la visualizzazione 3-D di mesh e superfici oltre l'importazione ed esportazione nei comuni formati grafici.

### 4.1.1 Strutture dati

Una mesh poligonale consiste di un insieme di vertici, lati, facce e delle relazione topologiche tra di essi. Basandosi su queste relazioni, una struttura dati definisce come dev'essere memorizzato ogni elemento e quali riferimenti ai suoi vicini sono necessari. Si è scelto di rappresentare esplicitamente la topologia di una mesh per capire quali relazioni sono più stringenti per l'implementazione di algoritmi di suddivisione. Pertanto una mesh è costituita dal vettore dei vertici, da quello delle facce e da quello dei lati.

Facendo riferimento all'esempio riportato nella Figura 4.1, il vettore dei vertici è  $[ \mathbf{p}_1 \ \cdots \ \mathbf{p}_8 ]$ , quello delle facce  $[ F_1 \ \cdots \ F_6 ]$  e quello dei lati  $[ e_1 \ \cdots \ e_{12} ]$ . Il pedice di ogni elemento corrisponde all'indice nel vettore corrispondente. Si è scelto di adottare una *struttura ad indici* perché garantisce tempi di accesso di ordine  $\mathcal{O}(1)$ , a fronte di una ridotta ridondanza dei dati allocati in memoria [33].

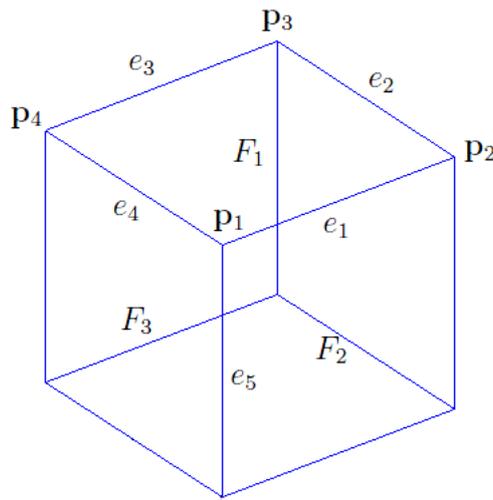


Figura 4.1: Esempio di una mesh nel prototipo in Matlab.

Per ogni vertice, oltre a memorizzare le coordinate cartesiane, sono allocati esplicitamente il vettore delle facce a cui appartiene il punto e quello dei lati cui appartiene. Dovendo gestire anche i vertici straordinari, i vettori delle facce e dei lati hanno dimensione variabile. Inoltre, le facce e i lati sono ordinati in senso anti-orario rispetto al vertice in considerazione. Dunque, facendo riferimento al punto  $\mathbf{p}_1$  della Figura 4.1, il vettore delle facce è  $[ 1 \ 3 \ 2 ]$ , mentre quello dei lati è  $[ 1 \ 4 \ 5 ]$ . L'ordinamento dei vettori è un aspetto critico per l'implementazione dell'algoritmo di suddivisione delle NURSS cubiche poiché, nelle equazioni di raffinamento (3.22),

(3.23) e (3.24), le spaziate nodali da utilizzare dipendono dai lati e dalle facce attraversati ruotando in senso orario o anti-orario attorno a un dato vertice. L'ordinamento delle facce risulta molto comodo anche per gli algoritmi duali, vale a dire Doo-Sabin e NURSS quadratiche, dove è necessario creare delle facce di vertice collegando i nuovi punti inseriti corrispondenti a un dato vertice originale.

Una faccia, invece, è descritta semplicemente dal vettore ordinato dei vertici che la compongono. Segue che la faccia  $F_1$  è definita dal vettore dei vertici  $\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$ .

Un lato è definito dai suoi due punti estremi; quindi il vettore dei vertici del lato  $e_1$  è  $\begin{bmatrix} 1 & 2 \end{bmatrix}$ . In più, per gestire gli algoritmi di suddivisione non uniformi, è necessario memorizzare le spaziate nodali. Le NURSS quadratiche necessitano di due spaziate per ciascun lato, mentre per le cubiche ne è sufficiente solo una, si vedano le Figure 3.21 e 3.22. Al fine di utilizzare una struttura dati comune a tutti gli algoritmi di suddivisione, si è scelto di associare due spaziate nodali per ciascun lato come richiesto dalle NURSS quadratiche. D'altra parte, tali spaziate sono sempre pari a 1 per le superfici uniformi, e, nel caso di NURSS cubiche, le due spaziate di ciascun lato sono sempre uguali.

Dallo sviluppo dei prototipi è emerso che l'implementazione efficiente degli schemi di suddivisione analizzati, specialmente per le NURSS, è fortemente legato alla scelta di una struttura dati che permetta di attraversare ordinatamente gli elementi del primo-anello<sup>1</sup> di ciascun vertice in un tempo minimo e possibilmente costante. Inoltre, dev'essere ottimizzata la gestione di vertici e facce a valenza variabile.

### 4.1.2 Soluzioni adottate nei casi limite

Sederberg, nel suo articolo sulle NURSS [27], presenta le regole topologiche e geometriche di tali schemi di suddivisione non uniformi; tuttavia non esplicita le espressioni da adottare nei casi limite dovuti a spaziate nodali nulle. Tali casi sono stati analizzati in questo lavoro di tesi e sono discussi nei prossimi paragrafi.

Si considerino le equazioni (3.13) e (3.14) per il raffinamento di una NURSS quadratica e la corrispondente Figura (3.21). Il denominatore di entrambe le espressioni è dato dalla sommatoria dei prodotti delle spaziate nodali "entranti" in ciascun vertice della faccia. D'altra parte, per definire spigoli vivi e sagome affilate, alcune spaziate nodali possono essere nulle e di conseguenza anche il denominatore in questione può azzerarsi, rendendo le regole di raffinamento non applicabili. In realtà, considerando le espressioni (3.15) e (3.16) per l'aggiornamento delle spaziate

---

<sup>1</sup>Il primo-anello di un vertice è l'insieme di tutti i vertici direttamente collegati al punto in questione, oltre ai lati e alle facce incidenti in esso.

nodali, risulta che anche se una sola spaziatura è nulla, dopo uno o due passi di suddivisione, le spaziature nodali di una faccia possano essere tali da annullare il denominatore delle equazioni (3.13) e (3.14). Tuttavia, se almeno una spaziatura nodale di ciascun vertice della faccia è nulla, si ricade nel caso di superfici di suddivisione uniformi e quindi possono essere applicate le regole di raffinamento definite da Doo e Sabin in [7] e riportate nelle espressioni (3.1) e (3.2).

L'equazione (3.17) per il calcolo di un punto di faccia di una NURSS cubica è una media pesata dei vertici della faccia stessa; i pesi sono dati dall'espressione (3.18). Analizzando quest'ultima si deduce che la somma dei pesi si annulla solo quando le spaziature nodali dei lati della faccia sono tutte nulle. In questa condizione, l'equazione (3.19) non è applicabile, però il vettore dei nodi corrispondente è uniforme e dunque il punto di faccia può essere definito come in una comune superficie Catmull-Clark, vale a dire calcolando la media algebrica dei vertici della faccia.

Un punto di lato nelle NURSS cubiche è definito dalla (3.19), cioè è dato da una combinazione lineare del punto medio  $\mathbf{M}$  e dei punti di faccia delle facce adiacenti al lato stesso. Il punto  $\mathbf{M}$  è calcolato attraverso la media pesata degli estremi del lato in considerazione, come riportato nell'equazione (3.21). Anche in questo caso le spaziature nodali in gioco possono essere nulle. Tuttavia, il denominatore della (3.21) è nullo solo quando tutte le spaziature nodali considerate sono nulle e quindi il vettore dei nodi è uniforme. In questa condizione, il punto  $\mathbf{M}$  può essere calcolato come media algebrica dei vertici del lato.

## 4.2 Implementazione per sistemi CAD

### 4.2.1 I sistemi CAD

I sistemi CAD (*Computer Aided Design*) rappresentano il settore dell'informatica volto all'utilizzo di tecnologie software, e in particolare della computer grafica, per supportare l'attività di progettazione di manufatti. I settori d'impiego dei software CAD sono i più disparati. Si spazia dall'architettura all'urbanistica, dall'elettrotecnica e meccanica all'industrial design, dall'impiantistica all'elettronica.

I sistemi CAD hanno come obiettivo la creazione di modelli, per lo più 3-D, del prodotto da realizzare e possono essere classificati, sulla base dell'estensione del campo di utilizzo, in CAD orizzontali e verticali. I primi sono software aventi dei domini molto ampi, utilizzabili con successo in contesti applicativi differenti, ad esempio nella progettazione architettonica e in quella meccanica. I comandi offerti da questi sistemi sono indipendenti da uno specifico ambiente applicativo. Si avranno,

pertanto, comandi come “traccia-linea” senza alcuna nozione se la linea rappresenta una parete di un edificio o lo spigolo di un supporto metallico. Dall'altra parte, i CAD verticali sono prodotti con dominio ristretto, orientati ad un particolare contesto applicativo, con comandi e funzionalità specifici per quel contesto. Ad esempio, un sistema CAD verticale per la progettazione di interni offrirà comandi per creare e posizionare differenti tipi di pareti e collocarvi porte e finestre.

Tutte le entità geometriche disegnate sono memorizzate e gestite da un apposito *database geometrico*, che rappresenta il cuore dei sistemi per la progettazione assistita al calcolatore. I prodotti CAD offrono comandi per il disegno di elementi grafici elementari, quali linee e archi, e comandi più complessi che consentono al disegnatore di realizzare con rapidità elementi grafici più elaborati come superfici 3-D. Questi comandi sono usualmente potenziati dall'abbinamento con modalità operative basate su sistemi di riferimento alternativi e dalla riferibilità di punti notevoli. Uno dei più evidenti vantaggi nell'utilizzo di un sistema CAD rispetto all'impiego di tecniche tradizionali, consiste nella grande facilità e rapidità con cui è possibile modificare, anche in modo radicale, un disegno per correggerlo o per creare una nuova versione. I sistemi CAD non si limitano alla sola automatizzazione delle attività tradizionali del disegno, ma offrono anche funzionalità di strutturazione possibili solo con l'ausilio di strumenti informatici. Il disegno, pertanto, cessa di essere un insieme uniforme di entità grafiche, per divenire una struttura, anche complessa, di aggregazioni di entità arricchite di attributi grafici e del contesto applicativo, come ad esempio materiali, note di lavorazione, costi, ecc. Queste funzionalità sono proposte all'utente del sistema come funzionalità supplementari: egli è responsabile di deciderne il migliore utilizzo in funzione delle proprie esigenze e delle modalità di lavoro dell'ambiente professionale in cui opera. Inoltre, l'utente di un sistema CAD può velocizzare in modo significativo il proprio lavoro creando degli speciali archivi, detti librerie, in cui raccogliere i disegni o i particolari di utilizzo più frequenti. Il disegno creato con un sistema CAD deve essere utilizzabile non solo come rappresentazione grafica, ma anche come sorgente di informazione sul progetto. È, dunque, importante che sia consentito l'accesso a tutta l'informazione contenuta nel disegno, sia essa in forma esplicita o implicita. I sistemi di disegno tipicamente supportano interrogazioni sulla geometria e sulla stima dei costi e della complessità.

L'attuale dimensione degli schermi per computer non è in alcun modo paragonabile alla dimensione di un tecnigrafo oppure di un foglio di formato A0; pertanto i sistemi CAD sono costretti ad offrire modalità alternative per la visualizzazione dei disegni. Le funzionalità essenziali di visualizzazione sono analoghe a quelle di una telecamera: operando sull'obiettivo si può ingrandire o rimpicciolire a piacere

il disegno passando da una visione globale dell'intero progetto ad una visione locale di una sua sottoparte. Le principali funzionalità per il controllo della visibilità sono lo zoom, il pan e le viste multiple.

Oltre a ciò, pressoché tutti i sistemi CAD possono essere personalizzati ed estesi al fine di migliorare la produttività dei progettisti e la qualità dei progetti. Le principali modalità per estendere un sistema CAD sono le macro, cioè comandi ottenuti componendo comandi più semplici tramite un linguaggio di programmazione. Per esempio, in un sistema CAD 2-D per fornire la funzione di disegno di muri, una macro può chiedere all'utente di inserire il punto iniziale, il punto finale e lo spessore del muro, e inserire automaticamente nel modello due linee parallele che rappresentano il muro.

Inoltre, la possibilità di scambiare dati tra sistemi CAD diversi e tra sistemi CAD e sistemi CAM (Computer-Aided Manufacturing) costituisce un elemento fondamentale nella valutazione delle funzionalità di un sistema.

Numerosi formati neutrali di dati sono stati proposti nel corso degli anni, ma nessuno di essi si è imposto con una diffusione sufficiente sugli altri. Attualmente sono utilizzati alcuni formati definiti da standard ufficiali, come *iges*, ed altri che sono standard de facto come il formato *dxf*.

## 4.2.2 Scelte di progetto

In ambito CAD, il linguaggio di programmazione largamente dominante è il C/C++. Ciò è dovuto all'efficienza e alla flessibilità delle sue istruzioni, oltre alla gestione diretta della memoria. Per tali motivi, si è scelto di utilizzare tale linguaggio per l'implementazione degli algoritmi di suddivisione.

In letteratura si distinguono strutture dati *face-based* e *edge-based* internamente implementate come indici o liste. Le prime memorizzano per ogni faccia i relativi vertici e le facce adiacenti. Ciò permette di attraversare tutte le facce che circondano un vertice, operazione molto comune in diversi algoritmi. Tuttavia, è necessario gestire diversi casi speciali quando si incontrano elementi con una valenza variabile, con un'inevitabile perdita di efficienza. Strutture dati *face-based* sono tipicamente utilizzate per mesh triangolari dove il numero di vertici e di lati per ciascuna faccia è costante. Strutture dati *edge-based*, invece, memorizzano i vertici del lato e i lati adiacenti. Dato che un lato ha sempre la stessa struttura topologica, è possibile gestire poligoni con valenza arbitraria in una sola mesh. Si è scelto quindi una struttura *edge-based*.

Esistono diverse varianti di strutture *edge-based* che differiscono solo nelle informazioni topologiche che memorizzano. La struttura dati *winged-edge* [1] memorizza

per ogni lato i riferimenti ai suoi vertici, ad entrambe le facce che condividono il lato e ai quattro lati adiacenti. Attraversare il vicinato richiede una selezione per ogni passo poiché i lati non memorizzano l'orientamento in senso esplicito. La struttura dati *half-edge* [10] risolve questo problema dividendo ogni lato in due metà: ogni half-edge mantiene un riferimento al suo half-edge opposto, ad un vertice incidente e ad una faccia adiacente. Si è deciso di utilizzare una struttura dati half-edge perché ottimizza l'attraversamento degli elementi del primo-anello di un vertice; questa proprietà, come emerso dallo studio dei prototipi, è critica per l'efficienza degli algoritmi di suddivisione.

La Figura (4.2) mostra l'architettura di una struttura dati half-edge: ogni vertice referencia un half-edge uscente (1), mentre una faccia mantiene il riferimento ad un half-edge che la delimita (2). Un half-edge, invece, memorizza il vertice a cui punta (3), la faccia a cui appartiene (4), il prossimo half-edge interno alla faccia (5), l'half-edge opposto (6) e opzionalmente l'half-edge precedente nella faccia (7).

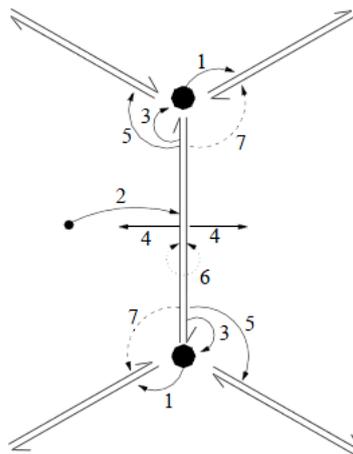
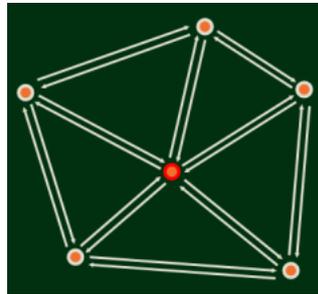


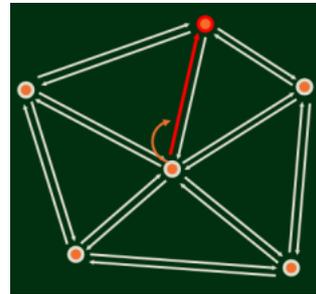
Figura 4.2: Struttura dati half-edge.

Gli algoritmi di suddivisione richiedono l'accesso frequente al vicinato di un vertice, talvolta in forma di vertici, altre in termini di lati e facce adiacenti. La struttura dati half-edge garantisce un accesso rapido e a tempo costante a tali informazioni. Nella Figura (4.3) è illustrato come attraversare tutti i vertici del vicinato di un dato vertice della mesh. Il primo vertice del vicinato è puntato dall'half-edge uscente dal vertice in questione; per identificare il prossimo vertice, basta passare all'half-edge opposto e quindi a quello successivo. Quest'ultimo half-edge punta direttamente al vertice cercato. Inoltre, iniziando dall'half-edge uscente da un vertice, il successivo

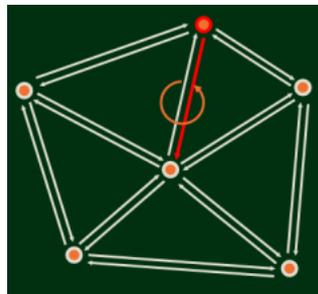
half-edge in senso orario può essere individuato con sole due referenziamenti: si passa all'half-edge opposto e quindi al prossimo half-edge della faccia. Le strutture dati face-based devono eseguire delle ricerche onerose per trovare l'half-edge in questione.



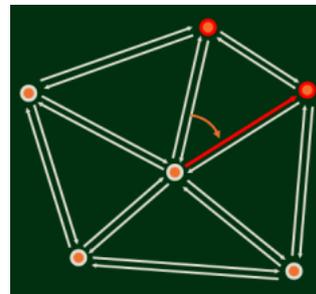
(a) Vertice di riferimento.



(b) Si identifica l'half-edge uscente dal vertice.



(c) Si passa all'half-edge opposto.



(d) L'half-edge successivo punta al prossimo vertice del vicinato.

Figura 4.3: Attraversamento del vicinato di un vertice mediante la struttura dati half-edge.

La libreria OpenMesh [12] è un'implementazione efficiente (*index-based*) e versatile della struttura dati half-edge pensata per rappresentare e manipolare mesh poligonali. Grazie al suo approccio basato sulla programmazione generica, oltre ad essere estremamente flessibile, è anche molto efficiente: evita l'overhead delle funzioni virtuali e risolve molte dipendenze a compile-time. Per quanto detto, si è scelto di utilizzare tale libreria per rappresentare le mesh poligonali.

### 4.2.3 Le classi

La natura del linguaggio C++, orientato agli oggetti, permette di garantire l'astrazione dei componenti. Questa proprietà nasce dalla separazione delle funzioni per la comunicazione esterna dalle operazioni interne. Ciò permette la modifica del

comportamento di un oggetto senza avere effetti sul modo in cui le diverse entità del sistema interagiscono. Di fatto l'astrazione è un requisito chiave per rendere il software facilmente manutenibile.

Nella Figura 4.4 è riportato il diagramma delle classi principali realizzate per implementare i diversi schemi di suddivisione uniformi e non uniformi presentati nel Capitolo 3. Tutte le classi qui riportate possono essere facilmente integrabili in un qualsiasi sistema CAD per gestire le superfici di suddivisione. Qualora fosse già presente una struttura dati per la rappresentazione di mesh poligonali o non si volesse utilizzare OpenMesh, l'architettura presentata è comunque valida, data l'astrazione concettuale su cui si fonda.

**CPolyMesh** E' la classe che rappresenta una mesh poligonale. Si basa sulle strutture dati e gli algoritmi forniti da OpenMesh. Attraverso degli iteratori è possibile recuperare tutti i vertici, le facce e i lati della mesh. Inoltre, sono disponibili dei *circulator* che permettono, tra l'altro, di attraversare i vertici e i lati di una faccia piuttosto che gli elementi del primo-anello di un vertice. Questa classe può essere facilmente integrata in un qualsiasi database geometrico per gestire le superfici di suddivisione oppure una generica polymesh.

**ISubdivider** Interfaccia che identifica un algoritmo di suddivisione. Data una polymesh iniziale, applica il numero di suddivisioni richiesto e ritorna la mesh raffinata. Questa interfaccia permette di separare la reale implementazione dell'algoritmo di suddivisione, sia esso uniforme o non uniforme, dal contesto di utilizzo. Grazie a questo disaccoppiamento è possibile introdurre nuovi algoritmi o modificare quelli già presenti in modo del tutto trasparente al database geometrico.

**CSubdividerFactory** Istanza la sotto-classe corrispondente all'algoritmo di suddivisione richiesto e ritorna solo un puntatore all'interfaccia **ISubdivider**, rendendo trasparente la tecnica di suddivisione utilizzata. Questo modus operandi corrisponde ad un noto design pattern chiamato *Factory Method*.

**CSubdivider** Classe base da cui derivano tutte le sotto-classi che implementano un algoritmo di suddivisione. Implementa l'interfaccia **ISubdivider** e tutte le operazioni comuni ai diversi algoritmi di suddivisione.

**CDooSabin** Classe che implementa l'algoritmo di suddivisione per superfici Doo-Sabin, deriva dalla classe **CSubdivider**. Precomputa i coefficienti definiti nell'equazione (3.2) per minimizzare i tempi di calcolo.

**CCatmullClark** Sotto-classe di **CSubdivider**, permette la creazione di superfici di suddivisione Catmull-Clark. Gestisce anche i vertici di confine mediante le maschere riportate nella Figura 3.11.

**CNurss2** Classe che implementa NURSS quadratiche, deriva da **CSubdivider**. Pre-calcola i coefficienti dell'equazione (3.13) e gestisce i casi limite dovuti a spaziature nodali nulle come descritto nella sezione 4.1.2.

**CNurss3** Classe che realizza le NURSS cubiche gestendo le spaziature nodali nulle come presentato nella sezione 4.1.2. Anche questa classe è una specializzazione della classe base **CSubdivider**.

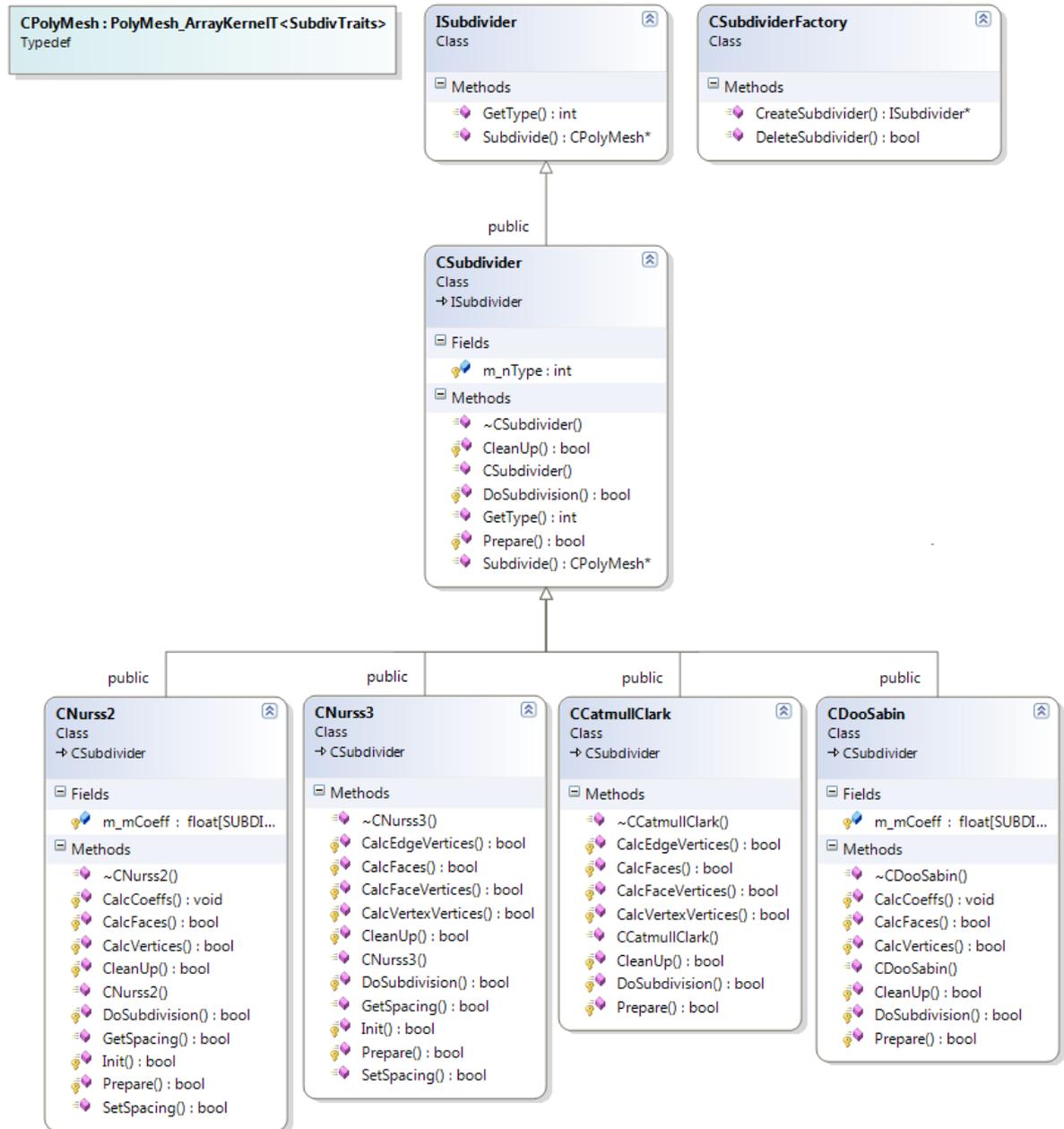


Figura 4.4: Diagramma delle classi.



# Capitolo 5

## Risultati numerici

Una volta implementati i diversi schemi di suddivisione presentati nel Capitolo 3, si è passati alla fase esecutiva generando diverse superfici di suddivisione tratte dalla letteratura e verificando così la bontà degli algoritmi scritti. Una volta validate le funzioni implementate, si sono confrontati lo schema di Doo-Sabin e quello di Catmull-Clark in termini d'accuratezza e tempi di calcolo; dapprima mediante i prototipi Matlab e successivamente con gli algoritmi in C++. Capite le differenze tra lo schema primale e quello duale, si è passati al confronto tra gli schemi uniformi e quelli non uniformi, focalizzando l'analisi sperimentale sui costi computazionali e sull'espressività aggiuntiva delle NURSS.

La Sezione 5.1 illustra i risultati numerici ottenuti, presentando prima le differenze tra lo schema primale Catmull-Clark e quello duale Doo-Sabin, e successivamente confrontando gli schemi uniformi e quelli non uniformi. La Sezione 5.2 offre, invece, alcune considerazioni sullo sviluppo delle superfici di suddivisione affinché possano essere utilizzate con profitto nell'ambito della progettazione industriale.

### 5.1 Confronto numerico fra i vari schemi

In questa sezione sono riportati i tempi di calcolo ottenuti con i prototipi in Matlab e le classi in C++ per la computazione di diverse superfici di suddivisione. I tempi registrati sono scomposti per ciascuna operazione; per lo schema Doo-Sabin, uniforme e non uniforme, sono riportati il tempo di computazione dei vertici (*vertici*), delle facce di tipo F (*facce F*), di tipo E (*facce E*) e di tipo V (*facce V*). Per lo schema Catmull-Clark, e per le NURSS cubiche, il tempo di calcolo è scomposto nella computazione dei punti di faccia (*vertici F*), di lato (*vertici E*) e di vertice (*vertici V*), e il tempo di calcolo delle nuove facce (*facce*). Inoltre, per ogni livel-

lo di suddivisione, sono riportati il numero di vertici che compongono la mesh di controllo.

Tutti i test sono stati eseguiti con un portatile Acer Aspire 5720G, processore Intel Core 2 Duo T5250 1.5 GHz, 2 GB di RAM DDR2, scheda video Nvidia GeForce 8400M GS, 256 MB di GRAM, risoluzione 1280x800, sistema operativo Windows Vista Home Premium 32 bit SP2.

### 5.1.1 Confronto tra schemi primali e duali

Nella Figura 5.1 sono illustrate le superfici di suddivisioni ottenute in Matlab a partire da un cubo applicando prima 3, e poi 5, passi degli schemi di Doo-Sabin e di Catmull-Clark. E' facile verificare che le superfici ottenute sono equivalenti a quelle riportate rispettivamente nelle Figure 3.6 e 3.13. La Tabella 5.1 riporta i tempi di calcolo, scomposti per tipo di operazione, relativi alla computazione delle superfici suddette.

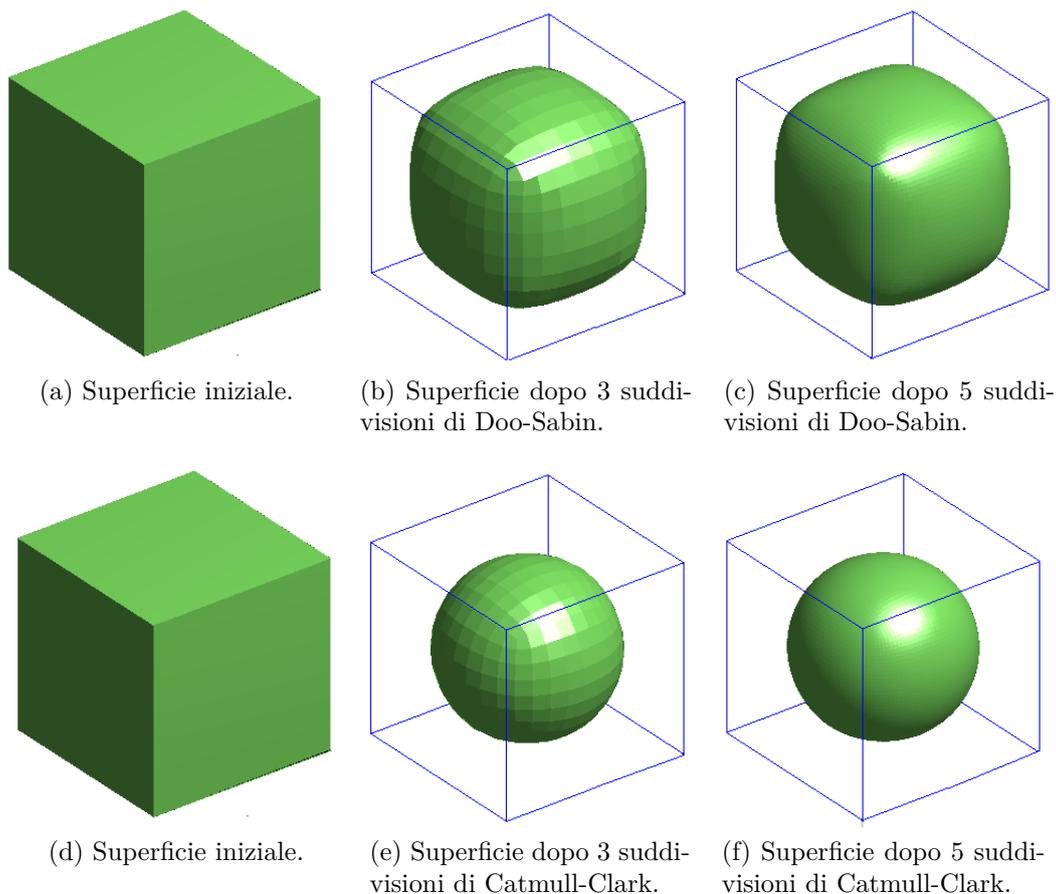


Figura 5.1: Illustrazione di un cubo ottenuta con i prototipi in Matlab.

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici	facce F	facce E	facce V	
3	384	0.0468	0.0780	0.0468	0.0780	0.2496
5	6144	0.7956	1.2012	0.8268	0.8268	3.6504

(a) Doo-Sabin.

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici F	vertici E	vertici V	facce	
3	386	0.0001	0.0780	0.0156	0.2028	0.2965
5	6146	0.1404	0.8580	0.4524	2.8704	4.3212

(b) Catmull-Clark.

Tabella 5.1: Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura 5.1.

Un secondo esempio è illustrato nella Figura 5.2 dove sono rappresentate le superfici di suddivisione Doo-Sabin e Catmull-Clark ottenute a partire da una mesh di controllo di 16 vertici che definisce un anello. I tempi di calcolo registrati in Matlab sono riportati nella Tabella 5.2.

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici	facce F	facce E	facce V	
3	1024	0.1716	0.1716	0.1404	0.1560	0.6396
5	16384	2.2620	3.3540	2.3088	2.1996	10.1244

(a) Doo-Sabin.

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici F	vertici E	vertici V	facce	
3	1024	0.0468	0.1092	0.0936	0.5148	0.7644
5	16384	0.5304	2.3088	1.2480	8.1589	12.2461

(b) Catmull-Clark.

Tabella 5.2: Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura 5.2.

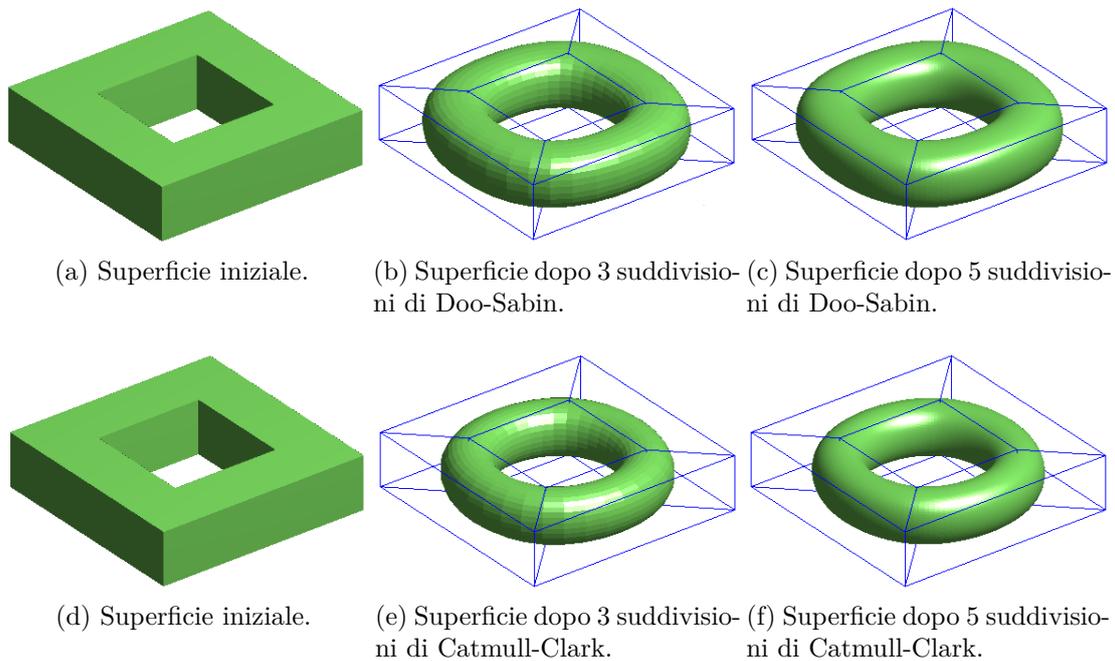


Figura 5.2: Illustrazione di un anello ottenuta con i prototipi in Matlab.

Analizzando le Tabelle 5.1 e 5.2 emerge che gli schemi Doo-Sabin e Catmull-Clark sono computazionalmente equivalenti, con prestazioni leggermente migliori per il primo algoritmo. Ciò giustifica la maggior diffusione dello schema Catmull-Clark che a fronte di un minimo overhead garantisce superfici  $C^2$ . Si può osservare, inoltre, che i tempi di calcolo dello schema Doo-Sabin sono uniformemente distribuiti nelle diverse sotto-operazioni. Mentre, per lo schema Catmull-Clark, il costo maggiore è dovuto alla definizione delle nuove facce; questo perché è necessario ricostruire tutte le informazioni topologiche della mesh.

Nella Figura 5.3 è illustrata la mesh di un vaso inizialmente costituita da 241 vertici. Sono stati successivamente applicati gli schemi Doo-Sabin e Catmull-Clark implementati in C++; i tempi di calcolo sono riportati nella Tabella 5.3. Le differenze computazionali dei due schemi si sono ulteriormente assottigliate. Si può osservare che adottando la struttura dati halfedge il costo di computazione è tutto nella creazione e nell'aggiornamento delle informazioni topologiche. Le diverse equazioni di raffinamento sono calcolate in tempi minimi anche a fronte di centinaia di migliaia di vertici poiché, grazie alla struttura halfedge, tutte le informazioni topologiche sono direttamente accessibili.

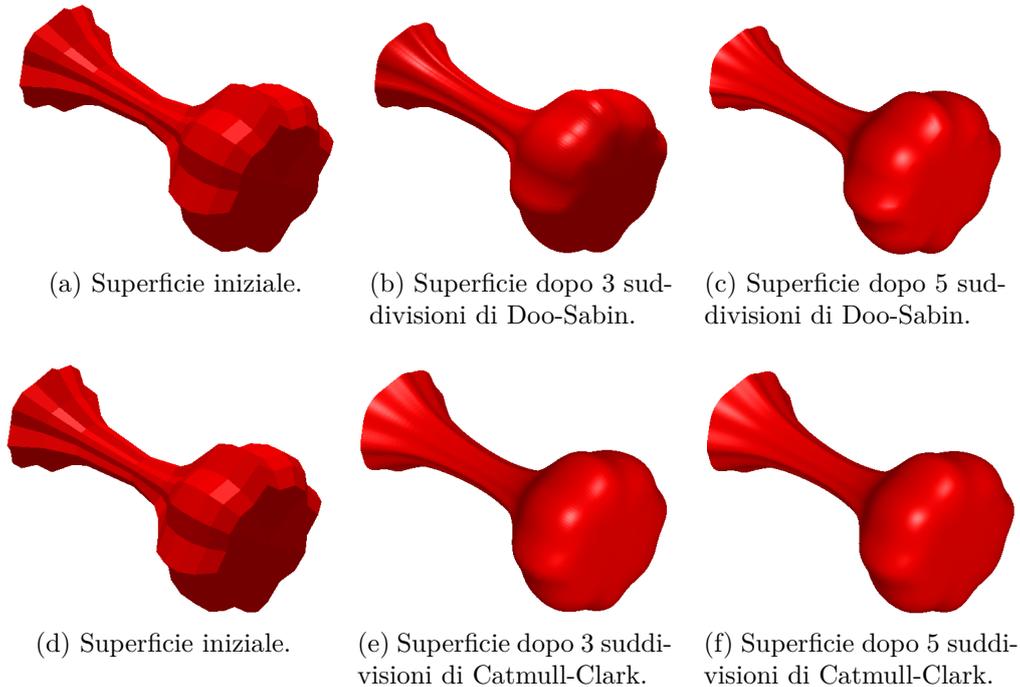


Figura 5.3: Illustrazione di un vaso ottenuta con le classi C++.

Gli schemi Doo-Sabin e Catmull-Clark, gestendo mesh quadrilatera, ad ogni passo di suddivisione accrescono di un fattore 4 il numero di vertici della mesh, come confermato dai valori delle Tabelle 5.1, 5.2 e 5.3. Questa crescita quadratica della complessità delle superfici di suddivisione si ripercuote nei tempi di calcolo che crescono di un ordine di grandezza per ogni livello di raffinamento. Ciò limita il numero di suddivisione effettivamente applicabili; mediante le classi implementate in C++ sono di fatto gestibili con un computer di fascia medio-bassa 6, 7 passi di raffinamento, arrivando a gestire circa un milione di vertici. Comunque, le superfici così ottenute sono sicuramente qualitativamente soddisfacenti.

### 5.1.2 Confronto tra schemi uniformi e non uniformi

La Figura 5.4 illustra la mesh di una croce composta inizialmente 24 vertici. A partire da tale mesh sono state ricavate le superfici di suddivisione, con 3 e 5 livelli di raffinamento, attraverso gli schemi Doo-Sabin e Catmull-Clark uniformi e non uniformi. Si può notare che le superfici di suddivisioni ottenute con le NURSS sono identiche alle rispettive controparti uniformi avendo utilizzato delle spaziature nodali equispaziate. I tempi di calcolo registrati in Matlab sono riportati nella Tabella 5.4. Si osserva che i costi computazionali delle NURSS quadratiche sono doppi rispetto a quelli dello schema Doo-Sabin, mentre per le NURSS cubiche i tempi sono quasi triplicati rispetto alla controparte uniforme. Ciò è dovuto alla ricerca delle

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici	facce F	facce E	facce V	
3	14240	0.0065	0.0336	0.0648	0.0322	0.1371
5	225920	0.1153	0.5846	1.1034	0.5588	2.3621

(a) Doo-Sabin.

Sudd.	Vertici	Tempi di calcolo [s]				totale
		vertici F	vertici E	vertici V	facce	
3	14801	0.0013	0.0026	0.0027	0.1412	0.1478
5	235841	0.0210	0.0458	0.0555	2.3964	2.5187

(b) Catmull-Clark.

Tabella 5.3: Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura 5.3.

spaziature nodali che per essere lette in maniera efficiente necessitano di una struttura dati che permetta di attraversare ordinatamente gli elementi del primo-anello di un vertice. La Figura 5.5 illustra la mesh di un pezzo meccanico inizialmente formata da 354 vertici. Sono stati applicati gli schemi di suddivisione uniformi e non uniformi implementati in C++ ottenendo i tempi di calcolo riportati nella Tabella 5.5. A differenza di quanto emerso con i prototipi Matlab, si osserva che le NURSS risultano del tutto equivalenti agli schemi Doo-Sabin e Catmull-Clark uniformi. Ciò è dovuto all'utilizzo della struttura dati halfedge che garantisce l'accesso efficiente agli elementi del primo-anello di un vertice. Si è visto, quindi, che le NURSS sono di fatto computazionalmente equivalenti alle superfici di suddivisione Doo-Sabin e Catmull-Clark. D'altra parte, gli schemi non uniformi sono molto più espressivi della controparte uniforme. Le NURSS permettono di rappresentare forme diverse mediante la stessa mesh di controllo; ad esempio, la Figura 5.6 mostra un superficie Doo-Sabin uniforme e tre sue varianti non uniformi ottenute impostando differenti spaziature nodali. Le NURSS permettono inoltre di rappresentare forme affilate non esprimibili con gli schemi Doo-Sabin e Catmull-Clark tradizionali. La Figura 5.7 riporta alcuni esempi di creste e cuspidi ottenute con le NURSS cubiche implementate in C++. A validazione degli algoritmi implementati, si fa notare la corrispondenza con le illustrazioni della Figura 3.29 tratte da [13]. In sintesi, le NURSS sono degli schemi di suddivisione molto interessanti perché, se opportunamente implementate, sono computazionali equivalenti ai rispettivi schemi uniformi, ma garantiscono un'espressività decisamente più ampia; infatti, a partire dalla stessa mesh iniziale, sono rappresentabili superfici di forme differenti e features affilate.

Sudd.	Vertici	Tempi di calcolo [s]				
		vertici	facce F	facce E	facce V	totale
3	1152	0.2028	0.2496	0.1248	0.1404	0.7176
5	18432	2.5896	3.8376	2.6832	2.4336	11.5440

(a) Doo-Sabin.

Sudd.	Vertici	Tempi di calcolo [s]				
		vertici	facce F	facce E	facce V	totale
3	1152	0.8268	0.4368	0.2028	0.1560	1.6224
5	18432	13.2757	7.0824	2.9016	2.4960	25.7557

(b) NURSS quadratiche.

Sudd.	Vertici	Tempi di calcolo [s]				
		vertici F	vertici E	vertici V	facce	totale
3	1154	0.0312	0.2028	0.0624	0.6084	0.9048
5	18434	0.5304	2.7456	1.3728	8.9389	13.5877

(c) Catmull-Clark.

Sudd.	Vertici	Tempi di calcolo [s]				
		vertici F	vertici E	vertici V	facce	totale
3	1154	0.3744	0.2808	0.4836	1.1856	2.3244
5	18434	6.5988	4.4772	8.8765	18.5641	38.5166

(d) NURSS cubiche.

Tabella 5.4: Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura 5.4.

		Tempi di calcolo [s]					
Sudd.	Vertici	vertici	facce F	facce E	facce V	totale	
3	22656	0.0110	0.0565	0.1112	0.0568	0.2355	
5	362496	0.1776	0.8740	1.7822	0.8646	3.6984	

(a) Doo-Sabin.

		Tempi di calcolo [s]					
Sudd.	Vertici	vertici	facce F	facce E	facce V	totale	
3	22656	0.0153	0.0654	0.1112	0.0549	0.2468	
5	362496	0.2500	0.9293	1.8364	0.9291	3.9448	

(b) NURSS quadratiche.

		Tempi di calcolo [s]					
Sudd.	Vertici	vertici F	vertici E	vertici V	facce	totale	
3	22656	0.0020	0.0040	0.0041	0.2158	0.2259	
5	362496	0.0329	0.0686	0.0887	3.5073	3.6975	

(c) Catmull-Clark.

		Tempi di calcolo [s]					
Sudd.	Vertici	vertici F	vertici E	vertici V	facce	totale	
3	22656	0.0057	0.0068	0.0048	0.2223	0.2396	
5	362496	0.0913	0.1275	0.0984	3.5467	3.8639	

(d) NURSS cubiche.

Tabella 5.5: Tempi di calcolo delle superfici di suddivisione per l'esempio di Figura 5.5.

## 5.2 Superfici di suddivisione per la progettazione industriale

In termini di interattività le superfici di suddivisione rappresentano una soluzione molto potente grazie alla possibilità di gestire mesh poligonali con topologia arbitraria. Gli algoritmi di suddivisione permettono agli utenti di sistemare i punti di controllo focalizzandosi sulle caratteristiche geometriche del modello, senza le problematiche dovute alla necessità di mantenere una mesh con struttura regolare. Ciò rende le superfici di suddivisione molto attraenti per la progettazione di superfici di forma libera. Il maggior difetto delle superfici di suddivisione è il ridotto controllo della superficie limite, un aspetto cruciale nella progettazione industriale.

Nei software per la progettazione industriale assistita al calcolatore, le Non-Uniform Rational B-Splines (NURBS) trimmate [23] sono lo standard *de-facto* per lo shape modeling. La modellazione mediante NURBS trimmate è ottenuta come interpolazione di una rete di curve. Il progettista deve seguire un flusso di lavoro obbligato: rappresenta le curve controllando accuratamente la regolarità delle stesse, le trimma per farle combaciare tra loro e solo in fine genera la superficie passante per le curve disegnate. E' una procedura piuttosto lunga e dispendiosa, ma garantisce un totale controllo della superficie progettata.

Gli schemi di interpolazione delle superfici di suddivisione non sono ancora ben compresi come quelli approssimanti e le difficoltà ancora presenti nella valutazione esatta della superficie limite riducono la loro applicazione in un contesto industriale. Un passo importante per un utilizzo diffuso delle superfici di suddivisione in ambito industriale è sicuramente legato allo sviluppo di un framework unificato, che permetta di eseguire con le superfici di suddivisione tutto quello che è fattibile con le NURBS. La modellazione di superfici di suddivisione a partire da reti di curve è un aspetto fondamentale, oltre alla definizione di forme regolari come sfere, cilindri, coni e superfici di rivoluzione [14].

Un secondo aspetto di sicuro interesse è la risoluzione dei problemi di interfacciamento e di compatibilità con software di modellazione basata sulle superfici parametriche [3, 17].

Infine, affinché le superfici di suddivisione possano essere utilizzate diffusamente, devono essere sviluppati adeguati strumenti di manipolazione che permettano operazioni booleane, offset, intersezioni e trimming [19]. Nonostante diversi lavori sono già stati presentati in questa direzione, c'è ancora molto spazio per ulteriori sviluppi.

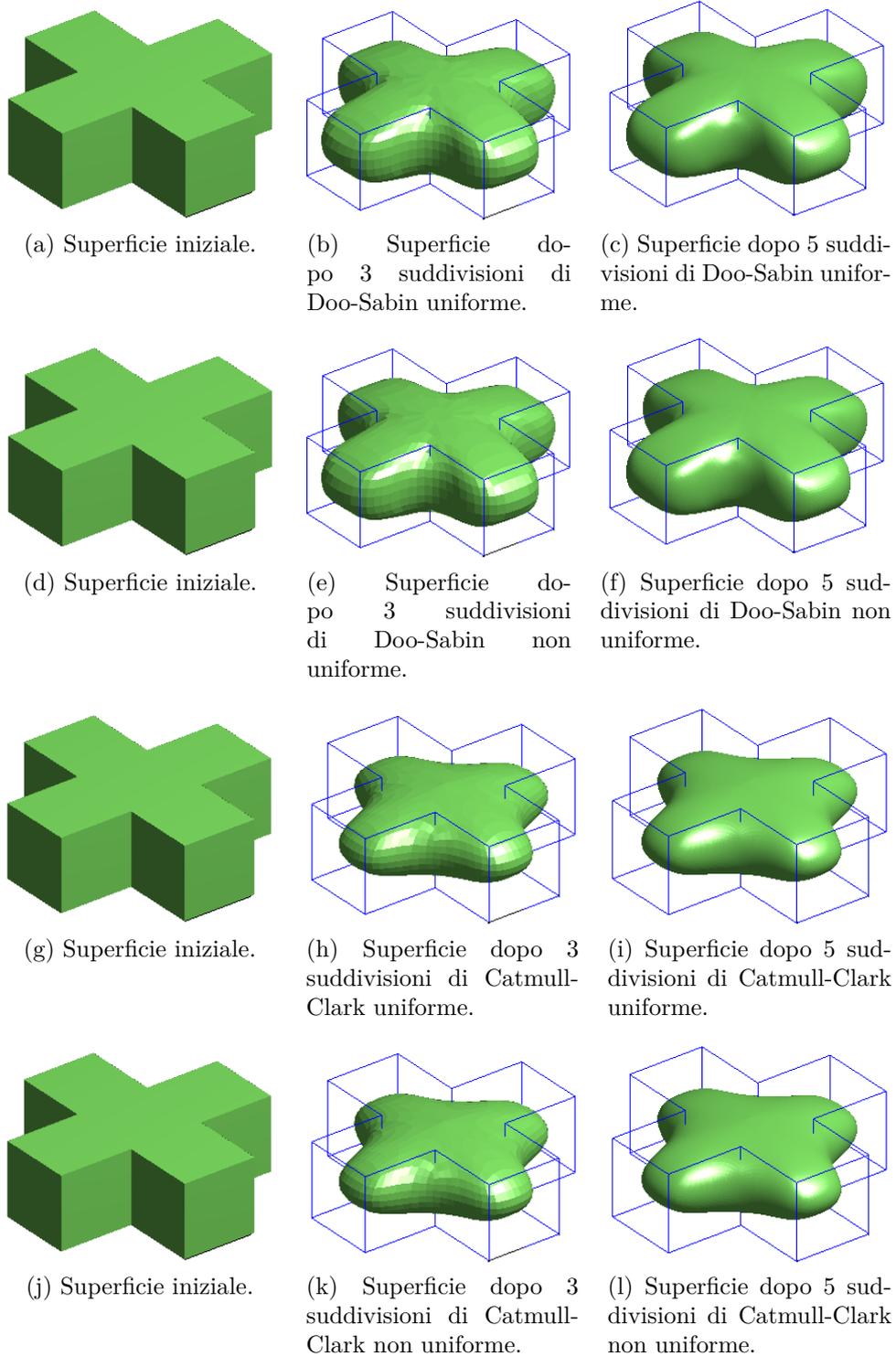
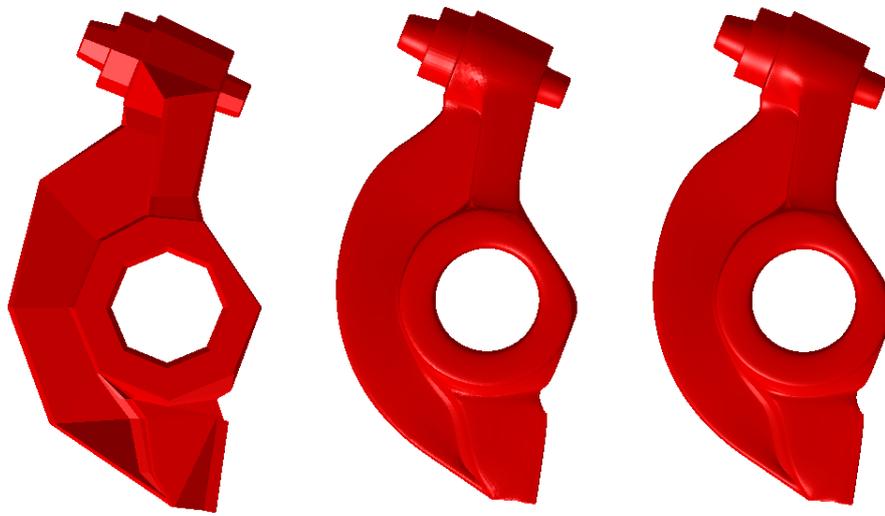
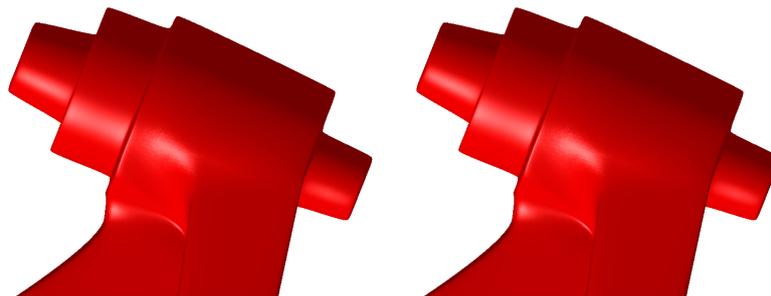


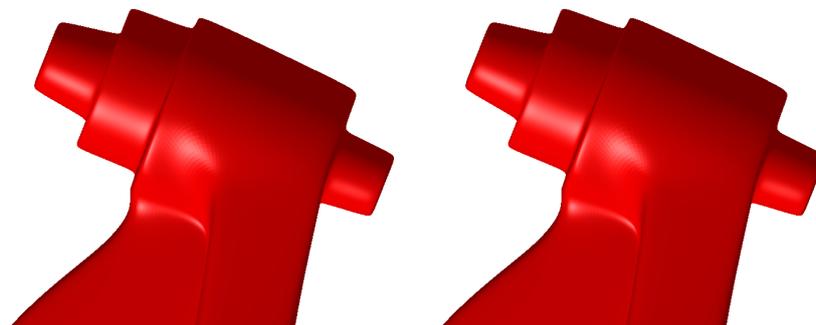
Figura 5.4: Illustrazione di una croce ottenuta con i prototipi in Matlab.



(a) Superficie iniziale. (b) Superficie dopo 3 suddivisioni di Doo-Sabin uniforme. (c) Superficie dopo 5 suddivisioni di Doo-Sabin uniforme.



(d) Zoom della superficie dopo 5 suddivisioni di Doo-Sabin uniforme. (e) Zoom della superficie dopo 5 suddivisioni di Doo-Sabin non uniforme.



(f) Zoom della superficie dopo 5 suddivisioni di Catmull-Clark uniforme. (g) Zoom della superficie dopo 5 suddivisioni di Catmull-Clark non uniforme.

Figura 5.5: Illustrazione di un pezzo meccanico ottenuta con le classi C++.

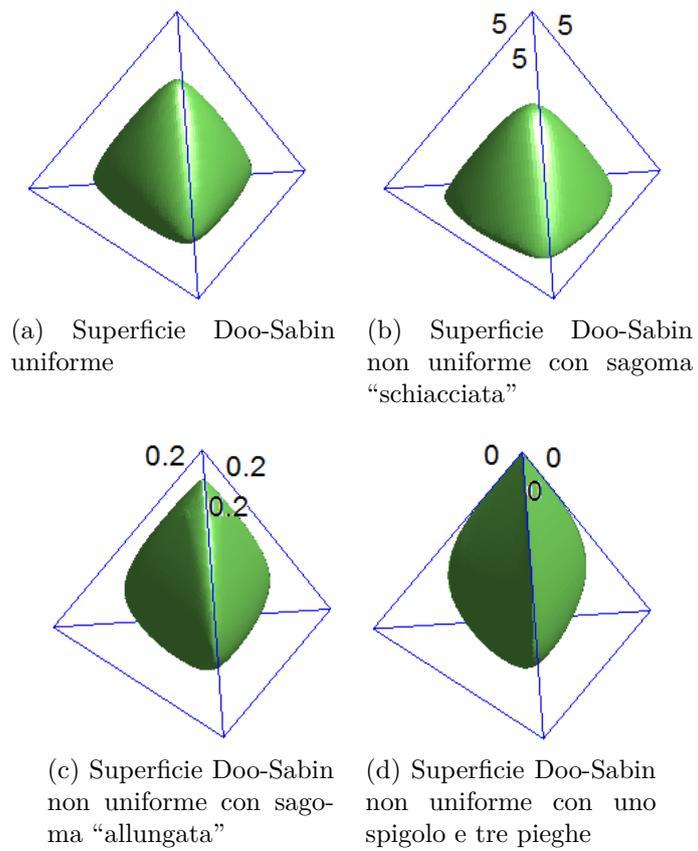


Figura 5.6: Superfici Doo-Sabin non uniformi con differenti spaziatore nodali ottenute mediante i prototipi in Matlab.

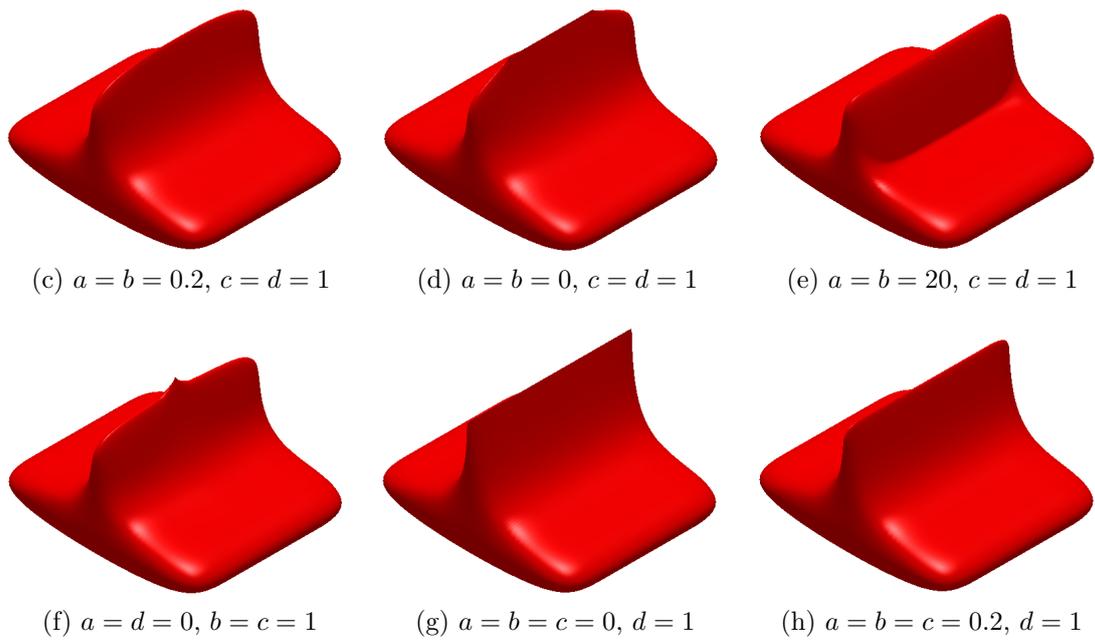
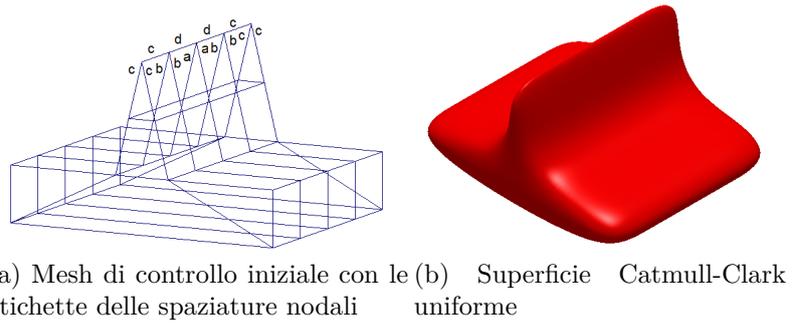


Figura 5.7: Superfici Catmull-Clark non uniformi con differenti spaziature nodali ottenute mediante le classi in C++.



# Capitolo 6

## Conclusioni

Le superfici NURBS sono lo strumento standard nella progettazione industriale per la modellazione geometrica con forme libere. Ma con l'aumento delle prestazioni dei calcolatori, le superfici di suddivisione sono divenute un'alternativa attraente. Con questo lavoro di tesi si è mostrato come sia possibile con le superfici di suddivisione superare il limite di topologie strettamente rettangolari imposto dalle superfici NURBS e di gestire features, quali creste e cuspidi, anche nelle applicazioni CAD per la progettazione industriale.

Grazie allo sviluppo di prototipi in Matlab degli schemi di suddivisione di Doo-Sabin e Catmull-Clark non uniformi, si è verificato che tali algoritmi possono rappresentare superfici con topologia arbitraria, come i tradizionali schemi uniformi, permettendo in più la definizione di creste, cuspidi, pieghe e spigoli, mantenendo lo stesso ordine di continuità della controparte uniforme. Inoltre, grazie ad un attento studio delle superfici NURSS, sono state presentate alcune soluzioni analitiche che permettono di gestire spaziature nodali nulle in situazioni limite non trattate esplicitamente dalla letteratura. In questo lavoro sono stati messi in evidenza anche gli aspetti critici per un'implementazione efficiente degli schemi di suddivisione suddetti; grazie ad un'accurata implementazione in C++ si è dimostrato come sia possibile avvalersi degli schemi non uniformi con costi computazionali equivalenti a quelli degli schemi uniformi. Le strutture dati utilizzate e gli algoritmi implementati permettono di gestire agevolmente 6, 7 passi di suddivisione e mesh con circa un milione di vertici anche mediante calcolatori con prestazioni medio-basse. Queste caratteristiche sono sufficienti per gestire pezzi industriali in maniera soddisfacente attraverso comuni applicazioni CAD. Tuttavia, per utilizzare con profitto le superfici di suddivisione in un contesto industriale, mancano opportuni strumenti di interpolazione, per la creazione di superfici da una rete di curve, e di manipolazione, per eseguire operazioni booleane e offset.

Come anticipato da Ma in [13], affinché le superfici di suddivisione possano essere utilizzate nell'ambito CAD, è necessario analizzare e studiare schemi di suddivisione che permettano di fare tutto ciò che è realizzabile con le NURBS. Per tale motivo, sviluppi futuri di questo lavoro di tesi potrebbero riguardare lo studio di schemi di suddivisione che si muovono in questa direzione. Innanzitutto si potrebbero analizzare le superfici T-NURCC (Non-Uniform Rational Catmull-Clark con giunzioni T) i cui fondamenti teorici sono presentati da Sederberg in [28]. Le T-NURCC sono una generalizzazione delle NURBS cubiche e delle superfici Catmull-Clark che permettono di gestire giunzioni a T e, se non sono presenti spaziature nodali nulle, il raffinamento locale senza alterare la superficie limite. In particolare, le NURCC sono identiche alle NURSS cubiche, quindi generano superfici limite  $C^2$  tranne nei punti straordinari e nelle features, ma le spaziature nodali dei lati opposti di ciascuna faccia quadrilatera devono essere uguali, ciò rendere le NURCC stazionarie. Le T-NURCC aggiungono alle NURCC la raffinatezza locale.

Un altro schema di suddivisione che sarebbe interessante analizzare è quello delle superfici DINUS (Double Insertion Non-Uniform Stationary) presentate da Muller in [16] che generalizzano le spline non uniformi cubiche e le superfici Catmull-Clark. Anche le DINUS sono stazionarie e gestiscono features e intervalli nodali arbitrari. A differenza delle NURSS, però, forniscono delle regole per il calcolo dei punti e delle normali limite fondamentali per molte applicazioni.

Di sicuro interesse sono anche le superfici di suddivisione compatibili con le NURBS proposte da Cashman in [3] che permettono di gestire superfici con topologia arbitraria e di rappresentare NURBS di qualsiasi grado. Tali superfici permettono di ottenere la curvatura voluta, di qualsiasi grado, anche in corrispondenza di punti straordinari.

Infine, altri settori esplorabili sono l'implementazione parallela dei diversi schemi di suddivisione sfruttando le capacità di calcolo delle moderne GPU e lo studio di schemi di suddivisione a mesh triangolare.

# Bibliografia

- [1] Baumgart B.G.: A Polyhedron Representation for Computer Vision. National Computer Conference, Anaheim, CA, 1975, 589-596.
- [2] Botsch M., Steinberg S., Bischoff S., Kobbelt L.: OpenMesh - a generic and efficient polygon mesh data structure. OpenSG Symposium, 2002.
- [3] Cashman T.J.: NURBS-compatible subdivision surfaces. Technical report 773, University of Cambridge, Computer Laboratory, 2010.
- [4] Catmull E., Clark J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design 10, 1978, 350-355.
- [5] Chaikin G. An algorithm for high-speed curve generation. Computer Graphics and Image Processing 3, 1974, 346-349.
- [6] Chieppa L., Liverani A.: Subdivision surfaces for industrial design. 1st Symposium on Multidisciplinary Studies of Design in Mechanical Engineering, 2008.
- [7] Doo D., Sabin M.: Behaviors of recursive division surfaces near extraordinary points. Computer Aided Design 10, 1978, 356-360.
- [8] Dyn N., Levin D., Gregory J.A.: A butterfly subdivision scheme for surface interpolation with tension control. ACM Trans. Graph. 9, 1990, 160-169.
- [9] Farin G.: Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide, Second Edition. Academic Press, Inc., 1990.
- [10] Kettner L.: Using Generic Programming for Designing a Data Structure for Polyhedral Surfaces, in Proc. 14th Annual ACM Symp. on Computational Geometry, 1998.
- [11] Kobbelt L.:  $\sqrt{3}$ -subdivision. Proceeding of ACM SIGGRAPH computer graphics 2000, 103-112.

- [12] Kobbelt L.: Online documentation of OpenMesh, <http://www.openmesh.org/>.
- [13] Ma W.: Subdivision surfaces for CAD- an overview. *Computer Aided Design* 37, 2005, 693-709.
- [14] Morin G., Warren J., Weimer H.: A subdivision scheme for surfaces of revolution. *Computer Aided Geometric Design* 18, 2001, 483-502.
- [15] Muller H., Jaeschke R.: Adaptive subdivision curves and surfaces. *Computer graphics International*, 1998.
- [16] Muller K., Funzig C., Reusche L., Hansford D., Farin G., Hagen H.: DINUS: Double Insertion, Nonuniform, Stationary, Subdivision Surfaces. *ACM Transaction on Graphics* 29, 2010.
- [17] Muller K., Reusche L., Fellner D.: Extended Subdivision Surfaces: building a bridge between NURBS and Catmull-Clark surfaces. *ACM Transaction on Graphics* 25, 2006.
- [18] Li G., Ma W., Bao H.:  $\sqrt{2}$  Subdivision for quadrilater meshes. *Vis Computer Graphics*, 2004, 180-198.
- [19] Litke N., Levin A., Schroder P.: Trimming for subdivision surfaces. *Computer Aided Geometric Design* 18, 2001, 463-481.
- [20] Loop C.: Smooth subdivision surfaces based on triangles, Master's Thesis, Department of Mathematics, University of Utah, 1987.
- [21] Miura K.T., Masuda H.: Selective Non-Uniform Subdivision. 10th Pacific Conference on Computer Graphics and Applications, 2002.
- [22] Peters J., Reif U.: *Subdivision Surfaces*. Springer Series in Geometry and Computing, 2008.
- [23] Piegl L., Tiller W.: *The NURBS Book* 2nd edition. Springer, 1997.
- [24] Romani L., Conti C., Gemignani L.: From approximating to interpolatory non-stationary subdivision schemes with the same generation properties. *Advanced in Computational Mathematics*, 2010.
- [25] Romani L., Beccari C., Casciola G.: Polynomial-based non-uniform interpolatory subdivision with features control. *Journal of Computational and Applied Mathematics*, 2010.

- [26] Sabin M.A.: Subdivision Surfaces. In: Farin G.E., Hoschek J., Kim M.S., editors. Handbook of computer aided geometric design. New York: North Holland; 2002, 309-325.
- [27] Sederberg T.W., Zheng J., Sewell D., Sabin M.: Non-uniform recursive subdivision surfaces. Proceeding of the ACM SIGGRAPH computer graphics 1998, 387-394.
- [28] Sederberg T.W., Zheng J., Bakenov A., Nasri A.: T-splines and T-nurccs. ACM Trans. Graph. 22, 2003, 477-484.
- [29] Sovakar A., Kobbelt L.: API design for adaptive subdivision schemes. Computer Graphics 24, 2004, 67-72.
- [30] Stam J.: Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. Proceedings of th ACM SIGGRAPH computer graphics, 1998, 395-404.
- [31] Stam J.: On Subdivision schemes generalizing uniform B-spline surfaces of arbitrary degree. Computer Aided Geometric Design 18, 2001, 383-396.
- [32] Stam J., Loop C.: Quad/triangle subdivision. Computer Graphics Forum 22, 2003, 1-7.
- [33] Tobler R.F., Maierhofer S.: A mesh data structure for rendering and subdivision. Short Communications proceedings, 2006.
- [34] Zorin D., Schroder P.: Subdivision for modeling animation. ACM SIGGRAPH Course notes, 2000.
- [35] Zorin D., Schroder P.: A unified framework for primal/dual quadrilateral subdivision scheme. Computer Aided Design 18, 2001, 429-454.