

L'ALGORITMO PAM NELLA MODELLAZIONE CON SUPERFICI SPLINE IN COORDINATE SFERICHE

Nota di ALESSANDRO AMOROSO, GIULIO CASCIOLA *

Presentata da ILIO GALLIGANI, Accademico Corrispondente Residente
nella seduta del 22 Febbraio 1995

Abstract. This note presents a PAM (Pseudopolar Array Mask) method application in geometric modeling area: spherical grid spline surfaces graphics visualization with hidden line removal. A PAM method efficient extension for contouring is presented too. Contouring is an often used and important tool for visualizing and interrogating surfaces and for judging the aesthetic quality of a surface in modeling stage.

1. Introduzione

Uno dei maggiori problemi nella Computer Aided Geometric Design (CAGD) è l'integrazione di tecniche per superfici a forma libera nei sistemi di modellazione solida. Nella modellazione di superfici vengono usate per lo più superfici in forma parametrica polinomiali a tratti. Queste superfici non sono in generale chiuse e quindi non è detto che definiscano un oggetto. Inoltre, se definiscono un oggetto solido, non esiste un algoritmo semplice per determinare se un dato punto sia interno od esterno al solido. A questo scopo sarebbero più utili le superfici in forma algebrica che definiscono in modo naturale un semispazio, ma per la difficoltà che si incontra nell'elaborarle il loro utilizzo è limitato a facce piane e quadriche. Un differente approccio è basato sull'uso di superfici ad un sol valore in coordinate cilindriche [5] o in coordinate sferiche (ρ, θ, ϕ) (vedi Figura 1). Infatti molte parti di oggetti di tipo ingegneristico possono essere descritte come superfici ad un sol valore, $\rho = \rho(\theta, \phi)$, in un sistema di coordinate sferiche.

Il vantaggio principale di questa rappresentazione è che la classificazione di un punto può essere realizzata in modo semplice: un dato punto $\mathbf{p}_0 = (\rho_0, \theta_0, \phi_0)$ è interno al volume definito dalla superficie se $\rho_0 < \rho(\theta_0, \phi_0)$. Queste superfici generano due semispazi che possono essere combinati mediante l'uso di operazioni Booleane per realizzare oggetti solidi chiusi e possono quindi essere incorporati come primitive in sistemi di modellazione solida.

*Questa ricerca è stata finanziata dal CNR-Italy, Contratto n.94.00132.CT01

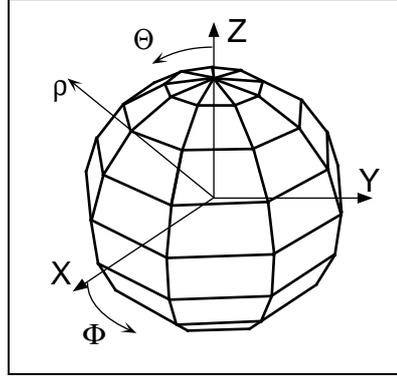


Figura 1: Sistema di coordinate sferiche

In questo lavoro si vuole da un lato proporre il metodo PAM come strumento per la visualizzazione con eliminazione di linee nascoste nella modellazione di superfici in coordinate sferiche e dall'altro si vuole proporre una variante di tale metodo per effettuare il contouring di superfici.

Si ricorda che il contouring o determinazione delle linee di livello è un approccio importante e spesso usato per il problema della visualizzazione e interrogazione di superfici nella modellazione e viene usato estensivamente nell'industria automobilistica per valutare la qualità estetica di una superficie.

2. Superfici spline in coordinate sferiche

In questa sezione definiamo funzioni spline ad un sol valore $s(\theta)$ di ordine m in coordinate polari (ρ, θ) [4, 6]. In modo simile al caso Cartesiano, si partiziona l'intervallo di definizione $[\theta_a, \theta_b]$ mediante una successione non decrescente di nodi $\Theta = \{\theta_i\}_{i=1, \dots, 2m+K}$ con

$$\theta_1 = \dots = \theta_m = \theta_a < \theta_{m+1} \leq \dots \leq \theta_{m+K} < \theta_b = \theta_{m+K+1} = \dots = \theta_{2m+K}$$

ed il vincolo che $\theta_{i+1} - \theta_i < \pi$.

Su questa partizione nodale si definiscono le seguenti funzioni in modo ricorrente:

$$M_{i,h}(\theta) = \sin\left(\frac{\theta - \theta_i}{m-1}\right)\beta_{i,h-1}(\theta) + \sin\left(\frac{\theta_{i+h} - \theta}{m-1}\right)\beta_{i+1,h-1}(\theta) \quad (1)$$

$$h = 2, \dots, m$$

dove

$$\beta_{i,h}(\theta) = \begin{cases} M_{i,h}(\theta) / \sin\left(\frac{\theta_{i+h} - \theta_i}{m-1}\right) & \text{se } \theta_i \neq \theta_{i+h} \\ 0 & \text{altrimenti} \end{cases}$$

e

$$M_{i,1}(\theta) = \begin{cases} 1 & \text{se } \theta_i \leq \theta < \theta_{i+1} \\ 0 & \text{altrimenti} \end{cases}$$

Si può dimostrare che queste funzioni sono linearmente indipendenti da cui una funzione spline può essere scritta in modo univoco come

$$\delta(\theta) = \sum_{i=1}^{m+K} \delta_i^{-1} M_{i,m}(\theta)$$

con $s(\theta) = \delta(\theta)^{-1}$ e $\mathbf{d}_i = (\delta_i, \xi_i)$ i punti di controllo localizzati sulle direzioni radiali di Greville

$$\xi_i = \frac{\theta_{i+1} + \dots + \theta_{i+m-1}}{m-1} \quad i = 1, \dots, m+K$$

Le funzioni spline in coordinate polari godono delle seguenti proprietà:

- precisione lineare: se m punti di controllo sono allineati, la spline riproduce questo segmento retto;
- guscio convesso: ogni punto della funzione giace nel guscio convesso di al più m punti di controllo;
- variation diminishing: ogni retta interseca la spline un numero inferiore od uguale di volte di quante interseca la poligonale dei punti di controllo;
- coniche: le spline di ordine $m = 3$ (grado 2) sono coniche a tratti.

Funzioni spline $s(\theta, \phi)$ ad un sol valore di ordine (m, n) in coordinate sferiche sono definite mediante due successioni non decrescenti di nodi $\Theta = \{\theta_i\}_{i=1, \dots, 2m+K}$ nella dimensione θ e $\Phi = \{\phi_i\}_{i=1, \dots, 2n+H}$ nella dimensione ϕ e da una griglia di punti di controllo $\mathbf{d}_{ij} = (\delta_{ij}, \xi_i, \eta_j)$ che sono localizzati sulle direzioni radiali date da

$$\xi_i = \frac{\theta_{i+1} + \dots + \theta_{i+m-1}}{m-1} \quad i = 1, \dots, m+K$$

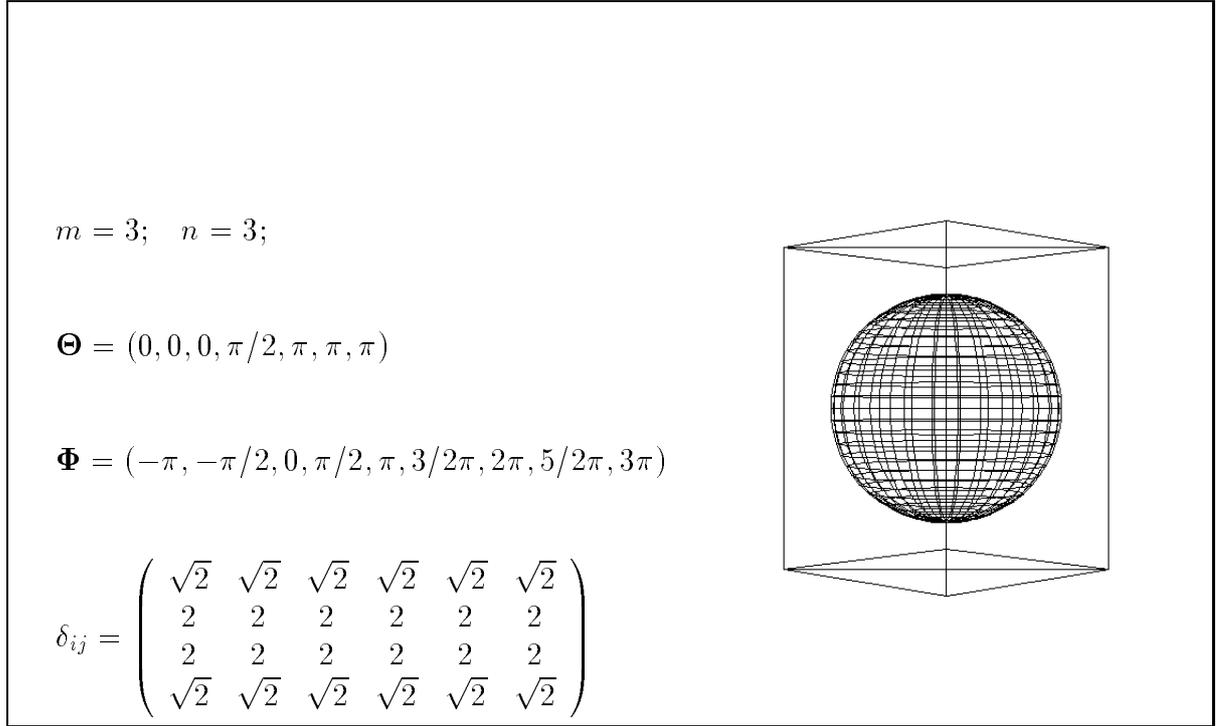


Figura 2: Esempio 1; sfera di centro l'origine e raggio unitario

$$\eta_j = \frac{\phi_{j+1} + \dots + \phi_{j+n-1}}{n-1} \quad j = 1, \dots, n+H$$

Allora la superficie $s(\theta, \phi)$ viene definita come $\delta(\theta, \phi)^{-1}$ con

$$\delta(\theta, \phi) = \sum_{i=1}^{m+K} \sum_{j=1}^{n+H} \delta_{ij}^{-1} M_{i,m}(\theta) M_{j,n}(\phi).$$

Nelle Figure 2, 3 e 4 vengono presentati tre semplici esempi di modellazione. L'esempio 1 illustra i parametri necessari per definire la superficie sfera di centro l'origine e raggio unitario, mentre gli esempi 2 e 3 mostrano due superfici a forma libera e le griglie dei punti di controllo che le generano.

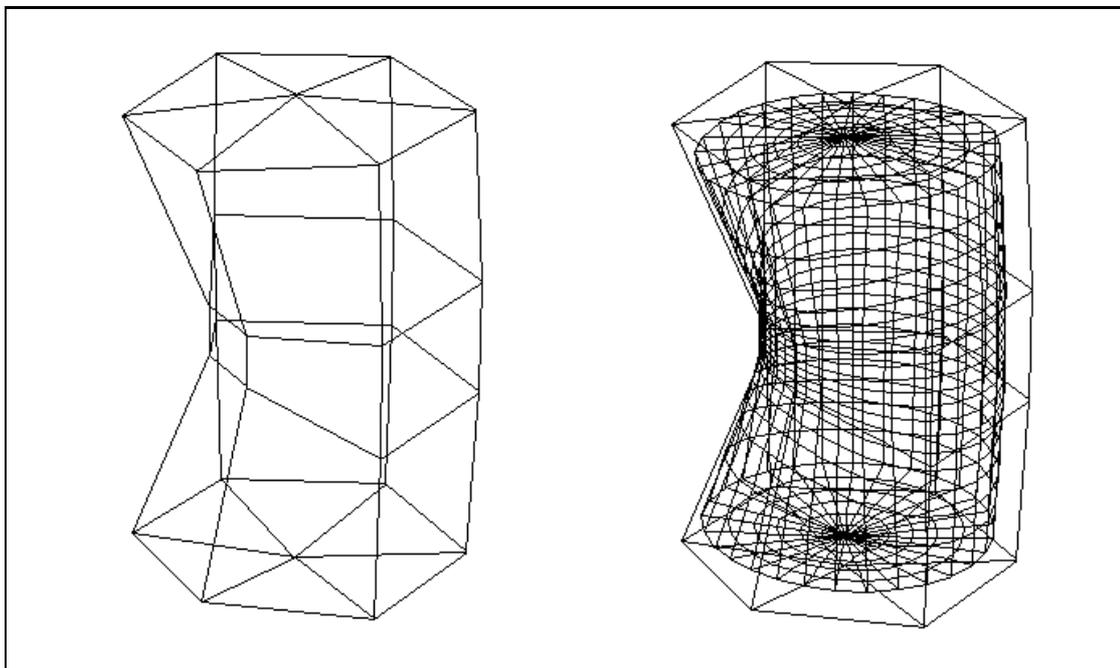


Figura 3: Esempio 2

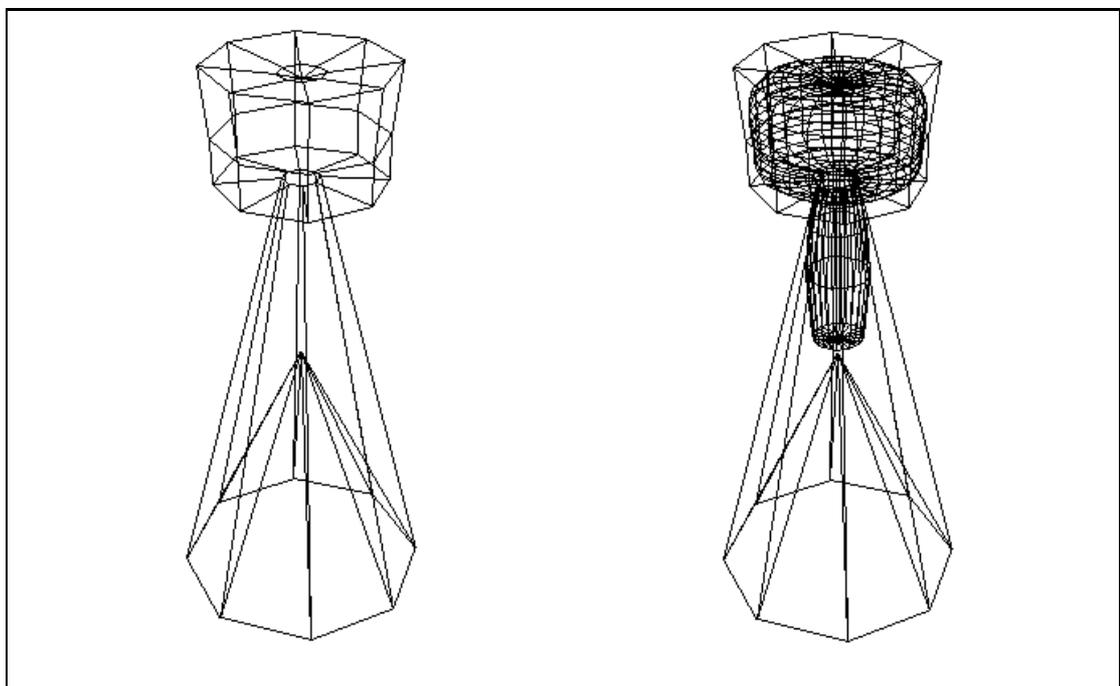


Figura 4: Esempio 3

3. L'algoritmo PAM

In [1, 2] è stato presentato l'algoritmo PAM (Pseudopolar Array Mask) per la rappresentazione prospettica con eliminazione delle linee nascoste di funzioni continue in due variabili in coordinate sferiche. Tale algoritmo opera in coordinate schermo e lavora quasi esclusivamente in aritmetica intera; è esatto al pixel e generale nel senso che può rappresentare una qualunque funzione continua da un punto di vista arbitrario. Nei lavori citati viene confrontato con gli algoritmi noti in letteratura risultando efficiente e competitivo soprattutto per alte grigliature, fornendo immagini ad alta definizione e a basso costo.

Nella fase della progettazione e modellazione di superfici si ha l'esigenza, spesso contrastante, di una rappresentazione in tempo reale e nel contempo di buona qualità; nel caso particolare di modellazione con superfici in coordinate sferiche tale esigenza può essere risolta in modo naturale dall'algoritmo PAM.

Nelle sue linee essenziali il metodo PAM consiste nel processare una grigliatura della superficie; quindi sfrutta la proprietà di funzione ad un sol valore per determinare in modo semplice un percorrimto degli elementi della griglia, ossia un ordine in cui processare tali elementi che sia corretto rispetto all'ordine di visibilità dal punto di vista fissato.

Infine sfrutta la proprietà della funzione di essere continua osservando che la regione di invisibilità, che viene generata proiettando gli elementi di superficie (facet) corrispondenti agli elementi griglia nell'ordine determinato degli elementi stessi, è connessa ed è star-convex rispetto all'origine del sistema sul piano di proiezione; ciò permette di conservare tale regione in modo semplice con una funzione discreta in coordinate polari (o pseudopolari definendo tali un sistema di coordinate polari nel discreto) cioè in termini implementativi con un array monodimensionale.

4. Linee di livello

Il problema delle linee di livello è raramente descritto in letteratura perché comporta due problemi non banali. Il problema principale è quello della determinazione delle linee di livello stesse, il secondo quello della rimozione delle linee nascoste. Gli algoritmi noti sono molto complicati, e l'enfasi maggiore è posta sulla problematica delle rimozione delle linee nascoste. Gli algoritmi noti usano tecniche di calcolo di linee di livello alle quali successivamente applicano quelle per la rimozione di linee nascoste, realizzando una sorta di esecuzione in serie ("pipe-line") tra i due metodi. In questo lavoro si propone una variante del metodo PAM che permette in modo semplice e non costoso il calcolo delle linee di livello in parallelo alla rimozione delle parti nascoste.

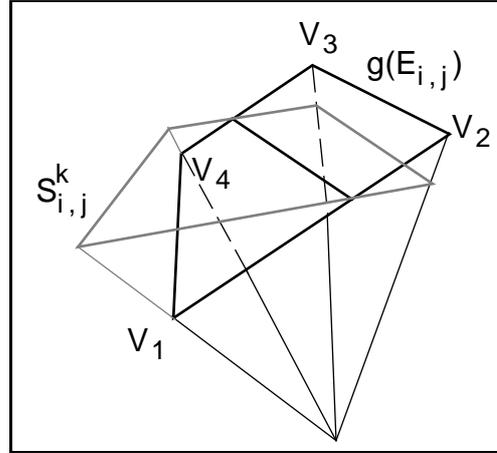


Figura 5: Intersezione tra la facet $g(E_{i,j})$ della funzione, in nero, e la corrispondente della k^{ma} sfera di livello, in grigio

4.1. Determinazione delle linee di livello

Esistono due approcci fondamentali per il calcolo delle linee di livello:

- *gridless*: le curve di livello vengono calcolate come espressione analitica dell'intersezione tra superfici, quella da rappresentare e quelle di livello
- *grid-type*: le curve di livello vengono calcolate in base ad una grigliatura della superficie; al rinfittirsi delle grigliatura aumenta l'accuratezza delle curve

L'approccio seguito in questo lavoro è del secondo tipo. Questo permette di modificare l'algoritmo PAM in modo che prima di elaborare i lati di una facet, vengano calcolate tutte le eventuali linee di livello su quella facet e disegnate secondo le stesse condizioni di visibilità alle quali è soggetta la facet stessa, ma senza comportare aggiornamento della regione di invisibilità; l'elaborazione procede quindi inalterata.

In coordinate polari le linee di livello sono generate dalle intersezioni della superficie con sfere concentriche equidistanti tra loro.

Ogni sfera di livello viene considerata come una funzione di valore costante, di raggio ρ_k ; si denota con $S_{i,j}^k$ la facet i,j della k^{ma} sfera di livello. Le intersezioni tra superficie ed ogni $S_{i,j}^k$ possono essere calcolate come quelle tra ogni facet $g(E_{i,j})$ della funzione e la corrispondente della sfera, $S_{i,j}^k$.

L'idea di base dell'algoritmo è di calcolare le intersezioni tra facet nello spazio oggetto, individuando i punti comuni ai lati di $g(E_{i,j})$ ed al piano passante per i vertici di $S_{i,j}^k$. Se esistono tali punti di intersezione, allora i segmenti che li uniscono sono le curve di livello per quella facet.

L'algoritmo adottato per il calcolo delle linee di livello può essere riassunto come indicato nello schema seguente:

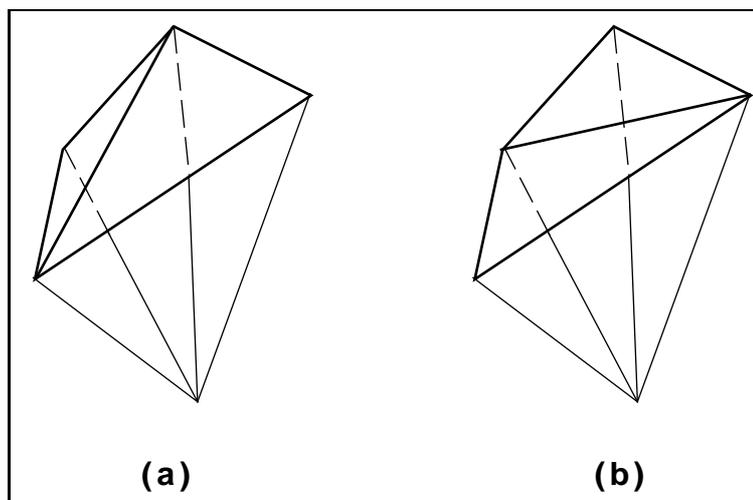


Figura 6: Possibili triangolarizzazioni di una facet quadrangolare

siano $V_i = (\rho_i, \Theta_i, \Phi_i)$ $i = 1, 2, 3$, e/o 4 le coordinate dei vertici della facet

siano i lati della facet $L_1 = \overline{V_1V_2}$, $L_2 = \overline{V_2V_3}$,

se la facet ha 3 lati allora

$$L_3 = \overline{V_3V_1}$$

altrimenti {la facet ha 4 lati}

$$L_3 = \overline{V_3V_4}, L_4 = \overline{V_4V_1}$$

per ogni sfera di livello, S^k , fai

calcola il piano Ω per $(\rho_k, \Theta_k, \Phi_k)$ $i = 1, 2, 3$

calcola i punti di intersezione PI_i tra Ω ed i lati L_i $i = 1, 2, 3$ e/o 4

per $i = 1, \dots, (\text{numero_dei_punti_di_intersezione} - 1)$, passo = 2

traccia la linea di livello tra PI_i e PI_{i+1}

Con un semplice test preliminare, sui valori ρ dei vertici V_i , si può determinare se ogni lato di una facet ha intersezione con la k^{ma} sfera di livello e se questa coincide con uno dei vertici.

Le possibili localizzazioni dei punti di intersezione tra i lati delle facet sono:

- non ci sono intersezioni
- due punti, su lati distinti, adiacenti od opposti;
- quattro punti, ognuno per lato

Ogni punto può coincidere con uno dei vertici del lato.

Ogni facet quadrangolare può essere considerata come formata da due triangoli contigui (vedi Figura 6). Potrebbe sorgere il problema di come considerare le linee

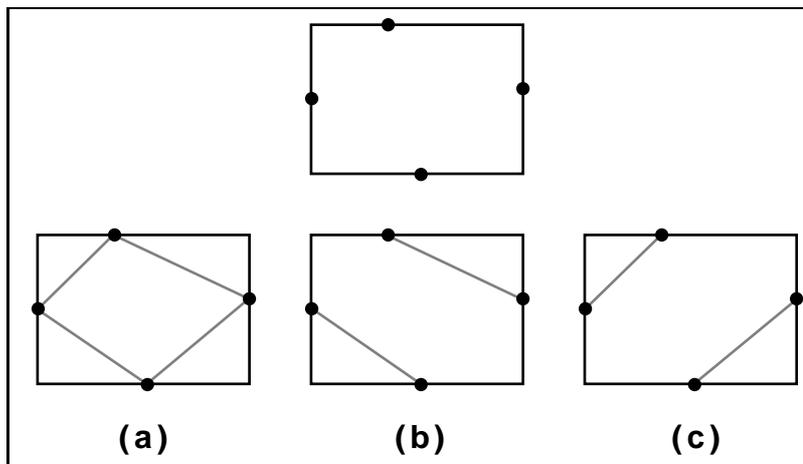


Figura 7: Possibili politiche di congiunzione di quattro intersezioni

di livello in base ai punti di intersezione calcolati se questi sono più di due. Facendo riferimento alla Figura 7 si considera che ogni punto può appartenere ad una sola linea di livello, infatti, per la triangolarizzazione della facet, non ha alcun senso geometrico la soluzione 7(a). Le due configurazioni (b) e (c) sono entrambe possibili, infatti la scelta tra di loro dipende dalla triangolazione, ma per le ipotesi sulla grigliatura non è possibile stabilire quale sia quella esatta. Queste considerazioni sono del tutto superflue nel caso di due soli punti di intersezione.

4.2. Rimozione delle linee nascoste

Il problema della rimozione delle linee nascoste viene risolto in modo automatico con l'applicazione dell'algoritmo PAM.

Per costruzione ogni segmento di linea di livello giace sulla superficie di una facet, per cui è soggetto alle sue stesse regole di visibilità. Ogni elemento di linea di livello viene quindi tracciato rispettando la regione di invisibilità, senza comportarne l'aggiornamento.

5. Alcuni risultati sperimentali

Il metodo proposto è stato implementato in Turbo Pascal 5.0 su PC 486DX a 33MHz. L'algoritmo è stato testato su molte superfici; in questa nota vengono presentati alcuni risultati relativi alle prestazioni su quattro superfici spline in coordinate sferiche generate per modellazione e presentate nelle Figure 8, 9, 10 e 11 dove vengono denominate A,B,C e D.

Ogni tabella presentata riporta le prestazioni ottenute elaborando una di tali superfici con una image-size di 600 pixel; viene confrontato il tempo impiegato dall'algoritmo PAM per la rimozione di linee nascoste su differenti grigliature con il tempo dell'algoritmo PAM modificato per fornire anche le linee di livello (PAM + LL). Il costo delle sole linee di livello rispetto al totale viene riportato in percentuale nell'ultima colonna.

È interessante analizzare separatamente il costo della determinazione delle linee di livello da quello della visualizzazione con rimozione delle parti nascoste; ciò permette di constatare come la prima parte sia nettamente più costosa della seconda.

Dalle prove presentate si osserva come la determinazione e la rimozione delle linee di livello, che come problemi a se stanti sono molto costosi, inseriti nell'algoritmo PAM risultano poco costosi, come si può osservare dai tempi ottenuti.

<i>griglia</i>	<i>Linee Livello</i>	<i>Tempo PAM</i>	<i>Tempo PAM + LL</i>	<i>calcolo LL</i>	<i>hidden LL</i>	<i>costo LL</i>
50 × 50	5	2.47	3.85	0.94	0.44	36%
50 × 50	10	2.47	5.18	1.82	0.89	52%
100 × 100	5	5.50	8.40	2.46	0.44	35%
100 × 100	10	5.50	11.36	4.88	0.98	52%
150 × 150	5	10.11	15.89	5.05	0.73	36%
150 × 150	10	10.11	21.48	9.99	1.38	53%

Tabella 1: Tempi di esecuzione [sec] per la superficie A

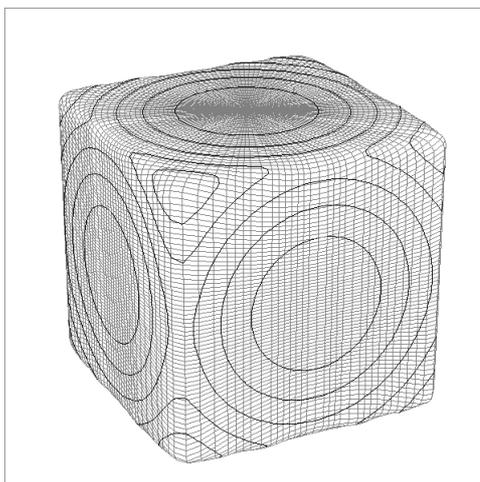


Figura 8: Superficie A: griglia 150 × 150, 5 linee di livello

<i>griglia</i>	<i>Linee Livello</i>	<i>Tempo PAM</i>	<i>Tempo PAM + LL</i>	<i>calcolo LL</i>	<i>hidden LL</i>	<i>totale LL</i>
50 × 50	5	2.58	3.68	0.77	0.33	30%
50 × 50	10	2.58	4.67	1.38	0.71	45%
100 × 100	5	5.99	9.02	2.63	0.40	34%
100 × 100	10	5.99	11.86	4.99	0.88	49%
150 × 150	5	10.50	16.26	5.20	0.56	35%
150 × 150	10	10.50	21.98	10.60	0.88	52%

Tabella 2: Tempi di esecuzione [sec] per la superficie B

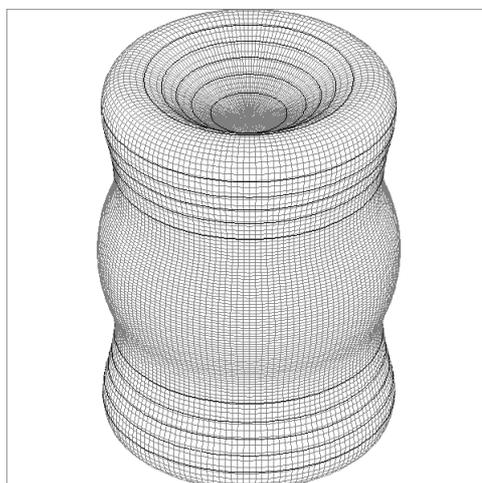
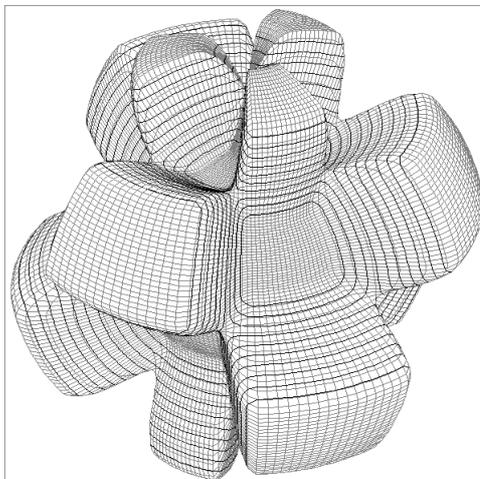


Figura 9: Superficie B: griglia 150 × 150, 5 linee di livello

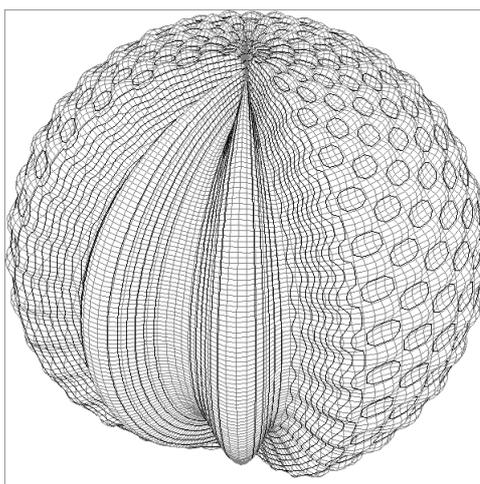
<i>griglia</i>	<i>Linee Livello</i>	<i>Tempo PAM</i>	<i>Tempo PAM + LL</i>	<i>calcolo LL</i>	<i>hidden LL</i>	<i>totale LL</i>
50 × 50	5	2.14	3.73	1.04	0.55	43%
50 × 50	10	2.14	5.06	1.81	1.11	58%
100 × 100	5	5.11	8.35	2.52	0.72	39%
100 × 100	10	5.11	11.44	5.04	1.29	55%
150 × 150	5	10.87	18.38	6.60	0.91	41%
150 × 150	10	10.87	25.16	12.48	1.81	57%

Tabella 3: Tempi di esecuzione [sec] per la superficie C

Figura 10: Superficie C: griglia 150×150 , 5 linee di livello

<i>griglia</i>	<i>Linee Livello</i>	<i>Tempo PAM</i>	<i>Tempo PAM + LL</i>	<i>calcolo LL</i>	<i>hidden LL</i>	<i>totale LL</i>
50×50	5	1.81	3.07	0.66	0.60	41%
50×50	10	1.81	3.89	1.10	0.98	53%
100×100	5	6.58	10.27	3.03	0.66	36%
100×100	10	6.58	14.88	6.66	1.64	56%
150×150	5	9.61	15.70	5.22	0.87	39%
150×150	10	9.61	22.68	10.89	2.18	58%

Tabella 4: Tempi di esecuzione [sec] per la superficie D

Figura 11: Superficie D: griglia 150×150 , 5 linee di livello

6. Conclusioni

In questo lavoro si è presentata una variante del metodo PAM per il contouring di superfici in coordinate sferiche.

Sia il metodo PAM che questa variante vengono proposti come tool efficienti nella modellazione con superfici spline in coordinate sferiche.

Riferimenti

- [1] Amoroso A., Casciola G., *Pseudopolar array mask algorithm for spherical coordinate grid surfaces*, Technical report Laboratory of Computer Science, University of Bologna. UBLCS-94-9 (May 1994).
- [2] Amoroso A., Casciola G., A hidden line algorithm for spherical coordinate functions, submitted to *ACM Transaction on Graphics*, (March 1994)
- [3] Suffern, K. Perspective views of polar coordinate functions, *Computer & Graphics*, 12, 3/4 (1988), 515-524.
- [4] Sánchez-Reyes J., Single-valued curves in polar coordinates (1990), *Computer-aided design*, 22,1,19-26;
- [5] Sánchez-Reyes J., Single-valued surfaces in cylindrical coordinates (1991), *Computer-aided design*, 23,8,561-568;
- [6] Sánchez-Reyes J., Single-valued spline curves in polar coordinates (1992), *Computer-aided design*, 24,6,307-315;