Modelli 3D

Categorie

di superficie

- **implicite** luogo di zeri di una funzione
- **parametriche** immagine di una funzione $\mathbb{R}^2 \to \mathbb{R}^3$
- **mesh poligonali** composizione di poligoni

volumetrici anche senza bordo (e.g. nuvola)

- **voxel** equivalente 3D delle immagini raster
- **CSG** operazioni insiemistiche su primitive volumetriche, rappresentazione ad albero (foglie: primitive, nodi interni: risultato di operazioni)
- **mesh poliedrali** tiicamenti di tetraedri e esaedri.

punti o segmenti

Mesh poligonali

Collezioni di poligoni adiacenti, cioè che condividono un lato. La risoluzione è il numero di poligoni usati; è spesso adattiva, cioè si utilizzano più poligoni dove servono. In CG si usano mesh di triangoli.

Una mesh 2-mainfold, orientata e chiusa delimita un volume.

Aperte e chiuse

Uno spigolo si dice *di bordo* se non è condiviso da due facce, altrimenti è *interno*. Un vertice è di bordo se fa parte di uno spigolo di bordo. Una mesh è *chiusa* se non ha spigoli di bordo.

Manifoldness

Una mesh è manifold se rappresenta effettivamente una superficie. In CG non è fondamentale che lo sia, ma è un requisito di molti algoritmi di geometry processing.

Una mesh è 2-manifold se si può far aderire un dischetto di gomma su qualsiasi punto della superficie (??)

- uno spigolo interno è 2-manifold se e solo se è condiviso da due facce;
- un vertice 2-manifold può essere solo in un fan (insieme di facce adiacenti che condividono un vertice).

Orientamento

L'orientamento di una faccia è definito dall'ordinamento dei vertici (orario o antiorario).

Una mesh 2-manifold in cui tutte le facce adiacenti hanno lo stesso orientamento si dice *orientata*, se può essere resa tale invertendo un sottinsieme delle facce allora è *orientabile*. Esempi di superfici non orientabili sono il nastro di Möbius e la bottiglia di Klein (hanno una sola faccia).

Cambiando l'orientamento cambia il verso della normale:

$$\frac{(v_1 - v_0) \times (v_2 - v_0)}{\|(v_1 - v_0) \times (v_2 - v_0)\|} = -\frac{(v_2 - v_0) \times (v_1 - v_0)}{\|(v_2 - v_0) \times (v_1 - v_0)\|}$$

Rappresentazione

polygon soup lista di poligoni, ciascun poligono è una lista di posizioni. Poco efficiente (vertici duplicati) e difficile da aggiornare (necessità di cambiare tutte le copie di un vertice);

mesh indicizzata una lista di vertici e una di poligoni, rappresentati come liste di riferimenti alla lista di vertici.

Spesso è utile associare dei valori alla mesh, per vertice (caso più comune, eventualmente con interpolazione) o per faccia (provoca discontinuità).

Coordinate baricentriche

Identificano un punto dentro un simplesso come combinazione dei suoi vertici. In un triangolo con vertici p_0, p_1, p_2:

$$p = w_0 p_0 + w_1 p_1 + w_2 p_2 \qquad w_0 + w_1 + w_2 = 1,$$

e i pesi si calcolano con:

$$w_0 = \frac{\text{area}(p, p_1, p_2)}{\text{area}(p_0, p_1, p_2)} \qquad w_1 = \frac{\text{area}(p, p_0, p_2)}{\text{area}(p_0, p_1, p_2)} \qquad w_2 = \frac{\text{area}(p, p_1, p_1)}{\text{area}(p_0, p_1, p_2)}$$

Sono usate per interpolare i valori (e.g. colore) dei vertici.

Normale

Vorremmo la normale alla superficie che stiamo approssimando con la mesh, ma solitamente abbiamo solo la tassellazione. Calcolare la normale di una faccia è facile:

$$n = \frac{(v_1 - v_0) \times (v_2 - v_0)}{\|(v_1 - v_0) \times (v_2 - v_0)\|},$$

ma usare questo valore per l'intera faccia evidenzia la tassellazione. Per un risultato più uniforme possiamo trovare le normali di ogni vertice e interpolarle all'interno della faccia.

Ci sono diverse definizioni di normale di un vertice, per esempio:

- media delle normali delle facce che condividono il vertice. Funziona solo se la mesh è sufficientemente regolare: più facce piccole contribuiscono maggiormente al risultato rispetto ad una faccia grande;
- la stessa media, ma pesata per l'angolo di ciascuna faccia sul vertice.

Questo approccio funziona dove la superficie è smussata (approssimabile con un piano intorno ai vertici); nel caso di spigoli nella superficie originale ciascun vertice deve avere più normali (creiamo copie o conserviamo le normali nelle facce). Per individuare questa situazione, fissiamo un *crease angle*: se l'angolo tra due facce adiacenti è maggiore del valore fissato, consideriamo la superficie non smussata in quel punto.

Marching Cubes

Algoritmo per la creazione di una mesh a partire da un campionamento regolare di un campo scalare $f : \mathbb{R}^3 \to \mathbb{R}$. Preso un cubo dalla griglia di campionamento, si stabilisce quali vertici siano all'interno ($f(v) < \alpha$) o all'esterno; in base a questo si seleziona una configurazione di triangoli da inserire nel cubo da una lookup table (16 elementi, si arriva a 2^8 tramite rotazione e riflessione). La posizione esatta dei vertici dei triangoli si trova con interpolazione lineare.

Problemi: non garantisce sempre coerenza (C0, 2-manifold, casi ambigui), correttezza (buona approssimazione della superficie reale), complessità (il numero di triangoli non dipendende dalla forma dell'isosuperficie) e qualità (triangoli arbitrariamente sottili) della mesh.

Superfici implicite

Luogo di zeri di una funzione $f : \mathbb{R}^3 \to \mathbb{R}$. Separano implicitamente esterno ($f(x) > 0$) e interno ($f(x) < 0$), perciò sono anche una rappresentazione volumetrica.

Normale

$$n_p = \frac{\nabla f(p)}{\|\nabla f(p)\|}$$

Operazioni booleane

I volumi delimitati da superfici implicite sono facili da combinare:

complemento $-f$

intersezione $\max(f_A, f_B)$

unione $\min(f_A, f_B)$

differenza $\max(f_A, -f_B)$

Metaballs

Creare una superficie a partire da sfere che tendono a fondersi quando sono vicine. Ciascuna sfera è definita da una funzione $f_i(x)$ con le seguenti caratteristiche:

- il valore dipende solo dalla distanza da un punto x_i;
- decrescente e liscia;
- $f(x_i) = 1$, $f'(x_i) = 0$;
- $f(x) = 0\ \forall x\ .\ \|x - x_i\| \geq R$, dove R è il *raggio di supporto*, e $f'(R) = 0$.

Definiamo poi f come la somma di tutte le f_i, e troviamo la isosuperficie associata ad un valore $0 < \alpha < 1$.

Per esempio,

$$f_i(x) = \begin{cases} 2\dfrac{r^3}{R^3} - 3\dfrac{r^2}{R^2} + 1 & r < R \\ 0 & r \geq R \end{cases}$$

Curve parametriche

Funzioni $f : \mathbb{R} \to \mathbb{R}^2$ o $\mathbb{R}^3$.

Spline

Curve polinomiali definite a tratti e di grado > 1. Sono definite in base a una sequenza di *punti di controllo*. Le curve che attraversano i loro punti di controllo sono dette *interpolanti*, quelle che ci passano vicino *approssimanti*.

$$f(x) = \sum_{i=0}^{n} p_i B_i(t) \qquad 0 \leq t \leq 1,$$

dove i p_i sono i punti di controllo e le B_i sono *funzioni di blending*.

Per un segmento, $B_0(t) = 1 - t$ e $B_1(t) = t$.

Curve di Bézier

Usano *polinomi di Bernstein* come funzioni di blending:

$$f(t) = \sum_{i=0}^{n} p_i B_{i,n}(t) \qquad B_{i,n} = \begin{cases} \dbinom{n}{i} t^i (1-t)^{n-i} & 0 \leq i \leq n \\ 0 & \text{altrimenti} \end{cases}$$

L'insieme $\mathcal{B}_n = \{B_{0,n}(t), \ldots, B_{n,n}(t)\}$ è detto base di Bernstein.

I polinomi di grado $n + 1$ si possono esprimere come combinazione lineare di polinomi di grado n:

$$B_{i,n+1}(t) = B_{i-1,n}(t)t + B_{i,n}(t)(1 - t),$$

che è utile per valutare la curva (*algoritmo di de Casteljau*).

Proprietà

- la somma degli elementi di $\mathcal{B}_n$ è 1, quindi le curve Bézier sono sempre contenute nell'involucro convesso dei punti di controllo;
- i punti di controllo per una curva di grado n sono $n+1$;
- il primo e ultimo punto di controllo sono interpolati;
- la curva è tangente al primo e ultimo segmento del poligono di controllo:

$$p_1 = p_1 + kf'(p_0) \qquad p_{n-1} = p_n + hf'(p_n);$$

- se un segmento attraversa il poligono di controllo n volte, interseca la curva al più n volte.

Esempi

quadratica $f(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$

cubica $f(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3$

Tangente

Derivata. Per curve di Bézier:

$$f'(t) = \sum_{i=0}^{n} p_i B'_{i,n}(t) \qquad B'_{i,n}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t)),$$

da cui si ricava:

$$f'(t) = n \sum_{i=0}^{n-1} B_{i,n-1}(t)(p_{i+1} - p_i).$$

quadratica $f'(t) = 2(1-t)(p_1 - p_0) + 2t(p_2 - p_1)$

cubica $f'(t) = 3(1-t)^2(p_1 - p_0) + 6t(1-t)(p_2 - p_1) + 3t^2(p_3 - p_2)$

Concatenazione

Nel punto di collegamento della concatenazione di curve ci sono diverse possibilità:

G0 le curve si toccano;

G1 hanno anche la stessa tangente unitaria;

G2 hanno anche la stessa derivata seconda.

In curve di Bézier,

G0 $v_n = u_0$;

G1 $v_{n-1}, (v_n = u_0), u_1$ sono collineari:

G2 G1 e $\|v_n - v_{n-1}\| = \|u_1 - u_0\|$.

Superfici parametriche

Funzioni $f : \mathbb{R}^2 \to \mathbb{R}^3$. Tangente: piano definito dai vettori $\frac{\partial f}{\partial s}(s,t)$ e $\frac{\partial f}{\partial t}(s,t)$.

Superficie di Bézier

$$f(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{n} p_{i,j} B_{i,n}(s) B_{j,n}(t) \qquad 0 \le s,t \le 1$$

- $(n+1)^2$ punti di controllo;
- interpolano sugli estremi $(0,0), (0,n), (n,0), (n,n)$;
- derivata:
$$\frac{\partial f}{\partial t}(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{n} p_{i,j} B'_{i,n}(s) B_{j,n}(t)$$
  analogamente per $\frac{\partial f}{\partial t}(s,t)$;
- la normale si trova con il prodotto vettoriale tra le derivate.

Conversione a mesh

Campionamento nel dominio dei parametri, e connessione dei punti risultanti con triangoli a griglia.

Voxel

Rappresentazione:

- griglia tridimensionale di booleani (vuoto/pieno) – spazio n^3;
- rappresentazione gerarchica (albero): lo spazio è diviso ricorsivamente in 8 parti, e la ricorsione si ferma quando il sottovolume è completamente pieno o vuoto, oppure dopo una profondità massima. $\mathrm{O}(n^2)$.

Nuvole di punti

Collezione non strutturata di punti, ottenuti tramite campionamento di una superficie. Per ciascuno si conserva posizione, normale e altri attributi. Realizzate con scanner 3D attivi (e.g. distanza con laser), structure from motion (tracciare un pixel tra più foto), ecc. Necessità di gestire sampling non uniforme (SfM riesce a tracciare solo pixel con colori particolari), rumore, allineamento dei risultati di più scan.

Sono efficienti, ma non hanno proprietà desiderabili: non definiscono una superrficie (difficile trovare volume o determinare se un punto è all'interno) né un grafo (non si conoscono i vicini di un punto).

Rendering

Tramite *surfel* (surface element): dischi con posizione, normale, colore e raggio. Facile da fare su GPU.