

Progettare Involucri di Forma Libera: (3) - Nuovi Strumenti per il Progetto

Davide Tonelli

Maggio 2013

1 Introduzione

I softwares CAD attualmente disponibili (vedi Rhinoceros[©] [3]) sono in grado di garantire la massima libertà espressiva alla creatività del progettista, mentre difficilmente riescono a tenere in considerazione le problematiche di carattere fisico, costruttivo e tecnologico tipiche delle costruzioni a forma complessa. Nei due articoli precedenti è stato invece evidenziato come una corretta progettazione delle forme libere passi necessariamente attraverso l'impiego di strumenti adeguati. Strumenti capaci di guidarne lo sviluppo dalla fase di concezione fino a quella di realizzazione, implementando in maniera naturale e continua tutti quei vincoli e quelle prescrizioni (di ordine geometrico, strutturale, tecnologico etc.), il cui mancato rispetto rende attualmente problematica la costruzione delle free forms.

In sintesi la *fattibilità a costi contenuti* deve essere il principio cardine dell'attività progettuale, ed è perciò proprio sullo sviluppo di strumenti atti al raggiungimento di questo obiettivo che si incentra l'attività del *G.R.I.F.F.*¹ dell'Università di Pisa, fondato e coordinato da Maurizio Froli. Attraverso una sinergia con alcuni membri del *V.C.G.*², sezione del CNR di Pisa, e con matematici topologici, il *G.R.I.F.F.* cerca di elaborare strumenti per la progettazione delle forme libere. La direzione di sviluppo è sia quella dell'*ottimale ingegnerizzazione dell'involucro*, sia quella della *progettazione integrata forma-struttura*.

2 Nuovi Metodi per Ingegnerezare l'Involucro

La volontà di instaurare una collaborazione tra il campo dell'Ingegneria Civile e quello della Computer Graphics, nasce dall'osservazione che i problemi di *ingegnerizzazione dell'involucro* sono almeno in parte affini a quelli affrontati nell'ambito della Computer Graphics. Di conseguenza le conoscenze e le tecniche precipue del suddetto settore risultano funzionali alla risoluzione delle problematiche in questione. In particolare sia il problema della razionalizzazione a posteriori - ovvero della pannellizzazione di una superficie di input, sia il problema dell'ottimizzazione a priori - ovvero della ricerca della superficie ottimale

¹G.R.I.F.F. - Gruppo di Ricerca sulle Free Forms. <http://www2.ing.unipi.it/griff/>

²V.C.G. - Visual Computing Lab.

sotto certe condizioni al contorno, sono tematiche abitualmente affrontate nella Computer Graphics. Il problema della *Pannellizzazione* è noto come *Surface Remeshing*, mentre la *Ricerca della Superficie Ottimale* è il tema centrale del *Surface Smoothing*.

2.1 Surface Remeshing

Il *Surface Remeshing* è una tecnica adoperata nella computer grafica per migliorare la qualità di una mesh. Il problema che affronta è il seguente: “data una mesh 3D, computare una nuova mesh i cui elementi soddisfino certi requisiti (definiti dall’utente) e che in contemporanea approssimi la superficie di input in maniera accettabile”. In gergo si parla di “*mesh quality improvement*”, con riferimento a parametri quali: densità, regolarità, dimensione, orientazione, allineamento, forma etc... delle facce della mesh.

Per quanto detto, è evidente che il tema della *pannellizzazione* di una superficie complessa può essere visto come un caso particolare di *remeshing*. La superficie di progetto, che generalmente è una NURBS (Non Uniform Rational Basis-Splines), può infatti essere banalmente convertita in una mesh³ grezza (al limite non strutturata), dopodiché il *pattern di suddivisione* prescelto scaturisce direttamente dall’applicazione dei metodi del surface remeshing.

Gli algoritmi di remeshing esistenti possono essere classificati in tre categorie fondamentali: “*greedy*”, *variazionali* e *incrementali*. Si assume implicitamente che tutti operino su meshes triangolari, e perciò anche su meshes esagonali in virtù della relazione di dualità che lega questi due patterns. Il remeshing quadrilatero viene invece affrontato con tecniche diverse.

Gli algoritmi “*greedy*” eseguono una sola operazione alla volta: ad esempio inserimento di vertici, rilocazione dei medesimi, rotazione dei lati etc... finché non viene raggiunto un determinato obiettivo.

³Questo è un processo triviale in quanto produce perdita d’informazione. Si intuisce perciò che il processo inverso sarà, qualora possibile, senz’altro non banale.

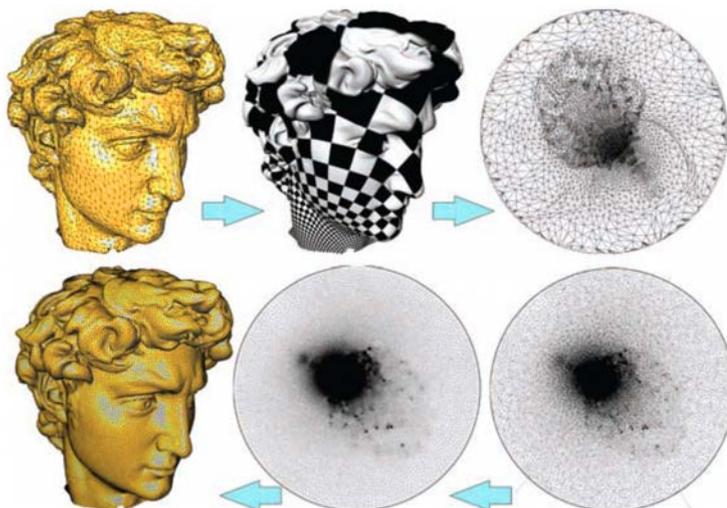


Figura 1: Procedura di Isotropic Remeshing con elementi triangolari [7].

Gli algoritmi *variazionali* vengono adoperati quando si vogliono ottenere meshes di qualità particolarmente elevata, ed operano applicando gli stessi strumenti di calcolo variazionale citati nella sezione 2.2. Ciò comporta la definizione di un criterio di ottimo per la mesh cercata: ad esempio è frequente uniformare gli angoli dei pannelli, i lati, l'area, oppure si può imporre una combinazione di questi ed altri aspetti etc... Spesso si vogliono creare pannelli identici: è il caso dell'*isotropic remeshing*, che si traduce in un problema di *isotropic point sampling* (ossia distribuzione di un insieme di punti sulla superficie, nella maniera più uniforme possibile). In questi casi è utile ricercare una “tassellazione centrale alla Voronoi” sul dominio di definizione della superficie $\Omega(u, v)$ (come spiegato nel precedente articolo). Dopodichè i baricentri delle celle di Voronoi così computate dovranno essere “mappati” sulla superficie $\mathbf{x}(u, v)$: questo processo introduce ovviamente delle distorsioni, cui si fa fronte mediante l'introduzione della “funzione densità variabile” $\rho(x)$ nel calcolo dei singoli baricentri nel dominio Ω (Figura 1).

Gli algoritmi *incrementali*, al contrario di quelli variazionali, non richiedono il calcolo di una parametrizzazione della superficie $\mathbf{x}(u, v)$. Il loro approccio è di tipo diretto in quanto prevede l'applicazione congiunta e ripetuta di una serie di operatori di remeshing locali. Una volta definito un criterio di ottimo quale ad esempio la lunghezza dei lati, l'algoritmo procede dividendo i lati lunghi, collassando quelli corti, ruotando i lati che generano vertici irregolari, rilocando i vertici in relazione ai vertici contigui etc... finchè il criterio di ottimo non è globalmente soddisfatto (Figura 2). Un esempio di pseudo-codice per questo tipo di algoritmo potrebbe essere il seguente:

```
remesh(target_edge_length)
low = 4/5 * target_edge_length
high = 4/3 * target_edge_length
for i = 0 to 10 do
    split_long_edges(high)
    collapse_short_edges(low, high)
    equalize_valences()
    tangential_relaxation()
    project_to_surface()
```

Infine tra gli algoritmi di *remeshing quadrilatero* quello di maggior interesse applicativo è il *curve based sampling*. E esso opera computando dapprima le direzioni e le curvature principali sulla superficie, dopodichè seleziona un set di linee di curvatura⁴ e le interseca. In corrispondenza delle suddette intersezioni vengono collocati i vertici della mesh quadrilatera. Osservando la Figura 3 si nota immediatamente come la mesh di output sia di tipo *quad-dominant* e non quadrilatera regolare: compaiono infatti sia “T-junctions” che vertici irregolari in corrispondenza dei punti umbilicali⁵. Si intuisce inoltre come il remeshing possa essere sia di tipo isotropo che anisotropo, in virtù della distanza che separa le linee di curvatura prescelte.

2.2 Surface Smoothing

Surface Smoothing, o anche *Surface Fairing*, è il nome che si dà in computer grafica alle procedure per l'individuazione, tra tutte le superfici rispettose delle

⁴curve che sono in ogni punto tangenti ad una delle due direzioni principali.

⁵punti in cui le curvature principali hanno valore uguale in modulo e segno: $k_1 = k_2$.

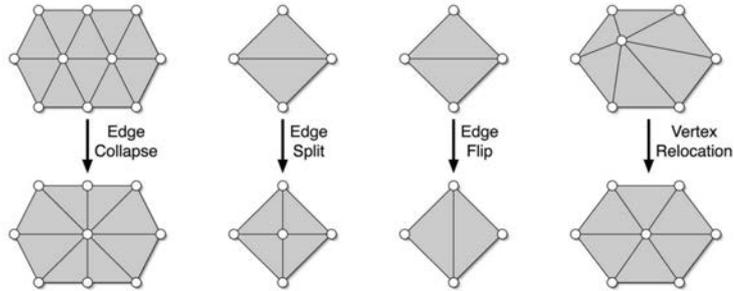


Figura 2: Operatori di remeshing locali [7].

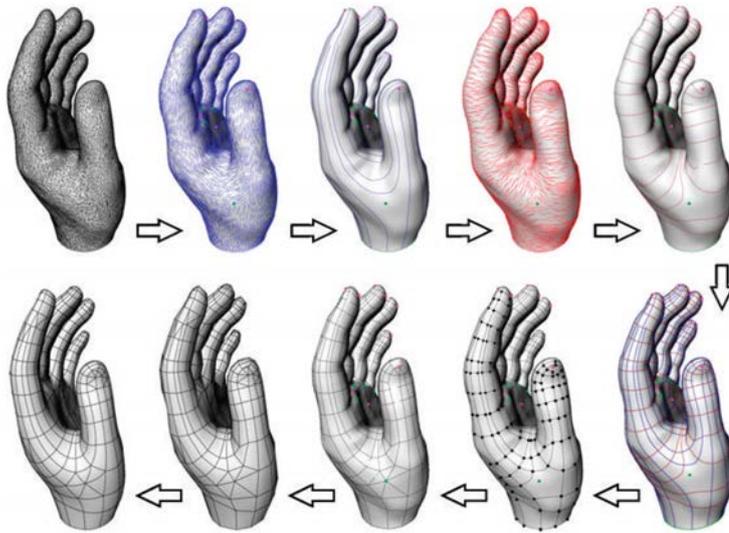


Figura 3: Quad Remeshing Anisotropo con *Curve based Sampling* [7].

condizioni ai limiti, di quella “più liscia possibile”. Queste procedure fanno un uso estensivo dei metodi di minimizzazione di funzionali tipici del calcolo delle variazioni⁶. I funzionali, formulati sotto forma di integrali, rappresentano delle energie il cui minimo individua proprio la superficie cercata.

Con riferimento a superfici continue, sono sostanzialmente tre i tipi di energia adoperati (nulla vieta però che ne possano essere definite altre ad hoc): l’“energia di membrana” (1) - cosiddetta perchè il suo minimo definisce superfici a membrana o di *area minima*, l’energia di piastra sottile” (2) - che computa superfici di curvatura minima, e l’“energia di *minima variazione di curvatura*” (3), riportate nel seguito.

(1) Energia di Membrana e sua linearizzazione (energia di Dirichlet):

$$E_M(\mathbf{x}) = \int \int_{\Omega} \sqrt{\det(\mathbf{I})} \, dudv \Rightarrow E_M(\mathbf{x}) = \int \int_{\Omega} \|\mathbf{x}_u\|^2 + \|\mathbf{x}_v\|^2 \, dudv$$

(2) Energia di Piastra sottile e sua linearizzazione:

⁶Il calcolo delle variazioni è quel campo dell’analisi che si occupa della ricerca dei punti estremali dei funzionali (ovvero di quelle funzioni il cui dominio è a loro volta un insieme di funzioni) e delle loro proprietà.



Figura 4: Esempi diversi, tutti validi, di *superficie più liscia possibile* [7].

$$E_{TP}(\mathbf{x}) = \int \int_{\Omega} k_1^2 + k_2^2 \, dudv \Rightarrow E_{TP}(\mathbf{x}) = \int \int_{\Omega} \|\mathbf{x}_{uu}\|^2 + 2\|\mathbf{x}_{uv}\|^2 + \|\mathbf{x}_{vv}\|^2 \, dudv$$

(3) Energia di Minima Variazione di Curvatura:

$$E_{MCV}(\mathbf{x}) = \int \int_{\Omega} \left(\frac{\partial k_1}{\partial \mathbf{t}_1}\right)^2 + \left(\frac{\partial k_2}{\partial \mathbf{t}_2}\right)^2 \, dudv$$

dove $\mathbf{x}(u, v)$ è una funzione vettoriale (mappa) definita sul dominio bidimensionale di definizione della superficie $\Omega(u, v)$, \mathbf{x}_{uv} è la derivata parziale mista della mappa rispetto alle variabili u e v , \mathbf{I} è la prima forma fondamentale, k_1 - k_2 e \mathbf{t}_1 - \mathbf{t}_2 sono rispettivamente le curvatures e le direzioni principali della superficie. Attraverso la teoria del calcolo delle variazioni⁷ si dimostra che minimizzare quei tre integrali equivale a risolvere tre equazioni alle derivate parziali⁸ del tipo:

$$(k) \quad \Delta^k \mathbf{x}(u, v) = 0$$

in $(u, v) \in \Omega$, con (k) numero dell'equazione integrale. In generale l'ordine k dell'equazione di Eulero-Lagrange ($\Delta^k \mathbf{x}(u, v) = 0$) determina la levigatezza massima al contorno, che è dimostrato essere pari a C^{k-1} (Figura 4).

Nell'ambito del *geometry processing* però non si ha mai a che fare con superfici continue e parametrizzate (come le NURBS ad esempio): le superfici sono tutte rappresentate attraverso mesh. Queste sono sostanzialmente un insieme di punti di coordinate note, cui vengono associate “informazioni di connettività” che servono a “contestualizzare il punto”, mettendolo in relazione con i punti vicini. Da un punto di vista computazionale compiere operazioni su una mesh risulta estremamente più semplice che eseguire le medesime azioni su una NURBS, ed è per questo che si preferiscono le prime alle seconde. In quanto “superfici discrete” però, per poterle utilizzare è indispensabile “discretizzare” la teoria finora introdotta al riguardo delle superfici continue. In pratica è necessario definire l'analogo discreto degli operatori matematici continui: nella fattispecie l'operatore di interesse è quello di Laplace-Beltrami⁹ (vedi [9]). Tale operatore è stato già definito (vedi [4], [5]) ed è una matrice quadrata \mathbf{L} di ordine n (n numero dei vertici della mesh).

Con tali premesse, risolvere le equazioni (1), (2) e (3) per una superficie discreta significa semplicemente risolvere tre sistemi di equazioni lineari analoghi del tipo: $\mathbf{L}^k \mathbf{x} = \mathbf{0}$, la cui soluzione restituirà proprio la posizione \mathbf{x} dei vertici della “*mesh più liscia possibile*” cercata.

⁷nella fattispecie è di interesse l'equazione di Eulero-Lagrange.

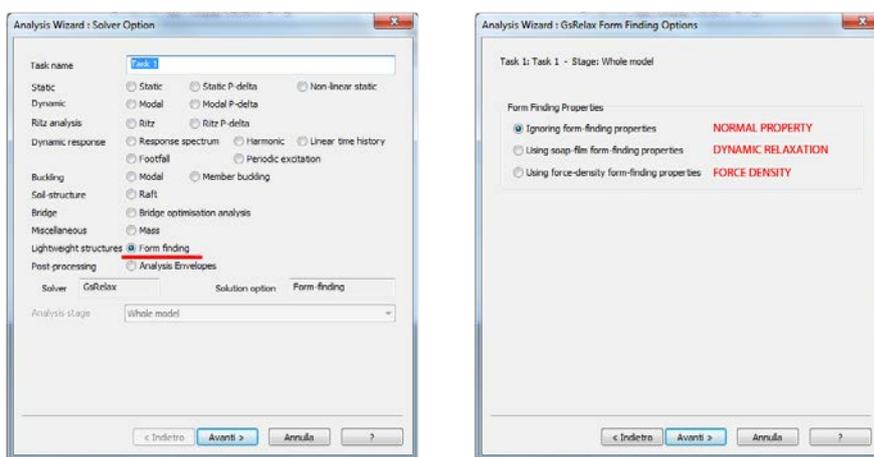
⁸detta per l'appunto equazione di Eulero-Lagrange.

⁹L'operatore di Laplace-Beltrami Δ_S estende il significato dell'operatore di Laplace Δ ($\Delta f(u, v) = f_{uu} + f_{vv}$) a funzioni definite su superfici S . Esso è definito come $\Delta_S \mathbf{x}(u, v) = -2H\mathbf{n}$, con H curvatura media della sup. e \mathbf{n} normale alla sup. nel punto di coordinate (u, v) .

3 Progettazione Integrata Forma-Struttura

Alla risoluzione dei problemi puramente geometrici succitati, si affianca poi ovviamente la problematica per così dire “classica” e prettamente “ingegneristica” della progettazione strutturale degli involucri. Come già sottolineato questi vengono realizzati come superfici discrete derivanti dalla pannellizzazione della superficie di progetto: si ricade così nell’ambito delle cosiddette “*grid-shells*”, ovvero di quelle strutture a guscio costituite da elementi portanti monodimensionali che vanno a costituire l’ossatura su cui poggiano i pannelli di chiusura portati. Strutture siffatte sono state oggetto di studi ingegneristici approfonditi a partire dalla metà del XX secolo [6] [19] [20] [12] [1] [11] [18], cosicchè ad oggi sono disponibili strumenti e metodi di progetto specifici: tra questi il *Transient Stiffness Method* [15] [13], il *Force Density Method* [18] [14] e il *Dynamic Relaxation Method* [17] [8] [2]. Tramite i suddetti metodi è possibile computare superfici discrete di tipo *funicolare*, ovvero superfici ottimali dal punto di vista statico che risultano essere in equilibrio mediante puri sforzi assiali di trazione (tenso-strutture) o di compressione (volte, gusci, *grid-shells*). Solo tramite un’accurata ricerca della forma ottimale infatti è possibile costruire strutture tanto ardite quali gli involucri a forma libera.

Attualmente questi metodi sono implementati in softwares di calcolo strutturale avanzati quali *GSA*® della Oasys (Figura 5), ma si iniziano già a vedere le prime implementazioni semplificate dei medesimi algoritmi anche in plug-in per softwares CAD quali Rhinoceros® (Figure 6 e 7). I software FEM sono infatti strumenti dedicati alla fase di analisi strutturale, e mal si prestano alle esigenze di flessibilità delle fasi preliminari di progettazione; i plug-in integrati in ambiente CAD quali ad esempio *RhinoVault* e *SMARTForm* invece si adattano perfettamente allo scopo.



(a) *Form Finding, Step 1.*

(b) *Form Finding, Step 2.*

Figura 5: Analysis Wizard del Software FEM *GSA*®.

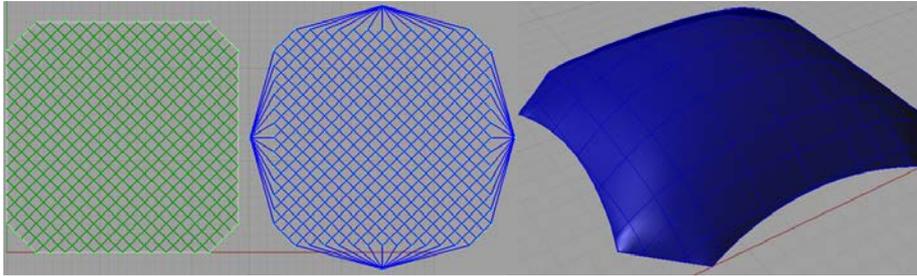


Figura 6: Superficie Funicolare, da plug-in per Rhinoceros[©] *Rhino Vault*.

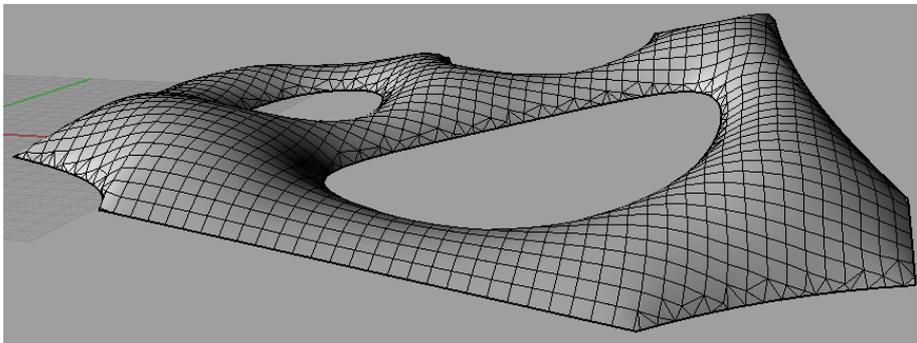


Figura 7: Superficie Funicolare, da plug-in per Rhinoceros[©] *SMARTForm*.

Riferimenti bibliografici

- [1] EN 1993-1-6. Eurocode 3: Design of steel structures - strength and stability of shell structures. Technical report, CEN, 2007.
- [2] J. H. Bunce A.S. Day. Analysis of cable networks by dynamic relaxation. *Civil Engineering and Public Works Review*, pages 383–386, 1970.
- [3] Robert McNeel & Associates. “NURBS Modeler for Windows - Rhinoceros”.
- [4] A. I. Bobenko and B. A. Springborn. “A discrete Laplace-Beltrami operator for simplicial surfaces”, 2006.
- [5] A. I. Bobenko and Y. B. Suris. *Discrete Differential Geometry (Graduate Studies in Mathematics)*. American Mathematical Society, December 2008.
- [6] S. Bonnemaïson. *LightWeight Structures in Urban Design*. PhD thesis, M.I.T., 1983.
- [7] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Levy. *Polygon Mesh Processing*. AK Peters, 2010.
- [8] A. S. Day. An introduction to dynamic relaxation. *The Engineer.*, Technical Contributor Section:220–221, 1965.

- [9] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. NJ: Prentice Hall, 1976.
- [10] Evolute GmbH. “Evolute Plugin for Rhinoceros”.
- [11] J.Lewis. *Tension Structures Form and behaviour*. Thomas Telford Publishing, 2003.
- [12] P. Singer D. Ströbel L. Gründig, E. Moncrieff. A history of the principal developments and applications of the force density method in germany 1970-1999. In *IASS-IACM*, 2000.
- [13] J. W. Leonard. *Tension Structures. Behaviour and Analysis*. 1985.
- [14] K. Linkwitz. Formfinding by the direct approach and pertinent strategies for the conceptual design of prepressed and hanging structures. *International Journal of Space Structures*, 2:73–87, 1999.
- [15] J. L. Meek. *Computer Methods in Structural Analysis*. 1991.
- [16] SMART Solutions Network. “SMARTForm Plugin for Rhinoceros”.
- [17] J. H. R. Otter. Computations for prestressed concrete reactor pressure vessels using dynamic relaxation. *Nuclear Structural Engineering*, 1:61–75, 1964.
- [18] H. J. Schek. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, 3:115–134, 1974.
- [19] J. Schlaich. On some recent lightweight structures. *Journal of International Association for Shell and Spatial Structures*, 43:69–79, 2002.
- [20] J. Knippers T. Bulenda. Stability of grid shells. *Computers and Structures*, 79, 2001.