

Estrazione algebrica di modelli (e proprietà) dalle immagini

Francesco Furiani[^], Claudio Paoluzzi^{*1}, Alberto Paoluzzi^{^2}

[^]Dipartimento di Matematica e Fisica, Università Roma Tre

^{*}Next Spa, Roma

Sommario

In this paper we discuss a novel algebraic representation of the topology of cell-decomposed domains, applicable to 2D and 3D images, finite element meshes, boundary representations of CAD models, as well as to NURBs, hierarchical assemblies and graphs. Our Linear Algebraic Representation (LAR) of topological models is based on well-known concepts from algebraic topology, namely on chain and cochain complexes supported by a cellular decomposition of the computational domain, and makes use of simple linear algebra operations, using a CSR (Compressed Sparse Row) representation of sparse binary matrices, well suited for GPGPU implementations on modern hardware. LAR defines a novel standard for model topology using a very general and simple representation scheme, with (co)chain complexes as mathematical models, represented by sparse binary matrices. LAR is based on chains, the domains of discrete integration, and cochains, the discrete prototype of differential forms, so naturally integrating the geometric shape with the physical properties. We show that (co)chain complexes, as formal models of geo-localised information, provide a kind of strongly-typed data objects with important descriptive, topological and algebraic properties. We also demonstrate here a simple, and topologically correct, extraction of complicated solid models from 3D images. This approach is being developed within the framework of the IEEE-SA P3333.2 — Standard for Three-Dimensional Model Creation Using Unprocessed 3D Medical Data.

Introduzione

Nelle scienze e nella tecnologia le applicazioni computazionali devono affrontare problemi geometrici di *complessità crescente*. La complessità delle informazioni geometriche deriva dal notevolissimo recente incremento *delle dimensioni, della diversità, e delle caratteristiche quantitative e qualitative* dei dati geometrici, sotto forma di immagini digitali, nuvole di punti da scanner 3D, rappresentazioni al contorno, rappresentazioni NURB, mesh di elementi finiti, immagini medicali 3D, etc. Tale complessità velocemente crescente delle informazioni e delle applicazioni geometriche, e gli obiettivi di *unificazione, scalabilità*, e supporto del calcolo distribuito *massivamente parallelo* spingono fortemente a ripensare i fondamenti del calcolo topologico e geometrico. In particolare, le emergenti applicazioni spaziali, le nano & bio tecnologie, e la medicina 3D, richiedono una nuova *convergenza tra metodi di sintesi e metodi di analisi* della forma, nelle discipline del computer imaging, nella computer grafica e nel CAD, in modo da rendere compatibili e interoperabili le rappresentazioni dei domini computazionali con le simulazioni fisiche.

Gli oggetti e le relazioni distribuite nel tempo e nello spazio sono rappresentabili come *complessi cellulari*. Ad esempio, le immagini digitali, le mesh di elementi finiti, le

¹C.P. performed the work while on sabbatical from consulting at Thales Alenia Space S.p.A, Rome.

²The research of A.P and F.F. was supported by a POC grant 2012/13 by SOGEI, the ICT company of the Italian Ministry of Economy and Finance.

rappresentazioni al contorno dei solidi, gli assiemi di componenti, i grafi, etc, sono tutti complessi di celle di varie dimensioni [1]. Tali partizioni di uno spazio in k -celle ($0 \leq k \leq d$) possono generare spazi lineari di k -catene, con elementi costituiti da combinazioni qualsiasi di k -celle, dove (a) insiemi contenenti una singola k -cella—“solitario” (singleton) di celle)—forniscono una base dello spazio, ovvero un insieme minimale di generatori, e dove (b) gli operatori lineari di bordo restituiscono la catena di bordo di qualsiasi catena data, per mezzo di una singola moltiplicazione SpMV (sparse matrix-vector).

Lo schema LAR (*Linear Algebraic Representation*) [1] è una rappresentazione semplice, generale, ed efficace di complessi di (co)catene [2], basata su una rappresentazione CSR (Compressed Sparse Row) [3] delle matrici caratteristiche di tali spazi lineari di (co)catene. LAR supporta tutte le strutture topologiche, è indipendente dalle dimensioni, e non è limitata a quelle regolari, per esempio ai complessi di dimensioni uniformi. LAR permette veloci test di validità della topologia dei modelli geometrici, anche generati da 3D scanner, o estratti da immagini 3D, utilizzando soltanto algebra lineare elementare, vale a dire moltiplicazioni di vettori e/o matrici sparse. Qualsiasi interrogazione circa l'occorrenza di relazioni tra catene e cocatene, di dimensioni uguali o differenti, si realizza per mezzo di singoli prodotti SpMV.

Background

Complessi cellulari

Tutti gli *oggetti geometrici* considerati in questo articolo sono *catene* in un *complesso cellulare* $\Lambda(X)$ ottenuto partizionando uno spazio (topologico) $X \subset E^d$. In estrema sintesi, un complesso cellulare è costituito di elementi di base chiamati *celle*, uniti insieme in modo appropriato [4].

Se vogliamo darne una definizione più formale, un *complesso cellulare* è uno spazio X di Hausdorff, ovvero uno spazio topologico nel quale punti distinti hanno intorni disgiunti, insieme ad una *partizione* $\Lambda = \Lambda_0 \cup \Lambda_1 \cup \dots \cup \Lambda_d$ di X in celle aperte (di varie dimensioni) che soddisfa alcune proprietà aggiuntive. Uno spazio topologico che ammette una triangolazione è necessariamente Hausdorff. In particolare lo sono gli spazi Euclidei.

Un sottospazio topologico compatto è una *cella convessa* se costituisce un insieme di soluzioni di uguaglianze e disuguaglianze affini. La *faccia* di una cella è la cella convessa ottenuta sostituendo delle disuguaglianze con delle uguaglianze. Una *faccetta* di una cella è una faccia definita da una sola uguaglianza. La *dimensione* n di una n -cella è quella del suo guscio affine, il sottospazio affine più piccolo in cui essa è contenuta.

Un *complesso di celle* convesse P , detto anche *complesso politopale*, è una unione finita di celle convesse tali che:

- (i) se A è una cella di P , allora lo sono anche le facce di A ;
- (ii) l'intersezione di due celle di P è una cella comune ad entrambe.

Un *complesso simpliciale* (rispettivamente, *cuboidale*) è un complesso politopale dove tutte le celle sono *simplessi* (rispettivamente, *cuboidi*). La dimensione di P è la dimensione massima delle celle di P . Lo r -scheletro P_r è il sottocomplesso formato dalle celle di dimensione $\leq r$. Lo 0-scheletro coincide con l'insieme $V(P)$ dei *vertici* di P .

Complessi politopali, simpliciali, cuboidali

Sono **complessi cellulari** caratterizzati da tipi di celle differenti. Nei primi le celle sono **politopi**, ovvero *insiemi convessi limitati*. Nei complessi simpliciali le celle sono **simplessi**, ovvero d -poliedri con $d + 1$ faccette (facce $(d - 1)$ -dimensionali) e $d + 1$ vertici. Nei complessi cuboidali le celle sono **cuboidi**, cioè (in generale, insiemi omeomorfi a) prodotti cartesiani di intervalli, ovvero d -poliedri con $2d$ faccette e 2^d vertici.

Un d -simpleso, o simpleso d -dimensionale, ha $d + 1$ *punti estremi* chiamati **vertici**. Uno spigolo (1-simpleso) ha $1 + 1 = 2$ vertici; un triangolo (2-simpleso) ha $2 + 1 = 3$ vertici; un tetraedro (3-simpleso) ha $3 + 1 = 4$ vertici, etc. Un d -cuboide ha invece 2^d vertici: un punto isolato (0-cuboide) ha $2^0 = 1$ vertici; uno spigolo (1-cuboide) ha $2^1 = 2$ vertici; un quadrilatero (2-cuboide) ha $2^2 = 4$ vertici; un esaedro (3-cuboide) ha $2^3 = 8$ vertici, e così via.

Supporto $|K|$ di un complesso cellulare K è l'unione dei punti delle sue celle. Una **triangolazione** di un politopo P è un complesso simpliciale K il cui supporto è $|K| = P$. Ad esempio, *una triangolazione di un poligono è una suddivisione di questo in triangoli*. Simplessi e cuboidi sono politopi. *Un politopo è sempre triangolabile*; ad esempio un quadrilatero si può suddividere in due triangoli, e un cubo in 5 oppure 6 tetraedri.

Le **celle** costituiscono i componenti elementari con cui si creano **modelli geometrici** a partire da "dati grezzi" (raw data), *ad esempio da nuvole di punti generati da scanner 3D*. Sotto alcune ulteriori specifiche sulla forma, i complessi simpliciali e cuboidali forniscono le **mesh** (o *griglie di calcolo*) per le **simulazioni** alle *differenze finite*, agli *elementi finiti* e ai *volumi finiti*.

Complessi di catene

Un *complesso di catene* in topologia algebrica è un *insieme di spazi lineari* (o vettoriali) C_k di oggetti (vettori) chiamati *catene*, che sono combinazioni lineari di *celle* in una partizione dello spazio, e tra cui sono definiti degli *operatori lineari di bordo* $\partial_k: C_k \rightarrow C_{k-1}$, che soddisfano delle condizioni topologiche del tipo “il bordo del bordo è vuoto”, $\partial_{k-1} \circ \partial_k = 0$. Si pensi ad esempio ad una palla 3D, il cui bordo è una superficie sferica, ovvero una superficie chiusa, cioè senza bordo.

Gli spazi C_k di catene possono essere definiti sul campo $Z_2 = \{0, 1\}$, e quindi una *catena* $\gamma \in C_k$ si ottiene combinando le *celle* della partizione con scalari binari. La rappresentazione coordinata $[\gamma]$ di una k -catena è data pertanto dal *vettore colonna* contenente gli scalari uno o zero che indicano presenza o assenza delle varie k -celle della base in essa.

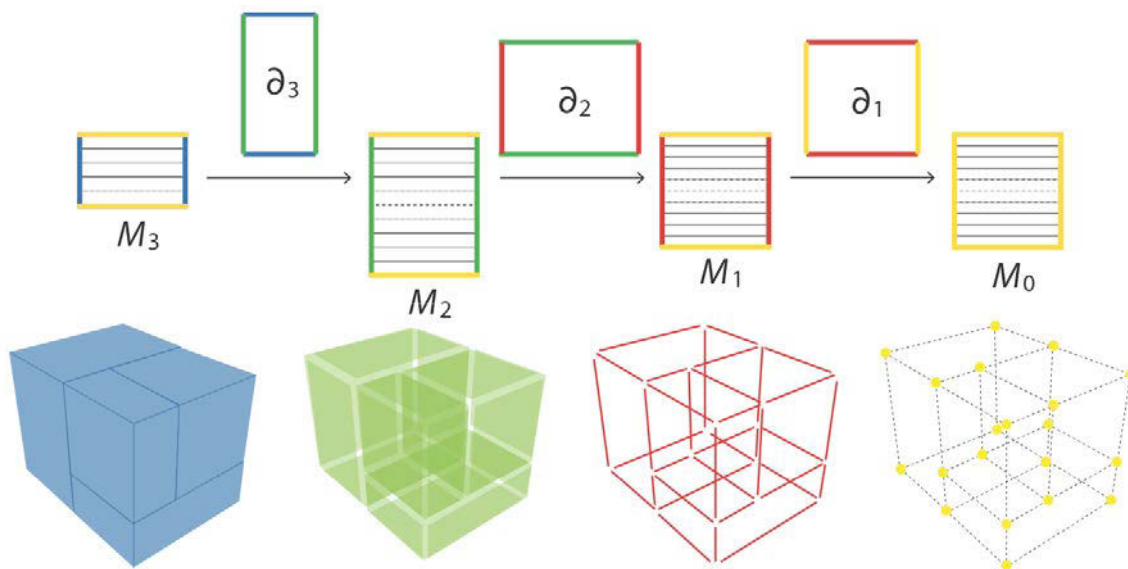


Figura 1: Complesso di catene definito da una partizione di uno spazio 3D. Le matrici M_3, M_2, M_1, M_0 , dette *matrici caratteristiche*, contengono per righe una base di celle delle varie dimensioni. Le matrici $\partial_3, \partial_2, \partial_1$ definiscono gli *operatori di bordo* in tali basi.

Le matrici caratteristiche M_k contengono per righe le rappresentazioni coordinate delle celle di dimensione k nella base di dimensione 0. Sono quindi una sequenza binaria contenente un elemento 1 per ogni vertice nel suo bordo e un elemento 0 per ogni vertice non appartenente al suo bordo. La Figura [fig:charmat] riporta le matrici caratteristiche M_3 (con 4 righe) ed M_2 (con 21 righe) dell'oggetto in figura. Si noti che il numero di colonne è costante, ed è pari al numero di vertici della partizione. Si noti anche che il numero di 1 in ogni riga è pari al numero di vertici incidenti sulla cella considerata. Ogni matrice M_k , che contiene per righe una base dello spazio C_k ($0 \leq k \leq d$), consente di generare qualunque k -catena (qualunque k -sottoinsieme discreto). Lo spazio lineare C_k ha dimensione pari al numero di righe della sua matrice caratteristica M_k .

Complessi di cocatene

Quando una quantità totale di una proprietà (fisica o astratta) è associata con le k -celle di un complesso di celle, assegnando un coefficiente a ciascuna k -cella, e i coefficienti possono essere sommati (in termini formali, appartengono ad un *gruppo commutativo*), la collezione

delle k -celle con coefficienti dallo stesso gruppo è chiamata *k-cocatena*. Essenzialmente le k -cocatene sono densità discrete di quantità contenute nelle k -celle della partizione Λ di uno spazio topologico X , come una immagine digitale o una mesh di elementi finiti, essendone k la dimensione.

Intuitivamente, una k -cocatena è una *funzione sopra le k -celle* di un complesso di celle $\Lambda(X)$. Nel limite di raffinamento infinito, le k -cocatene regolari tendono alle k -forme differenziali [5, 6].

Più in generale, in termini informatici, le cocatene offrono un meccanismo di rappresentazione *strongly-typed* di tutte le quantità associate con rappresentazioni discrete e combinatorie. E' facile mostrare che le k -cocatene sono i *funzionali lineari* agenti sulle k -catene e costituiscono pertanto lo *spazio duale* di quello delle k -catene. Esse possono essere usate come oggetti di base in un ricco e virtualmente illimitato calcolo delle proprietà fisiche del dominio considerato [7, 2].

$$\begin{array}{ccccccc}
 C^d & \xleftarrow{\delta^{d-1}} & C^{d-1} & \xleftarrow{\delta^{d-2}} & \dots & \xleftarrow{\delta^1} & C^1 & \xleftarrow{\delta^0} & C^0 \\
 \uparrow \cong & & \uparrow \cong & & & & \uparrow \cong & & \uparrow \cong \\
 C_d & \xrightarrow{\partial_d} & C_{d-1} & \xrightarrow{\partial_{d-1}} & \dots & \xrightarrow{\partial_2} & C_1 & \xrightarrow{\partial_1} & C_0
 \end{array}$$

Figura 2: Complesso di catene e cocatene, come sequenza di spazi lineari di catene C_k [sottoinsiemi del dominio di calcolo], e cocatene C_k [campi discreti definiti sul dominio], con operatori lineari di bordo ∂_k e cobordo δ^k , e con i vincoli $\partial_{k-1} \circ \partial_k = \delta^k \circ \delta^{k-1} = 0$. I simboli \cong denotano una gerarchia di isomorfismi tra k -catene e k -cocatene.

Gli operatori di cobordo $\delta^0, \delta^1, \delta^2$ di un complesso di cocatene su uno spazio 3D hanno una importante interpretazione come operatori differenziali discreti definiti su campi associati rispettivamente alle 0-catene (vertici), 1-catene (spigoli), 2-catene (facce), e corrispondono rispettivamente agli operatori discreti di gradiente, rotore e divergenza.

$$C^3 \xleftarrow[\text{div}]{\delta^2} C^2 \xleftarrow[\text{curl}]{\delta^1} C^1 \xleftarrow[\text{grad}]{\delta^0} C^0$$

Sussistono le seguenti relazioni tra operatori di bordo, cobordo e Laplaciano. In particolare, un operatore di cobordo $\delta^p: C^p \rightarrow C^{p+1}$ è semplicemente il trasposto dell'operatore di bordo $\partial_{p+1}: C_{p+1} \rightarrow C_p$ tra spazi di catene della stessa dimensione.

$$\boxed{\delta^p = \partial_{p+1}^\top} \quad \boxed{\Delta_p = (\delta^p)^\top \delta^p + \delta^{p-1} (\delta^{p-1})^\top}$$

Un semplicissimo algoritmo algebrico per il calcolo dell'operatore di bordo ∂_p ($1 \leq p \leq d$) e quindi per gli altri operatori differenziali discreti, è introdotto in [1]. Una sua implementazione efficiente in OpenCL [8] è descritta in [9].

Osservazione

La cocatena come *modello dell'informazione geo-localizzata* fornisce un tipo di “data object” con importanti proprietà descrittive, topologiche ed algebriche. Le cocatene possono fornire una sorta di ID collegato allo spazio descritto e alle sue proprietà, ovvero una sorta di *codice genomico* degli spazi o oggetti fisici, includendo la forma o il design del manufatto e le proprietà dei materiali costitutivi. Un ambiente di calcolo innovativo [7] basato sulle cocatene, al cui sviluppo stiamo collaborando, potrà operare con modelli di computazione che manipolano cocatene producendo cocatene. Una tale elaborazione sarà costruita come un mapping tra spazi di cocatene, che debba immergere, comprimere, raffinare, traslare, comporre, etc. informazioni utili codificate come cocatene a differenti livelli di astrazione, anche eventualmente derivate da immagini a differenti risoluzioni.

Cocatene valutate sulle catene

Riassumendo, dato uno spazio $\Lambda_d(X)$ suddiviso in d -celle connesse, con $|\Lambda_d| = k_d$, siamo dunque interessati alle d -catene, *insiemi* di d -celle, elementi dello spazio lineare C_d con scalari in. Di conseguenza, la *rappresentazione coordinata* di una cella $\lambda \in C_d$ è un *vettore binario* di lunghezza k_d e con un unico elemento pari ad 1.

Le *d-cocatene* sono gli elementi dello *spazio duale* $C^d := C_d^* = \{\varphi: C_d \rightarrow R\}$ dei funzionali lineari sulle catene. Per definizione, una *d-cocatena* è dunque una *rappresentazione funzionale* di un campo discreto su un d -complesso cellulare che partiziona uno spazio dato.

Ad esempio, per uno spazio discreto bidimensionale come un'immagine digitale, la base standard di C^2 è data dalle funzioni caratteristiche $\chi^i: C_2 \rightarrow R$ tali che $\chi^i(\lambda_j) = \delta_{ij}$, con $\delta_{ij} = 1$ (per $i = j$), e $\delta_{ij} = 0$ (per $i \neq j$). Una 2-cocatena è rappresentata come $\varphi = \sum_i \varphi^i \chi^i$, dove $\varphi^i(\lambda_j) \neq 0$ fornisce il valore di φ su λ_j , e $\varphi^i(\mu) = 0$ altrove. Il calcolo di una cocatena φ su una catena λ è dato da $\varphi(\lambda) := \langle \lambda, \varphi \rangle = \sum \lambda_i \varphi^i$, ed è il *prototipo discreto dell'operazione di integrazione*.

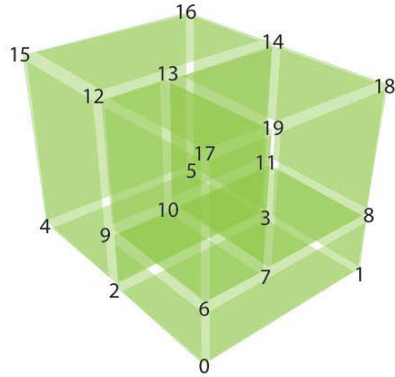
Rappresentazione Algebrica Lineare (LAR)

Definizione

Lo schema di rappresentazione [10] denominato LAR (*Linear Algebraic Representation*) [1] supporta tutte le costruzioni topologiche e le interrogazioni che si presentano in una tipica decomposizione cellulare dello spazio (immagini, griglie, bordi, ecc.).

Formalmente, LAR si basa su definizioni standard di topologia algebrica [4, 11]: nei complessi cellulari modulo 2, le d -catene sono *insiemi* di d -celle; la base standard dello spazio Z_2 -lineare C_d delle d -catene è fornita dai *solitari* delle d -celle; ciascuna d -cella è rappresentata da una funzione $C_d \rightarrow Z_2 C_0$, ad esempio da una *riga* di una matrice caratteristica binaria M_d . Ovviamente, ciascuna d -catena in C_d può essere generata da una combinazione Z_2 -lineare di righe di M_d .

La formulazione può essere estesa alle d -cocatene che rappresentano ogni possibile *campo* sulle *catene*. In questo contesto, il *bordo* è un operatore lineare ed il *cobordo* è il suo duale. Gli operatori di bordo e di cobordo forniscono gli strumenti computazionali previsti dalla versione discreta del teorema generalizzato di Stokes per l'integrazione di campi su domini d -dimensionali [5].



$$M_3 = \begin{pmatrix} 0011110001111111000 \\ 1111001111110000000 \\ 00000011011011000101 \\ 00000001101101100011 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1111000000000000000 \\ 1100001110000000000 \\ 1010001001000000000 \\ 0101000010010000000 \\ 0011110000000000000 \\ 0011000001110000000 \\ 0010100001001001000 \\ 0001010000010010100 \\ 0000110000000001100 \\ 0000001101100000000 \\ 00000011000000000101 \\ 00000010010010000100 \\ 0000000110110000000 \\ 00000001100000000011 \\ 00000001001001000001 \\ 00000000100100100010 \\ 00000000011011000000 \\ 00000000001101100000 \\ 0000000000011111000 \\ 00000000000011000101 \\ 00000000000001100011 \end{pmatrix}$$

Figura 3: Matrici caratteristiche delle 3-celle (parallelepipedi solidi della partizione) e delle 2-celle (faccette di bordo delle 3-celle)

Rappresentazione CSR e forma ridotta

Le matrici caratteristiche di un complesso di catene, ad esempio la matrice binaria che codifica l'incidenza delle d -celle sulle 0-celle, forniscono un comodo strumento per calcolare gli operatori di bordo (e di cobordo), e per rispondere alle interrogazioni sulle relazioni topologiche tra celle [1]. Come è facile immaginare, le matrici caratteristiche dei complessi che si utilizzano nelle applicazioni sono estremamente sparse, con sparsità crescente in $O(n^2)$, essendo n il numero delle celle, e possono essere efficientemente rappresentate in un formato standard CSR (Compressed Sparse Row) [12, 9], come illustrato in Figura 4a.

Il prodotto e la trasposizione di matrici CSR [12, 3], necessarie per calcolare il bordo, gli operatori di adiacenza e di incidenza tra questi spazi lineari, sono intrinsecamente efficienti, dato che il risultato della moltiplicazione matrice sparsa-vettore (SpMV), sparso anch'esso, ha dimensione lineare.

Sebbene per complessi di topologia qualunque LAR consista delle matrici $CSR(M_d), \dots, CSR(M_1)$, per due classi di complessi cellulari, importantissime nelle applicazioni, ovvero complessi *simpliciali* e *cuboidali*, dove le celle sono triangoli o tetraedri e quadrati o cubi, rispettivamente, LAR consiste di una *rappresentazione ridotta* (cfr. la Figura 4b), estremamente compatta.

Nella *LAR ridotta* una mesh 3D di tetraedri richiede uno spazio di memoria pari a $3V$ numeri reali per le coordinate dei vertici e $4Tet$ indici interi per la topologia, dove Tet è il numero dei tetraedri. Analogamente, una B-rep (rappresentazione al contorno) di un solido richiede $3Tria$ indici interi per la topologia, essendo ben noto che questo numero è pari a $2E$, con E il numero degli spigoli di bordo [13]. Tale rappresentazione è estremamente efficiente, considerando che la cosiddetta rappresentazione *Winged-Edge* di Baumgart (spesso usata come elemento di confronto) richiede memoria $8E$ per la sua tabella principale [13].

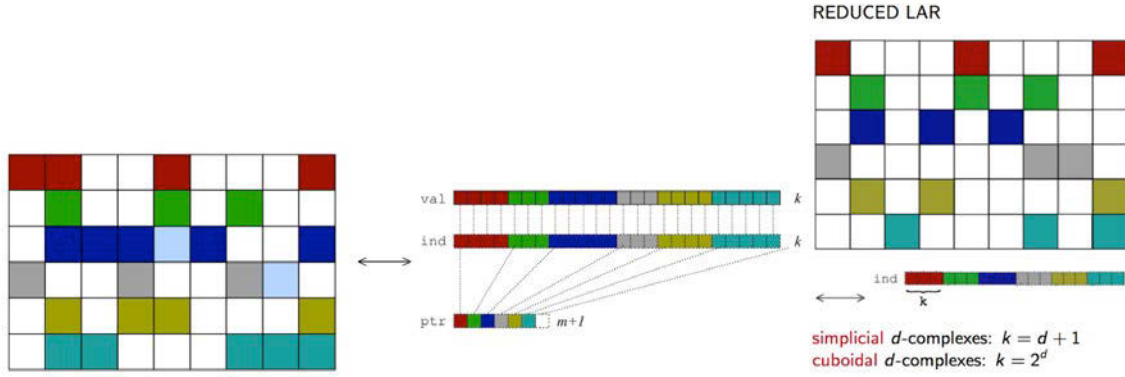


Figura 4: (a) Rappresentazione CSR (Compressed Sparse Row) di una matrice sparsa: richiede tre array, per memorizzare i valori, le colonne e l'inizio delle righe; (b) rappresentazione super-compatta di matrici binarie sparse relative a complessi simpliciali (triangoli, tetraedri) o cuboidali (rettangoli, esaedri).

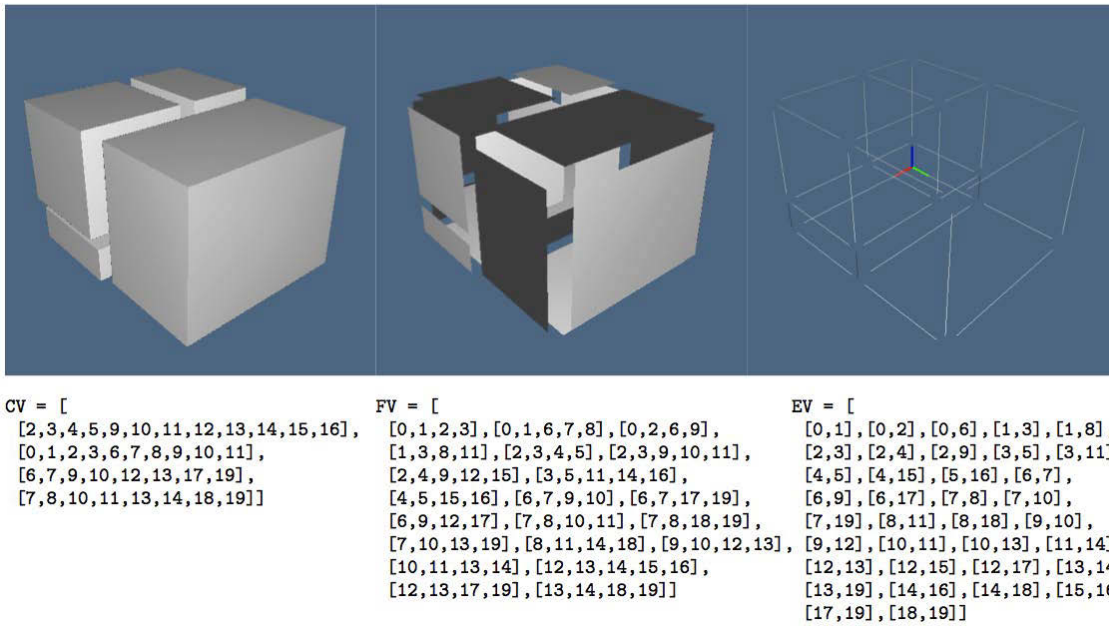


Figura 5: Rappresentazione compatta delle 3-, 2- e 1-celle del modello, fornita come relazioni celle-vertici, facce-vertici e spigoli-vertici. Si noti che gli array *CV* e *FV* corrispondono alle matrici M_3 ed M_2 .

Estrazione di (co)catene dalle immagini

La nostra implementazione [9] di LAR utilizza il modello computazionale data-parallel supportato dalle unità grafiche many-core, concepito per manipolare rapidamente la memoria e accelerare la creazione delle immagini da visualizzare, utilizzando lo standard industriale OpenCL [8]. Questo consente di scrivere programmi accelerati (kernel) che possano essere eseguiti su piattaforme eterogenee consistenti sia di unità centrali che di unità grafiche. Queste ultime sono presenti nei sistemi embedded, nei telefoni cellulari, nei personal computers, nelle workstations e nelle console di videogiochi. Ovviamente, le moderne GPU sono estremamente efficienti nel produrre computer-grafica, e la loro

struttura fortemente parallela è più efficiente delle CPU general-purpose per gestire quegli algoritmi in cui l'elaborazione di grandi blocchi di dati può essere effettuata in parallelo.

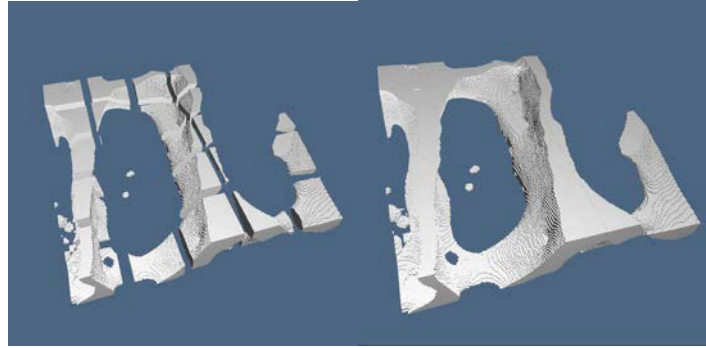


Figura 6: Dato che il collo di bottiglia della GPGPU è lo spostamento dei dati da memoria globale a locale, l'estrazione delle porzioni del modello è effettuata in parallelo utilizzando la stessa matrice $[\partial_3]$ per ogni sottoimmagine di ordine n^3 .

L'estrazione di un modello solido da una immagine è stata implementata utilizzando un approccio divide-et-impera (vedi Figura 6a). La partizione considerata è un insieme di n^3 voxel di immagine con $n \in \{64, 128\}$ che dipende dalla dimensione dello spazio di memoria della GPU.

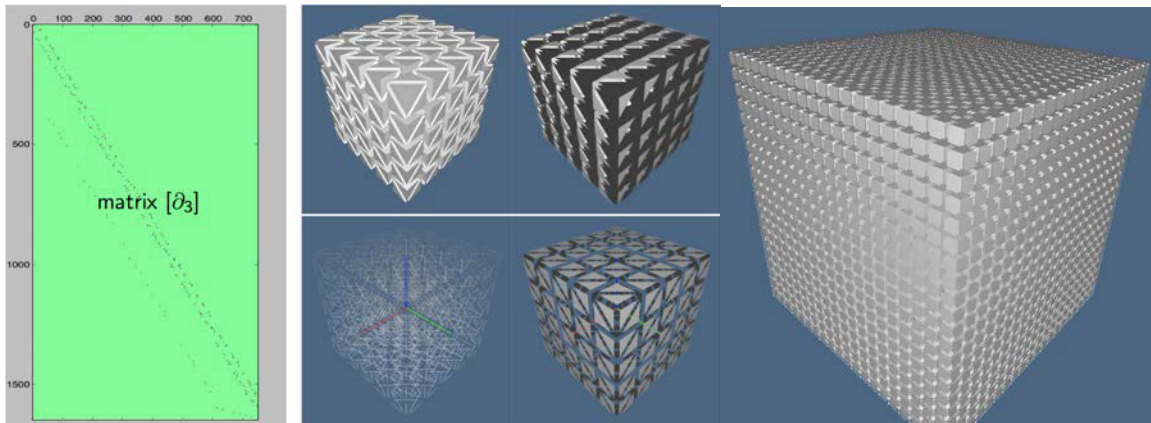


Figura 7: (a) Complesso simpliciale 3D: matrice di bordo $[\partial_3]$; complessi delle 3-, 2-, e 1-celle; 2-catena di bordo $[c_2]$ ottenuta dal prodotto SpMV con la 3-catena totale: $[c_2] = [\partial_3]\mathbf{1}$. (b) Complesso cuboidale corrispondente ad una porzione n^3 di un'immagine 3D, con $n = 20$; la corrispondente matrice CSR $[[\partial_3]]$ è precalcolabile, e caricabile *una tantum* nella memoria costante del *compute device*. Il bordo di qualunque catena (sottoinsieme) dell'immagine si ottiene con un solo prodotto SpMV, e fornisce un *modello solido* di topologia esatta.

La matrice $[\partial_3]$ dell'operatore di bordo definita sulla catena supportata da questa sottoimmagine è calcolata una sola volta, e memorizzata nella memoria costante della GPU, in modo che i bordi delle varie porzioni del modello contenuti nell'immagine possano essere *estratti in parallelo*, inviando ai *device work group* più insiemi di coordinate (stringhe di bit) per la rappresentazione delle catene di cui estrarre il bordo. Le coordinate per la rappresentazione dei bordi dei blocchi (cfr. la Figura 6a) sono calcolate effettuando molteplici operazioni SpMV in parallelo utilizzando la memoria locale dei *workgroup*. Un algoritmo *map-reduce* basato sul sort rimuove infine i contorni comuni a due blocchi.

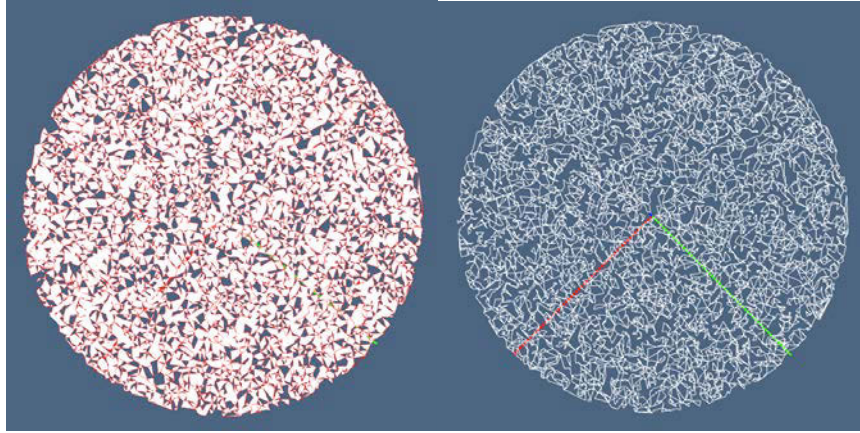


Figura 8: (a) Un poligono (una 2-catena) random nel cerchio unitario; (b) Il bordo della 2-catena, calcolato tramite una sola moltiplicazione SpMV. La stessa matrice $[\partial_2]$ viene utilizzata per estrarre il bordo di qualsiasi sottoinsieme del poligono, ovvero di qualunque 2-catena parziale di questo, moltiplicando $[\partial_2]$ per la rappresentazione coordinata (binaria) del sottoinsieme di 2-celle relativo.

In particolare, LAR è stato utilizzato in applicazioni biomediche che richiedono stringenti requisiti di performance nel manipolare grandi quantità di dati geometrici per operazioni topologiche come l'estrazione di un modello da immagini 3D (vedi Figura 9). I valori di densità nelle immagini medicali rappresentano campi scalari (cocatene) su complessi cellulari cubici, e LAR è utilizzata sia per garantire la correttezza topologica della segmentazione dell'immagine 3D, sia per estrarre modelli solidi dall'immagine (vedi Figura 9)³, la quale successivamente deve essere resa smooth utilizzando superfici parametriche interpolanti. Un interessante risultato di questo approccio è che *l'intera immagine* può essere partizionata in un insieme di cocatene associate agli intervalli di una suddivisione anche fine del campo scalare, incluso lo spazio interstiziale, in modo da fornire una mesh consistente e ben definita sia delle porzioni del modello di interesse che degli spazi esterni di varia natura.

La principale operazione che è stata implementata in OpenCL è ovviamente la moltiplicazione SpMV, per cui si sono utilizzati tre kernels, operanti su un insieme di vettori binari (rappresentazione coordinata di catene). Il risultato è prodotto nel formato CSR. Il primo kernel calcola il numero di elementi non-zero per ciascuna riga; il secondo scandisce il prefisso di ciascun elemento. Questo secondo kernel è stato implementato tramite la somma dei prefissi proposta in [14], con cache locale e blocchi per dividere le quote di lavoro tra unità distinte, secondo il paradigma *divide-et-impera*. I risultati parziali vengono raggruppati e diffusi a livello globale. Il terzo kernel costruisce e restituisce il risultato dell'operazione.

Nella libreria LAR, i kernel OpenCL per SpMV e SpMSpM (moltiplicazioni matrice sparsa-vettore e matrice sparsa-matrice sparsa) sono affiancati da numerosi altri componenti software. In particolare, sono necessarie varie computazioni lato host, che permettano di ricevere e manipolare l'ingresso, inviarlo alla GPU, verificare lo stato di esecuzione ed infine ritornare e visualizzare il risultato del calcolo. Inoltre, la libreria deve essere in grado di consentire il calcolo sia su macchine accelerate che non. Per conseguire questo obiettivo,

³Questo lavoro è stato eseguito nell'ambito del framework IEEE-SA Project P3333.2 - Standard for "Three-Dimensional Model Creation Using Unprocessed 3D Medical Data".

oltre ai moduli kernel, sono state concepite due differenti architetture: un servizio computazionale REST e un sistema pipelined single-host. Queste architetture sono state implementate in linguaggi differenti (javascript, java e C++) e utilizzando (varie combinazioni di) differenti tecnologie.

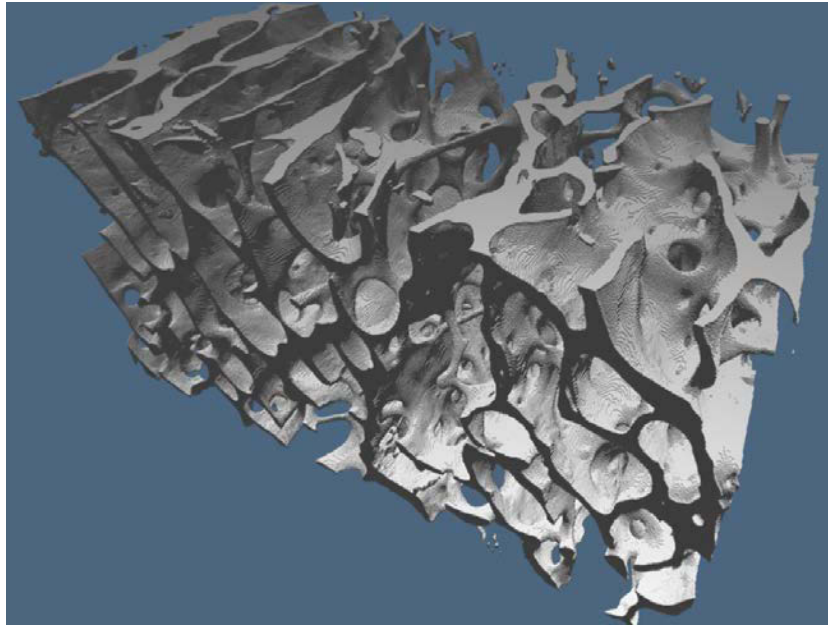


Figura 9: Modello solido di un campione di osso spugnoso estratto da un immagine 3D (μ NMR).

Applicazioni spaziali

Alcuni prodotti dell'industria dello spazio, come le immagini telerilevate da satellite e i sistemi di posizionamento globale (GPS), sono entrati da tempo nell'uso quotidiano. Un'altra tecnologia spaziale, il GPR (ground penetrating radar o radar sottosuperficiale), sta per imporsi sulla scena delle applicazioni spaziali di grande rilevanza economica. In particolare, i GPR sono già stati utilizzati per mappare le rovine di grandi siti archeologici e per studiare formazioni geologiche nel sottosuolo, come le riserve d'acqua nel sottosuolo marziano [15, 16]. La penetrazione sotto-superficiale delle microonde è maggiore, e il loro utilizzo più semplice e più economico rispetto ai metodi di indagine geologica tradizionali.

Alcune nuove tecnologie, come la risonanza dielettrica atomica (ADS, Atomic Dielectric Resonance), forniscono già ora servizi di geo-localizzazione, identificazione, mapping ed esplorazione delle risorse naturali nel sottosuolo. Perfino più promettente dello imaging satellitare della superficie terrestre, l'esplorazione con tecniche di telerilevamento degli strati superiori della crosta terrestre sta per produrre enormi quantità di dati digitali, che possono trovare impiego per numerosi nuovi prodotti e servizi per i cittadini, tra cui integrazione del mapping di ambienti interni/esterni, modelli di ricostruzione 3D di edifici, scoperta di siti archeologici inesplorati, mappe del sottosuolo, ispezione di strade e ponti, monitoraggio e servizi di allerta veloci per aree geologiche ad alto rischio, prevenzione dei disastri e ricostruzione, modelli geodinamici, controllo delle costruzioni abusive, operazioni di sicurezza in aree densamente abitate, e operazioni militari.

E' nostra opinione che le strutture topologiche descritte in questo articolo, arricchite con indici adeguati, saranno in grado sia di studiare la dinamica temporale dei fenomeni di utilizzo del suolo utilizzando stack 3D di immagini 2D, sia di analizzare efficientemente immagini 3D, producendo modelli geo-morfologici della crosta terrestre con partizioni di profondità variabile, anche allo scopo di supportare mesh di simulazione dei movimenti geodinamici. Ad esempio, i geologi sanno bene che per descrivere in modo più accurato lo sviluppo dei fenomeni di collisione orogenetici sono necessari dei modelli geodinamici pienamente 3D, caratterizzati da forte eterogeneità nelle condizioni al contorno e nelle proprietà litosferiche [17].

Gli autori stanno attualmente lavorando sullo smoothing e sulla trasformazione dei modelli estratti dalle immagini in mesh discrete di cocatene quoziente, allo scopo di ridurre drasticamente il numero di celle, caratterizzate da una densità costante della grandezza di interesse in ciascuna cella. In particolare, si stanno conducendo esperimenti con un insieme di algoritmi basati su matrici sparse derivate dal filtro di Taubin [18] e sui metodi di decomposizione di Morse-Smale della topologia derivati da [19].

Interrogazioni topologiche su modelli geometrici

Chiamiamo **modello geometrico** un *complesso cellulare*. Un modello 2D è costituito di **vertici** (V), **spigoli** (E , per "edges") e **facce** (F); un modello 3D di vertici, spigoli, facce e (3-) **celle** (C).

La **topologia** del modello riguarda le *relazioni binarie di incidenza* tra tali entità. Ad esempio, la relazione FE , sottoinsieme del prodotto cartesiano (insieme delle coppie ordinate) $F \times E$, specifica quali facce incidano su uno spigolo dato. Una **query topologica** di tipo $Fe \subset FE$ consiste nell'ottenere efficientemente una risposta alla domanda "quali facce incidono sullo spigolo e "? Nel piano abbiamo $3 \times 3 = 9$ relazioni topologiche binarie (in 3D ne avremo $4 \times 4 = 16$):

LAR trasforma le *relazioni tra entità topologiche* in **operatori lineari** tra catene (sottoinsiemi) di celle. Ad esempio, la relazione di **adiacenza** tra vertici VV diventa l'operatore VV , di matrice $[VV]$, calcolabile come prodotto della *matrice caratteristica* sparsa M_1 per la sua trasposta:

$$\begin{array}{lll} VV: C_0 \rightarrow C_0, [VV] = M_1^t M_1 & EV: C_0 \rightarrow C_1, [EV] = M_1 & FV: C_0 \rightarrow C_2, [FV] = M_2 \\ VE: C_1 \rightarrow C_0, [VE] = M_1^t & EE: C_1 \rightarrow C_1, [EE] = M_1 M_1^t & FE: C_1 \rightarrow C_2, [FE] = M_2 M_1^t \\ VF: C_2 \rightarrow C_0, [VF] = M_2^t & EF: C_2 \rightarrow C_1, [EF] = M_1 M_2^t & FF: C_2 \rightarrow C_2, [FF] = M_2 M_2^t \end{array}$$

LAR risponde a **qualunque** query topologica con un **prodotto** Rv tra la *matrice sparsa* R della relazione considerata (nell'esempio $R = [FE]$), e il *vettore colonna* v relativo all'elemento di query (nell'esempio e). Per rispondere a milioni o miliardi di query di tipo eF si calcolerà: (a) la matrice sparsa $R = [FE]$; (b) la rappresentazione coordinata w del sottoinsieme di spigoli di query; e infine (c) il prodotto Rw , calcolato in parallelo sugli acceleratori geometrici (usando OpenCL) come una singola operazione algebrica SpMV (prodotto matrice sparsa-vettore). Il *vettore colonna* risultante fornisce le **coordinate del sottoinsieme** (catena) di *facce incidenti* sugli spigoli di query.

Qualunque trasformazione su dati geo-localizzati può essere ricondotta ad una *sequenza di trasformazioni di (co)catene*. I risultati del calcolo potranno essere infine **visualizzati** sul dispositivo remoto dell'utente, sempre più **mobile**, anche nelle sue aspettative.

Dovrebbe essere facile per il lettore immaginare che questo approccio al calcolo topologico e geometrico si presta bene ad un tipo di elaborazione basata su **"open big raw data"** memorizzati in modo trasparente (**cloud computing**) e/o geograficamente distribuito, utilizzando i servizi locali o remoti di una infrastruttura di calcolo fortemente ottimizzata e parallela (**SaaS – software as a service**), ovvero gli strumenti di calcolo più performanti e pervasivi oggi disponibili.

Riferimenti bibliografici

- [1] A. DiCarlo, A. Paoluzzi, and V. Shapiro, "Linear algebraic representation for topological structures," in 2013 SIAM/ACM Joint Conference on Geometric and Physical Modeling, New York, NY, USA, 2013, SPM '13, Acm, To appear on Computer-Aided Design, Volume 46, January 2014, Pages 269–274.
- [2] A. DiCarlo, F. Milicchio, A. Paoluzzi, and V. Shapiro, "Chain-based representations for solid and physical modeling," Automation Science and Engineering, IEEE Transactions on, vol. 6, no. 3, pp. 454–467, 2009.
- [3] Aydın Buluc, and John R. Gilbert, "Parallel sparse matrix-matrix multiplication and indexing: Implementation and experiments," SIAM Journal of Scientific Computing (SISC), vol. 34, no. 4, pp. 170 – 191, 2012.
- [4] Allen Hatcher, Algebraic topology, Cambridge University Press, 2002.
- [5] Hassler Whitney, Geometric integration Theory, Dover, 1957.
- [6] Alain Bossavit, "Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism," IEE Proceedings, vol. 135, no. 8, pp. 493–500, November 1988.
- [7] Vadim Shapiro and William Regli, "The informatics of making," Tech. Rep., Ministry of Science & Technology Department of Science & Technology (DST), 2013, Innovation in Science Pursuit for Inspired Research (INSPIRE) Program.
- [8] Janusz Kowalik and Tadeusz Puźniakowski, Using OpenCL: Programming Massively Parallel Computers, IOS Press, 2012.
- [9] Francesco Furiani, "HPC acceleration of topological operators in the linear algebraic representation," M.S. thesis, Dept of Computer Science and Automation, University Roma Tre, Rome, 2013.
- [10] Aristides G. Requicha, "Representations for rigid solids: Theory, methods, and systems," ACM Comput. Surv., vol. 12, no. 4, pp. 437–464, 1980.
- [11] Jean-Claude Hausmann, Mod Two Homology and Cohomology, Université de Genève, 2012, Book Project.
- [12] Anton Lokhmotov, "Implementing sparse matrix-vector product in OpenCL," in OpenCL tutorial, 6th Int. Conference on High Performance and Embedded Architectures and Compilers (HiPEAC'11), 2012.
- [13] A. Paoluzzi, M. Ramella, and A Santarelli, "Boolean algebra over linear polyhedra," Comput. Aided Des., vol. 21, no. 10, pp. 474–484, Oct. 1989.
- [14] Shubhabrata Sengupta, Mark Harris, Michael Garland, and John D. Owens, "Efficient parallel scan algorithms for many-core GPUs," in Scientific Computing with Multicore and Accelerators, Jakub Kurzak, David A. Bader, and Jack Dongarra, Eds., Chapman and Hall/CRC Computational Science, chapter 19, pp. 413–442. Taylor and Francis, 2011.
- [15] G. Picardi, J. Plaut, O. Bombaci, D. Calabrese, M. Cartacci, A. Cicchetti, S. Clifford, P. Edenhofer, W. Farrell, C. Federico, A. Frigeri, D. Gurnett, T. Hagfors, E. Heggy, R. Huff, A. Ivanov, W.T.K. Johnson, R. Jordan, D. Kirchner, W. Kofman, C. Leuschen, E. Nielsen, R. Orosei, E. Pettinelli, R. Phillips, D.

Plettemeier, A. Safaeinili, R. Seu, E. Stofan, G. Vannaroni, T. Watters, and E. Zampolini, "Radar soundings of the subsurface of Mars," *Science*, vol. 310, pp. 1925–1929, December 2005.

[16] Claudio Paoluzzi, "Dual frequency algorithms for Sharad's SAR signal processing," M.S. thesis, InfoCom Dept, University La Sapienza, Rome, 2008, In Italian.

[17] S. Sol, A. Meltzer, R. Bu'rgmann, R. D. Van Der Hilst, R. King, Z. Chen, P. O. Koons, E. Lev, Y.P. Liu, P. K. Zeitler, X. Zhang, J. Zhang, and B. Zurek, "Geodynamics of the southeastern Tibetan Plateau from seismic anisotropy and geodesy," *Geology*, vol. 35, no. 6, pp. 563–566, June 2007.

[18] Gabriel Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1995, SIGGRAPH '95, pp. 351–358, Acm.

[19] Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci, "A practical approach to morse-smale complex computation: Scalability and generality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1619–1626, Nov. 2008.