



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI PROGETTAZIONE E GESTIONE INDUSTRIALE
DIPARTIMENTO DI PROGETTAZIONE AERONAUTICA



ADM ASSOCIAZIONE NAZIONALE
DISEGNO DI MACCHINE



*Embajada de España
Consejería de Educación
Italia*



ATTI

DEL

Seminario Italo-Español

**DISEGNO DI MACCHINE E
PROGETTAZIONE INDUSTRIALE**

**DIBUJO DE MÁQUINAS Y
DISEÑO INDUSTRIAL**

Facoltà di Ingegneria - Napoli
25 - 27 giugno 1996

Tecniche per la visualizzazione di immagini tomografiche

F. Bertolino, G. Gatto

Dipartimento d'Ingegneria Meccanica
Università di Cagliari
Piazza D'Armi - 09123 Cagliari

Introduzione

Tra le numerose tecniche per il controllo non distruttivo di componenti meccanici la Tomografia Computerizzata (TC) a raggi X si distingue per la sua capacità di localizzare in modo preciso il danneggiamento all'interno del volume. Con essa è possibile analizzare diversi tipi di materiali o verificare il corretto montaggio di un compressivo. La TC a raggi X consiste nella ricostruzione di una successione di sezioni trasversali dell'oggetto in esame a partire dalle sue radiografie acquisite da angoli diversi [1-4]. Gli autori hanno realizzato un microtomografo che consente di studiare oggetti di piccole dimensioni. A partire dalle informazioni relative ai vari strati del componente, un programma di visualizzazione è in grado di ricomporre l'intero volume e ne consente l'esame. Nel seguito, dopo una breve descrizione del sistema, saranno presentati gli algoritmi di visualizzazione implementati.

Descrizione del tomografo

Un sistema a raggi X (modello FAXITRON 43855A della HP, 10-120 kv e 3mA) adatto a radiografare oggetti di piccole dimensioni (contenuti in una sfera ideale di circa 20 cm. di raggio), è stato opportunamente strumentato ottenendo un prototipo di microtomografo di prima generazione (Fig.1). I raggi X viaggiano all'interno della camera in piombo dall'alto verso il basso; sulla base, a circa 80 cm dalla sorgente, è posizionato l'acquisitore puntuale costituito da un cristallo di silicio. Esso converte la radiazione trasmessa attraverso l'oggetto in un segnale elettronico adatto all'elaborazione automatica. Il campione da esaminare è disposto a sbalzo su un sistema di movimentazione e ciò ne limita il peso. Il motore lineare passo-passo consente spostamenti unitari di 25 μm mentre la rotazione è generata da un motore il cui passo è di 0.15 gradi. Un PC486 controlla il processo di acquisizione e consente di scegliere i passi di movimentazione ed i tempi d'integrazione. È inoltre possibile modificare il picco energetico emesso dal tubo. Il fascio di raggi X è collimato a monte dell'oggetto grazie ad un foro di 5 mm praticato su una lastra di piombo di 1 cm di spessore e a valle dell'oggetto grazie ad un foro di 0.4 mm praticato su una lastra di piombo di 2 mm. Non è stato possibile utilizzare fori più piccoli sia per la difficoltà di collimazione sia per la limitata quantità di energia che arriva all'acquisitore. Sono attualmente in corso una serie di misure per caratterizzare il tomografo rispetto ad alcuni parametri come la risoluzione spaziale, il contrasto, il rumore, etc. I dati così acquisiti sono inviati via Ethernet su una stazione Silicon Graphics per la ricostruzione tomografica e l'analisi dei risultati. Poiché i tempi di acquisizione sono molto lunghi (per registrare 200 dati lineari per 50 rotazioni sono necessarie circa 3 ore) non è possibile ottenere la mole di dati necessaria al buon funzionamento degli algoritmi di ricostruzione classici come il filtered back-projection o i metodi alle trasformate. In questo caso quindi è necessario ricorrere agli algoritmi algebrici [5-8], in grado di fornire ottimi risultati anche con un insieme di dati incompleto. Nel seguito sono descritti gli algoritmi implementati nel sistema per l'analisi dei risultati.

Algoritmi di visualizzazione

Il processo di ricostruzione tomografica si conclude con la generazione di un reticolo 3D in cui ad ogni nodo è associato un valore scalare che rappresenta il livello di opacità ai raggi X. La visualizzazione volumetrica è un metodo di selezione e rappresentazione delle informazioni significative contenute all'interno di questo reticolo, che può essere costituito da una mole gigantesca di dati (decine di megabytes). Il suo obiettivo consiste nello scrutare all'interno del volume per vedere ciò che normalmente è nascosto. L'algoritmo algebrico di ricostruzione considera dataset volumetrici regolari (intervalli costanti sui tre assi) di tipo *cell based*, in cui la generica informazione è associata ad un punto dello spazio; in questo senso si pone in alternativa agli algoritmi che forniscono dati di tipo *voxel based*, in cui l'informazione è considerata costante in un intorno del punto assunto come riferimento. Isolati nel reticolo 8 nodi contigui a formare un cubo, si ipotizza una variazione trilineare della funzione all'interno del volume e questo fatto viene utilizzato anche in fase di visualizzazione. La struttura del dataset è un parametro fondamentale in quanto una qualche forma di strutturazione consente d'individuare agevolmente, per ogni punto dello spazio d'interesse, gli elementi del dataset che "insistono" su quel punto. Nel nostro caso, in fase di ricostruzione tomografica si è fatto ricorso ad un mesh strutturato di forma cilindrica, mentre in fase di visualizzazione, per accelerare i calcoli, abbiamo mappato i valori calcolati, su un reticolo regolare in forma di parallelepipedo.

Le tecniche di visualizzazione possono dividersi in due famiglie [9, 10]: quelle che isolano delle superfici all'interno del dataset e provvedono alla loro visualizzazione, e le tecniche di "volume rendering", che visualizzano direttamente l'intero volume di dati.

Ricostruzione di superfici

Un approccio classico consiste nell'individuare una sequenza di isolinee sulle sezioni del volume, da connettere successivamente con dei triangoli [11,12]. Queste tecniche non forniscono una risposta corretta quando la superficie da ricostruire si divide in rami, specie se la distanza fra le sezioni è molto più grande della distanza tra i punti appartenenti alle sezioni stesse.

Scelta una soglia, la tecnica a *cube* consiste invece nel dividere i voxels tra interni ed esterni alla superficie che si desidera rappresentare [13]. Considerato ogni voxel come un piccolo cubo, si visualizza la superficie disegnandone le facce. I risultati sono soddisfacenti solo quando il numero di voxel è molto elevato.

L'algoritmo che si ritiene più interessante e che si è deciso di implementare prende il nome di *marching cubes* [14]. Stabilita la soglia da rappresentare, si esamina il mesh considerandolo come una successione ordinata di cubi ad ognuno dei quali si associa un indice, calcolato assegnando ad ogni nodo un numero, pari a zero o uno, secondo che esso si trovi dentro o fuori dalla superficie. Sebbene esistano 256 combinazioni possibili, si può dimostrare che esistono solo 15 configurazioni topologicamente distinte. A partire da queste, le altre si possono ottenere attraverso semplici operazioni di rotazione e di inversione. Una lookup table lunga 256 posizioni associa ad ogni indice una serie di triangoli, i vertici dei quali si trovano calcolando l'intersezione della superficie da visualizzare con il mesh. Per produrre un'immagine realistica, è necessario calcolare i gradienti della funzione nei vertici dei triangoli: questo implica la presenza contemporanea in memoria di solo quattro sezioni del volume, per formare tre strati di cubi. Al termine di queste operazioni si è in possesso di una lista di triangoli con le coordinate dei rispettivi vertici e i gradienti della funzione. Un programma di visualizzazione è a questo punto in grado di mostrare la superficie con gli algoritmi di Gouraud shading, z-buffer, animazione, etc [15].

L'algoritmo illustrato può essere utilizzato anche per la rappresentazioni di sezioni trasversali del dataset. Selezionato il piano di taglio, i nodi vengono classificati in base al semispazio di appartenenza: per il resto il procedimento è identico.

Volume rendering

La rappresentazione di isosuperfici è interessante quando si desidera isolare uno o più oggetti interni al volume. Dovendo applicare un operatore di soglia, questo tipo di rappresentazione è piuttosto delicata e può condurre ad errate interpretazioni. Con la visualizzazione di isosuperfici si perde inoltre la visione complessiva del dataset. Un metodo più sofisticato consiste nell'applicare al volume in esame le tecniche di ray tracing [16]: a fronte di una mole di calcoli non indifferente, è possibile osservare l'intero volume come se esso fosse semi-trasparente, una sorta di radiografia dell'oggetto. Una volta elaborata una serie di tali immagini da angoli diversi (il che può comportare alcune ore di calcolo), le si memorizza sul disco del calcolatore; successivamente è possibile rividerle in rapida sequenza con tecniche di animazione: l'esame in movimento fornisce un'idea piuttosto precisa dell'intero volume. L'algoritmo che abbiamo implementato è una versione semplificata di quello proposto da Upson e Keller [17]. Per ogni pixel dello schermo si traccia un raggio parallelo alla direzione di osservazione e si calcolano i punti d'ingresso ed uscita dal volume in esame. Sul segmento così isolato, si selezionano ad intervallo costante, un numero di punti pari al doppio dei cubi elementari intercettati. Per ciascuno di detti punti si individua il cubo elementare di appartenenza: a partire dai suoi valori nodali, ipotizzando un'interpolazione trilineare, si calcola il valore della funzione nel punto ed il suo gradiente. Il valore della funzione diventa l'input per 4 funzioni di trasferimento (rosso, verde, blu e opacità) scelte dall'utente. Ciò consente di isolare all'interno del volume zone omogenee di particolare interesse, pur continuando ad osservare l'intero volume. Si è ipotizzato un modello di illuminazione empirico molto semplificato, in cui si utilizza una sola fonte d'illuminazione a raggi paralleli e in cui sono assenti i termini d'illuminazione speculare ed ambiente:

$$I(\lambda, s) = K_d(\lambda) * (N * L) \quad \text{in cui} \quad \lambda = \text{rosso, verde, blu}$$

$I(\lambda, s)$ indica l'intensità luminosa percepita (funzione della lunghezza d'onda e del punto osservato), $K_d(\lambda)$ è il coefficiente di diffusione, N è il gradiente della funzione nel punto ed L è la direzione d'illuminazione. Il coefficiente di diffusione viene calcolato come segue:

$$K_d(\lambda) = D(\lambda) * LUT(\lambda, f(s)) \quad \text{in cui} \quad \lambda = \text{rosso, verde, blu}$$

in cui $D(\lambda)$ indica il coefficiente di diffusione del materiale, $LUT(\lambda, f)$ è la funzione di trasferimento ed $f(s)$ è il valore della funzione nel punto s lungo il raggio. I valori di $I(\lambda, s)$, calcolati nei punti selezionati lungo i raggi di osservazione, sono integrati come segue:

$$I(\lambda) = \sum [a(d) * LUT(\text{opacità}, f(s)) * I(\lambda, s)] * \Delta s$$

in cui $I(\lambda)$ indica l'intensità luminosa nel pixel dello schermo, $a(d)$ è un coefficiente di attenuazione (funzione della distanza d del punto dall'osservatore), Δs è il passo dei punti d'integrazione. La sommatoria termina quando sono esauriti i punti d'integrazione o quando l'opacità ha raggiunto il massimo.

Esempi

Gli algoritmi di visualizzazione sono stati implementati su una stazione INDIGO2 R4400 della Silicon Graphics con 32MB di RAM. Sono stati testati su mesh 3D sintetici costituiti da 100X100X100 nodi e su dati reali provenienti dalla ricostruzione di un cilindro in PVC di 2.5 cm di spessore, contenente tre fori passanti con assi paralleli a quello dell'oggetto, in uno dei quali si è impegnato un chiodo d'acciaio di 2.5 mm. La Fig.2 rappresenta la sezione trasversale di un mesh sintetico: il piano è costituito da circa 30000 triangoli ed il tempo di calcolo è stato di 17 secondi. La Fig.3 ne rappresenta una isosuperficie: il tempo di calcolo per isolare i suoi 180000 triangoli è stato pari a 31 secondi. La Fig.4 è il risultato dell'algoritmo di "volume rendering" che ha richiesto 41 secondi di calcolo. La Fig.5 rappresenta il livello di opacità ai raggi X di una sezione del provino in PVC. Il chiodo in esso contenuto è molto evidente, ma gli artefatti causati principalmente dalla scarsità dei dati acquisiti (circa 2000), nascondono i due fori.

Conclusioni

La caratterizzazione del microtomografo è ancora in corso. Per il momento, sono stati analizzati dei provini in PVC di forme e dimensioni appropriate per controllare la risoluzione spaziale, il contrasto, il rumore, etc.

Con l'algoritmo di *marching cubes*, i tempi e la memoria necessari sono legati evidentemente non solo alla dimensione del mesh, ma anche alla quantità di triangoli selezionati. Grazie al fatto che l'algoritmo richiede in memoria solo quattro sezioni per volta, è possibile analizzare dataset di grandi dimensioni. I risultati sono in ogni caso molto soddisfacenti.

Per ciò che riguarda l'algoritmo di *volume rendering* i tempi di calcolo dipendono dal numero di pixels sullo schermo su cui l'oggetto sarà proiettato. Esso può essere implementato in modo da richiedere in memoria solo i dati relativi ai cubi di volta in volta attraversati dai raggi di osservazione. Ciò comporta un accesso frequente alla memoria e rallenta il calcolo, ma consente di analizzare dataset molto grandi. La scelta delle funzioni di trasferimento consente di isolare zone di particolare interesse, ma richiede da parte dell'utente una certa esperienza. La lettura dei risultati è fortemente semplificata dalle tecniche di animazione.

Nella sua attuale implementazione, il programma di visualizzazione non prevede alcuna ricerca di coerenza spaziale e l'informazione è organizzata come una lunga sequenza di sezioni. Per le risoluzioni spaziali che ci siamo prefissi di raggiungere, questo tipo di organizzazione non è adatto in quanto richiede molta memoria; in futuro si prevede di inserire una fase di preanalisi dei dati che, attraverso una loro segmentazione, conduca ad una organizzazione più efficiente.

Bibliografia

- [1] Hall E.L.: *Computer Image Processing and Recognition*, Computer Science and Applied Mathematics, Academic Press, 1979.
- [2] Natterer F. (Editor), *Mathematical Aspects of Computerized Tomography*, Proceedings 1980, Lecture Notes in Medical Informatics, Springer-Verlag, 1981.
- [3] Rosenfeld A., Kak A.C., *Digital Picture Processing*, Computer Science and Applied Mathematics, Academic Press, 1982.
- [4] Natterer F., *The Mathematics of Computerized Tomography*, John Wiley & Sons Ltd. and B.G.Teubner, Stuttgart, 1989.
- [5] Gordon R., A Tutorial on ART (Algebraic Reconstruction Techniques), *IEEE Transactions on Nuclear Science*, Vol.NS-21, pp.78-93, June 1974.
- [6] Rangayyan R.M., Dhawan A.P., Gordon R., Algorithms for limited-view computed tomography: an annotated bibliography and a challenge, *Applied Optics*/Vol.24, No.23/ pp.4000-4012, 1985.
- [7] Verhoeven D., Multiplicative algebraic computed tomographic algorithms for the reconstruction of multidirectional interferometric data, *Optical Engineering*/ Vol.32 No.2, pp. 410-419, 1993.
- [8] Bertolino F., Ginesu F., Savona V., Applicazione della tecnica tomografica nella sperimentazione meccanica: elaborazione e rappresentazione 3D dei risultati, Atti del IX Convegno Nazionale ADM, pp.407-417, Caserta 1995.
- [9] Hohne K.H., Fuchs H., Pizer S.M. (Editors), *3D Imaging in Medicine: Algorithms, Systems, Applications*. NATO ASI Series, Springer-Verlag, 1990.
- [10] Kaufman A., Volume Visualization, IEEE Computer Society Press Tutorial, 1991.
- [11] Huijsmans, D.P., Jense G.J., Representation of 3D objects, reconstructed from series of parallel 2D slices, NATO ASI Series, Vol.F40, Theoretical Foundations of Computer Graphics and CAD, pp.1031-1037, Springer-Verlag, 1988
- [12] Boissonnat, J.D., Shape reconstruction from planar cross-sections. *Computer Graphics & Image Proc.* 44, pp.1-29, 1988
- [13] Herman G.T., Liu H.K., 3-D Display of Human Organs from Computed Tomograms, *Comp. Graph.& IM. Proc.* Vol.9(1), 1979.
- [14] Lorensen W.E., Cline H.E., Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer graphics* (Proc. Siggraph), Vol.21, No.4, pp.163-169, Aug.1987.
- [15] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., *Computer Graphics: Principles and Practice*, Addison-Wesley 1990.
- [16] Glassner A.S. (editor), *An Introduction to Ray Tracing*, Academic Press, 1989.
- [17] Upson C., Keller M., V-Buffer: Visible Volume Rendering, *Computer Graphics*, Vol. 22, N.4. pp.59-64, August 1988.

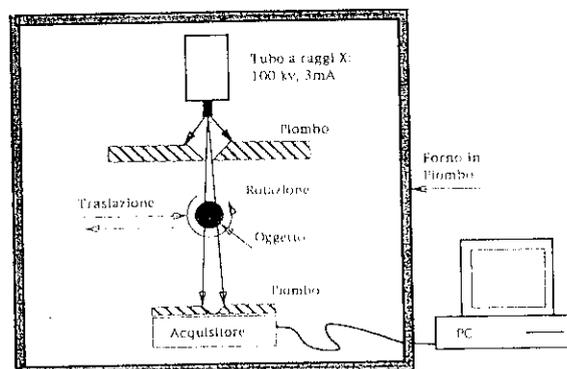


Fig.1 - Apparato sperimentale

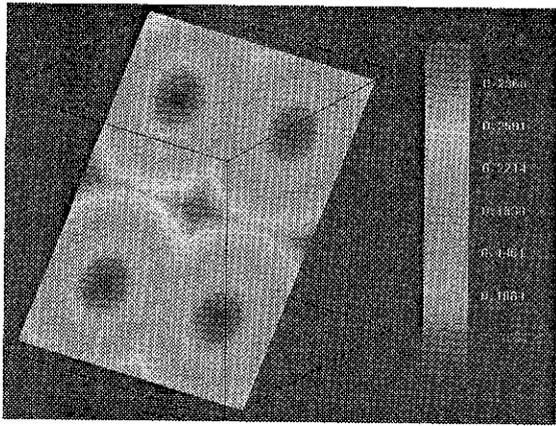


Fig. 2. Sezione trasversale del mesh sintetico

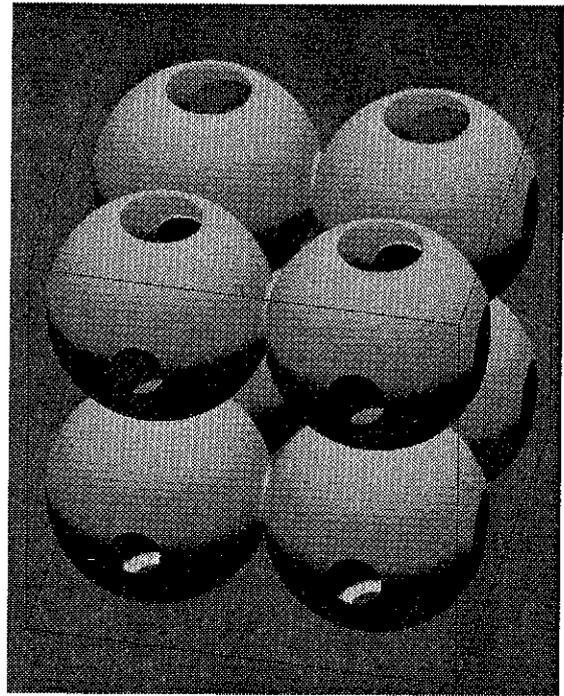


Fig. 3. Isosuperficie

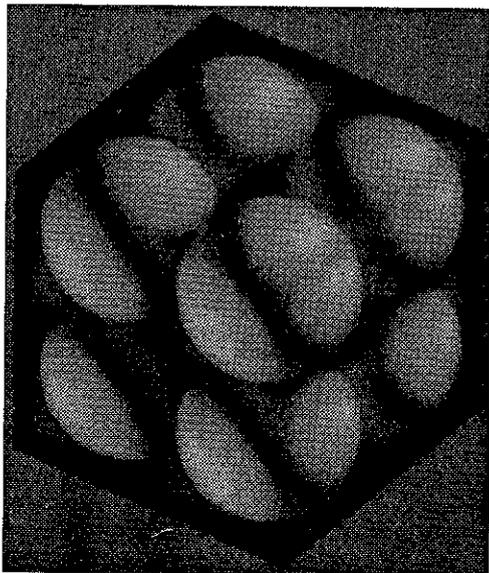


Fig. 4. Il mesh in "volume rendering"

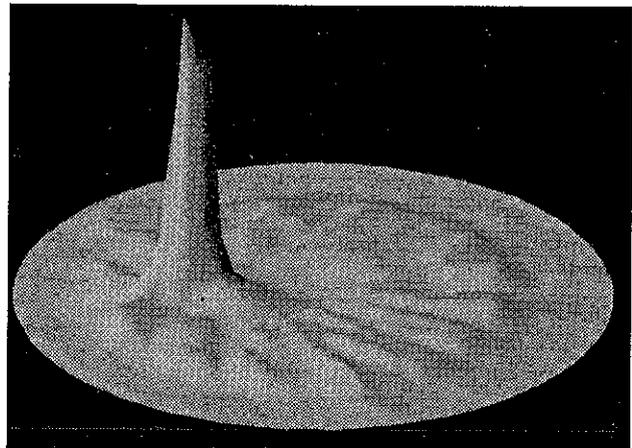


Fig. 5. Opacità ai raggi X di una sezione del provino in PVC