Equalizzazione Cromatica Semi-Automatica di Immagini per Texturing 3D

F. PIROTTI*, A. GUARNIERI*, A VETTORE*

*CIRGEO – Centro Interdipartimentale di Cartografia, Fotogrammetria, Telerilevamento e SIT - Università degli Studi di Padova – Viale dell'Università 16 – 35020 Legnaro (PD) – Tel. 049-8272688/2522 Fax. 049-8272686
e-mail: cirgeo@unipd.it – web: www.cirgeo.unipd.it

Abstract:

Il *texturing* mediante foto digitali di modelli 3D è un'operazione comunemente utilizzata per rendere questi ultimi più realistici, aggiungendo l'informazione cromatica al modello geometrico dell'oggetto rappresentato. Questa procedura consiste nell'assegnare al modello tridimensionale delle informazioni di colore alle sue singole parti, associando ad esse un numero di immagini necessario a ricoprirlo completamente. Il procedimento può essere descritto come un'operazione dove si ricorre a più immagini accostate tra loro per ricoprire il modello come se fosse un puzzle tridimensionale. Si presentano però una serie di difficoltà sia di tipo geometrico sia di tipo cromatico; queste ultime sono causate dalla diversa illuminazione, dalla diversa esposizione o anche dal diverso tipo di configurazione della camera e diventano evidenti quando nel modello le immagini vengono unite e si evidenziano le linee di confine tra una e l'altra proprio per queste differenze cromatiche. In questa nota si propongono soluzioni proprio a questo problema.

Il metodo proposto in questa nota consiste in un procedimento per equalizzare automaticamente le immagini utilizzate per il *texturing* di un modello ed eliminare quindi le differenze cromatiche nelle linee di unione. Il modello utilizzato, in un formato *Web3D* (*VRML*), viene analizzato per individuare i punti dove le immagini contigue vengono unite. Il procedimento poi ricava delle matrici di valori di colore dalle aree di sovrapposizione tra le due immagini, e infine, mediante analisi della correlazione e della distribuzione delle frequenze dei valori di grigio, ricava una *look – up - table* (LUT) per trasformare i valori di un immagine (denominata *slave*) verso quelli dell'immagine presa come riferimento (denominata *master*).

Sono stati così ottenuti dei risultati soddisfacenti per quanto riguarda la percezione delle immagini sul modello. Non viene osservata la differenza tra un'immagine e quella vicina come invece avveniva prima dell'elaborazione. Questo ha permesso di migliorare la qualità del modello nel suo insieme rendendolo più realistico e quindi più utile ai fini della visualizzazione. Il procedimento quindi fornisce uno strumento prezioso di *post-processing* che apporta un valore aggiunto di un certo peso al lavoro di modellazione dell'operatore.

Introduzione

Con lo sviluppo di calcolatori elettronici sempre più veloci nell'ultimo decennio la rappresentazione virtuale della realtà, sia in due dimensioni che in tre dimensioni, si sta facendo sempre più realistica. Un qualsiasi elemento può essere rappresentato a video conoscendo la sua forma ed il suo colore: è su questo che punta la tecnologia per il *virtual reality* - realtà virtuale, ossia modelli tridimensionali visualizzabili al computer che assomigliano sempre più all'oggetto reale grazie all'aumentata potenza dei calcolatori elettronici che permettono una maggior risoluzione del modello.

Il texturing consiste nel ricoprire l'oggetto con immagini dello stesso associando parti del modello a determinate zone delle immagini utilizzando dei punti di riferimento "modello? immagine". Si ricopre successivamente il modello con un processo di ricampionamento dell'immagine su una superficie tridimensionale. I problemi che questo comporta sono essenzialmente di due tipi: geometrico e cromatico. Geometrico perchè si vuole riportare un'immagine intrinsecamente a due dimensioni su una superficie che invece è tridimensionale. Si dovrà quindi apportare delle deformazioni all'immagine. Inoltre ci saranno distorsioni dovute all'angolo tra la normale alla superficie fotografata e la direzione di puntamento della fotocamera. Questo problema ha varie soluzioni sia di tipo empirico sia di tipo deterministico. Il secondo problema invece si presenta in quanto le varie immagini utilizzate per ricoprire il modello sono state prese in condizioni d'illuminazione diverse, quindi l'esposizione è stata necessariamente differente nelle varie immagini. Queste differenze sono molto chiare quando si visualizza il modello prima dell'elaborazione (vedi figura 1).

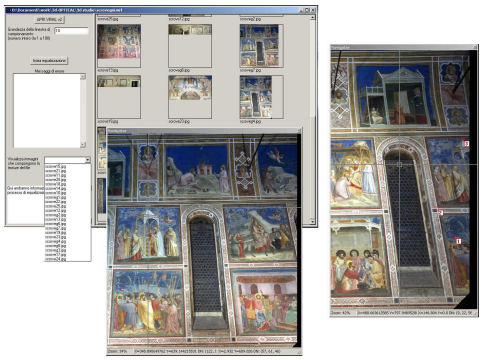

Figura 1 – Il programma di implementazione dell'algoritmo di equalizzazione con due delle immagini adiacenti attive e tre punti di controllo immessi dall'operatore visibili in quella di destra. Il menù a tendina illustra tutte le immagini utilizzate per il *texturing*.

Diverse ricerche sono state condotte per analizzare i metodi utili a comparare ed equalizzare delle immagini di un oggetto sottoposto a condizioni d'illuminazione diverse. La maggioranza di queste ricerche è stata fatta su casi che differiscono da quello preso in esame in questa nota in quanto ha considerato due immagini identiche tra loro e non appartenenti ad un modello tridimensionale. In questo caso le due immagini erano spesso di un pannello con colori di riferimento fotografato in condizioni d'illuminazione diverse. Altri studi ancora hanno analizzato il problema dal punto di vista della mosaicatura di immagini (es. strisciate di foto aeree oppure immagini satellitari) dove si lavora in due dimensioni e con errori di deformazione minori. In questi casi vengono utilizzate varianti della tecnica del *blending*, ossia una media dei colori delle due immagini contigue nella zona di confine per dare un'impressione di transizione meno netta. Nel caso in questa nota invece si hanno diverse immagini che messe insieme rappresentano l'oggetto nella sua totalità. Esse hanno delle zone di sovrapposizione con quelle confinanti, ed è proprio su questa limitata zona che si andrà ad operare per analizzare ed eliminare le differenze di cromaticità tra immagini contigue.

Il lavoro propone quindi una procedura automatica dove il programma identifica dei punti di controllo che saranno utilizzati per effettuare operazioni di analisi della correlazione tra coppie di immagini attigue. Un'immagine sarà fissata come riferimento e chiamata *master* ed una verrà modificata e quindi sarà chiamata *slave*. L'immagine *slave* viene modificata utilizzando operazioni sulla distribuzione delle frequenze dei valori di colore per avere un istogramma di valori più simile possibile a quello dell'immagine *master*. Questo permette di diminuire le differenze così nettamente visibili ad un livello non percepibile dall'occhio umano.

Metodi

L'obiettivo di questa nota consiste nel presentare la serie d'operazioni effettuate sul modello dall'applicativo appositamente implementato, che permettono di ottenere un modello con le informazioni di colore propriamente corrette. L'input necessario al programma è il modello completo dell'oggetto in formato *VRML v1.0*, con le varie parti della *texture* in immagini separate. Questa precisazione è necessaria in quanto alcuni modelli presentano la *texture* in un'unica grande immagine, in questo caso però l'applicativo non può equalizzare i colori all'interno della stessa. Il modello è letto e rielaborato in alcune delle sue parti per produrre alla fine dell'elaborazione un altro file nello stesso formato con le parti della *texture* equalizzate tra loro.

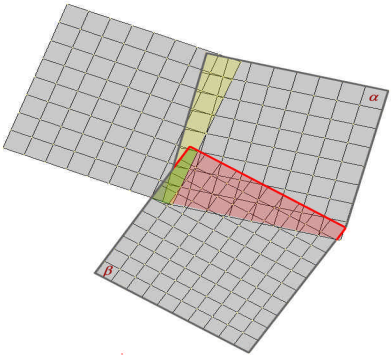

Figura 2 – Schema di tre ipotetiche immagini per texturing con zone di sovrapposizione. Le zone

Estrazione delle parti dell'immagine da equalizzare

La prima parte del lavoro ha richiesto l'estrazione di una serie di matrici (2,*n* dove *n* sono il numero di coppie di pixel sovrapposti – cfr. figura 3) contenenti i valori di colore tra due punti analoghi in due immagini adiacenti nel modello. Viene creata una matrice per ogni coppia di immagini vicine dove ogni colonna contiene una coppia di valori di colore (cfr. equazione 1). I valori di colore saranno triplette RGB per immagini a colore o singoli valori di grigio per immagini monocromatiche. Per comodità tra ogni coppia di immagini adiacenti a e b viene identificata una delle immagini come riferimento e l'altra come quella che subirà la modifica; vengono denominate rispettivamente *master* e *slave* (a_{master} e b_{slave}) per distinguerle. Queste immagini avranno una zona di sovrapposizione nel modello con dei pixel in comune, ossia ricampionate nello stesso punto del modello per ambedue le immagini, però con valori di colore (*r*, *g*, *b*) diversi (cfr. figura 3). I successivi riferimenti ai valori di colore per i singoli pixel dell'immagine saranno $a_{(xyz)}$ dove *x*, *y* e *z* sono le coordinate nel modello delle *texture*. Per rappresentare i pixel di due immagini che concorrono nello stesso spazio:

$$M = a_{(x\,y\,z)} \cap b_{(x\,y\,z)} = \begin{bmatrix} a_{(r_1\,g_1\,b_1)} & a_{(r_2\,g_2\,b_2)} \ldots a_{(r_n\,g_n\,b_n)} \\ b_{(r_1\,g_1\,b_1)} & b_{(r_2\,g_2\,b_2)} \ldots b_{(r_n\,g_n\,b_n)} \end{bmatrix} \qquad [Eq.\ 1]$$

Questo genere di campionamento apporta ad una serie di valori di coppie di pixel che dovrebbero essere analoghi e quindi essere il punto di partenza per elaborare i parametri di trasformazione. Prima però è necessaria un'ulteriore elaborazione per risolvere alcuni problemi.

Problemi e soluzioni adottate

Pixel analoghi nel campionamento

Il primo ostacolo che viene a porsi è l'estrazione di coppie di valori di punti analoghi tra immagine *master* e immagine *slave* per poter poi analizzarli e apportare la correzione all'immagine *slave*. Questo succede perché le due immagini avranno deformazioni diverse e quindi l'area di sovrapposizione non avrà i singoli pixel analoghi esattamente sovrapposti. Gli unici punti esattamente sovrapponibili sono quelli dei punti di controllo, anche se neanche questa affermazione è esatta come vedremo nel capitolo successivo. Per fare un esempio pratico, se si tira una linea tra due punti di controllo vedremo che questa non passa esattamente attraverso gli stessi elementi dell'immagine (cfr. immagine 4).

Figura 3 – Due immagini contigue con un dettaglio di un punto di controllo. E' visibile la deformazione dell'immagine dovuta all'orientamento fotocamera-oggetto.

La non esatta corrispondenza spaziale di pixel analoghi porterebbe ad un campionamento errato del valore di colore dei pixel, e successivamente ad una elaborazione meno precisa dell'equalizzazione.

La soluzione al problema è la stessa che viene proposta per il problema successivo, e quindi viene illustrata nel capitolo seguente.

Punti di controllo realmente analoghi

Quando l'operatore inserisce i punti di controllo dove le immagini contigue vanno a sovrapporsi, egli può fare degli errori che, anche se sono praticamente insignificanti dal punto di vista della costruzione del modello, sono invece importanti per il lavoro di equalizzazione. Ad esempio nella figura quattro di vede il punto 27 ed il punto 26 nelle due immagini. Il punto è rappresentato dalla punta della cornice sopra la testa del santo rappresentato. Questa cornice è duplice: in colore senape all'interno ed in colore bianco esternamente; l'operatore ha scelto la punta della cornice interna color senape, ma è possibile pensare che, selezionando con il mouse il punto, egli sia andato a mettere il punto sopra un pixel di colore bianco in un'immagine, e sopra un pixel di colore senape nell'altra. Un caso come questo rovinerebbe tutto il procedimento di equalizzazione perché l'algoritmo considera quel punto come un punto di controllo, ossia un punto che dovrebbe avere colore identico. Registrando un colore così diverso invece, l'algoritmo trasformerà tutta l'immagine *slave* con dei parametri errati.

La soluzione, che presenta anche la soluzione al problema precedente, è quella di analizzare una finestra intorno al pixel del punto di controllo, per determinare la correlazione tra le due finestre corrispondenti nelle due immagini. Nell'immagine *slave* la finestra viene poi spostata per fornire una matrice con i valori di correlazione dove il valore centrale sarà quello del punto originale di controllo (cfr. eq. 2). Se il valore di correlazione maggiore non è quello centrale, si sposta il punto di controllo tanto quanto è il valore di correlazione maggiore.

$$\rho_{i,j} = \frac{cov(X_i, X_j)}{\sigma_i \sigma_j}$$

[Eq. 2 - Matrice di correlazione tra finestre]

$$\boldsymbol{\rho} = \begin{pmatrix} \rho_{1,1} & \rho_{1,2} & \rho_{1,3} & \cdots & \rho_{1,n} \\ \rho_{2,1} & \rho_{2,2} & \rho_{2,3} & & \\ \rho_{3,1} & \rho_{3,2} & \rho_{3,3} & & \vdots \\ \vdots & & & \ddots & \\ \rho_{n,1} & & \cdots & & \rho_{n,n} \end{pmatrix}$$

Dove:

σ = Deviazione Standard

ρ = Indice di correlazione

cov = covarianza

X = matrici

i,j = immagini *slave* e *master*

Questa operazione porta ad "affinare" i punti dove vengono campionati i valori di colore, assicurando che siano veramente analoghi tra immagini. A questo punto il campionamento fornisce una serie di valori sulla quale si può procedere all'analisi e a trovare i parametri per l'equalizzazione.

Elaborazione dei valori ed equalizzazione

Le due righe della matrice contengono quindi i valori campionati dai quali si potrebbe già procedere

con l'equalizzazione. Si pone però il problema di prendere in considerazione non tanto la zona, ma la polilinea che delimita le due immagini (cfr. figura 3, la linea in rosso). Infatti, il programma effettuerà il *texturing* coprendo un'immagine con quella vicina e quindi la parte con differenze cromatiche visibili sarà lungo la linea di confine. Sono questi i pixel che vanno trasformati cromaticamente per ridurre le differenze tra il passaggio tra un'immagine e l'altra. L'algoritmo di equalizzazione farà pesare il fattore di trasformazione di più sulla zona immediatamente vicino a quella polilinea per diminuire poi in modo inversamente proporzionale alla distanza dalla linea (cfr. equazione 2)

$$w(d) = \frac{1}{dp}$$

d = distanza dalla linea p
w = peso del parametro di equalizzazione

[*Eq. 3*]

I valori lungo la linea di demarcazione delle due immagini vengono analizzati per ottenere due distribuzioni delle frequenze che possono essere rappresentate da un grafico per ogni immagine (cfr. figura 4).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.010000 & 0.990000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

[*Eq. 4*]

Dove:
Y = Componente acromatico
u' v' = Componenti cromatichee
R, G, B = valori di grigio per ogni banda

$$Y = 0.17697*R + 0.8124*G + 0.01063*B$$

$$u' = 4X / (X + 15Y + 3Z)$$
$$v' = 9Y / (X + 15Y + 3Z)$$

L'equalizzazione mediante corrispondenza dell'istogramma (*histogram matching*) può essere fatta direttamente su valori di RGB dell'immagine; però delle ricerche (Lucchese) hanno messo in evidenza che si può ottenere un risultato migliore se si trasformano le coordinate RGB in XYZ e successivamente in un canale cromatico e due canali acromatici (cfr. Eq. 3). Successivamente si procede con la trasformazione dei valori dell'immagine *slave* con corrispondenza dell'istogramma e poi si riportano i valori in RGB con il procedimento inverso.

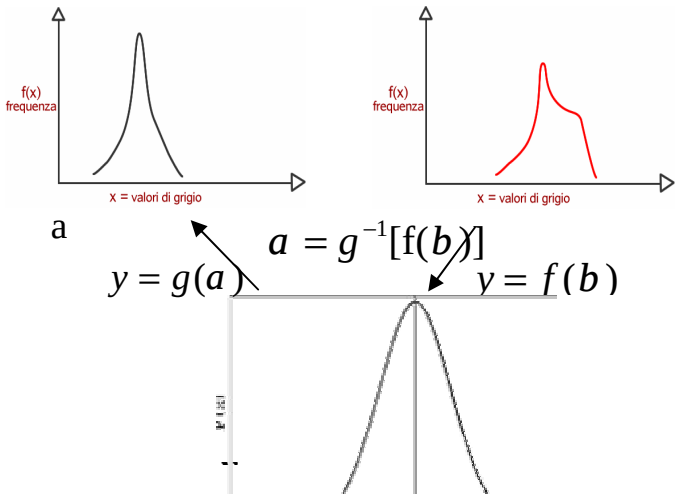

Figura 4 – Ipotetici grafici di distribuzione delle frequenze con le funzioni di equalizzazione dell'istogramma

Essendo i valori possibili per un'immagine digitale una quantità discreta espressa in bit (nel nostro caso 256 valori per ogni canale ossia 8 bit) la funzione di corrispondenza dell'istogramma è rappresentata da una *look-up-table* (LUT) ossia una tabella di corrispondenze tra i valori possibili nell'immagine originale e quelli da assegnare all'immagine trasformata (cfr. figura 5). La LUT viene fatta in due passaggi: prima si trova la serie di corrispondenze per equalizzare sia l'immagine *master* che quella *slave*, poi l'immagine *slave* equalizzate viene trasformata utilizzando la LUT usata per equalizzate la *master* (cfr. figura 4) .

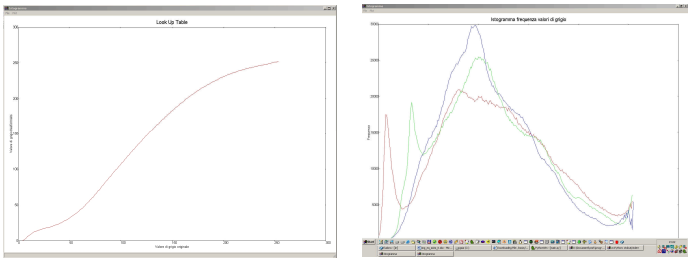

Figura 5 – A sinistra grafico della LUT e a destra istogrammi dei tre canali di una delle immagini da equalizzare.

Alcuni dei problemi riscontrati sono principalmente quelli della distorsione delle immagini, specialmente nella zona di sovrapposizione, in quanto è la zona ai margini dell'immagine. Infatti, se dalle immagini fossero assenti deformazioni, si potrebbe semplicemente fare una classica operazione di blending ossia colorare i pixel nella zona di sovrapposizione facendo la media dei valori delle due immagini pesata con la distanza dalla linea di margine. Questo non è stato possibile in quanto i pixel corrispondenti si sovrappongono in modo preciso solo lungo la linea di confine e non nel resto della zona di sovrapposizione. Per questo motivo l'approccio in questo lavoro è stato utilizzare proprio i pixel nella linea di confine e l'istogramma dell'immagine.

Analisi dei Risultati e Conclusioni

I diversi accorgimenti per portare ad una procedura di equalizzazione valida e robusta hanno dato risultati confortanti. I vari problemi che sono sorti nelle varie fasi del lavoro sono stati superati ed hanno apportato ad migliorare la solidità della procedura. La correzione dei punti di controllo mediante analisi della correlazione dei vicini si è rivelata necessaria per eliminare i problemi derivanti dalla deformazione delle immagini.
Sarebbe stato interessante apportare anche degli elementi di *blending* al sistema, per rendere più sicuro il risultato dell'equalizzazione. L'associazione della tecnica proposta al *blending* può rivelarsi proficua in quanto si può "sfumare" eventuali residui di differenze nella zona dove è più necessario, ossia lungo la linea di confine. La corrispondenza degli istogrammi infatti non assicura una totale corrispondenza tra coppie di pixel analoghi, ma lavora con una distribuzione statistica ed è quindi assoggettato a influenze aleatorie come ad esempio una distribuzione di colore molto limitata.

Referenze

Lucchese L. Mitra S. *A New Method for Color Image Equalization*, 2001 IEEE

Lenz R., Meer P., Kasary M.H., *Spectral-Based Illumination Estimation and Color Correction*, COLOR research and application, vol. 24, n 2 Aprile. 1999

H. H. Hou and H. C. Andrews, *Cubic Splines for Image Interpolation and Digital Filtering* , IEEE Trans. on Acoustics, Speech, and Signal Processing, ASSP-26(6), Dicembre 1978 pp 508-517

M. UNSER, A. ALDROUBI, M. EDEN, *Fast B-Spline Transforms for Continuous Image Representation and Interpolation* , IEEE Trans. on Pattern Analysis and Machine Intelligence Vol 13(3) Marzo 1991 pp 277-285