MATEMATICA NASCOSTA[1]

RICCARDO RICCI

Dipartimento di Matematica, Università di Firenze

Nell'esperienza quotidiana l'espressione "digitale" è talmente comune che forse si è perduto il senso di questo aggettivo. "Digitale", nel senso che gli si attribuisce parlando di "immagini digitali" o di "musica digitale" è una delle tante parole di origine latina reimportate in italiano tramite l'inglese: in questo caso la parola "digit", che ha lo stesso significato dell'italiana "cifra". Quindi "digitale" è tutto quello che viene affidato per la conservazione e la fruizione alla codifica (e decodifica) tramite una successione di numeri (di 0 e 1). È abbastanza ovvio che in questi processi la matematica debba avere la sua parte accanto all'informatica (anche senza voler pensare a quest'ultima come a una forma moderna della matematica stessa). Ma per quasi tutti gli utenti del "mondo digitale" non è per niente chiaro quale e quanta matematica sia coinvolta nella visione di una fotografia sulla schermo di un computer o nell'ascolto di una brano musicale su Ipod.

Scopo di questa lezione è quello di far emergere una parte della matematica che giace nascosta nella fruizione "digitale" delle immagini.

1. La fotografia digitale

Oggi la fotografia digitale ha quasi completamente soppiantato il vecchio procedimento fotografico basato sulla pellicola fotosensibile, tanto che il 30 dicembre 2010 a Pearson nel Kansas è stato sviluppato l'ultimo rotolino di pellicola Kodachrome, la più usata pellicola per foto a colori.

In una fotografia digitale possiamo individuare tre processi fondamentali. Il primo è quello ottico, del tutto analogo a quello delle vecchie macchine fotografiche, che consente di formare l'immagine su una superficie (la pellicola nella vecchia foto, il "sensore elettronico" per la foto digitale).

Il sensore è una griglia occupata da minuscoli semiconduttori (CCD o CMOS nello stato dell'arte attuale) il cui scopo è trasformare l'intensità della luce in un segnale elettrico. Questo viene a sua volta trasformato in un numero (questo avviene direttamente nel sensore nella tecnologia CMOS). I colori vengono "suddivisi" in tre colori fondamentali: rosso, verde e blu – *red, green, blue*, da cui la sigla RGB, in inglese – catturati da tre differenti sensori adiacenti. Il luogo da essi occupato è comunemente detto pixel. Una fotocamera di oggi (per uso meno che amatoriale) ha una "risoluzione" di almeno 10 megapixel, ovvero di dieci milioni di pixel.

---

1 Lezione tenuta a Firenze il 5 novembre 2010 presso il Liceo Classico Michelangiolo, nell'ambito dell'edizione 2010 di Pianeta Galileo.

Il terzo processo è lo stoccaggio di tutti questi numeri in un file. Il file prodotto (nel formato detto RAW) ha una dimensione ragguardevole. Infatti, ogni pixel produce tre numeri (in formato binario, cioè fatti da 0 e 1) che rappresentano l'intensità del rispettivo colore. Questa intensità è suddivisa in una scala che dipende da quanti bit sono dedicati a ogni colore (si usa la sigla bpp, bit per pixel). Una buona fotocamera usa 8 bit per colore (quindi 24 bpp) permettendo così di rappresentare un totale di 2 elevato alla 24, ovvero circa 16,8 milioni di colori. Quindi un file RAW per una foto occupa circa 240 milioni di bit, ovvero circa 30 Megabyte per ogni foto. Nonostante le capacità di memoria sempre più grandi dei dispositivi odierni, questa è una dimensione veramente grande. La grande dimensione di questi file è un difetto molto grave se si vogliono trasferire queste immagini tramite Internet.

Si pone quindi il problema di come ridurre la dimensione dei file (cioè dell'informazione "bruta" legata a ogni foto) senza un'eccessiva perdita di qualità delle immagini. A questo scopo sono nati vari algoritmi di compressione. Alcuni sono basati sulla creazione di "dizionari" che assegnano un "nome" (breve) a una serie di lunghezza fissata di bit: più sequenze uguali ci sono nel file (per esempio, se si ha una foto con molte zone di colore uniforme) più ci saranno serie uguali di bit, tutte condensate da un solo nome. Su questa strategia, che potremmo definire "informatica", si basano i formati GIF e PNG.

Ma il formato di compressione più noto per le fotografie a colori è certamente il formato JPG (o JPEG, acronimo di Joint Photographic Experts Group, la sigla del gruppo di esperti che creò lo standard verso la metà degli anni Ottanta).

Il formato JPEG si basa essenzialmente su una teoria della matematica classica, ovvero la trasformata e le serie di Fourier.

Le serie di Fourier appaiono per la prima volta nel lavoro dedicato dal matematico francese J.B. Joseph Fourier alla teoria della propagazione del calore.[2] In tale lavoro, Fourier egli sostiene che ogni funzione di una variabile può essere ottenuta come la somma di infiniti termini del tipo seno e coseno della variabile stessa, ciascuno moltiplicato per un'opportuna costante. Queste costanti, dette *coefficienti di Fourier* della funzione, permettono di identificare univocamente la funzione stessa e quindi la loro conoscenza equivale alla conoscenza della funzione di partenza.

Questa scomposizione di una funzione in somma di funzioni trigonometriche può essere fatta anche per funzioni di due variabili (x, y), moltiplicando funzioni trigonometriche della x con funzioni trigonometriche della y. Nella nostre fotografie digitali, il file in formato RAW può essere pensato come la raccolta dei valori che una funzione delle variabili (x, y), rispettivamente l'ascissa e l'ordinata dei punti del sensore, assume nei punti della griglia corrispondenti ai pixel (in realtà le funzioni sono tre, una per ogni colore della terna RGB). Ora possiamo pensare di sviluppare questa funzione usando la tecnica delle serie di Fourier. I dati del nostro file RAW sono quindi trasfor-

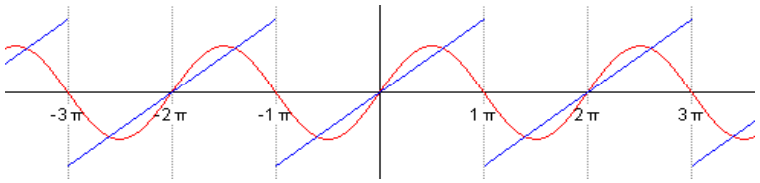
2 J.B.J. Fourier, *Mémoire sur la propagation de la chaleur dans les corps solides*, 1807, e *Théorie analytique de la chaleur*, 1822

mati nei *coefficienti di Fourier* di una opportuna funzione. Non sembra di aver fatto molti progressi, visto che abbiamo appena detto che le somme sono fatte da infiniti termini e quindi abbiamo infiniti coefficienti di Fourier.

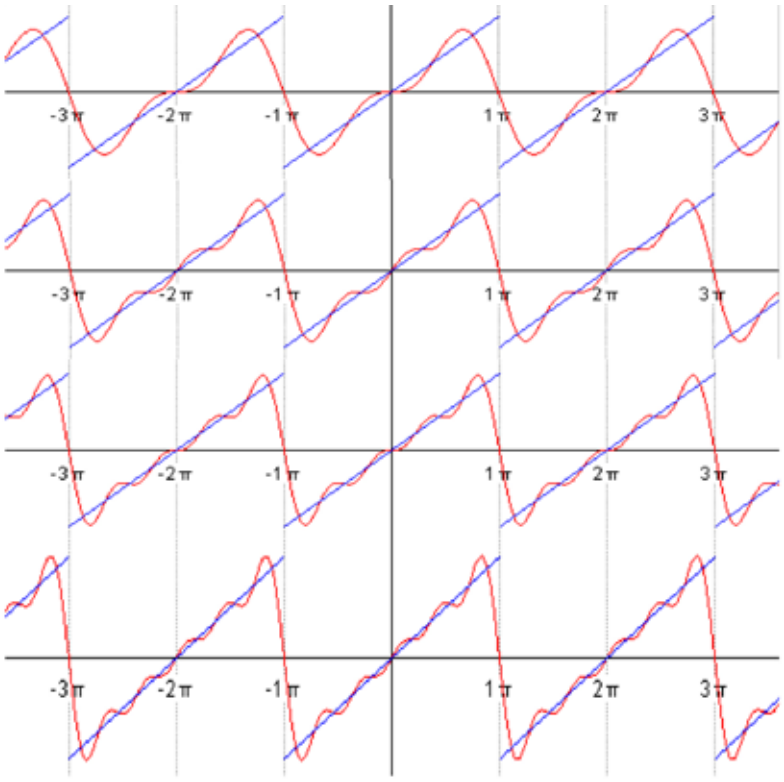
Ma quello che è importante per il nostro problema è che i coefficienti di Fourier decrescono molto rapidamente, ossia solo un numero molto piccolo di essi ha valori sensibilmente diversi da zero, e tutti gli altri (infiniti) termini possono essere trascurati. Perciò, conservando solo pochi termini della somma otteniamo una approssimazione della funzione che ha una forte somiglianza con la soluzione di partenza. Il punto chiave è che, se ci accontentiamo di quest'approssimazione, basta ricordarsi i pochi coefficienti di Fourier che abbiamo selezionato al posto di tutti i valori della funzione contenuti nel file RAW.

Vediamo un esempio (unidimensionale) per capire come funziona l'approssimazione.

Prendiamo la funzione $f(x)=x$ nell'insieme $(-\pi, \pi)$ ed estendiamola periodicamente fuori da questo intervallo, rappresentata in blu nel grafico



Nel grafico abbiamo sovrapposto (curva in rosso) il primo addendo della somma di Fourier. Continuiamo aggiungendo via via i termini successivi fino al quinto addendo.



Nell'ultima figura abbiamo già un buon accordo tra il grafico della funzione originaria e la sua approssimazione di Fourier. Quello che conta ora è che il grafico rosso è ricostruibile ricordando solo i coefficienti di Fourier (che in questo caso sono 2, -1, 2/3, -1/2 e 2/5) tramite la formula

$$\sum_{n=1}^{5}(-1)^{n+1}\frac{2}{n}\sin(nx)$$

Al posto delle valutazioni della funzione su tutti i punti della nostra griglia, ora dobbiamo ricordare solo i coefficienti dello sviluppo, con un notevole risparmio di memoria (dimensione del file).

C'è tuttavia una sostanziale differenza tra la curva blu e quella rossa: la prima è discontinua, ovvero mostra salti bruschi tra tratti di grafico in punti di ascissa vicina, la seconda è una curva continua anche se con tratti molto ripidi. Quando questa differenza viene trasformata nella "colorazione" dei differenti pixel, la figura generata dal formato JPEG mostra dei pixel colorati in zone dove non dovrebbero esserlo e viceversa pixel non sufficientemente colorati in zone che magari dovrebbero risultare uniformemente colorate (ciò avviene nelle zone di transizione da un colore all'altro). Questo fenomeno è particolarmente evidente se si salva in formato JPEG un documento che contiene caratteri a stampa o disegni fatti da curve molto nette (come questa pagina), mentre diventa assai poco percettibile nelle fotografie, dove i colori tendono naturalmente a "sfumare" uno nell'altro. Tutto ciò ha una qualche importanza pratica: se dovete digitalizzare un documento scritto, il formato JPEG dà risultati piuttosto deludenti (la scrittura risulta "sbavata") mentre una codifica con il formato PNG dà un risultato di gran lunga migliore per la leggibilità del documento (i grafici e la formula riportate qui sopra sono state digitalizzate in formato PNG). La situazione si ribalta nelle foto: per ottenere risultati paragonabili all'effetto visivo di un file JPEG, la codifica PNG (o GIFF) produce file di dimensioni considerevolmente maggiori.

L'analisi di Fourier è alla base anche delle codifiche digitali dei segnali sonori. La più comune codifica attuale (il formato MP3) è ottenuta con l'algoritmo di codifica MPEG che lavora sul cosiddetto dominio delle frequenze, ovvero sulla rappresentazione del segnale sonoro non in funzione del tempo (la rappresentazione "naturale") ma in funzione delle componenti del segnale ottenute tramite una trasformata di Fourier (è una generalizzazione dei coefficienti di Fourier di cui abbiamo parlato). Tuttavia il processo è molto più complesso di quanto sia quello per la digitalizzazione delle immagini e non può essere descritto in questa sede.

2. La *computer graphics*

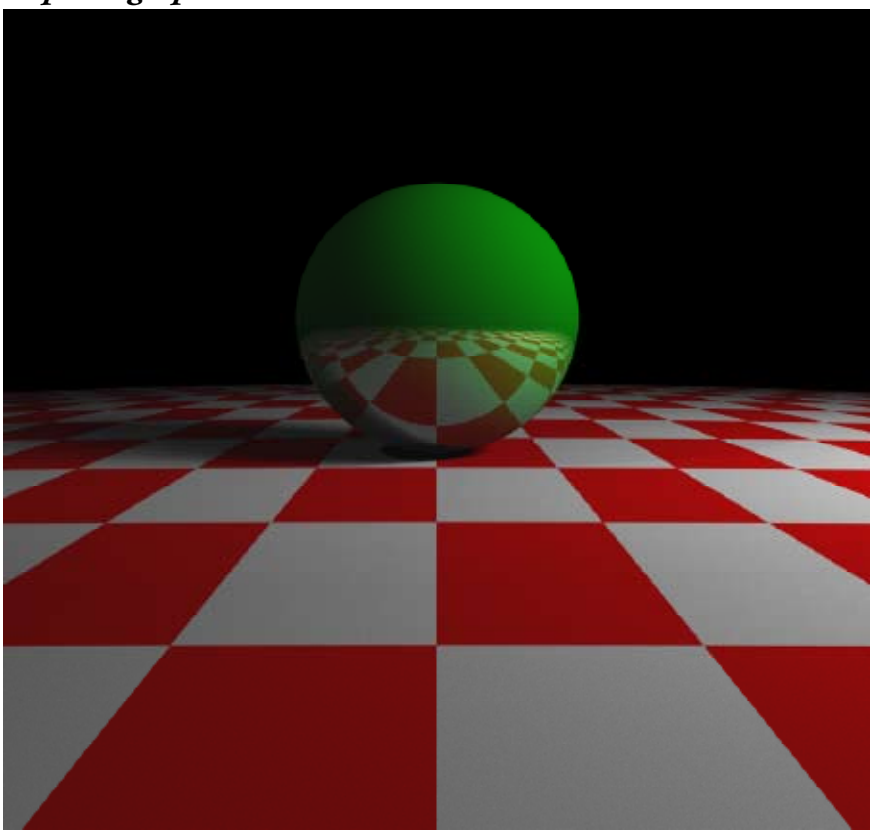


Figura 1. Figura realizzata dagli studenti del corso di Computer Graphics tenuto da Alberto Mancini (Dipartimento di Matematica, Università di Firenze).

La *computer graphics* (ormai il nome inglese ha soppiantato ogni tentativo di traduzione italiana) ha per scopo quello di *creare* immagini digitali senza che queste debbano essere ottenute da fotografie o filmati di oggetti reali. L'importanza, in primis commerciale, di quest'attività è stata illustrata anche in una lezione di Pianeta Galileo del 2009,[3] cui rimando per molte interessanti osservazioni generali sul "mestiere" di grafico. Questa lezione contiene una descrizione di molta della matematica coinvolta nella *computer graphics*, in particolare la geometria proiettiva e l'interpolazione polinomiale. Il ruolo della prima è essenzialmente quello di gestire la rappresentazione prospettica delle figure nello spazio, necessaria quando si voglia ottenere un effetto realistico come nelle due figure qui riportate. I suoi principi fondamentali datano dai grandi pittori e teorici della pittura del Rinascimento: Brunelleschi, Alberti, Piero della Francesca (autore di un trattato matematico sulla prospettiva).

Come si costruisce un'immagine al computer? Il primo passo è dare una descrizione matematica della forma degli oggetti. Per prima cosa dobbiamo individuare nel piano rappresentativo (che poi verrà visualizzato sullo schermo) gli oggetti presenti nella sce-

3 Marco Franciosi, La matematica dei videogiochi, pp. 81-94 di *Pianeta Galileo* 2009, a cura di A. Peruzzi, Consiglio regionale della Toscana, Firenze 2009.

na. Se ci riferiamo alla Figura 1, abbiamo una sfera e una sorta di "tappeto" quadrettato. La sfera sarà descritta da una griglia di punti che uniremo con dei segmenti ottenendo una cosa del tipo raffigurato nella Figura 2.

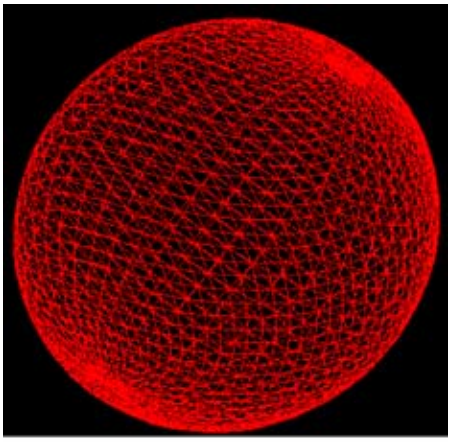


Figura 2.

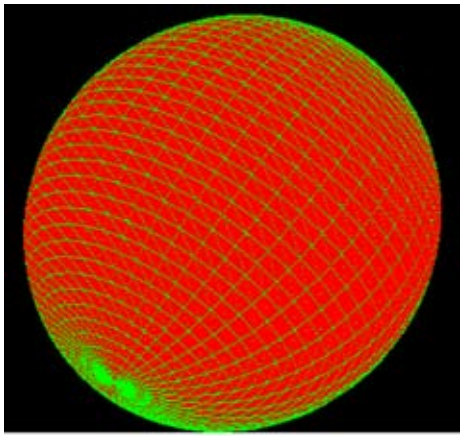


Figura 3.

Ma la sfera è un oggetto solido, quindi possiamo vedere solo i punti che stanno dalla parte dell'osservatore. Dobbiamo quindi rimuovere le "linee nascoste" ovvero quelle linee che delimitano parti della superficie che si trovano "dietro" (oppure che sono nascoste da altri oggetti). Questo è abbastanza facile da fare per un oggetto convesso come una sfera: la superficie è rimossa se la sua *normale esterna,* cioè la direzione perpendicolare alla superficie nel verso uscente dall'oggetto, forma un angolo acuto con la direzione dello sguardo dell'osservatore. Lo si può fare al computer tramite un algoritmo matematico noto come *backface culling*.

Più complicato è eliminare parti nascoste di oggetti non convessi. Per esempio, se rappresentiamo un bicchiere, possiamo vedere alcune parti della superficie interna, ma altre saranno nascoste dal resto del bicchiere anche se la normale esterna guarda verso l'osservatore. Stessa cosa se dobbiamo eliminare parti di un oggetto coperte alla vista da un altro oggetto presente nella scena. In questo caso si usa l'algoritmo detto "del pittore" che consiste nel rappresentare prima gli oggetti (parti di oggetti) più lontani e poi "dipingendoci sopra" quelli più vicini (*Z-sorting*). Si deve quindi decidere "matematicamente" quali sono gli oggetti più lontani, e per far ciò dobbiamo "ricordarci" della coordinata nella direzione della profondità (l'operazione è molto costosa dal punto di vista del calcolo perché si deve lavorare al livello dei singoli pixel e viene spesso implementata in un hardware dedicato, con acceleratori grafici).

Figura 4. Da M. Slater, The radiance equation.

Ma per raggiungere un effetto realistico è necessario integrare al disegno gli effetti della luce. Poiché gli oggetti rappresentati sono "nati dal nulla", cioè ottenuti tramite descrizioni matematiche degli oggetti e del loro posizionamento nello spazio, è necessario che anche l'illuminazione di questi oggetti venga "creata" da un algoritmo matematico.

Per capire la complessità del problema basta guardare con attenzione le due immagini qui presentate: Fig. 1 e Fig. 4. La seconda (che NON è una fotografia digitalizzata, ma un'immagine creata direttamente al computer) è di impressionante realismo, forse un po' troppo complessa per essere descritta nella sua interezza. Va però notata una sua caratteristica fondamentale: la luce proviene dall'alto a sinistra, per chi guarda, e illumina *direttamente* solo una parte della scena. La parte sinistra del portico si trova in ombra, ma è a sua volta illuminata dalla luce (parzialmente) riflessa dalle zone illuminate. A sua volta, per quanto in misura minore, anche gli oggetti illuminati di riflesso rinvieranno la luce nelle altre parti della scena, e così via.

Una situazione scenograficamente più semplice è presente nella Fig. 1 (risultato di una "esercitazione" di un corso tenuto da A. Mancini presso il Dipartimento di Matematica dell'Università di Firenze). Qui la luce proviene sempre dall'alto, ma dalla destra della scena. Colpisce direttamente una parte del pavimento e una parte della sfera che vi è appoggiata sopra. La sfera s'immagina che debba avere una superficie riflettente (almeno parzialmente) visto che la sua parte inferiore rispecchia il disegno

del pavimento. Anche qui si ha un gioco di riflessione parziale della luce che illumina le parti in ombra.

Dobbiamo ricordarci che le figure qui riprodotte sono la visualizzazione di file numerici a loro volta prodotti "matematicamente". Se gli effetti che vi si possono osservare sono realistici, ciò è ottenuto grazie alla possibilità di dare un "buon" *modello matematico* della propagazione, riflessione e diffusione della luce. La luce è un fenomeno fisico complesso che presenta aspetti che possono essere descritti talora in termini di onde luminose e talora tramite particelle (fotoni). Nel modello usato si fa uso del concetto di *radianza*, una misura dell'energia luminosa che si assume come propagantesi in raggi luminosi che, quando incontrano un oggetto, vengono in parte assorbiti e in parte riemessi dalla superficie dell'oggetto secondo caratteristiche tipiche della superficie dell'oggetto stesso (trascuriamo qui gli effetti di "attraversamento" di oggetti "trasparenti").

L'equazione che regola questo comportamento è nota in *computer graphics* con il nome di *rendering equation*, che esprime una particolare forma di conservazione dell'energia. Se indichiamo con L la radianza, abbiamo che a ogni punto della superficie e in ogni direzione, la luce uscente L_o è la somma della luce emessa L_e e della luce riflessa. Questa, a sua volta, è data dalla luce incidente L_i da tutte le direzioni, moltiplicata per una funzione di riflessione dipendente dalle caratteristiche della superficie e per il coseno dell'angolo di incidenza. Tutte queste quantità dipendono dalla lunghezza d'onda della luce stessa (dal "colore"). Il risultato è la seguente equazione di rendering

$$L_o(\mathbf{x},\omega,\lambda,t) = L_e(\mathbf{x},\omega,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega',\omega,\lambda,t) L_i(\mathbf{x},\omega',\lambda,t)(-\omega'\cdot\mathbf{n})d\omega'$$

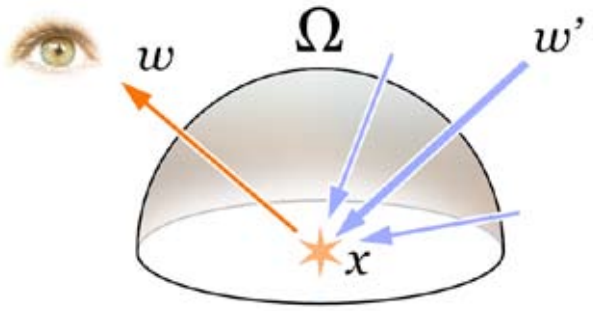


Figura 5. L'equazione di rendering.

L'equazione di rendering è un caso di equazione integrale, un tipo di equazione molto studiato in fisica matematica (un altro celebre esempio è l'equazione di Boltzmann della meccanica statistica).

Per determinare la corretta illuminazione di una scena dobbiamo trovare la soluzione dell'equazione di radianza in funzione della sorgente che illumina la scena (la luce "esterna") e della forma, posizione e caratteristiche degli oggetti presenti, calcolando la luce che finisce nel punto in cui si trova l'occhio dell'osservatore.

La soluzione tuttavia può essere costruita solo in modo approssimato. Negli anni

sono state sviluppate diverse strategie di soluzione, molte delle quali si basano sul principio del Metodo Montecarlo, che consiste nel generare un numero elevato di prove casuali (in questo caso si tratta di raggi luminosi) e fare una sorta di media dei risultati ottenuti. Questi metodi (*ray tracing*, *path tracing*, *Metropolis light trasport*) richiedono un enorme sforzo di calcolo e quindi molto tempo per ottenere un'immagine realistica. Ciò li rende adatti solo per immagini statiche, mentre non possono essere utilizzati per immagini in movimento come quelle dei videogiochi.