



UNIVERSITÀ DEGLI STUDI DI CAGLIARI
FACOLTA' DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA IN MATEMATICA

Modelli matematici e algoritmi numerici per la Photometric Stereo

RELATORE:

Prof. Giuseppe RODRIGUEZ

TESI DI LAUREA:

Gabriele STOCCHINO

Anno Accademico 2014/2015

*Ai miei genitori e a mio fratello,
per il supporto che mi hanno sempre dato*

Indice

1	Photometric Stereo	9
1.1	Formulazione del problema	9
1.2	Metodo alle differenze finite	11
1.3	Metodo ai minimi quadrati	13
2	Algoritmo alla Poisson	17
2.1	Costruzione del campo normale	17
2.2	Problema differenziale	19
2.2.1	Aspetti teorici dell'equazione di Laplace	20
2.2.2	Risoluzione mediante le differenze finite	21
2.3	Implementazione dell'algoritmo	25
3	Algoritmo alla Hamilton–Jacobi	29
3.1	Caso continuo con due immagini	29
3.2	Caso discreto con due immagini	32
3.3	Implementazione dell'Algoritmo con 2 luci	35
3.4	Caso con più luci	37
4	Esperimenti numerici	39
4.1	Esempio di ricostruzione di una superficie mediante l'Algoritmo alla Poisson	39
4.2	Esempio di ricostruzione di una superficie mediante l'Algoritmo alla Hamilton–Jacobi	42
4.3	Ricostruzione di una superficie con dati perturbati	45
4.4	Non derivabilità della superficie	45
4.5	Esperimenti sui tempi di calcolo dell'Algoritmo di Poisson	46
	Bibliografia	53

Introduzione

Un classico problema nell'ambito della Computer Vision è la ricostruzione 3D di un oggetto. Esso è un problema inverso, in quanto a partire da una o più immagini, si cerca di risalire alla superficie che le ha generate.

In origine l'obbiettivo era quello di ricreare la superficie avendo a disposizione solo un'immagine, ma ciò si è rilevato un problema mal posto ([1]) a causa della mancanza di unicità della soluzione. Per cui, un modo per evitare questo problema è quello di tener conto non di una sola ma di più immagini, le quali possono essere prese in due modi diversi.

1. Le immagini sono prese da diversi punti di vista sotto le stesse condizioni di luce.
2. Le immagini vengono prese da una stessa posizione, e ciò che cambia è la posizione della luce.

Gli algoritmi che fanno riferimento al primo punto rientrano nella categoria della *Stereovision*, mentre i secondi nella categoria della *Photometric Stereo*.

Inizialmente la Photometric Stereo è stata studiata da alcuni ottici ([2],[3],[4]), le cui ricerche riguardavano lo studio di mari della luna e la topografia lunare, e successivamente grazie ai contributi pionieristici di Horn e di altri colleghi del MIT ([5]) l'interesse fu sempre maggiore.

Un metodo concorrente alla Photometric Stereo è quello dello scanner laser, il quale ha però due grossi svantaggi:

1. La superficie dev'essere abbastanza piccola da poter essere contenuta in un rilevatore.
2. Lo scanner laser deve stare molto vicino all'oggetto, e ciò può essere un problema se si vuole ricostruire in 3D una statua o un'altra opera la cui incolumità è fondamentale.

La Photometric Stereo supera entrambe le difficoltà e proprio grazie al fatto che superi la seconda problematica trova applicazione nella ricerca archeologica.

La seguente tesi prenderà in esame il problema della Photometric Stereo nell'ipotesi aggiuntiva in cui è nota la posizione delle luci. In particolare, è strutturata come segue.

Nel capitolo 1 verrà modellizzato matematicamente il problema della Photometric Stereo, e verranno presentati alcune conoscenze matematiche necessarie alla comprensione della tesi. In particolare sarà utile introdurre il metodo alle differenze finite, che permette di risolvere numericamente una o più equazioni alle derivate parziali discretizzando il dominio e riducendo così il problema alla risoluzione di un sistema lineare, e il metodo ai minimi quadrati, necessario per lo studio di sistemi lineari sovradeterminati.

Nel capitolo 2 verrà esposto in dettaglio l'Algoritmo alla Poisson, il quale è un classico metodo per risolvere il problema di ricostruzione di una superficie Lambertiana sfruttando la formulazione matriciale della Legge di Lambert.

Nel capitolo 3 si tratterà dell'Algoritmo alla Hamilton-Jacobi, il cui algoritmo si basa sulla formulazione continua della Photometric Stereo esposta nel capitolo 1

Nel capitolo 4 verranno descritti alcuni esperimenti numerici utili per comprendere l'efficacia degli algoritmi proposti.

Capitolo 1

Photometric Stereo

1.1 Formulazione del problema

In questa sezione si cercherà di modellizzare il problema della ricostruzione 3D di un oggetto dove la macchina fotografica mantiene una posizione fissa e la posizione della luce varia.

Data una superficie M che appartiene allo spazio \mathbb{R}^3 , si considera il sistema di coordinate O_{xyz} e si suppone di fissare sull'asse z una macchina fotografica. Verrà risolto il problema della ricostruzione 3D sotto le seguenti ipotesi:

1. Le fonti luminose sono poste all'infinito.
2. La superficie è Lambertiana.
3. Non ci sono autoriflessioni.
4. Il punto ottico è sufficientemente lontano dalla superficie da evitare deformazioni di prospettiva

Dove con superficie Lambertiana si intende che per ogni punto della superficie vale la Legge di Lambert, o Legge del coseno, ossia: l'illuminamento prodotto da una sorgente luminosa sulla superficie è direttamente proporzionale all'intensità luminosa e al coseno dell'angolo tra la direzione dei raggi luminosi. In formule:

$$\rho(x, y) \langle \mathbf{n}(x, y), \ell \rangle = m(x, y) \quad (1.1)$$

Dove:

1. $\rho(x, y)$ è detto albedo e rappresenta la percentuale di luce riflessa dal punto (x, y) .

2. $\mathbf{n}(x, y)$ è il vettore normale alla superficie nel punto considerato
3. $m(x, y)$ è la radiazione riflessa dal punto quando è illuminata da una luce in direzione ℓ

Fissata una fonte luminosa, essa avrà direzione

$$\ell = \begin{pmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{pmatrix} \stackrel{\text{not}}{=} \begin{bmatrix} \tilde{\ell} \\ \ell_3 \end{bmatrix}$$

Per cui posta la superficie come $z = u(x, y)$, ciò che si vuole trovare è la quota z .

Si può notare che non si sta cercando di ricostruire una generica superficie, ma solo quelle esprimibili come grafici di funzione. In tal modo la superficie M si può parametrizzare come:

$$M = \{(x, y, z) | z = u(x, y)\} \quad (1.2)$$

Per cui il campo normale alla superficie si scrive come:

$$\mathbf{n}(x, y) = \frac{1}{\sqrt{1 + |\nabla u(x, y)|^2}} (-\nabla u(x, y), 1)^T \quad (1.3)$$

Dove $\nabla u(x, y) = \left(\frac{\partial u(x, y)}{\partial x}, \frac{\partial u(x, y)}{\partial y} \right)^T$

È noto che il problema può essere modellato dall'equazione d'irradianza

$$R(\mathbf{n}(x, y)) = I(x, y) \quad (1.4)$$

Dove $R(\mathbf{n}(x, y))$ indica la funzione di riflettanza, che rappresenta il valore della luce riemessa dalla superficie come funzione della sua orientazione, e $I(x, y)$ è l'irradianza che rappresenta il livello di grigio misurato nel punto (x, y) dell'immagine.

Una volta ricavate le immagini, ciò che si conosce è l'irradianza $I(x, y)$ poichè in termini di scala di grigio, ad ogni pixel dell'immagine viene associato un numero compreso tra 0 e 1, i cui estremi rappresentano rispettivamente il colore nero e bianco.

Per costruire un modello continuo si considera $I(x, y)$ come funzione $I : \bar{\Omega} \mapsto [0, 1]$ e la superficie come $u : \bar{\Omega} \mapsto \mathbb{R}$, dove $\bar{\Omega} \subset \mathbb{R}^2$ è detto dominio di ricostruzione, e nel caso considerato non è altro che un rettangolo dato che la macchina fotografica non cambia posizione.

Per cui, in virtù della (1.1) si ha che la (1.4) diventa:

$$R(\mathbf{n}(x, y)) = \rho(x, y) \langle \boldsymbol{\ell}, \mathbf{n}(x, y) \rangle \quad (1.5)$$

Da cui, usando la (1.3) si ha che:

$$I(x, y) = \frac{\rho(x, y)}{\sqrt{1 + |\nabla u(x, y)|^2}} (-\langle \tilde{\boldsymbol{\ell}}, \nabla u(x, y) \rangle + \ell_3)$$

E posto $\rho = 1$, l'equazione si riscrive come:

$$\sqrt{1 + |\nabla u(x, y)|^2} I(x, y) + \langle \tilde{\boldsymbol{\ell}}, \nabla u(x, y) \rangle - \ell_3 = 0 \quad (1.6)$$

Essa è un'equazione di Hamilton–Jacobi, e siccome dipende solo da $\nabla u(x, y)$, se u è una soluzione, lo è anche $u + c$ con c costante. Ciò implica che per avere un problema ben posto occorre aggiungere le condizioni al contorno su $\partial\Omega$.

Se al posto di avere un'unica immagine illuminata da una luce in direzione $\boldsymbol{\ell}$, si avessero n immagini illuminate dalla luce nelle direzioni $\boldsymbol{\ell}_i$ con $i = 1, \dots, n$, il problema si riscriverebbe come:

$$\begin{cases} H_i(x, y, \nabla u(x, y)) = 0 & \text{con } i = 1, \dots, n \\ u(x, y) = g(x, y) & \forall (x, y) = \partial\Omega \end{cases} \quad (1.7)$$

Dove H è l'Hamiltoniana del problema.

Per maggiori dettagli si veda [6].

1.2 Metodo alle differenze finite

In generale, un'equazione ellittica è un'equazione alle derivate parziali della forma:

$$u_{xx} + u_{yy} + p(x, y)u_x + q(x, y)u_y + r(x, y)u + s(x, y) = 0$$

Da notare che il termine u_{xy} non è presente in quanto per un'opportuna rotazione delle variabili può essere annullata.

Si parlerà di problema di Cauchy se all'equazione vengono associate delle condizioni iniziali generiche, un cui caso particolare è dato dal Problema di Dirichlet, ossia:

Definizione 1.1 (Problema di Dirichlet). Un problema di Cauchy è detto di Dirichlet se le condizioni iniziali sono date sulla frontiera del dominio di f .

In formula:

$$\begin{aligned} u_{xx} + u_{yy} + p(x, y)u_x + q(x, y)u_y + r(x, y)u + s(x, y) &= 0 \\ u(x, y) &= g_1(x, y) \quad (x, y) \in \partial\Omega \end{aligned} \quad (1.8)$$

Dato il problema (1.8) con le funzioni $p(x, y)$, $q(x, y)$, $r(x, y)$, $s(x, y)$ e $g_1(x, y)$ continue, si consideri come dominio $\Omega = [a, b] \times [c, d]$, e ciò è chiaro in quanto nel problema della Photometric Stereo la macchina fotografica non sta cambiando posizione per cui la regione dei punti interessata rappresenta un rettangolo.

Siccome il dominio è rettangolare, il modo migliore per risolvere il problema di Dirichlet è quello di usare il metodo alle differenze finite, il quale procede come segue.

Si discretizza il dominio nella seguente maniera.

$$\begin{aligned} x_i &= a + ih \text{ con } i = 0, \dots, n + 1 \text{ e } h = \frac{b-a}{n-1} \\ y_j &= b + jk \text{ con } j = 0, \dots, m + 1 \text{ e } h = \frac{d-c}{m-1} \end{aligned}$$

Per cui, una volta discretizzato il dominio, si deve discretizzare anche le derivate seconde. In generale, l'approssimazione della derivata prima tramite le differenze centrali è data da:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$$

e quindi la derivata seconda si discretizzerà nella seguente maniera:

$$f''(x_i) = \frac{1}{h} \left(\frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_i) - f(x_{i-1}))}{h} \right) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2}$$

Questa prende il nome di differenza centrale del secondo ordine, e si dimostra che:

$$f''(x_i) - \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} = O(h^2)$$

Posto $u_{i,j} = u(x_i, y_j)$, $p_{i,j} = p(x_i, y_j)$, $q_{i,j} = q(x_i, y_j)$, $r_{i,j} = r(x_i, y_j)$, $s_{i,j} = s(x_i, y_j)$ e $g_{1i,j} = g_1(x_i, y_j)$ il problema (1.8) diventa:

$$\begin{aligned} \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{k^2} + p_{i,j} \frac{u_{i+1,j} - u_{i-1,j}}{2h} + q_{i,j} \frac{u_{i,j+1} - u_{i,j-1}}{2k} + \\ + r_{i,j}u_{i,j} + s_{i,j} = 0 \text{ con } i = 1, \dots, n \text{ e } j = 1, \dots, m \end{aligned}$$

E ordinando si ricava che:

$$c_{i,j}u_{i,j-1} + b_{i,j}u_{i-1,j} - 2a_{i,j}u_{i,j} + \tilde{b}_{i,j}u_{i+1,j} + \tilde{c}_{i,j}u_{i,j+1} = -2h^2k^2s_{i,j}$$

con $i = 1, \dots, n$ e $j = 1, \dots, m$

Dove:

1. $c_{i,j} = h^2(2 - kq_{i,j})$
2. $b_{i,j} = k^2(2 - hp_{i,j})$
3. $a_{i,j} = 2(h^2 + k^2) - h^2k^2r_{i,j}$
4. $\tilde{b}_{i,j} = k^2(2 + hp_{i,j})$
5. $\tilde{c}_{i,j} = h^2(2 + kq_{i,j})$

Per cui, tramite la discretizzazione, si è passati dallo studio di un'equazione differenziale alle derivate parziali a quello di un sistema lineare, la cui matrice associata è:

$$\begin{pmatrix} a_{1,1} & \tilde{b}_{1,1} & 0 & 0 & \tilde{c}_{1,1} & & \\ b_{2,1} & a_{2,1} & \tilde{b}_{2,1} & & & \ddots & \\ 0 & \ddots & \ddots & \ddots & & & \\ 0 & & & & & & \\ c_{1,2} & & & & & & \\ & \ddots & & & & & \tilde{b}_{3,2} \\ & & c_{4,3} & 0 & 0 & b_{4,3} & a_{4,3} \end{pmatrix}$$

In altre parole si ha un sistema lineare di ordine $m \times n$ la cui matrice associata è pentadiagonale. Per una lettura approfondita sull'argomento si veda [7].

1.3 Metodo ai minimi quadrati

Spesso nelle applicazioni si incontrano sistemi lineari sovradeterminati della forma $Ax = b$ con $A \in \mathbb{R}^{m \times n}$ e $m > n$.

Questo problema è mal posto in quanto essendoci più equazioni che incognite, il problema potrebbe non avere alcuna soluzione. Inoltre, se la matrice A è stata costruita tramite dati raccolti attraverso misurazioni, allora la soluzione esatta, che si sa che esiste in quanto rappresenta un

fenomeno fisico, potrebbe non soddisfare alcuna equazione del sistema per via degli errori sperimentali introdotti.

Per cui, un modo per passare da un problema malposto ad un altro ben posto è quello di ricorrere al metodo dei minimi quadrati, ossia si cerca quel vettore x tale che:

$$\min_{x \in S} \|Ax - b\|_2 \text{ con } S = \{x \in \mathbb{R} \mid \|Ax - b\|_2 \text{ è minima}\} \quad (1.9)$$

Intuitivamente è chiaro. Dato che vengono introdotti degli errori sperimentali, per ogni x si ha che $Ax = b + \varepsilon$, dove ε dipende dalla varianza dei dati raccolti. Per cui, dato che l'errore sperimentale è sempre presente, risulta ragionevole minimizzare quest'ultima per realizzare una pulizia statistica dei dati raccolti e quindi si ha che:

$$\min_x |\varepsilon| = \min_x \|Ax - b\|_2$$

e tra tutti i vettori che soddisfano la formula, siccome la soluzione dev'essere unica, si prende quello di lunghezza minima. Il vettore x^* che soddisfa il problema (1.9) è detto soluzione nel senso dei minimi quadrati.

Per trovare la soluzione nel senso dei minimi quadrati si può ricorrere alla fattorizzazione SVD, ossia:

Teorema 1.3.1. *Sia $A \in \mathbb{C}^{m \times n}$ di rango r allora esistono due matrici unitarie $U \in \mathbb{C}^{m \times m}$ e $V \in \mathbb{C}^{n \times n}$, una matrice $\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix}$ con $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$ con $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ in modo tale che:*

$$A = U\Sigma V^H \quad (1.10)$$

Se la matrice $A \in \mathbb{R}^{m \times n}$, allora U e V sono ortogonali. La (1.10) è detta fattorizzazione ai valori singolari (SVD) della matrice A .

Si hanno le seguenti definizioni:

1. Le σ_i sono detti valori singolari.
2. Posto $U = (u_1, \dots, u_m)$ e $V = (v_1, \dots, v_n)$, u_i e v_i sono detti rispettivamente vettori singolari sinistri e destri associati al valore singolare σ_i .

Da notare che la fattorizzazione SVD è una generalizzazione della fattorizzazione spettrale. Infatti, se A è diagonalizzabile, allora vale la seguente: $A = U^T \Lambda U$ con U e Λ matrici rispettivamente ortogonale e diagonale, e quindi posto $V = U$, si ritrova la (1.10).

Il collegamento tra fattorizzazione ai valori singolari e il problema ai minimi quadrati è dato dal seguente teorema:

Teorema 1.3.2. Data la matrice $A \in \mathbb{C}^{m \times n}$ e $\text{rg}(A) = r$ e fissata la fattorizzazione $A = U\Sigma V^H$, allora il problema ai minimi quadrati (1.9) ammette un'unica soluzione data da:

$$x = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H b$$

Dimostrazione. Detto $z = V^H x = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ e $c = U^H b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ ho che:

$$\begin{aligned} \|b - Ax\|_2 &= \left\| U^H (b - AVV^H x) \right\|_2 = \left\| \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} - \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|_2 = \\ &= \left\| \begin{pmatrix} c_1 - \Sigma_1 z_1 \\ c_2 \end{pmatrix} \right\|_2 \end{aligned}$$

Per cui la norma è minima per $z_1 = \Sigma_1^{-1} c_1$. Inoltre, siccome si vuole minimizzare $x = Vz$ allora $z_2 = 0$. Da cui:

$$x = Vz = V \begin{pmatrix} \Sigma_1^{-1} c_1 \\ 0 \end{pmatrix} = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H b$$

□

Se si dá la seguente

Definizione 1.2. Si definisce pseudoinversa della matrice $A \in \mathbb{C}^{m \times n}$ la matrice $A^\dagger = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H$

Per cui, dato un sistema lineare $Ax = b$, il teorema precedente afferma che la soluzione del sistema è data da $x^* = A^\dagger b$. Un'esposizione più approfondita è presente in [8].

Capitolo 2

Algoritmo alla Poisson

In questo capitolo verrà proposto l'algoritmo alla Poisson per la ricostruzione 3-D di una superficie Lambertiana. Questo metodo è suddiviso in due parti.

Nella prima parte si estraggono le informazioni fotometriche dell'oggetto, ricostruendo il campo delle normali alla superficie.

Nella seconda, una volta forniti i dati fotometrici, si ricostruisce la superficie risolvendo un problema differenziale.

La peculiarità dell'algoritmo è che, dato il problema $RN^T L = M$, il campo normale N viene visto nella prima fase semplicemente come una matrice incognita e non come entità matematica derivante da una superficie e quindi dipendente solo da 2 variabili. Questa considerazione, perciò, obbliga ad avere a disposizione almeno 3 immagini dell'oggetto da ricostruire in quanto $N \in \mathbb{R}^{3 \times q}$.

Di seguito verrà mostrato in dettaglio l'algoritmo.

2.1 Costruzione del campo normale

La legge di Lambert si può riscrivere matricialmente come

$$RN^T L = M \tag{2.1}$$

Dove:

1. $R = \text{diag}(\rho_1, \dots, \rho_p)$ è l'albedo
2. $N = (n_{i,j}) = [\mathbf{n}_1, \dots, \mathbf{n}_p] \in \mathbb{R}^{3 \times p}$ è il campo normale
3. $L = [\ell_1, \dots, \ell_q] \in \mathbb{R}^{3 \times q}$ è la matrice contenente la posizione delle luci.

4. $M = [\mathbf{m}_1, \dots, \mathbf{m}_q] \in \mathbb{R}^{p \times q}$ rappresenta l'insieme delle immagini

M è stata costruita sfruttando l'ordine lessicografico, ossia:

date le immagini M_1, \dots, M_q , alla generica $M_j \in \mathbb{R}^{r \times s}$ si associa il vettore

$$\mathbf{m}_j = [M_{j1}^T, \dots, M_{js}^T]^T \in \mathbb{R}^p$$

dove M_{j1}, \dots, M_{js} indicano le colonne della matrice M_j e $p = rs$. In altre parole non si è fatto altro che riordinare i pixel dell'immagine M_j seguendo l'ordine delle colonne.

Segue quindi che $M = [\mathbf{m}_1, \dots, \mathbf{m}_p]$.

Da notare che senza l'ordine lessicografico, al punto (x_i, y_j) sono associati il valore di irradianza m_{ij} , il punto della superficie u_{ij} e le derivate parziali $(u_x)_{ij}$ e $(u_y)_{ij}$.

Invece, seguendo l'ordine lessicografico, al pixel k -esimo vengono associati il valore m_k , il punto della superficie u_k e le derivate parziali $(u_x)_k$ e $(u_y)_k$.

Per favorire una lettura più agevole del testo, di seguito si userà l'indice k laddove è presente l'ordine lessicografico e gli indici i, j quando non è presente

Un altro aspetto da rilevare è che R è una matrice diagonale in quanto, per via della legge di Lambert, i ρ_k rappresentano un fattore di scala dipendente solo dalla posizione delle luci.

Partendo da queste considerazioni, si suppone di avere a disposizione tre immagini e di conoscere la matrice rappresentante la posizione delle luci $L \in \mathbb{R}^{3 \times 3}$ così come la matrice delle immagini $M \in \mathbb{R}^{p \times 3}$, dove p indica il numero di pixel delle singole immagini.

L'obbiettivo di questa sezione è quello di ricostruire la matrice del campo delle normali $N \in \mathbb{R}^{3 \times p}$, supposto di avere incognita la matrice dell'albedo $R \in \mathbb{R}^{p \times p}$.

Se per notazione si pone che $\tilde{N}^T := RN^T$, la (2.1) diventa:

$$\tilde{N}^T L = M \tag{2.2}$$

Per cui si hanno due casi:

1. La matrice L è singolare, il che implica che le luci sono complanari. Fisicamente questo risultato si verifica se le luci ammettono la stessa direzione e variano solo di intensità luminosa, oppure se le luci provengono da un fonte luminosa che si muove su un piano che contiene l'oggetto osservato.
2. L non è singolare. In tal caso il sistema si risolve come $\tilde{N}^T = ML^{-1}$.

Una volta trovato $\tilde{N} = [\tilde{\mathbf{n}}_1 \ \dots \ \tilde{\mathbf{n}}_p]$ si ricava la matrice N normalizzando i vettori $\tilde{\mathbf{n}}_i$ a patto che non esista alcun vettore $\tilde{\mathbf{n}}_i$ nullo. In virtù della (2.1) ciò si verifica se:

$$\mathbf{0} = \tilde{\mathbf{n}}_i^T = L^{-1}(m^i)^T$$

dove m^i rappresenta la riga i -esima della matrice M .

L'ultima uguaglianza si può verificare se:

1. $m^i = \mathbf{0}$, ossia se esistono stesse zone d'ombra nelle immagini.
2. posto di chiamare $L^{-1} = [\bar{\mathbf{I}}_1 \ \bar{\mathbf{I}}_2 \ \bar{\mathbf{I}}_3]$, se $\mathbf{0} = \bar{\mathbf{I}}_1 m_{i,1} + \bar{\mathbf{I}}_2 m_{i,2} + \bar{\mathbf{I}}_3 m_{i,3}$
Ossia, se le colonne della matrice L^{-1} sono linearmente dipendenti, ma ciò è assurdo in quanto L^{-1} è invertibile.

Avendo trovato il campo normale, si può ricostruire anche l'albedo. Infatti, data la relazione $RN^T = \tilde{N}^T$, la riga i -esima si scrive come:

$$\rho_i \mathbf{n}_i^T = \tilde{\mathbf{n}}_i^T$$

passando alle norme, si ha che:

$$\rho_i = |\rho_i| = \|\rho_i \mathbf{n}_i^T\| = \|\tilde{\mathbf{n}}_i^T\| = \|\tilde{\mathbf{n}}_i\|$$

Se invece si hanno più di tre immagini allora il problema (2.2) si risolve ricorrendo alla matrice pseudoinversa come:

$$\tilde{N}^T = ML^\dagger$$

Il ricorso alla matrice pseudoinversa non crea alcun problema, poiché se la matrice è invertibile, allora $L^\dagger = L^{-1}$. Per quanto riguarda la costruzione dell'albedo, il procedimento è analogo a prima.

2.2 Problema differenziale

Una volta aver ricavato il campo N delle normali, si vuole ricostruire la superficie e per far ciò occorre risolvere l'equazione:

$$\Delta u(x, y) = g(x, y)$$

Dove $\Delta u(x, y) = u_{xx} + u_{yy}$ indica il laplaciano della funzione $u = u(x, y)$.

La risoluzione dell'equazione è chiara, in quanto si sta cercando la superficie $z = u(x, y)$ e dato che la normale è uguale a

$$\mathbf{n} = (-p, -q, 1)^T = \left(-\frac{\partial u}{\partial x}, -\frac{\partial u}{\partial y}, 1 \right)^T$$

allora l'equazione si riscrive come:

$$p_x + q_y = g(x, y)$$

Per cui, discretizzando il dominio si riesce a ricavare il termine noto $g(x, y)$, e applicando il metodo delle differenze finite si risolve l'equazione differenziale.

Di seguito verranno analizzate alcune proprietà analitiche dell'equazione, il metodo per la risoluzione numerica di queste e infine alcune prove numeriche.

2.2.1 Aspetti teorici dell'equazione di Laplace

L'equazione

$$\Delta u(x, y) = g(x, y) \tag{2.3}$$

è detta equazione di Poisson ed è un'equazione alle derivate parziali in quanto la funzione incognita compare derivata due volte rispetto ad entrambe le variabili.

Ecco un esempio:

Esempio 2.2.1. Dato il problema di Cauchy

$$u_{xx} + u_{yy} = 0$$

$$u(x, 0) = 0 \quad u_y(x, 0) = 0$$

Si dimostra che la soluzione del problema è data da

$$u(x, y) = 0 \quad \forall y > 0$$

Tuttavia, se si cambiano le condizioni iniziali come:

$$u(x, 0) = 0 \quad u_y(x, 0) = 10^{-4} \sin(10^4 x)$$

Allora la soluzione diventa

$$u(x, y) = 10^{-8} \sin(10^4 x) \sinh(10^4 y)$$

E per y sufficientemente grande $\sinh(10^4 y) \simeq e^{10^4 y}$ e quindi la soluzione cresce esponenzialmente.

L'esempio precedente mostra che il problema di Cauchy in generale è mal posto rispetto alle condizioni iniziali, e ciò suggerisce di risolvere l'equazione di Poisson utilizzando condizioni al contorno. Quest'idea è avvalorata dalla seguente proposizione:

Proposizione 2.2.1. *Se Ω è sufficientemente regolare e g_1 è continua su $\partial\Omega$ allora il problema di Dirichlet ammette una ed una sola soluzione e questa dipende con continuità dai dati*

Per continuità dai dati si intende che se una successione $\{g^n\}$ converge uniformemente a zero allora la successione delle soluzioni corrispondenti $\{u^n\}$ converge a zero.

Dal punto di vista pratico, la conoscenza delle condizioni al contorno può essere ricavata essenzialmente in due modi:

1. Applicando uno scanner laser nei punti corrispondenti al bordo delle immagini.
2. Applicando una maschera di nero nel contorno dell'immagine tramite un programma di elaborazione di immagini.

Entrambe le tecniche hanno i loro pro e contro.

La prima conserva la continuità della funzione ma presenta il grosso svantaggio del fatto che lo scanner laser sia uno strumento molto costoso e lento.

Nel secondo metodo, aggiungere una maschera di nero equivale a imporre che sulla frontiera la funzione valga zero, e ciò garantisce che questa sia una tecnica sempre applicabile. Tuttavia, in questo modo si introducono dei punti di discontinuità sulla funzione, e ciò si ripercuote sulla differenziazione numerica, come vedremo dagli esperimenti numerici.

2.2.2 Risoluzione mediante le differenze finite

Dato il problema (1.8) con $g(x, y)$ e $g_1(x, y)$ continue, si consideri come dominio $\Omega = [a, b] \times [c, d]$, e come è stato già detto, ciò è chiaro in quanto nel problema della Photometric Stereo la macchina fotografica è fissa per cui la regione dei punti interessata rappresenta un rettangolo.

Discretizzando l'operatore di Laplace mediante la formula a cinque punti, come è stato visto nel capitolo 1, si ricava che al problema di Dirichlet è associato il seguente sistema lineare:

$$u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i,j+1} + u_{i+1,j} = h^2 g(i, j) \quad (2.4)$$

Dove $i=1,\dots,n$ e $j=1,\dots,m$

Di seguito si cercherà di vedere come sono fatti il termine noto e la matrice del sistema.

Denotate le condizioni al contorno come:

$$u_{i,0} = u(x_i, y_0)$$

$$u_{i,m} = u(x_i, y_m)$$

$$u_{0,j} = u(x_0, y_j)$$

$$u_{n,j} = u(x_n, y_j)$$

Si pone

$$\mathbf{u}_i = \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ \vdots \\ u_{i,m} \end{bmatrix} \text{ con } i = 1, \dots, n \text{ e } \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}^T \in \mathbb{R}^{nm}$$

E posto

$$\mathbf{f}_i = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,m} \end{bmatrix} \text{ con } i = 1, \dots, n \text{ e } \mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} \in \mathbb{R}^{nm}$$

Il sistema (2.4) si riscrive come:

$$I\mathbf{u}_{i-1} + T\mathbf{u}_i + I\mathbf{u}_{i+1} = \tilde{\mathbf{f}}_i \text{ con } i = 1, \dots, n \quad (2.5)$$

$$\text{dove } T = \begin{bmatrix} -4 & 1 & & 0 \\ 1 & -4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -4 \end{bmatrix} \text{ e } \tilde{\mathbf{f}}_i = \begin{bmatrix} h^2 f_{i,1} - u_{i,0} \\ h^2 f_{i,2} \\ \vdots \\ h^2 f_{i,m-1} \\ h^2 f_{i,m} - u_{i,n+1} \end{bmatrix}$$

Per cui, sviluppando la (2.5) si ha che:

$$T\mathbf{u}_1 + I\mathbf{u}_2 = \tilde{\mathbf{f}}_1 - \mathbf{u}_0$$

$$I\mathbf{u}_{i-1} + T\mathbf{u}_i + I\mathbf{u}_{i+1} = \tilde{\mathbf{f}}_i \text{ con } i = 2, \dots, n-1$$

$$I\mathbf{u}_{n-1} + T\mathbf{u}_n = \tilde{\mathbf{f}}_n - \mathbf{u}_{n+1}$$

Ossia, la matrice associata all'equazione di Poisson discretizzata é data da:

$$A = \begin{bmatrix} T & I & & 0 \\ I & T & I & \\ & \ddots & \ddots & I \\ 0 & & I & T \end{bmatrix}$$

La matrice A gode di alcune proprietà:

- È simmetrica, infatti T e I sono simmetriche e sono disposte simmetricamente su A .
- È una matrice BTTB (*Block Toeplitz with Toeplitz blocks*), in quanto T e I sono matrici di Toeplitz.
- È pentadiagonale, in quanto T è tridiagonale e I è diagonale. Ciò implica che A è una matrice fortemente sparsa, in quanto, posto n la sua dimensione, contiene meno di $5n$ entrate non nulle, contro un totale di n^2 entrate.
- È definita negativa.

Per dimostrare quest'ultima proprietà verranno enunciati alcuni teoremi:

Teorema 2.2.2 (Primo teorema di Gerschgorin). *Definiti $\mathcal{R}_i = \{z \in \mathbb{C} \mid |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|\}$ con $i = 1, \dots, n$ i cerchi di riga, allora lo spettro di A è contenuto nell'unione dei cerchi di riga. Ossia,*

$$\sigma(A) \subseteq \bigcup_{i=1}^n \mathcal{R}_i = S_R$$

Definizione 2.1 (Matrice riducibile). Una matrice quadrata A di dimensione n si dice riducibile, se esiste una matrice di permutazione P tale che

$$PAP^T = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}$$

dove B_{11} e B_{22} sono matrici quadrate rispettivamente di dimensione $k \times k$ e $(n - k) \times (n - k)$.

Una matrice è irriducibile se non è riducibile.

Dalla sola definizione è difficile capire se una matrice è riducibile oppure no, per cui è necessario il seguente teorema:

Proposizione 2.2.3. *Una matrice quadrata A di dimensione n è irriducibile se e solo se esiste una catena per ogni coppia di indici (i, j) , cioè se per ogni indice (i, j) è possibile trovare una sequenza di elementi non nulli della matrice A del tipo*

$$a_{i,k_1}, a_{k_1,k_2}, \dots, a_{k_{n-1},k_n}, a_{k_n,j}$$

La proprietà di irriducibilità è utile per dimostrare il seguente teorema:

Teorema 2.2.4 (Terzo teorema di Gerschgorin). *Sia A una matrice irriducibile, se un suo autovalore appartiene alla frontiera di S_R , esso deve appartenere alla frontiera di ciascun cerchio \mathcal{R}_i*

Per cui ora si può dimostrare il seguente:

Proposizione 2.2.5. *La matrice associata all'equazione di Poisson è:*

1. *irriducibile*
2. *definita negativa*

Dimostrazione. Per quanto riguarda il primo punto occorre sfruttare la proposizione 2.2.3.

Dati due indici qualunque i e j ho tre casi:

1. $i=j$, per cui siccome in A $a_{i,i} \neq 0$ allora la catena si riduce alla coppia $(a_{i,i}, a_{i,i})$
2. $i>j$, in questo caso essendo le entrate $a_{i,i-1} \neq 0 \quad \forall i = 2, \dots, n$ allora si riesce a costruire la catena $(a_{i,i-1}, a_{i-1,i-2}, \dots, a_{j+2,j+1}, a_{j+1,j})$
3. $i<j$, per cui dato che $a_{i,i+1} \neq 0 \quad \forall i = 2, \dots, n$, allora si costruisce la catena $(a_{i,i+1}, a_{i+1,i+2}, \dots, a_{j-2,j-1}, a_{j-1,j})$

E siccome in tutti i tre casi si è riusciti a trovare una catena che colleghi a_i con a_j , allora la matrice è irriducibile.

Per il secondo punto, siccome la matrice associata all'equazione di Poisson è simmetrica, gli autovalori sono tutti reali, e sfruttando il Primo teorema di Gerschgorin, si riesce a costruire dei cerchi concentrici di centro -4 e raggio pari a $2, 3$ e 4 .

Per cui gli autovalori sono compresi nell'intervallo $(-8, 0)$ e questo intervallo è proprio aperto in virtù del fatto che la matrice è irriducibile, e quindi per il terzo teorema di Gerschgorin, 0 non può essere un autovalore in quanto non appartiene a tutti i cerchi \mathcal{R}_i . E siccome una matrice è definita negativa se e solo se tutti i suoi autovalori sono negativi, allora abbiamo dimostrato il teorema. \square

Tutte queste proprietà relative alla matrice associata all'equazione di Poisson sono molto utili per quanto riguarda la risoluzione del sistema lineare. Infatti, siccome la matrice è sparsa ed è definita negativa, la prima

scelta che si può compiere è quella di utilizzare il metodo del gradiente preconditionato, in quanto è noto che mantiene la struttura della matrice e converge per matrici definite positive. Per cui, anzichè risolvere il sistema

$$Ax = f$$

si dovrà risolvere il sistema

$$-Ay = f$$

dove $y = -x$

Per quanto riguarda il preconditionatore, una buona scelta è data dalla fattorizzazione di Cholesky incompleta, essendo la matrice $-A$ simmetrica e definita positiva.

2.3 Implementazione dell'algoritmo

Una volta analizzati il passo di ricostruzione della normale, e il successivo passo di risoluzione dell'equazione di Poisson, è possibile descrivere l'intero algoritmo come segue:

Algoritmo alla Poisson

Input :

M insieme delle immagini

L matrice contenente la posizione delle luci

h passo di discretizzazione

Output :

Z superficie ricostruita

N campo delle normali

R albedo

Passo 0 : Se h non è noto, ricostruirlo. Ciò è possibile ottenerlo inserendo un righello all'interno dell'ambiente della fotografia e così si può trovare h in base a come viene riprodotto lo strumento.

Passo 1 : Noti h , L e M , sfruttando la function **normale** si ricostruiscono il campo normale N e l'albedo R .

Passo 2 : Sfruttando un'opportuna function (vedi **Poissolve** o **Poissolvedue**) si risolve l'equazione di Poisson e così si trova la soluzione Z .

Dove la function **normale** presente al passo 1 è la seguente:

Normale

Input :

M insieme delle immagini

L matrice contenente la posizione delle luci

Output :

N campo delle normali

R albedo

Passo 0 : Si calcola $N = (ML^+)^T$.

Passo 1 : Si costruisce $R = \text{diag}(r_1, \dots, r_p)$ dove $r_i = \text{norm}(N(:, i))$ per ogni i .

Passo 2 : Si aggiorna $N = [\mathbf{n}_1 \ \dots \ \mathbf{n}_p]$ dove $\mathbf{n}_i = \frac{\mathbf{n}_i}{r_i}$ per ogni i

Siccome la matrice associata è fortemente sparsa, simmetrica e definita negativa, l'algoritmo di risoluzione del sistema lineare più naturale è quello del gradiente preconditionato tramite la fattorizzazione di Cholesky incompleta. Per cui è stata costruita la function **Poissolve** sfruttando questo algoritmo, ed è composta come segue:

Poissolve

Input :

F termine noto

h passo di discretizzazione

count condizioni al contorno

tol valore di tolleranza

nmax numero massimo di iterazioni

Output :

U soluzione del sistema

A matrice associata al sistema

Passo 0 : viene costruita la matrice di Poisson, e per far ciò si segue questo ragionamento:

$$A = \begin{bmatrix} T & I & & 0 \\ I & T & & I \\ & \ddots & \ddots & I \\ 0 & & I & T \end{bmatrix}$$

$$\text{Dove } T = \begin{bmatrix} -4 & 1 & & 0 \\ 1 & -4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -4 \end{bmatrix} \text{ e } S = \begin{bmatrix} 0 & 1 & & 0 \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 0 \end{bmatrix}$$

Per cui usando il prodotto di Kronecker, A si riscrive come:

$$A = I \otimes T + S \otimes I$$

Passo 1 : Note le condizioni al contorno, viene modificato il termine noto.

Passo 2 : Tramite le funzioni preimpostate di Matlab viene generato il preconditionatore R e a partire da questo si risolve il sistema col metodo del gradiente preconditionato.

e la function **Poissolvedue** è uguale a **Poissolve** se non per il Passo 2 in cui viene sfruttato il comando backslash di Matlab,, il quale è in grado di riconoscere la sparsità della matrice sfruttando il pacchetto UMFPACK (*Unsymmetric MultiFrontal Package*). Gli algoritmi di questo pacchetto sono dei metodi diretti per matrici sparse, e siccome la matrice di Poisson è fortemente sparsa, questi algoritmi risultano essere più veloci rispetto ai metodi iterativi per dimensioni sufficientemente grandi.

Capitolo 3

Algoritmo alla Hamilton–Jacobi

Nel seguente capitolo verrà proposto un algoritmo per la ricostruzione 3-D di una superficie Lambertiana. In particolare, verrà studiato il problema nel caso continuo, e da questo verrà discretizzato per ricavare l'algoritmo.

3.1 Caso continuo con due immagini

Come si è visto nel capitolo 1, sotto opportune ipotesi il problema di Photometric Stereo per superfici Lambertiane con sole due immagini si può formulare matematicamente, senza tener conto delle condizioni al contorno, come:

$$\begin{cases} \rho(x, y) \frac{\langle -\nabla u(x, y), \tilde{\ell}_1 \rangle + \ell_{1,3}}{\sqrt{1 + |\nabla u(x, y)|^2}} = I_1(x, y) \\ \rho(x, y) \frac{\langle -\nabla u(x, y), \tilde{\ell}_2 \rangle + \ell_{2,3}}{\sqrt{1 + |\nabla u(x, y)|^2}} = I_2(x, y) \end{cases} \quad (3.1)$$

dove

$$\nabla u(x, y) = \left[\frac{\partial u(x, y)}{\partial x} \quad \frac{\partial u(x, y)}{\partial y} \right]^T$$

è il gradiente della funzione $z = u(x, y)$ e i vettori che individuano le luci sono

$$\begin{aligned} \ell_1 &= (\ell_{1,1}, \ell_{1,2}, \ell_{1,3})^T := (\tilde{\ell}_1, \ell_{1,3})^T \\ \ell_2 &= (\ell_{2,1}, \ell_{2,2}, \ell_{2,3})^T := (\tilde{\ell}_2, \ell_{2,3})^T \end{aligned}$$

Nell'ipotesi che la posizione delle luci sia nota, Kozera [9] ha trovato i seguenti risultati.

Proposizione 3.1.1. Per ogni punto del dominio Ω le derivate prime di u possono essere espresse in termini di I_1, I_2, ℓ_1 e ℓ_2 nel seguente modo:

$$u_x^\pm = \frac{(\ell_{2,1}\langle \ell_1, \ell_2 \rangle - \ell_{1,1})I_1 + (\ell_{1,1}\langle \ell_1, \ell_2 \rangle - \ell_{2,1})I_2 \pm (\ell_{1,3}\ell_{2,2} - \ell_{1,2}\ell_{2,3})\varepsilon\sqrt{\Lambda}}{(\ell_{1,3} - \ell_{2,3}\langle \ell_1, \ell_2 \rangle)I_1 + (\ell_{2,3} - \ell_{1,3}\langle \ell_1, \ell_2 \rangle)I_2 \pm (\ell_{1,1}\ell_{2,2} - \ell_{1,2}\ell_{2,1})\varepsilon\sqrt{\Lambda}}$$

$$u_y^\pm = \frac{(\ell_{2,2}\langle \ell_1, \ell_2 \rangle - \ell_{1,2})I_1 + (\ell_{1,2}\langle \ell_1, \ell_2 \rangle - \ell_{2,2})I_2 \pm (\ell_{1,1}\ell_{2,3} - \ell_{1,3}\ell_{2,1})\varepsilon\sqrt{\Lambda}}{(\ell_{1,3} - \ell_{2,3}\langle \ell_1, \ell_2 \rangle)I_1 + (\ell_{2,3} - \ell_{1,3}\langle \ell_1, \ell_2 \rangle)I_2 \pm (\ell_{1,1}\ell_{2,2} - \ell_{1,2}\ell_{2,1})\varepsilon\sqrt{\Lambda}}$$

Dove

$$\Lambda = \Lambda(x, y) = 1 - I_1^2(x, y) - I_2^2(x, y) - \langle \ell_1, \ell_2 \rangle [\langle \ell_1, \ell_2 \rangle - 2I_1(x, y)I_2(x, y)]$$

e $\varepsilon = \varepsilon(x, y)$ e può assumere solo i valori ± 1 in modo tale che $f(x, y) = \varepsilon(x, y)\sqrt{\Lambda(x, y)}$

Da questo lemma segue il seguente:

Teorema 3.1.2. Sia Ω un sottoinsieme di \mathbb{R}^2 semplicemente connesso e siano I_1 e $I_2 \in C^1(\Omega)$. Supposto che $\Lambda > 0$ su Ω e per ogni scelta del segno \pm si ha che

$$\sigma^\pm = (\ell_{1,3} - \ell_{2,3}\langle \ell_1, \ell_2 \rangle)I_1 + (\ell_{2,3} - \ell_{1,3}\langle \ell_1, \ell_2 \rangle)I_2 \pm (\ell_{1,1}\ell_{2,2} - \ell_{1,2}\ell_{2,1})\varepsilon\sqrt{\Lambda}$$

non si annulla su Ω . Allora una condizione necessaria per l'esistenza di esattamente due soluzioni di classe C^2 è che

$$\frac{\partial}{\partial y}u_x^+(x, y) = \frac{\partial}{\partial x}u_y^+(x, y) \quad \text{o} \quad \frac{\partial}{\partial y}u_x^-(x, y) = \frac{\partial}{\partial x}u_y^-(x, y)$$

Corollario 3.1.3. Sia Ω un sottoinsieme di \mathbb{R}^2 semplicemente connesso e siano I_1 e $I_2 \in C^1(\Omega)$. Supposto che $\Lambda = 0$ su Ω si ha che

$$\sigma = (\ell_{1,3} - \ell_{2,3}\langle \ell_1, \ell_2 \rangle)I_1 + (\ell_{2,3} - \ell_{1,3}\langle \ell_1, \ell_2 \rangle)I_2$$

non si annulla su Ω . Allora condizione necessaria per l'esistenza e l'unicità di una soluzione di classe C^2 è che

$$\frac{\partial}{\partial y}u_x(x, y) = \frac{\partial}{\partial x}u_y(x, y)$$

Questo risultato presenta due problematiche:

1. L'insieme dei punti $D_0 = \{(x, y) \in \Omega \mid \Lambda(x, y) = 0\}$ spesso ha misura nulla, e quindi nella pratica il corollario non è applicabile.

2. Esiste un altro insieme D_1 in cui le derivate parziali sono uniche, e questo insieme di punti non ha alcun significato fisico. Inoltre si può notare che D_1 non è facile da trovare in quanto dipende dalla posizione delle luci. Per maggiori dettagli su questo caso si veda [10].

Ciò che penalizza il corollario è il fatto che si richieda che la soluzione sia di classe C^2 . Per cui occorre cercare di abbassare la regolarità della soluzione, mantenendo però l'unicità della soluzione.

Dato il sistema (3.1), si può notare che le due equazioni hanno in comune il denominatore che racchiude la nonlinearietà dell'equazione differenziale. Per cui dalla prima si ricava che:

$$\sqrt{1 + |\nabla u(x, y)|^2} = \rho \frac{\langle -\nabla u(x, y), \tilde{\ell}_1 \rangle + \ell_{1,3}}{I_1(x, y)}$$

e sostituendo alla seconda si ottiene

$$\begin{aligned} (I_2(x, y)\ell_{1,1} - I_1(x, y)\ell_{2,1}) \frac{\partial u}{\partial x} + (I_2(x, y)\ell_{1,2} - I_1(x, y)\ell_{1,1}) \frac{\partial u}{\partial y} = \\ = I_2(x, y)\ell_{1,3} - I_1(x, y)\ell_{2,3} \end{aligned}$$

Posto

$$\begin{aligned} \mathbf{b}(x, y) &= (I_2(x, y)\ell_{1,1} - I_1(x, y)\ell_{2,1}, I_2(x, y)\ell_{1,2} - I_1(x, y)\ell_{1,1})^T, \\ f(x, y) &= I_2(x, y)\ell_{1,3} - I_1(x, y)\ell_{2,3} \end{aligned}$$

il sistema (3.1) si riformula come.

$$\begin{cases} \langle \mathbf{b}(x, y), \nabla u(x, y) \rangle = f(x, y) & \forall (x, y) \in \Omega \\ u(x, y) = g(x, y) & \forall (x, y) \in \partial\Omega \end{cases} \quad (3.2)$$

E si ha il seguente:

Teorema 3.1.4. *$u(x, y)$ è una soluzione della (3.1) se e solo se è soluzione della (3.2).*

Corollario 3.1.5. *Posto di avere due fotografie e di conoscere la posizione delle luci, il sistema (3.1) ammette un'unica soluzione, e questa è Lipschitziana.*

Per le dimostrazioni si veda [6].

Una volta trovata la soluzione, è possibile ricavare l'albedo in questo modo:

$$\rho(x, y) = \frac{I_j}{\langle \ell_j(x, y), \mathbf{n} \rangle}$$

per $j = 1$ o $j = 2$.

I risultati visti fino ad ora, sono stati ottenuti sfruttando la formulazione continua del problema della Photometric Stereo. Nella prossima sezione verrà discretizzato il problema per ottenere lo stesso risultato nel caso discreto. Come si vedrà, la versione discreta permetterà di mettere in evidenza alcune considerazioni non esplicitamente visibili nel caso continuo.

3.2 Caso discreto con due immagini

Data la forma matriciale della legge di Lambert

$$RN^T L = M \quad (3.3)$$

Dove:

1. $R = \text{diag}(\rho_1, \dots, \rho_p)$ è l'albedo
2. $N = (n_{i,j}) = [\mathbf{n}_1, \dots, \mathbf{n}_p] \in \mathbb{R}^{3 \times p}$ è il campo normale
3. $L = [\ell_1, \dots, \ell_q] \in \mathbb{R}^{3 \times q}$ è la matrice contenente la posizione delle luci. In questo caso $q=2$.
4. $M = [\mathbf{m}_1, \dots, \mathbf{m}_q] \in \mathbb{R}^{p \times q}$ rappresenta l'insieme delle immagini siccome

$$\mathbf{n}_k = \frac{1}{\sqrt{1 + (u_x^2)_k + (u_y^2)_k}} \begin{pmatrix} -(u_x)_k \\ -(u_y)_k \\ 1 \end{pmatrix} \quad \forall k = 1, \dots, p$$

La (3.3) si riscrive come:

$$RD\tilde{N}^T L = M$$

dove

1. $D = \text{diag}(d_1, \dots, d_p)$ con $d_k = \frac{1}{\sqrt{1 + (u_x^2)_k + (u_y^2)_k}}$,

2. $\tilde{N} = \begin{bmatrix} -(u_x)_1 & \dots & -(u_x)_p \\ -(u_y)_1 & \dots & -(u_y)_p \\ 1 & \dots & 1 \end{bmatrix}$.

Se si pone $\tilde{R} = RD = \text{diag}[r_1, \dots, r_p]$ con $r_k = \rho_k d_k$, la forma matriciale del problema diventa:

$$\tilde{R}\tilde{N}^T L = M \quad (3.4)$$

Introduciamo le seguenti notazioni:

$$\tilde{\mathbf{n}}_k = \begin{bmatrix} -(u_x)_k \\ -(u_y)_k \end{bmatrix} \text{ con } k = 1, \dots, p \text{ e } N_1 = [\tilde{\mathbf{n}}_1, \dots, \tilde{\mathbf{n}}_p] \in \mathbb{R}^{2 \times p}$$

$$\tilde{\ell}_i = \begin{bmatrix} l_{1,i} \\ l_{2,i} \end{bmatrix} \text{ con } i = 1, 2 \text{ e } L_1 = [\tilde{\ell}_1, \tilde{\ell}_2] \in \mathbb{R}^{2 \times 2}$$

$$\mathbf{v} = (l_{3,1}, l_{3,2})^T \text{ da cui } L = \begin{pmatrix} L_1 \\ \mathbf{v}^T \end{pmatrix}$$

$$r_k \tilde{\mathbf{n}}_k^T \tilde{\ell}_j + r_k l_{j3} = m_{kj},$$

con $k = 1, \dots, p$ e $j = 1, 2$.

Posto $j = 1$, si ricava che

$$r_k = \frac{m_{k,1}}{\tilde{\mathbf{n}}_k^T \tilde{\ell}_1 + l_{3,1}}$$

Sostituendo nell'espressione per $j = 2$, si ha

$$m_{k1}(\tilde{\mathbf{n}}_k^T \tilde{\ell}_2 + l_{32}) = m_{k2}(\tilde{\mathbf{n}}_k^T \tilde{\ell}_1 + l_{31}),$$

da cui, mettendo in evidenza il vettore normale, si ottiene

$$\begin{aligned} 0 &= \tilde{\mathbf{n}}_k^T (m_{k2} \tilde{\ell}_1 - m_{k1} \tilde{\ell}_2) + m_{k2} l_{31} - m_{k1} l_{32} \\ &= \tilde{\mathbf{n}}_k^T L_1 \begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} + \begin{pmatrix} l_{31} & l_{32} \end{pmatrix} \begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} \end{aligned}$$

e quindi:

$$(\tilde{\mathbf{n}}_k^T L_1 + \mathbf{v}^T) \begin{bmatrix} m_{k2} \\ -m_{k1} \end{bmatrix} = 0. \quad (3.5)$$

Dalla (3.5) si può notare che:

1. Se $\begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ allora $\tilde{\mathbf{n}}_k^T L_1 + \mathbf{v}^T$ non è definito univocamente, e dato che l'unica incognita è il vettore normale $\tilde{\mathbf{n}}_k^T$ allora quest'ultimo non è unico. Ciò implica che il problema è mal posto se entrambe le immagini presentano zone d'ombra nella stessa posizione

2. Se $\tilde{\mathbf{n}}_k^T L_1 + \mathbf{v}^T = \mathbf{0}$ allora, avendo posto $L_1 = [\tilde{\ell}_1, \tilde{\ell}_2]$ si ricava

$$\begin{cases} \tilde{\mathbf{n}}_k^T \tilde{\ell}_1 + l_{31} = 0 \\ \tilde{\mathbf{n}}_k^T \tilde{\ell}_2 + l_{32} = 0 \end{cases} \Rightarrow \begin{cases} [\tilde{\mathbf{n}}_k^T & 1] \ell_1 = 0 \\ [\tilde{\mathbf{n}}_k^T & 1] \ell_2 = 0 \end{cases} \Rightarrow \begin{cases} \mathbf{n}_k \perp \ell_1 \\ \mathbf{n}_k \perp \ell_2 \end{cases}$$

Confrontando con la relazione $\rho_k n_k^T l_j = m_{kj}$ si ha che $n_k^T l_j = 0 \forall j = 1, 2$, da cui $m_{kj} = 0$ e quindi si ricade nelle considerazioni fatte al punto 1.

Ciò si verifica, ad esempio, nel caso in cui la superficie è piana e la luce è radente.

Esclusi questi casi, la (3.5) si può scrivere come

$$\tilde{\mathbf{n}}_k^T L_1 \begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} = m_{k1} l_{23} - m_{k2} l_{13} \text{ con } k = 1, \dots, p \quad (3.6)$$

E la (3.6) è la forma discreta della (3.2).

Ora, siccome $L_1 = [\tilde{\ell}_1 \ \tilde{\ell}_2]$, allora la (3.6) si riscrive come

$$a_k (u_x)_k + b_k (u_y)_k = c_k \quad (3.7)$$

dove

1. $a_k = l_{2,1} m_{k1} - l_{1,1} m_{k2}$
2. $b_k = l_{2,2} m_{k1} - l_{1,2} m_{k2}$
3. $c_k = l_{2,3} m_{k1} - l_{1,3} m_{k2}$

con $k = 1, \dots, p$.

Siccome le incognite sono le $(u_x)_k$ e $(u_y)_k$ e p indica il numero di pixel nelle singole immagini, si può notare che si è trovato un sistema con $2p$ incognite e p equazioni e quindi il sistema discretizzato è sottodeterminato. Ciò implica che il problema potrebbe non avere un'unica soluzione se si vedono le incognite come variabili indipendenti e non come derivate della funzione $z = u(x, y)$.

Un altro modo per vedere questo risultato è il seguente. Fissato un valore k , le variabili corrispondenti $(u_x)_k$ e $(u_y)_k$ sono presenti solo nella k -esima equazione. Ciò implica che risolvere il sistema lineare così come si presenta equivale a risolvere p equazioni lineari a due variabili indipendenti tra loro, e quindi per ogni coppia $((u_x)_k, (u_y)_k)$ esistono ∞^1 soluzioni.

Per cui, come è stato detto prima, un modo per risolvere il sistema è quello di ricorrere al metodo alle differenze finite discretizzando le derivate con le differenze centrali.

3.3 Implementazione dell'Algoritmo con 2 luci

Come è stato detto prima, si vuole risolvere la (3.7) applicando le differenze finite centrali. Prima di far ciò notiamo che la relazione (3.7) vale per ogni pixel dell'immagine, per cui tornando all'ordine non lessicografico, questa si può riscrivere come:

$$a_{i,j}(u_x)_{i,j} + b_{i,j}(u_y)_{i,j} = c_{i,j}$$

Avendo posto il dominio $\Omega = [a, b] \times [c, d]$ e discretizzando quest'ultimo come $i = 0, \dots, n+1$ e $j = 0, \dots, m+1$.

Ora, applicando il metodo alle differenze centrali, l'equazione si riscrive come

$$a_{i,j}u_{i+1,j} - a_{i,j}u_{i-1,j} + b_{i,j}u_{i,j+1} - b_{i,j}u_{i,j-1} = 2hc_{i,j} \quad (3.8)$$

dove $u_{i,j} = u(x_i, y_j)$.

Per cui, posto: $\mathbf{u}_i = [u_{i,1} \ u_{i,2} \ \dots \ u_{i,m}]^T$ da cui $\mathbf{u} = [\mathbf{u}_1^T \ \mathbf{u}_2^T \ \dots \ \mathbf{u}_m^T]^T$ e $\mathbf{a}_i = [a_{i,1} \ a_{i,2} \ \dots \ a_{i,m}]^T$ così come $\mathbf{b}_i = [b_{i,1} \ b_{i,2} \ \dots \ b_{i,m}]^T$ e $\mathbf{f}_i = 2h [c_{i,1} \ c_{i,2} \ \dots \ c_{i,m}]^T$

La (3.8) si riscrive come

$$-D_i \mathbf{u}_{i-1} + B_i \mathbf{u}_i + D_i \mathbf{u}_{i+1} = \tilde{\mathbf{f}}_i$$

Dove $D_i = \text{diag}(\mathbf{a}_i)$ con $i = 1, \dots, n$ e

$$B_i = \begin{bmatrix} b_{i,1} & & & \\ & \ddots & & \\ & & b_{i,m} & \end{bmatrix} \begin{bmatrix} 0 & -1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & 1 & 0 \end{bmatrix} := \text{diag}(\mathbf{b}_i) T_i$$

e $\tilde{\mathbf{f}}_i = \mathbf{f}_i + [b_{i,1}u_{i,0} \ 0 \ \dots \ 0 \ -b_{i,m}u_{i,m+1}]^T$
e sviluppando lungo i ho che:

$$B_1 \mathbf{u}_1 + D_1 \mathbf{u}_2 = \tilde{\mathbf{f}}_1 + D_1 \mathbf{u}_0$$

$$-D_i \mathbf{u}_{i-1} + B_i \mathbf{u}_i + D_i \mathbf{u}_{i+1} = \tilde{\mathbf{f}}_i \quad \text{con } i = 2, \dots, n-1$$

$$-D_n \mathbf{u}_{n-1} + B_n \mathbf{u}_n = \tilde{\mathbf{f}}_n - D_n \mathbf{u}_{n+1}$$

Abbiamo trovato così un sistema lineare $S\mathbf{u} = \mathbf{g}$ dove la matrice associata al sistema è uguale a:

$$S = \begin{bmatrix} B_1 & D_1 & & & & \\ -D_2 & B_2 & D_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & D_{n-1} & \\ & & & -D_n & & B_n \end{bmatrix}$$

È noto dal teorema di Kramer che il sistema lineare ammette un'unica soluzione se e solo se il determinante della matrice associata è non singolare. Per cui, siccome è fondamentale avere un problema ben posto, occorre analizzare i casi in cui il determinante della matrice S si annulla.

Un caso è sicuramente quando $a_{ij} = b_{ij} = 0$, ossia quando

$$[\tilde{\ell}_1 \quad \tilde{\ell}_2] \begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} = 0$$

. Si possono perciò verificare le seguenti situazioni:

1. $\begin{pmatrix} m_{k2} \\ -m_{k1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ e come è stato visto nel paragrafo precedente ciò si verifica se si hanno stesse zone d'ombra nelle due immagini.
2. $[\tilde{\ell}_1 \quad \tilde{\ell}_2] = (0 \quad 0)$ il che implica che le luci sono radenti al piano $z = 0$.
3. $\tilde{\ell}_1 m_{k2} = \tilde{\ell}_2 m_{k1}$ e siccome si sta escludendo il caso 1, uno tra i due valori m_{k1} e m_{k2} è diverso da zero. Supposto che $m_{k1} \neq 0$ l'equazione si riscrive come:

$$\tilde{\ell}_2 = \tilde{\ell}_1 \frac{m_{k2}}{m_{k1}}$$

Ossia, le proiezioni delle luci sul piano $z = 0$ sono proporzionali.

Si può notare come le problematiche legate alla singolarità della matrice dipendono esclusivamente dalla posizione delle luci, per cui sono facilmente controllabili nell'atto dell'acquisizione delle immagini.

Per cui, l'algoritmo alla Hamilton-Jacobi si descrive nel modo seguente:

Algoritmo alla Hamilton-Jacobi con due luci

Input :

M insieme delle immagini

L matrice contenente la posizione delle luci

h passo di discretizzazione

Output :

Z superficie ricostruita

Passo 0 : Se h non è noto, ricostruirlo. Ciò è possibile ottenerlo inserendo un righello all'interno dell'ambiente della fotografia e così si può trovare h in base a come viene riprodotto lo strumento.

Passo 1 : Noti h , L e M , si costruisce la matrice S e il termine noto g tramite un'opportuna function **sistzluci**.

Passo 2 : Sfruttando un'opportuna function si risolve il sistema $S\mathbf{u} = \mathbf{g}$ e così si trova la soluzione Z .

3.4 Caso con più luci

Se si hanno a disposizione $n > 2$ immagini M_1, \dots, M_n associate a n luci $\tilde{\ell}_1, \dots, \tilde{\ell}_n$, allora data ℓ_1 è possibile costruire $n - 1$ coppie

$$(M_1 \ M_2), \dots, (M_1 \ M_n)$$

e per ciascuna di queste coppie si considerano i sistemi lineari costruiti come nel paragrafo precedente

$$\begin{cases} S_{1,2}\mathbf{x} = \mathbf{b}_{1,2} \\ \vdots \\ S_{1,n}\mathbf{x} = \mathbf{b}_{1,n} \end{cases} \quad (3.9)$$

Siccome \mathbf{x} è in comune a tutti i sistemi, la (3.9) si riscrive come:

$$\begin{bmatrix} S_{1,2} \\ \vdots \\ S_{1,n} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b}_{1,2} \\ \vdots \\ \mathbf{b}_{1,n} \end{bmatrix}$$

il quale è un sistema sovradeterminato, e in quanto tale si può risolvere nel senso dei minimi quadrati.

Da notare che la scelta della luce $\tilde{\ell}_1$ è arbitraria purchè soddisfa i criteri posti all'algoritmo alla Hamilton-Jacobi con 2 luci. Ossia, che le

luci non devono generare zone d'ombra comuni alle immagini, e che non siano perpendicolari ad un vettore normale \mathbf{n}_k fissato.

Per cui, l'algoritmo si può scrivere come segue:

Algoritmo alla Hamilton–Jacobi con n luci

Input :

M insieme delle immagini

L matrice contenente la posizione delle luci

h passo di discretizzazione

Output :

Z superficie ricostruita

Passo 0 : Se h non è noto, ricostruirlo. Ciò è possibile ottenerlo inserendo un righello all'interno dell'ambiente della fotografia e così si può trovare h in base a come viene riprodotto lo strumento.

Passo 1 : Si sceglie $\tilde{\ell}_j$ in modo tale che soddisfi le condizioni di idoneità per le luci. Si pone $\tilde{\ell}_j = \tilde{\ell}_1$ e i restanti pari a $\tilde{\ell}_2, \dots, \tilde{\ell}_n$.

Passo 2 : Sfruttando la function **sistzluci** dell'Algoritmo alla Hamilton–Jacobi con duelluci, per ogni coppia (M_1, M_j) si costruisce la matrice associata $S_{1,j}$ e il termine noto $\mathbf{b}_{1,j}$. Una volta ricavate le $S_{1,j}$ e le $\mathbf{b}_{1,j}$

è possibile costruire la matrice $S = \begin{bmatrix} S_{1,2} \\ \vdots \\ S_{1,n} \end{bmatrix}$ così come il termine noto

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_{1,2} \\ \vdots \\ \mathbf{b}_{1,n} \end{bmatrix}$$

Passo 3 : Sfruttando un'opportuna function si risolve il sistema $S\mathbf{u} = \mathbf{g}$ e così si trova la soluzione Z .

Capitolo 4

Esperimenti numerici

Di seguito verranno mostrati alcuni esperimenti numerici legati agli algoritmi proposti nei capitoli precedenti.

4.1 Esempio di ricostruzione di una superficie mediante l'Algoritmo alla Poisson

In questo paragrafo verranno proposti degli esempi che mostreranno come l'algoritmo alla Poisson sia efficace nella ricostruzione di una superficie. Ciò è necessario in quanto si vuole misurare la bontà dell'algoritmo e, una volta ricavate le foto da dati reali, la superficie ricostruita potrebbe essere indistinguibile da quella reale a occhio nudo. Ciò è dovuto al fatto che al di sotto di una precisione relativa di 10^{-2} l'occhio non è praticamente in grado di notare le differenze. Ragion per cui occorre testare questo procedimento su dati sintetici.

Per far questo, prima di tutto si è fatto un test per mostrare l'efficacia delle function **Poissolve** e **Poissolvedue** non tenendo conto della ricostruzione della normale. Per cui si è costruito in ambiente Matlab un file **Testpois** strutturato come segue:

Testpois

Passo 0 : Si sceglie la funzione test f per la quale sono note le derivate parziali e il laplaciano. Avendo a disposizione f , si può conoscere quanto vale la funzione in tutto il dominio, e quindi a maggior ragione si sa quanto vale anche lungo il bordo, conoscenza che non è assolutamente banale da ottenere nei dati reali e che è in-

dispensabile per imporre le condizioni al contorno dell'equazione differenziale.

Passo 1 : Si fissa un passo di discretizzazione h e in base a questo viene discretizzato il dominio di definizione di f . Questo punto serve per simulare i pixel di una foto.

Passo 2 : Tramite la discretizzazione si ricostruisce il termine noto come spiegato in 2.2.2 sfruttando la conoscenza delle derivate parziali della funzione f .

Passo 3 : Dato il termine noto, il passo di discretizzazione della foto e le condizioni al contorno, si risolve il sistema lineare (2.4) tramite **Poissolve** e **Poissolvedue**.

Passo 4 : Una volta trovata la soluzione numerica, vengono mostrati i grafici di quest'ultima, della soluzione esatta e dell'errore commesso nel dominio discretizzato.

Di seguito è mostrato un esempio di come le due function ricostruiscono una generica superficie.

Da questo esempio si può notare come ad occhio nudo non vi sia alcuna differenza tra le tre superfici, tuttavia, se si considerano i grafici 4.1 e 4.1 è possibile accorgersi di come abbiano lavorato i due algoritmi.

Come si può vedere, siccome **Poissolve** usa un metodo diretto per risolvere il sistema lineare, l'errore commesso nella risoluzione segue globalmente l'andamento della superficie.

Di contro, **Poissolvedue** usando un metodo iterativo, si arresta non appena viene raggiunto una sensibilità fissata o se si è raggiunto un numero di iterazioni fissato.

In questo caso, si è fissata una sensibilità di 10^{-2} in quanto, come è stato già detto, l'occhio oltre questa soglia non riesce a notare le differenze e l'algoritmo si è arrestato alla tredicesima iterazione, e come si può notare l'errore non si è distribuito perfettamente sul dominio.

Da notare che il grafico è ribaltato rispetto all'asse delle z . Ciò significa semplicemente che **Poissolve** dà un'approssimazione per eccesso della superficie.

Una volta testato l'effetto delle due function, è lecito domandarsi quanto la ricostruzione della normale peggiori la soluzione. Per cui, si è costruito in ambiente Matlab un file **Testpoisdue** identico a **Testpois** tranne per il fatto che in quest'ultima la normale è esatta.

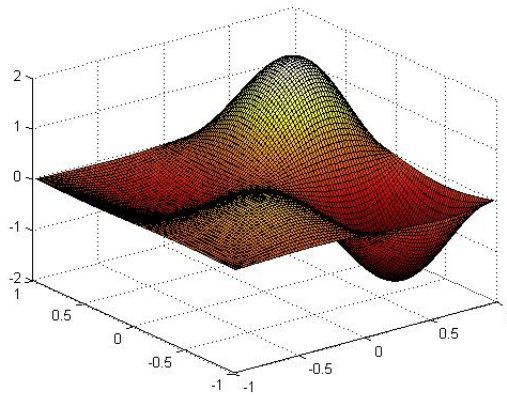


Figura 4.1: Superficie originale

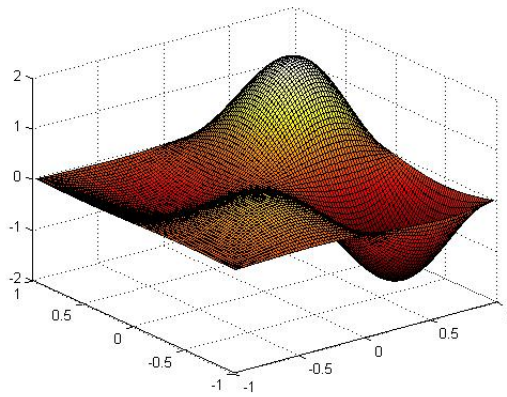


Figura 4.2: Superficie ricostruita tramite **Poissolve**

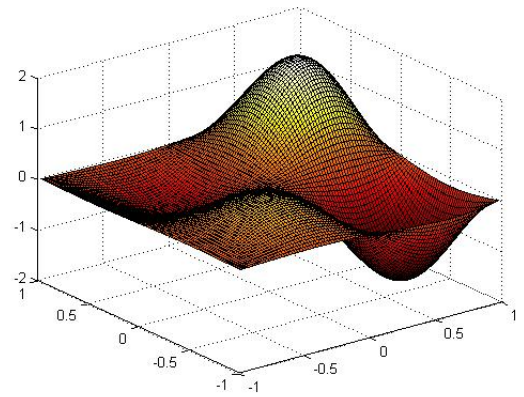


Figura 4.3: Superficie ricostruita tramite **Poissolvedue**

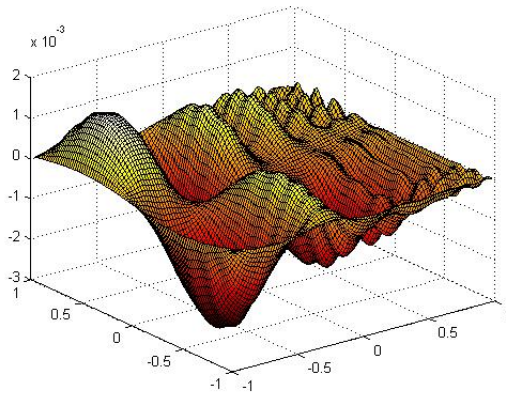


Figura 4.4: Errore commesso da **Poissolve**

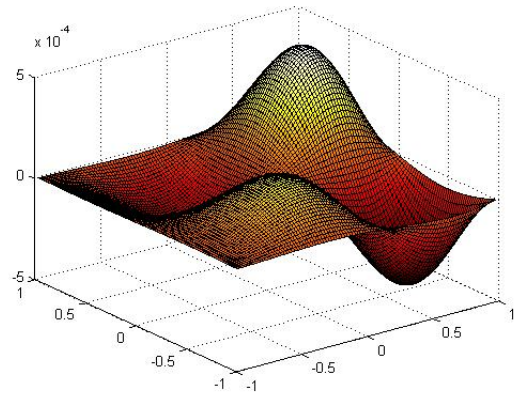


Figura 4.5: Errore commesso da **Poissolvedue**

Anche in questo caso le due superfici sono identiche all'originale. E confrontando gli errori commessi dalle due function, si può notare come la ricostruzione della normale non abbia introdotto errori significativi.

4.2 Esempio di ricostruzione di una superficie mediante l'Algoritmo alla Hamilton–Jacobi

Di seguito verrà mostrato un esempio di come opera l'Algoritmo alla Hamilton–Jacobi nel caso di due immagini.

Nel paragrafo precedente si erano costruite due function per la risoluzione del sistema lineare in quanto erano note alcune proprietà di cui gode la matrice di Poisson.

Nell'algoritmo alla Hamilton–Jacobi, invece, la matrice associata al sistema dipende fortemente dalla posizione delle luci e ciò non ha permesso di individuare proprietà utili per la ricerca di algoritmi di risoluzione ad hoc. Per questo motivo la risoluzione del sistema lineare è affidata al comando backslash di Matlab.

Per cui, si è costruito in ambiente Matlab un file **Test1luci** analogo al file **Testpois**.

In un primo caso si è scelta come prima luce una posta a 0 radianti rispetto alla macchina fotografica, e la seconda a π radianti. Le figure 4.11 e 4.12 fanno riferimento alle immagini fornite in input, mentre le figure

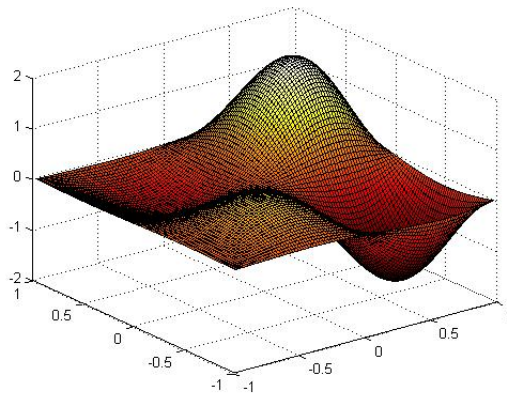


Figura 4.6: Superficie originale

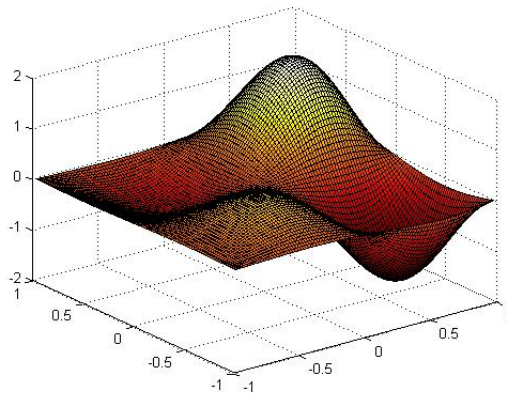


Figura 4.7: Superficie ricostruita tramite **Poissolve**

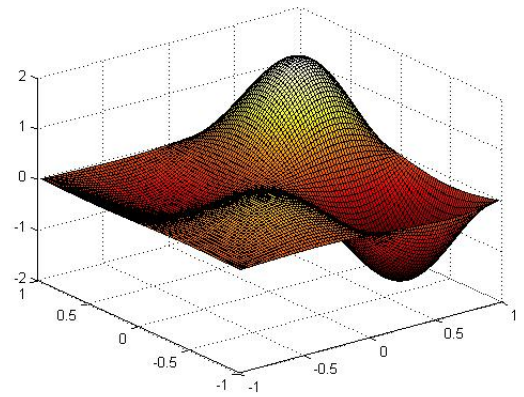


Figura 4.8: Superficie ricostruita tramite **Poissolvedue**

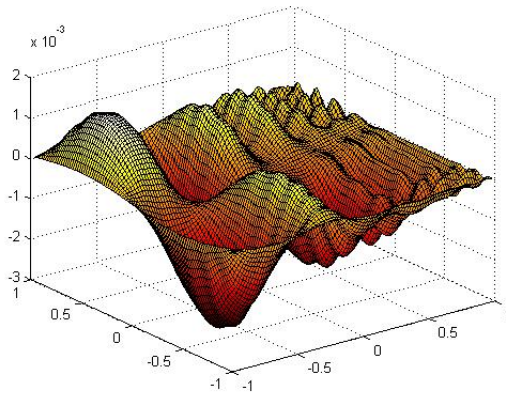


Figura 4.9: Errore commesso da **Poissolve**

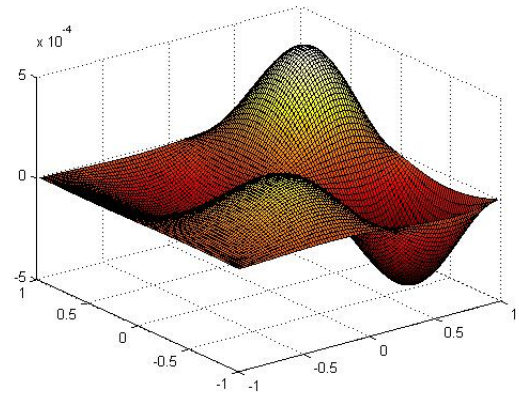


Figura 4.10: Errore commesso da **Poissolvedue**

4.13, 4.14 e 4.15 fanno riferimento rispettivamente alla superficie esatta, quella ricostruita e l'errore commesso dall'algoritmo.

Come secondo caso si è presa la prima luce a 0 radianti dalla macchina fotografica, e la seconda a $\frac{\pi}{2}$ radianti. Le figure 4.16 e 4.17 fanno riferimento alle immagini fornite in input, mentre le figure 4.18, 4.19 e 4.20 fanno riferimento rispettivamente alla superficie esatta, quella ricostruita e l'errore commesso dall'algoritmo. Da come si può notare dalle immagini la superficie ricostruita non sembra corrispondere alla superficie esatta.

Ciò è dovuto al fatto che le luci hanno generato delle zone d'ombra in comune e questo fatto ha generato degli errori numerici significativi. Questo esempio mostra come l'algoritmo alla Hamilton-Jacobi con 2 luci dal punto di vista teorico è molto interessante in quanto afferma che, note le posizioni delle luci, con sole due immagini è possibile ricostruire la superficie. Tuttavia, dal punto di vista pratico è preferibile avere più di due immagini per due motivi

1. In presenza di errori sperimentali i minimi quadrati smorzano l'influenza degli errori perchè minimizzano la varianza rispetto ai dati
2. C'è minore probabilità che le immagini presentino stesse zone d'ombra, il che implica che la matrice ha meno probabilità di risultare a rango non pieno

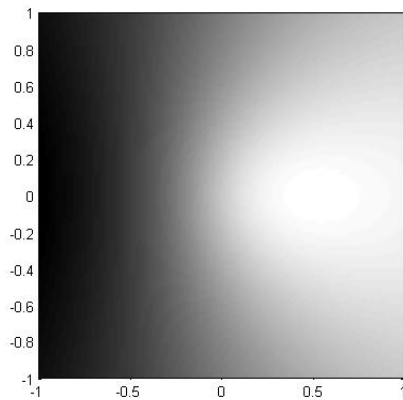


Figura 4.11: Prima immagine

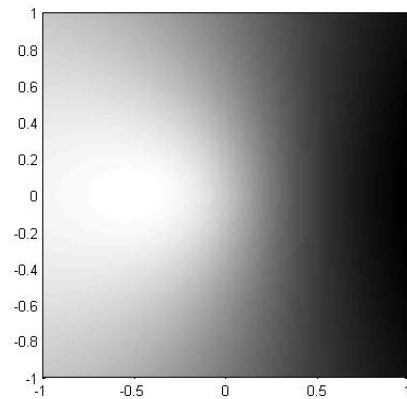


Figura 4.12: Seconda immagine

4.3 Ricostruzione di una superficie con dati perturbati

Di seguito viene proposto un test nel quale la matrice L contenente la posizione delle luci e la matrice M rappresentante l'insieme delle immagini vengono perturbati.

La perturbazione dei dati in L simula delle misurazioni imprecise sulla provenienza delle fonti luminose, invece la perturbazione in M simula il noise sulla ricostruzione della matrice. In particolare si sono considerati degli errori che seguono una distribuzione Gaussiana di media 0 e varianza 10^{-2} .

Come si vede dalle immagini 4.21 e 4.22, l'algoritmo alla Hamilton-Jacobi risulta molto più sensibile agli errori rispetto all'algoritmo alla Poisson. Ciò conferma le conclusioni poste nel paragrafo precedente secondo cui un problema sovradeterminato realizza una pulizia statistica sui dati affetti da errore.

4.4 Non derivabilità della superficie

Nella risoluzione dell'equazione di Poisson è implicita l'ipotesi che la superficie sia derivabile.

Ciò è dovuto al fatto che le derivate seconde vengono approssimate tramite le differenze centrali, e se questa non è derivabile la stessa

equazione differenziale perde di significato.

Questo ragionamento è facilmente visualizzabile dalle figure 4.24 e 4.25.

4.5 Esperimenti sui tempi di calcolo dell'Algoritmo di Poisson

Una volta create le function **Poissolve** e **Poissolvedue** necessarie per la risoluzione dell'equazione di Poisson, si è effettuato un test legato ai tempi di calcolo.

Il risultato è dato dal grafico 4.5 e confronta l'algoritmo **Poissolve** con diversi livelli di sensibilità con l'algoritmo **Poissolvedue** il quale, come è stato già detto, sfrutta il comando backslash di Matlab.

Si può notare come per dimensioni non troppo grandi l'algoritmo che sfrutta un metodo diretto risulta essere più competitivo rispetto a quelli che utilizzano metodi iterativi, ma una volta superato una dimensione di soglia la situazione si ribalta.

Occorre ricordare che in letteratura esistono algoritmi, detti FPS (*Fast Poisson Solver*), ad hoc per sistemi associati all'equazione di Poisson i quali potrebbero avere tempi di calcolo minori rispetto a **Poissolve**.

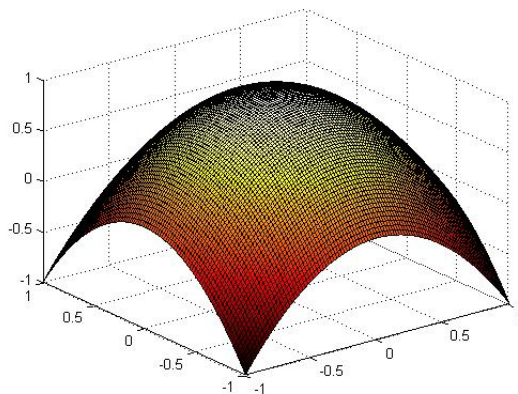


Figura 4.13: Superficie originale

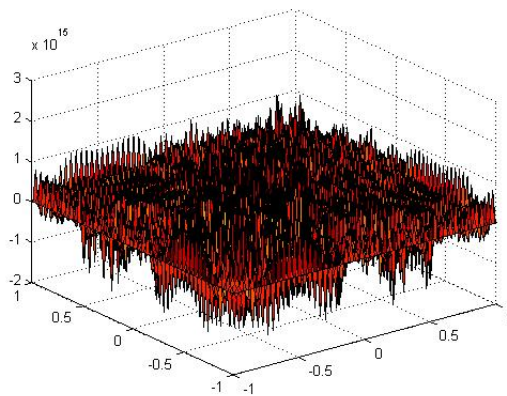


Figura 4.14: Superficie ricostruita

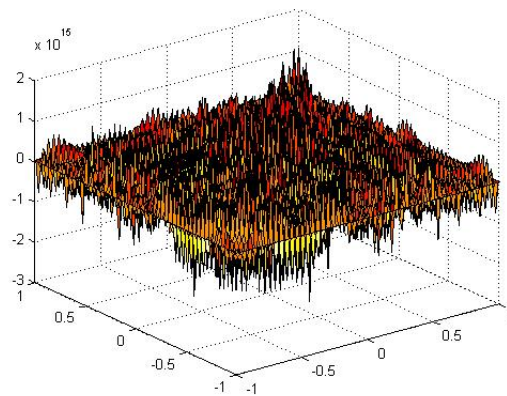


Figura 4.15: Errore commesso

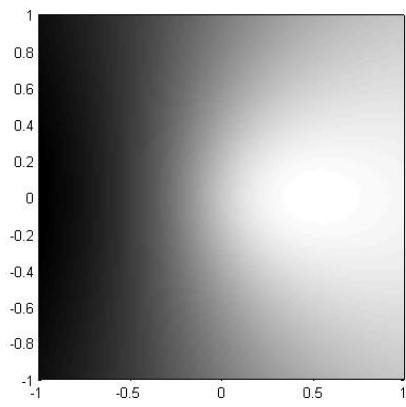


Figura 4.16: Prima immagine

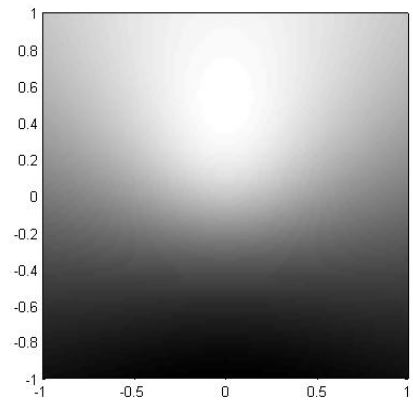


Figura 4.17: Seconda immagine

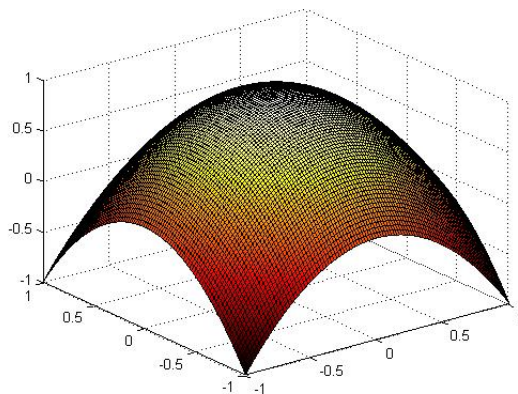


Figura 4.18: Superficie originale

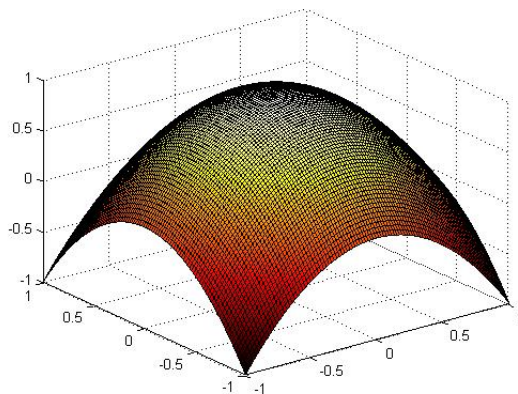


Figura 4.19: Superficie ricostruita

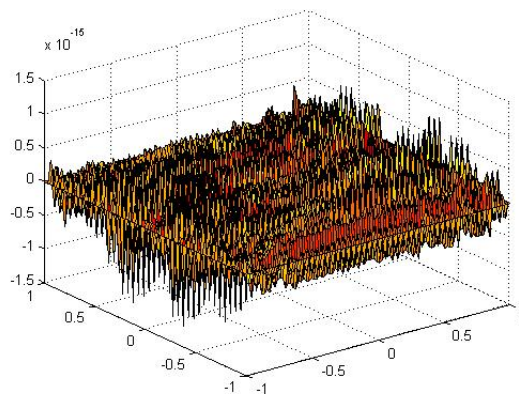


Figura 4.20: Errore commesso

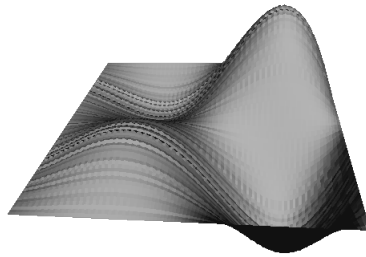


Figura 4.21: Superficie ricostruita mediante Hamilton-Jacobi

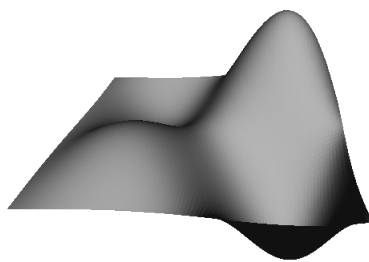


Figura 4.22: Superficie ricostruita mediante Poisson

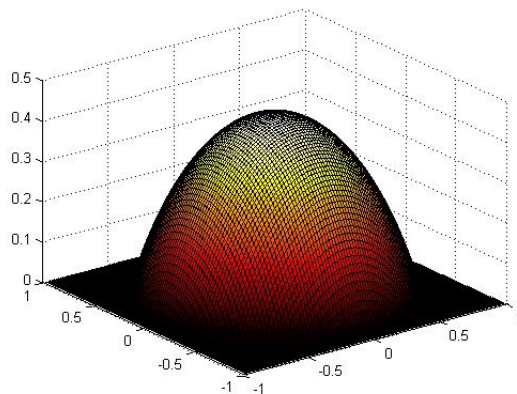


Figura 4.23: Superficie originale

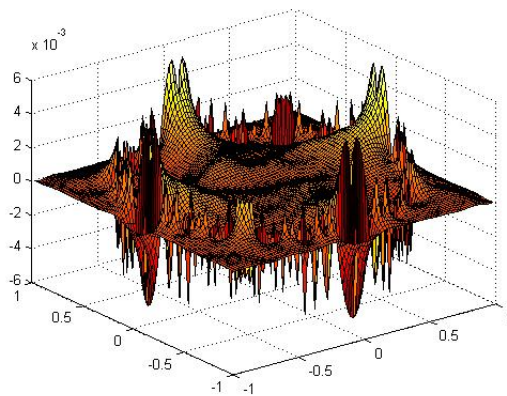


Figura 4.24: Errore commesso da **Poissolve**

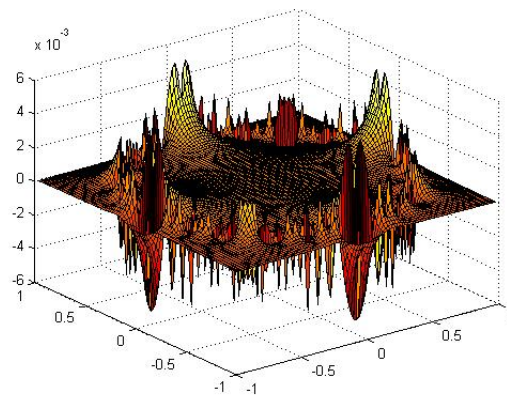


Figura 4.25: Errore commesso da **Poissolvedue**

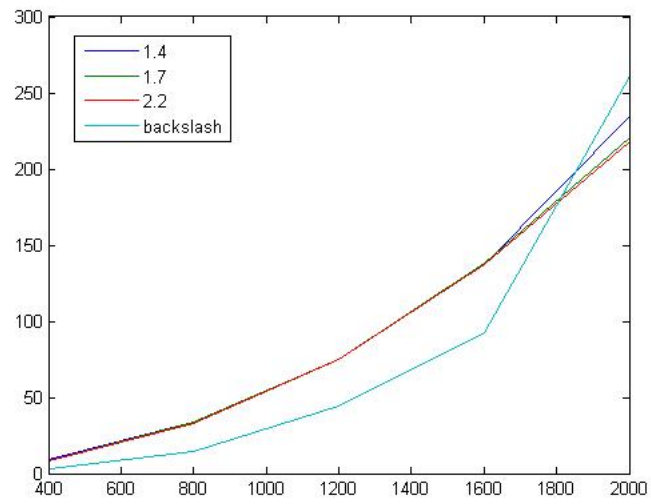


Figura 4.26: Confronto dei tempi di calcolo

Bibliografia

- [1] R. Zhang, P. S. Tsai, J. E. Cryer e M. Shah: *Shape-from-Shading* IEEE Trans. Pattern Anal. Mach. Intell., 1999
- [2] J. Van Diggelen: *A photometric investigation of the slopes and heights of the ranges of hills in the Maria of the Moon* Bull. Astron. Inst. Netherlands 1951
- [3] B. W. Hapke: *A theoretical photometric function for the lunar surface* J. Geophys, 1963
- [4] T. Rindfleisch: *Photometric method for lunar topography* Photometric Engineering, 1966
- [5] B. K. P. Horn e M. J. Brooks: *Shape from Shading* The MIT Press, 1989
- [6] R. Mecca e M. Falcone: *Uniqueness and Approximation of a Photometric Shape-from-Shading Model* SIAM J. Imaging Sciences, 2013
- [7] S. Seatzu e P. Contu: *Equazioni alle derivate parziali. Una introduzione ai metodi di risoluzione analitica e numerica* Pitagora, 2012
- [8] A. Bjork: *Numerical methods for least squares problems* SIAM, 1996
- [9] R. Kozera: *Existence and uniqueness in photometric stereo* Appl. Math. Comput., 1991
- [10] R. Mecca e J.-D Drouot: *Unambiguous photometric stereo using two images* Springer-Verlag, 2011
- [11] R. Dessì: *Algoritmi ottimizzati per la Photometric Stereo applicati all'archeologia* Tesi di laurea in Ingegneria Elettronica, Università degli studi di Cagliari, 2012-2013
- [12] V. Comincioli: *Analisi numerica: metodi, modelli, applicazioni* Apogeo, 2005