POLITECNICO DI TORINO

Corso di Laurea
in Matematica per l'Ingegneria

Tesi di Laurea

Filtraggio di immagini

Relatori
prof. Nome Cognome
prof. Nome Cognome
firma dei relatori

. . . .
. . . .

Candidato
Raffaello Ippolito

firma del candidato

. . . .

Anno Accademico 2021-2022

♡ *Alla mia regina*

Sommario

Lo studio delle Equazioni alle derivate parziali ed il loro impiego in ambito di filtraggio di immagini.

Ringraziamenti

I candidati ringraziano vivamente il Granduca di Toscana per i mezzi messi loro a disposizione, ed il signor Von Braun, assistente del prof. Albert Einstein, per le informazioni riservate che egli ha gentilmente fornito loro, e per le utili discussioni che hanno permesso ai candidati di evitare di riscoprire l'acqua calda.

Indice

Elenco delle tabelle

Elenco delle figure

If you cannot understand my
argument, and declare
it’s Greek to me
you are quoting Shakespeare.

[B. LEVIN, Quoting Shakespeare]

Parte I

Introduzione

Capitolo 1

Nozioni introduttive

1.1 Introduzione

Le immagini hanno un ruolo fondamentale nelle nostre vite, viviamo di immagini, ne guardiamo tutti i giorni in tutti i contesti, la percezione visiva è sempre il nostro primo riferimento senza la quale ci si sentiamo persi. Viviamo in una società consapevole di ciò e che sfrutta questo aspetto quanto più possibile nei campi più disparati, fotografie, radiografie, ecografie, poster pubblicitari, progetti, etc. E' quindi un problema sempre di grande interesse cercare di sfruttarle al meglio, a tal fine esistono metodi cosìdetti di "filtraggio", tramite i quali intendiamo migliorare la qualità delle nostre immagini, mettere in risalto determinate caratteristiche o nasconderne altre.

Viviamo in un era digitale, le immagini passano generalmente per un calcolatore prima di essere stampate, o in ogni caso possono essere sempre scannerizzate (con strumenti più o meno precisi) così da averne una copia digitale. E' in questa fase che l'immagine subisce il processo di filtraggio, nel calcolatore, quando pè in formato digitale, Per capire cos'è un filtro occorre dunque chiedersi cosa sia un'immagine digitale.

1.2 Immagini digitali

Per capire come codificare un'immagine per memorizzarla in fgormato digitale ci chiediamo prima che cos'è un'immagine.

> Forma esteriore degli oggetti corporei, in quanto viene percepita attraverso il senso della vista, o si riflette – come realmente è, o variamente alterata – in uno specchio, nell'acqua e sim., o rimane impressa in una lastra o pellicola o carta fotografica.
>
> Vocabolario Treccani

Un'immagine viene rappresentata, impressa quindi su superfici, cioè oggetti bidimensionali, di dimensioni finite e le vediamo perchè i nostri occhi percepiscono il susseguirsi di colori diversi. Come codificare tali entità? Come tutti gli oggetti reali, sebbene abbiano dimensioni finite, il susseguirsi delle immagini avviene in una maniera che possiamo considerare come continua. Questo è il primo problema che ci si pone quando si pensa a come codificare delle immagini. La soluzione più largamente utilizzata è anche quella più semplice ed intuitiva, ossia di discretizzare tale distribuzione di colori. Dividiamo l'immagine con una griglia ed ad ogni casella, che d'ora in poi chiameremo "pixel", assegnamo un colore. E' ovvio che così facendo si perdono dei dettagli, la quantità di dettagli che riusciamo a conservare piò variare enormemente, un minimo si perde sempre ma è un prezzo che siamo disposti a pagare.

Facciamo un esempio

Figura 1.1. Confronto immagine originale e immagine codificate utilizzando una griglia 4x4.

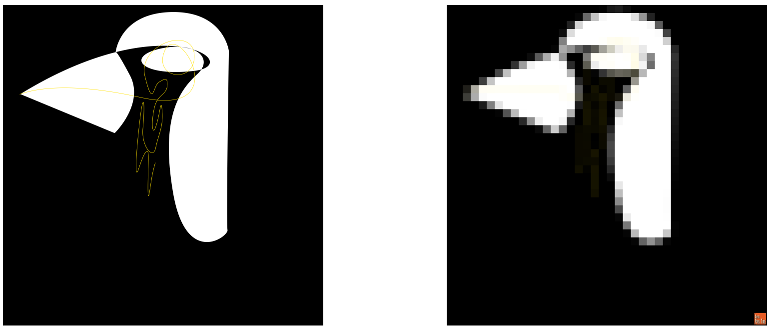

Figura 1.2. Confronto immagine originale e immagine codificate utilizzando una griglia 80x80.

E' semplice vedere come ad una griglia più fitta corrisponda una miglior qualità dell'immagine, questo è il concetto di "risoluzione di un'immagine". Una miglior risoluzione però costa, come anticipato, in termini di memoria. Una griglia 4x4 corrisponde a 16 pixel, ad una griglia 80x80 corrispondono invece 6400 pixel! Questo vuol dire che la seconda immagine pesa 400 volte di più della prima.

Volendo definire in maniera più precisa cosa e' un'immagine digitale, diremmo che quest'ultima è una funzione da $\mathbb{R}^2$ in $\mathbb{R}^3$ cioè, date in input due coordinate, essa restituisce un colore (che è formato da 3 canali RGB). Se però l'immagine è in bianco e nero la questione si semplifica: l'immagine diventa una funzione da $\mathbb{R}^2$ in $\mathbb{R}$, dal momento che per codificare un colore appartenente alla scala di grigio basta un solo canale, il livello di luminosità. Difatti, la riduzione da tre ad un solo canale rappresenta un grosso vantaggio, permettendo di diminuire di due terzi lo spazio di memoria occupato.

1.3 Cosa sono i filtri

Una volta capito che un'immagine è una funzione possiamo definire un filtro come una seconda funzione che convoluta alla prima da il risultato richiesto.

Una equazione alle derivate parziali (PDE) esprime una evoluzione, sia u la nostra immagine e u_0 lo stato in cui si trova inzialmente, allora per convoluzione possiamo dire che

$$u(x) = \frac{1}{w(x)} \int \int d(x-\xi)\tilde{d}(u_0(x) - u_0(\xi))u_0(\xi)d\xi. \tag{1.1}$$

con $w(x) = \int \int d(x-\xi)\tilde{d}(u_0(x) - u_0(\xi))d\xi$

Definiti questi concetti siamo pronti ad iniziare la trattazione vera e propria

Parte II

Prima Parte

Capitolo 2

Equazione del calore e metodo Perna-Malik

2.1 L'equazione del calore

Prendiamo in esame l'equazione del calore

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) - \Delta u(t,x) = 0 \; x \in \mathbb{R}^2, t \geq 0 \; . \\ u(0,x) = u_0(x) \; . \end{cases} \tag{2.1}$$

Prendiamo in esame l'eq del calore.

L'equazione del calore, come intuibile dal nome che porta, è stata formulata per determinare l'evoluzione di un sistema isolato che presenta al suo interno una data distribuzione di calore. E' banale pensare che con il passare del tempo il calore si distribuisca, tendendo per un tempo infinito ad una distribuzione uniforme.

Euristicamente, non è difficile pensare che possiamo codificare (pensando ad un'immagine in bianco e nero) i pixel più luminosi come punti "più caldi" e quelli più scuri come punti "più freddi" ed applicare così l'equazione del calore.
Otteniamo quindi un'immagine sempre più "liscia", otteniamo di fatto una sfocatura, e per un numero di iterazioni idealmente infinito ci ritroveremmo con una distribuzione uniforme di colore, ossia una tinta unita.

Vediamo con uno script matlab come opera nella pratica.
Per fare ciò operiamo una approssimazione alle differenze finite per il calcolo delle derivate.
Questa si rifà molto alla definizione in se di derivata. Decidiamo quindi di approssimare

$$\frac{du}{dx} \approx \frac{u_i + 1 - u_i}{delta(x)}$$

Se la derivata seconda è la derivata della derivata, allora approssimiamo

$$
\begin{aligned}
\frac{d^2u}{dx^2} \approx & \frac{(\frac{u_i+1-u_i}{\Delta(x)})_i+1-(\frac{u_i+1-u_i}{\Delta(x)})_i}{\Delta(x)} \\
= & \frac{\frac{(u_i+1-u_i)_i+1}{\Delta(x)}-\frac{(u_i+1-u_i)_i}{\Delta(x)}}{\Delta(x)} \\
= & \frac{\frac{(u_i+2-u_i+1)}{\Delta(x)}-\frac{(u_i+1-u_i)}{\Delta(x)}}{\Delta(x)} \\
= & \frac{\frac{(u_i+2-u_i+1)-(u_i+1-u_i)}{\Delta(x)}}{\Delta(x)} \\
= & \frac{(u_i+2-u_i+1)-(u_i+1-u_i)}{\Delta(x)^2} \\
& \frac{u_i+2-u_i+1-u_i+1+u_i}{\Delta(x)^2}
\end{aligned}
$$

Per una migliore approssimazione è bene utilizzare un valore di delta(x) quanto più basso possibile. Il massimo che possiamo fare è vedere la differenza tra un pixel e quello adiacente ossia $\Delta(x) = 1$, ma allora

$$\frac{d^2u}{dx^2} \approx \frac{u_i + 2 - u_i + 1 - u_i + 1 + u_i}{\Delta(x)^2} = u_i + 2 - 2u_i + 1 + u_i$$

Capiamo che scorrendo tutti gli indici questo è equivalente a $u_i + 1 - 2u_i + u_i - 1$.
In sintesi utilizzeremo per il calcolo del laplaciamo l'approssimazione

$$\frac{d^2u}{dx^2} \approx u_i + 1 - 2u_i + u_i - 1$$

Vediamo lo script:

```


Im=imread('parrot.jpeg');     %Apro l'immmagine
Im=rgb2gray(Im);              %La trasformo in bianco e nero
Im = imnoise(Im,'gaussian');%Aggiungo del rumore

%---Definisco le costanti e le condizioni iniziali

[ny, nx, ~]=size(Im)          %Dimensioni dell'immagine
dt=0.25;                      %Passo temporale
u=double(Im);                 %Copia dell'immagine originale su cui lavorare
T=3                           %Tempo, ossia T/dt + 1 definisce il numero di
                               iterazioni da eseguire
k=0.5;

%---Mostro l'immagine originale
imshow(uint8(Im))
title("immagine originale");

%---Metodo eq del calore
u=double(Im);
```

```
for t = 0:dt:T
   u_xx = u(:,[2:nx nx],:) - 2*u + u(:,[1 1:nx-1],:);  % derivata seconda lungo x
   u_yy = u([2:ny ny],:,:) - 2*u + u([1 1:ny-1],:,:);  % derivata seconda lungo y
   u= u + k*dt*(u_xx+u_yy);
   temp=u;
end
```

Provando a cambiare il tempo, ossia il numero di iterazioni, osserviamo come un maggior lasso di tempo produca immagini più sfocate. Questo metodo però non è poi molto utile in generale, sì, il rumore viene eliminato o almeno ridotto ma si perdono importanti dettagli.

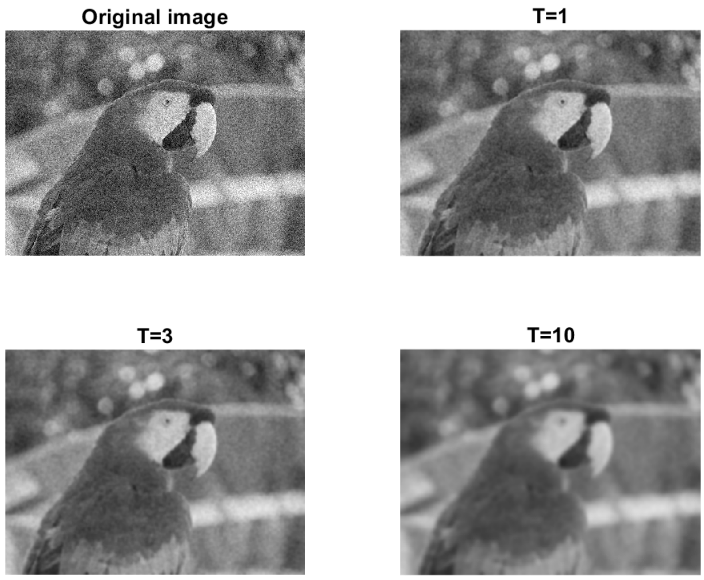

Figura 2.1. Effetti dell'eq del calore nel tempo.

Operando una derivata in una data direzione, per il significato in sè di derivata, questa assume valori più elevati quando la variazione è elevata, e assume valori nulli quando non c'è variazione in quella direzione. Per questo motivo, applicata ad un' immagine, ne rileviamo i bordi!

Presa una tinta unita la derivata sarà quindi nulla in ogni suo punto (è intuitivo, se un'immagine è una funzione che, date due coordinate restituisce un colore, allora una tinta unita è una funzione costante ed in quanto tale ha derivata nulla).

Operando una derivata seconda in una data direzione, per il significato in sè di derivata seconda, questa assume valori più elevati quando la concavità è più stretta, e assume valori

pressocchè nulli quando non ci sono concavità (possiamo pensare alle concavità come a dei picchi o dei ventri, su di una immagine vuol dire chiazze di colore diverso).
Presa una sfumatura di colore che varia in maniera lineare, la derivata seconda sarà nulla in ogni suo punto, la derivata prima sarà invece costante. Scriviamo un piccolo script su MATLAB che ci mostri questo aspetto

```
%---Operazioni preliminari
Im=imread("nome_immagine.png"); %Apro l'immmagine

[ny, nx, ~]=size(Im)          %Memorizzo le dimensioni dell'immagine
u=double(Im);                 %Copia dell'immagine originale su cui lavorare
h=80;                         %Definisco un parametro che usero' per
                               enfatizzare i bordi in fase di stampa


%---Calcolo tutte le derivate
u_x =  u(:,[1 1:nx-1],:) - u;                             %derivata
                                                           prima lungo x
u_xx = u(:,[2:nx nx],:) - 2*u + u(:,[1 1:nx-1],:);  %derivata
                                                           seconda lungo x
u_y =  u([1 1:ny-1],:,:) - u;                             %derivata
                                                           prima lungo y
u_yy = u([2:ny ny],:,:) - 2*u + u([1 1:ny-1],:,:);  %derivata
                                                           seconda lungo y
u_xy = u_x([1 1:ny-1],:,:) - u_x;                         %derivata
                                                           seconda mista

%---Stampo i risultati
figure()
subplot(2,3,2),text(0.3,0,nome,"FontSize",20); axis off
subplot(2,3,4), imshow(Im)
title("Immagine originale")
subplot(2,3,5), imshow(uint8(h*abs(u_x)))
title("h*u_x")
subplot(2,3,6), imshow(uint8(h*abs(u_y)))
title("h*u_y")

figure()
subplot(2,3,2),text(0.3,0,nome,"FontSize",20); axis off
subplot(2,3,4), imshow(Im)
title("Immagine originale")
subplot(2,3,5), imshow(uint8(h*abs(u_x + u_y)))
title("h*(u_x + u_y)")
subplot(2,3,6), imshow(uint8(h*abs(u_xx + u_yy)))
title("h*(u_xx + u_yy)")
```

Vediamo con diverse immagini molto semplici se otteniamo i risultati attesi.

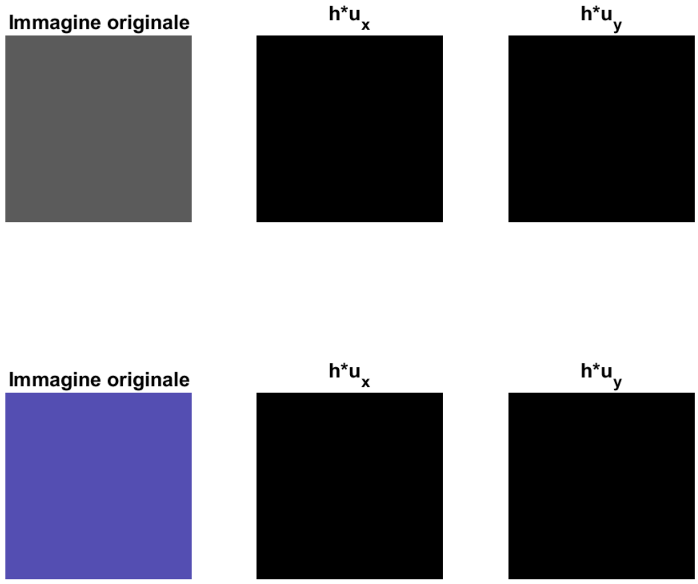

Figura 2.2. Derivate parxiali di una tinta unita.

Possiamo vedere come con un'immagine a tinta unita (che sia essa in bianco e nero o a colori) le derivate sono nulle, quindi lo saranno anche gradiente e laplaciano. Confermiamo inoltre che questi principi valgono sia a colori che in bianco e nero.

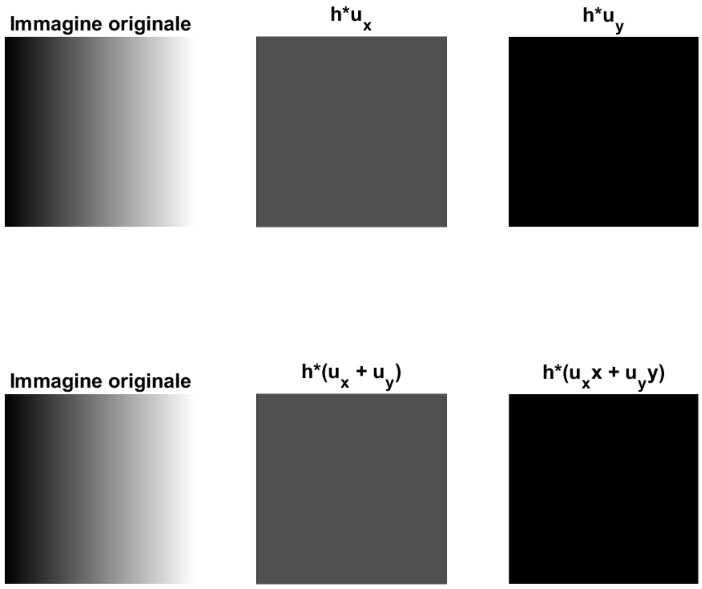

Figura 2.3. Derivate parziali, gradiente e laplaciano di una sfumatura orizzontale.

Guardando invece ad una immagine che presenta una sfumatura lineare lungo l'asse x, la derivata lungo x assume un valore costante mentre la derivata lungo y è nulla, proprio perchè lungo y non c'è variazione mentre lungo x c'è una variazione costante.
Ovviamente, date queste premesse, il gradiente sarà costante uguale u_x (siccome $u_y = 0$) e quindi il laplaciano sarà nullo. Il fatto che in entrambi questi esempi il laplaciano sia nullo è un buon segno, lo useremo per rilevare i bordi ed in queste immagini non ve ne sono, quindi è giusto che il laplaciano sia nullo.

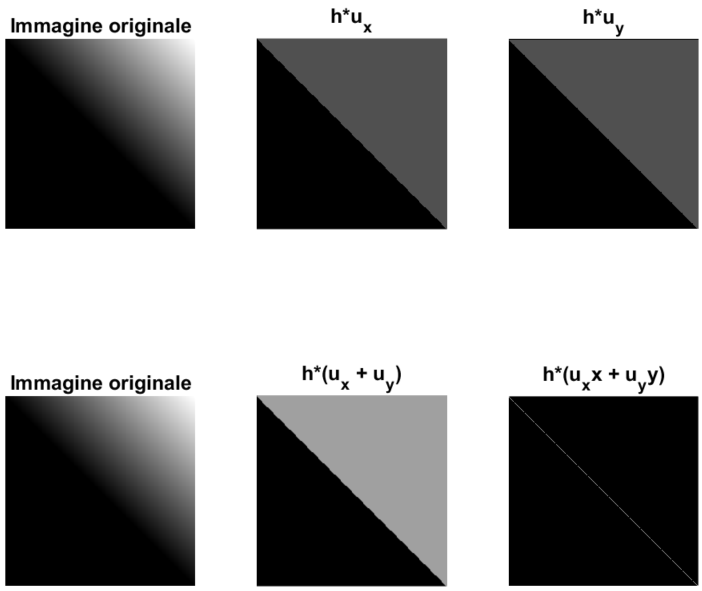

Figura 2.4. Derivate parziali, gradiente e laplaciano di una sfumatura diagonale.

Se prendiamo una sfumatura diagonale ma solo su metà immagine vediamo dei riultati interessanti: entrambe le derivate parziali sono nulle nelle regioni in cui non c'è sfumatura, esattamente come nel caso della tinta unita, ed entrambe sono costanti dove c'è sfumatura (che ricordiamo essere lineare).
Tutto ciò riconferma quanto visto dai punti precedenti, decidiamo quindi di volgere uno sguardo al gradiente ed al laplaciano e noteremo qualcosa di interessante, il gradiente ha un aspetto molto simile alle due derivate parziali, sommando i loro valori è semplicemente più luminoso.
Per quanto riguarda il gradiente la storia cambia. Le derivate seconde sono indicatrici della variazione delle derivate prime, cioè della variazione della variazione del valore della funzione, ma l'unica variazione che hanno le derivate prime è lungo la diagonale. Abbiamo così individuato il nostro primo bordo, cioè la diagonale che divide di fatto due regioni, una in cui il colore è costante ed una in cui sfuma.

Facciamo ancora un esempio, proviamo ad introdurre una semplice figura e vediamo cosa accade.

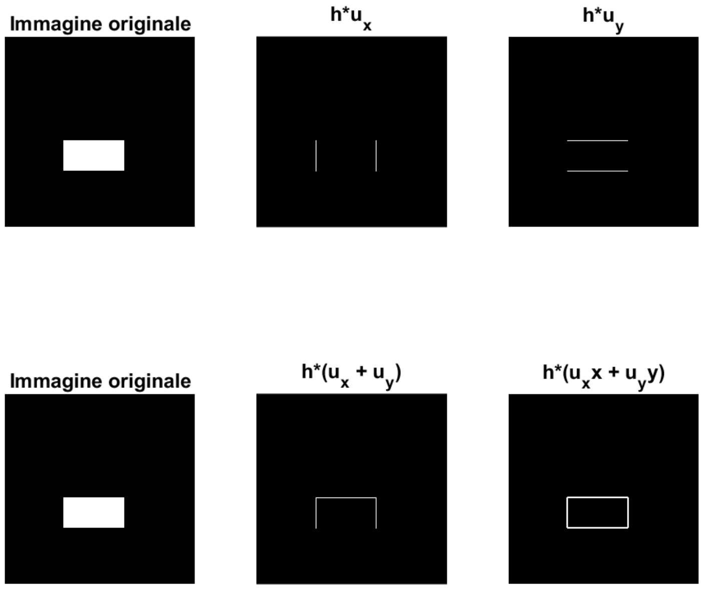

Figura 2.5. Derivate parziali, gradiente e laplaciano di una figura semplice.

Abbiamo importato un rettangolo nero su di uno sfondo bianco, come confermato anche dalla prima sfumatura, la derivata lungo x rileva i bordi verticali, quella lungo y i bordi orizzontali, dalla loro somma (quindi dal gradiente) otteniamo già il bordo del rettangolo. Il bordo così ottenuto è un bordo che idealmente rimarrà inalterato a prescindere dall'ordine della derivata, in particolare quindi anche per derivate seconde, quindi il laplaciano continua a soddisfare la nostra richiesta di determinare i bordi.
Come detto: "Il bordo così ottenuto è un bordo che idealmente rimarrà inalterato a prescindere dall'ordine della derivata" cerchiamo di capire perchè. Presa una striscia di pixel, cioè uno strato della nostra immagine (immaginiamola quindi come una funzione da R in R), otterremo quindi una funzione porta!
La funzione porta non è derivabile in senso classico, ripensando alla definizione di derivata avremmo un valore di +infinito prima e -infinito poi. La sua derivata sarà quindi una coppia di delta di Dirac