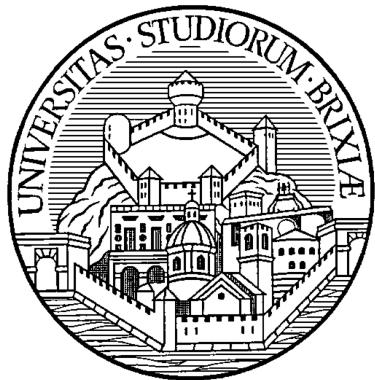


UNIVERSITÀ DEGLI STUDI DI BRESCIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE  
Corso di Laurea Magistrale in Ingegneria Informatica



Tesi di Laurea

**Tecniche di miglioramento di immagini basate  
sulla distribuzione locale del colore**

**Relatore:**

Prof. Nicola Adami

**Laureando:**

**Correlatori:**

Prof. Keiichiro Shirai

Maurizio Zucchelli

Prof. Sergio Benini

mat. 91665

Anno Accademico 2013-2014

# Indice

<b>Indice</b>	<b>ii</b>
<b>Elenco delle figure</b>	<b>iv</b>
<b>1 Introduzione</b>	<b>2</b>
1.1 Approssimazioni locali di immagini a superfici planari . . . . .	3
1.2 Trasformazioni di colore tra coppie di immagini . . . . .	5
<b>2 Approssimazioni locali di immagini a superfici planari</b>	<b>9</b>
2.1 Modello Lineare Locale nel dominio del colore . . . . .	10
2.2 Modello Lineare Locale nel dominio dello spazio . . . . .	10
2.2.1 Soluzione rispetto a $d'$ . . . . .	11
2.2.2 Soluzione rispetto a $z$ . . . . .	11
2.2.3 Soluzione rispetto a $\mathbf{n}'$ . . . . .	12
2.2.4 Combinazione con Equazioni Differenziali convenzionali . . . . .	13
2.3 Relazione con i Gradient Domain Operators . . . . .	13
<b>3 Trasformazioni di colore tra coppie di immagini</b>	<b>16</b>
3.1 Filtro guidato per regioni dinamiche . . . . .	16
3.1.1 Soluzione rispetto a $\mathbf{b}_i$ . . . . .	17
3.1.2 Soluzione rispetto a $\mathbf{A}_i$ . . . . .	17
3.1.3 Trasformazione globale del colore . . . . .	18
3.2 Analisi e riduzione delle decolorazioni . . . . .	19
3.2.1 Correzione dell'ordine e del segno degli autovettori . . . . .	19
3.2.2 Correzione degli autovalori di $\mathbf{\Xi}_i$ e $\mathbf{\Sigma}_i$ . . . . .	20
3.3 Classificazione di regioni statiche e dinamiche, e blending . . . . .	20

3.3.1	Addestramento della Support Vector Machine . . . . .	21
3.3.2	Classificazione e metodi di blending . . . . .	22
<b>4</b>	<b>Applicazioni del Modello Lineare Locale</b>	<b>24</b>
4.1	Inter-conversione tra Vettori Normali e Gradienti . . . . .	24
4.2	Applicazione alla Rimozione di Riflessi . . . . .	25
4.3	Applicazione alla Rimozione di Ombre . . . . .	26
4.3.1	Rimozione di Ombre basata su immagini invarianti all'illuminazione . . .	26
4.3.2	Rimozione di Ombre basata su piramidi Laplaciane . . . . .	30
<b>5</b>	<b>Risultati sperimentali</b>	<b>40</b>
5.1	Modello Locale Lineare . . . . .	40
5.1.1	Inter-conversione di Luminanza e Gradienti . . . . .	40
5.1.2	Rimozione di Riflessi . . . . .	41
5.1.3	Rimozione di Ombre . . . . .	44
5.2	Trasformazioni di colore tra coppie di immagini . . . . .	44
<b>6</b>	<b>Conclusioni</b>	<b>49</b>
<b>7</b>	<b>Stesura della Tesi</b>	<b>50</b>
<b>A</b>	<b>Matrice di Centratura Locale</b>	<b>52</b>
<b>B</b>	<b>Calcolo efficiente di <math>z</math></b>	<b>53</b>
	<b>Bibliografia</b>	<b>55</b>

# Elenco delle figure

1.1	Esempio di image matting . . . . .	3
1.2	Illustrazione del processo di funzionamento del filtraggio guidato . . . . .	6
1.3	Esempio di applicazione del filtraggio guidato . . . . .	7
2.1	La definizione del LLM adottata nei domini del colore e dello spazio . . . . .	9
2.2	Relazione tra un vettore normale ed il gradiente di profondità sul piano $x$ - $z$ . . . . .	14
3.1	Trasformazioni di colore e contrasto tramite GF, per regioni statiche e dinamiche . . . . .	21
3.2	Esempio di set di immagini utilizzate nella fase di addestramento della SVM . . . . .	22
4.1	Determinazione della superficie dell'ombra da rimuovere . . . . .	31
4.2	Istogramma delle distanze . . . . .	32
4.3	Determinazione della maschera dell'ombra da rimuovere . . . . .	33
4.4	Ristorazione dell'ombra . . . . .	36
4.5	Ristorazione tramite piramide Laplaciana . . . . .	37
5.1	Risultati della inter-conversione di luminanza e gradienti . . . . .	41
5.2	Un'applicazione del LLM alla rimozione di riflessi . . . . .	42
5.3	Viste in dettaglio di Figura 5.2 . . . . .	42
5.4	Coppie di immagini utilizzate per la rimozione di ombre . . . . .	43
5.5	Sfocature della rimozione di ombre tramite LLM . . . . .	44
5.6	Risultati della rimozione di ombre . . . . .	45
5.7	Dettagli dei risultati di Figura 5.6 . . . . .	46
5.8	Risultato della fase di test della SVM . . . . .	46
5.9	Risultati della composizione di immagini flash/no-flash . . . . .	47

## Sommario

Nel presente documento verranno presentati due metodi di miglioramento di immagini digitali: il *modello lineare locale applicato al dominio dello spazio* e il *filtraggio guidato basato su corrispondenze tra covarianze*.

Il modello lineare locale [1] è un potente metodo per l'elaborazione di immagini volta all'ottimizzazione, applicato nel dominio del colore in vari ambiti come immagini intrinseche [2], tone mapping [3], filtraggio guidato [4], ed altri. La prima tecnica qui analizzata è quindi l'applicazione del modello lineare locale al dominio dello spazio, costruendo così uno strumento robusto per il calcolo di gradienti e la risoluzione di equazioni di Poisson. Questo metodo è applicabile in una varietà di ambiti, come lo stereo fotometrico [5], la rimozione di riflessi [6], la rimozione di ombre [7], per i quali sono mostrati le applicazioni in questo documento, ed altri.

Il filtraggio guidato [4] è una tecnica di filtraggio esplicita altamente efficiente ed in grado di ottenere risultati spesso migliori dei filtri classici, oltre a permettere nuove applicazioni del filtraggio tramite la definizione implicita di kernel traslazione-varianti. Per il corretto funzionamento del filtraggio in molte di queste nuove applicazioni (in particolare, riduzione del rumore e tone-mapping) è importante che l'immagine guida sia perfettamente allineata con l'immagine in input, pena la comparsa di sfumature e decolorazioni.

Questo problema viene affrontato tramite la seconda tecnica qui proposta: una nuova variante del filtraggio guidato basata sulle corrispondenze tra covarianze in grado di rilassare il vincolo sull'allineamento ed ottenere buoni risultati anche in caso l'immagine in input e l'immagine guida non siano ben allineate. Il metodo proposto non è comunque privo di difetti, e può causare la comparsa di artefatti in regioni di colore costante. Per questo motivo viene descritta inoltre una tecnica di combinazione delle due tipologie di filtraggio guidato basata sulla classificazione delle regioni delle immagini migliori per ognuno dei due metodi. Come applicazione del metodo proposto viene mostrata la combinazione di immagini con/senza flash [8].

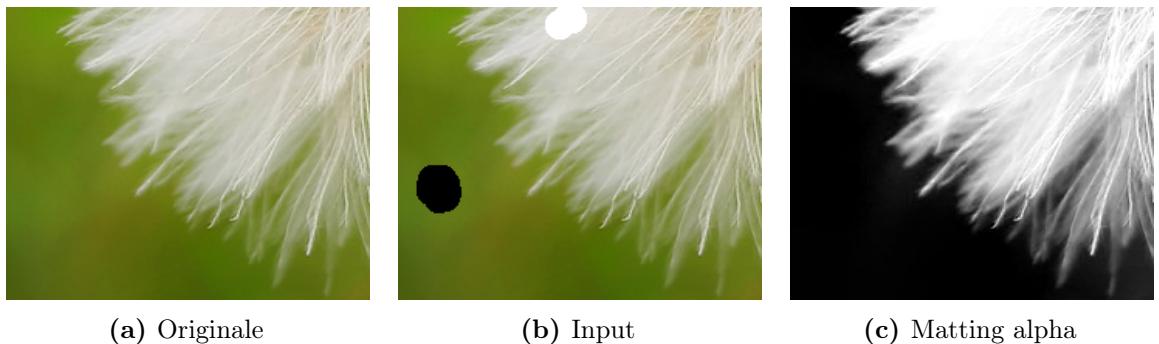
# Capitolo 1

## Introduzione

Il miglioramento di immagini digitali è un compito importante sia a livello artistico dove, per quanto la gradevolezza sia un parametro soggettivo, è spesso importante che le immagini abbiano rumore ridotto, e colori naturali e non eccessivamente saturi; sia a livello di computer vision, dove un’immagine pre-trattata adeguatamente può semplificare e favorire compiti come il riconoscimento di oggetti, il loro tracciamento, la segmentazione, ecc. In questo documento verranno presentate due tecniche applicabili in questo ambito.

La prima, introdotta in Sezione 1.1 e descritta in Capitolo 2, sfrutta le intensità dell’immagine per costruire un’approssimazione della stessa tramite delle superfici planari definite in ogni pixel. Le inclinazioni di queste superfici rappresentano una sorta di gradiente dell’immagine, col vantaggio aggiuntivo di una robustezza maggiore al rumore. Questo nuovo metodo può essere adottato in sostituzione a qualsiasi applicazione impieghi l’equazione di Poisson. Alcune sue applicazioni saranno mostrate in Capitolo 4.

La seconda tecnica analizzata, introdotta in Sezione 1.2 e descritta in Capitolo 3, è un nuovo approccio al filtraggio guidato incentrato sul rilassamento del forte requisito di allineamento tra i pixel dell’immagine in input e l’immagine guida. L’utilità di questo metodo risulta evidente quando sia l’immagine in input che l’immagine guida sono fotografie di scene naturali ottenute in condizioni—e quindi istanti temporali—differenti: è facile che le due immagini non risultino perfettamente allineate in qualche area (ad esempio, le foglie degli alberi). Gli approcci classici al filtraggio guidato tendono a causare, in questo caso, la comparsa di artefatti come sfocature o decolorazioni. La tecnica proposta riesce a risolvere questi problemi, al costo di artefatti nelle regioni statiche. Per questo motivo, in Sezione 3.2, viene proposto un metodo per combinare il filtraggio guidato classico con il filtraggio guidato proposto.



**Figura 1.1:** Esempio di image matting

Dall'immagine (a) viene estratta la maschera rappresentante il dente di leone (c). In (b) è mostrato l'input dell'utente (le macchie nere indicano la regione da considerare come sfondo, mentre quelle bianche indicano la regione da considerare come primo piano) necessario per ottenere il risultato (c) tramite l'algoritmo proposto da Levin et al. [1].

### 1.1 Approssimazioni locali di immagini a superfici planari

Un importante compito nell’elaborazione di immagini e video è il *matting digitale interattivo*, ossia il processo tramite cui un oggetto in primo piano viene estratto da un’immagine basandosi su input limitato da parte dell’utente Figura 1.1. Dal punto di vista della computer vision, questo compito è una grande sfida in quanto consiste in un problema estremamente mal posto, richiedendo la stima in ogni pixel dell’immagine del colore del primo piano, dello sfondo, e dell’opacità del primo piano (*matting alpha*), tutto da una singola misurazione del colore.

I metodi classici [9–14] affrontavano il problema restringendo l’analisi ad una piccola regione dell’immagine, effettuando la stima basandosi sui pixel vicini, dove noti, oppure tramite una stima iterativa e non lineare composta dall’alternanza della stima del colore di primo piano e sfondo con quella del *matting alpha*. Nel 2008, Levin et al. [1] hanno proposto una soluzione in forma chiusa all’*image matting* naturale. Il metodo deriva una funzione di costo da assunzioni locali sull’omogeneità dei colori di primo piano e sfondo e, da questa, ottiene una funzione di costo quadratica con solamente alpha come incognita.

Questo metodo è stato chiamato Local Linear Model (LLM) ed è ora un potente strumento per l'elaborazione di immagini volta all'ottimizzazione, non solo come termine di *data fidelity*, ma anche come vincolo versatile, ed ha visto applicazioni in più ambiti, come immagini intrinseche<sup>1</sup> [2], *tone mapping*<sup>2</sup> [3], filtraggio guidato [4], ed altri.

---

<sup>1</sup>La tecnica delle **immagini intrinseche** consiste nella scomposizione di un'immagine tra componenti intrinseche di riflessione, contenente informazioni sul colore degli oggetti nella scena, ed illuminazione, contenente informazioni sulle texture.

<sup>2</sup>Il **tone mapping** è una tecnica che consiste nel creare una relazione tra due set di colori con lo scopo di

Per quanto molti studi siano stati fatti riguardo al metodo nel dominio del colore, non sono note sue applicazioni a quello dello spazio. Sono stati comunque studiati metodi alternativi nell’ambito dell’approssimazione di intensità, ad esempio:

- Nello *stereo matching* [15], la continuità della profondità viene ottenuta usando superfici piatte a tratti. Tuttavia, dato che questi metodi sono espressi come operazioni non lineari ed hanno alta complessità computazionale, non sono adatti all’uso per la soluzione di problemi di ottimizzazione.
- Lo stereo fotometrico [5; 16] è un metodo basato sull’ottimizzazione che intende recuperare una mappa di profondità di un oggetto da informazioni su vettori normali alle superfici, quali la luminanza. Per fare questo, la superficie approssimata viene collegata implicitamente al problema del restauro in ogni pixel e l’equazione di Poisson associata viene risolta con il tradizionale filtraggio Laplaciano iterativo: neanche qui, quindi viene fatto uso esplicito del LLM.

Similarmente a come fatto negli studi sopracitati, verrà assunto che le distribuzioni delle intensità di luminanza possano essere approssimate localmente con piani tangenti ad una superficie. Basandosi su questa assunzione, verrà descritto un metodo di ottimizzazione applicabile a problemi di ricostruzione involgenti vettori normali a piani tangenziali e gradienti di immagini ortogonali alle normali. Questo metodo può essere applicato ad una varietà di problemi quali lo stereo fotometrico, la rimozione di riflessi [6], la rimozione di ombre [7; 17; 18], e altri metodi chiamati Gradient Domain Operator (GDO) [19–21].

A differenza dei convenzionali operatori differenziali che operano a corto raggio, il LLM può rappresentare variazioni di luminanza in una regione più ampia, fornendo risultati più robusti. Mentre filtri differenziali con risposte all’impulso più lunghe, come quelli usati in [5; 22] o anche i filtri di Sobel<sup>3</sup>  $3 \times 3$  e di Scharr [23], causano la comparsa di artefatti a griglia nelle immagini risultanti, il LLM riesce a rimuovere il rumore ed a fornire risultati più naturali. Lo svantaggio consiste, però, nella perdita di definizione degli spigoli rispetto alla differenza finita.

---

approssimare immagini HDR per la visualizzazione tramite un mezzo con range dinamico limitato.

<sup>3</sup>I **filtri di Sobel** consistono in coppie di filtri (uno orizzontale e la sua versione trasposta, verticale) solitamente utilizzati per il rilevamento di bordi. Comunemente sono di dimensione  $3 \times 3$  e corrispondono alle matrici

$$\mathbf{G}_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ e } \mathbf{G}_y = \mathbf{G}_x^\top.$$

Per questo motivo è possibile adottare una combinazione tra i due metodi in grado di compensare gli svantaggi di entrambi e fornire risultati migliori.

I contributi apportati da questo studio consistono in:

**Modello Lineare Locale per il Dominio dello Spazio:** applicabile ai GDO che includono l’equazione di Poisson:  $\arg \min_I \sum_i \|\nabla I_i - \mathbf{J}_i\|_2^2$  (con  $I_i \in \mathbb{R}$  e  $\mathbf{J}_i \in \mathbb{R}^2$ ), o la sua versione estesa con il LLM. Questo metodo fornisce risultati migliori dei filtri differenziali con risposta all’impulso lunga.

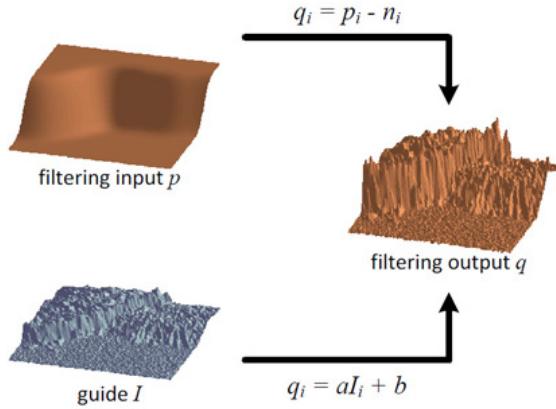
**Alta gestibilità:** questo LLM può essere usato in congiunzione con la convenzionale equazione di Poisson come vincolo e stabilizzatore, con lo scopo di migliorare la qualità dell’immagine. Verranno mostrati un metodo di calcolo efficiente in grado di gestire grandi kernel ed una tecnica [24] che fa uso del metodo del Gradiente Coniugato [25; 26] e dell’*immagine integrale* [27].

## 1.2 Trasformazioni di colore tra coppie di immagini

Il secondo tema affrontato riguarda il restauro di immagini deteriorate da rumore e saturazione dell’intensità del colore. In questo ambito, l’utilizzo di più immagini spesso porta a risultati migliori dell’utilizzo di una sola immagine. I metodi classici per effettuare queste operazioni richiedono, solitamente, che le posizioni dei soggetti presenti nelle due immagini siano *completamente allineati*, pena la comparsa di sfocature ed il fallimento nella correzione dell’immagini.

Tra i metodi che sfruttano molteplici immagini sono stati considerati i Guided Filter (GF) [4; 28–30], ossia filtri che usano un’immagine della stessa scena ottenuta in condizioni differenti come guida—e.g, per restaurare un’immagine di una scena notturna ottenuta senza l’utilizzo del flash, quindi con rumore dovuto all’alta sensibilità ISO necessaria, potrebbe venire utilizzata come immagine guida un’immagine della stessa scena, ottenuta però con l’ausilio del flash e, quindi, con rumore molto ridotto al costo della comparsa di alterazioni dei colori dovute alla luce del flash ed eventuali saturazioni. Nonostante questi metodi siano stati applicati in una varietà di situazioni, come correzione del contrasto [31] e *stereo matching* [32], le applicazioni obiettivo del metodo studiato sono la rimozione del rumore ed il *tone mapping* [28; 30; 31].

I filtri guidati nascono come alternativa ai classici filtri lineari invarianti alla traslazione (LTI), permettendo tempi di elaborazione genericamente inferiori (risultano infatti richiedere tempi lineari, indipendentemente dalla dimensione del kernel). Possono essere utilizzati come operatori



**Figura 1.2:** Illustrazione del processo di funzionamento del filtraggio guidato

L’immagine mostra, in particolare, l’applicazione del filtraggio per il trasferimento della struttura.  
Immagine presa da [4].

liscianti in grado di preservare i contorni meglio di filtri, e.g., quello bilaterale[Tomasi:1998aa] e, soprattutto, permettono nuove applicazioni del filtraggio grazie alla possibilità di trasferire strutture dell’immagine guida sull’immagine in uscita Figura 1.2.

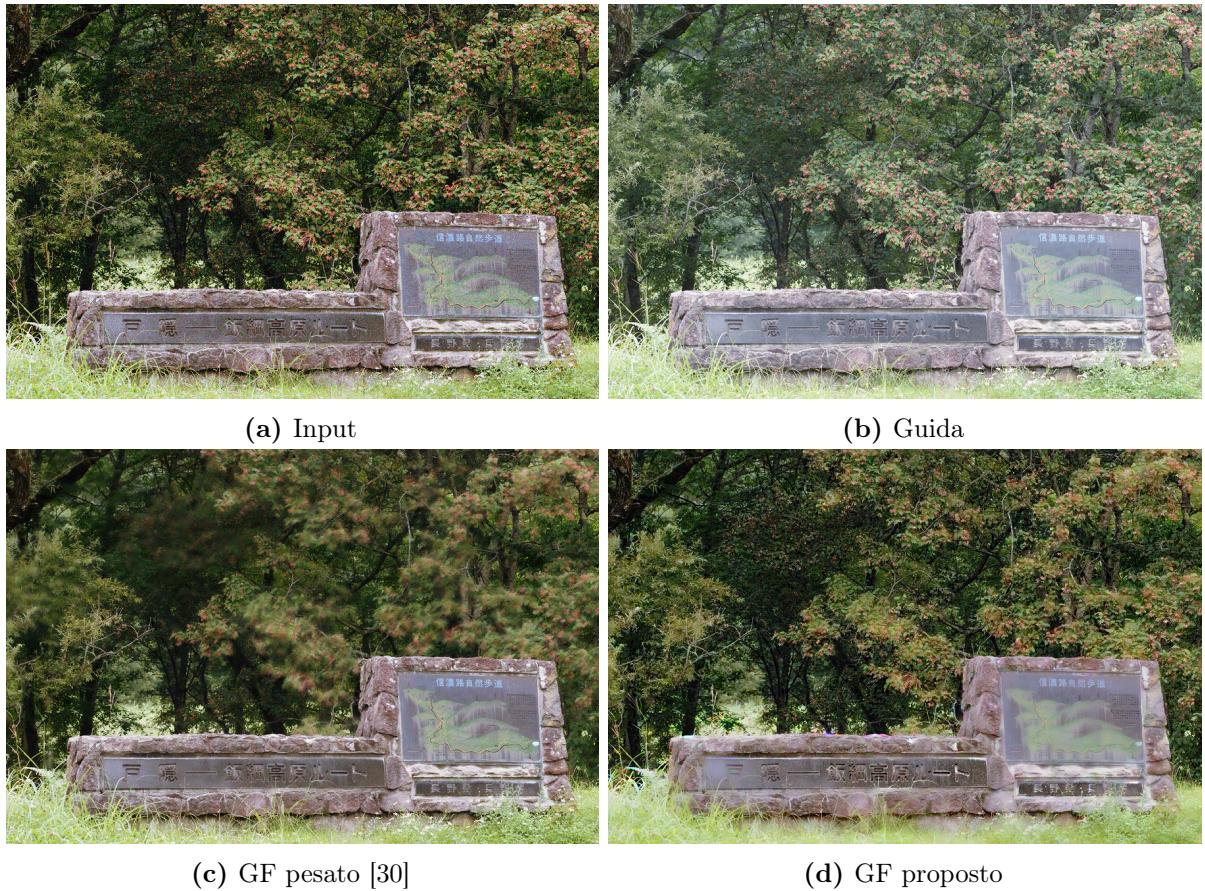
In genere, la distribuzione del colore su una piccola regione di un’immagine (chiamata anche *patch*) presenta una caratteristica denominata *color-line*<sup>4</sup> [33]. In patch corrispondenti di coppie di immagini rappresentanti la stessa scena ma ottenute in condizioni diverse, le due distribuzioni del colore tendono ad assomigliarsi e, pertanto, una di esse può essere ottenuta in modo approssimato come rotazione e traslazione dell’altra. Basandosi su questa osservazione, i GF esistenti [4; 28–30] affrontano il problema tramite la determinazione di una matrice di trasformazione che sia in grado di minimizzare l’errore della trasformazione commesso in ogni patch, i.e., esaminando un pixel  $i$  ed i suoi vicini  $j$  appartenenti alla patch  $\Omega_i$ , i colori locali dei pixel dell’immagine guida  $\mathbf{I}_j \in \mathbb{R}^3$  (e.g., RGB o YCbCr) sono trasformati linearmente tramite una matrice  $\mathbf{A}_i \in \mathbb{R}^{3 \times 3}$  ed un vettore  $\mathbf{b}_i \in \mathbb{R}^3$  comuni all’intera patch in modo da renderli il più simile possibile ai colori locali dei pixel dell’immagine fornita in input  $\mathbf{p}_j \in \mathbb{R}^3$ :

$$\arg \min_{A,b} \sum_i \sum_{j \in \Omega_i} w_{ij} (\|\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i - \mathbf{p}_j\|_2^2 + R) \quad (1.1)$$

dove  $w_{ij}$  sono pesi introdotti per rimuovere *outliers* ed  $R$  è un termine di regolarizzazione atto a garantire l’esistenza di una soluzione.

---

<sup>4</sup>La forma di una distribuzione di colore tende ad assumere un’estensione lineare o planare a causa di ombreggiature, luci, e per la mescolanza di più colori. Ciò significa che, applicando una trasformazione lineare affine alle distribuzioni di colore in un’immagine, è possibile trasformare i colori e anche levigarli [4; 30].



**Figura 1.3:** Esempio di applicazione del filtraggio guidato

Lo sfondo alberato risulta non allineato tra le immagini in input (a) e guida (b). Questo porta a sfocature nel risultato della combinazione delle due immagini tramite GF classici (c). Il metodo proposto (d) riesce a gestire questo problema.

Nell'applicazione di questo GF a *regioni dinamiche*, come foglie su alberi o superfici d'acqua, l'errore commesso nell'approssimazione aumenta a causa del disallineamento tra i corrispondenti  $\mathbf{p}_j$  e  $\mathbf{I}_j$ . Questo a sua volta causa un decremento delle norme delle matrici di trasformazione ( $\|\mathbf{A}_i\| \rightarrow 0$ ) e, di conseguenza, una riduzione del contrasto di queste patch:  $\mathbf{A}_i \mathbf{I}_j \rightarrow 0$  (Figura 1.3).

Il metodo qui proposto è costruito apposta per gestire le regioni dinamiche e, per ottenere questo, lavora sul rendere conformi tra di loro le due *matrici di covarianza* della coppia di patch. Questo approccio ottiene l'effetto desiderato—di trasformare una distribuzione di colore in un'altra—grazie al fatto che la forma assunta da una distribuzione di colore è numericamente rappresentabile tramite la sua matrice di covarianza. Inoltre, visto che le due matrici sono calcolabili separatamente, è possibile rilassare la corrispondenza pixel-per-pixel ed ottenere così buoni risultati sulle regioni dinamiche. Tuttavia, con il mancato utilizzo di queste corrispondenze, possono sorgere decolorazioni (intese come *asperità* di colore) durante la trasformazione, rendendo

la qualità delle *regioni statiche* inferiore a quella dei GF classici e, in particolare, del GF proposto da Iwata et al. [30].

Sia il metodo proposto che il metodo convenzionale presentano, quindi, vantaggi e svantaggi su regioni pressoché complementari. È quindi possibile combinare i due metodi per ottenere un risultato che presenta i vantaggi di entrambe le idee e riduce al minimo i loro svantaggi; questo viene fatto sfruttando un classificatore addestrato in modo da riuscire a distinguere tra regioni statiche e dinamiche dell'immagine ed utilizzando il metodo più consono ad ogni regione dell'immagine.

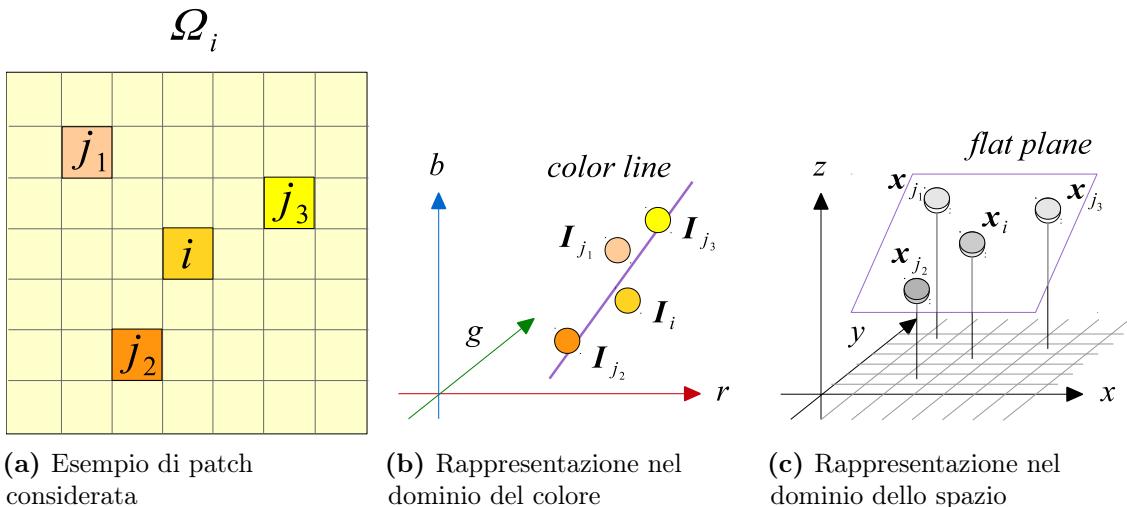
Sfortunatamente, a causa delle differenze sostanziali nei colori dell'immagine di input e dell'immagine guida, non è possibile utilizzare metodi esistenti di elaborazione di immagini o video. Il classificatore basa le sue decisioni quindi sugli errori commessi nella trasformazione (Equazione 1.1) e sulle norme delle matrici usate per riconoscere la comparsa di sfocature nell'immagine in uscita. Come classificatore viene adottato un Support Vector Machine (SVM) [34; 35], mentre il GF usato per il calcolo delle *feature* necessarie alla classificazione è [30].

## Capitolo 2

# Approssimazioni locali di immagini a superfici planari

In questo capitolo, verrà prima mostrata l'equazione di base per il LLM per il *dominio del colore* e, successivamente, verranno descritti il LLM per il *dominio dello spazio* e la sua risoluzione.

Innanzitutto è necessario definire le convenzioni utilizzate per gestire insiemi di pixel locali: chiamando  $i$  l'indice del pixel analizzato,  $\Omega_i$  corrisponde ad una patch quadrata centrata nel pixel  $i$ , mentre  $j \in \Omega_i$  corrisponde ad un pixel vicino, incluso nella patch. Il numero di pixel nell'intera immagine e nella patch sono chiamati rispettivamente  $N$  ed  $M$ , con  $M < N$ .



**Figura 2.1:** La definizione del LLM adottata nei domini del colore e dello spazio

I colori dei pixel componenti una patch (a) sono rappresentabili con una color-line (b), mentre le loro intensità sono rappresentabili con un piano (c).

## 2.1 Modello Lineare Locale nel dominio del colore

Il LLM convenzionale, proposto da Levin et al. [1] per l’*image matting* (Figura 1.1) ed accennato nell’introduzione (Sezione 1.1), è definibile matematicamente tramite la seguente trasformazione lineare, chiamando  $\mathbf{I}_j = [I_j^r, I_j^g, I_j^b]^\top \in \mathbb{R}^3$  i colori inclusi in una patch e  $\mathbf{a}_i \in \mathbb{R}^3$  e  $b_i \in \mathbb{R}$  rispettivamente i vettori di trasformazione e gli *offset* assegnati alla patch:

$$\arg \min_{p, \mathbf{a}, b} \sum_i \left( \sum_{j \in \Omega_i} w_{ij} \left\| \mathbf{a}_i^\top \mathbf{I}_j + b_i - p_j \right\|_2^2 + R_{\|\cdot\|^2} \right) + C \quad (2.1)$$

dove  $p_j \in \mathbb{R}$  sono le intensità da ottenere,  $\|\cdot\|_2$  corrisponde alla norma  $\ell_2$  e  $R_{\|\cdot\|^2}$  è un termine di regolarizzazione—utile, ad esempio, a fornire stabilità numerica—per le variabili  $\mathbf{a}_i$  e  $b_i$ , basato sulla norma  $\ell_2$  o sulla norma di Frobenius<sup>1</sup>,  $C$  denota altri vincoli di data fidelity e di regolarizzazione, e  $w_{ij} \geq 0$  sono pesi utilizzabili per eliminare gli outliers.

Una procedura tipica utilizzabile per ottenere la soluzione ottima  $p_j^*$  [1–3] consiste nel risolvere Equazione 2.1 rispetto ad  $\mathbf{a}_i$  e  $b_i$ , dopodiché sostituirli nuovamente in Equazione 2.1 e risolvere infine l’equazione risultante rispetto a  $p_j$ . Nel caso della applicazione ai filtri guidati classici [24; 29; 30], viene utilizzata un’ottimizzazione alternata, ottenuta fissando i valori iniziali di  $p_j$  tramite un’immagine guida e procedendo alla risoluzione prima rispetto a  $\mathbf{a}_i$  e  $b_i$  e successivamente rispetto a  $p_j$ .

## 2.2 Modello Lineare Locale nel dominio dello spazio

Il LLM proposto è ispirato dallo stereo fotometrico [5; 16], un metodo basato sull’ottimizzazione che, da una serie di fotografie di un oggetto illuminato da più angolazioni, punta ad ottenere una rappresentazione tridimensionale dell’oggetto stesso. Il LLM prende spunto in particolare dalla fase di recupero della mappa di profondità a partire dalla mappa di vettori normali costruita combinando le immagini in ingresso.

Chiamando un set di coordinate  $(x_j, y_j)$  ed il valore<sup>2</sup> del pixel corrispondente  $z_j = I(x_j, y_j) \in \mathbb{R}$ , vengono definiti un vettore di coordinate  $\mathbf{x}_j = [x_j, y_j, z_j]^\top$  ed un vettore unitario  $\mathbf{n}_i =$

---

<sup>1</sup>La norma di Frobenius corrisponde alla norma matriciale  $L_{2,2}$ , ossia

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

---

<sup>2</sup>L’ottimizzazione in questo caso viene effettuata per ogni canale separatamente.

$[a_i, b_i, c_i]^\top$  s.t.  $\|\mathbf{n}_i\|_2 = 1$ . L'equazione di un piano nello spazio è definibile quindi come:

$$a_i x_j + b_i y_j + c_i z_j + d_i = \mathbf{n}_i^\top \mathbf{x}_j + d_i = 0 \quad (2.2)$$

dove  $d_i$  indica la distanza del piano dall'origine degli assi. Per semplicità nei calcoli successivi, Equazione 2.2 viene divisa per  $-c_i$  e vengono definite le nuove variabili  $\mathbf{n}'_i = [a'_i, b'_i]^\top = [-\frac{a_i}{c_i}, -\frac{b_i}{c_i}]^\top$ ,  $\mathbf{x}'_j = [x_j, y_j]^\top$  e  $d'_i = -\frac{d_i}{c_i}$ . Così facendo, Equazione 2.2 assume una forma simile all'argomento della norma di Equazione 2.1, ossia:

$$\mathbf{n}'_i^\top \mathbf{x}'_j + d'_i - z_j = 0 \quad (2.3)$$

L'equazione finale viene ottenuta considerando un'ottimizzazione su tutta l'immagine, definendo la funzione energia come:

$$E(z, d') = \sum_i \sum_{j \in \Omega_i} w_{ij} \left\| \mathbf{n}'_i^\top \mathbf{x}'_j + d'_i - z_j \right\|_2^2 + C \quad (2.4)$$

In questa trattazione, il termine di regolarizzazione di Equazione 2.1 non viene utilizzato ( $R_{\|\cdot\|_2} = 0$ ), mentre alcuni dei possibili utilizzi dei vincoli  $C$  verranno trattati in Sottosezione 2.2.4 e Sezione 4.3.2.

### 2.2.1 Soluzione rispetto a $d'$

L'ottimo locale  $d'^*$  viene ottenuto ponendo la derivata prima di Equazione 2.4 uguale a zero, come in [24; 30]. Ciò risulta in:

$$d'^*_i = \bar{z}_i - \mathbf{n}'_i^\top \bar{\mathbf{x}}'_i \quad (2.5)$$

dove  $\bar{z}_i = \frac{\sum_j w_{ij} z_j}{\sum_j w_{ij}}$  e  $\bar{\mathbf{x}}'_i = \frac{\sum_j w_{ij} \mathbf{x}'_j}{\sum_j w_{ij}}$  denotano le medie pesate. Sostituendo Equazione 2.5 in Equazione 2.4 per eliminare  $d'_i$ , si ottiene l'equazione principale del metodo proposto:

$$E(z) = \sum_i \sum_{j \in \Omega_i} w_{ij} \left\| \mathbf{n}'_i^\top \tilde{\mathbf{x}}'_j - \tilde{z}_j \right\|_2^2 \quad (2.6)$$

dove  $\tilde{\mathbf{x}}'_j = \mathbf{x}'_j - \bar{\mathbf{x}}'_i$  e  $\tilde{z}_j = z_j - \bar{z}_i$  corrispondono alle variabili centrate rispetto alla loro media pesata.

### 2.2.2 Soluzione rispetto a $z$

Operando similmente al caso precedente, Equazione 2.6 viene derivata—questa volta rispetto a  $z_j$ —ed il valore ottimo  $z_j^*$  viene ottenuto ponendo la derivata a zero. Il differenziale è in questo

caso complesso a causa delle operazioni di centratura effettuate in Equazione 2.6. Per questo motivo viene introdotto un vettore  $\mathbf{z} = [z_1, \dots, z_N]^\top \in \mathbb{R}^N$  e l'operazione viene espressa tramite una matrice di centratura  $\mathbf{V}_i \in \mathbb{R}^{M \times N}$  come descritto in Appendice A. Introducendo inoltre i vettori di coordinate  $\mathbf{x} = [x_1, \dots, x_N]^\top$  e  $\mathbf{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$  ed una matrice diagonale di pesi  $\mathbf{W}_i = \text{diag}(w_{ij_1}, \dots, w_{ij_M}) \in \mathbb{R}^{M \times M}$ , Equazione 2.6 può essere riscritta come:

$$E(\mathbf{z}) = \sum_i \left\| \mathbf{W}_i^{\frac{1}{2}} (a'_i \mathbf{V}_i \mathbf{x} + b'_i \mathbf{V}_i \mathbf{y} - \mathbf{V}_i) \right\|_2^2 \quad (2.7)$$

Derivando Equazione 2.7 rispetto a  $\mathbf{z}$  e chiamando  $\mathbf{U}_i = \mathbf{V}_i^\top \mathbf{W}_i \mathbf{V}_i$ , si ottiene infine la seguente equazione:

$$\left( \sum_i \mathbf{U}_i \right) \mathbf{z} = \sum_i (a'_i \mathbf{U}_i \mathbf{x} + b'_i \mathbf{U}_i \mathbf{y}) \quad (2.8)$$

Questa equazione si presenta sotto la forma standard  $\mathbf{A}\mathbf{x} = \mathbf{b}$  e, dato che  $\mathbf{A} = \sum_i \mathbf{U}_i \in \mathbb{R}^{N \times N}$  è una matrice simmetrica, la soluzione può essere ottenuta applicando il Gradiente Coniugato (CG) [25]. Nell'implementazione viene utilizzato un metodo di calcolo efficiente (Appendice B) per ridurre la complessità della computazione, necessario per ottenere tempi di elaborazione ragionevoli a causa dell'enorme dimensione delle matrici.

Si noti che il valore ottenuto per  $z_i^*$  risulta *incompleto* se non vengono impostate le condizioni al contorno appropriate nel termine di regolarizzazione  $C$  (Equazione 2.4). In modo simile alle equazioni di Poisson con condizioni al bordo di Neumann<sup>3</sup>, il rango della matrice non è pieno:  $\text{rank}(\mathbf{A}) = N - 1$ . Ciò significa che esiste un autovettore  $\mathbf{v}$ —corrispondente all'autovalore nullo— tale per cui  $\mathbf{A}\mathbf{v} = 0$  e la soluzione presenta quindi un'ambiguità del tipo  $\mathbf{A}(\mathbf{z} + s\mathbf{v}) = \mathbf{b}, s \in \mathbb{R}$ . È quindi necessario aggiungere un offset  $s\mathbf{v}$  alla soluzione per ottenere un risultato visualmente piacevole:

$$\hat{\mathbf{z}}^* = \mathbf{z}^* + s\mathbf{v} \quad (2.9)$$

Il valore di  $s\mathbf{v}$  verrà specificato per ogni applicazione mostrata.

### 2.2.3 Soluzione rispetto a $\mathbf{n}'$

Come verrà mostrato in Sezione 2.3, esistono applicazioni per le quali il LLM proposto può essere utilizzato con l'intento di trovare i valori di  $\mathbf{n}'$ . Per fare ciò viene minimizzata Equazione 2.6 rispetto ad  $\mathbf{n}'$ :

$$\arg \min_{\mathbf{n}'} \left( E(\mathbf{n}') = \sum_i \sum_{j \in \Omega_i} w_{ij} \left\| \mathbf{n}'^\top \tilde{\mathbf{x}}'_j - \tilde{z}_j \right\|_2^2 \right) \quad (2.10)$$

---

<sup>3</sup>Anch'esse rappresentabili come  $\mathbf{A}\mathbf{z} = \mathbf{b}$ , utilizzando come matrice  $\mathbf{A}$  una matrice di filtraggio Laplaciana.

Il valore ottimo  $\mathbf{n}'_i^*$  viene ottenuto nuovamente ponendo la derivata prima di Equazione 2.10 a zero, risultando in:

$$\mathbf{n}'_i^* = \left( \sum_{j \in \Omega_i} w_{ij} \tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^\top \right)^{-1} \left( \sum_{j \in \Omega_i} w_{ij} \tilde{z}_j \tilde{\mathbf{x}}_j \right) \quad (2.11)$$

Dato che la dimensione della matrice inversa è  $2 \times 2$ , viene ottenuta semplicemente ed in modo analitico utilizzando la formula standard dei cofattori. In aggiunta, quando i pesi sono uniformi ( $w_{ij} = 1$ ), Equazione 2.11 può essere risolta efficientemente usando l'*immagine integrale* [24; 27]. Infine, la matrice di covarianza  $\sum_j \tilde{\mathbf{x}}_j \tilde{\mathbf{x}}_j^\top$  diviene diagonale e, se la dimensione delle patch è fissa, la stessa matrice di covarianza viene ottenuta per ogni patch, indipendentemente dalle loro coordinate  $x_i$  ed  $y_i$ .

#### 2.2.4 Combinazione con Equazioni Differenziali convenzionali

Confrontato con la soluzione convenzionale dell'equazione di Poisson Equazione 4.1 che considera una derivata in un'area ristretta, il LLM lavora su una patch locale ma ampia. Ciò permette di avere una robustezza al rumore rapidamente crescente al crescere della dimensione della patch. Questo va, però, a discapito della preservazione dei bordi (esempi di questo problema sono visibili in Sezione 5.1). Anche con una patch di dimensione  $3 \times 3$ , la capacità di ricostruzione dei bordi risulta leggermente inferiore rispetto a quella fornita dai filtri convenzionali come la differenza finita in avanti (intesa come  $\Delta z_i = z_i - z_{i+1}$ ).

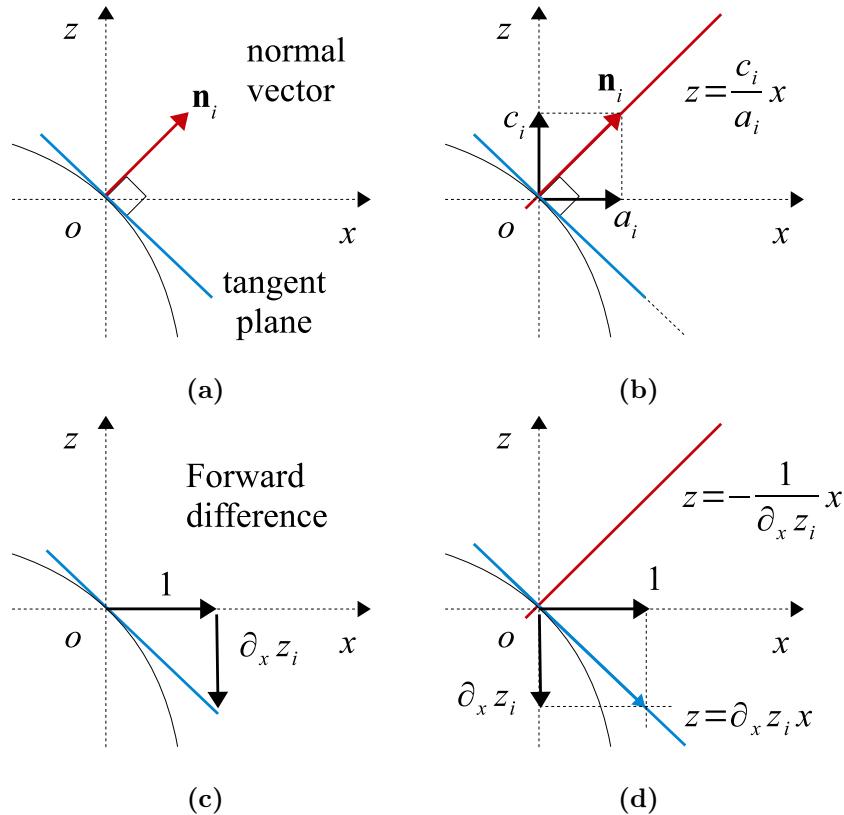
Un modo per ovviare a questo problema può essere la combinazione delle funzioni energia del metodo proposto (Equazione 2.6) e del metodo convenzionale (Equazione 4.1) per compensare le mancanze di ognuno dei due:

$$\arg \min_z \alpha \sum_i \sum_{j \in \Omega_i} w_{ij} \left\| \mathbf{n}'_i^\top \tilde{\mathbf{x}}'_j - \tilde{z}_j \right\|_2^2 + (1 - \alpha) \sum_i \|\nabla z_i - \dot{\mathbf{n}}'_i\|_2^2 \quad (2.12)$$

dove  $\alpha \in [0, 1]$  è un parametro di *blending*, e  $\dot{\mathbf{n}}'_i$  indica i gradienti calcolati tramite la differenza finita in avanti. Visto che entrambe sono equazioni quadratiche, la loro combinazione risulta semplice.

### 2.3 Relazione con i Gradient Domain Operators

Per chiarire la relazione tra un vettore normale utilizzato nel LLM ed i gradienti di intensità usati nei GDO, viene qui presa come esempio la ricostruzione della mappa di profondità da una



**Figura 2.2:** Relazione tra un vettore normale ed il gradiente di profondità sul piano  $x$ - $z$

Quando il vettore normale viene decomposto lungo le direzioni  $x$  e  $z$  (a) utilizzando i vettori di base  $[a_i, 0, 0]$  e  $[0, 0, c_i]$ , l'equazione di una retta passante per il vettore normale è data da  $z = \frac{c_i}{a_i}x$  (b); per quanto riguarda il gradiente del piano tangente (c), la variazione di profondità associata alla differenza in avanti di un pixel è data da  $\partial_x z_i$  ed un'equazione perpendicolare al piano (d) è data da  $z = -\frac{1}{\partial_x z_i}x$ . Ciò significa che esiste una relazione di identità  $\frac{c_i}{a_i} = -\frac{1}{\partial_x z_i}$ .

mappa normale nell'ambito dello stereo fotometrico [5; 16]<sup>4</sup>.

In questa applicazione, i gradienti  $\nabla z_i = [\partial_x z_i, \partial_y z_i]^\top$  (dove  $\partial_x$  e  $\partial_y$  sono definiti come filtri differenziali) vengono trovati osservando l'angolo di un vettore normale ad ogni pixel  $\mathbf{n}_i = [a_i, b_i, c_i]^\top$ .

Come mostrato graficamente per il piano  $xz$  in Figura 2.2,  $\mathbf{n}$  giace sulla retta di equazione

$$z = \frac{c_i}{a_i}x \quad (2.13)$$

mentre la differenza in avanti  $\partial_x z_i$  porta alla definizione della retta

$$z = \partial_x z_i x \quad (2.14)$$

---

<sup>4</sup>Questo concetto verrà ripreso in Capitolo 4, dove verranno mostrate le applicazioni del metodo in un algoritmo di recupero della profondità per lo stereo fotometrico, oltre che nella rimozione di riflessi e nella rimozione di ombre.

La retta ortogonale a questa risulta essere per definizione parallela alla retta di Equazione 2.13 e descritta dall'equazione

$$z = -\frac{1}{\partial_x z_i} x \quad (2.15)$$

pertanto si può dedurre una relazione del tipo  $\frac{c_i}{a_i} = -\frac{1}{\partial_x z_i}$  tra i due vettori. La stessa relazione vale per il piano  $yz$ , ossia  $\frac{c_i}{b_i} = -\frac{1}{\partial_y z_i}$ . Queste relazioni possono essere infine riassunte come:

$$[\partial_x z_i, \partial_y z_i]^\top = \left[ -\frac{a_i}{c_i}, -\frac{b_i}{c_i} \right]^\top \implies \nabla z_i = \mathbf{n}'_i \quad (2.16)$$

Ciò significa che i vettori  $\mathbf{n}'$  utilizzati nel LLM applicato al dominio dello spazio corrispondono ai gradienti  $\nabla z$  dell'immagine e che, quindi, il metodo può essere sostituito direttamente sia nelle fasi di calcolo dei gradienti di un'immagine (Sottosezione 2.2.3), sia nelle fasi di ricostruzione di un'immagine a partire dai suoi gradienti (Sottosezione 2.2.2).

## Capitolo 3

# Trasformazioni di colore tra coppie di immagini

In questo capitolo, verrà prima mostrato il filtro guidato proposto in questo lavoro (accennato in Sezione 1.2), basato sulla trasformazione tra covarianze, e la sua risoluzione. Successivamente verranno analizzate alcune cause della comparsa di decolorazioni e verranno proposti alcuni rimedi. Verrà infine trattato l'utilizzo di un classificatore per unire i risultati ottenuti dall'uso del GF qui sviluppato con i risultati ottenuti dall'utilizzo del GF proposto da Iwata et al. [30].

### 3.1 Filtro guidato per regioni dinamiche

L'utilizzo di filtri guidati classici per il restauro di immagini deteriorate da rumore e saturazione dell'intensità del colore porta a risultati che presentano sfocature e ridotto contrasto nelle regioni in cui le due immagini utilizzate non siano perfettamente allineate. Per gestire queste regioni è stato sviluppato un GF in grado di rilassare la corrispondenza pixel-per-pixel.

L'obiettivo del metodo qui proposto per le regioni dinamiche resta comunque quello di trovare le matrici di trasformazione  $\mathbf{A}_i$  ed i vettori di traslazione  $\mathbf{b}_i$ . Per ottenerli, però, viene tenuto conto del fatto che la forma di una distribuzione di colore è numericamente rappresentabile tramite la sua matrice di covarianza e, per questo, viene formulato un problema di ottimizzazione con un vincolo aggiunto sulle covarianze:

$$\text{Eq. (1.1) s.t. } \text{Cov}(\{\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i\}) = \text{Cov}(\{\mathbf{p}_j\}) \quad (3.1)$$

Dove la covarianza viene calcolata sfruttando dei pesi  $w_{ij}$ , usati per la rimozione di *outliers*,

risultando quindi definita come:

$$\text{Cov}(\{x_j\}) = \frac{1}{\sum_j w_{ij}} \sum_{j \in \Omega_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^\top \in \mathbb{R}^{3 \times 3} \quad (3.2)$$

Dove  $\bar{\mathbf{x}} = \frac{1}{\sum_j w_{ij}} \sum_{j \in \Omega_i} w_{ij} \mathbf{x}_j$  denota la media pesata. La selezione dei pesi dipende dall'applicazione e, nell'applicazione mostrata in Sezione 5.2, vengono calcolati implicitamente sfruttando un'implementazione efficiente del filtraggio bilaterale [36].

L'aggiunta di questo vincolo lascia comunque possibile l'esistenza di più di una soluzione: il problema sarà descritto in Sezione 3.2.

### 3.1.1 Soluzione rispetto a $\mathbf{b}_i$

Espandendo il vincolo sulle covarianze ( $\text{Cov}(\{\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i\}) = \text{Cov}(\{\mathbf{p}_j\})$ ) di Equazione 3.1 con la definizione di covarianza data in Equazione 3.2 e sostituendo  $\mathbf{I}'_j = \mathbf{I}_j - \bar{\mathbf{I}}_i$  e  $\mathbf{p}'_j = \mathbf{p}_j - \bar{\mathbf{p}}_i$ , si ottiene:

$$\begin{aligned} 0 &= \text{Cov}(\{\mathbf{A}_i \mathbf{I}'_j + \mathbf{b}_i\}) - \text{Cov}(\{\mathbf{p}'_j\}) \\ &= \mathbf{A}_i \left( \frac{1}{\sum_j w_{ij}} \mathbf{I}'_j \mathbf{I}'_j^\top \right) \mathbf{A}_i^\top - \left( \frac{1}{\sum_j w_{ij}} \mathbf{p}'_j \mathbf{p}'_j^\top \right) \\ &= \mathbf{A}_i \text{Cov}(\{\mathbf{I}'_j\}) \mathbf{A}_i^\top - \text{Cov}(\{\mathbf{p}'_j\}) \end{aligned} \quad (3.3)$$

In questo processo, il vettore di *offset*  $\mathbf{b}_i$  scompare, risultando apparentemente scorrelato dall'equazione. Derivando invece Equazione 1.1 rispetto a  $\mathbf{b}_i$  e ponendo il risultato a zero, si ottiene  $\mathbf{b}_i = \bar{\mathbf{p}}_i - \mathbf{A}_i \bar{\mathbf{I}}_i$ . Sostituendo nuovamente l'espressione trovata per  $\mathbf{b}_i$  all'interno della trasformata affine  $\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i$  si ottiene  $\mathbf{A}_i(\mathbf{I}_j - \bar{\mathbf{I}}_i) + \bar{\mathbf{p}}_i = \mathbf{A}_i \mathbf{I}'_j + \bar{\mathbf{p}}_i$ . Confrontando questo risultato con l'argomento della covarianza in Equazione 3.3, si ottiene:

$$\mathbf{b}_i = \bar{\mathbf{p}}_i \quad (3.4)$$

### 3.1.2 Soluzione rispetto a $\mathbf{A}_i$

La soluzione rispetto ad  $\mathbf{A}_i$  si basa sui concetti di **decorrelazione** e di **ricostruzione** usati nella Principal Component Analysis (PCA).

Innanzitutto, le matrici di covarianza  $\text{Cov}(\{\mathbf{I}'_j\})$  e  $\text{Cov}(\{\mathbf{p}'_j\})$  sono scomposte usando la SVD:

$$\text{Cov}(\{\mathbf{I}'_j\}) = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{U}_i^\top, \quad \text{Cov}(\{\mathbf{p}'_j\}) = \mathbf{V}_i \boldsymbol{\Xi}_i \mathbf{V}_i^\top \quad (3.5)$$

Successivamente vengono separate nel prodotto di due matrici:

$$\mathbf{Q}_i \mathbf{Q}_i^\top = (\mathbf{U}_i \boldsymbol{\Sigma}_i^{\frac{1}{2}}) (\boldsymbol{\Sigma}_i^{\frac{1}{2}} \mathbf{U}_i^\top), \quad \mathbf{R}_i \mathbf{R}_i^\top = (\mathbf{V}_i \boldsymbol{\Xi}_i^{\frac{1}{2}}) (\boldsymbol{\Xi}_i^{\frac{1}{2}} \mathbf{V}_i^\top) \quad (3.6)$$

dove  $\mathbf{U}_i = [\mathbf{u}_{1,i} | \mathbf{u}_{2,i} | \mathbf{u}_{i,3}]$  e  $\mathbf{V}_i = [\mathbf{v}_{1,i} | \mathbf{v}_{2,i} | \mathbf{v}_{i,3}] \in \mathbb{R}^{3 \times 3}$  sono matrici ortogonali composte dagli autovettori delle matrici di covarianza, rappresentanti i tre assi principali di ogni distribuzione di colore e  $\boldsymbol{\Sigma} = \text{diag}(\sigma_{1,i}^2, \sigma_{2,i}^2, \sigma_{3,i}^2)$  e  $\boldsymbol{\Xi} = \text{diag}(\xi_{1,i}^2, \xi_{2,i}^2, \xi_{3,i}^2) \in \mathbb{R}^{3 \times 3}$  sono le matrici degli autovalori delle matrici di covarianza, rappresentanti le varianze lungo gli assi principali di ogni distribuzione, con  $\sigma_k \geq \sigma_{k+1}$  e  $\xi_k \geq \xi_{k+1}$ .

**Decorrelazione** Il calcolo di  $\{\tilde{\mathbf{I}}'_j\} = \mathbf{Q}_i^{-1}\{\mathbf{I}'_j\}$  permette di decorrelare ogni patch, in modo che la sua covarianza corrisponda ad una matrice identità:  $\text{Cov}(\{\tilde{\mathbf{I}}'_j\}) = \mathbf{1}$

**Ricostruzione** Il calcolo di  $\{\hat{\mathbf{p}}'_j\} = \mathbf{R}_i\{\tilde{\mathbf{I}}'_j\}$  ricostruisce i dati, in modo tale da rendere la loro covarianza uguale a quella dell'immagine in input:  $\text{Cov}(\{\hat{\mathbf{p}}'_j\}) = \text{Cov}(\{\mathbf{p}'_j\})$ .

La soluzione di  $\mathbf{A}_i$  risulta essere quindi:

$$\mathbf{A}_i = \mathbf{R}_i \mathbf{Q}_i^{-1} = \left( \mathbf{V}_i \boldsymbol{\Xi}_i^{\frac{1}{2}} \right) \left( \boldsymbol{\Sigma}_i^{-\frac{1}{2}} \mathbf{U}_i^\top \right) \quad (3.7)$$

Sostituendo le equazioni (Equazione 3.5), (Equazione 3.6), (Equazione 3.7) in Equazione 3.3, si può notare che il vincolo Equazione 3.1 viene soddisfatto:

$$\mathbf{R}_i \mathbf{Q}_i^{-1} (\mathbf{Q}_i \mathbf{Q}_i^\top) \mathbf{Q}_i^\top \mathbf{R}_i^\top - \mathbf{R}_i \mathbf{R}_i^\top = 0 \quad (3.8)$$

Questo, però, non rende unica la soluzione, in quanto l'ordine delle componenti di  $\mathbf{Q}_i$  e  $\mathbf{R}_i$  è arbitrario. Alcune di queste soluzioni potrebbero causare decolorazioni; un metodo per gestire questo problema è descritto in Sezione 3.2.

### 3.1.3 Trasformazione globale del colore

La trasformazione locale del colore di ogni patch è assunta essere  $\{\mathbf{A}_i \mathbf{I}_i + \mathbf{b}_i\}_{j \in \Omega_i}$  mentre, in pratica, la trasformazione per l'intera immagine è fatta in modo da ottenere i valori  $\mathbf{q}_i^*$  ottimali utilizzando tutti i valori calcolati per  $\mathbf{A}_i$  e  $\mathbf{b}_i$ . La funzione energia utilizzata è definita come:

$$\arg \min_q \sum_i \sum_{j \in \Omega_i} w_{ij} \|\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i - \mathbf{q}_j\|_2^2 \quad (3.9)$$

Ponendo i pesi costanti  $w_{ij} = 1$ , Equazione 3.9 corrisponde al GF standard [4].

La soluzione viene ottenuta come:

$$\mathbf{q}_i^* = \bar{\mathbf{A}}_i \mathbf{I}_i + \bar{\mathbf{b}}_i \quad (3.10)$$

dove  $\bar{\mathbf{A}}_i = \frac{1}{N} \sum_{j \in \Omega_i} \mathbf{A}_i$  e  $\bar{\mathbf{b}}_i = \frac{1}{N} \sum_{j \in \Omega_i} \mathbf{b}_i$  sono le medie dei valori attorno al pixel  $i$ ,  $N$  è il numero di pixels ed i pesi sono normalizzati in modo da ottenere  $N = \sum_j w_{ij}$ . Quando  $w_{ij} \in \{0, 1\}$ , un metodo approssimato ma rapido per calcolare le medie viene descritto in [29]. Negli altri casi le medie sono ottenute tramite la convoluzione con filtri tempo-varianti con coefficienti  $w_{ij}$  [30].

## 3.2 Analisi e riduzione delle decolorazioni

Alcune matrici di trasformazione  $\mathbf{A}_i$  ottenute come descritto nella Sottosezione 3.1.2 hanno degli effetti collaterali e possono causare decolorazioni quando applicate direttamente. In questa sezione sono mostrati due problemi che causano decolorazioni e le relative soluzioni adottate.

### 3.2.1 Correzione dell'ordine e del segno degli autovettori

Nella derivazione della matrice di trasformazione del colore, vengono utilizzate le matrici di autovettori  $\mathbf{U}_i$  e  $\mathbf{V}_i$  (Equazione 3.7). Queste presentano però due ambiguità:

1. L'ordine degli autovettori principali.

Questo problema sorge dal fatto che la distribuzione di colore di una regione uniforme diviene sferica, con la conseguenza che l'ordine degli assi è facilmente scambiabile a causa del rumore—e.g.,  $[\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3] \rightarrow [\mathbf{u}_1|\mathbf{u}_3|\mathbf{u}_2]$ .

2. La direzione (i.e., il segno) degli autovettori.

Questo problema riguarda il fatto che in genere il segno degli autovettori non viene specificato, lasciando quindi la possibilità di un'inversione di segno in ogni colonna—e.g.,  $[\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3] \rightarrow [\pm \mathbf{u}_1| \pm \mathbf{u}_2| \pm \mathbf{u}_3]$ .

Per gestire questi problemi, vengono fatte due assunzioni:

1. Una coppia di patch presenta distribuzioni di colore di forma simile tra di loro. Questo implica che l'ordine degli assi principali sia lo stesso. Inoltre, nel caso in cui la distribuzione divenga sferica in una regione di colore uniforme e che, quindi, l'ordine degli assi non sia decidibile, questo non risulta un problema visto che la varianza lungo ogni asse è simile.
2. Coppie corrispondenti di autovettori di una coppia di patch hanno lo stesso orientamento. La direzione è quindi correggibile usando il segno del prodotto scalare:  $\mathbf{u}_k := \mathbf{u}_k * \text{sgn}(\mathbf{u}_k^\top \mathbf{v}_k)$ .

### 3.2.2 Correzione degli autovalori di $\Xi_i$ e $\Sigma_i$

La matrice  $\mathbf{R}_i$  in  $\mathbf{A}_i = \mathbf{R}_i \mathbf{Q}_i^{-1}$ , i.e.,  $\mathbf{R}_i = \mathbf{V}_i \Xi_i^{\frac{1}{2}}$  viene ottenuta da dati rumorosi, e l'influenza del rumore appare principalmente in  $\Xi_i$ . Ciò significa che, quando una distribuzione di colore viene alterata dall'influenza del rumore, le sue varianze  $\{\xi_1^2, \xi_2^2, \xi_3^2\}$  aumentano. Questo porta ad un'espansione eccessiva della distribuzione trasformata e, quindi, ad un'enfasi eccessiva del colore.

Dall'esperienza descritta in [37; 38], le deviazioni standard lungo il secondo e, in particolare, il terzo asse ( $\{\xi_2, \xi_3\}$ ) sono collegate alle decolorazioni: anche nel caso in cui i dati non contengano alcun rumore, un'espansione della distribuzione del colore lungo questi due assi risulta in un'enfatizzazione delle decolorazioni.

La gestione di questi problemi viene fatta basandosi sui rapporti tra le coppie di autovalori corrispondenti delle matrici  $\Xi_i^{\frac{1}{2}}$  e  $\Sigma_i^{-\frac{1}{2}}$ , ossia gli elementi della diagonale di  $\Xi_i^{\frac{1}{2}} \Sigma_i^{-\frac{1}{2}} = \text{diag}(\frac{\xi_1}{\sigma_1}, \frac{\xi_2}{\sigma_2}, \frac{\xi_3}{\sigma_3})$ , presente nel calcolo di  $\mathbf{R}_i \mathbf{Q}_i^{-1}$ . La riduzione di ogni elemento di  $\Xi_i$  viene fatta come segue:

- Il **primo** autovalore viene utilizzato senza effettuare alcuna correzione;
- Il **secondo** autovalore viene sostituito direttamente con il corrispondente ottenuto senza rumore—i.e.,  $\xi_2 := \sigma_2 \implies \frac{\xi_2}{\sigma_2} = 1$ ;
- Il **terzo** autovalore, il quale è la causa principale delle decolorazioni, viene mantenuto il più piccolo possibile, sostituendolo con il più piccolo tra  $\xi_3$  e  $\sigma_3$ —i.e.,  $\xi_3 := \min(\xi_3, \sigma_3) \implies \frac{\xi_3}{\sigma_3} \leq 1$ .

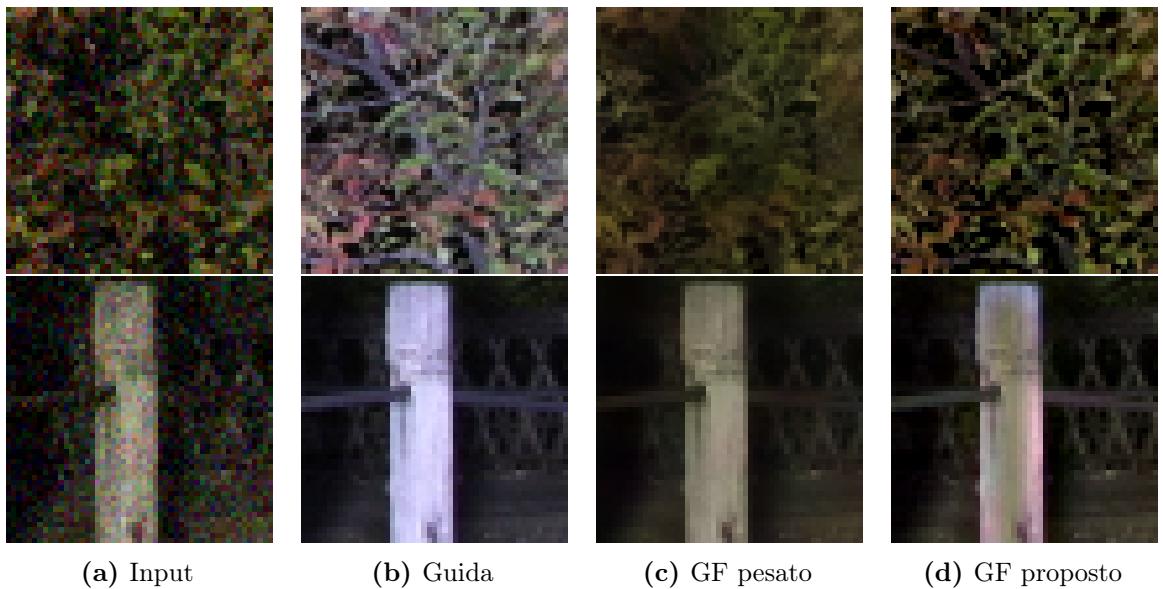
## 3.3 Classificazione di regioni statiche e dinamiche, e blending

I risultati dell'utilizzo di entrambi i GF sono mostrati in Figura 3.1: si può notare in Figura 3.1d che il GF proposto in questo lavoro produce un'immagine naturale e senza sfumature nella regione dinamica, mentre purtroppo causa la comparsa di decolorazioni nella regione statica a causa della presenza di rumore. Visto che ogni GF ha vantaggi e svantaggi, l'obiettivo di questo capitolo è di riuscire ad utilizzare il GF adeguato in modo selettivo, regione per regione, in modo da ottenere un risultato più naturale.

Analizzando le matrici di trasformazione ottenute con i GF esistenti, si nota che la presenza di sfocature è associabile alla norma delle matrici  $\|\mathbf{A}_i\|$ , i.e., giocano un ruolo da fattore di

---

<sup>1</sup>Per enfatizzare sia i problemi sia l'efficacia dei metodi, è stato aggiunto del rumore Gaussiano con deviazione standard 0.06.



**Figura 3.1:** Trasformazioni di colore e contrasto tramite GF, per regioni statiche e dinamiche.

Alto: risultati su una regione dinamica. Basso: risultati su una regione statica.

Da sinistra a destra: immagini in input, rumorose<sup>1</sup> ma con contrasto ideale (a); immagini guida, con messa a fuoco ideale ma contrasto alterato (b); risultato del GF [30] (c); risultato del GF proposto (d).

riscalatura del contrasto di ogni patch e, decrescendo, fanno sì che il valore di  $\mathbf{A}_i \mathbf{I}_i$  si avvicini a zero e, il valore finale di  $\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i$  tenderà alla media  $\mathbf{b}_i$  dei pixel della patch. Questa osservazione implica che anche le regioni dinamiche sono classificabili sulla base delle norme e degli errori di trasformazione di Equazione 1.1.

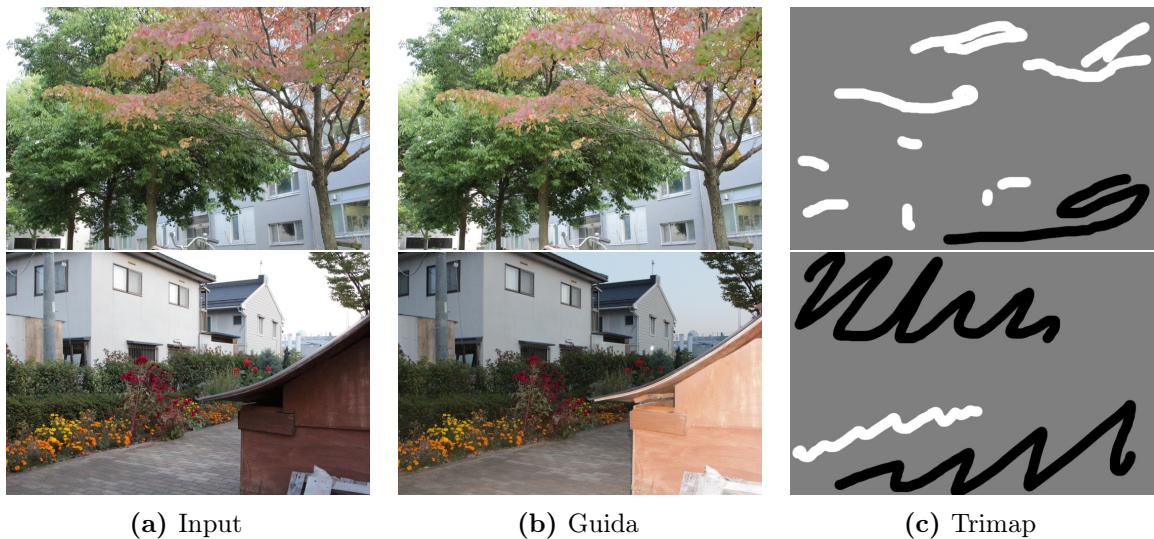
Per effettuare questa classificazione tra regioni statiche e regioni dinamiche, viene utilizzata una SVM [34; 39]<sup>2</sup>. La SVM richiede una fase di addestramento ed una di testing: i dettagli della prima sono descritti nella prossima sezione, mentre i risultati della seconda fase sono descritti in Sezione 5.2.

### 3.3.1 Addestramento della Support Vector Machine

Per la fase di addestramento, è stato costruito un *dataset* di triple di immagini organizzate come in Figura 3.2: alle due immagini di input e guida viene associata una maschera utilizzata per specificare le regioni statiche (tratti neri) e le regioni dinamiche (tratti bianchi) Figura 3.2c. I pixel corrispondenti a regioni statiche vengono etichettate come  $y_i = -1$ , le regioni dinamiche vengono etichettate come  $y_i = +1$ , mentre le regioni in grigio non vengono utilizzate.

---

<sup>2</sup>Nell'implementazione, è stato usato  $SVM^{perf}$  [35; 40; 41] con un radial basis function kernel [34; 39] ed i parametri di default.



**Figura 3.2:** Esempio di set di immagini utilizzate nella fase di addestramento della SVM

I *feature vectors* ( $\in \mathbb{R}^3$ ) usati per l’addestramento vengono poi definiti come composizione delle seguenti *features*:

- La norma di Frobenius<sup>3</sup> della matrice di trasformazione, i.e.,  $\|\mathbf{A}_i\|_F^2 \in \mathbb{R}$ ;
- La norma  $\ell_2$  del vettore di *offset*, i.e.,  $\|\mathbf{b}_i\|_2^2 \in \mathbb{R}$ .

Questa *feature* è pensata per le decolorazioni nelle regioni scure, dove la media dei pixel tende a zero e la forma delle distribuzioni di colore diviene sferica.

- Il termine di fedeltà:  $\sum_{j \in \Omega_i} w_{ij} \|\mathbf{A}_i \mathbf{I}_j + \mathbf{b}_i - \mathbf{p}_j\|_2^2 \in \mathbb{R}$ .

Utilizzando queste etichette e questi vettori di caratteristiche per ogni coppia di immagini nel *dataset*, viene effettuato l’addestramento della SVM.

### 3.3.2 Classificazione e metodi di blending

I risultati della classificazione, effettuata basandosi sull’etichettatura descritta nella Sottosezione 3.3.1, consistono in valori positivi ( $y_i > 0$ ) per le regioni dinamiche e valori negativi ( $y_i < 0$ ) per le regioni statiche.

---

<sup>3</sup>La norma di Frobenius corrisponde alla norma matriciale  $L_{2,2}$ , ossia

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

Il risultato finale viene ottenuto come *blending pixel-wise* dei risultati del GF pesato  $\mathbf{q}_i$  e del GF proposto  $\mathbf{q}'_i$ :

$$\mathbf{q}_i^* = (1 - \alpha(y_i))\mathbf{q}_i + \alpha(y_i)\mathbf{q}'_i \quad (3.11)$$

dove  $\alpha(\cdot) \in [0, 1]$  denota una funzione di *blending* come una sigmoide—e.g.,  $\alpha(y_i) = (1 + \exp(-\beta y_i))^{-1} \in [0, 1]$ , dove  $\beta$  è un parametro che controlla la larghezza della transizione. I risultati sperimentali ottenuti utilizzando questo metodo sono mostrati in Sezione 5.2.

## Capitolo 4

# Applicazioni del Modello Lineare Locale

### 4.1 Inter-conversione tra Vettori Normali e Gradienti

Nel caso dello stereo fotometrico convenzionale,  $z_i$  nell'Equazione 2.16 viene trattata come incognita mentre  $\mathbf{n}_i$  è dato da uno step di stima effettuato in precedenza. Per ottenere i valori ottimi  $z_i^*$ , viene risolta la seguente equazione di Poisson:

$$\arg \min_z \sum_i \|\nabla z_i - \mathbf{n}'_i\|_2^2 + C \quad (4.1)$$

dove  $C$  riassume i termini di regolarizzazione che, per semplicità, non verranno trattati in questa sezione. Il valore ottimo  $z_i^*$  viene ottenuto ponendo a zero la prima derivata di Equazione 4.1, ottenendo:

$$z_i^* = (\nabla^{*\top} \nabla)^{-1} (\nabla^{*\top} \mathbf{n}'_i) \quad (4.2)$$

dove  $\nabla^{*\top} = [\partial_x^* \partial_y^*]^\top$  indica la derivata all'indietro quando  $\nabla$  è definito come derivata in avanti. Equazione 4.2 è calcolabile in modo efficiente sfruttando la trasformata di Fourier [7]. La risoluzione di questo problema può essere effettuata applicando il LLM al posto dell'equazione di Poisson (Equazione 4.1) e risolvendo rispetto a  $\mathbf{z}$  come descritto nella Sottosezione 2.2.2.

Come altro caso di GDO, viene considerata la situazione in cui  $z_i$  sono noti e si intende ricostruire l'immagine  $\widehat{z}_i$  da una versione modificata dei suoi gradienti  $\nabla \widehat{z}_i = f(\nabla z_i)$ , ossia risolvendo:

$$\arg \min_{\widehat{z}} \sum_i \|\nabla \widehat{z}_i - f(\nabla z_i)\|_2^2 \quad (4.3)$$

Questo problema è risolvibile, come descritto sopra, utilizzando Equazione 2.8 ma richiede prima il calcolo dei gradienti  $\nabla z_i$ . Mentre i metodi convenzionali utilizzano i valori ottenuti

tramite filtri con risposta all’impulso corta o lunga—e.g., differenze finite in avanti o indietro tra due pixel oppure filtri più tolleranti al rumore come filtri di Sobel, di Scharr [23] od altri [22]—, è possibile ottenere i gradienti applicando il LLM come descritto nell’Equazione 2.11.

## 4.2 Applicazione alla Rimozione di Riflessi

Come applicazione possibile del LLM proposto, viene mostrato l’utilizzo del metodo nella risoluzione del problema della *self reflection removal* descritto in [6]. Il problema si presenta quando vengono scattate fotografie attraverso vetri a soggetti illuminati in posti scuri: lo scatto di una fotografia senza flash presenta rumore dovuto all’alta sensibilità ISO necessaria e, in particolare, spesso accade che il fotografo venga riflesso dal vetro; l’utilizzo del flash risolve questi problemi ma satura l’illuminazione originale della scena. Lo scopo è quindi la ricostruzione di un’immagine pulita come quella con il flash ma con i colori naturali dell’immagine senza flash.

Chiamando  $A$  l’immagine senza flash e  $F$  l’immagine con il flash, una coppia di gradienti di luminanza ideali al pixel  $i$  e per il colore  $c \in \{r, g, b\}$  viene denotata come  $\{\nabla \hat{A}_i^c, \nabla \hat{F}_i^c\}$ . In situazioni pratiche, compaiono degli *offset*  $\{\nabla R_{A_i}^c, \nabla R_{F_i}^c\}$  a causa di riflessi e rumore. I gradienti  $\{\nabla A_i^c, \nabla F_i^c\}$  osservati risultano quindi esprimibili dalla seguente relazione:

$$\begin{cases} \nabla A_i^c = \nabla \hat{A}_i^c + \nabla R_{A_i}^c \\ \nabla F_i^c = \nabla \hat{F}_i^c + \nabla R_{F_i}^c \end{cases} \quad (4.4)$$

dove il nostro obiettivo è ottenere  $\hat{A}_i^c$ .

Per la risoluzione del problema vengono effettuate due assunzioni:

1. L’immagine con flash non include artefatti ( $\nabla R_{F_i}^c = 0$ ) ed è ottenuta come riscalatura dell’immagine originale:  $\nabla \hat{F}_i^c = \nabla F_i^c = k_i^c \nabla \hat{A}_i^c$
2. I vettori di *offset* sono ortogonali ai gradienti ideali, ossia  $\nabla \hat{A}_i^c \nabla R_i^c = 0$ .

Sulla base di queste assunzioni, pre-moltiplicando Equazione 4.4 per  $\nabla F_i^{c\top}$ , si ottiene:

$$\begin{cases} \nabla F_i^{c\top} \nabla A_i^c = \nabla F_i^{c\top} \nabla \hat{A}_i^c + \nabla F_i^{c\top} \nabla R_{A_i}^c \\ \|\nabla F_i^c\|_2^2 = k_i^c \nabla F_i^{c\top} \nabla \hat{A}_i^c \end{cases} \quad (4.5)$$

Si può notare che, in quanto  $\nabla \hat{F}_i^{c\top} \nabla R_{A_i}^c = k_i^c \nabla \hat{A}_i^{c\top} \nabla R_{A_i}^c = 0$ , la prima equazione può essere semplificata come  $\nabla F_i^{c\top} \nabla A_i^c = \nabla F_i^{c\top} \nabla \hat{A}_i^c$ . Sostituendo questo risultato nella seconda

equazione si ottiene  $k_i^c = \frac{\|\nabla F_i^c\|_2^2}{\nabla F_i^c \top \nabla A_i^c}$ . Sostituendo nuovamente il valore di  $k_i^c$  trovato all'interno della definizione data nella prima assunzione ( $\nabla \hat{A}_i^c = \frac{1}{k_i^c} \nabla F_i^c$ ), si ottiene infine la seguente relazione tra le due immagini:

$$\nabla \hat{A}_i^c = \frac{\nabla F_i^c \top \nabla A_i^c}{\|\nabla F_i^c\|_2^2} \nabla F_i^c \quad (4.6)$$

Il problema è quindi risolvibile usando Equazione 4.3 ponendo  $\hat{z}_i = \hat{A}_i^c$  e  $f(\nabla z_i) = \frac{\nabla F_i^c \top \nabla A_i^c}{\|\nabla F_i^c\|_2^2} \nabla F_i^c$ .

Nell'applicazione del LLM per ottenere  $\nabla \hat{A}_i^c$ , prima vengono calcolati i vettori normali delle immagini  $A$  ed  $F$  usando Equazione 2.11, successivamente vengono calcolati i gradienti  $\nabla \hat{A}_i^c$  usando Equazione 4.6 ed infine viene ricostruita l'immagine originale risolvendo Equazione 2.8 con il CG.

### 4.3 Applicazione alla Rimozione di Ombre

Una ulteriore possibile applicazione del metodo è la *rimozione di ombre*, descritto in Finlayson et al. [7]; Shor and Lischinski [18]. Il problema si può presentare in applicazioni *artistiche*, ossia quando si intende rimuovere ombre da fotografie per motivi estetici, come ombre proiettate dal fotografo o causate dall'uso del flash, oppure in applicazioni di *computer vision*, come pre-elaborazione utile a semplificare ed incrementare la probabilità di successo di compiti come la segmentazione di immagini, il *tracking*, o il riconoscimento di oggetti.

Per affrontare questo problema sono state analizzate le due differenti metodologie descritte da Finlayson et al. [42] e da [18].

#### 4.3.1 Rimozione di Ombre basata su immagini invarianti all'illuminazione

Il metodo proposto da Finlayson [7; 42] si incentra sul costruire una rappresentazione dell'immagine invariante rispetto al colore ed all'illuminazione, ed utilizza questo risultato per rimuovere l'ombra dall'immagine originale, considerandola come una semplice diminuzione dei gradienti lungo il suo bordo. Per poter agire in questo modo vengono fatte alcune assunzioni sul processo di generazione dell'ombra.

1. La prima assunzione è che l'immagine sia formata secondo un modello Lambertiano, ossia che i valori di ogni pixel siano linearmente legati all'intensità della luce incidente e che non vi siano effetti come specularità o inter-riflessioni nell'immagine. Sotto questa assunzione,

la risposta dei sensori della fotocamera è esprimibile come:

$$\rho_k(x, y) = \sigma(x, y) \int E(\lambda, x, y) S(\lambda, x, y) Q_k(\lambda) d\lambda \quad (4.7)$$

dove  $E(\lambda, x, y)$  denota la SPD della luce,  $S(\lambda, x, y)$  la riflettanza della superficie,  $Q(\lambda)$  la sensibilità spettrale del  $k$ -esimo sensore ( $k = 1, 2, 3$ ) e  $\sigma(x, y)$  è un fattore costante che denota il termine di ombreggiatura Lambertiano per un dato pixel.

2. La seconda assunzione, che verrà successivamente rilassata, è che i sensori della fotocamera che ha scattato la fotografia siano perfettamente a banda stretta, vale a dire con risposta ad una singola lunghezza d'onda della luce:

$$Q_k(\lambda) = q_k \delta(\lambda - \lambda_k) \quad (4.8)$$

3. La terza ed ultima assunzione è che gli illuminanti della scena siano Planckiani. Modellando l'illuminazione tramite l'approssimazione di Wien della legge di Planck [43], questa assunzione significa che la SPD dell'illuminante può essere espressa in funzione della temperatura  $T$ :

$$E(\lambda, T) = I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}} \quad (4.9)$$

dove  $c_1$  e  $c_2$  sono costanti ed  $I$  è una variabile che controlla l'intensità della luce.

Basandosi su queste assunzioni, si può esprimere la risposta dei sensori in modo approssimato sostituendo Equazione 4.8 in Equazione 4.7 e Equazione 4.9 nel risultato ottenuto:

$$\rho_k = \sigma E(\lambda_k) S(\lambda_k) q_k = \sigma I c_1 \lambda_k^{-5} e^{-\frac{c_2}{T\lambda_k}} S(\lambda_k) q_k \quad (4.10)$$

dove la dipendenza di  $\rho_k$  dalla posizione nello spazio viene temporaneamente ignorata.

Formando il vettore di cromaticità  $\chi$  come rapporto tra bande, si può notare che vengono rimosse le informazioni riguardanti l'intensità e l'ombreggiatura:

$$\chi_j = \frac{\rho_k}{\rho_p} = \frac{\lambda_k^{-5} e^{\frac{c_2}{T\lambda_p}} S(\lambda_p) q_p}{\lambda_p^{-5} e^{\frac{c_2}{T\lambda_p}} S(\lambda_p) q_p}, \quad k \in \{1, 2, 3\}, \quad k \neq p, \quad j = 1, 2 \quad (4.11)$$

Calcolando il logaritmo  $\chi'$  di queste quantità, si ottiene:

$$\chi'_j = \log(\chi_j) = \log\left(\frac{s_k}{s_p}\right) + \frac{1}{T}(e_k - e_p) \quad (4.12)$$

dove  $s_k = \lambda_k^{-5} S(\lambda_k) q_k$  e  $e_k = -\frac{c_2}{\lambda_k}$ . Rappresentando le log-cromaticità sotto forma vettoriale:

$$\chi' = \mathbf{s} + \frac{1}{T} \mathbf{e} \quad (4.13)$$

si può notare che sono composte da due termini:  $\mathbf{s}$ , dipendente dalla superficie e dalla camera ma indipendente dall'illuminante e  $\mathbf{e}$ , indipendente dalla superficie, ma ancora dipendente dalla fotocamera.

Osservando questa rappresentazione, si può osservare che, al variare della temperatura dell'illuminazione, il vettore delle log-cromaticità  $\chi'$  si muove lungo una linea dritta, con caratteristiche dipendenti solamente dalla fotocamera. Ciò significa che, riuscendo a determinare la direzione della variazione (il vettore  $\mathbf{e}$ ), si può determinare una rappresentazione monodimensionale indipendente dall'illuminazione proiettando le log-cromaticità sul vettore ortogonale  $\mathbf{e}^\perp$ :

$$\mathcal{I}' = \chi'^\top \mathbf{e}^\perp, \quad \mathcal{I} = \exp(\mathcal{I}') \quad (4.14)$$

Per rimuovere qualsiasi polarizzazione dovuta alla scelta del canale di colore da utilizzare come denominatore nella definizione delle cromaticità, si può utilizzare la media geometrica  $\rho_M = \sqrt[3]{R \times G \times B}$  in Equazione 4.11 al posto di  $\rho_p$ , mantenendo comunque la dipendenza lineare. Così facendo si ottiene un vettore tridimensionale ed i rapporti giacciono su un piano ortogonale a  $\mathbf{u} = [1, 1, 1]^\top$ .

Questa derivazione, descritta in [7], si basa su una completa conoscenza<sup>1</sup> dei sensori della fotocamera, informazione spesso non disponibile. Nel caso generico, l'immagine indipendente dall'illuminazione è comunque ottenibile, come descritto in [17; 42], analizzando l'entropia di varie proiezioni delle log-cromaticità sui possibili  $\mathbf{e}^\perp$ . L'idea fondamentale è che, anche senza aver disponibili più fotografie per la calibrazione dell'algoritmo, una proiezione delle log-cromaticità nella corretta direzione causa la minimizzazione dell'entropia dell'immagine monodimensionale risultante. Il metodo proposto è quindi, nella versione più semplice, di attraversare tutti i possibili orientamenti tra  $0^\circ$  e  $180^\circ$  e proiettare  $\chi$  in quella direzione: per ogni angolazione si può applicare la definizione di entropia di Shannon dopo aver calcolato l'istogramma dell'immagine. Dividendo l'istogramma per il numero di elementi in modo da ottenere delle probabilità  $p_i$ , l'entropia è calcolabile come:

$$\eta = \sum_{i=1}^N -p_i \log_2(p_i) \quad (4.15)$$

La rappresentazione invariante all'illuminazione descritta riesce a rimuovere le ombre, ma ciò facendo viene rimossa anche l'informazione sul colore dell'immagine. Osservando che il processo

---

<sup>1</sup>Finlayson et al. [7] espone una metodologia per la stima di queste informazioni, ma ciò richiede di avere un set di immagini di superfici di riferimento (come una Macbeth Color Checker Chart, composta da 19 superfici di differenti cromaticità) sotto una serie di  $n$  luci differenti.

per ricavare questa rappresentazione consiste nel proiettare i vari punti delle log-cromaticità lungo la retta di direzione  $\mathbf{e}^\perp$ , si può provare a rappresentare il punto in cui ogni pixel viene proiettato tramite le sue coordinate sul piano, mantenendo così qualche informazione sul colore. Per fare ciò viene costruita un'invariante all'illuminazione bidimensionale sfruttando la matrice di proiezione  $P_{\mathbf{e}^\perp}$ , definita come:

$$P_{\mathbf{e}^\perp} = \mathbf{e}^\perp \mathbf{e}^{\perp \top} \quad (4.16)$$

ed applicandola come segue:

$$\tilde{\chi}' = P_{\mathbf{e}^\perp} \chi' \quad (4.17)$$

Questo miglioramento, pur essendo molto utile, risulta comunque in un'immagine dai colori innaturali e mancante dell'informazione sull'intensità. In molte applicazioni è preferibile avere un'immagine identica all'originale, ad eccezione delle ombre, ma per ottenere ciò bisogna adottare strategie differenti. L'algoritmo proposto in [7; 42] prende spunto dai metodi di *lightness recovery*<sup>2</sup>: questi algoritmi ricevono in ingresso un'immagine a colori tridimensionale e restituiscono due immagini intrinsiche: una basata sulla riflettanza (l'immagine *lightness*) e l'altra sull'illuminazione. Il calcolo della *lightness* si basa sull'assunzione che l'illuminazione vari lentamente lungo un'immagine dove i cambiamenti nella riflettanza sono lenti. Da questo segue che ponendo una soglia sulla derivata dell'immagine, è possibile rimuovere cambiamenti lenti ed il risultato è ottenibile dall'integrazione di questa immagine.

Dato che le ombre consistono in cambiamenti tendenzialmente rapidi dell'illuminazione, queste non sono rimosse nell'immagine *lightness* ma, applicando questa idea alle due immagini disponibili nel problema della rimozione delle ombre—l'immagine 3-d originale e l'immagine 2-d priva di ombre—è possibile distinguere i bordi di oggetti materiali dai bordi di ombre ed eliminarli di conseguenza.

---

<sup>2</sup>La percezione del colore mostra un'alta correlazione con la riflettanza, anche se la quantità di luce visibile che raggiunge l'occhio dipende dal prodotto di riflettanza ed illuminazione. Il sistema visivo deve essere in grado di raggiungere questo risultato sfruttando uno schema che non misura il flusso e questo modello è descritto come base della teoria “retinex” da Land and McCann [44] (il termine “retinex” è un portmanteau composto da “retina” e “cortex”, per suggerire che in questo processo sono coinvolti sia occhi che cervello). Questa teoria assume che ci siano tre sistemi di coni indipendenti, ognuno con un picco di recettori per bande differenti (lunghe, medie e corte fasce di lunghezze d'onda dello spettro visibile). Ogni sistema forma un'immagine separata del mondo in termini di luminosità che mostra una forte correlazione con la riflettanza nella sua specifica banda di lunghezze d'onda. Queste immagini non vengono mescolate per ottenere il risultato finale, bensì confrontate per generare i colori percepiti. Vari metodi sono stati proposti per ottenere questi risultati a livello numerico (e.g., [44–46]).

Il metodo proposto da Finlayson et al. [7] consiste quindi nell’identificazione e nel confronto dei bordi in ognuna delle due immagini: quelli scomparsi nell’immagine sono trattati come bordi delle ombre e la derivata dell’immagine originale lungo di essi viene posta a zero. Un metodo iterativo viene proposto per effettuare un’integrazione dell’immagine che includa anche un semplice processo di *inpainting*<sup>3</sup> dei bordi rimossi.

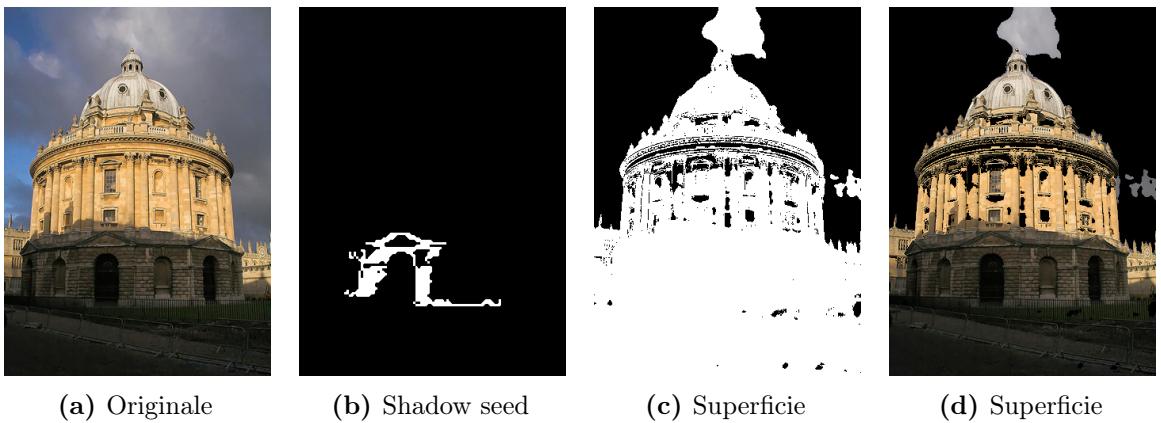
La strategia proposta prevede un’alternanza tra integrazione e smussatura dei gradienti del bordo, in modo da ammorbidire il passaggio dalla regione in luce alla regione in ombra ristorata. Il LLM è applicabile in questa fase del metodo, sostituendolo al calcolo dei gradienti ed alla loro integrazione in ogni passaggio dell’algoritmo, rispettivamente nelle forme di Equazione 2.11 ed Equazione 2.8.

Purtroppo questo metodo di rilevamento delle ombre è molto sensibile ai parametri utilizzati ed alla qualità dell’immagine in ingresso. L’immagine indipendente dall’illuminazione presenta inoltre bordi leggermente alterati rispetto all’immagine originale, rendendo difficile la corrispondenza tra i contorni delle due immagini e, quindi, la corretta identificazione dei contorni delle ombre. Ulteriori problemi sono riscontrabili quando le ombre da eliminare non consistono in variazioni nette dell’illuminazione, bensì sfumate, ampie anche pochi pixel: risulta necessario definire ad-hoc la larghezza dell’area in cui la derivata viene posta a zero (pena l’incompleta rimozione dell’ombra) e l’utilizzo di ampiezze troppo elevate comporta difficoltà nella ricostruzione con qualità apprezzabile, causando la perdita della texture dell’immagine lungo il bordo dell’ombra.

Per questi motivi non è stato possibile applicare il LLM al problema della rimozione delle ombre seguendo questo metodo e si è scelto di affrontare il problema seguendo l’idea proposta da Shor and Lischinski [18], descritta nella sezione seguente.

#### 4.3.2 Rimozione di Ombre basata su piramidi Laplaciane

Il metodo proposto da Shor and Lischinski [18] si basa sull’identificazione dell’ombra e della sua superficie, ed il restauro della prima tramite la stima di un modello di formazione delle ombre.



**Figura 4.1:** Determinazione della superficie dell’ombra da rimuovere

Da un punto indicato nell’immagine originale (a) viene determinato il *shadow seed* (b). Dalla soglia determinata in Figura 4.2 si deriva la maschera (c), corrispondente nell’immagine a (d).

### Identificazione della regione dell’ombra

Per l’identificazione dell’ombra è richiesto un minimo input da parte dell’utente, il quale deve specificare un punto dell’ombra. L’algoritmo, innanzitutto, espande la regione tramite region growing—il risultato di questo processo è una patch che viene denominata *shadow seed* (Figura 4.1b)—ed identifica nell’immagine la superficie su cui giace la parte dell’ombra selezionata sfruttando un’idea simile all’invarianza all’illuminazione proposta da Finlayson et al. [47]: utilizzando il colore mediano della regione selezionata come riferimento, viene calcolata la sua distanza da ogni pixel definita come:

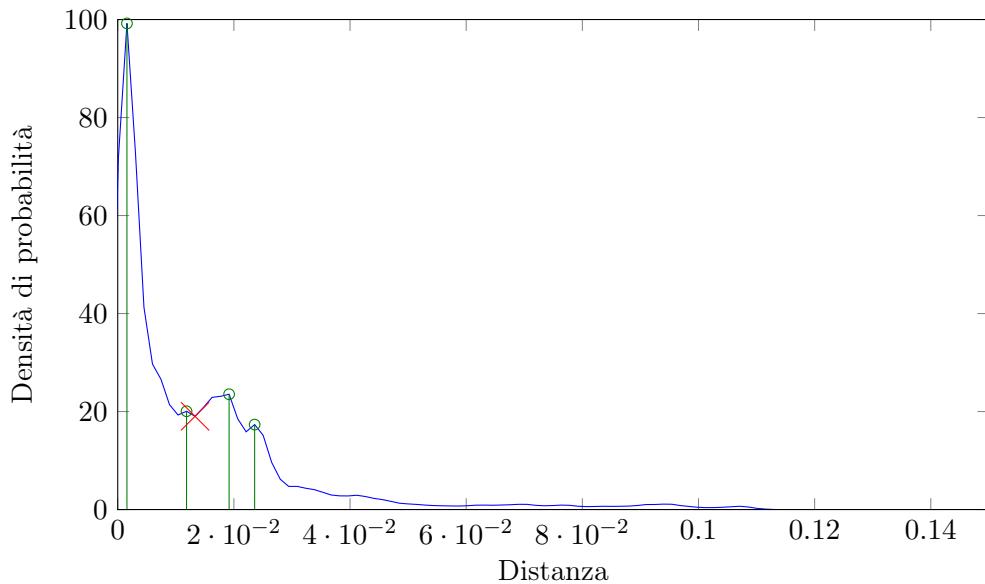
$$1 - |\cos(\theta)| \quad (4.18)$$

dove  $\theta$  è l’angolo tra il vettore tridimensionale rappresentante il colore del pixel in analisi e quello corrispondente al colore di riferimento. L’intuizione dietro a questa metrica è che il colore di punti con riflettanza simile corrisponda a vettori circa collineari e che la presenza dell’ombra comporti principalmente una riduzione della loro norma. Questa osservazione è stata sfruttata anche da Omer and Werman [33] riguardo alle *color-lines*.

Osservando l’istogramma delle distanze (e.g., Figura 4.2), si può notare la presenza, solitamente, di un picco a distanza prossima a zero, corrispondente alla superficie su cui giace l’ombra. Impostando una soglia sulla distanza al primo minimo dell’istogramma dopo questo picco, è possibile quindi classificare i pixel appartenenti alla stessa superficie dell’ombra (Figura 4.1c,

---

<sup>3</sup>L’**inpainting** è un processo tramite il quale viene ricostruita dell’informazione mancante da un’immagine (ad esempio, dovuta a testo sovrascritto all’immagine, usura di vecchie immagini scansionate o, nel caso qui trattato, rimossa volontariamente) osservando i pixel circostanti l’area di applicazione dell’algoritmo.



**Figura 4.2:** Istogramma delle distanze

In blu, la *finestra di Parzen* [48] relativa alle distanze, in verde sono indicati i principali massimi locali dell'istogramma, e in rosso il valore scelto come soglia.  
Le distanze sono calcolate utilizzando Equazione 4.18.

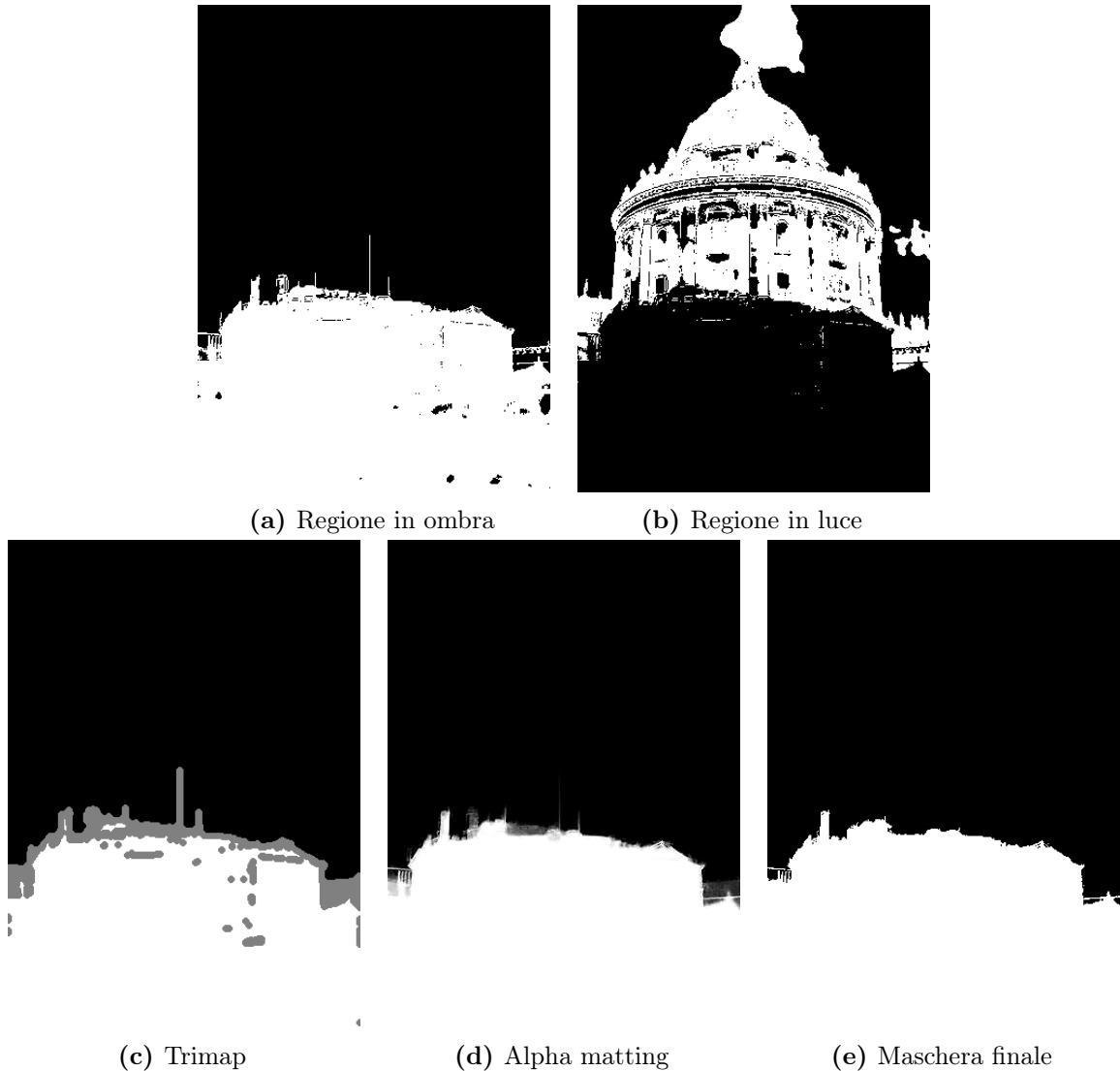
Figura 4.1d). Nella pratica, capita spesso che compaiano due picchi: uno a distanza prossima a zero, rappresentante i pixel corrispondenti alla parte di superficie su cui giace l'ombra ed uno a distanza leggermente maggiore, corrispondente ai pixel appartenenti alla parte di superficie illuminata. Per questo motivo si è deciso di utilizzare come valore di soglia, il primo minimo della derivata prima dopo il secondo massimo dell'istogramma (Figura 4.2).

Il passo successivo consiste nella divisione della superficie tra parte effettivamente ombreggiata e parte illuminata (Figura 4.3); ciò viene fatto nuovamente con un processo di *region growing*, questa volta utilizzando solo i punti ottenuti dal passo precedente e, ad ogni iterazione, estendendo la patch corrispondente al *shadow seed*. Queste iterazioni di *region growing* utilizzano una regola di terminazione aggiuntiva rispetto alle versioni classiche, ossia viene richiesto che venga preservata la condizione che la varianza della parte in ombra resti inferiore alla varianza della parte illuminata:

$$\text{Var}[I(M_s)] \leq \text{Var}[I(M_l)] \quad (4.19)$$

dove  $M_s$  ed  $M_l$  sono rispettivamente la maschera dei pixel della superficie ombreggiata e della superficie illuminata.

La maschera corrispondente all'ombra da eliminare viene poi completata includendo tutte le regioni dell'immagine non appartenenti a  $M_l$  circondate da pixel identificati come appartenenti a  $M_s$  ed infine raffinata tramite l'algoritmo di *image matting* descritto in [1]: viene costruita una



**Figura 4.3:** Determinazione della maschera dell'ombra da rimuovere

La superficie di Figura 4.1c viene divisa in regione in ombra (a) e regione in luce (b). La maschera viene raffinata applicando l'image matting [1] alla trimap (c), ottenendo (d). La maschera finale (e) viene ottenuta applicando una soglia  $\alpha = 0.5$  a (d).

*trimap* (Figura 4.3c) indicante le regioni certamente appartenenti all'ombra (nell'implementazione, questa regione corrisponde alla versione di  $M_s$  completata e leggermente erosa), certamente non appartenenti all'ombra (corrispondente al complemento di una leggera dilatazione della versione di  $M_s$  completata) e le regioni ignote (tutti i punti restanti). Al risultato dell'*image matting* (Figura 4.3d) viene poi applicata una soglia  $\alpha = 0.5$  per ottenere la maschera binaria finale, denominata  $M_{shadow}$  (Figura 4.3e).

### Ristorazione dell'ombra

Il modello di formazione dell'immagine utilizzato in questo metodo è quello descritto da Barrow and Tenenbaum [49]:

$$I(x, \lambda) = L(x, \lambda)R(x, \lambda) \quad (4.20)$$

dove  $I(x, \lambda)$  è l'intensità riflessa da un punto  $x$  della scena ad una lunghezza d'onda  $\lambda$ , mentre  $L$  ed  $R$  sono rispettivamente l'illuminazione e la riflettanza per lo stesso punto e lunghezza d'onda. Questo modello assume uno scenario dove le ombre proiettate siano dovute ad una singola sorgente di illuminazione (come il sole in scene all'aperto). Se un punto  $x$  nella scena è privo di ombra, la sua illuminazione è esprimibile come somma dell'illuminazione diretta  $L^d$  e dell'illuminazione indiretta  $L^a$ , dovuta all'ambiente:

$$L(x, \lambda) = L^d(x, \lambda) + L^a(x, \lambda) \quad (4.21)$$

$$I^{lit}(x, \lambda) = L^d(x, \lambda)R(x, \lambda) + L^a(x, \lambda)R(x, \lambda) \quad (4.22)$$

Supponendo che l'ombra sia causata dalla presenza di un oggetto nella scena che occlude la sorgente primaria di luce, proiettando così un'ombra sul punto  $x$ , l'intensità riflessa da questo punto risulta quindi:

$$I^{shadow}(x, \lambda) = a(x)L^a(x, \lambda)R(x, \lambda) \quad (4.23)$$

dove  $a(x)$  è un fattore spazio-variante che tiene conto dell'attenuazione dell'illuminazione dell'ambiente, ulteriore effetto della presenza dell'occlusore. Qui viene assunto che l'illuminazione ambientale abbia pressoché la stessa Distribuzione Spettrale di Potenza (SPD) per qualsiasi direzione incidente, altrimenti il termine  $a$  dovrebbe dipendere anche dalla lunghezza d'onda  $\lambda$ .

Combinando le ultime due equazioni possiamo esprimere l'intensità su un punto illuminato come funzione affine dell'intensità dello stesso punto se fosse presente un'ombra:

$$I^{lit}(x, \lambda) = L^d(x, \lambda)R(x, \lambda) + \frac{1}{a(x)}I^{shadow}(x, \lambda) \quad (4.24)$$

Quando la scena viene fotografata, il colore del pixel  $p$  corrispondente al punto  $x$  della la scena viene ottenuto integrando  $I(x, \lambda)$  con la risposta spettrale del sensore della fotocamera. Essendo questa operazione lineare, la natura affine della relazione tra pixel con e senza ombra non cambia, risultando quindi in:

$$I_k^{lit}(p) = \alpha_k(p) + \gamma(p) I_k^{shadow}(p) \quad (4.25)$$

dove  $\alpha_k$ ,  $k \in \{R, G, B\}$ , è la risposta della fotocamera all'illuminazione diretta riflessa lungo i tre canali *RGB*, e  $\gamma(p) = \frac{1}{a(x)}$  è l'inverso del fattore di attenuazione dell'ambiente che, sotto le assunzioni fatte, non dipende dalla lunghezza d'onda ed è quindi costante attraverso i tre canali.

Equazione 4.25 può essere sfruttata per ricostruire un pixel illuminato dal suo colore sottoposto ad ombra stimando quattro parametri. Questo viene fatto analizzando due regioni dell'immagine, una illuminata ed una ombreggiata, e sfruttando le medie e le deviazioni standard delle luminanze dei pixel corrispondenti—o con colore appartenente con sufficiente frequenza ai colori della rispettiva maschera—alla superficie dell'ombra:

$$\gamma = \frac{\sigma(\mathcal{L})}{\sigma(\mathcal{S})} \quad (4.26)$$

$$\alpha_k = \bar{\mathcal{L}} - \gamma \bar{\mathcal{S}}, \quad k \in \{R, G, B\} \quad (4.27)$$

dove  $\mathcal{L}$  ed  $\mathcal{S}$  sono le versioni estese come scritto sopra delle maschere  $M_l$  ed  $M_s$ . Questo metodo di restauro ricalca il metodo di trasferimento del colore proposto da Reinhard et al. [50] ed utilizzato anche da Wu et al. [51]. Figura 4.4a mostra il risultato dell'utilizzo di questo metodo per il recupero dell'ombra di Figura 4.1.

**Ristorazione basata su piramidi** Semplicemente stimando il modello una volta sola per ogni pixel dell'ombra (Figura 4.4a) può causare la comparsa di due artefatti:

1. La texture della regione recuperata risulta avere un contrasto minore della regione originalmente illuminata.

Una causa di questo problema è che l'illuminazione diretta direzionale nell'area senza ombra comporta un contrasto maggiore dell'illuminazione indiretta emisferica nell'area con ombra.

Inoltre, né questo modello né il precedente tengono conto che regioni con texture articolate presentano spesso un fenomeno denominato *self-shadowing*<sup>4</sup> anche nelle regioni illuminate.

---

<sup>4</sup>Con **self-shadowing** si intende la proiezione di ombre da parte di oggetti su sé stessi. Un esempio di questo fenomeno su scala fine può essere l'ombra generata da ogni filo d'erba su un prato, indispensabile per dare realismo all'immagine e tendenzialmente inutile ed impratica da rimuovere per applicazioni di computer vision.



(a) Ristorazione “*naïve*”      (b) Ristorazione piramidale      (c) Ristorazione non-uniforme

**Figura 4.4:** Ristorazione dell’ombra

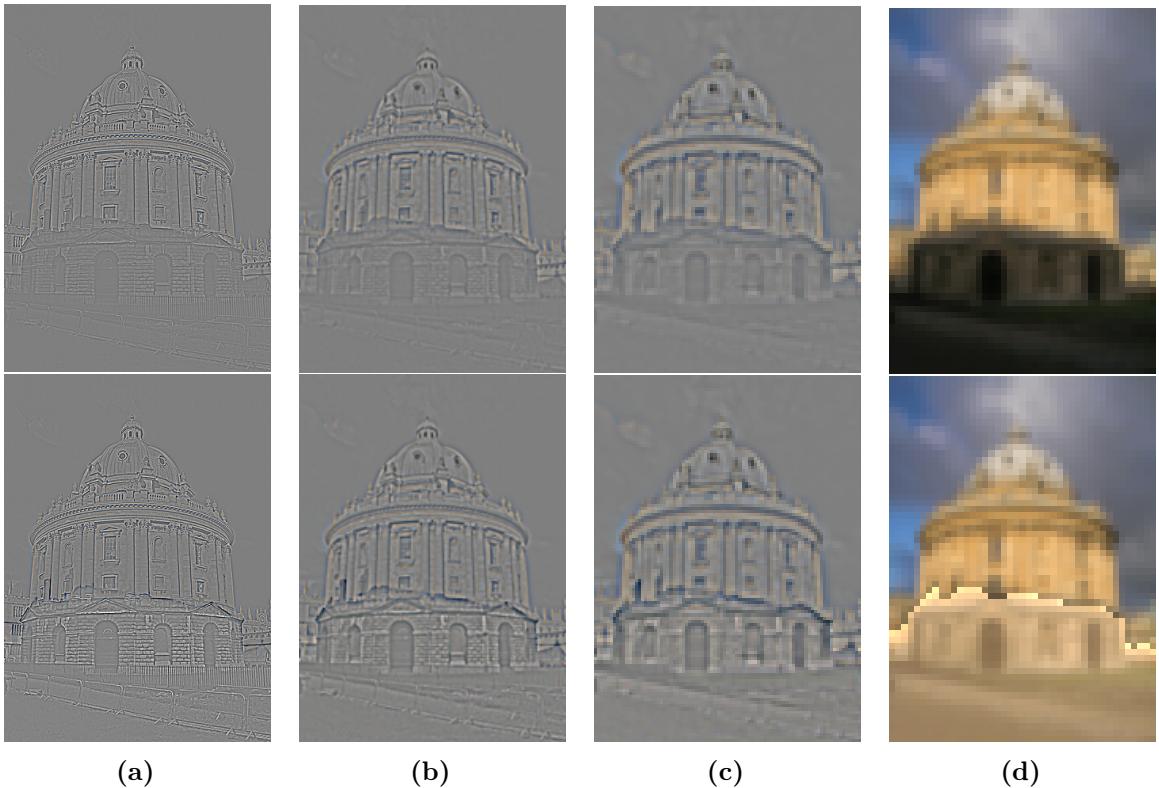
L’immagine originale (Figura 4.1a) viene restaurata applicando semplicemente il modello di Equazione 4.25, ottenendo (a). Effettuando il restauro su ogni livello della piramide Laplaciana (Figura 4.5), si ottiene (b) e, raffinando ulteriormente il recupero lavorando su strisce concentriche nel livello più grossolano, si ottiene (c).

2. Il secondo artefatto è dovuto al rumore: la regione ombreggiata è più prona al rumore delle aree circostanti e una semplice riscalatura dei valori tende ad esaltare questo problema. Questo effetto è particolarmente evidente quando vengono restaurate ombre proiettate su superfici scure o lisce.

La strategia adottata per affrontare questi problemi è di effettuare la stima dei parametri ed il restauro dell’ombra utilizzando una piramide Laplaciana (Figura 4.5) come rappresentazione dell’immagine: vengono calcolate la rappresentazione piramidale dell’immagine e le versioni sotto-campionate delle maschere, e, per ogni livello, viene effettuato il restauro come descritto sopra. Il risultato ottenuto ricomponendo la piramide restaurata è mostrato in Figura 4.4b.

Questo procedimento permette di ridurre entrambi gli artefatti descritti sopra: da un lato, nelle regioni riguardanti le texture, il contrasto della trama viene aumentato lavorando sulle sue statistiche per renderle come quelle delle texture illuminate. Dall’altro lato, nelle regioni prive di texture, l’abbinamento delle statistiche con quelle delle aree illuminate più lisce tende a ridurre l’ampiezza del rumore.

**Trattamento di ombre non uniformi** Un altro problema che è possibile affrontare con questo metodo, contrariamente al precedente, è la gestione di ombre non uniformi. Tipicamente, le ombre



**Figura 4.5:** Ristorazione tramite piramide Laplaciana

Nella riga superiore viene mostrato ogni livello (dal più fine (a) al più grossolano (d)) della scomposizione Laplaciana dell'immagine originale.

Nella riga inferiore viene mostrato il corrispondente livello restaurato secondo Equazione 4.25.

proiettate tendono a scurire avvicinandosi all'oggetto occludente: un semplice annullamento della derivata lungo il bordo lascerebbe questo effetto, risultando in un recupero incompleto dell'ombra. La strategia adottata consiste nell'effettuare la stima ed il restauro su sottili strisce concentriche dell'ombra singolarmente (Figura 4.4c), invece che sull'intera maschera in un passaggio solo. Questo processo, visto l'aumento di iterazioni che comporta e, soprattutto, la sua utilità incentrata su variazioni graduali dell'ombra, viene applicato solamente al livello della piramide corrispondente alle basse frequenze spaziali.

Un miglioramento possibile di questa idea, benché un'ulteriore complicazione, potrebbe essere la suddivisione delle strisce stesse in sotto-segmenti. Questo permetterebbe di gestire meglio le variazioni dell'ombra e, soprattutto, terrebbe meglio conto dello scurirsi dell'ombra in direzione dell'oggetto occludente invece che verso il centro della stessa.

**Gestione del bordo dell'ombra** Il bordo dell'ombra richiede un trattamento speciale, questo è dovuto principalmente a due cause:

1. La maschera  $M_{shadow}$  è tipicamente imprecisa lungo il bordo dell'ombra. Questo è evidente se si pensa che il bordo di un'ombra può essere sfocato lungo qualche pixel, mentre la maschera rappresenta una distinzione netta tra regioni con e senza ombra.
2. I bordi delle ombre corrispondono tipicamente a rapidi cambiamenti di luminosità e, benché la gestione tramite strisce riduca ampiamente l'area da trattare, per un recupero efficace sarebbe necessaria una stima a livello di singoli pixel, evidente controsenso con la natura statistica del metodo.

Shor and Lischinski [18] propongono quindi un trattamento della regione tramite inpainting basato sulla sintesi di texture descritto da Kwatra et al. [52]. Un'alternativa può essere il restauro della texture tramite algoritmi di *Poisson image editing* [53; 54]. L'algoritmo viene utilizzato come strumento di *seamless cloning*—in particolare per il *texture transfer*—come descritto da Pérez et al. [54]: la texture dell'immagine originale lungo il bordo dell'ombra viene imposta sull'immagine ristorata. Questa operazione permette di mantenere un'ottima qualità nei dettagli originali dell'immagine ed al contempo di restaurare implicitamente—anche se solo in modo parziale—l'ombra lungo il bordo.

Adottando le convenzioni descritte all'inizio di Capitolo 2, e chiamando  $\mathbf{p}_i$  l' $i$ -esimo pixel dell'immagine originale,  $\mathbf{I}_i^*$  l' $i$ -esimo pixel dell'immagine da cui è stata rimossa l'ombra e  $M_b$  la maschera rappresentante il bordo (ottenuta tramite dilatazione degli estremi di  $M_{shadow}$ ), l'equazione di Poisson per ottenere l'immagine finale  $\mathbf{I}$  risulta essere:

$$\arg \min_{\mathbf{I}|_{M_b}} \sum_i \sum_{j \in \Omega_i} (\mathbf{I}_i - \mathbf{I}_j - (\mathbf{p}_i - \mathbf{p}_j))^2 \quad \text{s. t. } \mathbf{I}_i = \mathbf{I}_i^*, \quad \forall i \notin \Omega \quad (4.28)$$

Per i pixel esterni alla maschera, l'equazione consiste in una semplice identità (i.e., le intensità dei pixel di  $\mathbf{I}^*$  vengono mantenute inalterate in  $\mathbf{I}$ ). Per quanto riguarda i pixel  $i$  interni alla maschera, invece, il loro valore è ottenibile risolvendo:

$$|\Omega_i| \mathbf{I}_i - \sum_{j \in \Omega_i} \mathbf{I}_j = |\Omega_i| \mathbf{p}_i - \sum_{j \in \Omega_i} \mathbf{p}_j \quad (4.29)$$

Ciò risulta in un sistema sparso e definito positivo, risolvibile quindi direttamente. Quando  $\Omega_i$  corrisponde ai pixel *4-adiacenti* ad  $i$ , Equazione 4.29 equivale a richiedere che l'uguaglianza tra i gradienti calcolati tramite l'operatore di Laplace dell'immagine in uscita coincidano con quelli dell'immagine  $\mathbf{p}$ . Si noti che  $i \in M_b \Rightarrow j \in M_b$ , ossia che alcuni valori a sinistra dell'uguale in Equazione 4.29 sono soggetti al vincolo di Equazione 4.28. Questo fa sì che venga comunque preservata la continuità dei gradienti finali di  $\mathbf{I}$  lungo i bordi di  $M_s$ .

La risoluzione dell'equazione di Poisson (Equazione 4.28) tramite il LLM viene effettuata calcolando prima i gradienti di  $\mathbf{p}$  secondo Equazione 2.11 e poi ricostruendo l'immagine finale  $\mathbf{I}$  secondo Equazione 2.8. Per tener conto del vincolo aggiunto, è necessario apportare una modifica al metodo risolutivo basato sul Gradiente Coniugato. Questa consiste nell'utilizzo dell'immagine  $\mathbf{I}^*$  come stima iniziale della soluzione e nell'annullamento ad ogni iterazione dei residui per i pixel da mantenere fissi. Considerando queste condizioni al contorno, la soluzione risulta essere unica e ciò rende non necessaria la definizione del vettore di *offset*  $\mathbf{v}$  di Equazione 2.9.

# Capitolo 5

## Risultati sperimentali

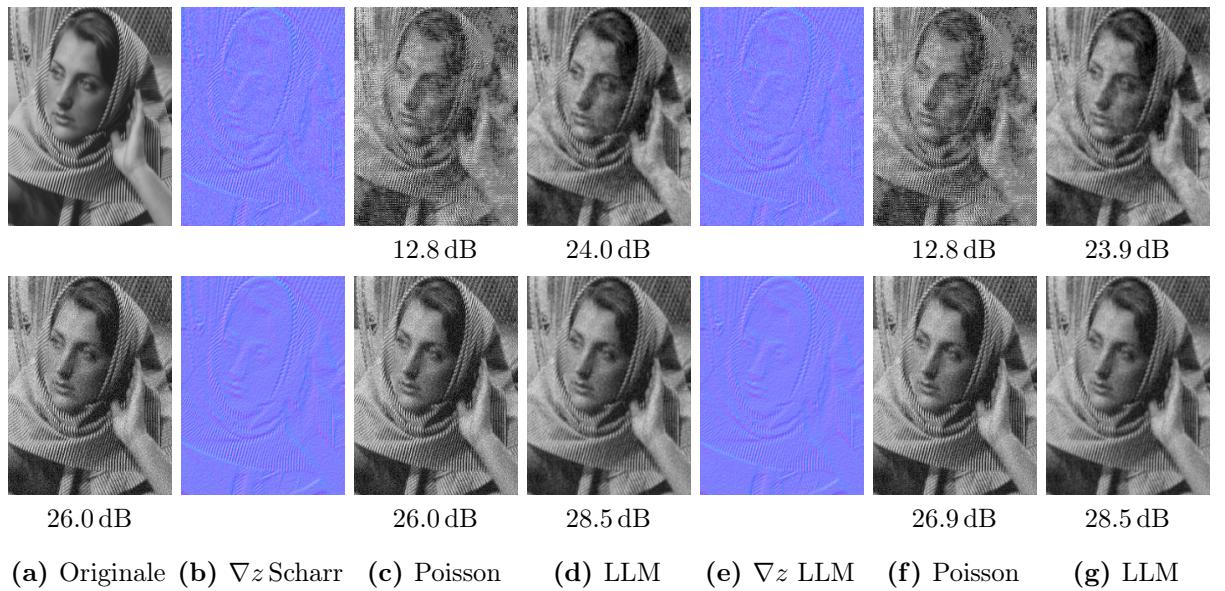
### 5.1 Modello Locale Lineare

In questa sezione, verranno descritti alcuni risultati di applicazioni del LLM e verranno confrontati i risultati ottenibili convenzionalmente tramite la risoluzione delle corrispondenti equazioni di Poisson.

Le applicazioni trattate sono quelle descritte in Capitolo 4, ossia stereo fotometrico (Sezione 4.1), rimozione di riflessi (Sezione 4.2), e rimozione di ombre (Sezione 4.3.2); mentre i metodi utilizzati come confronto sono la differenza finita in avanti e l'operatore di Scharr [23]  $3 \times 3$ .

#### 5.1.1 Inter-conversione di Luminanza e Gradienti

Figura 5.1 mostra i risultati della inter-conversione di un'immagine ed una mappa di gradienti. Si può notare che questo problema è equivalente allo **stereo fotometrico** considerando la luminosità come profondità ed i gradienti come vettori normali. Per esaminare la tolleranza al rumore, nella riga superiore, è stato aggiunto un rumore Gaussiano  $\sim \mathcal{N}(0, 0.05)$  alla mappa dei gradienti dopo la conversione e, nella riga inferiore, un rumore con la stessa distribuzione è stato aggiunto all'immagine originale. Come *offset sv* viene aggiunto il valore medio dell'immagine originale all'immagine recuperata (Equazione 2.9). Per una valutazione numerica, viene mostrato il Peak Signal-to-Noise Ratio (PSNR) tra l'immagine originale ed i risultati. Per quanto riguarda il metodo di Poisson, nella riga superiore è possibile notare la comparsa di artefatti a griglia: sono dovuti alla natura instabile dell'algoritmo attorno alla soluzione, causata dall'avvicinarsi a 0 dei coefficienti nelle alte frequenze della trasformata di Fourier. La riga inferiore mostra che il metodo non riesce a gestire il rumore dell'immagine originale. Per quanto riguarda il metodo LLM, la riga



**Figura 5.1:** Risultati della inter-conversione di luminanza e gradienti

Riga superiore: del rumore viene aggiunto alle mappe di gradienti (b) ed (e).

Riga inferiore: del rumore viene aggiunto all'immagine originale (a).

In (b), (c), ed (f), viene utilizzato un operatore di Scharr  $3 \times 3$  come filtro differenziale.

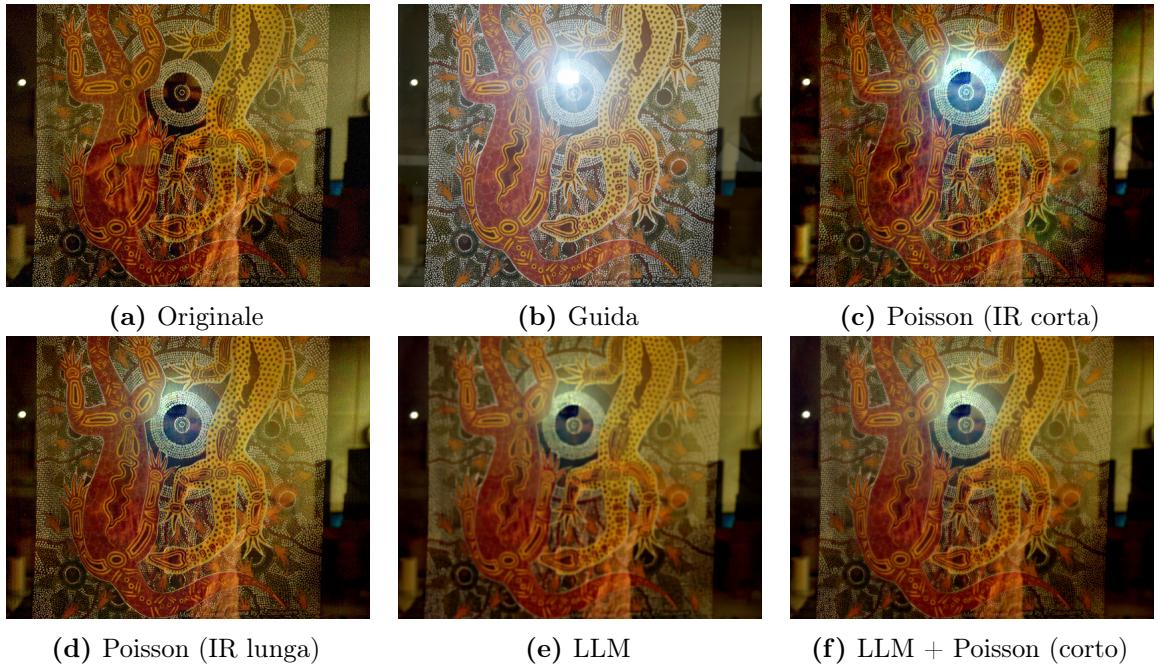
In (d), (e), e (g), viene utilizzato il LLM con patch  $3 \times 3$ .

I gradienti vengono rappresentati come mappa normale.

superiore mostra la possibilità di ottenere risultati stabili, anche se i bordi vengono leggermente sfocati in confronto al metodo di Poisson. La riga inferiore mostra che il LLM fornisce un effetto levigato e migliora il PSNR dell'immagine. L'effetto di levigatura può essere considerato simile a quello del filtro Trilaterale che approssima le intensità con superfici piatte a tratti (contrapposto al LLM, che effettua questa approssimazione pixel per pixel).

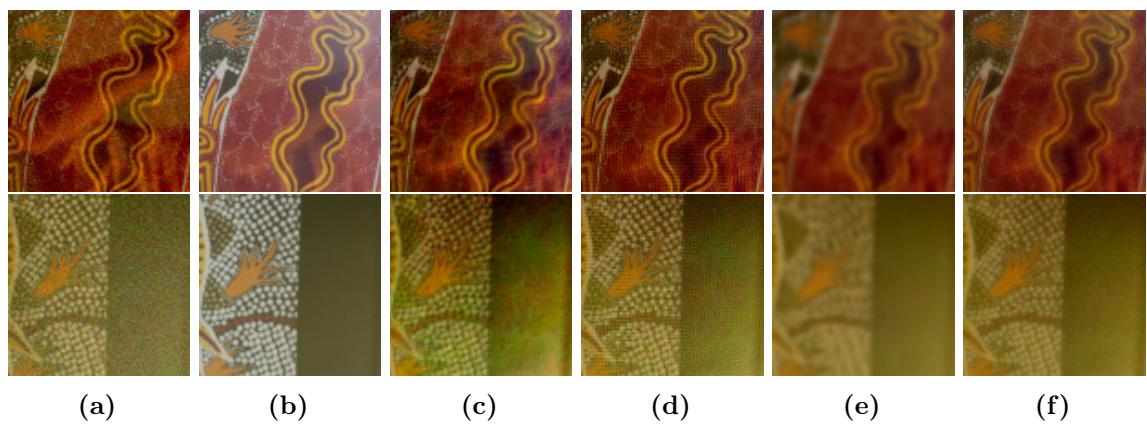
### 5.1.2 Rimozione di Riflessi

Figura 5.2 mostra i risultati della rimozione di riflessi tramite il LLM. Figura 5.3 mostra alcuni dettagli delle immagini. L'obiettivo è di rimuovere il riflesso del fotografo (ed il rumore dovuto al non utilizzo del flash in condizioni di scarsa illuminazione) dall'immagine originale (a) utilizzando l'immagine guida (b). Come vettore di offset  $sv$  (Equazione 2.9) viene utilizzato nuovamente il colore medio dell'immagine originale (a) nel restauro delle immagini (c-f). I risultati (c) e (d) sono ottenuti tramite il metodo di Poisson utilizzando la differenza finita in avanti con l'operatore di Scharr: si può notare che tendono a sorgere artefatti come rumore, decolorazioni e pattern a griglia. Utilizzando il LLM come descritto in Sezione 4.2, si ottiene (e), che appare privo degli artefatti dei casi precedenti, anche se leggermente sfocato. Infine, il risultato (f) viene ottenuto



**Figura 5.2:** Un'applicazione del LLM alla rimozione di riflessi

In (c) e (d), come filtri differenziali, vengono utilizzati rispettivamente la differenza in avanti e l'operatore di Scharr  $3 \times 3$ . In (e), viene utilizzato il LLM con una patch quadrata  $5 \times 5$ .



**Figura 5.3:** Viste in dettaglio di Figura 5.2.

Da sinistra a destra: (a) Originale, (b) Guida, (c) Poisson con corta risposta all'impulso, (d) Poisson con lunga risposta all'impulso, (e) LLM, (f) combinazione del LLM con Poisson a breve risposta all'impulso.



(a) Ombra proiettata sulla Radcliffe Camera (Oxford University)



(b) Ombra di un uomo proiettata su un prato

**Figura 5.4:** Coppie di immagini utilizzate per la rimozione di ombre

A sinistra è mostrata l'immagine originale, a destra l'immagine con l'ombra restaurata, a meno del bordo.

combinando i metodi (c) e (e) come descritto in Sottosezione 2.2.4. Confrontato con entrambi i metodi, (f) presenta decolorazioni ridotte mantenendo una buona nitidezza dei bordi.



**Figura 5.5:** Sfocature della rimozione di ombre tramite LLM

Dettagli del risultato della ristorazione del bordo Poisson image editing utilizzando il LLM con regioni  $\Omega_i$  di raggio unitario quando applicato a (a) Figura 5.4a e (b) Figura 5.4b.

### 5.1.3 Rimozione di Ombre

Figura 5.6 mostra i risultati della rimozione di ombre applicata alle immagini di Figura 5.4. Come è evidente da Figura 5.5, la regione considerata dal LLM per il calcolo e l'integrazione dei gradienti è eccessiva per questo compito, anche ponendo il raggio delle regioni  $\Omega_i$  a 1. Per questo motivo, in Figura 5.6b e Figura 5.6c, la ristorazione è stata effettuata su versioni delle immagini interpolate bi-cubicamente di un fattore 4—questo equivale ad adottare finestre di raggio  $\frac{1}{4}$  di pixel.

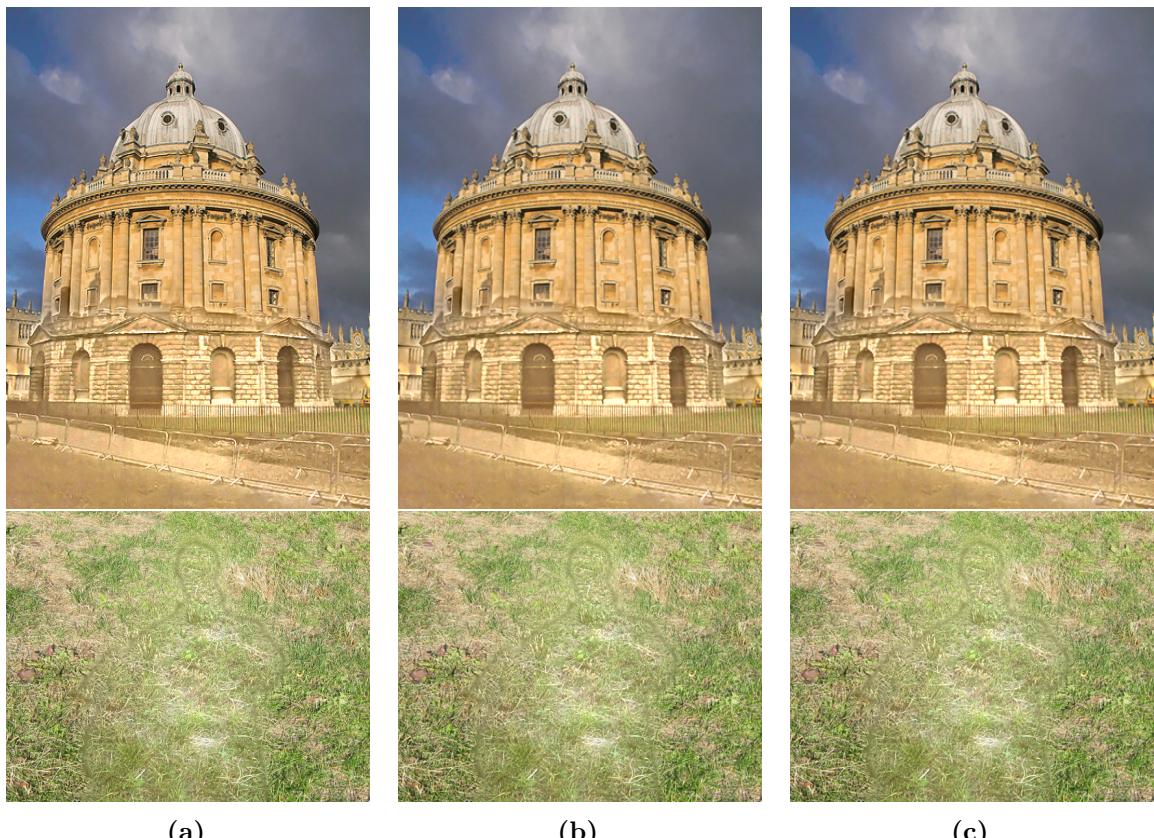
Figura 5.7 mostra un dettaglio dei risultati di Figura 5.6. Si può notare in questa figura che le differenze tra i risultati ottenuti sono visivamente minime ed in buona parte dovute al processo di interpolazione e decimazione.

## 5.2 Trasformazioni di colore tra coppie di immagini

Come applicazione è stata scelta la *composizione di immagini flash/no-flash*<sup>1</sup> tramite GF (questa applicazione è stata originariamente proposta in [8; 55]) in quanto l'immagine *flash* (guida) e l'immagine *no-flash* (input) includono molte corrispondenze errate causate da sotto/sovraesposizioni in regioni statiche. Per valutare i risultati della classificazione tramite SVM e del *blending* effettuato con Equazione 3.11, è stata addestrata la SVM su un dataset di 30 coppie di immagini corredate da *trimap*, come descritto in Sottosezione 3.3.1. Alle immagini di input è

---

<sup>1</sup>La composizione di immagini flash/no-flash consiste nel restauro di un'immagine di una scena notturna ottenuta senza l'utilizzo del flash, quindi con rumore dovuto all'alta sensibilità necessaria, effettuata utilizzando come immagine guida un'immagine della stessa scena, ottenuta però con l'ausilio del flash e, quindi, con rumore molto ridotto ma con colori alterati dalla luce del flash ed eventuali saturazioni.



**Figura 5.6:** Risultati della rimozione di ombre

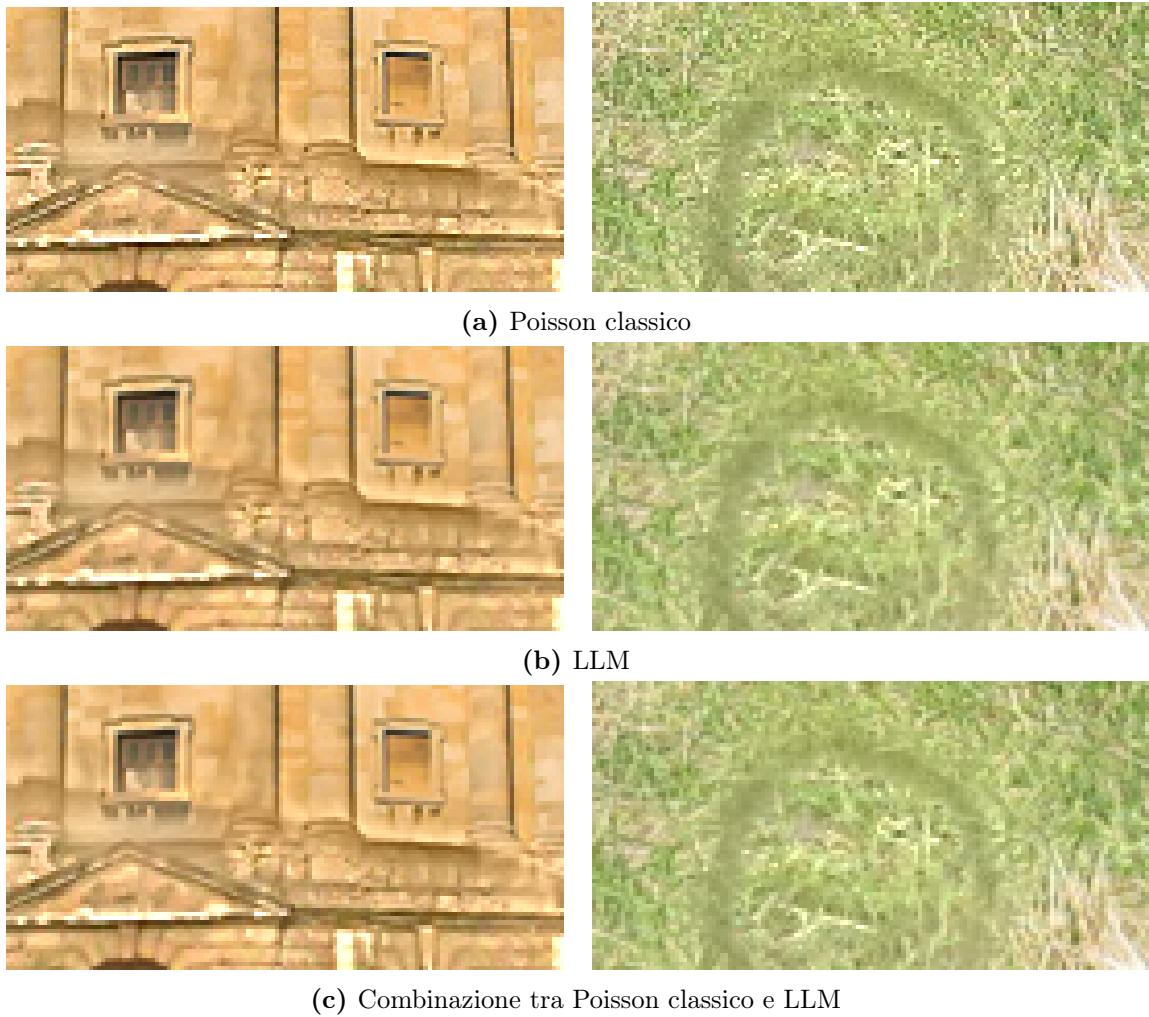
In alto: risultati relativi alle immagini di Figura 5.4a, in basso: risultati relativi alle immagini di Figura 5.4b.

Il bordo è stato restaurato applicando il Poisson image editing secondo (a) secondo l'algoritmo classico [54], (b) tramite LLM, (c) combinando il LLM con l'algoritmo classico.

stato aggiunto del rumore Gaussiano  $\sim \mathcal{N}(0, 0.06)$ , e le intensità delle immagini utilizzate per l'addestramento e per il test sono state normalizzate tra  $[0, 1]$ .

La Figura 5.8 mostra il risultato della classificazione fatta con la SVM: le aree dinamiche nella coppia di immagini Figura 5.8a e Figura 5.8b vengono automaticamente classificate come mostrato in Figura 5.8c, la quale mostra i valori di  $\alpha(y_i)$  come definito in Equazione 3.11. Le regioni statiche appaiono quindi in nero ( $\alpha(y_i) = 0$ ), mentre le regioni dinamiche appaiono in bianco ( $\alpha(y_i) = 1$ ).

La Figura 5.9 mostra un confronto tra il GF pesato, il GF proposto, il GF ottenuto tramite blending ed alcuni metodi standard [4; 8; 56]. Il BM3D [56] (Figura 5.9a) viene mostrato come rappresentante dei metodi di *denoising* su un'immagine singola, anche se causa pattern innaturali nelle immagini risultanti. I risultati ottenuti tramite il metodo originale di questa applicazione [8] (Figura 5.9b) ed il GF standard [4] (Figura 5.9c) sono simili e le loro regioni dinamiche



**Figura 5.7:** Dettaglio dei risultati di Figura 5.6

A sinistra: dettaglio dei risultati relativi alle immagini di Figura 5.4a, a destra: dettaglio dei risultati relativi alle immagini di Figura 5.4b.



**Figura 5.8:** Risultato della fase di test della SVM

Utilizzando la coppia di immagini (a) e (b), la SVM fornisce in output (c).



**Figura 5.9:** Risultati della composizione di immagini flash/no-flash

soffrono di sfocature e decolorazioni, principalmente causate dal disallineamento tra i pixel. Il GF pesato [30] (Figura 5.9d) è in grado di ridurre le sfocature preservando meglio i bordi delle regioni dinamiche, ma lasciando comunque alcune decolorazioni. Il GF proposto (Figura 5.9e) ottiene buoni risultati nelle regioni dinamiche ma causa la comparsa di decolorazioni nelle aree scure. Il risultato finale (Figura 5.9f) ottenuto come *blending* del GF pesato e del GF proposto presenta un aspetto naturale, riducendo le decolorazioni causate da entrambi i metodi.

La performance della classificazione effettuata dalla SVM (mostrata in Figura 5.8c) può essere considerata leggermente eccessiva in quanto alcune regioni statiche sono state erroneamente classificate (e.g., le travi del campanile), ma in queste regioni i risultati di entrambi i GF risultano simili e, in ogni caso, il problema è riducibile specificando queste situazioni nel dataset di addestramento. Questo problema è trattabile in modo simile all'*image matting* [1; 57], ossia, etichette definite come scarabocchi o trimap possono essere estese automaticamente a regioni simili. Questa tecnica potrebbe quindi essere utilizzata per migliorare la precisione di classificazione.

# Capitolo 6

## Conclusioni

In questo documento sono stati presentati ed analizzati due metodi di elaborazione di immagini volte all'ottimizzazione.

Il primo è un'applicazione del LLM, già studiato per il dominio del colore, al dominio dello spazio. Questa ha portato alla costruzione di un metodo molto generico applicabile come alternativa robusta al rumore nella risoluzione di equazioni di Poisson. Il problema principale del metodo risulta essere la perdita di definizione lungo i bordi dell'immagine, per cui viene proposto un *blending* con un metodo classico meno prone a questo problema. In futuro, per gestire meglio questo effetto, si potrebbe analizzare la selezione di patch di dimensione adattiva, come fatto da Lu et al. [29] per il dominio del colore.

Un'ulteriore ottimizzazione applicabile al metodo è la gestione di patch locali di dimensione frazionaria. Questo permetterebbe l'utilizzo di finestre di dimensione molto ridotta senza che sia richiesta l'elaborazione su una versione interpolata dell'immagine, con conseguente miglioramento dei tempi di elaborazione in questi casi. Un esempio di situazione in cui l'ottimizzazione risulterebbe utile è stato mostrato in Sottosezione 5.1.3.

Il secondo metodo analizzato è un GF per regioni dinamiche aiutato da un metodo per la classificazione delle stesse. Invece di usare corrispondenze tra pixel, il metodo introduce corrispondenze tra matrici di covarianza locali. Nella classificazione, è stata utilizzata una SVM con valori numerici ottenuti dalle equazioni dei GF come *feature*. La combinazione di GF può risultare in immagini con ridotte quantità di artefatti: in futuro, altri metodi di classificazione, *blending*, e combinazione potrebbero essere esplorati per ridurre ulteriormente questi effetti.

## Capitolo 7

# Stesura della Tesi

Questo documento è stato scritto interamente<sup>1</sup> in MultiMarkdown (MMD) [58] e, successivamente, tradotto in L<sup>A</sup>T<sub>E</sub>X.

Markdown, di cui MMD è un'estensione, è uno strumento ideato da Gruber [59] ed originariamente pensato per permettere a scrittori web di lavorare utilizzando un formato testuale semplice da leggere e semplice da scrivere, convertibile in XHTML strutturalmente valido. Visto il suo successo, Penney [58] ha deciso di costruire un'estensione del linguaggio che permetesse di esportare i documenti non solo in HTML, bensì anche in L<sup>A</sup>T<sub>E</sub>X (compilabile quindi come PDF), OpenDocument (ODF)—e, di conseguenza, RTF, Microsoft Word, e virtualmente qualsiasi altro formato di word-processing—, e fornisse una quantità di features più estesa, come tabelle, note a piè pagina, o citazioni, oltre alla possibilità di includere **metadata**.

Per poter scrivere un'intera tesi in questo linguaggio è stato necessario creare un template ad-hoc, distribuito come repository GitHub<sup>2</sup>, insieme ad una breve documentazione ed al codice di questo documento (fornito come esempio d'uso e su un repository separato<sup>3</sup>). Il template comprende il trattamento delle informazioni fornite come metadata per creare in modo semplice e standardizzato il frontespizio del documento, una serie di pacchetti L<sup>A</sup>T<sub>E</sub>X ritenuti utili e pratici, e già configurati per fornire un ottimo output con i minori aggiustamenti possibili da parte dell'utente.

---

<sup>1</sup>Gli strumenti forniti da MMD sono pensati per essere molto semplici, per cui non permette una gestione fine di figure ed altri oggetti flottanti. In particolare per la combinazione di sottofigure, è stato necessario scrivere il codice L<sup>A</sup>T<sub>E</sub>X relativo, cosa fattibile comunque comodamente all'interno del documento MMD, inserendo il codice all'interno di commenti.

<sup>2</sup><http://github.com/Denaun/mmd-thesis>

<sup>3</sup><http://github.com/Denaun/master-thesis>

**Listato 7.1:** Codice per il file *header.mmd*

```

1 LaTeX Input: template/mmd-thesis-header
2 LaTeX Mode: memoir
3 Base Header Level: 2
4 ToC Depth: 2
5 Title: Tecniche di miglioramento di immagini basate sulla
       distribuzione locale del colore
6 Formatted Title: Tecniche di miglioramento di immagini
       basate sulla distribuzione locale del colore
7 Author: Maurizio Zucchelli
8 Registration Number: 91665
9 Supervisor: Prof. Nicola Adami
10 Assistant Supervisor: Prof. Keiichiro Shirai
11 Second Assistant Supervisor: Prof. Sergio Benini
12 University: Università degli Studi di Brescia
13 Faculty: Dipartimento di Ingegneria dell'Informazione
14 Course: Corso di Laurea Magistrale in Ingegneria
15           Informatica
16 Academic Year: 2013-2014
17 BibTeX: papers/bibliography.bib
18 Languages: italian
19 LaTeX Input: template/mmd-natbib-plain
20 LaTeX Input: template/mmd-thesis-begin-doc
21 LaTeX Input: mmd-equation-mode
22 LaTeX Footer: template/mmd-thesis-footer

```

L'intento di questo lavoro è stato quello di uniformare, in particolare all'interno dell'Università degli Studi di Brescia, la struttura dei documenti di tesi senza per questo appesantire il lavoro degli studenti ma, anzi, fornendo loro un metodo più efficace per descrivere il lavoro svolto. Essendo il template stato utilizzato solamente per la scrittura di questo documento, è possibile che siano necessari ulteriori aggiustamenti<sup>4</sup> ed inclusioni di packages—e.g., per la gestione di tabelle. Tutto sommato si ritiene che il lavoro abbia portato ad un template già in grado di produrre ottimi risultati. La semplicità della scrittura di documenti di tesi tramite la combinazione di MMD ed il template costruito risulta evidente osservando il frontespizio (Listato 7.1): fornendo solamente le informazioni *fondamentali*, si ottiene un frontespizio correttamente formattato.

Un'estensione interessante del lavoro potrebbe essere l'ampliamento del linguaggio MultiMarkdown. Questo permetterebbe la gestione nativa di sotto-figure e di tutto ciò che viene trattato in L<sup>A</sup>T<sub>E</sub>X all'interno del template. Si potrebbe così sfruttare a pieno la potenza del compilatore, permettendo di visualizzare i documenti di tesi anche in formato HTML, secondo necessità.

---

<sup>4</sup>La proposta di modifiche al template è facilmente gestibile attraverso il sistema di **gestione di issues** fornito da GitHub, secondo cui qualsiasi utente è libero di segnalare problemi, proporre modifiche e/o soluzioni ed anche derivare una propria versione del template partendo dall'originale (*fork*).

## Appendice A

# Matrice di Centratura Locale

Definendo i vettori di campionamento  $\boldsymbol{\delta}_i$  e  $\boldsymbol{\mu}_i \in \mathbb{R}^N$  come

$$\boldsymbol{\delta}_i(k) = \begin{cases} 1 & \text{se } k = i \\ 0 & \text{altrimenti} \end{cases} \quad \text{e} \quad \boldsymbol{\mu}_i(k) = \begin{cases} \frac{w_{ik}}{\sum_j w_{ij}} & \text{se } k \in \Omega_i \\ 0 & \text{altrimenti} \end{cases} \quad (\text{A.1})$$

in modo rispettivamente da estrarre l'elemento  $i$ -esimo e calcolare la media degli elementi appartenenti ad  $\Omega_i$  di  $\mathbf{z}$  se applicati come  $z_i = \boldsymbol{\delta}_i^\top \mathbf{z}$  e  $\bar{z}_i = \boldsymbol{\mu}_i^\top \mathbf{z}$  e le matrici di campionamento da essi derivanti  $\mathbf{S}_i$  e  $\mathbf{M}_i$

$$\mathbf{S}_i = [\boldsymbol{\delta}_{j_1}, \dots, \boldsymbol{\delta}_{j_M}]^\top \quad \text{e} \quad \mathbf{M}_i = [\boldsymbol{\mu}_i, \dots, \boldsymbol{\mu}_i]^\top \in \mathbb{R}^{N \times M} \quad (\text{A.2})$$

in grado di estrarre un set di elementi e medie locali come vettori  $\mathbf{z}_i = [z_{j_1}, \dots, z_{j_M}]^\top$  e  $\bar{\mathbf{z}}_i = [\bar{z}_i, \dots, \bar{z}_i]^\top \in \mathbb{R}^M$  se pre-moltiplicati a  $\mathbf{z}$  (i.e.,  $\mathbf{z}_i = \mathbf{S}_i \mathbf{z}$  e  $\bar{\mathbf{z}}_i = \mathbf{M}_i \mathbf{z}$ ), è possibile definire una matrice di centratura locale come:

$$\mathbf{V}_i = \mathbf{S}_i - \mathbf{M}_i \implies \mathbf{V}_i \mathbf{z} \rightarrow \{z_j - \bar{z}_i \text{ s.t. } j \in \Omega_i\} \quad (\text{A.3})$$

## Appendice B

### Calcolo efficiente di $\mathbf{z}$

Equazione 2.8 contiene un'enorme—anche se sparsa—matrice di dimensione  $N \times 4M$ , precisamente in  $\sum_i \mathbf{V}_i^\top \mathbf{W}_i \mathbf{V}_i$ . Per questo è richiesto un metodo di calcolo efficiente in modo da ridurre la sua complessità e rendere la risoluzione dell'equazione un problema gestibile.

Qui viene descritto il caso semplice in cui i pesi siano costanti ( $w_{ij} = 1$ ). In questo caso, i costi del calcolo possono essere ridotti drasticamente con l'aiuto del metodo dell'immagine integrale (*summed-area table*, o SAT) [27]. Nel caso generico in cui i pesi  $w_{ij} \geq 0$  siano adattivi su ogni patch, il calcolo è effettuabile filtrando ogni riga di  $\mathbf{V}_i^\top \mathbf{W}_i \mathbf{V}_i$  come nel metodo descritto da Levin et al. [60].

Quando tutti i pesi sono ignorabili, ossia  $w_{ij} = 1$ , l'operazione è semplificabile come  $\mathbf{V}_i^\top \mathbf{W}_i \mathbf{V}_i = \mathbf{V}_i^\top \mathbf{V}_i$ . Questa matrice risulta essere simmetrica e sparsa, con elementi nelle stesse posizioni di  $\mathbf{V}_i$ , i.e., gli elementi non nulli di  $(\mathbf{V}_i^\top \mathbf{V}_i)\mathbf{x}$  coincidono con quelli di  $\mathbf{V}_i \mathbf{x}$ . Ogni elemento di  $\widehat{\mathbf{x}}_i = a'_i (\mathbf{V}_i^\top \mathbf{V}_i) \mathbf{x}$  consiste quindi ancora in un'operazione di centratura:

$$\widehat{\mathbf{x}}_i(k) = \begin{cases} a'_i(x_k - \bar{x}_i) & \text{se } k \in \Omega_i \\ 0 & \text{altrimenti} \end{cases} \quad (\text{B.1})$$

Ciò significa che il  $k$ -esimo elemento di  $\widehat{\mathbf{x}} = \sum_i \widehat{\mathbf{x}}_i$  può essere scritto come:

$$\widehat{\mathbf{x}}(k) = \sum_{i \in \Omega_j} a'_i(x_k - \bar{x}_i) = \left( x_k \sum_i a'_i \right) - \left( \sum_i a'_i \bar{x}_i \right) \quad (\text{B.2})$$

Lo stesso ragionamento è applicabile agli altri termini con forma simile—i.e.,  $\sum_i b'_i (\mathbf{V}_i^\top \mathbf{V}_i) \mathbf{y}$  e  $\sum_i (\mathbf{V}_i^\top \mathbf{V}_i) \mathbf{z}$  sono calcolati nello stesso modo. Equazione 2.8 può quindi essere riscritta come:

$$\left( Mz_k - \sum_{i \in \Omega_k} \bar{z}_k \right) = \left( x_k \sum_i a'_i \right) + \left( y_k \sum_i b'_i \right) + \sum_i (a'_i \bar{x}_i + b'_i \bar{x}_i) \quad (\text{B.3})$$

Il calcolo delle sommatorie e delle medie può essere effettuato tramite *box filtering* e, inoltre, utilizzando il SAT, l'operazione di filtraggio può essere calcolata in  $O(1)$ , indipendentemente dalla dimensione della patch. Infine si può notare che, nel comune caso in cui la patch sia quadrata e centrata attorno al pixel  $i$ , i valori medi delle coordinate della patch coincidono con le coordinate del pixel  $i$ —i.e.,  $\bar{x}_i \rightarrow x_i$  e  $\bar{y}_i \rightarrow y_i$ .

Questa ottimizzazione rende implicita la definizione della matrice  $\mathbf{A} = \sum_i \mathbf{U}_i$  e pertanto obbligatoria la ricerca di una soluzione tramite metodi indiretti, come il Gradiente Coniugato.

# Bibliografia

- [1] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, 2008. 1, 3, 10, 32, 33, 48
- [2] Adrien Bousseau, Sylvain Paris, and Frédéric Durand. User-assisted intrinsic images. In *ACM Transactions on Graphics (TOG)*, volume 28, page 130. ACM, 2009. 1, 3
- [3] Qi Shan, Jiaya Jia, and Michael S Brown. Globally optimized linear windowed tone mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 16(4):663–675, 2010. 1, 3, 10
- [4] Kaiming He, Jian Sun, and Xiaou Tang. Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409, 2013. 1, 3, 5, 6, 18, 45, 47
- [5] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 536–543. ACM, 2005. 1, 4, 10, 14
- [6] Amit Agrawal, Ramesh Raskar, Shree K Nayar, and Yuanzhen Li. Removing photography artifacts using gradient projection and flash-exposure sampling. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 828–835. ACM, 2005. 1, 4, 25
- [7] Graham D Finlayson, Steven D Hordley, Cheng Lu, and Mark S Drew. On the removal of shadows from images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):59–68, 2006. 1, 4, 24, 26, 28, 29, 30
- [8] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. In *ACM transactions on graphics (TOG)*, volume 23, pages 664–672. ACM, 2004. 1, 44, 45

- [9] Nicholas Apostoloff and Andrew Fitzgibbon. Bayesian video matting using learnt image priors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–407. IEEE, 2004. 3
- [10] Arie Berman, Arpag Dadourian, and Paul Vlahos. Comprehensive method for removing from an image the background surrounding a selected subject, October 17 2000. US Patent 6,134,345.
- [11] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Transactions on Graphics (TOG)*, 21(3):243–248, 2002.
- [12] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2001.
- [13] Mark A Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 18–25. IEEE, 2000.
- [14] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 315–321. ACM, 2004. 3
- [15] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 15–18. IEEE, 2006. 4
- [16] Amit Agrawal, Ramesh Raskar, and Rama Chellappa. What is the range of surface reconstructions from a gradient field? In *Computer Vision–ECCV 2006*, pages 578–591. Springer, 2006. 4, 10, 14
- [17] Graham D Finlayson, Mark S Drew, and Cheng Lu. Intrinsic images by entropy minimization. In *Computer Vision–ECCV 2004*, pages 582–595. Springer, 2004. 4, 28
- [18] Yael Shor and Dani Lischinski. The shadow meets the mask: Pyramid-based shadow removal. In *Computer Graphics Forum*, volume 27, pages 577–586. Wiley Online Library, 2008. 4, 26, 30, 38

- [19] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 249–256. ACM, 2002. 4
- [20] Matthew Harker and Paul O’Leary. Least squares surface reconstruction from measured gradient fields. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.
- [21] Mushfiqur Rouf, Cheryl Lau, and Wolfgang Heidrich. Gradient domain color restoration of clipped highlights. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–14. IEEE, 2012. 4
- [22] Chih-Tsung Shen and Wen-Liang Hwang. Color image enhancement using retinex with robust envelope. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3141–3144. IEEE, 2009. 4, 25
- [23] Hanno Scharr. *Optimal operators in digital image processing*. PhD thesis, 2000. 4, 25, 40
- [24] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2165–2172. IEEE, 2010. 5, 10, 11, 13
- [25] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. National Bureau of Standards Washington, DC, 1952. 5, 12
- [26] Richard Barrett, Michael W Berry, Tony F Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*, volume 43. Siam, 1994. 5
- [27] Franklin C Crow. Summed-area tables for texture mapping. *ACM SIGGRAPH computer graphics*, 18(3):207–212, 1984. 5, 13, 53
- [28] Keiichiro Shirai, Masayuki Okamoto, and Masaaki Ikebara. Noiseless no-flash photo creation by color transform of flash image. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3437–3440. IEEE, 2011. 5, 6

- [29] Jiangbo Lu, Keyang Shi, Dongbo Min, Liang Lin, and Minh N Do. Cross-based local multipoint filtering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 430–437. IEEE, 2012. 10, 19, 49
- [30] Keisuke Iwata, Keiichiro Shirai, and Masahiro Okuda. Local color distribution projection filtering using a homogeneous coordinate system. *ICIP*, 2013. 5, 6, 7, 8, 10, 11, 16, 19, 21, 47, 48
- [31] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *Computer Vision–ECCV 2012*, pages 771–785. Springer, 2012. 5
- [32] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3017–3024. IEEE, 2011. 5
- [33] Ido Omer and Michael Werman. Color lines: Image specific color representation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–946. IEEE, 2004. 6, 31
- [34] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995. 8, 21
- [35] Thorsten Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999. 8, 21
- [36] Kenjiro Sugimoto, Keiichiro Shirai, and Sei-ichiro Kamata. O(1) transposed bilateral filtering for optimization. In *Proc. APSIPA Conf. Signal Process.*, 2014. 17
- [37] Keiichiro Shirai, Masahiro Okuda, and Masaaki Ikebara. Color-line vector field and local color component decomposition for smoothing and denoising of color images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3050–3053. IEEE, 2012. 20
- [38] Shunsuke Ono and Isao Yamada. A convex regularizer for reducing color artifact in color image recovery. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1775–1781. IEEE, 2013. 20

- [39] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998. 21
- [40] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006. 21
- [41] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009. 21
- [42] Graham D Finlayson, Mark S Drew, and Cheng Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85(1):35–57, 2009. 26, 28, 29
- [43] Gunter Wyszecki and Walter Stanley Stiles. *Color science*, volume 8. Wiley New York, 1982. 27
- [44] Edwin H Land and John McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971. 29
- [45] Andrew Blake. Boundary conditions for lightness computation in mondrian world. *Computer Vision, Graphics, and Image Processing*, 32(3):314–327, 1985.
- [46] Gavin Brelstaff and Andrew Blake. Computing lightness. *Pattern Recognition Letters*, 5(2):129–138, 1987. 29
- [47] Graham D Finlayson, Steven D Hordley, and Mark S Drew. Removing shadows from images. In *Computer Vision—ECCV 2002*, pages 823–836. Springer, 2002. 31
- [48] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, pages 1065–1076, 1962. 32
- [49] HG Barrow and JM Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer vision systems*, pages 3–26, 1978. 34
- [50] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 35
- [51] Tai-Pang Wu, Chi-Keung Tang, Michael S Brown, and Heung-Yeung Shum. Natural shadow matting. *ACM Transactions on Graphics (TOG)*, 26(2):8, 2007. 35

- [52] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003. 38
- [53] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):125, 2010. 38
- [54] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 38, 45
- [55] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. In *ACM transactions on graphics (TOG)*, volume 23, pages 673–678. ACM, 2004. 44
- [56] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007. 45
- [57] Yuanzhen Li, Edward Adelson, and Aseem Agarwala. Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. In *Computer Graphics Forum*, volume 27, pages 1255–1264. Wiley Online Library, 2008. 48
- [58] Fletcher T. Penney. Multimarkdown. <http://fletcherpenney.net/multimarkdown/>, 2009. 50
- [59] John Gruber. Markdown. <http://daringfireball.net/projects/markdown/>, 2004. 50
- [60] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM Transactions on Graphics (TOG)*, volume 26, page 70. ACM, 2007. 53