

# Deblur delle immagini

Dario Mylonopoulos

06-03-2021

# Indice

<b>1</b>	<b>Immagini</b>	<b>1</b>
1.1	Immagini in bianco e nero . . . . .	1
1.2	Immagini a colori . . . . .	2
<b>2</b>	<b>Deblur delle immagini</b>	<b>2</b>
2.1	Acquisizione di immagini e fonti di blur . . . . .	3
2.2	Point Spread Functions (PSF) . . . . .	3
2.3	Convoluzione . . . . .	4
2.4	Rumore . . . . .	7
<b>3</b>	<b>Metodi per il deblur e regolarizzazione</b>	<b>7</b>
3.1	Soluzione naive . . . . .	7
3.2	Regolarizzazione . . . . .	8
<b>4</b>	<b>Risultati</b>	<b>9</b>

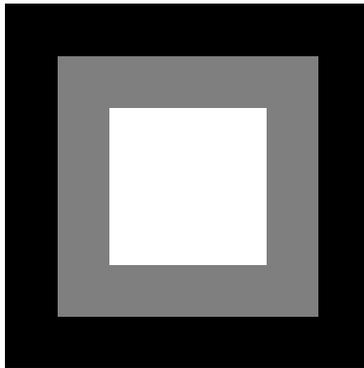
# 1 Immagini

Per rappresentare un'immagine nella memoria di un computer ed effettuare delle operazioni su di esse è necessario codificarne il contenuto attraverso dei numeri. Le immagini sono spesso memorizzate come array bidimensionali e possono essere raffigurate come matrici di numeri i cui elementi rappresentano l'intensità di una piccola sezione quadrata dell'immagine detta pixel. Il numero di pixel in un'immagine è detto risoluzione, più pixel sono presenti più un'immagine apparirà nitida e dettagliata. Per motivi di efficienza relativa allo spazio occupato le immagini sono memorizzate su disco con numeri interi a 8 bit, con valori tra 0 (intensità minima) e 255 (intensità massima). Quando si effettuano operazioni sulle immagini è utile rappresentare i pixel in memoria con numeri floating point compresi tra 0 (intensità minima) e 1 (intensità massima), utilizzeremo quindi quest'ultima rappresentazione.

## 1.1 Immagini in bianco e nero

Per un'immagine in bianco e nero è sufficiente un solo numero per rappresentare il colore di un pixel, al numero 0 è associato il colore nero, al numero 1 il bianco e a tutti i numeri compresi la scala di grigi. Ad esempio la matrice seguente rappresenta un'immagine di dimensione  $7 \times 7$  con un quadrato bianco su sfondo nero con un bordo grigio di un pixel.

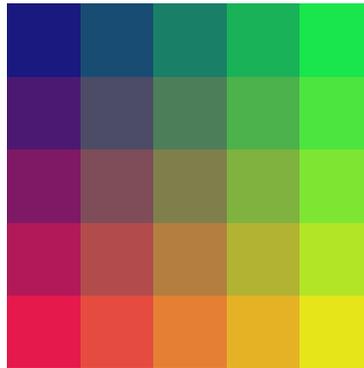
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 1 & 1 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 1 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 1 & 1 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



## 1.2 Immagini a colori

Le immagini a colori possono essere rappresentate in diversi formati, uno dei più comuni è il formato RGB, per ogni pixel si memorizzano 3 numeri, detti canali, che rappresentano l'intensità del colore rosso, verde e blue rispettivamente. Gli altri colori sono ottenuti attraverso la somma di quest'ultimi come accade per gli schermi LCD in cui ogni pixel è formato da tre LED molto vicini di questi colori. Le immagini a colori si possono quindi rappresentare con tre matrici di numeri, una per canale.

$$R = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \end{bmatrix} \quad G = \begin{bmatrix} 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\ 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\ 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\ 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \\ 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.45 & 0.4 & 0.35 & 0.3 \\ 0.45 & 0.4 & 0.35 & 0.3 & 0.25 \\ 0.4 & 0.35 & 0.3 & 0.25 & 0.2 \\ 0.35 & 0.3 & 0.25 & 0.2 & 0.15 \\ 0.3 & 0.25 & 0.2 & 0.15 & 0.1 \end{bmatrix}$$



Per semplicità considereremo solo immagini in bianco e nero, le tecniche utilizzate possono però essere estese a immagini a colori operando sui tre canali individualmente.

## 2 Deblur delle immagini

Con il termine deblur ci si riferisce al recupero di un'immagine nitida a partire da una sfocata o mossa, questa operazione serve per migliorarne la qualità e renderla più utile per lo scopo per cui è stata catturata. Non è quasi mai possibile recuperare l'immagine originale in modo esatto poiché molte delle informazioni e dei dettagli sono perduti nella codifica dell'immagine e possono essere coperti dal rumore, cioè valori presenti nell'immagine causati da elementi esterni all'oggetto osservato. Vedremo in che modo le tecniche di deblur utilizzano un modello matematico per impostare un'equazione la cui soluzione approssimi al meglio l'immagine originale.

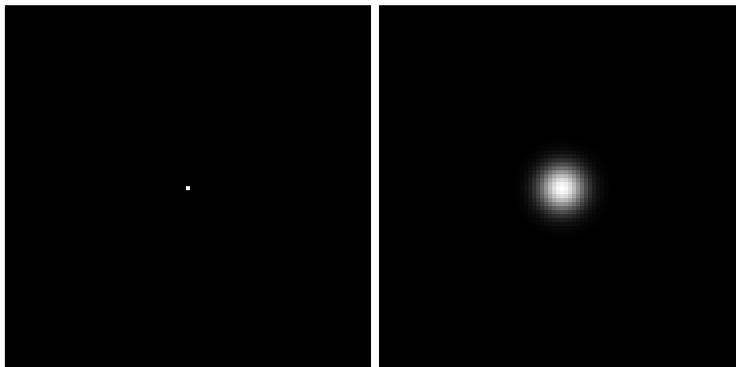
## 2.1 Acquisizione di immagini e fonti di blur

Quando si scatta una fotografia digitale la luce entra attraverso l'obiettivo della macchina fotografica e viene misurata da un sensore formato da una matrice di elementi sensibili alla luce che corrispondono ai pixel dell'immagine. In passato la tecnologia più diffusa per la cattura di immagini digitali era il CCD (Charge Coupled Device), un dispositivo in grado di trasformare la luce in un segnale analogico che viene poi convertito in numeri digitali. Più di recente i sensori utilizzano invece la tecnologia CMOS (Complementary MOS sensors), che sono ormai presenti nella maggior parte delle macchine fotografiche e degli smartphone poiché più efficienti e meno costosi.

Il primo passo per delineare il modello è individuare le fonti di blur dovute al processo di acquisizione delle immagini. La forma più comune di sfocatura è causata dalla mancata messa a fuoco della macchina fotografica che fa sì che la luce che passa attraverso la lente dell'obiettivo influenzi un'area di pixel più ampia del dovuto. In altri casi le cause di blur sono esterne alla macchina, ad esempio nel caso di movimento o di vibrazioni. Le condizioni atmosferiche possono influire sulla qualità dell'immagine catturata, in particolare per immagini acquisite attraverso telescopi astronomici o satelliti.

## 2.2 Point Spread Functions (PSF)

Per ottenere informazioni su come il sistema di cattura influisce sull'immagine ottenuta possiamo condurre il seguente esperimento. Se catturiamo un'immagine con un'unica fonte di luce puntiforme al centro ci aspettiamo idealmente di ottenere un'immagine completamente nera tranne per un unico pixel bianco in posizione centrale. Nel caso reale invece, l'immagine ottenuta in questo modo mostra come la fonte di luce influisce sul colore dei pixel adiacenti, per questo motivo è chiamata Point Spread Function (PSF). Spesso si può fare l'approssimazione che il modo in cui un pixel bianco influenza i pixel vicini sia lo stesso indipendentemente dalla sua posizione nell'immagine, in questo caso la PSF rappresenta completamente l'effetto del blur.



(a) Pixel bianco al centro

(b) Point Spread Function

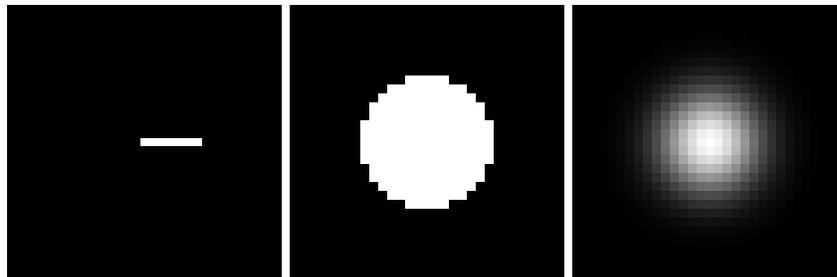
Un modo alternativo di ottenere la PSF è di approssimarla attraverso una funzione matematica che si comporti in modo simile al sistema reale. Ad esempio il blur dovuto al movimento uniforme orizzontale di un pixel può essere rappresentato con una riga di pixel la cui somma delle intensità corrisponde all'intensità del pixel iniziale. Il blur dovuto alla mancata messa a fuoco può essere approssimato dalla formula seguente

$$p_{ij} = \begin{cases} \frac{1}{\pi r^2}, & \text{if } (i - k)^2 + (j - l)^2 \leq r^2 \\ 0, & \text{altrimenti} \end{cases}$$

$p_{ij}$  rappresenta il pixel alla riga  $i$  e colonna  $j$  nella PSF,  $r$  il raggio del blur,  $k$  e  $l$  le coordinate del pixel centrale. La PSF risultante è un cerchio di raggio  $r$  la cui somma delle intensità dei pixel è uguale a 1. Un'altra funzione utilizzata spesso per descrivere il blur dovuto alle condizioni atmosferiche è la funzione Gaussiana bidimensionale, che nel caso di un blur simmetrico è descritta da

$$p_{ij} = \exp\left(-\frac{1}{2}\left(\frac{(i - k)}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{(j - l)}{\sigma}\right)^2\right)$$

dove  $\sigma$  rappresenta la deviazione standard della funzione Gaussiana e influisce sulla dimensione del blur. Come nei casi precedenti la PSF del blur Gaussiano deve essere successivamente normalizzata affinché la somma dei suoi pixel sia uguale a 1.



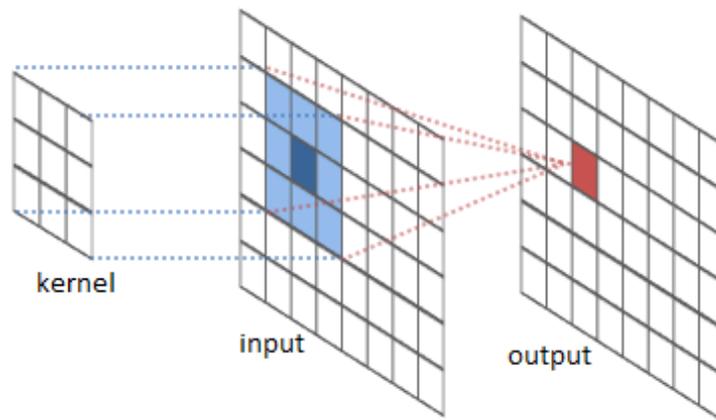
(a) PSF movimento      (b) PSF non a fuoco      (c) PSF blur Gaussiano

### 2.3 Convoluzione

Vediamo come la PSF può essere utilizzata per ottenere un'immagine sfocata a partire da una nitida. La PSF rappresenta l'impatto che ogni pixel dell'immagine originale ha sui pixel vicini, un modo equivalente di pensare questa operazione è che ogni pixel dell'immagine sfocata è una somma pesata dei pixel vicini nell'immagine originale, dove i pesi sono dati dai valori dei pixel della PSF. La convoluzione è un'operazione matematica che nel caso continuo unidimensionale è descritta dalla formula

$$b(s) = \int_{-\infty}^{\infty} p(s - t)x(t) dt$$

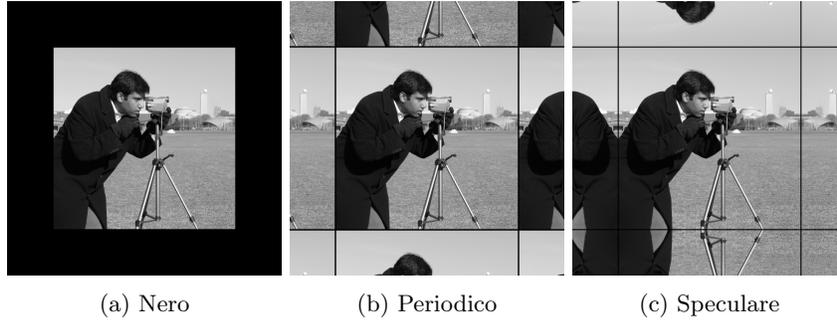
Il risultato  $b(s)$  è una somma pesata dei valori della funzione  $x(t)$  dove i pesi sono dati dalla funzione  $p$ . La funzione dei pesi è specchiata rispetto all'asse delle ordinate  $p(-t)$  e traslata per ottenere  $p(s-t)$ . Nel caso discreto l'integrale diventa una somma finita di termini,  $x$  rappresenta l'immagine originale,  $b$  l'immagine dopo il blur e  $p$  la PSF. La convoluzione di un'immagine con una PSF può quindi essere pensata nel modo seguente, per ogni pixel dell'immagine in output si sovrappone la PSF specchiata orizzontalmente e verticalmente (detta kernel) all'immagine in input in modo tale che sia centrata sul pixel in considerazione, si moltiplica ogni pixel dell'immagine con il pixel del kernel sovrapposto e si sommano tutti i termini per ottenere il pixel in output.



Quando consideriamo i pixel che si trovano sul bordo parte del kernel esce dall'immagine, in questo caso le informazioni per calcolare il colore del pixel in output sono incomplete poiché l'immagine ha una dimensione finita. In base alla tipologia di immagine e al fine per cui è stata catturata può essere utile fare diversi tipi di assunzioni riguardo ai pixel mancanti, ad esempio per una fotografia astronomica può essere una valida approssimazione assumere che tutti i pixel mancanti rappresentino lo spazio vuoto e siano neri. In altri casi si può assumere che l'immagine si ripeta in modo periodico, cioè che una copia dell'immagine si trovi ai lati dell'immagine originale. Similmente si può considerare che la copia dell'immagine sia speculare rispetto all'originale in modo tale che in bordi combacino senza creare discontinuità. Le varie possibilità prendono il nome di condizioni al contorno (boundary conditions).

L'operazione di convoluzione può essere espressa anche in forma di applicazione lineare. Sia  $X$  l'immagine originale di dimensione  $n \times m$  e  $x$  il vettore ottenuto concatenando le colonne di  $X$  di dimensione  $N = n \times m$ , allo stesso modo  $b$  è il vettore di dimensione  $N$  delle colonne di  $B$  immagine sfocata e  $A$  è la matrice  $N \times N$  che rappresenta l'applicazione lineare i cui elementi sono ottenuti dalla PSF e dipendono dall'approssimazione che si decide di utilizzare per le informazioni mancanti al bordo dell'immagine. La formula che descrive la convoluzione è

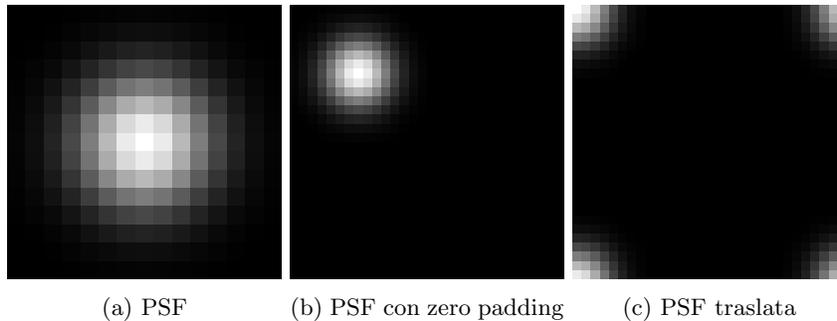
$$b = Ax$$



Quando si considerano le condizioni al contorno periodiche la matrice di blur  $A$  presenta una struttura particolare detta BCCB (Block Circulant with Circulant Blocks). In questo caso la matrice può essere decomposta nel modo seguente:

$$A = F^{-1} \Lambda F$$

$F$  rappresenta la matrice della trasformata di Fourier discreta (DFT) bidimensionale e  $F^{-1}$  la trasformata inversa.  $\Lambda$  è una matrice diagonale che contiene gli autovalori di  $A$ . Calcolare il prodotto  $Ax$  equivale quindi a effettuare la trasformata di Fourier di  $x$ , moltiplicare elemento per elemento con gli autovalori di  $A$  ed effettuare la trasformata inversa. Questa decomposizione rende l'applicazione del blur particolarmente efficiente poiché esiste un algoritmo chiamato Fast Fourier Transform (FFT) per calcolare la trasformata e la sua inversa in tempo  $O(N \log N)$  invece del tempo  $O(N^2)$  necessario per il prodotto tra matrice e vettore. Inoltre non è necessario memorizzare  $A$  interamente in memoria, ma è sufficiente memorizzare i suoi autovalori che si possono ottenere a partire dalla PSF nel modo seguente. Ricordiamo che la PSF può essere di dimensioni inferiori rispetto all'immagine su cui vogliamo applicare il blur, bisogna quindi aggiungere gli zeri necessari a raggiungere la stessa dimensione (zero padding). Successivamente occorre traslare la PSF in modo tale che il pixel in posizione centrale si sposti in posizione  $(0,0)$  cioè nell'angolo in alto a sinistra. Infine si esegue la trasformata di Fourier bidimensionale per ottenere la diagonale della matrice  $\Lambda$ .



## 2.4 Rumore

Le immagini acquisite contengono quasi sempre del rumore che può essere di vari tipi e provenire da diverse fonti. Il rumore più comune che consideriamo è il rumore additivo, cioè che si ottiene sommando dei valori a singoli pixel dell'immagine. La fonte più comune di rumore nel caso di immagini catturate attraverso il sensore di una macchina fotografica è dovuta ad errori nella lettura del segnale analogico. Questo rumore è formato da valori indipendenti e distribuiti in modo identico (rumore bianco) che provengono da una distribuzione Gaussiana con media zero e deviazione standard proporzionale all'ampiezza del rumore, per questo motivo è chiamato rumore bianco Gaussiano (Gaussian white noise). Un'altra fonte comune di rumore è data dall'errore introdotto nella quantizzazione dei valori analogici in digitale, cioè al momento della codifica dell'immagine i valori letti dal sensore sono approssimati al valore discreto più vicino. In questo caso il rumore ha sempre una distribuzione uniforme e la deviazione standard è inversamente proporzionale al numero di bit utilizzato nella codifica.

Nel modello per il deblur consideriamo il rumore Gaussiano uniforme, che è descritto dalla formula seguente

$$B = B_{exact} + E$$

$B_{exact}$  rappresenta l'immagine  $m \times n$  sfocata senza la presenza di rumore,  $E$  è la matrice  $m \times n$  che contiene i valori del rumore bianco Gaussiano e  $B$  è l'immagine corrotta osservata.

## 3 Metodi per il deblur e regolarizzazione

Abbiamo visto come l'effetto del blur e del rumore su un'immagine è descritto dalla formula

$$b = Ax + e$$

nella quale  $b$  è l'osservazione di cui siamo in possesso,  $A$  rappresenta la funzione di blur che approssimiamo a partire dalla PSF,  $x$  l'immagine originale ed  $e$  il rumore bianco Gaussiano. Il problema di deblur consiste quindi nella ricerca della  $x$  che soddisfi questa equazione.

### 3.1 Soluzione naive

Se ignoriamo la presenza del rumore nella formula appena vista la soluzione  $x_{naive}$  può essere trovata risolvendo il sistema lineare ed è quindi data da

$$x_{naive} = A^{-1}b$$

o equivalentemente dalla soluzione del problema di minimo

$$x_{naive} = \arg \min_x \frac{1}{2} \|Ax - b\|_2^2$$

Tuttavia la soluzione ottenuta con questo procedimento risente in modo consistente della presenza del rumore nell'osservazione, infatti la soluzione naive è in realtà il risultato della formula

$$x_{naive} = A^{-1}b_{exact} + A^{-1}e$$

cioè

$$x_{naive} = x + A^{-1}e$$

Il termine  $A^{-1}e$  rappresenta la presenza del rumore nella ricostruzione ed è detto rumore inverso, quasi sempre ha valori molto elevati e domina nella soluzione trovata.



(a) Osservazione



(b) Ricostruzione naive

### 3.2 Regolarizzazione

Per limitare i problemi relativi alla presenza del rumore inverso si può introdurre un termine di regolarizzazione che garantisca alcune caratteristiche nella soluzione trovata. Un esempio di regolarizzazione è il metodo di Tichonov che fa sì che la soluzione del problema sia data dalla formula

$$x = \arg \min_x \frac{1}{2} \|Ax - b\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$$

$\|Ax - b\|_2^2$  è detto termine di fedeltà ed indica la coerenza tra la ricostruzione e l'immagine di partenza.  $\lambda \|x\|_2^2$  è il termine di regolarizzazione di parametro  $\lambda$ , più  $\lambda$  è elevato più peso è dato alla regolarizzazione. La scelta del parametro di regolarizzazione  $\lambda$  può dipendere dalle caratteristiche dell'immagine di partenza e dalla tipologia ed entità del rumore presente. Una possibile scelta nel caso in cui si è a conoscenza della norma del rumore  $\|e\|_2^2$  è il principio di discrepanza: si provano diversi valori di  $\lambda$  e si sceglie il più grande per cui

$$\|Ax_\lambda - b\|_2^2 \leq \|e\|_2^2$$

dove  $x_\lambda$  rappresenta la soluzione ottenuta utilizzando  $\lambda$  come parametro di regolarizzazione.

## 4 Risultati

Per valutare la qualità della ricostruzione rispetto all'immagine originale oltre al confronto visivo utilizziamo il valore del peak signal-to-noise ratio (PSNR) definito in termini dell'errore quadratico medio (MSE) nel modo seguente

$$\text{MSE} = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^m (x_{ij}^* - x_{ij})^2}{nm}}$$

$$\text{PSNR} = 20 \log \left( \frac{\max x^*}{\sqrt{\text{MSE}}} \right)$$

dove  $x^*$  rappresenta l'immagine originale di dimensioni  $n \times m$  e  $x$  la ricostruzione ottenuta, con  $\max x^*$  si indica il valore del pixel massimo nell'immagine  $x^*$ . Maggiore è il valore del PSNR più fedele è la ricostruzione.

Utilizziamo come esempio una fotografia corrotta con un blur Gaussiano di dimensione  $7 \times 7$  e deviazione standard 0.5 e rumore bianco Gaussiano di deviazione standard 0.02. Riportiamo 2 soluzioni ottenute con il metodo di Tichonov per due diverse scelte del parametro di regolarizzazione  $\lambda$



(a) Originale



(b) Corrotta, PSNR = 24.35



(c)  $\lambda = 0.02$ , PSNR = 25.33



(d)  $\lambda = 0.04$ , PSNR = 25.39

Infine riportiamo un grafico del PSNR al variare di  $\lambda$  tra i valori 0.01 e 0.1, indicando il valore del PSNR per l'immagine corrotta e il valore di  $\lambda$  ottenuto con il principio di discrepanza.

