

Restauro cieco di immagini mediante ottimizzazione a sciame di particelle

Luigi Pinca

Dipartimento di matematica e informatica
Università degli Studi di Perugia

9 Luglio, 2009



Degradazioni trattate

Durante il processo di acquisizione e memorizzazione è possibile che le immagini digitali vengano deteriorate da effetti indesiderati:

- la sfocatura
- il rumore



Degradazioni trattate

Durante il processo di acquisizione e memorizzazione è possibile che le immagini digitali vengano deteriorate da effetti indesiderati:

- la sfocatura
- il rumore



originale



degradata

Definizione formale

L'intensità di ogni pixel sostituita con la media pesata del valore di quelli adiacenti. I pesi sono le entrate di una matrice

$M \in \mathbb{R}^{(2h+1) \times (2h+1)}$, $h \in \mathbb{N}^+$. Es: $h = 4$

$$M = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$



Degradazioni trattate

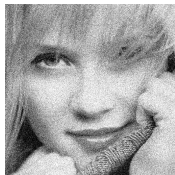
Durante il processo di acquisizione e memorizzazione è possibile che le immagini digitali vengano deteriorate da effetti indesiderati:

- la sfocatura
- il rumore

Si assume Gaussiano, con media nulla e varianza σ^2 nota. Es: $\sigma = 40$

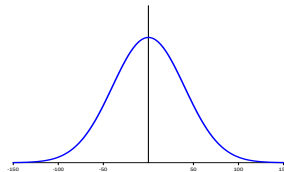


originale



degradata

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



Modello matematico

FORMA LESSICOGRAFICA

$$\mathbf{x} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix} \Rightarrow \vec{\mathbf{x}} = \begin{pmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ x_{1,2} \\ x_{2,2} \\ x_{3,2} \\ x_{1,3} \\ x_{2,3} \\ x_{3,3} \end{pmatrix}$$



Modello matematico

FORMA LESSICOGRAFICA

$$\mathbf{x} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix} \Rightarrow \vec{\mathbf{x}} = \begin{pmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ x_{1,2} \\ x_{2,2} \\ x_{3,2} \\ x_{1,3} \\ x_{2,3} \\ x_{3,3} \end{pmatrix}$$

PROCESSO DI DEGRADAZIONE

$$\vec{\mathbf{y}} = A\vec{\mathbf{x}} + \vec{\mathbf{n}}$$

$\vec{\mathbf{y}} \in \mathbb{R}^{m \times n}$ immagine osservata

$\vec{\mathbf{x}} \in \mathbb{R}^{m \times n}$ immagine ideale

$\vec{\mathbf{n}} \in \mathbb{R}^{m \times n}$ rumore

$A \in \mathbb{R}^{(m \times n) \times (m \times n)}$ operatore lineare



Invertire il processo di degradazione

Problema inverso

Il problema del **restauro d'immagine** consiste nel trovare la migliore stima di $\vec{x}$, conoscendo la matrice di sfocatura A , l'immagine osservata $\vec{y}$ e la varianza σ^2 del rumore.



Invertire il processo di degradazione

Problema inverso

Il problema del **restauro d'immagine** consiste nel trovare la migliore stima di $\vec{x}$, conoscendo la matrice di sfocatura A , l'immagine osservata $\vec{y}$ e la varianza σ^2 del rumore.

Cattive notizie

Il problema è mal-posto nel senso di Hadamard ovvero la soluzione può:

- non esistere
- non essere unica
- non essere stabile a variazioni sui dati



Regolarizzazione del problema

Per rendere il problema trattabile è necessaria una **tecnica di regolarizzazione** che impone i seguenti vincoli sulla soluzione:

- consistenza con i dati
- uniformità
- conservazione dei bordi



Regolarizzazione del problema

Per rendere il problema trattabile è necessaria una **tecnica di regolarizzazione** che impone i seguenti vincoli sulla soluzione:

- consistenza con i dati
- uniformità
- conservazione dei bordi

Per non perdere le discontinuità bisogna usare le **variabili di linea**:

Definizione

Una variabile di linea b_c è una variabile booleana associata ad una **clique** c . Se in c è presente una discontinuità allora la corrispondente variabile di linea b_c assumerà il valore 1, zero altrimenti.



Clique

Definizione

Sia $Q_n = ([1, n] \cap \mathbb{N}) \times ([1, n] \cap \mathbb{N})$ per ogni $n \in \mathbb{N}$. Una clique di ordine k è l'insieme dei punti di Q_n in cui risultano definite le differenze finite di ordine k .

CLIQUE DEL PRIMO ORDINE

$$(1) \quad s \bullet t \bullet$$

$$(2) \quad \begin{matrix} s \bullet \\ t \bullet \end{matrix}$$

$$D_c^{1\vec{x}} = x_s - x_t, \quad \forall c \text{ del tipo (1) e (2).}$$

CLIQUE DEL SECONDO ORDINE

$$(1) \quad s \bullet t \bullet r \bullet$$

$$(2) \quad \begin{matrix} s \bullet \\ t \bullet \\ r \bullet \end{matrix}$$

$$D_c^{2\vec{x}} = x_s - 2x_t + x_r, \quad \forall c \text{ del tipo (1) e (2).}$$



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

$$E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- **Fedeltà con i dati**

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- Fedeltà con i dati

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$

- Vincolo di uniformità

Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- Fedeltà con i dati

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$

- Vincolo di uniformità

$\vec{b}$ è l'insieme di tutte le variabili di linea b_c



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- Fedeltà con i dati

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} [\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c] .$$

- Vincolo di uniformità

$\vec{b}$ è l'insieme di tutte le variabili di linea b_c

C_k è l'insieme di tutte le clique di ordine k



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- Fedeltà con i dati

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$

- Vincolo di uniformità

$\vec{b}$ è l'insieme di tutte le variabili di linea b_c
 C_k è l'insieme di tutte le clique di ordine k
 $\lambda \in \mathbb{R}$ è un parametro di regolarizzazione



Funzione energia

La soluzione del problema può essere definita come quella coppia $(\vec{x}, \vec{b})$ che minimizza la seguente **funzione energia**:

- Fedeltà con i dati

$$E(A, \vec{x}, \vec{b}) = ||\vec{y} - A\vec{x}||^2 + \sum_{c \in C_k} \left[\lambda^2 (D_c^k \vec{x})^2 (1 - b_c) + \alpha b_c \right].$$

- Vincolo di uniformità

$\vec{b}$ è l'insieme di tutte le variabili di linea b_c

C_k è l'insieme di tutte le clique di ordine k

$\lambda \in \mathbb{R}$ è un parametro di regolarizzazione

$\alpha \in \mathbb{R}$ è un costo da pagare in presenza di discontinuità



Eliminazione del processo di linea

È possibile trattare le variabili di linea b_c in maniera implicita:

$$E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} h_{\alpha, \lambda}(D_c^k \vec{x}, b_c),$$



Eliminazione del processo di linea

È possibile trattare le variabili di linea b_c in maniera implicita:

$$E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} h_{\alpha, \lambda}(D_c^k \vec{x}, b_c),$$

$$h_{\alpha, \lambda}(t, b_c) = \lambda^2 t^2 (1 - b_c) + \alpha b_c.$$

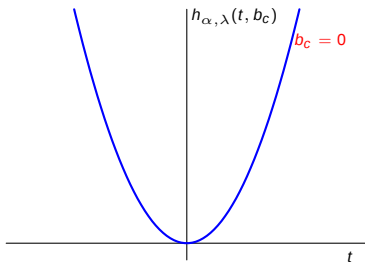


Eliminazione del processo di linea

È possibile trattare le variabili di linea b_c in maniera implicita:

$$E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} h_{\alpha, \lambda}(D_c^k \vec{x}, b_c),$$

$$h_{\alpha, \lambda}(t, b_c) = \lambda^2 t^2 (1 - b_c) + \alpha b_c.$$

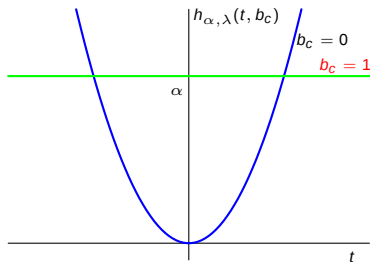


Eliminazione del processo di linea

È possibile trattare le variabili di linea b_c in maniera implicita:

$$E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} h_{\alpha, \lambda}(D_c^k \vec{x}, b_c),$$

$$h_{\alpha, \lambda}(t, b_c) = \lambda^2 t^2 (1 - b_c) + \alpha b_c.$$



Funzione energia duale

La **funzione energia duale** è definita come segue:

$$E_d(A, \vec{x}) = \inf_{\vec{b}} E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \vec{x}),$$

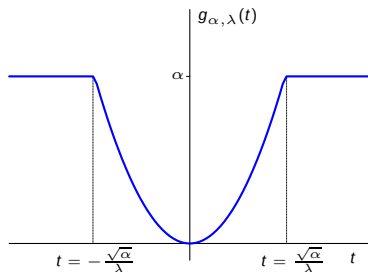


Funzione energia duale

La **funzione energia duale** è definita come segue:

$$E_d(A, \vec{x}) = \inf_{\vec{b}} E(A, \vec{x}, \vec{b}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \vec{x}),$$

$$g_{\alpha, \lambda}(t) = \min_{b_c \in \{0,1\}} h_{\alpha, \lambda}(t, b_c) = \begin{cases} \lambda^2 t^2 & \text{se } |t| < \frac{\sqrt{\alpha}}{\lambda}, \\ \alpha & \text{altrimenti.} \end{cases}$$



Algoritmi di minimizzazione

Rimane il problema di trovare quella $\vec{x}^*$ tale che:

$$\vec{x}^* = \arg \min_{\vec{x}} E_d(A, \vec{x}).$$



Algoritmi di minimizzazione

Rimane il problema di trovare quella $\vec{x}^*$ tale che:

$$\vec{x}^* = \arg \min_{\vec{x}} E_d(A, \vec{x}).$$

Cattive notizie

$$E_d(A, \vec{x}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \vec{x}).$$

Questo termine non è convesso ←



Algoritmi di minimizzazione

Rimane il problema di trovare quella $\vec{x}^*$ tale che:

$$\vec{x}^* = \arg \min_{\vec{x}} E_d(A, \vec{x}).$$

Cattive notizie

$$E_d(A, \vec{x}) = \|\vec{y} - A\vec{x}\|^2 + \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \vec{x}).$$

Questo termine non è convesso ←

Per risolvere il problema esistono sia algoritmi stocastici che deterministici.



Algoritmi deterministici

Una tecnica classica utilizzata per minimizzare la funzione energia duale è l'algoritmo GNC (*Graduated Non-Convexity*).

Si basa sulla costruzione di una famiglia di funzioni approssimanti di E_d

$$\left\{ E_d^{(p)} \right\}_p, \quad p \in [0, 2],$$

di cui la prima $E_d^{(2)}$ è convessa e l'ultima, $E_d^{(0)}$, coincide con E_d .



Algoritmi deterministici (2)

Algoritmo GNC

Require: $\vec{y}$ l'immagine osservata

$\vec{x} \leftarrow \vec{y}$

$p = 2$

while $p \neq 0$ **do** $\{E_d^{(p)} \neq E_d = E_d^{(0)}\}$

$\vec{x} \leftarrow \arg \left(\min E_d^{(p)} \text{ trovato partendo da } \vec{x} \right)$

$p \leftarrow p - r$ $\{r \text{ è un fattore di riduzione}\}$

end while

Il punto stazionario di ogni approssimante viene trovato utilizzando un metodo di discesa classico: NL-SOR (*Non-Linear Successive Over Relaxation*).



Algoritmi deterministici (2)

Algoritmo GNC

Require: $\vec{y}$ l'immagine osservata

$\vec{x} \leftarrow \vec{y}$

$p = 2$

while $p \neq 0$ **do** $\{E_d^{(p)} \neq E_d = E_d^{(0)}\}$

$\vec{x} \leftarrow \arg \left(\min E_d^{(p)} \text{ trovato partendo da } \vec{x} \right)$

$p \leftarrow p - r$ $\{r \text{ è un fattore di riduzione}\}$

end while

Il punto stazionario di ogni approssimante viene trovato utilizzando un metodo di discesa classico: NL-SOR (*Non-Linear Successive Over Relaxation*).

L'algoritmo ha un costo computazionale stimato come $O(k\lambda N^2)$, dove $O(\lambda N^2)$ è la complessità del NL-SOR.



Definizione del problema

Molto spesso le caratteristiche del rumore e della sfocatura sono sconosciute e il problema si complica:

Restauro cieco d'immagine

Trovare quella coppia $(A^*, \vec{x}^*)$ tale che:

$$(A^*, \vec{x}^*) = \arg \min_{(A, \vec{x})} E_d(A, \vec{x}).$$



Definizione del problema

Molto spesso le caratteristiche del rumore e della sfocatura sono sconosciute e il problema si complica:

Restauro cieco d'immagine

Trovare quella coppia $(A^*, \vec{x}^*)$ tale che:

$$(A^*, \vec{x}^*) = \arg \min_{(A, \vec{x})} E_d(A, \vec{x}).$$

Problema

La minimizzazione congiunta della funzione energia non è banale poichè questa è fortemente non lineare.



Un possibile approccio

Algoritmo di minimizzazione in successione

Require: A una stima dell'operatore di sfocatura

Require: $\vec{y}$ l'immagine osservata

$\vec{x} \leftarrow \vec{y}$

while non è stato trovato un punto fisso **do**

$\vec{x} \leftarrow \arg \min_{\vec{x}} E_d(A, \vec{x})$

$A \leftarrow \arg \min_A E_d(A, \vec{x})$

end while

Per valori di λ piccoli l'algoritmo ha un punto fisso banale vicino a $(I, \vec{y})$, dove I è la matrice identità.



Approccio proposto

La coppia $(A^*, \vec{x}^*)$ può essere trovata risolvendo il seguente problema di minimizzazione annidato:

$$A^* = \arg \min_A E_d(A, \vec{x}(A)),$$



Approccio proposto

La coppia $(A^*, \vec{x}^*)$ può essere trovata risolvendo il seguente problema di minimizzazione annidato:

$$A^* = \arg \min_A E_d(A, \vec{x}(A)),$$

dove

$$\vec{x}(A) = \arg \min_{\vec{x}} E_d(A, \vec{x}),$$

corrisponde al problema del restauro associato all'operatore A e



Approccio proposto

La coppia $(A^*, \vec{x}^*)$ può essere trovata risolvendo il seguente problema di minimizzazione annidato:

$$A^* = \arg \min_A E_d(A, \vec{x}(A)),$$

dove

$$\vec{x}(A) = \arg \min_{\vec{x}} E_d(A, \vec{x}),$$

corrisponde al problema del restauro associato all'operatore A e

$$\vec{x}^* = \vec{x}(A^*).$$



Approccio proposto

La coppia $(A^*, \vec{x}^*)$ può essere trovata risolvendo il seguente problema di minimizzazione annidato:

$$A^* = \arg \min_A E_d(A, \vec{x}(A)),$$

dove

$$\vec{x}(A) = \arg \min_{\vec{x}} E_d(A, \vec{x}),$$

corrisponde al problema del restauro associato all'operatore A e

$$\vec{x}^* = \vec{x}(A^*).$$

Per Trovare A^* si è implementato un algoritmo basato sul PSO (*Particle Swarm Optimization*).



Caratteristiche fondamentali

È un algoritmo evolutivo basato su una popolazione di individui che consente di risolvere diversi problemi di ottimizzazione.

Ogni particella ha le seguenti proprietà:

- ha una posizione $\vec{x}_i$ che rappresenta una possibile soluzione al problema
- è in grado di valutare la “bontà” della propria posizione tramite una funzione obiettivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$
- modifica la propria posizione in funzione della sua esperienza e di quella dei suoi vicini.



Aggiornamento della posizione

Indicando con $s \in \mathbb{N}^+$ la dimensione dello sciame, la posizione di ogni particella i , $i = 1, \dots, s$ è aggiornata aggiungendo una velocità alla posizione corrente:

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)}. \quad (1)$$



Aggiornamento della posizione

Indicando con $s \in \mathbb{N}^+$ la dimensione dello sciame, la posizione di ogni particella i , $i = 1, \dots, s$ è aggiornata aggiungendo una velocità alla posizione corrente:

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)}. \quad (1)$$

È $\vec{v}_i^{(t+1)}$ che guida il processo di ottimizzazione.



Aggiornamento della posizione

Indicando con $s \in \mathbb{N}^+$ la dimensione dello sciame, la posizione di ogni particella i , $i = 1, \dots, s$ è aggiornata aggiungendo una velocità alla posizione corrente:

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)}. \quad (1)$$

È $\vec{v}_i^{(t+1)}$ che guida il processo di ottimizzazione.

La miglior posizione visitata da i è data da:

$$\vec{z}_i^{(t+1)} = \begin{cases} \vec{z}_i^{(t)} & \text{se } f(\vec{x}_i^{(t+1)}) \geq f(\vec{z}_i^{(t)}), \\ \vec{x}_i^{(t+1)} & \text{se } f(\vec{x}_i^{(t+1)}) < f(\vec{z}_i^{(t)}). \end{cases}$$



Aggiornamento della posizione

Indicando con $s \in \mathbb{N}^+$ la dimensione dello sciame, la posizione di ogni particella i , $i = 1, \dots, s$ è aggiornata aggiungendo una velocità alla posizione corrente:

$$\vec{x}_i^{(t+1)} = \vec{x}_i^{(t)} + \vec{v}_i^{(t+1)}. \quad (1)$$

È $\vec{v}_i^{(t+1)}$ che guida il processo di ottimizzazione.

La miglior posizione visitata da i è data da:

$$\vec{z}_i^{(t+1)} = \begin{cases} \vec{z}_i^{(t)} & \text{se } f(\vec{x}_i^{(t+1)}) \geq f(\vec{z}_i^{(t)}), \\ \vec{x}_i^{(t+1)} & \text{se } f(\vec{x}_i^{(t+1)}) < f(\vec{z}_i^{(t)}). \end{cases}$$

La miglior posizione trovata dallo sciame è:

$$\hat{\mathbf{z}}^{(t)} = \arg \min_{\vec{x} \in \{\vec{z}_0, \dots, \vec{z}_s\}} f(\vec{x}).$$



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

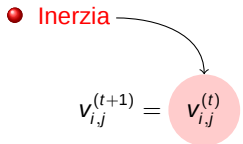
$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_{1j}^{(t)} [z_{i,j}^{(t)} - x_{i,j}^{(t)}] + c_2 r_{2j}^{(t)} [\hat{z}_j^{(t)} - x_{i,j}^{(t)}]. \quad (2)$$



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

● Inerzia


$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_{1j}^{(t)} [z_{i,j}^{(t)} - x_{i,j}^{(t)}] + c_2 r_{2j}^{(t)} [\hat{z}_j^{(t)} - x_{i,j}^{(t)}]. \quad (2)$$



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

- Inerzia

$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_{1,j}^{(t)} [z_{i,j}^{(t)} - x_{i,j}^{(t)}] + c_2 r_{2,j}^{(t)} [\hat{z}_j^{(t)} - x_{i,j}^{(t)}]. \quad (2)$$

- Componente cognitiva



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

- Inerzia

$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_{1,j}^{(t)} [z_{i,j}^{(t)} - x_{i,j}^{(t)}] + c_2 r_{2,j}^{(t)} [\hat{z}_j^{(t)} - x_{i,j}^{(t)}]. \quad (2)$$

- Componente cognitiva

- **Componente sociale**



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

- Inerzia

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 r_{1j}^{(t)} [z_{ij}^{(t)} - x_{ij}^{(t)}] + c_2 r_{2j}^{(t)} [\hat{z}_j^{(t)} - x_{ij}^{(t)}]. \quad (2)$$

- Componente cognitiva

- Componente sociale

$c_1, c_2 \in \mathbb{R}$ sono coefficienti di accelerazione



Aggiornamento della velocità

Il vettore velocità associato alla particella i è calcolato per ogni dimensione j , $j = 1, \dots, n$ come:

- Inerzia

$$v_{i,j}^{(t+1)} = v_{i,j}^{(t)} + c_1 r_{1,j}^{(t)} [z_{i,j}^{(t)} - x_{i,j}^{(t)}] + c_2 r_{2,j}^{(t)} [\hat{z}_j^{(t)} - x_{i,j}^{(t)}]. \quad (2)$$

- Componente cognitiva

- Componente sociale

$c_1, c_2 \in \mathbb{R}$ sono coefficienti di accelerazione

$r_{1,j}^{(t)}, r_{2,j}^{(t)} \sim U(0,1)$ sono sequenze casuali



Algoritmo

PSO originale

Require: inizializzazione di uno sciame n -dimensionale

repeat

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{\mathbf{z}})$ **then**

$\hat{\mathbf{z}} \leftarrow \vec{z}_i$

end if

end for

for all $i \in s$ **do**

 aggiorna la velocità di i con la (2)

 aggiorna la posizione di i con la (1)

end for

until non è verificata una condizione di uscita



Algoritmo

PSO originale

Require: **inizializzazione di uno sciame n -dimensionale**

repeat

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{z})$ **then**

$\hat{z} \leftarrow \vec{z}_i$

end if

end for

for all $i \in s$ **do**

 aggiorna la velocità di i con la (2)

 aggiorna la posizione di i con la (1)

end for

until non è verificata una condizione di uscita



Algoritmo

PSO originale

Require: inizializzazione di uno sciame n -dimensionale

repeat

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{z})$ **then**

$\hat{z} \leftarrow \vec{z}_i$

end if

end for

for all $i \in s$ **do**

 aggiorna la velocità di i con la (2)

 aggiorna la posizione di i con la (1)

end for

until non è verificata una condizione di uscita



Algoritmo

PSO originale

Require: inizializzazione di uno sciame n -dimensionale

repeat

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{z})$ **then**

$\hat{z} \leftarrow \vec{z}_i$

end if

end for

for all $i \in s$ **do**

 aggiorna la velocità di i con la (2)

 aggiorna la posizione di i con la (1)

end for

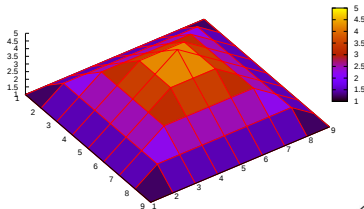
until non è verificata una condizione di uscita



Codifica della posizione

La posizione di ogni particella specifica un operatore di sfocatura rappresentato dalla rispettiva matrice M in forma lessicografica con $M \in \mathcal{S}^{(2h+1) \times (2h+1)}$, $h \in \mathbb{N}^+$, $\mathcal{S} = [0, \mu]$ e μ fissato.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 4 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 4 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$



Fitness

Funzione di fitness

La funzione obiettivo è la seguente:

$$f(A) = E_d(A, \vec{x}(A)),$$



Fitness

Funzione di fitness

La funzione obiettivo è la seguente:

$$f(A) = E_d(A, \vec{x}(A)),$$

dove $\vec{x}(A) = \arg \min_{\vec{x}} E_d(A, \vec{x})$ è approssimato dall'algoritmo GNC avviato con l'input $(A, \vec{y})$.



Fitness

Funzione di fitness

La funzione obiettivo è la seguente:

$$f(A) = E_d(A, \vec{x}(A)),$$

dove $\vec{x}(A) = \arg \min_{\vec{x}} E_d(A, \vec{x})$ è approssimato dall'algoritmo GNC avviato con l'input $(A, \vec{y})$.

Osservazione

La valutazione della funzione di fitness ha un costo proibitivo nelle immagini con un numero di pixel elevato.



Calcolare la funzione obiettivo velocemente

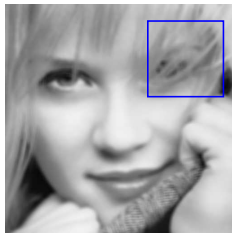
Per ridurre il costo computazionale della funzione di fitness si è scelto di:



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

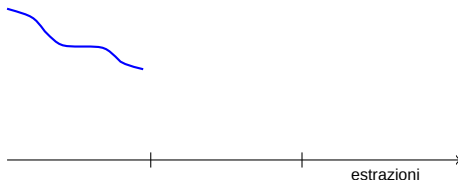
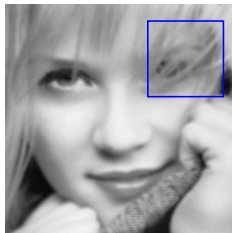
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

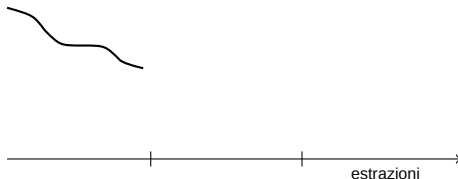
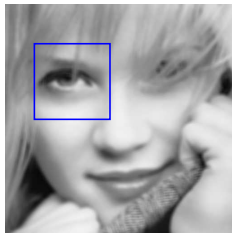
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

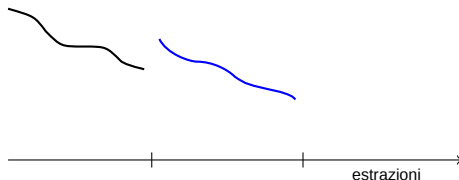
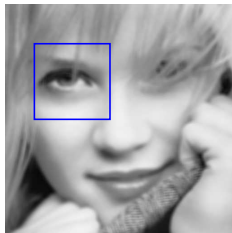
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata
- Ripetere il processo dopo un numero fissato ν di movimenti



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

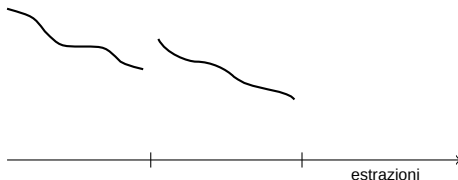
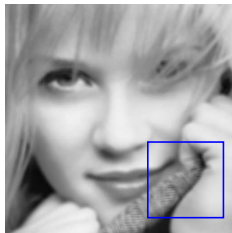
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata
- Ripetere il processo dopo un numero fissato ν di movimenti



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

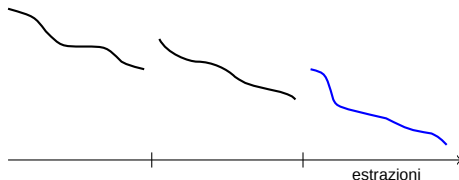
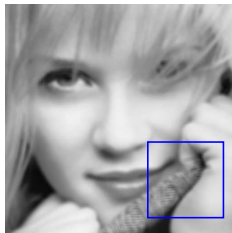
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata
- Ripetere il processo dopo un numero fissato ν di movimenti



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

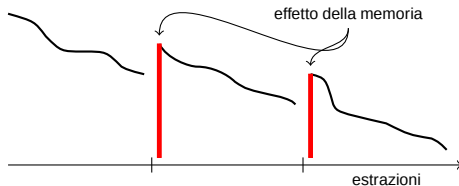
- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata
- Ripetere il processo dopo un numero fissato ν di movimenti



Calcolare la funzione obiettivo velocemente

Per ridurre il costo computazionale della funzione di fitness si è scelto di:

- Estrarre a caso una sottoimmagine $\hat{N} \times \hat{N}$ dai dati
- Fare evolvere lo sciame valutando la funzione di fitness solo sulla sottoarea considerata
- Ripetere il processo dopo un numero fissato ν di movimenti



Procedura di estrazione della sottoimmagine

Non tutte le sottoimmagini sono “significative” per il processo di ottimizzazione. Per fare in modo che ogni immagine estratta abbia la stessa importanza è necessario imporre alcune restrizioni sulla sua “regolarità”.

Una sottoimmagine $\hat{\mathbf{y}}, \hat{N} \times \hat{N}$ è ricavata a caso dai dati $\vec{\mathbf{y}}$ in modo tale che la sua regolarità $S(\hat{\mathbf{y}})$ cada in un intervallo prefissato $[l, u]$ dove



Procedura di estrazione della sottoimmagine

Non tutte le sottoimmagini sono “significant” per il processo di ottimizzazione. Per fare in modo che ogni immagine estratta abbia la stessa importanza è necessario imporre alcune restrizioni sulla sua “regolarità”.

Una sottoimmagine $\hat{\mathbf{y}}, \hat{N} \times \hat{N}$ è ricavata a caso dai dati $\vec{\mathbf{y}}$ in modo tale che la sua regolarità $S(\hat{\mathbf{y}})$ cada in un intervallo prefissato $[l, u]$ dove

$$S(\hat{\mathbf{y}}) = \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \hat{\mathbf{y}}).$$



Procedura di estrazione della sottoimmagine

Non tutte le sottoimmagini sono “significative” per il processo di ottimizzazione. Per fare in modo che ogni immagine estratta abbia la stessa importanza è necessario imporre alcune restrizioni sulla sua “regolarità”.

Una sottoimmagine $\hat{\mathbf{y}}, \hat{N} \times \hat{N}$ è ricavata a caso dai dati $\vec{\mathbf{y}}$ in modo tale che la sua regolarità $S(\hat{\mathbf{y}})$ cada in un intervallo prefissato $[l, u]$ dove

$$S(\hat{\mathbf{y}}) = \sum_{c \in C_k} g_{\alpha, \lambda}(D_c^k \hat{\mathbf{y}}).$$

Se i parametri l e u sono scelti in maniera corretta ogni sottoimmagine ottenuta con questa procedura sarà ugualmente significativa.



Fattore di costrizione

L'equazione di aggiornamento della velocità definisce la dimensione del movimento fatto dalle particelle in ogni istante temporale t . Se il passo è troppo grande può succedere che le particelle divergano e abbandonino lo spazio di ricerca: **swarm explosion**.



Fattore di costrizione

L'equazione di aggiornamento della velocità definisce la dimensione del movimento fatto dalle particelle in ogni istante temporale t . Se il passo è troppo grande può succedere che le particelle divergano e abbandonino lo spazio di ricerca: **swarm explosion**.

Soluzione: limitare le velocità con una costante moltiplicativa χ .

$$v_{i,j}^{(t+1)} = \chi \left[v_{i,j}^{(t)} + c_1 r_{1,j} \left(y_{i,j}^{(t)} - x_{i,j}^{(t)} \right) + c_2 r_{2,j} \left(\hat{y}_j^{(t)} - x_{i,j}^{(t)} \right) \right],$$



Fattore di costrizione

L'equazione di aggiornamento della velocità definisce la dimensione del movimento fatto dalle particelle in ogni istante temporale t . Se il passo è troppo grande può succedere che le particelle divergano e abbandonino lo spazio di ricerca: **swarm explosion**.

Soluzione: limitare le velocità con una costante moltiplicativa χ .

$$v_{i,j}^{(t+1)} = \chi \left[v_{i,j}^{(t)} + c_1 r_{1,j} \left(y_{i,j}^{(t)} - x_{i,j}^{(t)} \right) + c_2 r_{2,j} \left(\hat{y}_j^{(t)} - x_{i,j}^{(t)} \right) \right],$$

dove

$$\chi = \frac{2k}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|}, \quad \phi = c_1 + c_2.$$



Fattore di costrizione

L'equazione di aggiornamento della velocità definisce la dimensione del movimento fatto dalle particelle in ogni istante temporale t . Se il passo è troppo grande può succedere che le particelle divergano e abbandonino lo spazio di ricerca: **swarm explosion**.

Soluzione: limitare le velocità con una costante moltiplicativa χ .

$$v_{i,j}^{(t+1)} = \chi \left[v_{i,j}^{(t)} + c_1 r_{1,j} \left(y_{i,j}^{(t)} - x_{i,j}^{(t)} \right) + c_2 r_{2,j} \left(\hat{y}_j^{(t)} - x_{i,j}^{(t)} \right) \right],$$

dove

$$\chi = \frac{2k}{\left| 2 - \phi - \sqrt{\phi^2 - 4\phi} \right|}, \quad \phi = c_1 + c_2.$$

Se $\phi \geq 4$ e $k \in [0, 1]$ lo sciame converge verso un punto stabile.



Confinamento

Il fattore di costrizione non garantisce che la posizione di ogni particella in ogni dimensione j rimanga confinata nell'intervallo $[0, \mu]$.

Soluzione: Ad ogni movimento si sceglie a caso con probabilità $\frac{1}{2}$ uno tra due possibili metodi di confinamento.



Confinamento

Il fattore di costrizione non garantisce che la posizione di ogni particella in ogni dimensione j rimanga confinata nell'intervallo $[0, \mu]$.

Soluzione: Ad ogni movimento si sceglie a caso con probabilità $\frac{1}{2}$ uno tra due possibili metodi di confinamento.

Confinamento ibrido

```
for all  $i \in s$  do  
   $p \leftarrow$  pseudo-casuale  $\in [0, 1]$   
  if  $p \leq 0.5$  then  
    muovi  $i$  con il Confinamento iperbolico  
  else  
    muovi  $i$  con il Confinamento random back  
  end if  
end for
```



Confinamento iperbolico

Limitazione della velocità

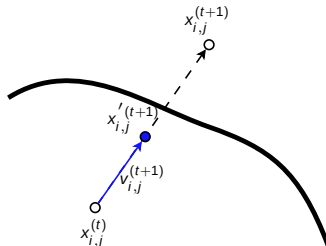
if $v_{i,j}^{(t+1)} > 0$ then

$$v_{i,j}^{(t+1)} \leftarrow \frac{v_{i,j}^{(t+1)}}{1 + \left| \frac{v_{i,j}^{(t+1)}}{\mu - x_{i,j}^{(t)}} \right|}$$

else

$$v_{i,j}^{(t+1)} \leftarrow \frac{v_{i,j}^{(t+1)}}{1 + \left| \frac{v_{i,j}^{(t+1)}}{x_{i,j}^{(t)}} \right|}$$

end if

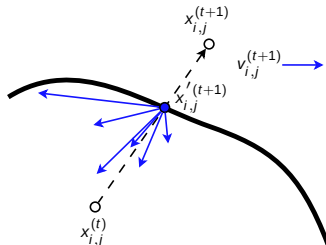


Confinamento random back

Vincolo sulla posizione

```

if  $x_{i,j}^{(t+1)} > \mu$  then
     $x_{i,j}^{(t+1)} \leftarrow \mu$ 
     $\{r_{3,j} \sim U(0, 1)\}$ 
     $v_{i,j}^{(t+1)} \leftarrow -r_{3,j} v_{i,j}^{(t+1)}$ 
end if
if  $x_{i,j}^{(t+1)} < 0$  then
     $x_{i,j}^{(t+1)} \leftarrow 0$ 
     $v_{i,j}^{(t+1)} \leftarrow -r_{3,j} v_{i,j}^{(t+1)}$ 
end if
  
```



Schema dell'algoritmo proposto

Algoritmo proposto

Require: $\vec{y}$ l'immagine osservata

repeat

 estrai una sottoimmagine $\hat{y}$ dai dati $\vec{y}$

 Inizializza lo sciame

for $n = 1, \dots, \nu$ **do**

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{z})$ **then**

$\hat{z} \leftarrow \vec{z}_i$

end if

end for

$\forall i$ aggiorna la velocità e la posizione con il Confinamento Ibrido

end for

until $\hat{z}$ non è la stessa per t estrazioni consecutive **or** il massimo numero di queste è stato raggiunto

Schema dell'algoritmo proposto

Algoritmo proposto

Require: $\vec{y}$ l'immagine osservata

repeat

 estrai una sottoimmagine $\hat{y}$ dai dati $\vec{y}$

 Inizializza lo sciame

for $n = 1, \dots, \nu$ **do**

for all $i \in s$ **do**

if $f(\vec{x}_i) < f(\vec{z}_i)$ **then**

$\vec{z}_i \leftarrow \vec{x}_i$

end if

if $f(\vec{z}_i) < f(\hat{z})$ **then**

$\hat{z} \leftarrow \vec{z}_i$

end if

end for

$\forall i$ aggiorna la velocità e la posizione con il Confinamento Ibrido

end for

until $\hat{z}$ non è la stessa per t estrazioni consecutive **or** il massimo numero di queste è stato raggiunto

Parametri

In ogni test sono stati usati i seguenti parametri:

- numero di particelle uguale a 30
- massimo elemento della matrice di sfocatura $\mu = 15$
- coefficienti di accelerazione $c_1 = 2.8$, $c_2 = 1.3$
- numero di movimenti per ogni sottoimmagine $\nu = 100$
- valori di α e λ scelti in maniera empirica caso per caso
- dimensione della sottoimmagine fissata caso per caso



Test 1: immagine stilizzata



Maschera di sfocatura:

$$M_1 = \begin{pmatrix} 2 & 1 & 3 & 1 & 2 \\ 4 & 2 & 6 & 2 & 4 \\ 8 & 4 & 12 & 4 & 8 \\ 4 & 2 & 6 & 2 & 4 \\ 2 & 1 & 3 & 1 & 2 \end{pmatrix}.$$

Rumore: $\sigma^2 = 0$.

Parametri: $\alpha = 400$, $\lambda = 2$.

Clique del primo ordine.

Figura: Immagine osservata 160×160 .

Date le dimensioni ridotte e la particolare struttura geometrica, si è lavorato sull'intera immagine.



Test 1: immagine osservata



Test 1: restauro non cieco



Test 1: restauro cieco



Test 2: immagine reale senza rumore

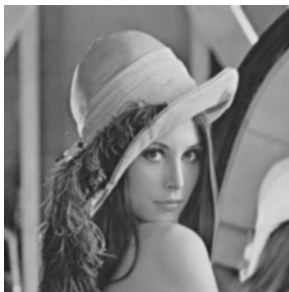


Figura: Immagine osservata 512×512 .

Maschera di sfocatura:

$$M_2 = \begin{pmatrix} 1 & 1 & 2 & 3 & 2 & 1 & 1 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 4 & 4 & 8 & 12 & 8 & 4 & 4 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 1 & 1 & 2 & 3 & 2 & 1 & 1 \end{pmatrix}.$$

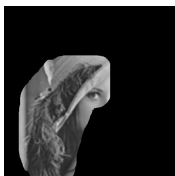
Rumore: $\sigma^2 = 0$.

Parametri: $\alpha = 4$, $\lambda = 2$, $\hat{N} = 100$.

Clique del secondo ordine.



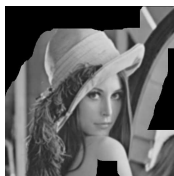
Test 2: scelta di l e u



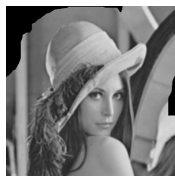
$[60000, +\infty[$



$[55000, +\infty[$



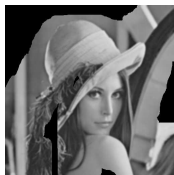
$[50000, +\infty[$



$[45000, +\infty[$



$[45000, 52000]$



$[48000, 54000]$



$[50000, 52000]$



$[51000, 52000]$

Figura: Parti della foto Test 2 coperte dalla procedura di estrazione della sottoimmagine con diversi valori di l e u .



Test 2: immagine osservata



Test 2: restauro non cieco



Test 2: restauro cieco



Test 2: nuova codifica della posizione

Si è cercato di migliorare il risultato del restauro cieco facendo delle ipotesi sulla geometria della maschera di sfocatura:

$$M_2 = \begin{pmatrix} 1 & 1 & 2 & 3 & 2 & 1 & 1 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 4 & 4 & 8 & 12 & 8 & 4 & 4 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 1 & 1 & 2 & 3 & 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} (1 \ 1 \ 2 \ 3 \ 2 \ 1 \ 1).$$



Test 2: nuova codifica della posizione

Si è cercato di migliorare il risultato del restauro cieco facendo delle ipotesi sulla geometria della maschera di sfocatura:

$$M_2 = \begin{pmatrix} 1 & 1 & 2 & 3 & 2 & 1 & 1 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 4 & 4 & 8 & 12 & 8 & 4 & 4 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 1 & 1 & 2 & 3 & 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} (\textcolor{red}{1} \textcolor{red}{1} \textcolor{red}{2} \textcolor{red}{3} 2 1 1).$$

I due vettori sono simmetrici pertanto l'informazione significativa è contenuta solo nelle prime $h + 1$ componenti.



Test 2: nuova codifica della posizione

Si è cercato di migliorare il risultato del restauro cieco facendo delle ipotesi sulla geometria della maschera di sfocatura:

$$M_2 = \begin{pmatrix} 1 & 1 & 2 & 3 & 2 & 1 & 1 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 4 & 4 & 8 & 12 & 8 & 4 & 4 \\ 3 & 3 & 6 & 9 & 6 & 3 & 3 \\ 2 & 2 & 4 & 6 & 4 & 2 & 2 \\ 1 & 1 & 2 & 3 & 2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} (1 \ 1 \ 2 \ 3 \ 2 \ 1 \ 1).$$

I due vettori sono simmetrici pertanto l'informazione significativa è contenuta solo nelle prime $h + 1$ componenti.

Lo spazio di ricerca è stato ridotto codificando la posizione con due vettori di questo tipo inizializzati con valori a caso nell'intervallo $[0, \mu]$.



Test 2: restauro cieco (nuova codifica)



Test 3: immagine reale senza rumore



Figura: Immagine osservata 512×512 .

Maschera di sfocatura:

$$M_3 = \begin{pmatrix} 1 & 2 & 3 & 3 & 3 & 2 & 1 \\ 2 & 4 & 6 & 6 & 6 & 4 & 2 \\ 3 & 6 & 9 & 9 & 9 & 6 & 3 \\ 3 & 6 & 9 & 9 & 9 & 6 & 3 \\ 3 & 6 & 9 & 9 & 9 & 6 & 3 \\ 2 & 4 & 6 & 6 & 6 & 4 & 2 \\ 1 & 2 & 3 & 3 & 3 & 2 & 1 \end{pmatrix}.$$

Rumore: $\sigma^2 = 0$.

Parametri: $\alpha = 4$, $\lambda = 0.5$, $\hat{N} = 100$.

Clique del secondo ordine.



Test 3: scelta di l e u

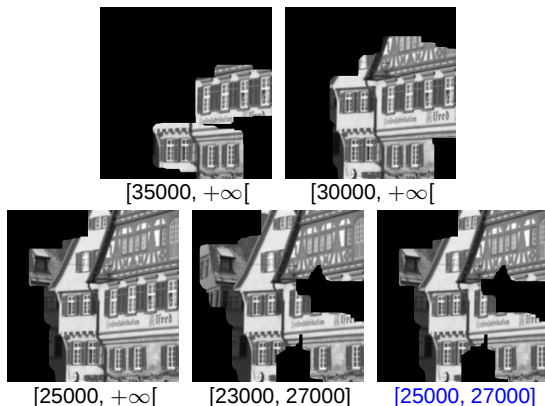


Figura: Parti della foto Test 3 coperte dalla procedura di estrazione della sottoimmagine con diversi valori di l e u .



Test 3: immagine osservata



Test 3: restauro non cieco



Test 3: restauro cieco



Test 4: immagine reale con rumore

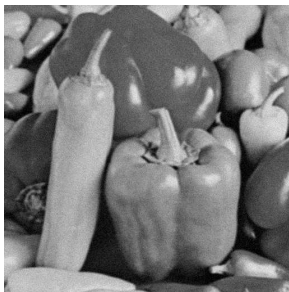


Figura: Immagine osservata 512×512 .

Maschera di sfocatura:

$$M_4 = \begin{pmatrix} 2 & 3 & 4 & 3 & 2 \\ 4 & 6 & 8 & 6 & 4 \\ 6 & 9 & 12 & 9 & 6 \\ 4 & 6 & 8 & 6 & 4 \\ 2 & 3 & 4 & 3 & 2 \end{pmatrix}.$$

Rumore: $\sigma^2 = 100$.

Parametri: $\alpha = 625$, $\lambda = 8$, $\hat{N} = 50$.

Clique del secondo ordine.



Test 4: scelta di l e u (valori $\times 1000$)



[2750, 2800]



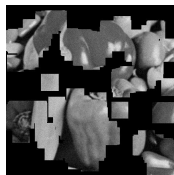
[2750, 2790]



[2760, 2780]



[2760, 2785]

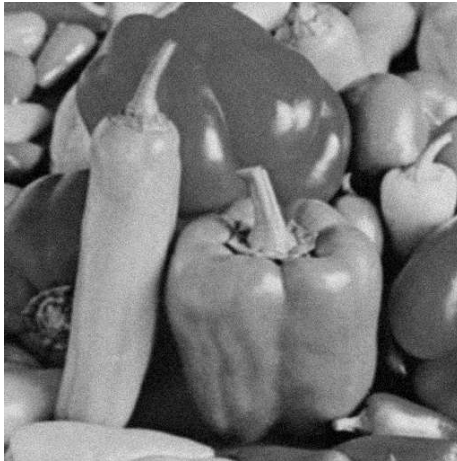


[2760, 2783]

Figura: Parti della foto Test 4 coperte dalla procedura di estrazione della sottoimmagine con diversi valori di l e u .



Test 4: immagine osservata



Test 4: restauro non cieco



Test 4: restauro cieco



Confronto numerico della qualità dei risultati

Risultati soddisfacenti sono stati ottenuti in termini di qualità delle immagini ottenute.

In tabella è riportato il valore dell'errore quadratico medio calcolato tra il risultato del restauro non cieco e l'immagine ideale (RNC) e tra quest'ultima e quella ottenuta con l'algoritmo proposto (RC).

Test	σ^2	dimensione	maschera	RNC	RC
1	0	160×160	5×5	0.7825	0.8246
2	0	512×512	7×7	39.9972	40.6230
3	0	512×512	7×7	318.2831	342.4217
4	100	512×512	5×5	46.4230	46.5552



Fine.

