

Pseudo-Code 1/2

Algorithm 1 Hybrid optimization routine

E	Energy function to be minimized
γ	Step size of the gradient
L_{\max}^D	Number of iterations (Gradient)
r	Radius of exploration, real $\in]0, 1[$
T_0	Initial Temperature
T_f	Final Temperature
L_{\max}^S	Number of iterations (Metropolis)

1. Deterministic Search

CONJUGATE_GRADIENT_ $(E; \gamma, L_{\max}^D)$ [Algo. 2]
(with $\gamma \leftarrow \gamma/2$ if E increases between 2 iterations)

2. Stochastic Local Exploration

METROPOLIS_EXPLOR $(E; r, T_0, T_f, L_{\max}^S)$ [Algo. 3]

Algorithm 2 Conjugate gradient

E	Energy function to be minimized
∇E	Gradient vector of E
I	Image with pixels in lexicographic order as one column-vector
$I_s^{[l]}$	Image vector at site s and iteration l
$\langle \cdot, \cdot \rangle$	Scalar product
l	Iteration step

γ	Step size of the gradient
L_{\max}^D	Maximal number of iterations

Repeat

if $l = 0$ **then**

$$d^{[l]} = -\nabla E(I^{[l]})$$

$$= 4 \sum_{s,t \substack{s \neq t \\ t \in \mathcal{N}_s^3}} [I_s^{[l]} - I_t^{[l]}] (\beta_{[1]s,t}^2 - [I_s^{[l]} - I_t^{[l]}]^2)$$

else

$$b^{[l-1]} = \frac{\langle \nabla E(I^{[l]}) - \nabla E(I^{[l-1]}), \nabla E(I^{[l]}) \rangle}{\langle \nabla E(I^{[l-1]}), \nabla E(I^{[l-1]}) \rangle}$$

$$d^{[l]} = -\nabla E(I^{[l]}) + b^{[l-1]} d^{[l-1]}$$

$$I^{[l+1]} = I^{[l]} + \gamma d^{[l]}$$

$$l = l + 1$$

Until $l < L_{\max}^D$ **or** $\frac{(E^{[l]} - E^{[l-1]})^2}{(E^{[l]})^2} \approx 0;$

Algorithm 3 Local exploration with Metropolis

E	Energy function to be minimized
T_l	Temperature at Iteration step l
a	Cooling schedule parameter
r	Radius of exploration, real $\in]0, 1[$
T_0	Initial Temperature
T_f	Final Temperature
L_{\max}^S	Maximal number of iterations

1. Initialization

$$a \leftarrow \left(\frac{T_f}{T_0} \right)^{\frac{1}{L_{\max}^S}}$$

2. Local Exploration

while $l < L_{\max}^S$ **do**

for each pixel with value x_s at site s do

- Compute $\Delta \text{Energy} = E(y_s) - E(x_s)$ with $y_s \in [x_s - r : x_s + r]$ and $y_s \in [0.0 : 1.0]$
- **If** ($\Delta \text{Energy} < 0$) Replace x_s by y_s
- **Else** Replace x_s by y_s with probability $\triangleright p = \exp\left(-\frac{\Delta \text{Energy}}{T_l}\right)$

3. $l \leftarrow l + 1$ and $T_l \leftarrow T_0 a^l$

Algorithm 4 Optimization-Based Edge Specification Algorithm OBEHS Algorithm

E	Energy function to be minimized
$D_{\mathcal{B}}$	Bhattacharya distance between the target and output edge histogram
l	Iteration step

Stopping Criteria

E_{\min}	Threshold energy value
$D_{\mathcal{B}_{\max}}$	Threshold Bhattacharya distance
L_{\max}^H	Maximal number of iterations

while $l < L_{\max}^H$ **or** $E > E_{\min}$ **or** $D_{\mathcal{B}} > D_{\mathcal{B}_{\max}}$ **do**

1. Specification

ALGORITHM B, C OR D

2. Optimization

HYBRID_OPTIMIZATION_ (E) [Algo. 1]

Pseudo-Code 2/2

[Algorithm A]:

- Order pixels: $I(x_1, y_1) \prec I(x_2, y_2) \prec \dots \prec I(x_N, y_M)$
- Split this pixel ordering relation from left to right in L groups, such as group j has h_j pixels.
- For all the pixels in a group j , assign gray-level j .

[Algorithm B]:

- **1. Ordering relation**
 - Normalize H in order that $W = \sum_{k=0}^{Z-1} h_k$
 - Order the $W = 8 \cdot N \cdot M$ pairwise pixel absolute differences:

$$|I_s - I_t| \prec |I_u - I_v| \prec \dots \prec |I_x - I_y|$$
 - Split this pixel absolute difference ordering relation from left to right in Z groups, such as group j has h_j elements, i.e., h_j couples of pixels.
 - For all pairs of pixels or pair of sites (s, t) whose absolute difference is in a group j , assign $\beta_{[1] s, t} = j$.

- **2. Optimize:**

$$\hat{I} = \arg \min_I \sum_{\langle s, t \rangle_{s \neq t} \in \mathcal{N}_s^1} \left(\beta_{[1] s, t}^2 - (I_s - I_t)^2 \right)^2$$

[Algorithm C]:

- **1. Specification with ordering relation**
 - For $l = 1$ to n_t
 - * $W_{[l]} = 8 \cdot l \cdot N \cdot M$
 - * Normalize $H_{[l]}$ such that $W_{[l]} = \sum_{k=0}^{Z-1} h_{[l], k}$
 - * Order the $W_{[l]}$ pairwise pixel absolute differences:

$$|I_s - I_t| \prec |I_u - I_v| \prec \dots \prec |I_x - I_y|$$
 - * Split this pixel absolute difference ordering relation from left to right in Z groups, such as group j has $h_{[l], j}$ elements, i.e., $h_{[l], j}$ couples of pixels.
 - * For all pairs of pixels or pair of sites (s, t) whose absolute difference is in a group j , assign $\beta_{[l] s, t} = j$.

- **2. Optimize:**

$$\hat{I} = \arg \min_I \sum_{\langle s, t \rangle_{s \neq t} \in \mathcal{N}_s^1} \left(\beta_{[1] s, t}^2 - (I_s - I_t)^2 \right)^2$$

$$\text{for } t \in \mathcal{N}_s^1 \cup \mathcal{N}_s^2 \cup \dots \cup \mathcal{N}_s^l$$

[Algorithm D]:

It simply consists in alternating Algorithm C and Algorithm A until a stability criterion is reached (i.e., the output image does not change too much between two iterations).

In the case where the input image is specified from a target image for which we want to keep its color palette, there are two different ways:

- 1) either the histogram of the components L , A and B of the input image is specified (Algorithm A) from the components L , A and B of the target image
- 2) or, one has to define a strict ordering relation among color image pixels and a possible solution is to use the luminance or the gray value for that. In this case, the color histogram specification procedure proceeds as follows:
 - Order color pixels of the input image (I) from their luminance or grey value:

$$I(x_1, y_1) \prec I(x_2, y_2) \prec \dots \prec I(x_N, y_M)$$
 - Order color pixels of the target image (T) from their luminance or grey value:

$$T(u_1, v_1) \prec T(u_2, v_2) \prec \dots \prec T(u_N, v_M)$$
 Note that if the size of the target image is different from the size of the input image, an up-sampling or a sub-sampling procedure should be used.
 - Assign to $I(x_i, y_j)$ the color value $T(u_i, v_j)$ for all $i < N$ and $j < M$.