



HAL
open science

Fast Ordering Algorithm for Exact Histogram Specification

Mila Nikolova, Gabriele Steidl

► **To cite this version:**

Mila Nikolova, Gabriele Steidl. Fast Ordering Algorithm for Exact Histogram Specification. 2013.
hal-00870501

HAL Id: hal-00870501

<https://hal.science/hal-00870501>

Preprint submitted on 7 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast Ordering Algorithm for Exact Histogram Specification

Mila Nikolova, and Gabriele Steidl

Abstract

This paper provides a fast algorithm to order in a strict way the integer gray values in digital images which can be applied for exact histogram specification. Our algorithm is based on the ordering procedure by the variational approach in [9]. This variational method was shown to be superior to other state-of-the art ordering algorithms in terms of faithful total strict ordering but not in speed. Indeed, the relevant functionals are in general difficult to minimize because their gradient is nearly flat over vast regions.

In this paper we propose a simple and fast fixed point algorithm to minimize the functionals. The fast convergence of our algorithm results from known analytical properties of the model. In particular the original image is a good starting point for the iterations and the involved inverse functions admit a simple explicit form. Only a few iteration steps of this algorithm provide an image whose pixels can be ordered in a strict and faithful way. Numerical experiments confirm that our algorithm outperforms by far its main competitors in speed and quality. Moreover, in contrast to other ordering algorithms we can handle large images commonly taken by commercial cameras.

Concerning applications the proposed ordering algorithm is the basis of the hue and range preserving color image enhancement method proposed in [11].

I. INTRODUCTION

Histogram processing is a technique with numerous applications, e.g., in invisible watermarking, image normalization and enhancement, object recognition [3], [5], [11], [13]. The goal of exact histogram specification (HS) is to transform an input image into an output image having a prescribed histogram. For a uniform target histogram we speak about histogram equalization (HE).

Consider digital (i.e. quantized) $M \times N$ images f with L gray values $\mathcal{Q} := \{q_1, \dots, q_L\}$. For 8-bit images we have $L = 256$ and $\mathcal{Q} = \{0, \dots, 255\}$. We reorder the image columnwise into a vector of size $n := MN$ and address the pixels by the index set $\mathbb{I}_n := \{1, \dots, n\}$. The histogram of f , denoted by h_f , is given by $h_f[q_k] = \# \{i \in \mathbb{I}_n \mid f[i] = q_k\}$, $k = 1, \dots, L$, where $\#$ stands for cardinality.

In theory, histogram specification uses the relation between the cumulative density function of an arbitrary distributed continuous random variable and a uniformly distributed one, see [5]. However, for digital images we are confronted with a large number of n discrete random variables taking only L possible values (i.e., $n \gg L$).

M. Nikolova is with the CMLA, ENS Cachan, CNRS, PRES UniverSud. 61 av. du President Wilson, 94235 Cachan, France (email: nikolova@cmla.ens-cachan.fr).

G. Steidl is with the Dept. of Mathematics, University of Kaiserslautern, 67663 Kaiserslautern, Germany (email: steidl@mathematik.uni-kl.de).

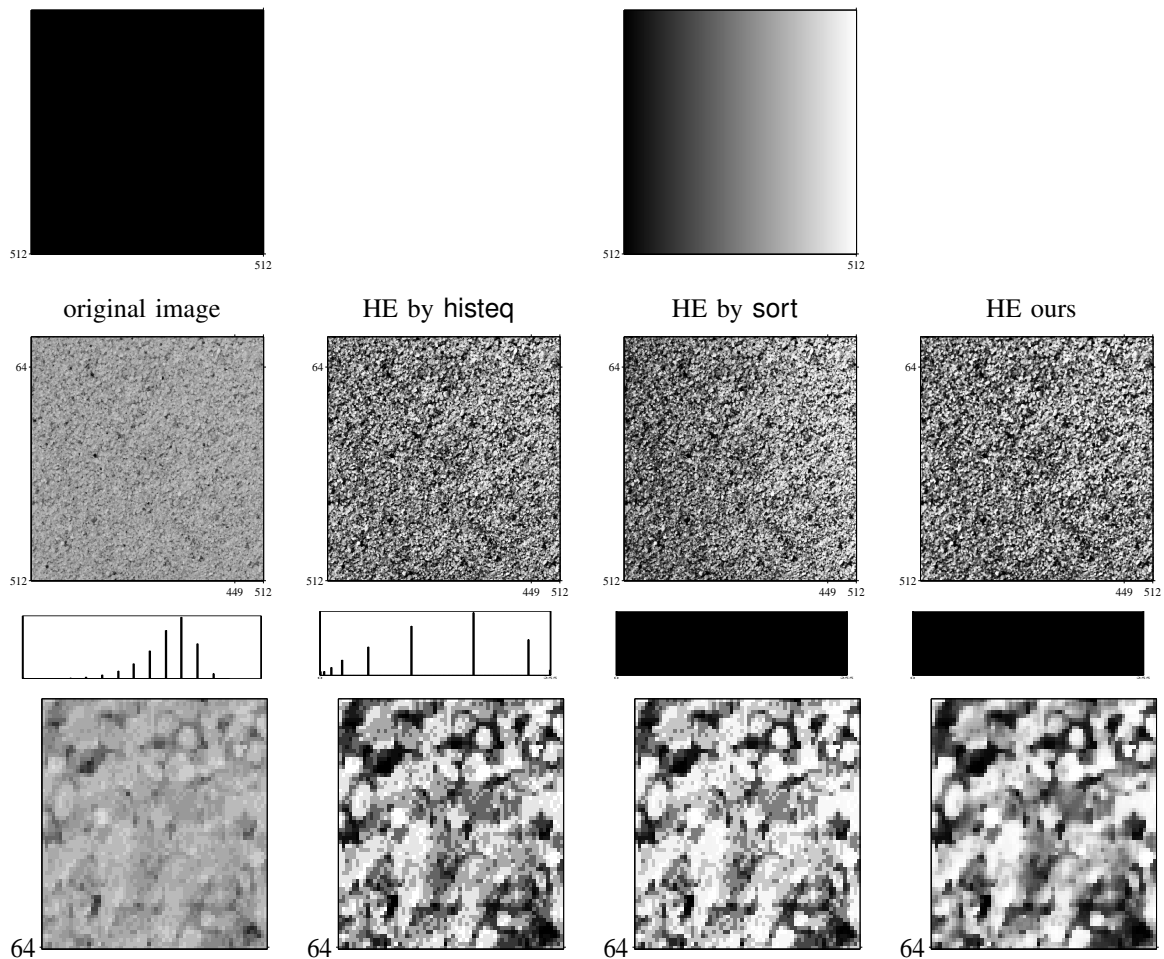


Fig. 1. Illustration of the importance of ordering for histogram equalization (HE). First row: The pixels of a completely black image (left) are strictly ordered using the Matlab routine `sort` which sorts equal pixels columnwise ascending. The resultant HE image (right) is perfectly equalized and becomes lighter from left to right. Second row: The original image 'sand' and different equalizations, namely Matlab `histeq`, Matlab `sort` and perfect ordering, our sorting algorithm. The third image still has the lighting effects from left to right. Third row: The corresponding histograms. Fourth row: Zooms of these images (rows from 1 to 64 and columns from 449 to 512). It can be seen that the texture generated by our HE algorithm is more regular than the other ones.

Then the target histogram can almost never be satisfied exactly. Histogram specification is an ill-posed problem for digital images. The Matlab function `histeq` is expected to produce HE; see Fig. 1 and the comments in the caption. Additional effort should be paid to improve the specification as e.g. in [1], where the appearing gaps in the target histogram of the HE were circumvented by so-called “probability smoothing”.

In this paper we focus on *exact* histogram specification to a target histogram $\hat{h} = (\hat{h}_1, \dots, \hat{h}_L)$ for the gray values $\mathcal{P} = \{0, \dots, L-1\}$. If the pixels values of our image are pairwise different so that they can be strictly ascending ordered, exact histogram specification can be easily done by dividing the corresponding ordered list of indices into L groups and assigning gray value 0 to the first \hat{h}_1 pixels, gray value 1 to the second \hat{h}_2 pixels and so on until gray value $L-1$ to the last \hat{h}_L pixels. However, this simple procedure requires a *meaningful* strict ordering of all pixels in the input image. Fig. 1 demonstrates the importance of ordering for histogram equalization.

Research on ordering problems has been conducted for four decades already [6]. State-of-the-art methods are

- the local mean ordering (LM) of Coltuc, Bolon and Chassery [4],
- the wavelet-based ordering (WA) of Wan and Shi [14], and
- the special variational approach (VA) of Nikolova, Wen and Chan [9].

The first two methods extract for any pixel $f[i]$ in the input image K auxiliary informations, say $\kappa_k[i]$, $k \in \mathbb{I}_K$, where $\kappa_1 := f$. Then an ascending order “ \prec ” for all pixels could ideally be obtained using the rule

$$i \prec j \quad \text{if} \quad \kappa_s[i] < \kappa_s[j] \text{ for some } s \in \mathbb{I}_K \quad \text{and} \quad \kappa_k[i] = \kappa_k[j] \text{ for all } 1 \leq k < s.$$

The third method uses an iterative procedure to find the minimizer of a certain functional related to f which components can be ordered in a strict way. The numerical results in [9] have shown that VA clearly outperforms LM and WA in terms of quality of the ordering and memory requirements, but not in speed. The *main contribution of this paper* consists in providing a simple fixed point minimization algorithm that attains the minimizer with remarkable speed and precision. Convergence and parameter selection are discussed. In particular, only few iterations are sufficient to provide the information needed for a meaningful strict ordering of the pixels. Numerical tests confirm that the VA method along with the new fixed point algorithm outperforms by far all other relevant ordering methods in terms of quality and speed. In particular our algorithm can easily handle large size “mega-pixel” images where other algorithms as LM and WA run into storage problems without pertinent sorting strategies. Preliminary results of this paper were published in the conference paper [10].

As already pointed out, one can design fast HS methods based on our ordering algorithm. The present paper thus provides the background for HS based methods for hue and range preserving color enhancement developed in our subsequent paper [11].

The outline of the paper is as follows: In Section II we review the specialised variational approach in [9] and some of its properties proven in [2]. Then, in Section III, we propose a simple fixed point algorithm to find a minimizer of the our model which is indeed fast for two reasons: first we know a good starting point for the iterations, second the appearing inverse functions are given explicitly and need simple operations. Section IV contains numerical examples. We compare speed and accuracy in the sense of a faithful total strict ordering of our algorithm with state-of-the art algorithms and provide a histogram equalization inversion comparison. We will see that only few iterations are necessary to obtain promising ordering results.

II. THE MINIMIZERS OF A FULLY SMOOTHED ℓ_1 —TV MODEL

Consider a functional of the form

$$J(u, f) := \Psi(u, f) + \beta\Phi(u), \quad \beta > 0 \tag{1}$$

with

$$\begin{aligned} \Psi(u, f) &:= \sum_{i \in \mathbb{I}_n} \psi(u[i] - f[i]), \\ \Phi(u) &:= \sum_{j \in \mathbb{I}_r} \varphi(g_j u), \end{aligned} \tag{2}$$

where $g_i \in \mathbb{R}^{1 \times n}$, $i \in \mathbb{I}_r$ correspond to vertical and horizontal differences, i.e.,

$$\begin{aligned} g_i[i] &= -1, & g_i[i+1] &= 1 & \text{and} & g_i[k] &= 0 & \forall k \in \mathbb{I}_n \setminus \{i, i+1\}, \\ g_j[j] &= -1, & g_j[j+M] &= 1 & \text{and} & g_j[k] &= 0 & \forall k \in \mathbb{I}_n \setminus \{j, j+M\}. \end{aligned} \quad (3)$$

In this case, the number r in the definition of Φ is $r = 2MN - (M + N)$. One can also add diagonal differences¹ to improve the rotation invariance of $\Phi(u)$. The experiments in [9] have shown that the simpler case when only the differences in (3) are used is enough to enable the minimizers of J to give rise to a prompt sorting. In all cases Neumann boundary conditions are adopted. The functions $\psi(\cdot) := \psi(\cdot, \alpha_1) : \mathbb{R} \rightarrow \mathbb{R}$ and $\varphi(\cdot) := \varphi(\cdot, \alpha_2) : \mathbb{R} \rightarrow \mathbb{R}$ depend on two parameters $\alpha_1 > 0$ and $\alpha_2 > 0$, respectively. When necessary, we also shall use the notation $\psi(\cdot, \alpha_1)$ and $\varphi(\cdot, \alpha_2)$. The parameters α_1 and α_2 shall be omitted when they are not explicitly involved in our derivations.

Following [9], the hint for achieving strict ordering is that the functions ψ and φ belong to a *family of functions* $\theta(\cdot, \alpha) : \mathbb{R} \rightarrow \mathbb{R}$, $\alpha > 0$, satisfying the requirements in assumptions H1 and H2 described next. The rationale for these choices is extensively discussed in [9]. For simplicity, we denote $\theta'(t, \alpha) := \frac{d}{dt}\theta(t, \alpha)$ and $\theta''(t, \alpha) := \frac{d^2}{dt^2}\theta(t, \alpha)$.

H1: For any fixed $\alpha > 0$ the function $t \mapsto \theta(t, \alpha)$ is in $C^s(\mathbb{R}^n)$, $s \geq 2$ and even, i.e., $\theta(-t, \alpha) = \theta(t, \alpha)$ for all $t \in \mathbb{R}$. Its derivative $\theta'(t, \alpha)$ is strictly increasing with $\lim_{t \rightarrow \infty} \theta'(t, \alpha) = 1^2$ and its second derivative $\theta''(t, \alpha)$ is decreasing on $[0, +\infty)$.

H2: For fixed $t > 0$, the function $\alpha \mapsto \theta(t, \alpha)$ is strictly decreasing on $(0, +\infty)$ with

$$\lim_{\alpha \rightarrow 0} \theta'(t, \alpha) = 1 \quad \text{and} \quad \lim_{\alpha \rightarrow \infty} \theta'(t, \alpha) = 0 .$$

From H1 we see that $\theta'(t, \alpha)$ is odd and has an inverse function

$$\xi(t, \alpha) := (\theta')^{-1}(t, \alpha)$$

which is also odd and strictly increasing.

Under these assumptions, the functional $J(\cdot, f)$ amounts to a fully smoothed ℓ_1 -TV model. There are many possible choices of functions θ meeting H1 and H2, see [9]. In our numerical examples we will focus on the functions θ given in Table I.

Since $J(\cdot, f)$ is a strictly convex, coercive functional it has a unique minimizer $\hat{u} \in \mathbb{R}^n$. The following theorems collect properties of this minimizer which are important for our faithful and fast sorting algorithm. The first theorem which proof can be found in [9, Theorem 1] guaranties that the entries of the minimizer differ in general pairwise from each other so that \hat{u} provides an auxiliary information for ordering the pixels of f .

Theorem 1: (Strict ordering information) Let ψ and φ fulfill H1 and H2. Then there exists a *dense open subset* \mathbb{K}^n of \mathbb{R}^n such that for any $f \in \mathbb{K}^n$ the minimizer \hat{u} of $J(\cdot, f)$ satisfies

$$\begin{aligned} \hat{u}[i] &\neq \hat{u}[j], & \forall i, j \in \mathbb{I}_n, & \quad i \neq j, \\ \hat{u}[i] &\neq f[i], & \forall i \in \mathbb{I}_n. \end{aligned} \quad (4)$$

¹Diagonal differences are given by

$$\begin{aligned} g_i[i] &= -1, & g_i[i+M-1] &= 1 & \text{and} & g_i[k] &= 0 & \forall k \in \mathbb{I}_n \setminus \{i, i+M-1\}, \\ g_j[j] &= -1, & g_j[j+M+1] &= 1 & \text{and} & g_j[k] &= 0 & \forall k \in \mathbb{I}_n \setminus \{j, j+M+1\}. \end{aligned}$$

²The upper bound on θ' is set to 1 just for definiteness.

	θ	θ'	θ''	$\xi = (\theta')^{-1}$	ξ'
$\theta 1$	$\sqrt{t^2 + \alpha}$	$\frac{t}{\sqrt{t^2 + \alpha}}$	$\frac{\alpha}{(\sqrt{t^2 + \alpha})^3}$	$t\sqrt{\frac{\alpha}{1-t^2}}$	$\frac{\sqrt{\alpha}}{(\sqrt{1-t^2})^3}$
$\theta 2$	$ t - \alpha \log\left(1 + \frac{ t }{\alpha}\right)$	$\frac{t}{\alpha + t }$	$\frac{\alpha}{(\alpha + t)^2}$	$\frac{\alpha t}{1 - t }$	$\frac{\alpha}{1 - t }$

TABLE I

CHOICES FOR $\theta(\cdot, \alpha)$ TOGETHER WITH THE USED DERIVATIVES AND INVERSE FUNCTIONS.

The fact that \mathbb{K}^n is dense and open in \mathbb{R}^n means that the property in (4) is generically true. This result is much stronger than saying that (4) holds true almost everywhere on \mathbb{R}^n .³ Therefore, the elements of $\mathbb{R}^n \setminus \mathbb{K}^n$ are highly exceptional in \mathbb{R}^n .

The second theorem provides an estimate of $\|f - \hat{u}\|_\infty$ which was proved by the authors in [2, Theorems 1 and 2]. It shows in particular that for our parameter settings in the numerical part we can guarantee that $|f[i] - \hat{u}[i]| < 0.5$, for any $i \in \mathbb{I}_n$. Consequently, if for $f[i] \in \{0, \dots, 255\}$, $i \in \mathbb{I}_n$, the relation $f[i] < f[j]$ holds true, then also $\hat{u}[i] < \hat{u}[j]$ such that the initial ordering of pairwise different pixels is preserved.

Theorem 2: (Distance of \hat{u} from f)

Let ψ and φ fulfill H1 and H2. Then, for $\beta < \frac{1}{4}$, the minimizer \hat{u} of $J(\cdot, f)$ satisfies

$$\|\hat{u} - f\|_\infty \leq (\psi')^{-1}(4\beta, \alpha_1) = \xi(4\beta, \alpha_1), \quad (5)$$

where $\xi := (\psi')^{-1}$. Further,

$$\|\hat{u} - f\|_\infty \nearrow \xi(4\beta, \alpha_1) \quad \text{as} \quad \alpha_2 \searrow 0$$

under the assumption that $\nu_f := \max_{i \in \mathcal{I}} \min_{j \in \mathcal{N}_i} (|f[i] - f[j]|) > 2\xi(4\beta, \alpha_1)$, where

$\mathcal{I} := \{i \in \text{int } \mathbb{I}_n : \text{sign}(f[i] - f[j]) = \sigma, \forall j \in \mathcal{N}_i \text{ with } \sigma \in \{-1, +1\}\}$, and $\nu_f := 0$ if $\mathcal{I} = \emptyset$.

The above assumption on ν_f is quite realistic for natural images with 8 bit gray values. For the collection of natural images considered in [2] we have for example $\text{mean}(\nu_f) \approx 224$. In our numerical examples we will use the parameters $\beta = 0.1$ and $\alpha_1 = \alpha_2 = 0.05$. For these values we obtain by (5) the following upper estimates:

$$\begin{aligned} \psi = \varphi = \theta 1 : \quad & \|\hat{u} - f\|_\infty \leq 0.0976, \\ \psi = \varphi = \theta 2 : \quad & \|\hat{u} - f\|_\infty \leq 0.0\bar{3}. \end{aligned}$$

III. FAST MINIMIZATION AND SORTING ALGORITHM

The function \hat{u} is a minimizer of $J(\cdot, f)$ in (1) if and only if $\nabla J(\hat{u}, f) = 0$. This is equivalent to $\nabla \Psi(\hat{u}, f) = -\beta \nabla \Phi(\hat{u})$.

By (2) we have

$$\frac{d\Psi(u, f)}{du[i]} = \psi'(u[i] - f[i]) \quad \text{and} \quad \frac{d\Phi(u)}{du[i]} = \sum_{j \in \mathbb{I}_r} \varphi'(g_j u) g_j[i]$$

so that the minimizer \hat{u} is determined by

$$\psi'(\hat{u}[i] - f[i]) = -\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i], \quad i \in \mathbb{I}_n.$$

With $\xi := (\psi')^{-1}(\cdot, \alpha_1)$ this can be rewritten as

$$\hat{u}[i] = f[i] + \xi\left(-\beta \sum_{j \in \mathbb{I}_r} \varphi'(g_j \hat{u}) g_j[i]\right), \quad i \in \mathbb{I}_n. \quad (6)$$

³An almost everywhere true property requires *only* that \mathbb{K}^n is dense in \mathbb{R}^n . But \mathbb{K}^n may not contain open subsets. There are many examples. For instance, $\mathbb{K} := [0, 1] \setminus \{x \in [0, 1] : x \text{ is rational}\}$ is dense in $[0, 1]$ and \mathbb{K} does not contain open subsets.

Using the vector notation

$$G := \begin{pmatrix} g_1 \\ \vdots \\ g_r \end{pmatrix} \in \mathbb{R}^{r \times n}, \quad \varphi(Gu) := (\varphi(g_j u))_{j=1}^r, \quad \text{and} \quad \xi(u) := (\xi(u_i))_{i=1}^n$$

and recalling that ξ is odd, we obtain from (6) the fixed point equation

$$\hat{u} = T(\hat{u}), \quad T(u) := f - \xi(\beta G^T \varphi'(Gu)). \quad (7)$$

We propose to apply a simple *fixed point algorithm* to compute \hat{u} :

Minimization Algorithm

Initialization: $u^{(0)} = f$

For $r = 1, \dots, R$ compute until a convergence criterion is reached

$$u^{(r)} = T(u^{(r-1)}). \quad (8)$$

As stopping criterion we propose $\|\nabla J\|_\infty \leq 10^{-6}$. For the images used in our numerical examples and in many more experiments the required precision was reached in general within less than 35 iterations.

The efficiency of the algorithm relies on two clues:

- By Theorem 2 the vector $u^{(0)} = f$ is very close to the fixed point \hat{u} and is therefore a good starting point.
- The functions φ' and ξ appearing in the algorithm are given explicitly, in our experiments by Table 1.

The following theorem provides a convergence result for our fixed point algorithm.

Theorem 3: (Convergence of Fixed Point Algorithm) Let ψ and φ fulfill H1. Let $\alpha_1, \alpha_2 > 0$ and $\beta > 0$ be chosen such that $\beta < \frac{1}{4}$ and

$$8\beta \xi'(4\beta, \alpha_1) \varphi''(0, \alpha_2) < 1. \quad (9)$$

Then the sequence $\{u^{(r)}\}_r$ generated by the minimization algorithm converges to the minimizer \hat{u} of $J(\cdot, f)$.

Proof. By Ostrowski's theorem [12] it is enough to prove that the Jacobian matrix $\nabla T(u)$ becomes smaller than 1 in some norm on \mathbb{R}^n for all $u \in \mathbb{R}^n$. Since

$$\nabla T(u) = \beta \text{diag}(\xi'(\beta G^T \varphi'(Gu))) G^T \text{diag}(\varphi''(Gu)) G$$

we obtain

$$\|\nabla T(u)\|_2 \leq \beta \|\text{diag}(\xi'(\beta G^T \varphi'(Gu)))\|_2 \|G^T\|_2 \|\text{diag}(\varphi''(Gu))\|_2 \|G\|_2.$$

Since φ'' is monotone decreasing on $[0, +\infty)$ we get $\|\text{diag}(\varphi''(Gu))\|_2 \leq \varphi''(0)$. Further, we have by definition (3) of G that $\|G^T\|_2 \|G\|_2 = \|G^T G\|_2 < 8$. Note that $G^T G$ is a discrete Laplacian with Neumann boundary conditions and that the bound is sharp in the sense that $\|G^T G\|_2$ approaches 8 as $n \rightarrow \infty$.

It remains to estimate $\xi'(\beta G^T \varphi'(Gu))$. Regarding that $|\varphi'(t)| \leq 1$ for all $t \in \mathbb{R}$ we conclude $\|G^T \varphi'(Gu)\|_\infty \leq \|G\|_1 \leq 4$ where the later follows by (3). Since ξ' increases on $(0, 1)$ and $4\beta < 1$ we obtain finally

$$\|\text{diag}(\xi'(\beta G^T \varphi'(Gu)))\|_2 \leq \xi'(4\beta).$$

Multiplying the parts together we obtain the assertion. □

If both $\psi(\cdot, \alpha_1)$ and $\varphi(\cdot, \alpha_2)$ are equal to θ_1 , resp. θ_2 in Table I, then the left-hand side of (9) becomes

$$\frac{8\beta \xi'(4\beta, \alpha_1) \varphi''(0, \alpha_2)}{\sqrt{\frac{\alpha_1}{\alpha_2} \frac{8\beta}{\sqrt{(1-(4\beta)^2)^3}}}} \left| \begin{array}{c|c} \theta_1 & \theta_2 \\ \hline \frac{\alpha_1}{\alpha_2} \frac{8\beta}{1-|4\beta|} & \end{array} \right|,$$

and in case $\alpha_1 = \alpha_2$ condition (9) reads as

$$\beta < 0.0976 \text{ if } \psi = \varphi = \theta_1, \quad \beta < 0.0833 \text{ if } \psi = \varphi = \theta_2, \quad (10)$$

In practice, we are not really interested in the minimizer \hat{u} of $J(\cdot, f)$, but want to use the sorting of its components to get a meaningful ordering of the original image. Here we observed the following phenomenon:

Observation. The approximation $u^{(r)}$ obtained after $r = 6$ steps of the minimization algorithm leads nearly to the same ordering as the ordering obtained when the convergence criterion was reached. More precisely, a nearly strict ordering is always possible after these few steps, see Table II. In the histogram inversion experiments, the obtained PSNR values are the same, see Table III and the difference images are not distinguishable for the naked eye, see Fig. 3.

This fact led us to propose the following efficient ordering algorithm for $R \ll 35$:

Ordering Algorithm

Initialization: $u^{(0)} = f$

1. For $r = 1, \dots, R$ compute

$$u^{(r)} = T(u^{(r-1)}).$$

2. Order the values in \mathbb{I}_n according to the corresponding ascending entries of $u^{(r)}$.

Note that step 1 of this algorithm is a nonlinear filtering of f .

Applying the same technique as in [2, Theorems 1] we can prove the following corollary.

Corollary 1: (Distance of $u^{(r)}$ from f : upper bound) Let ψ and φ fulfill H1. Then, for $\beta < \frac{1}{4}$, all iterates $u^{(r)}$ generated by the fixed point algorithm satisfy

$$\|u^{(r)} - f\|_\infty \leq (\psi')^{-1}(4\beta, \alpha_1) = \xi(4\beta, \alpha_1).$$

Proof. By (7), (8) and H1 we can estimate

$$\|u^{(r)} - f\|_\infty \leq (\psi')^{-1}(\beta \|h^{(r-1)}\|_\infty),$$

where

$$h[i]^{(r-1)} := \sum_{j \in \mathbb{I}_r} \varphi'(g_j u^{(r-1)}) g_j[i], \quad i \in \mathbb{I}_n.$$

Using $|\varphi'(t)| \leq 1$ and the definition of the $g_j[i]$ we obtain $|h[i]^{(r-1)}| \leq \sum_{j \in \mathbb{I}_r} g_j[i] \leq 4$ and since $(\psi')^{-1}$ is increasing on $[-1, 1]$ for $\beta < \frac{1}{4}$ finally

$$\|u^{(r)} - f\|_\infty \leq (\psi')^{-1}(4\beta, \alpha_1).$$

□

The corollary has two important consequences: first, since for $\beta < \frac{1}{4}$ the sequence of iterates $\{u^{(r)}\}_{r \in \mathbb{N}}$ is bounded, it has a convergent subsequence. Convergence of the whole sequence is ensured under the more restrictive conditions posed in Theorem 3. Second, for appropriately chosen parameters β and α_1 as those in our numerical experiments all iterates fulfill again the important property $|f[i] - u^{(r)}[i]| < 0.5$, $i \in \mathbb{I}_n$. This means if $f[i] < f[j]$, $f[i] \in \mathbb{Z}$, then also $u^{(r)}[i] < u^{(r)}[j]$, so that the pixel ordering in all iterates always respects the pixel ordering in the original image.

IV. NUMERICAL COMPARISON OF SORTING ALGORITHMS

From Table I one sees that the main ingredients of the fixed point algorithm (7), namely ξ and φ' involve much easier arithmetical operations for θ_2 than for θ_1 . So we can expect that using θ_2 would give rise to a faster algorithm than using θ_1 .

In this section we demonstrate that our variational approach together with the proposed sorting algorithm for the functions $\psi = \varphi = \theta_2$ is actually the best way (in terms of speed and quality) to order the pixels in digital images. Note that extensive qualitative comparisons of the variational ordering method with a slower algorithm were done in [9]. These experiments have always shown that the variational method clearly outperforms other state-of-the-art algorithms as LM [4] and WA [14] concerning quality. Here we want to demonstrate that our new sorting algorithm ensures nearly the same quality, in particular a faithful strict ordering, but is much faster than the previous implementations. All algorithms are implemented in MATLAB and executed on a computer with an Intel Core i7-870 Processor (8M Cache, 2.93 GHz) and 8 GB physical memory, 64 Bit Linux.

The performance was done for the 24 digital 8-bit images in Fig. 2, where the first six images have size 256×256 , the next six images 512×512 , then 1024×1024 and finally size 2048×2048 . The tables give the average computation time of 50 runs of the algorithms, respectively.

We compare the ordering methods:

- a) LM: local mean ordering [4] with parameter $K = 6$,
- b) WA: wavelet-based ordering [14] with parameter $K = 9$,

with our variational approach (VA) with parameters $(\beta, \alpha_1, \alpha_2) = (0.1, 0.05, 0.05)$ and

- c) VA-PR: the Polak-Ribière algorithm and θ_1 , stop if $\|\nabla J\|_\infty < 10^{-6}$ but at most 35 iterations as proposed in [9],
- d) VA- θ_1 : our fixed point sorting algorithm with $R = 35$ steps and function θ_1 proposed in [10],
- e) VA- θ_2 : our fixed point sorting algorithm with $R = 35$ steps and function θ_2 ,
- f) VA- $\theta_2 f$: our *fast* fixed point sorting algorithm with $R = 6$ steps and function θ_2 .

We want to mention that the estimate in Theorem 3, see (10) is too restrictive and we have chosen β slightly larger which still provides a convergent iteration scheme.

Concerning the speed of WA and LM we like to emphasize the following: In our present implementation we have to keep for the LM method six images of the size of the original image into the storage and for the WA method nine images. These 6, resp. 9 images were ordered at once to achieve the fast speed commented in the tables. For larger images, e.g., of size 5616×3744 taken by usual commercial cameras, both the LM and WA algorithms do not work in the present form due to storage problems while our algorithm performs fine.

First we compare how faithful the algorithms can order natural images and the corresponding computational time. The results are reported in Tab. II. Here `Fail%` gives the percentage of image pixels which cannot be faithfully ordered. All VA algorithms clearly outperform the LM and WA algorithms also if only a reduced number of 6 iterations was applied. Concerning computational time the VA with the fixed point algorithm and θ_2 is the fastest and VA- $\theta_2 f$ clearly outperforms LM and WA.

Next we compare the performance of the several algorithms for histogram equalization inversion: First the original 8-bit image f with histogram h_f is mapped to an 8-bit image g which histogram resembles a uniform distribution. This requires the first application of an ordering algorithm. Then g is transformed to an 8-bit image \tilde{f} with histogram h_f which requires a second time an ordering algorithm. Tab. III shows the PSNR $20 \log_{10}(255M \cdot N / \|f - \tilde{f}\|_2)$, the percentage of pixels `Fail%` which cannot be faithfully ordered averaged over the two applied ordering procedures and the computational time of the whole histogram equalization inversion process. Since VA-PR and VA- θ_1 give qualitatively, in terms of PSNR and FAIL, the same results as VA- θ_2 but VA- θ_2 is faster, we consider only VA- θ_2 and its faster variant VA- $\theta_2 f$. The VA-algorithms outperform LM and WA wrt PSNR and FAIL. Moreover, VA- $\theta_2 f$ is the fastest algorithm. The quality of our VA algorithms is emphasized by Fig. 3 which shows three difference images between the original f and the image \tilde{f} obtained by histogram equalization inversion. The first row contains the original images, respectively, a part of them. The following rows contains

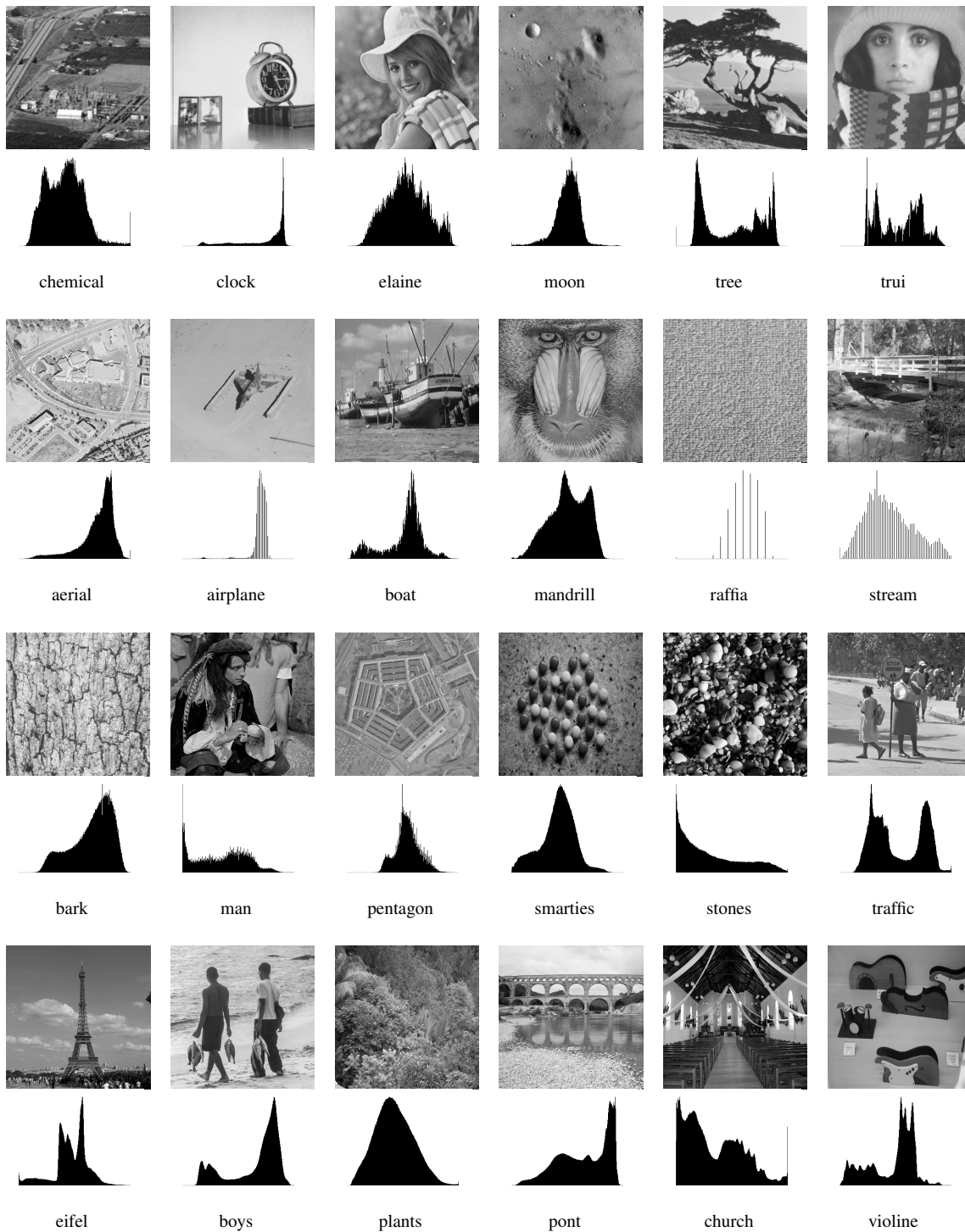


Fig. 2. All 24 digital 8-bit images with their histograms used for our comparison. The size of the images ranges from 256×256 in the first row to 2048×2048 in the fourth row.

the difference images between the original image and the image obtained by histogram equalization inversion. The second and third rows show the results obtained by the LM and WA ordering, respectively. The fourth row shows the difference images by VA-02 and the fifth row by VA-02f. The VA methods, in particular the fast one, are able to reconstruct the original image more precisely than its competitors in particular at edges.

Acknowledgements. The work of Mila Nikolova was supported in part by the “FMJH Program Gaspard Monge in

method	Fail %						Computation Time					
				VA						VA		
	LM	WA	PR	θ_1	θ_2	$\theta_2 f$	LM	WA	PR	θ_1	θ_2	$\theta_2 f$
chemical	0.01	0.06	0.00	0.00	0.00	0.00	0.03	0.04	0.59	0.08	0.06	0.01
clock	1.57	4.52	0.00	0.00	0.00	0.00	0.03	0.04	0.27	0.08	0.06	0.01
elaine	0.00	0.02	0.00	0.00	0.00	0.00	0.03	0.03	0.61	0.08	0.07	0.02
moon	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.61	0.08	0.06	0.02
tree	0.02	0.19	0.00	0.00	0.00	0.00	0.03	0.04	0.59	0.08	0.06	0.01
trui	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.30	0.08	0.06	0.02
means	0.27	0.80	0.00	0.00	0.00	0.00	0.03	0.04	0.50	0.08	0.06	0.01
aerial	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.16	2.28	0.40	0.29	0.07
airplane	5.30	17.70	0.00	0.00	0.00	0.00	0.16	0.31	1.16	0.40	0.29	0.07
boat	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.15	1.40	0.40	0.29	0.07
mandrill	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.14	2.25	0.40	0.29	0.07
raffia	13.66	16.05	0.00	0.00	0.00	0.00	0.20	0.35	1.04	0.39	0.29	0.07
stream	0.41	0.75	0.00	0.00	0.00	0.07	0.13	0.19	1.03	0.41	0.29	0.07
means	3.23	5.75	0.00	0.00	0.00	0.01	0.14	0.22	1.53	0.40	0.29	0.07
bark	0.00	0.00	0.00	0.00	0.00	0.00	0.64	0.80	10.16	1.79	1.33	0.26
man	0.34	0.68	0.00	0.00	0.00	0.00	0.61	0.82	4.47	1.62	1.22	0.26
pentagon	0.00	0.00	0.00	0.00	0.00	0.00	0.66	0.87	9.48	1.80	1.32	0.26
smarties	0.08	0.05	0.00	0.00	0.00	0.00	0.68	0.90	9.43	1.82	1.46	0.30
stones	1.14	1.39	0.00	0.00	0.00	0.03	0.66	0.90	4.76	1.78	1.32	0.25
traffic	0.10	0.07	0.00	0.00	0.00	0.00	0.65	0.88	4.96	1.80	1.44	0.28
means	0.28	0.36	0.00	0.00	0.00	0.00	0.65	0.86	7.21	1.77	1.35	0.27
eifel	0.57	0.37	0.00	0.00	0.00	0.00	4.24	6.55	22.29	8.08	6.58	1.38
boys	0.02	0.00	0.00	0.00	0.00	0.00	3.84	5.46	42.64	8.06	6.60	1.38
plants	0.00	0.00	0.00	0.00	0.00	0.00	3.74	4.74	21.61	8.08	6.62	1.38
pont	1.96	5.77	0.00	0.00	0.00	0.00	3.98	5.70	43.15	8.08	6.63	1.46
church	0.61	0.78	0.13	0.05	0.07	0.26	3.62	5.16	20.74	8.08	6.60	1.39
violine	0.25	0.23	0.00	0.00	0.00	0.00	3.76	5.95	43.25	8.07	6.59	1.37
means	0.57	1.19	0.02	0.01	0.01	0.04	3.86	5.59	32.28	8.07	6.60	1.39

TABLE II

COMPARISON OF DIFFERENT ORDERING METHODS FOR THE IMAGES IN FIG. 2.

optimization and operation research”, and by the support to this program from EDF.

REFERENCES

- [1] N. Bassiou and C. Kotropoulos, *Color image histogram equalization by absolute discounting back-off*, Computer Vision and Image Understanding, 107 (2007), pp. 108–122.
- [2] F. Bauss, M. Nikolova, and G. Steidl, *Fully smoothed ℓ_1 -TV models: Bounds for the minimizers and parameter choice*, Journal of Mathematical Imaging and Vision, (2013), online.
- [3] V. Caselles, J. L. Lisani, J. M. Morel, and G. Sapiro, *Shape preserving local histogram modification*, IEEE Trans. on Image Processing, 8 (1999), pp. 220–229.
- [4] D. Coltuc, P. Bolon, and J.-M. Chassery, *Exact histogram specification*, IEEE Trans. on Image Processing, 15 (2006), pp. 1143–1152.
- [5] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley, 1993.
- [6] E. L. Hall, *Almost uniform distributions for computer image enhancement*, IEEE Trans. on Computers, C-23 (1974), pp. 207–208.
- [7] J. H. Han, S. Yang, and B. U. Lee, *A novel 3-D color histogram equalization method with uniform 1-D gray scale histogram*, IEEE Trans. on Image Processing, 20 (2011), pp. 506–512.
- [8] S. F. Naik and C. A. Murthy, *Hue-preserving color image enhancement without gamut problem*, IEEE Trans. on Image Processing, 12 (2003), pp. 1591–1598.

method	PSNR				Fail %				Computation Time			
			VA				VA				VA	
	LM	WA	θ_2	θ_{2f}	LM	WA	θ_2	θ_{2f}	LM	WA	θ_2	θ_{2f}
chemical	49.34	48.90	49.67	49.67	0.03	0.10	0.00	0.00	0.06	0.07	0.13	0.03
clock	51.69	51.56	51.78	51.78	0.87	2.41	0.00	0.00	0.06	0.08	0.14	0.03
elaine	49.51	49.66	49.90	49.91	0.00	0.02	0.00	0.00	0.06	0.07	0.13	0.03
moon	47.36	46.50	47.82	47.80	0.06	0.11	0.00	0.00	0.05	0.07	0.13	0.03
tree	51.94	51.84	52.01	52.01	0.03	0.18	0.00	0.00	0.05	0.07	0.12	0.03
trui	52.70	52.51	52.86	52.85	0.04	0.06	0.00	0.00	0.05	0.07	0.13	0.03
means	50.42	50.16	50.67	50.67	0.17	0.48	0.00	0.00	0.05	0.07	0.13	0.03
aerial	48.36	48.06	50.05	50.05	0.00	0.02	0.00	0.00	0.25	0.33	0.57	0.12
airplane	46.74	46.26	47.25	47.25	2.68	8.96	0.00	0.00	0.27	0.49	0.56	0.12
boat	49.51	49.58	49.89	49.89	0.07	0.09	0.00	0.00	0.24	0.32	0.57	0.12
mandrill	48.27	49.47	49.75	49.75	0.00	0.00	0.00	0.00	0.23	0.30	0.56	0.12
raffia	41.12	41.12	41.12	41.12	6.85	8.18	0.00	0.00	0.32	0.54	0.57	0.11
stream	44.75	45.00	45.07	45.07	0.40	0.71	0.00	0.06	0.24	0.36	0.56	0.12
means	46.46	46.58	47.19	47.19	1.67	2.99	0.00	0.01	0.26	0.39	0.57	0.12
bark	51.28	51.17	51.30	51.30	0.00	0.01	0.00	0.00	1.29	1.60	2.60	0.52
man	49.22	49.18	49.44	49.43	0.19	0.40	0.00	0.00	1.20	1.59	2.55	0.50
pentagon	50.69	50.62	51.35	51.35	0.01	0.01	0.00	0.00	1.29	1.66	2.53	0.53
smarties	51.47	51.09	51.64	51.64	0.63	0.91	0.00	0.03	1.34	1.77	2.55	0.53
stones	51.28	50.95	51.60	51.60	0.97	1.22	0.00	0.04	1.31	1.78	2.52	0.51
traffic	50.73	50.58	51.01	51.01	0.25	0.30	0.00	0.01	1.31	1.73	2.53	0.51
means	50.78	50.60	51.06	51.05	0.34	0.48	0.00	0.01	1.29	1.69	2.55	0.52
eifel	48.61	48.48	48.74	48.72	0.88	0.97	0.00	0.05	8.17	12.36	13.15	2.75
boys	51.63	51.43	51.71	51.71	0.09	0.13	0.00	0.00	7.45	10.44	13.15	2.77
plants	48.96	48.86	49.52	49.52	0.18	0.24	0.00	0.02	7.35	9.29	13.15	2.75
pont	51.48	51.30	51.55	51.52	1.42	3.50	0.01	0.09	7.78	10.90	13.24	2.85
church	50.94	50.80	51.28	51.28	1.23	1.73	0.06	0.28	7.16	10.23	13.16	2.78
violine	51.66	51.34	51.85	51.79	1.22	1.59	0.00	0.22	7.25	11.29	13.12	2.73
means	50.55	50.37	50.78	50.76	0.84	1.36	0.01	0.11	7.53	10.75	13.16	2.77

TABLE III

COMPARISON OF HISTOGRAM EQUALIZATION INVERSION ALGORITHMS.

- [9] M. Nikolova, Y. Wen, and R. Chan, *Exact histogram specification for digital images using a variational approach*, Journal of Mathematical Imaging and Vision, (2012), online.
- [10] M. Nikolova, *A fast algorithm for exact histogram specification. Simple extension to color images*, Scale Space and Variational Methods in Computer Vision, A. Kuijpers, K. Bredies and T. Pock (eds.), LNCS 7893, Springer-Verlag, Berlin, pp. 174–185 (2013).
- [11] M. Nikolova, and G. Steidl, *Fast Hue and Range preserving Histogram Specification. Theory and New Algorithms for Image Enhancement*, HAL Preprint, 2013.
- [12] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
- [13] D. Sen and P. Sankar, *Automatic exact histogram specification for contrast enhancement and visual system based quantitative evaluation*, IEEE Trans. on Image Processing, 20 (2011), pp. 1211–1220.
- [14] Y. Wan and D. Shi, *Joint exact histogram specification and image enhancement through the wavelet transform*, IEEE Trans. on Image Processing, 16 (2007), pp. 2245–2250.

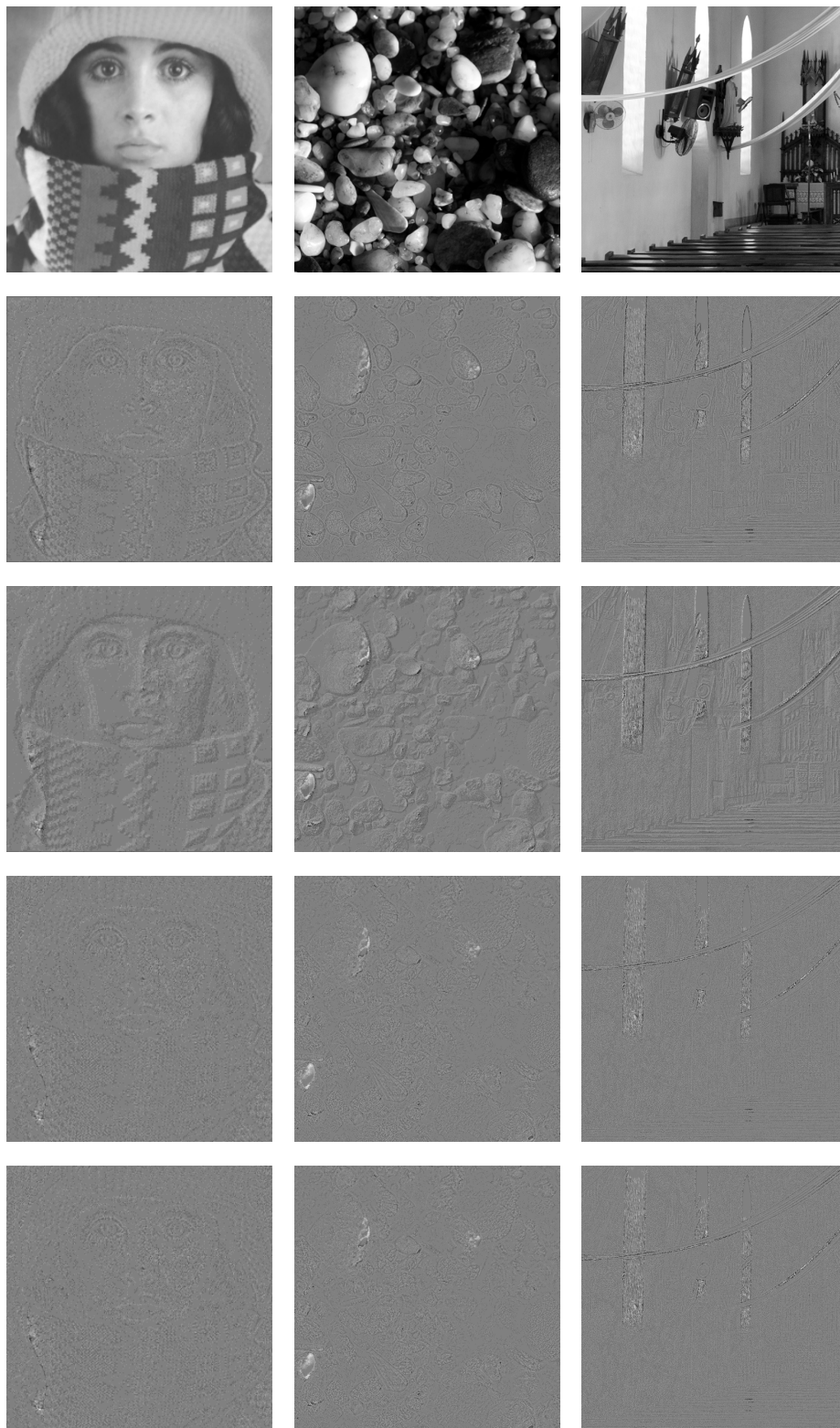


Fig. 3. Comparison of ordering methods for histogram equalization inversion. First row: true images or parts of them. The following rows show the difference images between the original one and those obtained after histogram equalization inversion. Second row: LM method. Third row: WA method. Fourth row: variational method e). Fifth row: variational method f). The variational methods (VA) contain much less errors than those achieved by LM and WA. Moreover there is no visual difference between the VA with 35 iterations and its faster variant with only 6 iterations.