

# Dithering by Differences of Convex Functions

T. Teuber\*, G. Steidl\*, P. Gwosdek†, Ch. Schmaltz†, J. Weickert†

April 29, 2010

## Abstract

Motivated by a recent halftoning method which is based on electrostatic principles, we analyse a halftoning framework where one minimizes a functional consisting of the difference of two convex functions (DC). One of them describes attracting forces caused by the image gray values, the other one enforces repulsion between points. In one dimension, the minimizers of our functional can be computed analytically and have the following desired properties: the points are pairwise distinct, lie within the image frame and can be placed at grid points. In the two-dimensional setting, we prove some useful properties of our functional like its coercivity and propose to compute a minimizer by a forward-backward splitting algorithm. We suggest to compute the special sums occurring in each iteration step by a fast summation technique based on the fast Fourier transform at non-equispaced knots which requires only  $\mathcal{O}(m \log m)$  arithmetic operations for  $m$  points. Finally, we present numerical results showing the excellent performance of our dithering method.

## 1 Introduction

Digital halftoning is a method for creating the illusion of a continuous tone image with a device having only a limited number of tones available, see [55] for an introduction and Fig. 1 for illustration. Applications of halftoning include printing and geometry processing [49] as well as sampling problems occurring in rendering [54], re-lighting [31] or object placement and artistic non-photorealistic image visualization [4, 45]. Halftoning, resp. dithering has been an active field of research for many years. An early dithering technique proposed in [17] perturbs the initial image by white Gaussian noise prior to quantization to avoid quantization boundaries. Point processes as the so-called *ordered dithering* approximate colour values locally by predefined regular local patterns [7] or by more random patterns as the method proposed by Purgathofer et al. [43].

Due to their run-time efficiency *error diffusion* methods as those of Floyd and Steinberg [14] are very popular halftoning algorithms. Extensions of the Floyd-Steinberg algorithm use different neighborhoods or grids [25, 46, 47]. To enhance the visual quality rather than approximating the image from the mathematical point of view edge enhancing preprocessing steps of the initial image were suggested in [25, 26]. These methods are very similar to the unsharp masking method in [16]. A variant of error diffusion called *dot diffusion* was developed in [30]. All these methods produce unpleasant visual artefacts, especially so-called

---

\*University of Mannheim, Dept. of Mathematics and Computer Science, A5, 68131 Mannheim, Germany

†Saarland University, Dept. of Mathematics and Computer Science, Mathematical Image Analysis Group, Campus E1.1, 66041 Saarbrücken, Germany



Figure 1: Original  $650 \times 560$  image of a cat, cf. [38] (left) and the stippled image by our proposed method with  $m = 194682$  dots (right).

worms and disturbing patterns. Ostromoukhov [37] detected that the disturbing patterns in the Floyd-Steinberg algorithm occur mainly at certain gray values. Therefore he introduced a special distribution mask for the gray values. Further improvements were suggested by Zhou et al. [58] using threshold modulation as in [29] and stencils optimized for this situation. Recently, Chang et al. [9] proposed a so-called *structure-aware error diffusion* algorithm. Their modulation depends on the local frequency content of the image (local dominant frequency/orientation/contrast). The amount of thresholding is manually calibrated and stored in five lookup tables in such a way that for each specific combination of the above quantities the error diffusion system produces a halftone output which is perceptually close to the source image. A different technique which also performs scan line processing optimizes the halftoned result in a local neighborhood based on an electrostatic analysis and has been patented in [24].

Another class of digital halftoning methods applies iterative procedures such as global search or direct binary search [1, 5]. The *structure-aware halftoning* of Pang et al. [38] introduces an iterative optimization method to maintain both the local tone and the original texture of the image with the drawback of a long execution time. Finally, we mention two techniques in [15] by neural networks and Markovian simulations which reduce local errors by optimizing the frequencies occurring within local neighborhoods.

*Stippling* algorithms which are closely related to halftoning techniques use the continuous domain as possible point positions. Non-photorealistic stippling places black dots in such a way that their distribution gives the impression of tone. One prominent example of this technique was proposed by Secord [45]. It is based on *weighted centroidal Voronoi tessellations* and *Lloyd's iterative algorithm* [33]. Recall that Lloyd's algorithm tries to find for a given weight function  $w : \Omega \rightarrow [0, 1]$  on a bounded domain  $\Omega \in \mathbb{R}^2$  and given  $m \in \mathbb{N}$  a useful minimizer of the error functional

$$L((p_k, V_k)_{k=1}^m) := \sum_{k=1}^m \int_{V_k} w(x) |x - p_k|^2 dx$$

over any set of points  $\{p_k\}_{k=1}^m$  belonging to  $\bar{\Omega}$  and any tessellation  $\{V_k\}_{k=1}^m$  of  $\bar{\Omega}$ . Note that the above functional was also examined for other than the Euclidian norm. A necessary condition for  $L$  to be minimized is that the  $\hat{V}_k$ 's are the Voronoi regions corresponding to the  $\hat{p}_k$ 's, and, simultaneously, the  $\hat{p}_k$  are the centroids of the corresponding  $\hat{V}_k$ 's, see, e.g., [11]. Starting with an initial guess of  $\{p_k^{(0)}\}$  Lloyd's algorithm iteratively finds the corresponding Voronoi tessellation and then the centroids corresponding to this tessellation. For the continuous setting this requires  $\mathcal{O}(m \log m)$  arithmetic operation per iteration step mainly due to the construction of the Voronoi tessellation. The local convergence of the algorithm was proved in 1D for differentiable, strictly logarithmical concave weights  $w$  in [28]. The convergence analysis in the multidimensional case is far from being complete. Few results are known, see [10]. For a comprehensive overview and references up to 1999 we refer to [11]. Recently, a capacity-constrained variant of Lloyd's algorithm was introduced by Balzer et al. [4]. This approach aims to overcome the drawback of Lloyd's algorithm which introduces regularity artefacts if not stopped at a suitable iteration step and requires some additional computational expenses.

In this paper, we propose a completely different approach both for halftoning and stippling. We minimize a global energy functional which is the difference of two functionals (DC), namely one for the attraction of the dots by the image gray values and the other one (with the negative sign) for the repulsion between the dots. Our method was inspired by the electrostatically motivated paper [44] and is to the best of our knowledge the first attempt to treat the dithering problem from a convex analysis point of view. Our functional is the difference of two continuous, convex functions so that we can build on existing results for the computation. To get an idea about the behavior of our functional we first deal with the one-dimensional setting. We show that after assuming an ordering of the point positions the functional becomes convex and the minimizers can be determined analytically. We prove that the minimizing points have the following desired properties: they are pairwise distinct, lie within the image frame and can always be placed at grid positions. Some of these useful properties carry over to the actual two-dimensional case. We propose to use a forward-backward splitting (FBS) algorithm for DC programming also known as proximal point (type) algorithm or Moreau-Yoshida regularization [23, 20, 48]. The proposed algorithms involve the computation of special sums in each iteration step where straightforward computation would require  $\mathcal{O}(m^2)$  arithmetic operations for  $m$  points. We suggest to compute these sums by a fast summation procedure based on the fast Fourier transform at non-equispaced knots. Similar to the fast multipole method this summation algorithm requires only  $\mathcal{O}(m \log m)$  arithmetic operations. In the rest of the paper, we will not strictly distinguish between halftoning/dithering and stippling since the used technique becomes clear from the context. We will mainly use the first term.

This paper is organized as follows: In Section 2 we introduce our functional whose minimizers will serve as dithering/stippling results. In Section 3 we examine the simpler one-dimensional setting in order to get an idea about the behavior of our functional and move to the two-dimensional case in Section 4. We prove some useful simple properties of our functional in Subsection 4.1 and review the forward-backward splitting (FBS) algorithm for DC programming. In Subsection 4.2 we show how the large sums involved in our iterative minimization algorithms can be computed by a fast summation procedure based on the fast Fourier transform at non-equispaced knots. Finally, we present numerical results showing the excellent

quality of our dithering/stippling method in Section 5. The paper ends with conclusions in Section 6.

## 2 General Problem

We consider images  $u : G \rightarrow [0, 1]$  on an integer grid  $G := \{1, \dots, n_x\} \times \{1, \dots, n_y\}$ , where 'black' is 0 and 'white' 1. Let  $w := 1 - u$  denote the corresponding weight distribution. Let  $m$  be the number of black pixels produced by the dithering process and  $p := (p_k)_{k=1}^m = ((p_{k,x}, p_{k,y})^T)_{k=1}^m \in \mathbb{R}^{2m}$  be their position vector. By  $|p_k| := \sqrt{p_{k,x}^2 + p_{k,y}^2}$  we denote the Euclidian norm of the position of the  $k$ -th black pixel. Let  $\lambda$  be chosen by

$$\lambda := \frac{1}{m} \sum_{(i,j) \in G} w(i,j).$$

In the following, we want to determine minimizers  $\hat{p}$  of the functional

$$E(p) = \underbrace{\sum_{k=1}^m \sum_{(i,j) \in G} w(i,j) \left| p_k - \begin{pmatrix} i \\ j \end{pmatrix} \right|}_{F(p)} - \lambda \underbrace{\sum_{k=1}^m \sum_{l=k+1}^m |p_k - p_l|}_{G(p)}. \quad (1)$$

Note that

$$F(p) := \sum_{k=1}^m f(p_k), \quad f(q) := \sum_{(i,j) \in G} w(i,j) \left| q - \begin{pmatrix} i \\ j \end{pmatrix} \right|.$$

The intention for considering minimizers of this functional as point positions for dithering is as follows: consider the points with positions  $p_k$  as small particles of equal size moving in an environment, e.g., a glass pane above the image  $w$ . The particles are *attracted* by the image forces  $w(i, j)$  at the points  $(i, j) \in G$ . On the other hand, there is a force of *repulsion* between the particles modeled by the negative sign of  $G$  which tries to maximize the distances between the particles.

The functional (1) is the difference of two continuous, convex functionals  $F$  and  $G$  so that we can built on existing results of *DC (difference of convex functions, delta-convex functions)* computation. For a recent overview on DC functions see [3].

### Remark 2.1.

- i) **Role of  $\lambda$ .** The parameter  $\lambda$  is not a usual regularization parameter but an equilibration parameter between the 'opposite' functionals. Since  $\frac{1}{\lambda}E(p)$  has the same minimizer as  $E(p)$  one can set  $\lambda := 1$  and use  $\frac{1}{\lambda}w(i, j)$  instead of  $w(i, j)$  in the functional. Therefore we restrict our attention mainly to  $\lambda = 1$  and  $m = \sum_{(i,j) \in G} w(i, j)$ . In applications we often use  $m := \text{round}(\sum_{(i,j) \in G} w(i, j))$  to *preserve the mean gray value of the image*.
- ii) **Continuous setting.** Sometimes we will also consider the continuous counterpart of (1), namely

$$\sum_{k=1}^m \int_{\Omega} w(x) |p_k - x| dx - \lambda \sum_{k=1}^m \sum_{l=k+1}^m |p_k - p_l|,$$

where  $\Omega := (\frac{1}{2}, \frac{1}{2} + n_x) \times (\frac{1}{2}, \frac{1}{2} + n_y)$ .

iii) **Other functionals.** More general functionals of the form

$$E_\varphi(p) = \sum_{k=1}^m \sum_{(i,j) \in G} w(i,j) \varphi \left( \left| p_k - \begin{pmatrix} i \\ j \end{pmatrix} \right| \right) - \lambda \sum_{k=1}^m \sum_{l=k+1}^m \varphi(|p_k - p_l|), \quad (2)$$

with  $\varphi : [0, \infty) \rightarrow \mathbb{R}$  may also be of interest. In our setting (1), we have  $\varphi(s) = s$ . We will see that for  $\varphi(s) := s^\alpha$  it makes only sense to consider these functionals for  $\alpha \in (0, 2)$  since otherwise  $E_\varphi$  is not bounded from below or not coercive. To obtain differentiable functionals one could also replace the Euclidian norm  $|q| = |(q_x, q_y)^T|$  in (2) by

$$|q|_\varepsilon = |(q_x, q_y)^T|_\varepsilon := \sqrt{q_x^2 + q_y^2 + \varepsilon^2}, \quad \varepsilon > 0.$$

We denote the resulting functional by  $E_{\varphi, \varepsilon}$ .

In [44] an iterative global dithering algorithm was proposed which can be considered as a gradient descent algorithm of the functional  $E_{\varphi, \varepsilon}$  with the function  $\varphi(s) = \log(s)$ . Note that  $\log(|q|_\varepsilon)$  is neither convex nor concave which also holds true for the two sums in  $E_{\varphi, \varepsilon}$ . The motivation in [44] came from physical principles of electrostatics, namely from the movement of small equally-charged particles in a global system. Fig. 2 shows the functionals  $E(p_1, p_2)$  and  $E_{\log, \varepsilon}(p_1, p_2)$  in one dimension for a signal  $w \equiv 2/n$  consisting of  $n$  equal weights.

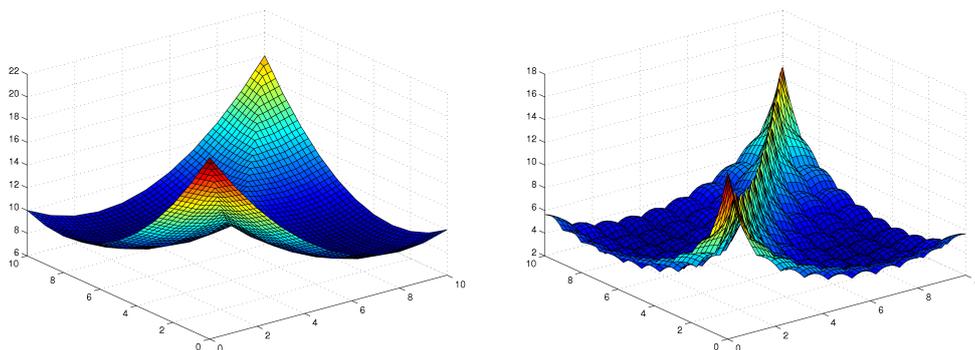


Figure 2: The functionals  $E(p_1, p_2)$  (left) and  $E_{\log, \varepsilon}(p_1, p_2) := \sum_{k=1}^2 \sum_{j=1}^n w(j) \log(|p_k - j|^2 + \varepsilon^2) - \log(|p_1 - p_2|^2 + \varepsilon^2)$  (right) with  $\varepsilon = 0.1$  in one dimension for  $w = (w(j))_{j=1}^{10} = (1/5)_{j=1}^{10}$ .

### 3 One-dimensional Problem

We consider 1D signals with  $n = n_x$ . Then the functional we have to minimize reads

$$E(p) = \sum_{k=1}^m \sum_{j=1}^n w(j) |p_k - j| - \sum_{k=1}^m \sum_{l=k+1}^m |p_k - p_l|.$$

This non-convex 1D minimization problem can be simplified if we assume  $p$  to be an ordered vector with  $p_1 \leq \dots \leq p_m$ . Then we have

$$\sum_{k=1}^m \sum_{l=k+1}^m |p_k - p_l| = \sum_{k=1}^m \sum_{l=k+1}^m p_l - p_k = \sum_{k=1}^m -(m - (2k - 1)) p_k.$$

Hence, instead of minimizing  $E(p)$  over  $p$ , we can consider the convex minimization problem

$$\operatorname{argmin}_{p_1 \leq \dots \leq p_m} E(p) = \operatorname{argmin}_p \sum_{k=1}^m \left( \sum_{j=1}^n w(j) |p_k - j| + (m - (2k - 1)) p_k \right). \quad (3)$$

Note that a minimizer  $\hat{p}$  exists because the functional is continuous and coercive. The following theorem determines the minimizers analytically and shows that they have the useful properties we expected, namely:

- the points  $\hat{p}_k$  are within the signal interval  $[1, n]$ ,
- the points  $\hat{p}_k$  are pairwise different,
- there exists a minimizer within the grid  $\{1, \dots, n\}$ .

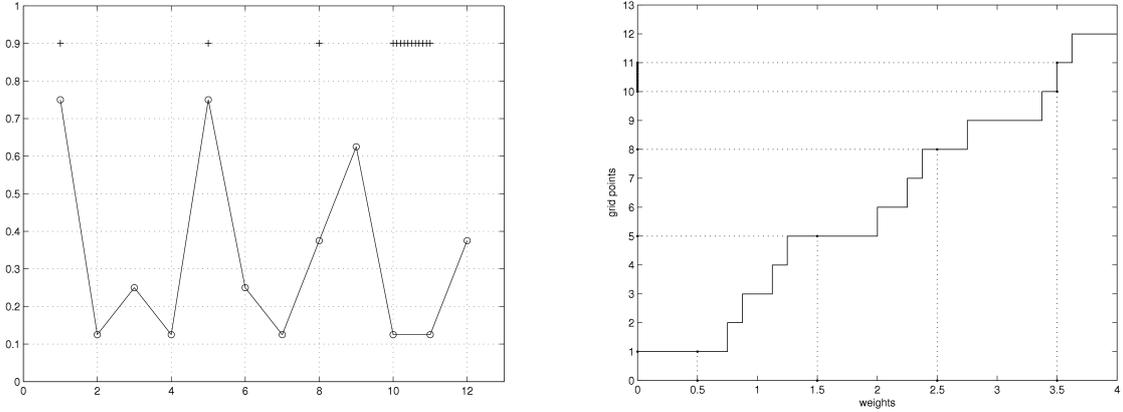


Figure 3: Graphical illustration of the solution of (3). Left: Signal  $w$  consisting of  $n = 12$  points linearly connected. Right: The solution (4) with  $m = 4$  is  $\hat{p}_1 = 1$ ,  $\hat{p}_2 = 5$ ,  $\hat{p}_3 = 8$ ,  $\hat{p}_4 \in [10, 11]$ .

We define

$$a_0 := 0, \quad a_r := \sum_{j=1}^r w(j), \quad r = 1, \dots, n$$

so that the interval  $[0, m]$  can be divided into  $n$  subintervals  $[a_{r-1}, a_r]$  of length  $w(r)$ . Furthermore let  $\partial F(p^*) := \{p : F(x) - F(p^*) \geq \langle p, x - p^* \rangle \forall x\}$  denote the *subdifferential* of  $F$  at  $p^*$ , see [22].

**Theorem 3.1. (Analytical solutions in 1D)**

Let  $m := \sum_{j=1}^n w(j) \in \mathbb{N}$  and  $w(j) \in [0, 1]$ ,  $j = 1, \dots, n$ . Then the solutions  $\hat{p}$  of the minimization problem (3) are given by

$$\hat{p}_k \in \begin{cases} r & \text{if } k - \frac{1}{2} \in (a_{r-1}, a_r), \\ [r, r + s] & \text{if } k - \frac{1}{2} = a_r = \dots = a_{r+s-1} \neq a_{r+s}, s \geq 1, a_r \neq a_{r-1}. \end{cases} \quad (4)$$

In particular, we have that  $\hat{p}_k \in [1, n]$  for all  $k = 1, \dots, m$ . If  $0 \leq w(j) < 1$ ,  $j = 1, \dots, n$ , then for any minimizer  $\hat{p}$  of (3) the relation  $\hat{p}_k \neq \hat{p}_l$  holds true for  $k \neq l$ . There always exists a minimizer  $\hat{p} \in \mathbb{N}^m$  with pairwise different entries.

**Proof:** The functional in (3) can be minimized separately for each  $p_k$ , i.e., we have to find a minimizer  $\hat{p}_k$  of

$$E_k(q) := \sum_{j=1}^n w(j) |q - j| + (m - (2k - 1))q \quad (5)$$

for all  $k = 1, \dots, m$ . Note that the minimizer of the first sum on the right-hand side is the weighted median of  $(j)_{j=1}^n$ . By Fermat's rule,  $\hat{p}_k$  is a solution of (5) if and only if

$$\begin{aligned} 0 &\in \partial E_k(\hat{p}_k) = \sum_{j=1}^n w(j) \operatorname{sgn}(\hat{p}_k - j) + (m - (2k - 1)), \\ k - \frac{1}{2} &\in \frac{m}{2} + \frac{1}{2} \sum_{j=1}^n w(j) \operatorname{sgn}(\hat{p}_k - j), \end{aligned} \quad (6)$$

where

$$\operatorname{sgn}(q) := \begin{cases} -1 & \text{if } q < 0, \\ [-1, 1] & \text{if } q = 0, \\ 1 & \text{if } q > 0. \end{cases}$$

We will use that

$$\frac{m}{2} + \frac{1}{2} \left( \sum_{j=1}^r w(j) - \sum_{j=r+1}^n w(j) \right) = \sum_{j=1}^r w(j) = a_r. \quad (7)$$

Now we distinguish two cases. For illustration the reader may have a look at Fig. 3, where we have depicted the solution of a special problem graphically.

*Case 1:* Assume that  $k - \frac{1}{2} \in (a_{r-1}, a_r)$  for some  $r \in \{1, \dots, m\}$ . Then we obtain by (6) and (7) that  $\hat{p}_k = r$ .

*Case 2:* Assume that  $k - \frac{1}{2} = a_r$  where  $a_r \neq a_{r-1}$  and  $a_r = \dots = a_{r+s-1} \neq a_{r+s}$ ,  $s \geq 1$ . Then we see by (6) and (7) that  $\hat{p}_k \in [r, r + s]$ . The rest of the assertion follows directly from (4).  $\square$

Figs. 4 and 5 show minimizers  $\hat{p} \in \mathbb{N}^m$  obtained by (4) for different underlying functions  $w$ . In Fig. 5 an interesting disadvantage of Lloyd's algorithm can be seen for the one-dimensional case. If the underlying image contains large uniform regions, Lloyd's algorithm can be incapable of bypassing these regions. The problem is, that since there is no capacity constraint, a stable state can be reached where the sum of each Voronoi region strongly differs. This drawback may be avoided by initializing the halftone image with a sufficient approximation of the result where each area is initialized with about the right number of pixels.

The attraction of the grid points disperses to the whole line if we deal with the continuous counterpart of (3) given by

$$\sum_{k=1}^m \int_{\frac{1}{2}}^{n+\frac{1}{2}} w(x) |p_k - x| dx + (m - (2k - 1))p_k. \quad (8)$$

If we assume for example that

$$w(x) := \begin{cases} w(j) & x \in [j - \frac{1}{2}, j + \frac{1}{2}), \\ 0 & \text{otherwise,} \end{cases}$$

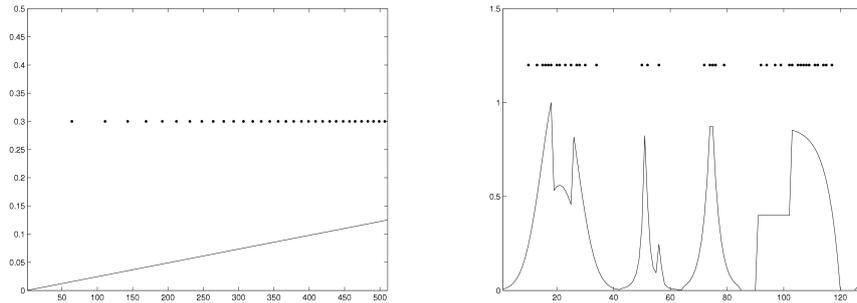


Figure 4: Dithering results (dots) by an integer solution of (4). Left:  $w = (\frac{1}{8 \cdot 511}, \dots, \frac{511}{8 \cdot 511})$  and  $m = 32$ . Right:  $w$  is the 'Piecewise-Regular' signal of length 128 from the WaveLab-Toolbox and  $m = 38$ .

the functional (8) becomes

$$\sum_{k=1}^m \sum_{\substack{j=1 \\ j \neq [p_k]}}^n w(j) |p_k - j| + w([p_k]) (p_k - [p_k])^2 + (m - (2k - 1)) p_k,$$

where  $[p_k]$  denotes the nearest grid neighbor of  $p_k$  in the sense that  $[p_k] = r \in \mathbb{Z}$  if  $p_k \in [r - 1/2, r + 1/2)$ . Of course,  $(p_k - [p_k])^2$  is a weaker penalizer than  $|p_k - [p_k]|$ . If  $w(x)$  is continuous, e.g., the linear interpolation between the  $w(j)$ , then the solution of (8) is given by  $\int_{\frac{1}{2}}^{\hat{p}_k} w(x) dx = k - \frac{1}{2}$ ,  $k = 1, \dots, m$ . In the example in Fig. 3 one would get with linearly interpolated weights and Neumann boundary conditions that  $\hat{p}_1 = 9/14 \approx 0.6429$ ,  $\hat{p}_2 = 59/14 \approx 4.2143$ ,  $\hat{p}_3 = 7.25$  and  $\hat{p}_4 = 10$ . Finally, we mention that in 1D the Voronoi cells corresponding to the  $\hat{p}_k$ 's are determined by  $V_k = [(\hat{p}_{k-1} + \hat{p}_k)/2, (\hat{p}_k + \hat{p}_{k+1})/2]$ , where we set  $\hat{p}_0 := 0$  and  $\hat{p}_{m+1} := n$ . If  $w(x) = w_0$  is constant, then the centroids of  $V_k$  coincide with the  $\hat{p}_k$ 's.

**Remark 3.2. (Relaxed grid attraction by linear interpolation)**

In the two-dimensional computations it will be efficient to replace a special subgradient  $Df(q) \in \partial f(q)$  by its interpolation at the grid points. In 1D this means that we replace

$$Df(q) = \sum_{j=1}^n w(j) \text{sign}(q - j), \quad \text{sign}(x) = \begin{cases} \text{sgn}(x) & x \neq 0, \\ 0 & \text{otherwise} \end{cases}$$

by

$$f'_I(q) := \sum_{r=1}^n Df(r) M_2(q - r) = \sum_{j=1}^n w(j) \sum_{r=1}^n \text{sign}(r - j) M_2(q - r),$$

where  $M_2$  denotes the centered, cardinal  $B$ -spline of order 2. Then

$$f_I(q) = \sum_{j=1}^n w(j) \sum_{r=1}^n \text{sign}(r - j) \psi(q - r)$$

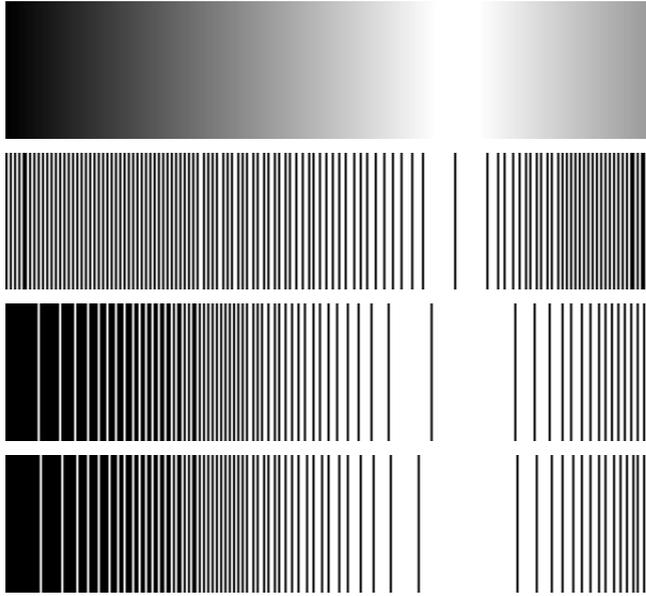


Figure 5: From top to bottom: Image of a 1D signal which is repeated in vertical direction. Results by Lloyd's method, the method of Balzer et al. and our approach as given by (4). We initialized the first two methods with a vector of uniformly distributed points.

with

$$\psi(x) := \begin{cases} -\frac{1}{2} & x \in (-\infty, -1), \\ x + \frac{1}{2}x^2 & x \in [-1, 0), \\ x - \frac{1}{2}x^2 & x \in [0, 1), \\ \frac{1}{2} & x \in [1, \infty) \end{cases}$$

and after some straightforward computation this becomes (up to a constant)

$$f_I(q) = \sum_{j=1}^n w(j)|q - j| + \frac{1}{2}w([q]) (|q - [q]| - 1)^2. \quad (9)$$

In other words, the interpolated subgradient is the derivative of the functional (9) which shows a relaxed grid attraction.

## 4 Two-dimensional Problem

### 4.1 DC Programming

The functional  $E$  in (1) is the difference of the two convex functions  $F$  and  $G$ . A *necessary condition* for  $\hat{p}$  to be a local minimizer of  $E$  is

$$\partial G(\hat{p}) \subset \partial F(\hat{p}). \quad (10)$$

For conditions on global minimizers we refer to [22]. Using (10) we can prove the following useful property of our local minimizers.

**Proposition 4.1. (Property of local minimizers of  $E$ )**

Let  $\hat{p}$  be a local minimizer of (1) and let  $0 \leq w(i, j) < 1$ ,  $(i, j) \in G$ . Then  $\hat{p}_k \neq \hat{p}_l$  holds true for  $k \neq l$ .

**Proof:** If  $\hat{p}$  is a local minimizer of (1), then we know by (10) that

$$\sum_{l=1}^m \partial |\cdot| (\hat{p}_k - \hat{p}_l) \subset \sum_{(i,j) \in G} w(i, j) \partial |\cdot| \left( \hat{p}_k - \begin{pmatrix} i \\ j \end{pmatrix} \right) \quad \forall k = 1, \dots, m.$$

If  $\hat{p}_k = \hat{p}_l$ ,  $k \neq l$ , then the left-hand side contains an area  $q + \{x : |x| \leq 1\}$  which must be contained in the right-hand side. This is only possible if  $\hat{p}_k$  coincides with some grid point and the corresponding weight is equal to 1.  $\square$

In general, condition (10) is hard to be reached and may be relaxed to

$$0 \in \partial_C E(\hat{p}) = \partial F(\hat{p}) - \partial G(\hat{p}) \quad \Leftrightarrow \quad \partial F(\hat{p}) \cap \partial G(\hat{p}) \neq \emptyset, \quad (11)$$

where  $\partial_C E(\hat{p})$  denotes the *Clarke subdifferential* of  $E$  at  $\hat{p}$  given by

$$\partial_C E(\hat{p}) := \left\{ z \in \mathbb{R}^{2m} : \langle z, u \rangle \leq \limsup_{\substack{y \rightarrow \hat{p} \\ t \rightarrow 0_+}} \frac{E(y + tu) - E(y)}{t} \quad \forall u \in \mathbb{R}^{2m} \right\},$$

see [21]. A point  $\hat{p}$  fulfilling (11) is called a *critical point* due to Toland [50]. For Gâteaux-differentiable functions  $F$  the subdifferential coincides with the gradient and the above conditions (10) and (11) are the same. For our application, the following subdifferentials/gradients at  $q = (q_x, q_y)^T$  are of interest:

$$\begin{aligned} \nabla (q_x^2 + q_y^2 + \varepsilon^2)^{\frac{1}{2}} &= \frac{1}{2} q / (q_x^2 + q_y^2 + \varepsilon^2)^{\frac{1}{2}}, \quad \varepsilon > 0, \\ \partial |q| &= \begin{cases} q/|q| & \text{for } q \neq 0, \\ \{x : |x| \leq 1\} & \text{for } q = 0. \end{cases} \end{aligned}$$

The following theorem shows some useful properties of  $E_\varphi$ .

**Theorem 4.2. (Properties of the 2D functional  $E_\varphi$ )**

Let  $m = \sum_{(i,j) \in G} w(i, j) \in \mathbb{N}$  and  $\varphi : [0, \infty) \rightarrow \mathbb{R}$ . Then the functional  $E_\varphi$  in (1) has the following properties:

- i) If  $\varphi$  is concave and monotone increasing on  $[0, \infty)$  and  $\varphi(0)$  is finite, then  $E_\varphi$  is bounded from below and coercive, i.e.,  $E_\varphi(p) \rightarrow \infty$  if  $\|p\| \rightarrow \infty$ .
- ii) If  $\varphi$  grows for  $s \rightarrow \infty$  as  $s^\alpha$ ,  $\alpha > 2$ , then  $E_\varphi$  is not bounded from below.
- iii) If  $\varphi(s) = s^2$ , then  $E_\varphi$  is not coercive.

**Proof:** i) Let us first restrict our attention to functions  $\varphi$  with  $\varphi(0) = 0$ . Let  $0 \leq a \leq b < \infty$  be arbitrary points and let  $g(x) = cx = \frac{\varphi(a+b)}{a+b} x$  denote the line through the origin and  $(a+b, \varphi(a+b))$ . Since  $\varphi$  is concave and  $\varphi(0) = 0$  we have for  $\lambda \in [0, 1]$  that  $\varphi(\lambda(a+b)) \geq$

$\lambda\varphi(a+b) = \lambda g(a+b)$ . Setting  $\lambda := a/(a+b)$  and  $\lambda := b/(a+b)$ , resp., we see that  $\varphi(a) \geq g(a)$  and  $\varphi(b) \geq g(b)$  and consequently

$$\varphi(a) + \varphi(b) \geq g(a) + g(b) = g(a+b) = \varphi(a+b).$$

In other words,  $\varphi$  is subadditive. Since  $\varphi$  is moreover monotone increasing, we get by the triangle inequality for the Euclidian norm of  $q_1, q_2 \in \mathbb{R}^2$  that

$$\varphi(|q_1 + q_2|) \leq \varphi(|q_1| + |q_2|) \leq \varphi(|q_1|) + \varphi(|q_2|).$$

Using this inequality we obtain

$$\begin{aligned} E(p) &\geq \sum_{k=1}^m \sum_{(i,j) \in G} w(i,j) \left( \varphi(|p_k|) - \varphi\left(\left|\binom{i}{j}\right|\right) \right) - \sum_{k=1}^m \sum_{l=k+1}^m \varphi(|p_k|) + \varphi(|p_l|) \\ &= m \sum_{k=1}^m \varphi(|p_k|) - m \sum_{(i,j) \in G} w(i,j) \varphi\left(\left|\binom{i}{j}\right|\right) - (m-1) \sum_{k=1}^m \varphi(|p_k|) \\ &= \sum_{k=1}^m \varphi(|p_k|) - m \sum_{(i,j) \in G} w(i,j) \varphi\left(\left|\binom{i}{j}\right|\right) \geq -m \sum_{(i,j) \in G} w(i,j) \varphi\left(\left|\binom{i}{j}\right|\right). \end{aligned} \quad (12)$$

Hence  $E$  is bounded from below and we see that  $E(p) \rightarrow \infty$  as  $\|p\| \rightarrow \infty$ .

Finally, we can also draw the conclusions in (12) for  $\varphi + C$  with a constant  $C$ . Hence the assertion i) holds true for  $\varphi$  with finite  $\varphi(0)$ .

ii) Assume w.l.o.g. that the number  $m$  of points is even. We consider  $p = (p_k)_{k=1}^m$  with  $p_1 = \dots = p_{m/2} = q$  and  $p_{m/2+1} = \dots = p_m = -q$  which leads to

$$E_\varphi(p) = \frac{m}{2} \sum_{(i,j) \in G} w(i,j) \left( \varphi\left(\left|q - \binom{i}{j}\right|\right) + \varphi\left(\left|q + \binom{i}{j}\right|\right) \right) - \left(\frac{m}{2}\right)^2 \varphi(2|q|) - c\varphi(0)$$

with some constant  $c$ . For  $|q| \rightarrow \infty$ , the first sum grows as  $m^2|q|^\alpha$  and the negative part as  $m^2\frac{2^\alpha}{4}|q|^\alpha$ . Consequently, for  $\alpha > 2$ , the function  $E_\varphi(p)$  is not bounded from below as  $|q| \rightarrow \infty$ .

iii) For  $\varphi(s) = s^2$  we obtain by straightforward computation that

$$\begin{aligned} E_\varphi(p) &= \sum_{k=1}^m \sum_{(i,j) \in G} w(i,j) \left| p_k - \binom{i}{j} \right|^2 - \sum_{k=1}^m \sum_{l=k+1}^m |p_k - p_l|^2 \\ &= \left( \sum_{k=1}^m p_{k,x} - \sum_{(i,j) \in G} i w(i,j) \right)^2 + \left( \sum_{k=1}^m p_{k,y} - \sum_{(i,j) \in G} j w(i,j) \right)^2 + C \end{aligned}$$

with some constant  $C$ . Thus  $\|p\| \rightarrow \infty$  does not imply  $E_\varphi(p) \rightarrow \infty$  and we are done.  $\square$

**Remark 4.3.** Note that the following functions  $\varphi : [0, \infty) \rightarrow \mathbb{R}$  fulfill the conditions of Theorem 4.2 i):  $\varphi(s) := s^\alpha$ ,  $\alpha \in (0, 1]$  and for  $\varepsilon > 0$

$$\varphi(s) := \begin{cases} \log(s) & \text{for } s \in [\varepsilon, \infty) \\ \frac{1}{\varepsilon} s + \log \varepsilon - 1 & \text{for } s \in [0, \varepsilon), \end{cases} \quad \varphi(s) := \begin{cases} -s^{-\alpha} & \text{for } s \in [\varepsilon, \infty), \\ \frac{\alpha}{\varepsilon^{\alpha+1}} s - \frac{\alpha+1}{\varepsilon^\alpha} & \text{for } s \in [0, \varepsilon), \end{cases} \quad \alpha > 0.$$

Next we are looking for algorithms to find a critical point of  $E$ . Condition (11) is equivalent to

$$\begin{aligned}\hat{p} + \mu\partial F(\hat{p}) &\in \hat{p} + \mu\partial G(\hat{p}), \quad \mu > 0, \\ \hat{p} &\in (I + \mu\partial F)^{-1}(I + \mu\partial G)(\hat{p}),\end{aligned}\tag{13}$$

where  $\partial F(\hat{p}) \in \partial G(\hat{p})$  means that there exists an element  $x \in \partial F(\hat{p})$  such that  $x \in \partial G(\hat{p})$ . Furthermore we have used that the resolvent  $(I + \mu\partial F)^{-1}$  of  $\mu\partial F$  is single-valued again, see [2]. Hence a necessary and sufficient condition for  $\hat{p}$  to be a critical point of  $E$  is that  $\hat{p}$  is a fixed point of  $(I + \mu\partial F)^{-1}(I + \mu\partial G)$  in (13). The following Picard iteration is called *forward-backward splitting* (FBS) algorithm, *proximal point (type) algorithm* [48] or *Moreau-Yoshida regularization* [20].

**Algorithm (FBS for DC programming)**

*Initialization:*  $p^{(0)}$

For  $r = 0, 1, \dots$  repeat until a convergence criterion is reached

$$\begin{aligned}v^{(r)} &\in \partial G(p^{(r)}), \\ p^{(r+1)} &= (I + \mu\partial F)^{-1}(p^{(r)} + \mu v^{(r)}).\end{aligned}$$

Note that the FBS algorithm is closely related to the DC algorithm of Pham and Souad [39, 40] which computes iteratively

$$v^{(r)} \in \partial G(p^{(r)}), \quad p^{(r+1)} \in \partial F^*(v^{(r)}),\tag{14}$$

where  $F^*$  denotes the conjugate function of  $F$ . If we split

$$E(p) = \underbrace{\left(\frac{1}{2}\|p\|_2^2 + \mu F(p)\right)}_{\tilde{F}(p)} - \underbrace{\left(\frac{1}{2}\|p\|_2^2 + \mu G(p)\right)}_{\tilde{G}(p)}$$

into the strongly convex parts  $\tilde{F}, \tilde{G}$  and use these functions instead of  $F, G$  in (14), we obtain exactly the FBS algorithm. For continuously differentiable  $G$  there also exists a relation of FBS to the algorithm in [34].

The following theorem ensures that convergent subsequences of  $\{p^{(r)}\}$  generated by the FBS algorithm converge to a critical point of  $E$ .

**Theorem 4.4. (Convergence of the FBS algorithm for DC)**

*The sequence  $\{p^{(r)}\}$  generated by the FBS algorithm fulfills*

$$E(p^{(r+1)}) \leq E(p^{(r)}) - \frac{1}{\mu}\|p^{(r+1)} - p^{(r)}\|_2^2\tag{15}$$

*Any convergent subsequence of  $\{p^{(r)}\}$  converges to a critical point of  $E$ .*

The estimate (15) was proved in [48]. By Theorem 4.2 the sequence  $\{p^{(r)}\}$  generated by the FBS algorithm is bounded so that it follows from Zangwill's convergence theorem [57] that any convergent subsequence converges to a critical point of  $E$ .

Finally, let us have a closer look at the FBS algorithm for our special setting. A subgradient  $v^{(r)} = (v_k^{(r)})_{k=1}^m$  of  $G$  at  $p^{(r)}$  is given by

$$v_k^{(r)} = \sum_{\substack{l=1 \\ p_l^{(r)} \neq p_k^{(r)}}}^m \frac{p_k^{(r)} - p_l^{(r)}}{|p_k^{(r)} - p_l^{(r)}|}$$

with the corresponding modifications if we use the  $\varepsilon$ -smoothed variant. Set  $y^{(r)} := p^{(r)} + \mu v^{(r)}$ . Next,  $p^{(r+1)} = (I + \mu \partial F)^{-1}(y^{(r)})$  requires the minimization of

$$\frac{1}{2} \|p - y^{(r)}\|_2^2 + \mu F(p)$$

which can be done componentwise for each  $p_k$ , i.e.,

$$p_k^{(r+1)} = \operatorname{argmin}_q \left\{ \frac{1}{2} \|q - y_k^{(r)}\|_2^2 + \mu f(q) \right\}, \quad k = 1, \dots, m. \quad (16)$$

Note that in one dimension this problem can be solved analytically, see [32]. The simplest way to solve (16) numerically is to use a (sub)gradient algorithm, see [35]. If one uses only one inner iteration the overall FBS algorithm is just a (sub)gradient algorithm applied to  $E$ :

**Algorithm ((Sub)gradient algorithm applied to  $E$ )**

*Initialization:*  $p^{(0)}$

For  $r = 0, 1, \dots$  repeat until a convergence criterion is reached

$$p_k^{(r+1)} = p_k^{(r)} - \tau \left( \sum_{\substack{(i,j) \in G \\ (i,j)^T \neq p_k^{(r)}}} w(i,j) \frac{p_k^{(r)} - \binom{i}{j}}{|p_k^{(r)} - \binom{i}{j}|} - \sum_{\substack{l=1 \\ p_l^{(r)} \neq p_k^{(r)}}}^m \frac{p_k^{(r)} - p_l^{(r)}}{|p_k^{(r)} - p_l^{(r)}|} \right). \quad (17)$$

**Remark 4.5.** Another possibility to solve (16) in its differentiable variant is the *Newton method* which computes in each iteration

$$q^{(s+1)} = q^{(s)} + J_H(q^{(s)})^{-1} H(q^{(s)}),$$

where

$$\begin{aligned} H(q) &:= q - y_k^{(r)} + \mu \sum_{\substack{(i,j) \in G \\ p_k^{(r)} \neq \binom{i}{j}}} w(i,j) \frac{p_k^{(r)} - \binom{i}{j}}{|p_k^{(r)} - \binom{i}{j}|}, \\ J_H(q)^{-1} &:= \frac{1}{j_{20}j_{02} - j_{11}^2} \begin{pmatrix} 1 + j_{02} & -j_{11} \\ -j_{11} & 1 + j_{20} \end{pmatrix}, \\ j_{ab} &:= \mu \sum_{\substack{(i,j) \in G \\ q \neq \binom{i}{j}}} w(i,j) \frac{(q_x - i)^a (q_y - j)^b}{|q - \binom{i}{j}|^3}, \quad a, b \in \{0, 1, 2\}. \end{aligned}$$

This requires the computation of two more sums involving  $w(i, j)$ .

## 4.2 Fast Summation by NFFT

The above algorithms involve the computation of special sums of the form

$$s_1(p_k) := \sum_{\substack{(i,j) \in G \\ (i,j)^T \neq p_k}} w(i,j) \frac{p_k - \binom{i}{j}}{|p_k - \binom{i}{j}|} \quad \text{and} \quad s_2(p_k) := \sum_{\substack{l=1 \\ p_l \neq p_k}}^m \frac{p_k - p_l}{|p_k - p_l|} \quad (18)$$

for  $k = 1, \dots, m$  in each iteration. The straightforward computation of these sums is too time consuming to make the algorithm work for applications with a large number  $m$  of black pixels.

We propose to compute the first sum by bilinear interpolation if we are looking for continuous particle locations, see Remark 3.2. In other words, we precompute  $s_1(s, t)$ ,  $(s, t) \in G$ . Since this requires a (noncyclic) convolution of  $w(i, j)$  with  $i/(i^2 + j^2)^{\frac{1}{2}}$ , such precomputation can be computed very efficiently via 3 FFTs of size  $2n \times 2n$  in  $\mathcal{O}(n^2 \log n)$  arithmetic operations. The bilinear interpolation at the points  $p_k$ ,  $k = 1, \dots, m$  requires  $\mathcal{O}(m)$  arithmetic operations. In this paper, we suggest to compute the second sum by fast summation methods based on the fast Fourier transform for non-equispaced data (NFFT) developed in [13, 41]. Material about the NFFT can be found, e.g., in [12, 42]. Alternatively one could use *fast multipole methods* [8, 18], see also *fast mosaic-skeleton matrix multiplication* [51] or *fast  $\mathcal{H}$ -matrix multiplication* [19]. All these methods require  $\mathcal{O}(m \log m)$  arithmetic operations. Since these methods are designed for radial kernels, i.e., for sums of the form

$$s(p_k) := \sum_{l=1}^m \alpha_l \mathcal{K}(p_k - p_l) = \sum_{l=1}^m \alpha_l K(|p_k - p_l|), \quad k = 1, \dots, m, \quad (19)$$

we decompose  $s_2$  for  $p_k = (p_{k,x}, p_{k,y})$  as

$$\begin{aligned} s_2(p_k) &:= \sum_{l=1}^m \frac{p_k - p_l}{|p_k - p_l|} \\ &= p_k \sum_{l=1}^m \frac{1}{|p_k - p_l|} - \sum_{l=1}^m \frac{p_l}{|p_k - p_l|} \\ &= p_k \sum_{l=1}^m \frac{1}{|p_k - p_l|} - \binom{1}{0} \sum_{l=1}^m \frac{p_{l,x}}{|p_k - p_l|} - \binom{0}{1} \sum_{l=1}^m \frac{p_{l,y}}{|p_k - p_l|}. \end{aligned} \quad (20)$$

Note that the first sum  $s_1$  can be decomposed similarly.

In the following, we sketch our NFFT-based summation algorithm for the computation of each sum in (20) with the kernel  $K(x) = 1/|x|$ . For more information, including error estimates we refer to [13, 41]. We assume that our image is centered at the origin and that the image size is scaled by a factor  $\frac{\sqrt{2}(1-\varepsilon_B)}{4n} \geq \frac{1}{4n}$  with  $0 < \varepsilon_B \ll 1$  so that  $|p_k - p_l| \leq \frac{1-\varepsilon_B}{2}$ . We regularize  $\mathcal{K}$  near  $(0, 0)$  and near the boundary  $\pm 1/2$  in each direction to obtain a bivariate *smooth 1-periodic kernel*  $\mathcal{K}_R$  in the Sobolev space  $H^p(\mathbb{T}^2)$ , i.e.,

$$\mathcal{K}_R(x) = \begin{cases} K_I(|x|) & \text{if } |x| \in [0, \frac{\varepsilon_I}{2}], \\ K_B(|x|) & \text{if } |x| \in [\frac{1-\varepsilon_B}{2}, \frac{1}{2}], \\ K_B(\frac{1}{2}) & \text{if } |x| \geq \frac{1}{2}, \\ K(|x|) & \text{otherwise,} \end{cases}$$

with  $0 < \varepsilon_I \ll 1$ . This regularization can be done in different ways, e.g., by two-point Taylor interpolation with polynomials, see [13]. If  $p$  is large enough,  $\mathcal{K}_R$  can be approximated with only a small error by its truncated Fourier series

$$\mathcal{K}_{RF}(x) := \sum_{j \in J_N} b_j e^{2\pi i \langle j, x \rangle}, \quad b_j := \frac{1}{N^2} \sum_{k \in J_N} \mathcal{K}_R \left( \frac{k}{N} \right) e^{-2\pi i \langle j, k \rangle / N}, \quad (21)$$

where  $J_N := \{-N/2, \dots, N/2 - 1\} \times \{-N/2, \dots, N/2 - 1\}$  for even  $N$ . In [41] the authors propose to choose  $N \sim \sqrt{pm}$  and  $\varepsilon_I \sim 2p/N$ . Now the original kernel  $\mathcal{K}$  can be written

$$\mathcal{K} = (\mathcal{K} - \mathcal{K}_R) + (\mathcal{K}_R - \mathcal{K}_{RF}) + \mathcal{K}_{RF} = \mathcal{K}_{NE} + \mathcal{K}_{ERR} + \mathcal{K}_{RF}.$$

Since  $\mathcal{K}_{ERR}$  is small, we approximate  $\mathcal{K}$  by  $\mathcal{K}_{NE} + \mathcal{K}_{RF}$  and compute instead of  $s(p_k)$  in (19) the sums

$$\tilde{s}(p_k) := \sum_{l=1}^m \alpha_l \mathcal{K}_{NE}(p_k - p_l) + \sum_{l=1}^m \alpha_l \mathcal{K}_{RF}(p_k - p_l). \quad (22)$$

Indeed this can be done in a fast way by the following two steps:

- *Near field computation*

Suppose that the points  $p_k$  are sufficiently uniformly distributed so that each ball of radius  $\varepsilon_I/2$  contains at most a fixed number of  $\nu$  points. Since  $|p_k - p_l| < \frac{1-\varepsilon_B}{2}$  and  $\text{supp}(\mathcal{K}_{NE}) \cap [-\frac{1-\varepsilon_B}{2}, \frac{1-\varepsilon_B}{2}] \times [-\frac{1-\varepsilon_B}{2}, \frac{1-\varepsilon_B}{2}] = [-\frac{\varepsilon_I}{2}, \frac{\varepsilon_I}{2}] \times [-\frac{\varepsilon_I}{2}, \frac{\varepsilon_I}{2}]$  the evaluation of the first sum in (22) requires  $\leq m\nu$ , i.e.,  $\mathcal{O}(m)$  arithmetic operations.

- *NFFT based summation*

By (21) the second sum in (22) can be rewritten as

$$\begin{aligned} \sum_{l=1}^m \alpha_l \mathcal{K}_{RF}(p_k - p_l) &= \sum_{l=1}^m \alpha_l \sum_{j \in J_N} b_j e^{2\pi i \langle j, p_k - p_l \rangle} \\ &= \sum_{j \in J_N} b_j \left( \sum_{l=1}^m \alpha_l e^{-2\pi i \langle j, p_l \rangle} \right) e^{2\pi i \langle j, p_k \rangle}. \end{aligned}$$

The expression  $c_j$ ,  $j \in J_N$  in the inner brackets can be computed by a bivariate NFFT. This will be followed by  $N^2$  multiplications with  $b_l$  and completed by another bivariate NFFT to compute the outer summation with the complex exponentials. It can be shown that this summation procedure requires  $\mathcal{O}(m \log m)$  arithmetic operations.

**Remark 4.6.** The fast summation method can be applied for other kernels as well, e.g., for

$$s_1(p_k) := \sum_{(i,j) \in G} w(i,j) \frac{p_k - \binom{i}{j}}{|p_k - \binom{i}{j}|^2 + \varepsilon^2} \quad \text{and} \quad s_2(p_k) := \sum_{l=1}^m \frac{p_k - p_l}{|p_k - p_l|^2 + \varepsilon^2} \quad (23)$$

which will be used in our numerical examples for the logarithmic setting  $\varphi = \log$ .

## 5 Numerical Examples

In this section, we demonstrate the performance of our algorithms by numerical examples. Our algorithms were implemented in C using double precision arithmetic and tested on a dual core desktop with two 2.4 GHz processors and 3 GB physical memory.

The numerical results were produced by the subgradient algorithm (17). Until stated otherwise no smoothing parameter  $\varepsilon$  was used. Our fast summation algorithm applies the NFFT-package from <http://www-user.tu-chemnitz.de/~potts/nfft> which incorporates the FFTW. To guarantee that all points remain in the image domain during our iterative procedure, we project back to the image domain at the end of every iteration.

We compare our stippling results with those obtained by the method of Balzer et al. [4]. The dithering results, where the points have to meet grid positions, are compared with those computed by the algorithms of Floyd-Steinberg [14], Ostromoukhov [37], Pang et al. [38], Zhou et al. [58] and Purgathofer et al. [43]. While implementations of the algorithms of Ostromoukhov and Balzer et al. are freely available, we have implemented the remaining methods from scratch.

**Remark 5.1.** In particular, we compare our approach for the functional  $E_\varphi$  with  $\varphi(s) = s$  with the gradient descent algorithm for the functional  $E_{\varphi,\varepsilon}$  with the logarithmic function  $\varphi(s) = \log(s)$ . To this end, we just have to replace the involved sums by those in (23). This approach was suggested from an electrostatical point of view in [44] and leads to qualitatively very good results. However, also for this case our computations involve two novelties:

- i) The involved large sums are computed by a fast summation algorithm on a usual PC now instead on a Graphics Processing Unit (GPU) without fast algorithm.
- ii) While the stippling results are achieved by a bilinear interpolation of the first sum, the halftoning method, which requires point positions on the grid, is simply obtained by applying a better approximation of the first sum. As better approximation we have again used our fast summation approach. In [44] a more complex method including a third function to concentrate the dots in the vicinity of the grid points was suggested.

**Continuous pixel location.** Our stippling results were obtained by computing the first sum in (18), resp. (23) by bilinear interpolation which relaxes the grid attraction. The second sum was computed by our fast summation algorithm.

**Grid pixel location.** To place the points at/near grid positions we need a better approximation of the first sums in (18), resp. (23). In our computations, we use a fast summation algorithm for the first sum, too, which includes NFFTs and ordinary FFTs now due to the involved grid points  $\binom{i}{j}$ . In our computations with the logarithmic kernel we set  $\varepsilon := 0.2$  in the first sum. Moreover, the minimization procedure for the logarithmic setting tends to end in a local minimum yielding suboptimal results. Therefore we introduce a 'shaking or warming procedure': Every ten iterations, each pixel is moved into a random direction with a random magnitude, where the maximal displacement 'cools down' with an increasing iteration number. More precisely, the displacement at the  $k$ -th iteration is set to

$$c \cdot e^{-\frac{3k}{K}}, \quad c := \max\left(0, \frac{\log_2(3K) - 6}{10}\right),$$

where  $K$  is the total number of iterations to be done.

In our experiments, the method with the logarithmic function places nearly all points  $p_k$  at grid positions (up to a small error), while for our method with  $\varphi(s) = s$  approximately two third of the points lie on the grid. We push the remaining points to the nearest grid positions.

**First example.** In our first example, we evaluate the visual quality of the images produced by our method in comparison with other methods both for the continuous and the grid setting for a natural image. Fig. 6 shows the stippling results for the algorithm of Balzer et al. and for the continuous methods with  $\varphi(s) = s$  ( $\tau = 0.01$ ) and  $\varphi(s) = \log(s)$  ( $\tau = 0.1$ ). Note that the different time step sizes  $\tau$  do not mean that the first method requires more iterations than the other since the involved sums have different magnitudes. All three methods show a good performance. The algorithm with  $\varphi(s) = \log(s)$  tends to arrange dots in large uniform regions in an energetically optimal hexagonal structure. Such 'regularity artefacts' which also appear if Lloyd's algorithm is not stopped at a suitable iteration step may attract the attention of the observer and are therefore undesirable in some applications. In [44] an additional jittering procedure was suggested to avoid this phenomenon. Note that with our present implementation one iteration for this image requires a runtime of approximately a second and one needs less than 1000 iterations to get an image with a very good visual quality.

Next, we compare various dithering approaches in Figs. 10 and 11 (left). The error diffusion methods of Floyd and Steinberg and of Ostromoukhov show the expected disturbing patterns. The iterative algorithm of Pang et al. with the parameters proposed in [38] has the structure enhancing properties which are often considered as its strength but fails to approximate the overall image very well, see Fig. 11 (left).

At a certain distance an observer should not recognize the difference between the stippled/halftoned image and the original one. Therefore we analyze the Gaussian scale space properties of our images. For this, the original and the stippled images are convolved with a Gaussian of standard deviation  $\sigma$ . The results for  $\sigma = 1$  are depicted in Fig. 7 together with the differences between the convolved original image and the results. The image resulting from the continuous method with  $\varphi(s) = \log(s)$  is superior here. As it can be seen in Fig. 6 (top right), the algorithm of Balzer et al. produces some artefacts at the boundary. Therefore, for comparing the scale space properties, we mirrored the image prior to the application of the method of Balzer et al. and applied the Gaussian blurring to the resulting image. Nevertheless, our method with  $\varphi(s) = s$  gives a better result than those of Balzer et al. Finally, Figs. 8 and 9 compare the Peak Signal to Noise Ratio (PSNR) between the convolved stippled/dithered and original images for increasing standard deviations  $\sigma$ . The above findings were confirmed by this plot. For a reference image  $u$  and the computed image  $\tilde{u}$  the PSNR is defined by

$$PSNR := 20 \log_{10} \left( \frac{1}{\sqrt{MSE}} \right), \quad MSE := \frac{1}{n_x n_y} \sum_{(i,j) \in G} (u(i,j) - \tilde{u}(i,j))^2.$$

**Second example.** In our second example, we dither a Gaussian of standard deviation  $\sigma = 40$ . Fig. 12 shows the results for the stippling methods. All three methods perform very well. We added the PSNR versus standard deviation plot in Fig. 14. Next, Figs. 13 and 11 (right) contain the results for the various dithering methods. As we see here, the Floyd-Steinberg method cannot keep the rotation invariance of the Gaussian. Moreover,



Figure 6: Top left: Original  $256 \times 256$  image. Top right: Stippling result by the algorithm of Balzer et al. Bottom: Stippling results by the continuous method with  $\varphi(s) = s$  (left) and  $\varphi(s) = \log(s)$  (right). Here we use  $m = 30150$  points.

some artefacts at special gray values are visible. Ostromoukhov's approach and the algorithm of Zhou et al. perform better, but there is still a lack in rotation invariance and there are artefacts at special gray values in Ostromoukhov's algorithm. The method of Purgathofer et al. spreads the pixels too much. The result for the method of Pang et al. with the parameters proposed in [38] shows again that the good structure preservation of the algorithm is paid with a lack in tone preservation. The scale space behavior of the various dithering results for the Gaussian can be found in Fig. 15.

**Third example.** In our third example, we distribute black pixels equally within an image of constant gray value  $u_c \in [0, 1]$  by our methods. It is generally accepted that the ideal halftone image has a special *blue noise spectrum* [52, 53]. To this end, the Fourier spectrum of the dithered image without its lowest frequency coefficient (in other words, the Fourier spectrum of the difference between the dithered image and the original one) is radially averaged which

ideally results in the curve depicted in Fig. 16. The curve shows a low energy content up to the principal frequency  $f_p := \sqrt{0.5 - |u_c - 0.5|}$ , where a sharp transition region occurs. Above this frequency blue noise behavior is visible, but much lower than the peak. Fig. 17 shows the results obtained by the stippling method with  $\varphi(s) = s$  and  $\varphi(s) = \log(s)$ , and by its dithering variants together with the corresponding blue noise spectrum. For robustness we have averaged the power spectra of ten randomly selected patches with a sufficient distance to the image boundary as it is common, see [6]. For the blue noise behavior of other stippling/dithering methods we refer to [44].

**Remark 5.2.** We like to stress that optional extensions as *edge enhancement* by applying a preprocessing step to the initial image  $u$ , *different pixel sizes* by varying  $m$ , resp.  $\lambda$  or the extension to the case of RGB or CMY *colour models* by treating the channels separately can be included into our computations without difficulties.

## 6 Conclusions

This paper is a first attempt to treat the problem of halftoning as a global optimization problem of the form (1). We have proved some facts concerning this functional with  $\varphi(s) = s$ , but there remain various questions starting from the appropriate choice of the functional  $\varphi$  and its properties up to algorithms to find a local/global minimizer. It would be interesting if other functions as, e.g., those considered in [36] are also useful in our context. In this paper, we have combined a simple subgradient algorithm with a fast summation algorithm for radial kernels. However, we have sketched that other algorithms, e.g., an FBS method applied to the ordinary splitting of  $E$  together with a Newton method for the inner iterations can be used as well. We remark that there may also work a Weiszfeld-like algorithm. The Weiszfeld algorithm and its modified/accelerated versions are the best-known algorithms for solving the problem of the spatial median [27, 56]. The examination of different algorithmical approaches in connection with the appropriate adaptation of our fast summation algorithm is object of future research. To this end, we have to further speed up our adapted fast summation algorithm. In [44] some of the authors have implemented the electrostatic halftoning on a Graphics Processing Units (GPU). In our future work we plan to combine our fast summation algorithm with GPU programming.

**Acknowledgment.** Many thanks to S. Setzer (UCLA) and D. Potts (TU Chemnitz) for fruitful discussions. Our research was partially funded by the Deutsche Forschungsgemeinschaft (Project STE 571/9-1).

## References

- [1] M. Analoui and J. P. Allebach. Model-based halftoning using direct binary search. *Proc. SPIE*, 1666:96–108, 1992.
- [2] J.-P. Aubin and H. Frankowska. *Set-valued Analysis*. Birkhäuser, Boston, 1990.
- [3] M. Bacák and J. M. Borwein. On difference convexity of locally Lipschitz functions. *Preprint, University of Newcastle*, 2010.

- [4] M. Balzer, T. Schlömer, and O. Deussen. Capacity-constrained point distributions: A variant of Lloyd’s method. *ACM Transactions on Graphics*, 28(3):Article 86, 2009.
- [5] F. A. Baqai and J. P. Allebach. Halftoning via direct binary search using analytical and stochastic printer models. *IEEE Transactions on Image Processing*, 12(1):1–15, 2003.
- [6] M. S. Bartlett. *An Introduction to Stochastic Processes with Special Reference to Methods and Applications*. Cambridge University Press, Cambridge, 1978.
- [7] B. E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. In *Conference Record, IEEE International Conference on Communications*, volume 1, pages (26–11)–(26–15), 1973.
- [8] R. K. Beatson and G. N. Newsam. Fast evaluation of radial basis functions: I. *Computers and Mathematics with Applications*, 24(12):7–19, 1992.
- [9] J. Chang, B. Alain, and B. Ostromoukhov. Structure-aware error diffusion. *ACM Transactions on Graphics*, 28(5):Article 162, 2009.
- [10] Q. Du, M. Emelianenko, and L. Ju. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. 44(1):102–119, 2006.
- [11] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [12] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM Journal on Scientific and Statistical Computing*, 14(6):1368–1393, 1993.
- [13] M. Fenn and G. Steidl. Fast NFFT based summation of radial functions. *Sampling Theory in Signal and Image Processing*, 3(1):1–28, 2004.
- [14] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *Proc. Society of Information Display*, 17:75–77, 1976.
- [15] R. Geist, R. Reynolds, and D. Suggs. A Markovian framework for digital halftoning. *ACM Transactions on Graphics*, 12(2):136–159, 1993.
- [16] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, 2008.
- [17] W. M. Goodall. Television by pulse code modulation. *Bell System Technical Journal*, 30:33–49, 1951.
- [18] L. F. Greengard and V. Rokhlin. Fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.
- [19] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing*, 62(2):89–108, 1999.
- [20] A. Hamdi. A Moreau-Yoshida regularization of a difference of two convex functions. *Applied Mathematics E-Notes*, 5:164–170, 2005.

- [21] J.-B. Hiriart-Urruty. Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. In *Convexity and Duality in Optimization*, volume 256 of *Lecture Notes in Economical and Mathematical Systems*, pages 37–70. Springer, 1985.
- [22] J.-B. Hiriart-Urruty. From convex optimization to nonconvex optimization: Necessary and sufficient conditions for global optimality. In *Nonsmooth Optimization and Related Topics*, pages 219–240. Plenum Press, 1989.
- [23] J.-B. Hiriart-Urruty. How to regularize a difference of convex functions. *Journal of Mathematical Analysis and Applications*, 162:196–209, 1991.
- [24] P. W. M. Ilbery. Force field halftoning, U.S. patent 6124844, 2000.
- [25] J. F. Jarvis, C. N. Judice, and W. H. Ninke. A survey of techniques for the display of continuous tone pictures on bilevel displays. *Computer Graphics and Image Processing*, (5):13–40, 1976.
- [26] J. F. Jarvis and C. S. Roberts. A new technique for displaying continuous tone images on a bilevel display. *IEEE Transactions on Communications*, (24):891–898, 1976.
- [27] T. Kärkkäinen and S. Äyrämö. On computation of spatial median for robust data mining. In R. Schilling, W. Haase, J. Periaux, H. Baier, and G. Bugeada, editors, *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN*, pages 1–14, Munich, 2005.
- [28] J. C. Kiefer. Uniqueness of locally optimal quantizer for log-concave density and convex error function. *IEEE Transactions on Information Theory*, 29(1):42–47, 1983.
- [29] K. T. Knox. Edge enhancement in error diffusion. In *Advance Printing of Paper Summaries, SPSE's 42nd Annual Conference*, pages 310–313, Boston, 1989.
- [30] D. E. Knuth. Digital halftones by dot diffusion. *ACM Transactions on Graphics*, 6(4):245–273, 1987.
- [31] T. Kolling and A. Keller. Efficient illumination by high dynamic range images. In *Proceedings of the 14th Eurographics Workshop on Rendering*, volume 44 of *ACM International Conference Proceeding Series*, pages 45–50, 2003.
- [32] Y. Li and S. Osher. A new median formula with applications to PDE based denoising. *Communications in Mathematical Sciences*, 7(3):741–754, 2009.
- [33] S. P. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [34] H. Mine and M. Fukushima. A minimization method for the sum of a convex function and a continuously differentiable function. *Journal of Optimization Theory and Applications*, 33:9–23, 1981.
- [35] Y. Nesterov. *Introductory Lectures on Convex Optimization: Basic Course*. Kluwer, Boston, 2003.

- [36] M. Nikolova. Analysis of the recovery of edges in images and signals by minimizing non-convex regularized least-squares. *SIAM Journal on Multiscale Modeling and Simulation*, 4(3):960–991, 2005.
- [37] V. Ostromoukhov. A simple and efficient error-diffusion algorithm. In E. Fiume, editor, *Proc. SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 567–572, Los Angeles, 2001.
- [38] W.-M. Pang, Y. Qu, T.-T. Wong, D. Cohen-Or, and P.-A. Heng. Structure-aware halftoning. *ACM Transactions on Graphics*, 27(3):89:1–89:8, 2008.
- [39] T. Pham Dinh and H. A. Le Thi. Convex analysis approaches to DC programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):287–367, 1997.
- [40] T. Pham Dinh and E. B. Soaud. Algorithms for solving a class of nonconvex optimization problems: Methods of subgradient. In *Fermat Days 85: Mathematics of optimization*, pages 249–270, North Holland Mat. Stud. 129, 1986. Elsevier.
- [41] D. Potts, G. Steidl, and A. Nieslony. Fast convolution with radial kernels at nonequispaced knots. *Numerische Mathematik*, 98(2):329–351, 2004.
- [42] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, pages 247–270. Birkhäuser, 2001.
- [43] W. Purgathofer, R. F. Tobler, and M. Geiler. Forced random dithering: Improved threshold matrices for ordered dithering. In *Proceedings of 1st IEEE International Conference on Image Processing*, volume 2, pages 1032–1035, Austin, Texas, 1994.
- [44] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert. Electrostatic halftoning. Report No. 260, Dept. of Mathematics, Saarland University, Saarbrücken, Germany, 2010.
- [45] A. Secord. Weighted Voronoi stippling. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, pages 37–43. ACM Press, 2002.
- [46] R. L. Stevenson and G. R. Arce. Binary display of hexagonally sampled continuous-tone images. *Journal of the Optical Society of America*, A 2(7):1009–1013, 1985.
- [47] P. Stucki. MECCA - a multiple-error correcting computation algorithm for bilevel hard-copy reproduction. Tech. Report RZ1060, IBM Research Lab, Zürich, Switzerland, 1981.
- [48] W. Sun, R. J. B. Sampaio, and M. A. B. Candido. Proximal point algorithm for minimization of DC function. *Journal of Computational Mathematics*, 21(4):451–462, 2003.
- [49] V. Surazhsky, P. Alliez, and C. Gotsman. Isotropic remeshing of surfaces: A local parameterization approach. In *Proceedings of the 12th International Meshing Roundtable*, pages 215–224, 2003.
- [50] J. F. Toland. A duality principle for non-convex optimization and the calculus of variations. *Archive for Rational Mechanics and Analysis*, 71(1):41–61, 1979.
- [51] E. E. Tyrtshnikov. Mosaic-skeleton approximation. *Calcolo*, 33:47–57, 1996.

- [52] R. Ulichney. Digital halftoning. *MIT Press*, 1987.
- [53] R. Ulichney. Dithering with blue noise. *Proceedings of the IEEE*, 76(1):56–79, 1988.
- [54] D. Vanderhaeghe and V. Ostromoukhov. Polyomino-based digital halftoning. In P. Isaías, editor, *IADIS International Conference on Computer Graphics and Visualization*, pages 11–18, 2008.
- [55] L. Velho, A. C. Frery, and J. Gomes. *Image Processing for Computer Graphics and Vision*. Springer, London, 2009.
- [56] E. Weiszfeld. Sur le point pour lequel les sommes des distances de  $n$  points donnés et minimum. *Tôhoku Mathematical Journal*, 43:355–386, 1937.
- [57] W. I. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice-Hall, 1969.
- [58] B. Zhou and X. Fang. Improving mid-tone quality of variable-coefficient error diffusion using threshold modulation. *ACM Transactions on Graphics*, 22(3):437–444, 2003.

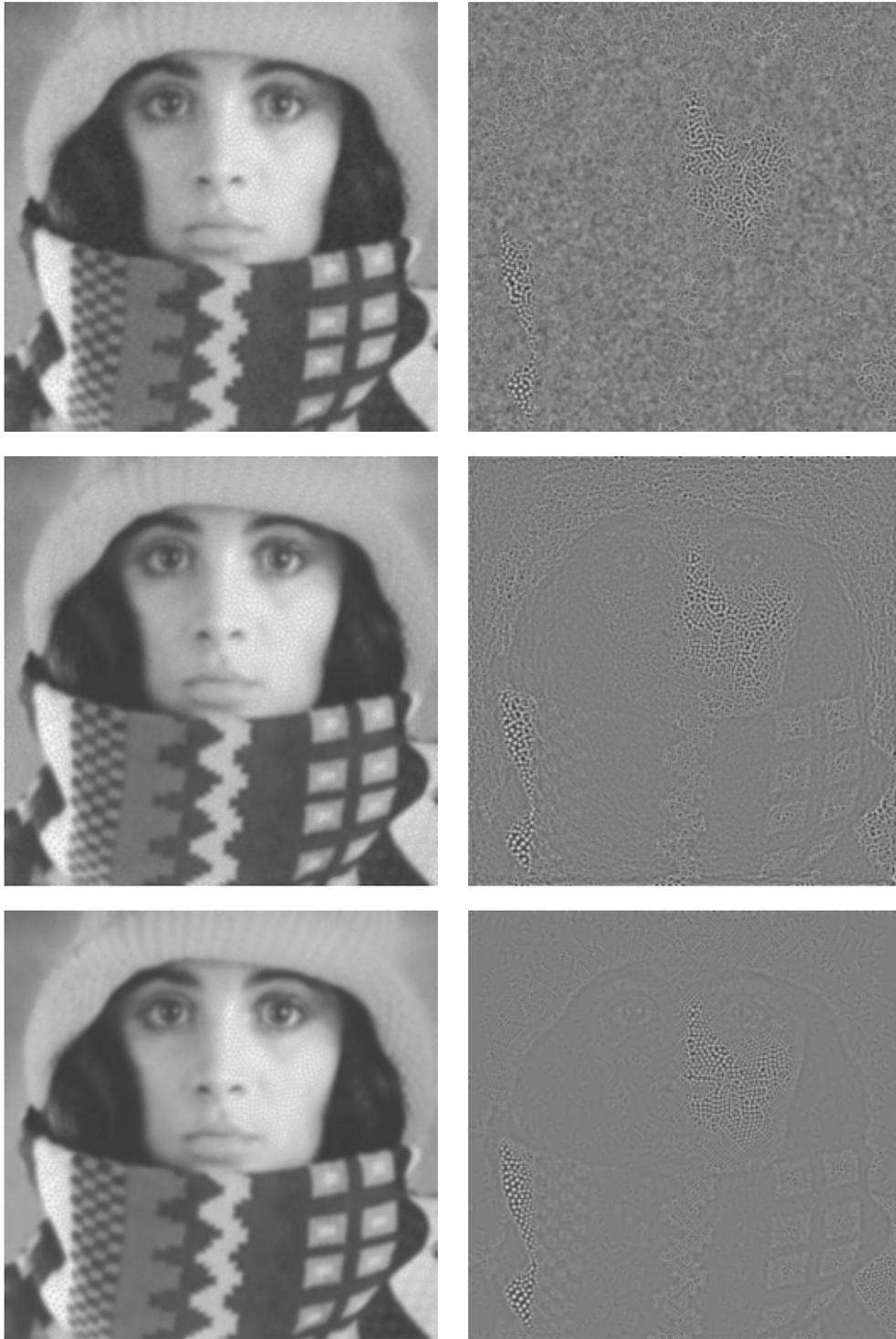


Figure 7: Stippled images convolved with a Gaussian of standard deviation  $\sigma = 1$  (left) and the error between the convolved original and stippled images (right). Top: Method of Balzer et al. Middle: Continuous method with  $\varphi(s) = s$ . Bottom: Continuous method with  $\varphi(s) = \log(s)$ .

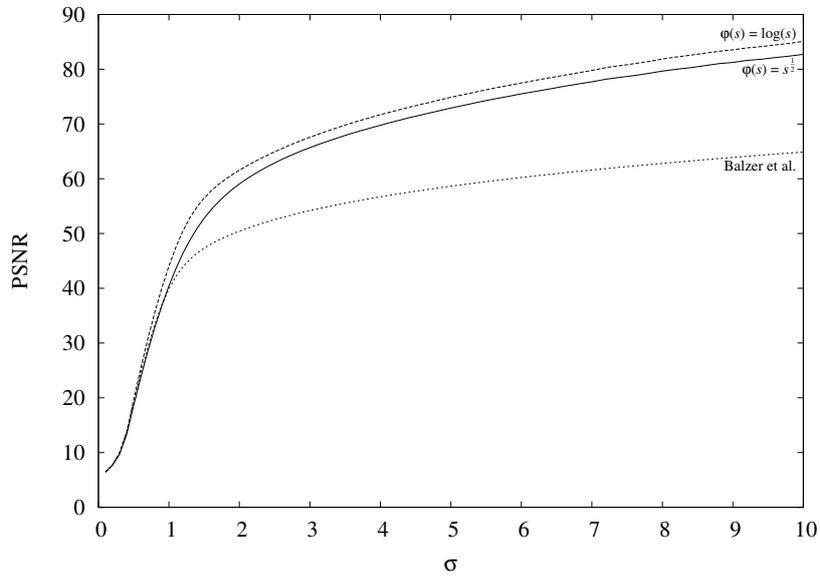


Figure 8: PSNR for stippling results in Fig. 6 under the convolution with Gaussians of various standard deviations  $\sigma$ .

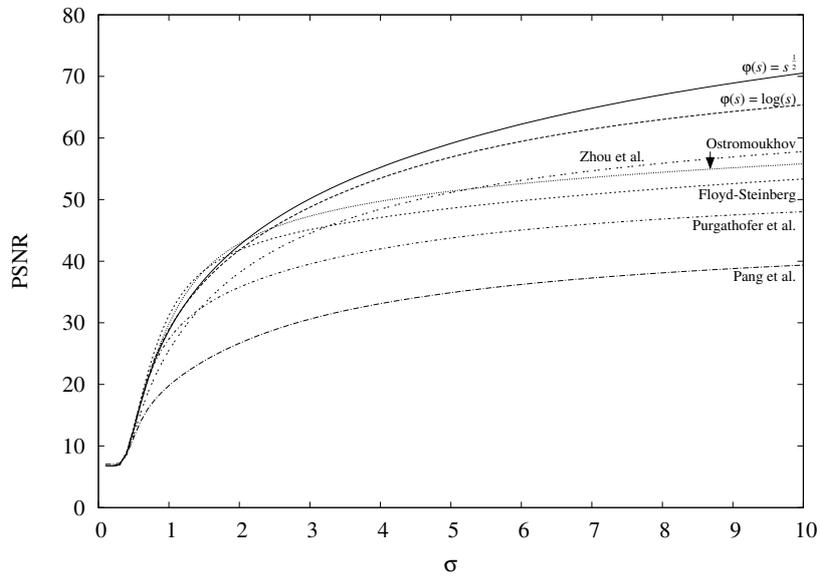


Figure 9: PSNR for dithering results in Figs. 10 and 11 under the convolution with Gaussians of various standard deviations  $\sigma$ .



Figure 10: Dithering results by our grid method with  $\varphi(s) = s$  (top left) and  $\varphi(s) = \log(s)$  (top right), the Floyd-Steinberg algorithm (middle left), Ostromoukhov's algorithm (middle right), the algorithm of Purgathofer et al. (bottom left) and the algorithm of Zhou et al. (bottom right).

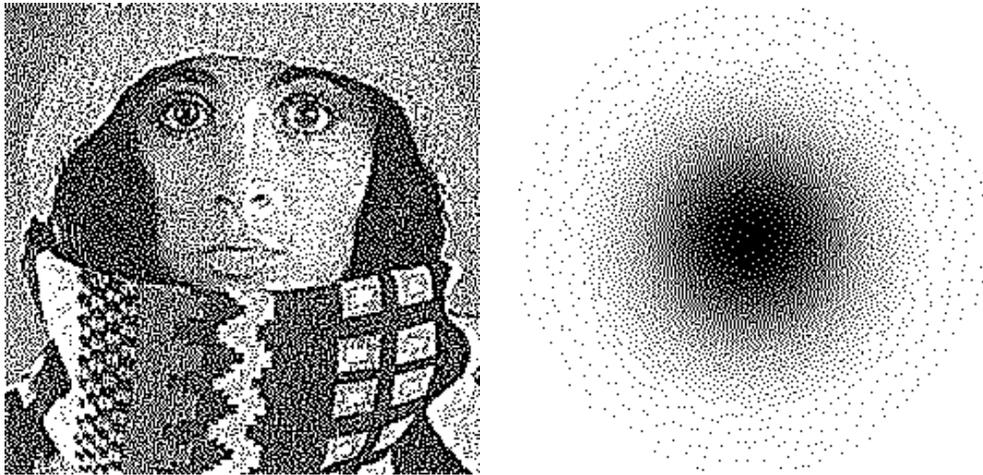


Figure 11: Dithering results by the algorithm of Pang et al. with the parameter setting proposed in [38] for the image in Fig. 6 (left) and the Gaussian in Fig. 12 (right).

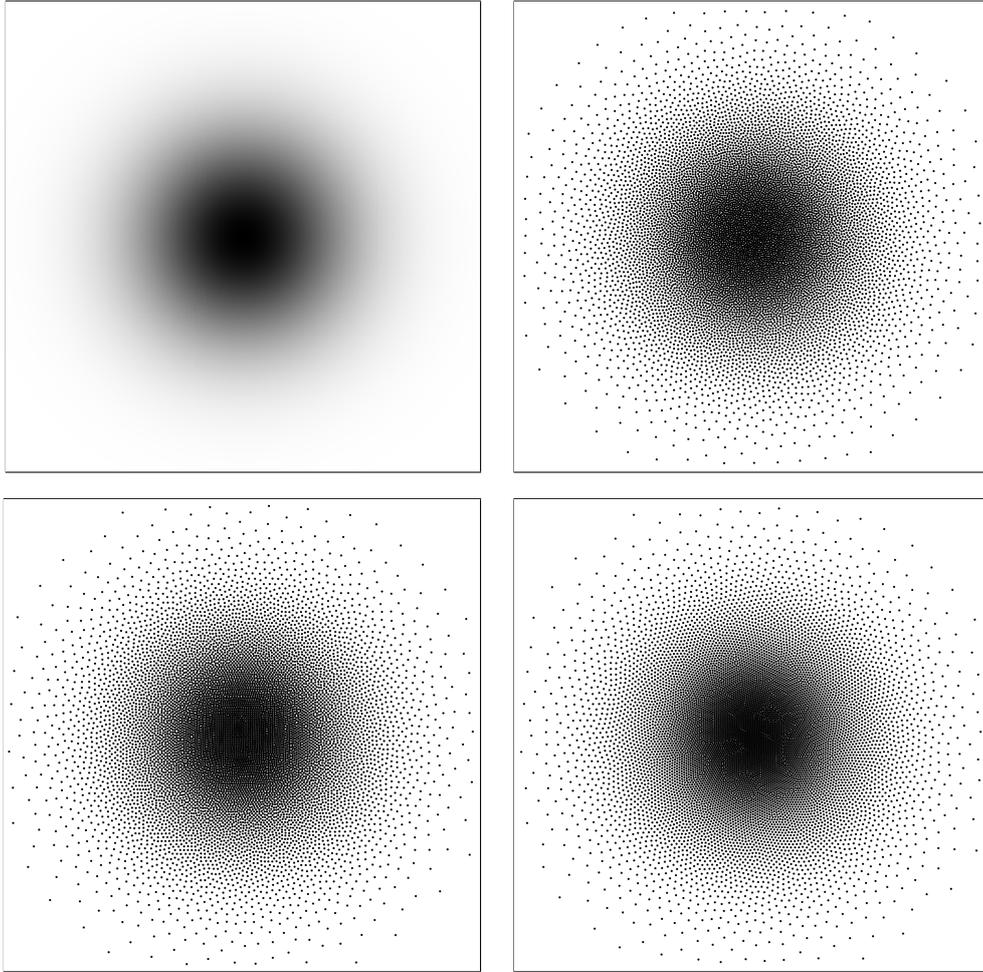


Figure 12: Top left: Original  $256 \times 256$  image. Top right: Stippling result by the algorithm of Balzer et al. Bottom: Stippling results by the continuous method with  $\varphi(s) = s$  (left) and  $\varphi(s) = \log(s)$  (right). Here we use  $m = 10023$  points.

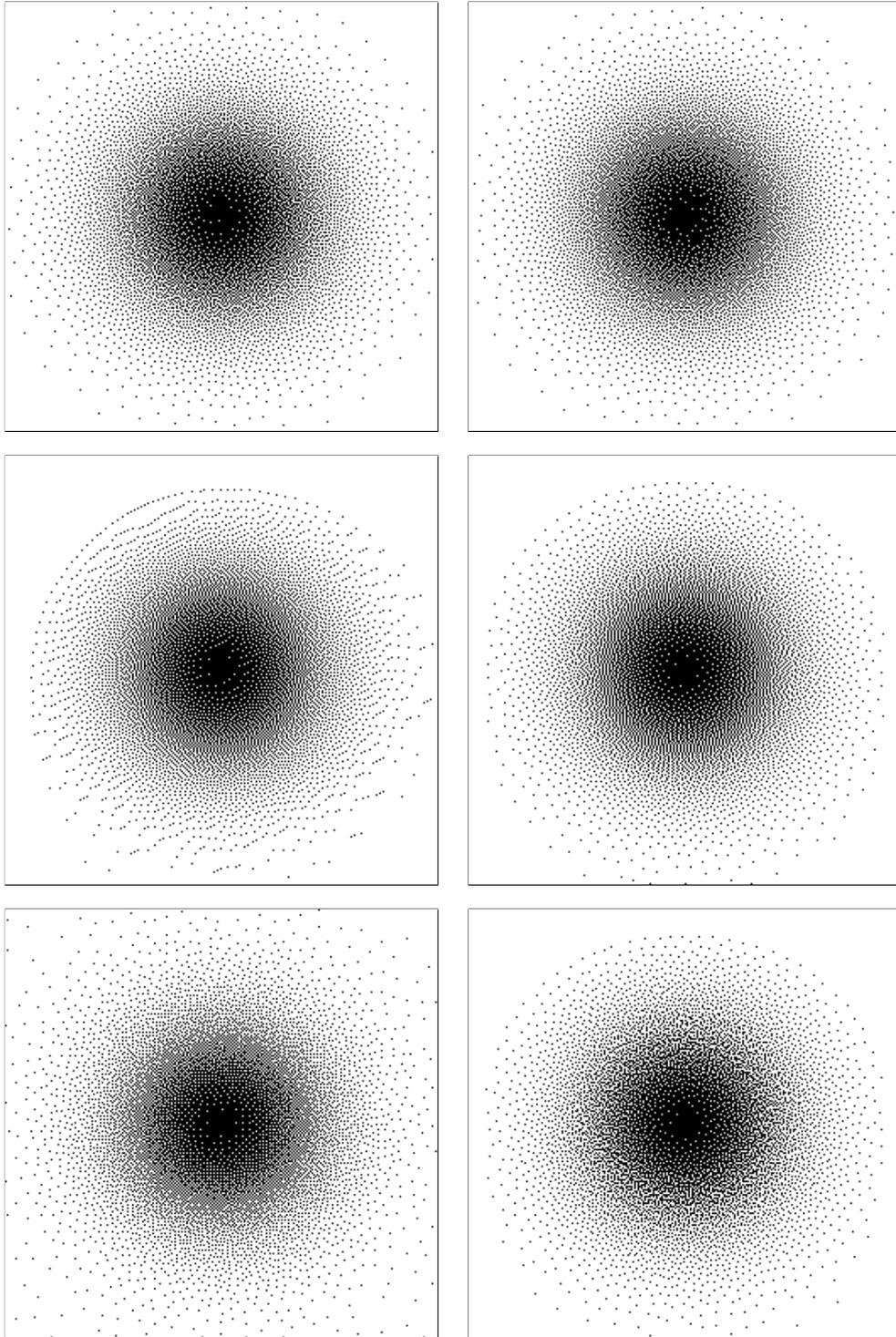


Figure 13: Dithering results by our grid method with  $\varphi(s) = s$  (top left) and  $\varphi(s) = \log(s)$  (top right), the Floyd-Steinberg algorithm (middle left), Ostromoukhov's algorithm (middle right), the algorithm of Purgathofer et al. (bottom left), the algorithm of Zhou et al. (bottom right).

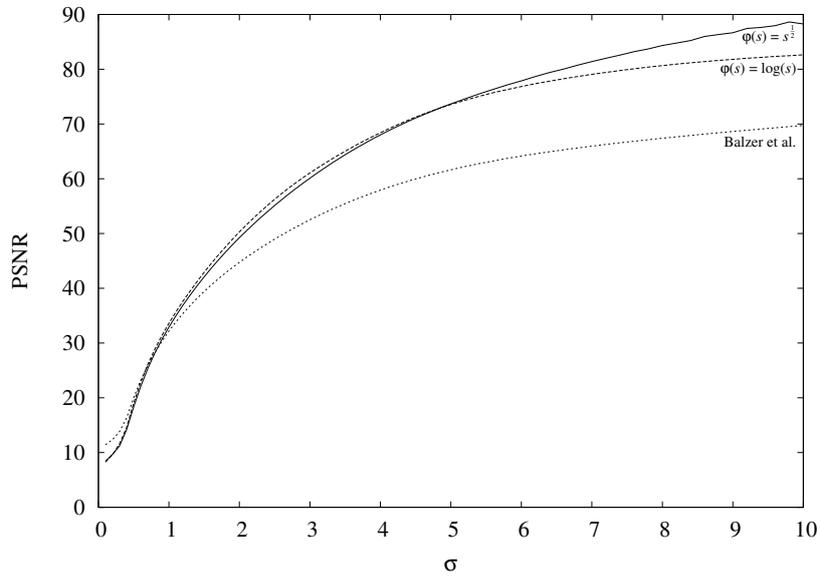


Figure 14: PSNR for stippling results in Fig. 12 under the convolution with Gaussians of various standard deviations  $\sigma$ .

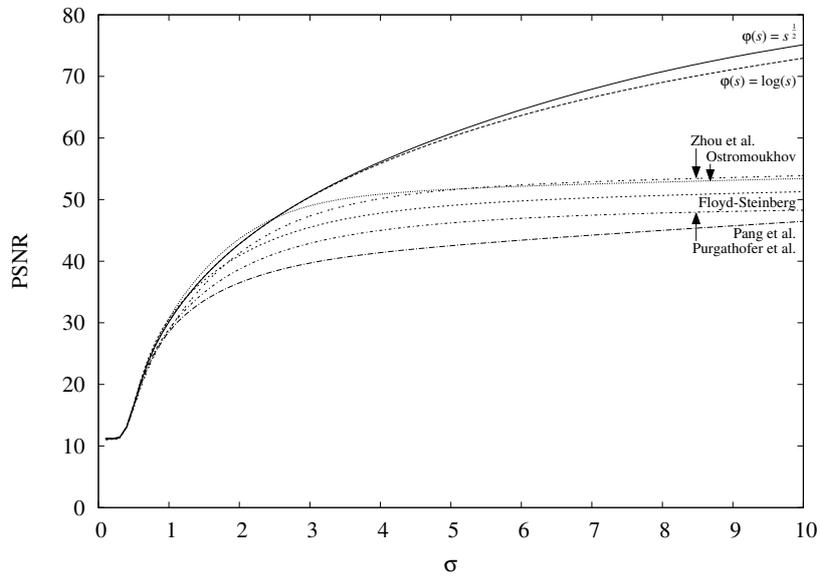


Figure 15: PSNR for dithering results in Figs. 13 and 11 under the convolution with Gaussians of various standard deviations  $\sigma$ .

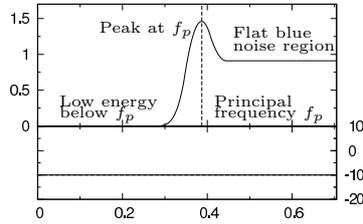


Figure 16: Blue noise behavior of an ideal halftone image proposed in [52, 53].

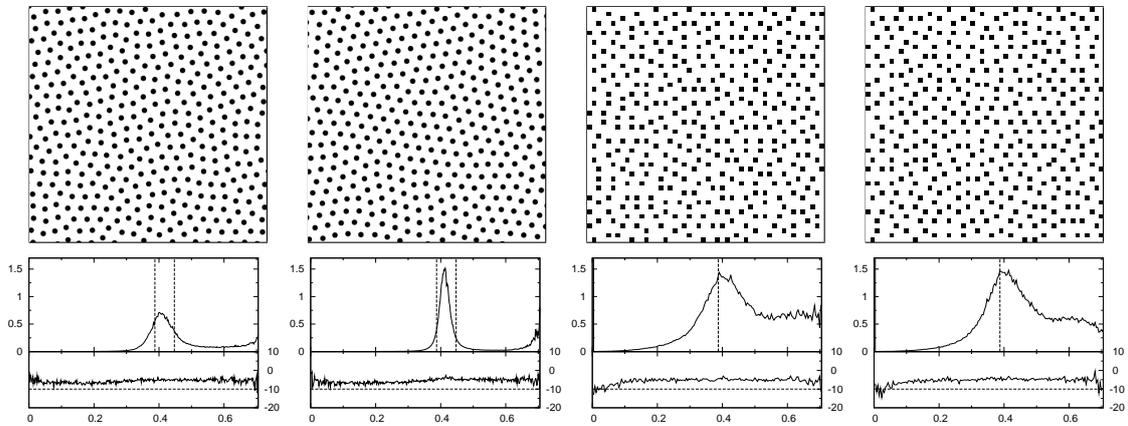


Figure 17: Results for an  $512 \times 512$  image of constant gray value  $u_c = 0.85$  obtained by the stippling method for  $\varphi(s) = s$  and  $\varphi(s) = \log(s)$  and by their discrete dithering versions (left to right). Here  $m = 39322$  points were used. Top: Zoom into the point distributions. Bottom: Radially averaged power spectrum (upper graph) and the anisotropy of the power spectrum in decibels (lower graph).