

# Rotation-Invariance Can Further Improve State-of-the-Art Blind Deconvolution Techniques

Fernando Cervantes<sup>1</sup>, Bryan Usevitch<sup>1</sup>,  
and Vladik Kreinovich<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering

<sup>2</sup>Department of Computer Science

University of Texas at El Paso

500 W. University

El Paso, Texas 79968, USA

fcervantes@miners.utep.edu, usevitch@utep.edu,

vladik@utep.edu

**Abstract**—In many real-life situations, we need to reconstruct a blurred image in situations when no information about the blurring is available. This problem is known as the problem of blind deconvolution. There exist techniques for solving this problem, but these techniques are not rotation-invariant. Thus, the result of using this technique may change with rotation. So, if we rotate the image a little bit, the method, in general, leads to a different deconvolution result. Therefore, even when the original reconstruction is optimal, the reconstruction of a rotated image will be different and, thus, not optimal. To improve the quality of image decomposition, it is desirable to modify the current state-of-the-art techniques by making them rotation-invariant. In this paper, we show how this can be done, and we show that this indeed improves the quality of blind deconvolution.

## I. BLIND IMAGE DECONVOLUTION: FORMULATION OF THE PROBLEM

When we observe an image, we usually observe the image intensity values  $y(i, j)$  at different locations  $(i, j)$  on a rectangular grid, i.e., at a spatial location  $(u_0 + i \cdot \Delta u, v_0 + j \cdot \Delta v)$ , where  $(u_0, v_0)$  is the starting point and  $\Delta u$  and  $\Delta v$  are distances between the neighboring pixels in the  $u$ - and  $v$ -directions.

Each observed value  $y(i, j)$  is, in general, different from the actual (desired) value  $x(i, j)$  of the corresponding intensity. First, there is noise (measurement error), and second, the image is blurred, in the sense that the observed signal  $y(i, j)$  reflects not only the actual intensity  $x(i, j)$  at the same spatial location  $(i, j)$ , but also the intensities  $x(i', j')$  at nearby locations. Under the assumption that the dependence of  $y(i, j)$  on  $x$  is linear, we conclude that

$$y(i, j) = \sum_{i', j'} h(i, j, i', j') \cdot x(i', j') + n(i, j)$$

for some coefficients  $h(i, j, i', j')$ , where  $n(i, j)$  denotes the (additive) noise.

Usually, the blurring is the same for all the pixels, so the coefficients  $h(i, j, i', j')$  depend only on the differences

$$(i - i', j - j')$$

between the spatial locations  $(i, j)$  and  $(i', j')$ :

$$y(i, j) = \sum_{i'} \sum_{j'} h(i - i', j - j') \cdot x(i', j') + n(i, j). \quad (1)$$

Thus, the observed image  $y(i, j)$  is obtained from the actual image  $x(i, j)$  by a convolution. Based on the observed image  $y(i, j)$ , we need to reconstruct the original image  $x(i, j)$ , i.e., to perform a *deconvolution*.

In many real-life situations, we do not have any information about the blurring  $h$ . The corresponding deconvolution problem is known as the problem of *blind deconvolution*.

*Comment.* In vector form, the above relation can be described as  $y = \mathbf{H}x$ , where  $x$  is the vector containing all the intensities of the original image, and  $\mathbf{H}$  is an appropriate matrix. This matrix is called a *convolution matrix* corresponding to the vector  $h$ .

## II. NEED TO USE SPARSITY-BASED TECHNIQUES

In situations when we do not have a partial knowledge of the blurring function, what often helps is the observation that the observed image has a *sparsity property*: namely, if we represent the observed image as a linear combination of functions from an appropriate basis (e.g., an appropriate wavelet basis), then most of the coefficients  $a = (a_1, a_2, \dots)$  in this representation will be 0s (or almost 0s).

In such situations, we have a bound on the number of non-zero coefficients:  $\|a\|_0 \leq c$ , where  $\|a\|_0 \stackrel{\text{def}}{=} \#\{i : a_i \neq 0\}$  and  $c$  is a known constant.

In general, once we have a constraint  $f(a) \leq c$ , then the original problem of minimizing an objective function  $J$  gets transformed into a constrained optimization problem of minimizing the objective function  $J$  under this constraint  $f(a) \leq c$ . To solve this constraint optimization problem, we can use the Lagrange multiplier approach, according to which the above constraint optimization problem is equivalent to minimizing a function

$$J + \lambda \cdot f(a)$$

for some constant  $\lambda$  (known as *Lagrange multiplier*) which needs to be determined from the condition that  $f(a) \leq c$ .

For  $f(a) = \|a\|_0$ , no efficient algorithms are known for minimizing such an objective function: most efficient optimization algorithms require that the objective function be differentiable (or at least continuous), and the expression  $\|a\|_0$  is not even continuous. The good news is that under some reasonable conditions, minimizing this function is equivalent to minimizing the similar expression

$$J + \lambda \cdot \|a\|_1$$

with a continuous  $\ell^1$ -norm  $\|a\|_1 \stackrel{\text{def}}{=} \sum_i |a_i|$  instead of the discontinuous expression  $\|a\|_0$ ; see, e.g., [2]. The  $\ell_1$ -norm is convex, so if  $J$  is also convex (and it often is), then we get an additional advantage of being able to use known algorithms for minimizing convex functions.

### III. STATE-OF-THE-ART TECHNIQUES FOR SPARSITY-BASED BLIND IMAGE DECONVOLUTION

In [1], the following algorithm was proposed to solve the blind deconvolution problem. We know that  $y \approx \mathbf{H}x$ . We also know that  $y$  has the sparsity property, i.e., that  $y \approx \mathbf{W}a$ , where  $\mathbf{W}$  is the matrix describing the corresponding decomposition (e.g., into wavelets), and the vector  $a$  is sparse. We also impose additional restrictions  $R_1(x) \leq \text{const}$  and  $R_2(h) \leq \text{const}$  that imply that  $x$  and  $h$  are sufficiently smooth.

Since  $y \approx \mathbf{W}a$ , the condition  $y \approx \mathbf{H}x$  can be equivalently described as  $\mathbf{W}a \approx \mathbf{H}x$ . The least square approach thus leads us to minimizing the square the the  $\ell_2$ -norm, i.e., that value  $\|y - \mathbf{W}a\|_2^2$ , under the constraints:

- that  $\mathbf{W}a \approx \mathbf{H}x$  (i.e., that  $\|\mathbf{W}a - \mathbf{H}x\|_2^2 \leq \text{const}$ ),
- that  $a$  is sparse (i.e., that  $\|a\|_1 \leq \text{const}$ ),
- that  $R_1(x) \leq \text{const}$ , and
- that  $R_2(h) \leq \text{const}$ .

Applying the Lagrange multiplier technique to this constraint optimization problem, we can reduce it to the unconstrained optimization problem of minimizing

$$Q(a, x, h) \stackrel{\text{def}}{=} \frac{\beta}{2} \cdot \|y - \mathbf{W}a\|_2^2 + \frac{\eta}{2} \cdot \|\mathbf{W}a - \mathbf{H}x\|_2^2 + \tau \cdot \|a\|_1 + \alpha \cdot R_1(x) + \gamma \cdot R_2(h), \quad (2)$$

for appropriate parameters  $\beta$ ,  $\eta$ ,  $\tau$ ,  $\alpha$ , and  $\gamma$ .

Specifically, the authors select

$$R_1(x) = \sum_{d \in D} 2^{1-o(d)} \sum_i |\Delta_i^p(x)|^p, \quad (3)$$

where  $o(d) \in \{1, 2\}$  is the order of the difference operator  $\Delta_i^p(x)$ ,  $0 < p < 1$ , and  $d \in D = \{h, v, hh, vv, hv\}$ . Here,  $\Delta_i^h$  and  $\Delta_i^v$  correspond, respectively to horizontal and vertical first order differences at pixel  $i$ :

$$(\Delta^h x)(n_x, n_y) \stackrel{\text{def}}{=} x(n_x, n_y) - x(n_x - 1, n_y)$$

and

$$(\Delta^v x)(n_x, n_y) \stackrel{\text{def}}{=} x(n_x, n_y) - x(n_x, n_y - 1).$$

and the operators  $\Delta_i^{hh}$ ,  $\Delta_i^{vv}$ ,  $\Delta_i^{hv}$  correspond to second order horizontal, vertical, and horizontal-vertical differences at pixel  $i$ :

$$\Delta_i^{hh}(x) \stackrel{\text{def}}{=} \Delta_i^h(\Delta_i^h(x)), \quad \Delta_i^{vv}(x) \stackrel{\text{def}}{=} \Delta_i^v(\Delta_i^v(x)),$$

$$\text{and } \Delta_i^{hv}(x) \stackrel{\text{def}}{=} \Delta_i^h(\Delta_i^v(x)).$$

The authors also select

$$R_2(h) = \|\mathbf{C}h\|^2, \quad (4)$$

where  $\mathbf{C}$  is the circulant matrix that represents the convolution with the discrete Laplacian operator:

$$(\mathbf{C}h)(n_x, n_y) = h(n_x - 1, n_y) + h(n_x + 1, n_y) + h(n_x, n_y - 1) + h(n_x, n_y + 1) - 4h(n_x, n_y).$$

The goal is to find the values  $x$ ,  $h$ , and  $a$  that minimize the objective function  $Q$ . The algorithm for optimizing this objective function is iterative. It starts with some first approximations to the blur. Then, the algorithm interchangingly uses two steps:

- first, we fix  $a$  and find  $h$  and  $x$  that minimize  $Q$ ;
- then, they fix  $x$  and  $h$  and find  $a$  that minimizes  $Q$ .

The process stops when the images  $x^k$  and  $x^{k-1}$  on the two consequent iterations are sufficiently close to each other, i.e., when

$$\frac{\|x^k - x^{k-1}\|}{\|x^{k-1}\|} < \varepsilon$$

for some pre-determined small threshold  $\varepsilon > 0$ . In [1], the authors select  $\varepsilon = 0.01$ .

To minimize over  $a$ , the authors use an *ll-ls* method described in [5]. Minimization over  $h$  is easy, since the objective function  $Q$  is quadratic in  $h$  and thus, we can differentiate with respect to  $h$ , equate derivatives to 0, and get a system of linear equations for determining  $h$ .

For  $x$ , the situation is not so simple, since in addition to the quadratic term proportional to  $\|\mathbf{W}a - \mathbf{H}x\|^2$ , we also have a non-quadratic term  $R_1(x)$  that includes terms proportional to  $|L(x)|^p$  for some linear operators  $L$ . To perform the corresponding minimization, the authors take into account that this non-quadratic term can be represented as

$$|L(x)|^p = \frac{|L(x)|^2}{|L(x)|^{2-p}}.$$

Thus, to minimize this expression over  $x$ , we can perform the following iterative approach: we start with some initial value  $x$ , and then on each  $(\ell + 1)$ -th iteration, we minimize the quadratic expression

$$\frac{|L(x)|^2}{|L(x^\ell)|^{2-p}},$$

where in the denominator, we use the value  $x^\ell$  from the previous iteration. By explicitly differentiating the corresponding expressions and equating the derivatives to 0, we arrive at the following algorithm.

First, we select  $\alpha, \beta, \gamma, \tau, \theta, \eta^1$ , and  $0 < p < 1$ . We then select some initial estimate  $h^{1,1}$  of the blur, and the initial values of an auxiliary vector  $v_d^{1,1}$ , where  $d \in D = \{h, v, hh, vv, hv\}$ .

The simplest idea is to select each of the components of each initial vector  $v_d^{1,1}$  by using a random number generator that generates numbers uniformly distributed on the interval  $[0, 1]$ .

Then, for  $k = 1, 2, \dots$ , we perform the following until the above stopping criterion is met:

1) For  $\ell = 1, \dots, L_0$  for some  $L_0$ :

a) Compute  $x^{k,\ell+1}$  as

$$\left[ \eta^k (\mathbf{H}^{k,\ell})^T (\mathbf{H}^{k,\ell}) + \alpha p \sum_{d \in D} 2^{1-o(d)} (\Delta^d)^T \mathbf{B}_d^{k,\ell} \Delta^d \right]^{-1} \cdot \eta^k (\mathbf{H}^{k,\ell})^T \mathbf{W} a^k,$$

where  $B_d^{k,\ell}$  is a diagonal matrix with entries

$$B_d^{k,\ell}(i, i) = \left( v_{d,i}^{k,\ell} \right)^{p/2-1}$$

and  $\Delta^d$  is the convolution matrix of the difference operator  $\Delta_i^d(\cdot)$ . For solving this system of linear equations, the authors use the Fourier transform approach.

b) Compute

$$h^{k,\ell+1} = \left[ \eta^k (\mathbf{X}^{k,\ell})^T (\mathbf{X}^{k,\ell}) + \gamma \mathbf{C}^T \mathbf{C} \right]^{-1} \cdot \eta^k (\mathbf{X}^{k,\ell})^T \mathbf{W} a^k,$$

where  $\mathbf{X}^{k,\ell}$  is the convolution matrix of the image  $x^{k,\ell}$ .

For solving this system of linear equations, the authors also use the Fourier transform approach.

c) For each  $d \in D = \{h, v, hh, vv, hv\}$ , calculate

$$v_{d,i}^{k,\ell+1} = \left[ \Delta_i^d(x^{k,\ell}) \right]^2.$$

2) Set  $x^k = x^{k,\ell+1}$ ,  $h^k = h^{k,\ell+1}$ , and  $v_{d,i}^k = v_{d,i}^{k,\ell+1}$ .

3) Now, we need to find  $a^{k+1}$  by minimizing the expression

$$\frac{\beta}{2} \cdot \|y - \mathbf{W}a\|_2^2 + \frac{\eta}{2} \cdot \|\mathbf{W}a - \mathbf{H}x\|_2^2 + \tau \cdot \|a\|_1.$$

This can be done by applying the *II-ls* algorithm for minimizing the equivalent minimization problem

$$\|y' - \Phi' \mathbf{W} a^k\|^2 + \tau \|a^k\|_1,$$

where

$$y' = \begin{bmatrix} \sqrt{\frac{\beta}{2}} y \\ \sqrt{\frac{\eta}{2}} \mathbf{H}^k x^k \end{bmatrix} \text{ and } \Phi' = \begin{bmatrix} \sqrt{\frac{\beta}{2}} \mathbf{I} \\ \sqrt{\frac{\eta}{2}} \mathbf{I} \end{bmatrix}.$$

4) Set  $\eta^{k+1} = \theta \eta^k$ .

*Comment.* Some of the ideas behind this algorithm are heuristic; in [3] and [4], we show that fuzzy techniques can help provide a theoretical justification for these ideas.

## IV. NEED FOR IMPROVEMENT

The current state-of-the-art method for blind image deconvolution is based on minimizing the sum

$$|\Delta_x I_{i,j}|^p + |\Delta_y I_{i,j}|^p \quad (5)$$

for some  $p < 2$ , where  $\Delta_x I_{i,j} \stackrel{\text{def}}{=} I_{i,j} - I_{i-1,j}$ , and  $\Delta_y I_{i,j} \stackrel{\text{def}}{=} I_{i,j} - I_{i,j-1}$ . This is a discrete analog of the term

$$\left| \frac{\partial I}{\partial x} \right|^p + \left| \frac{\partial I}{\partial y} \right|^p. \quad (6)$$

In the traditional least squares approach, when  $p = 2$ , the corresponding expression

$$\left| \frac{\partial I}{\partial x} \right|^2 + \left| \frac{\partial I}{\partial y} \right|^2 \quad (7)$$

is rotation-invariant: namely, it describes the square of the length of the gradient vector

$$\nabla I \stackrel{\text{def}}{=} \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right). \quad (8)$$

However, for  $p \neq 2$ , the corresponding expression is no longer rotation-invariant.

Since the current state-of-the-art technique is not rotation-invariant, the result of using this technique may change with rotation. Thus, if we rotate the image a little bit, the method, in general, leads to a different deconvolution result. So, even when the original reconstruction is optimal, the reconstruction of a rotated image will be different and, thus, not optimal.

## V. HOW TO IMPROVE: MAIN IDEA

As we have just mentioned, the main problem with the current state-of-the-art blind deconvolution techniques comes from the fact that these techniques are not rotation-invariant. To improve the quality of image decomposition, it is therefore desirable to modify the current state-of-the-art techniques by making them rotation-invariant.

In other words, instead of the above non-rotational-invariant expression, we need a rotation-invariant one. Let us first consider the continuous approximation. In this approximation, the desired expression depends on the components  $\frac{\partial I}{\partial x}$  and  $\frac{\partial I}{\partial y}$  of the gradient vector (8). When we rotate the coordinate system, the components of the gradient vector change.

In general, a 2-D vector can be characterized by its length and its direction. When we rotate the coordinate system, the direction changes but the length remains unchanged. Thus, the only rotation-invariant characteristic of a vector  $\vec{a}$  is its length  $\|\vec{a}\|$ . Thus, since we want the desired expression to be rotation-invariant, it must depend only on the length  $\|\nabla I\|$  of the gradient vector, i.e., only on the expression

$$\|\nabla I\| = \sqrt{\left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2}. \quad (9)$$

The actual images are discrete. Thus, as we have mentioned earlier, instead of the derivatives, we have finite differences

$\Delta_x I_{i,j}$  and  $\Delta_y I_{i,j}$ , and instead of  $\|\nabla I\|$ , we have an expression

$$\sqrt{(\Delta_x I_{i,j})^2 + (\Delta_y I_{i,j})^2}. \quad (10)$$

So, the desired rotation-invariant expression  $E$  must have the form

$$E = f\left(\sqrt{(\Delta_x I_{i,j})^2 + (\Delta_y I_{i,j})^2}\right) \quad (11)$$

for an appropriate function  $f(x)$ .

To find the function  $f(x)$ , let us consider a degenerate case, in which the image, in effect, is 1-dimensional, i.e., when the intensity does not change in the  $y$ -direction, it only varies in the  $x$ -direction:  $I(i, j) = I(i)$ . In this degenerate case,  $\Delta_x I_{i,j} = \Delta_x I_i$  and  $\Delta_y I_{i,j} = 0$ . Thus, the above expression (11) takes the form

$$E = f(\sqrt{(\Delta_x I_i)^2}) = f(|\Delta_x I_i|). \quad (12)$$

On the other hand, We have already discussed, in the previous section, that in the 1-D case, the corresponding expression should be proportional to  $|\Delta_x I_i|^p$  for some  $p$ , i.e., it should take the form

$$E = c \cdot |\Delta_x I_i|^p \quad (13)$$

for some  $c$  and  $p$ . By comparing the formulas (12) and (13), we conclude that

$$f(x) = c \cdot |x|^p.$$

Substituting this expression for  $f(x)$  into the formula (11) that describes the general 2-D case, we thus conclude that

$$E = c \cdot \left| \sqrt{(\Delta_x I_{i,j})^2 + (\Delta_y I_{i,j})^2} \right|^p = c \cdot ((\Delta_x I_{i,j})^2 + (\Delta_y I_{i,j})^2)^{p/2}. \quad (14)$$

Thus, we arrive at the following conclusion: to make the blind deconvolution method rotation-invariant, we need to replace the non-rotation-invariant expression (5) with a rotation-invariant expression

$$E = c \cdot ((\Delta_x I_{i,j})^2 + (\Delta_y I_{i,j})^2)^{p/2}, \quad (15)$$

for an appropriate constant  $c$ .

## VI. FROM THE IDEA TO THE ALGORITHMIC DETAILS

How does the above change in the objective function affect the resulting blind deconvolution algorithm? In terms of the algorithm, we replace the sum

$$|\Delta_i^h(x)|^p + |\Delta_i^v(x)|^p \quad (16)$$

with the new expression proportional to

$$((\Delta_i^h(x))^2 + (\Delta_i^v(x))^2)^{p/2}. \quad (17)$$

According to the above description of the state-of-the-art algorithm, to minimize the expression (6.3.1), we represent it as

$$(\Delta_i^h(x))^2 \cdot v_{h,i}^{p/2-1} + (\Delta_i^v(x))^2 \cdot v_{v,i}^{p/2-1}, \quad (18)$$

where:

- $v_{h,i}$  is the value of  $(\Delta_i^h(x))^2$  on the previous iteration, and
- $v_{v,i}$  is the value of  $(\Delta_i^v(x))^2$  on the previous iteration.

We can apply the same idea to minimize the new expression (17). Specifically, to minimize this expression (17), we represent it as

$$((\Delta_i^h(x))^2 + (\Delta_i^v(x))^2) \cdot v^{p/2-1}, \quad (19)$$

where  $v$  is the value of the sum  $(\Delta_i^h(x))^2 + (\Delta_i^v(x))^2$  on the previous iteration.

The expression (19) can be described in the form similar to (18), as

$$(\Delta_i^h(x))^2 \cdot (v'_{h,i})^{p/2-1} + (\Delta_i^v(x))^2 \cdot (v'_{v,i})^{p/2-1}, \quad (20)$$

where  $v'_{h,i} = v'_{v,i}$  is proportional to the value of the sum  $(\Delta_i^h(x))^2 + (\Delta_i^v(x))^2$  on the previous iteration. We have already denoted the values of the squares of differences  $(\Delta_i^h(x))^2$  and  $(\Delta_i^v(x))^2$  on the previous iteration by  $v_{h,i}$  and  $v_{v,i}$ . Thus, we have

$$v'_{h,i} = v'_{v,i} = C \cdot (v_{h,i} + v_{v,i}). \quad (21)$$

In other words, at each spatial location  $i$ , instead of possible different values  $v_{h,i} \neq v_{v,i}$ , we apply equal weights  $v'_{h,i} = v'_{v,i}$  to horizontal and vertical differences.

The current method has been tuned to work well. So, it makes sense to make the difference between the current method and its proposed modification to be as small as possible. When  $v_{h,i} \neq v_{v,i}$ , our new method differs from the current one, but when  $v_{h,i} = v_{v,i}$ , there is no reason for it to differ. It is therefore reasonable to select a constant  $C$  in such a way that when  $v_{h,i} = v_{v,i}$ , the new method will lead to exactly the same result as the current one. In other words, when  $v_{h,i} = v_{v,i}$ , we should have  $v'_{h,i} = v_{h,i}$  and  $v'_{v,i} = v_{v,i}$ . Substituting these values into the formula (21), we conclude that  $C = \frac{1}{2}$ . Thus, the formula (21) takes the following final form:

$$v'_{h,i} = v'_{v,i} = \frac{1}{2} \cdot (v_{h,i} + v_{v,i}). \quad (22)$$

So, we arrive at the following modification of the state-of-the-art blind deconvolution algorithm.

## VII. RESULTING MODIFICATION OF THE STATE-OF-THE-ART BLIND DECONVOLUTION ALGORITHM

The only modification is on Step 1(c), where after computing, for each spatial location  $i$ , the horizontal and vertical values  $v_{h,i}$  and  $v_{v,i}$ , we then average these two values before performing further computations:

First, we select  $\alpha, \beta, \gamma, \tau, \theta, \eta^1$ , and  $0 < p < 1$ . We then select some initial estimate  $h^{1,1}$  of the blur, and the initial values of an auxiliary vector  $v_d^{1,1}$ , where  $d \in D = \{h, v, hh, vv, hv\}$ .

Then, for  $k = 1, 2, \dots$ , we perform the following until the above stopping criterion is met:

- 1) For  $\ell = 1, \dots, L_0$  for some  $L_0$ :
  - 1a) Compute  $x^{k,\ell+1}$  as

$$\left[ \eta^k (\mathbf{H}^{k,\ell})^T (\mathbf{H}^{k,\ell}) + \alpha p \sum_{d \in D} 2^{1-o(d)} (\Delta^d)^T \mathbf{B}_d^{k,\ell} \Delta^d \right]^{-1} \cdot \eta^k (\mathbf{H}^{k,\ell})^T \mathbf{W} a^k,$$

where  $B_d^{k,\ell}$  is a diagonal matrix with entries

$$B_d^{k,\ell}(i, i) = \left( v_{d,i}^{k,\ell} \right)^{p/2-1}$$

and  $\Delta^d$  is the convolution matrix of the difference operator  $\Delta_i^d(\cdot)$ . For solving this system of linear equations, the authors use the Fourier transform approach.

1b) Compute

$$h^{k,\ell+1} = \left[ \eta^k (\mathbf{X}^{k,\ell})^T (\mathbf{X}^{k,\ell}) + \gamma \mathbf{C}^T \mathbf{C} \right]^{-1} \cdot \eta^k (\mathbf{X}^{k,\ell})^T \mathbf{W} a^k,$$

where  $\mathbf{X}^{k,\ell}$  is the convolution matrix of the image  $x^{k,\ell}$ .

For solving this system of linear equations, the authors also use the Fourier transform approach.

1c) For each  $d \in \{h, v, hh, vv, hv\}$ , calculate

$$v_{d,i}^{k,\ell+1} = \left[ \Delta_i^d(x^{k,\ell}) \right]^2;$$

for  $d \in \{h, v\}$ , calculate

$$v_{h,i}^{k,\ell+1} = v_{v,i}^{k,\ell+1} = \frac{1}{2} \cdot \left( \left[ \Delta_i^h(x^{k,\ell}) \right]^2 + \left[ \Delta_i^v(x^{k,\ell}) \right]^2 \right);$$

2) Set  $x^k = x^{k,\ell+1}$ ,  $h^k = h^{k,\ell+1}$ , and  $v_{d,i}^k = v_{d,i}^{k,\ell+1}$ .

3) Now, we need to find  $a^{k+1}$  by minimizing the expression

$$\frac{\beta}{2} \cdot \|y - \mathbf{W}a\|_2^2 + \frac{\eta}{2} \cdot \|\mathbf{W}a - \mathbf{H}x\|_2^2 + \tau \cdot \|a\|_1.$$

This can be done by applying the *l1-ls* algorithm for minimizing the equivalent minimization problem

$$\|y' - \Phi' \mathbf{W} a^k\|^2 + \tau \|a^k\|_1,$$

where

$$y' = \begin{bmatrix} \sqrt{\frac{\beta}{2}} y \\ \sqrt{\frac{\eta}{2}} \mathbf{H}^k x^k \end{bmatrix} \text{ and } \Phi' = \begin{bmatrix} \sqrt{\frac{\beta}{2}} \mathbf{I} \\ \sqrt{\frac{\eta}{2}} \mathbf{I} \end{bmatrix}.$$

4) Set  $\eta^{k+1} = \theta \eta^k$ .

### VIII. TESTING THE NEW ALGORITHM: PRELIMINARY RESULTS

To test the new method, we compared it with the original methods on the same well-known benchmark ‘‘Cameraman’’ image on which the authors of the original paper [1] tested their method. (In the future, we also plan to test our method on the two other benchmark images used in [1].) For the ‘‘Cameraman’’ image, we used the same values of the parameters that the authors of [1] used:

$$\{\alpha, \beta, \gamma, \tau, \eta^1\} = \{1, 1/\sigma^2, 5e5, 0.125, 1042\},$$

where  $\sigma^2$  denotes the noise variance.

Following [1], we also applied, to the original image, the Gaussian blurring with the variance of 5.

We selected  $\sigma = 0.001$ , which is consistent with the signal-to-noise ratio used in [1]. Following [1], we used the mean square difference  $\|x - \hat{x}\|_2$  between the original image  $x$  and the reconstructed image  $\hat{x}$  to gauge the quality of deconvolution.

When comparing the results of the two algorithms, we need to take into account that both the original and the modified algorithms start with randomly selected initial values  $v_d^{1,1}$ . Because of this, the results of both algorithm may differ slightly when we re-apply the same algorithm to the same blurred image. Because of the statistical character of the results, to compare the two algorithms, we need to apply both algorithms to the same blurred image several times, and then use statistical criteria to decide which method is better.

To perform this comparison, we applied each of the two algorithms 30 times, and for each application, we computed the distance  $\|x - \hat{x}\|_2$ . To make the results of the comparison more robust, for each of the algorithms, we eliminated the smallest and the largest value of this distance, and got a list of 28 values. For the original algorithm, we get the following results:

1192.44, 1192.97, 1202.01, 1196.93, 1191.03, 1195.04,  
1195.28, 1204.42, 1194.01, 1192.15, 1195.05, 1191.27,  
1190.42, 1192.78, 1192.20, 1196.84, 1202.12, 1194.88,  
1192.15, 1195.05, 1189.90, 1189.36, 1191.27, 1190.42,  
1192.78, 1192.20, 1196.84, 1202.12.

The average of these values is 1195.21.

For the modified method, we get the following 28 values:

1195.78, 1190.74, 1188.81, 1188.60, 1190.43, 1189.07,  
1191.83, 1187.20, 1189.36, 1191.77, 1189.05, 1189.24,  
1189.36, 1198.58, 1193.91, 1188.08, 1192.04, 1191.54,  
1189.36, 1191.77, 1189.05, 1189.24, 1189.36, 1198.58,  
1193.91, 1188.08, 1192.03, 1191.54.

The average of these values is 1191.01, which is smaller than the average distance corresponding to the original algorithm.

To check whether this difference is statistically significance, we applied the t-test for two independent means. In this test, given two samples of sizes  $N_1$  and  $N_2$ , we compute the corresponding sample means  $\bar{X}_1$  and  $\bar{X}_2$ , sample variances  $s_1^2$  and  $s_2^2$ , and the value

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\left( \frac{(N_1 - 1) \cdot s_1^2 + (N_2 - 1) \cdot s_2^2}{N_1 + N_2 - 2} \right) \cdot \left( \frac{1}{N_1} + \frac{1}{N_2} \right)}}.$$

Based on the value of this statistic, we decide whether the null hypothesis – that both samples comes from the populations with same mean – can be rejected.

For the two above samples, computations lead to rejection with  $p = 0.002$ . This is much smaller than the  $p$ -values 0.01

and 0.05 normally used for rejecting the null hypothesis. So, we can conclude that the null hypothesis can be rejected, and that, therefore, the *modified algorithm is statistically significantly better than the original one*.

#### Comments

- The difference between the two methods is relatively small: on average, about 0.5% on average. This is somewhat expected: we are trying to improve over the state-of-the-art method, a method which has already been optimized a lot and is, therefore, almost as optimal as possible under the given noise level.
- Since the difference between the images obtained by the original method and by the proposed modification is small, a naked eye cannot see this difference; we need to process the two images to realize that there is a difference.

#### ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award “UTEP and Prudential Actuarial Science Academy and Pipeline Initiative” from Prudential Foundation.

The authors are thankful to the anonymous referees for valuable suggestions.

#### REFERENCES

- [1] B. Amizic, L. Spinoulas, R. Molina, and A. K. Katsaggelos, “Compressive blind image deconvolution”, *IEEE Transactions in Image Processing*, 2013, Vol. 22, pp. 3994–4006.
- [2] M. Argaez, C. Ramirez, and R. Sanchez, “An  $l_1$ -algorithm for under-determined systems and applications”, *Proceedings of the 2011 Annual Conference of the North American Fuzzy Information Processing Society*, March 2011, pp. 1–6.
- [3] F. Cervantes, B. Usevitch, L. Valera, and V. Kreinovich, “Why sparse? fuzzy techniques explain empirical efficiency of sparsity-based data- and image-processing algorithms”, *Proceedings of the 2016 World Congress on Soft Computing*, Berkeley, California, May 22–25, 2016, to appear.
- [4] F. Cervantes, B. Usevitch, and V. Kreinovich, “Why  $\ell_p$ -methods in signal and image processing: a fuzzy-based explanation”, *Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society NAFIPS’2016*, El Paso, Texas, October 31 – November 4, 2016, to appear.
- [5] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, “An interior-point method for large-scale  $l^1$ -regularized least squares”, *IEEE Journal on Selected Topics in Signal Processing*, 2007, Vol. 1, No. 4, pp. 606–617.