Feedback Loop Design and Analysis for Iterative Localized Image Deblurring

Dimitry Gorinevsky, *Senior Member, IEEE*

***Abstract*—This paper considers image deblurring algorithm based on model-based iterative update of the image estimate. The update has two principal terms: (i) integral feedback of blurred image prediction error and (ii) feedback of the past deblurred image estimate. The two feedback terms are computed by applying two FIR operators. The paper shows how these feedback update operators can de designed with a given degree of localization to provide an optimal tradeoff between design objectives such as convergence speed, noise amplification, quality of image restoration, and robustness. The solution to the feedback operator design problem is obtained by Linear Programming optimization. A parallelized implementation of the update algorithm using localized feedback operators is scalable to very large image sizes.**

I. Introduction

IMAGE distortions including noise contamination and blurring are encountered in many imaging applications. The blurring might be caused by the imaging system being off-focus, or by the atmospheric distortions. In computed tomography, 3-D images are blurred by a Radon transform.

The existing deblurring algorithms and methods require significant computing power and for high resolution images could take some time even with modern powerful computers. A processing time delay is acceptable for deblurring of static images, such as astronomical images. Yet, processing of streaming images - such as digital video - in real-time, at the frame update rate, might be impossible

A blurred image is a transformation of an ideal image that can be described as convolving the ideal image with a Point Spread Function (PSF) of the blur and adding a noise. The PSF describes a spatial intensity pattern observed in the blurred image for a point source in the ideal image. Herein, it is assumed that the PSF is known and spatially invariant – the same across the image.

Known approaches to computing the deblurred image given the blur PSF fall into two main classes [6,8,12,13,14]. First, there are frequency domain approaches such as Wiener filter or regularized inversion in frequency domain. These approaches are described in more detail below and require computation of a Fourier (Cosine) transform of the deblurred image. Applying the inverse Fourier transform then yields the sought deblurred image. Second, iterative approaches, such as Lucy-Richardson update, optimize loss indices based on the image statistics. The commonly used iterative update methods are localized, in the sense that an update of the image estimate at a given pixel uses information from neighborhood pixels only.

The existing methods work well for static images but are not well suited for real-time deblurring of streaming video. The frequency domain methods are centralized and take excessive computing time even for modern powerful processors. The iterative methods require supervision and might diverge if run continuously. This paper considers an iterative update that is explicitly designed for convergence and performance. The update is localized and could be implemented with a systolic array processor to provide fast computations necessary for streaming image processing.

The proposed update can be considered as an iterative implementation of a 2D non-causal IIR filter, see [2,5,6,7]. In effect, such filter can provide a localized approximation of a deblurring operator designed in the frequency domain. The key innovation of this work is in applying methods of feedback system design and analysis to the estimation filter. The proposed optimal design of the iterative update is based on frequency-domain engineering specifications for filter performance, noise amplification, and, possibly, robustness that are implemented as optimization constraints. The approach is related to recent work on design and analysis of feedback systems [1,3,4,9,10,15,16]. The cited papers consider distributed control problems. This paper considers feedback in distributed estimation – a dual problem. A related multidimensional filter design work is [18].

II. Problem Formulation

To introduce a more technical description of the deblurring problem and the above mentioned deblurring methods, consider the following model of the distorted image

$$y_b(m,n) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} h(k,l)u_d(m-k,n-l) + e(m,n), \quad (1)$$

where m and n are column and row indices of an image pixel element, $y_b(m,n)$ is the observed blurred image, $u_d(m,n)$ is the ideal undistorted image, $e(m,n)$ is the additive noise, and $h(k,l)$ is the blur operator PSF. The image boundaries and boundary effects are herein ignored by assuming that the image extends infinitely in the positive and negative directions. The deblurring goal is to restore the original image u_d given the blurred image y_b.

The convolution (1) can be compactly written in the form $y_b = H * u_d + e$. The blur operator H can be represented through a two-dimensional discrete transfer function

Manuscript received March 7, 2005. Dimitry Gorinevsky is with the Honeywell Labs, San Jose, CA 94134 USA (fax: 208-545-6408; e-mail: dimitry.gorinevsky@honeywell.com).

$$H(\lambda_1,\lambda_2)=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}h(k,l)\lambda_1^k\lambda_2^l, \quad (2)$$

where λ_1 and λ_2 are spatial one-pixel shift operators.

The algorithm explanations below use Fourier transforms of various spatial operators. The Fourier transform of the convolution kernel h in (1) has the form

$$\tilde{h}(v_1,v_2)=H(e^{iv_1},e^{iv_2}), \quad (3)$$

The frequency domain function (3) is known in the field as Optical Transfer Function (OTF). In the frequency domain, when using the Fourier transforms, the convolution (1) takes the form

$$\tilde{y}_b(v_1,v_2)=\tilde{h}(v_1,v_2)\tilde{u}_d(v_1,v_2)+\tilde{e}(v_1,v_2) \quad (4)$$

A. Frequency-domain solutions

By ignoring the noise *e,* one can arrive to the following 'inverse filter' solution for the deblurring problem

$$\tilde{u}(v_1,v_2)=\frac{1}{\tilde{h}(v_1,v_2)}\cdot\tilde{y}_b(v_1,v_2) \quad (5)$$

The inverse filter solution (5) is discussed in many textbooks on image processing and has well recognized deficiency of amplifying high frequency noise, e.g., see [6,8,13]. The blur operator typically has low-pass filtering properties which means $\tilde{h}(v_1,v_2)$ is vanishing small on high spatial frequencies and might cross zero. When inverted, it yields an exceedingly large or infinite gain leading to overwhelming noise amplification.

A reasonable tradeoff between high-frequency noise attenuation and image restoration quality can be achieved by using a Wiener filter approach. It assumes that e and u in (1) are two independent Markov fields of normally distributed variables. In that case an optimal estimate of the image u can be obtained as a Maximum A Posteriori Probability estimate in the form

$$\tilde{u}(v_1,v_2)=\frac{1}{\tilde{h}(v_1,v_2)}\cdot\frac{\left|\tilde{h}(v_1,v_2)\right|^2}{\left|\tilde{h}(v_1,v_2)\right|^2+L(v_1,v_2)}\tilde{y}_b(v_1,v_2), \quad (6)$$

where $L(v_1,v_2)$ is the ratio of the spectral power $S_{ee}(v_1,v_2)$ of the noise e and the spectral power $S_{xx}(v_1,v_2)$ of the signal u

$$L(v_1,v_2)=S_{ee}(v_1,v_2)/S_{xx}(v_1,v_2)$$

Typically, the spectral powers $S_{ee}(v_1,v_2)$, $S_{xx}(v_1,v_2)$ are not known and the ratio $L(v_1,v_2)$ in (6) is manually tuned to satisfy various engineering design considerations.

The Wiener filter deblurring is also known in the field as Constrained Least Squares [11] and is one of the best available methods. Though the Wiener filter deblurring computations can be efficiently implemented, this is still essentially a centralized computational method that requires access to the entire data array.

A closely related frequency domain deblurring approach is given by regularized inversion. This approach follows the general regularization concept of [17]. The image estimate is obtained by minimizing the loss index including quadratic penalties for the restoration error and for the excessive values in the restored image

$$J=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\left|y_b(k,l)-(H*u)(k,l)\right|^2 + \lambda\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\left|(C*u)(k,l)\right|^2\rightarrow\min, \quad (7)$$

where as before * denotes convolution and C is a regularization operator. The operator C is usually chosen empirically and might be a unity operator, or a high pass operator, such as a discrete Laplacian. When the regularization parameter λ is small, the regularized solution closely resembles the inverse filter solution (5). Note that the regularized solution has the same form (6) as the Wiener filter solution with

$$L(v_1,v_2)=\lambda\left|C(v_1,v_2)\right|^2$$

The regularized inversion deblurring has the same advantages and disadvantages as Wiener Filter deblurring.

B. Iterative Update

Closer to the spirit of this work are iterative methods. Historically, many of these methods were developed in the early years of image processing and are very economical in terms of the required computing power and memory access. All of the three methods mentioned below include localized updates. An update for each pixel is based on the data for the pixels in the immediate neighborhood. Such updates can be efficiently implemented through parallel processing using systolic array similar to the proposed update. Yet, as will be described below, the known iterative methods have a fundamental deficiency which is addressed by the proposed update.

The three iterative updates described below are most often used and are described in [2, 6, 8, 12, 13, 14]. The first iterative method is the Successive Approximation (Landweber) Method also known as an "iteration with reblurring". It corresponds to steepest descent optimization of quadratic error - the first term in (7) and has the form

$$u(n+1)=u(n)-\gamma H^*\left(H*u(n)-y_b\right), \quad (8)$$

where $u(n)$ is the image array at iteration n, γ is a scalar factor used to adjust the convergence speed, and H^* is an adjunct operator for H. If H corresponds to a convolution kernel $h(k,l)$ as in (1), then H^* corresponds to a convolution kernel $h^*(-k,-l)$ where * denotes a complex conjugate.

A variation of the update is Van Cittered method which can be described as stochastic approximation (least-mean-square update) minimizing the same quadratic performance index.

$$u(n+1)=u(n)-\gamma\left(H*u(n)-y_b\right) \quad (9)$$

The updates (8) and (9) are linear. A nonlinear update that is supposed to converge faster is Lucy-Richardson update. While the update (8) can be considered as Maximum Likelihood optimization with Gaussian noise model, the Lucy-Richardson update is a Maximum Likelihood optimization for a Poisson noise model. Assuming that H is

normalized, such that $\sum h(k,l) = 1$, the Lucy-Richardson update can be written in a compact form as

$$u(n+1) = \left(\frac{y_b}{H * u(n)} * H^* \right) \cdot u(n), \qquad (10)$$

where the division (fraction) inside the big brackets and the multiplication outside of the brackets are pixel-wise operations. The convolutions denoted by * are computed in the usual way.

The iterative updates (8)—(10) are the most representative ones, but numerous minor variations are known. These updates have two key features highly relevant to this work. First, the computations in (8)—(10) are localized to the extent that blur PSF operator H is localized. Second, the updates (8)—(10) suffer from common problems including slow convergence noise amplification and 'ringing' (producing edge artifacts and spurious "sources"). For practical implementation of these updates, stopping the update in time is of foremost importance. The quality of the recovered image is first improved in the update and then starts deteriorating. Stopping an update in time to achieve optimal quality of the recovered image requires supervision and is unacceptable for an implementation in a systolic array processor.

C. Needs and gaps to address

This work addresses the iterative update convergence issue and, thus, enables an embedded systolic array implementation. The main reason for the eventual divergence of the updates (8)—(10) is that each of these updates converges to the inverse solution (4) as its steady state. As discussed in the previous subsection, the inverse solution is unacceptable because of high frequency noise amplification. To achieve good quality of deblurring, there is a need for high-frequency regularization, an optimal trade off between idealized restoration accuracy and noise amplification. This is exactly what the proposed update does, while keeping within the localized computations framework. A frequency domain solution of the form (6) can be used to achieve the same goal, but it is not localized and, thus, is unsuitable for a systolic array implementation.

Essentially the proposed approach introduces regularization into a localized iterative update. The localized error feedback and the regularization terms in the update are used as design knobs in a formal specification-based optimal design procedure to achieve a image restoration performance, noise rejection, convergence speed, robustness objectives. The procedure is similar to optimal design of distributed feedback control.

III. Proposed Approach

The iterative update studied in this work has the form

$$u(n+1) = u(n) - K * \left(H * u(n) - y_b \right) - S * u(n), \qquad (11)$$

where * denotes convolution, K is a FIR feedback update operator with a convolution kernel $k(i,j)$, and S is a FIR 'smoothing' operator with a convolution kernel $s(i,j)$.

The fundamental difference between the update (11) and the updates (8)—(9) is in the last term on the r.h.s. (11). Without this term, the update describes an integrator that will accumulate the estimate u as long as there is a remaining error. The term $S*u(n)$ introduces integrator leakage limiting potentially unchecked growth of u in the update. A more detailed analysis and design of the operators K and S is presented below.

The update (11) can be considered as iterative implementation of a 2D IIR deblurring filter, see [5,6,7]. This IIR filter essentially approximates a frequency-domain deblurring operator, such as Wiener filter. We will discuss the design of the update (11) in frequency domain. In the update, the operator S corresponds to regularization term in the frequency-domain design. Unlike a Wiener filter, the update (11) provides a localized implementation, since K and S are FIR operators.

A. Systolic array implementation

The update (11) is amenable to a systolic array processor implementation. Such processor consists of an array of simple processing elements (logic blocks), one per image pixel. The processing logic blocks are interconnected such that each can exchange data with its immediate neighbors. Each of the processing logic blocks also has some local data memory and is capable of simple arithmetic operations such as addition, subtraction, and multiplication. Using the interconnect logic, the blurred image can be downloaded into the processor array, each pixel into a respective processing logic block. By sequentially exchanging data with the nearest neighbors, each logic block can accumulate data from any predefined number of neighbors within certain reach. Of course this would require multiple update cycles, a larger number of cycles for a larger reach.

By accumulating the data within the reach of the localized spatial operators, the update (11) for each pixel can be implemented within the respective processor. This presumes preloading and storing the operators H, K, and S in each of the array processing logic blocks. For each pixel, the computations (11) are extremely simple and require a number of additions, subtractions, and multiplications, proportional to the squared size of the FIR operators H, K, and S. Apart from the data transfer associated with downloading the blurred image and uploading the deblurred image estimate, the computational demand on each of the processing logic blocks does not at all depend on the image size. Of course, the overall amount of computations grows linearly with the image size. These computations are, however, performed in parallel by the processing blocks of the array. The data transfer requirements grow very slowly, as a square root of the image pixel size. Thus, the proposed approach is fully scalable and can be implemented at high speed for high-resolution video images.

B. Analysis of feedback loop

The technical approach to analysis and design of the spatial update operators in (11) largely follows the distributed control technique presented in papers [9,15]. The approach will be briefly reviewed here for completeness and as it applies to the image restoration problem in question.

The analysis to follow assumes that PSF of the blur is centrally symmetric, such that

$$H(\lambda_1,\lambda_2) = H(-\lambda_1,-\lambda_2) \quad (12)$$

The central symmetry described by (12) is also known as (2-fold) symmetry. Other symmetry types are possible and can be conveniently handled. These symmetry types are commonly considered in image processing and include 4-fold symmetry

$$H(\lambda_1,\lambda_2) = H(-\lambda_1,-\lambda_2) = H(-\lambda_1,\lambda_2) = H(\lambda_1,-\lambda_2) \quad (13)$$

or an 8-fold symmetry

$$\begin{aligned} &H(\lambda_1,\lambda_2) = H(-\lambda_1,-\lambda_2) = H(-\lambda_1,\lambda_2) = H(\lambda_1,-\lambda_2) \\ &= H(\lambda_2,\lambda_1) = H(-\lambda_2,-\lambda_1) = H(-\lambda_2,\lambda_1) = H(\lambda_2,-\lambda_1) \end{aligned} \quad (14)$$

It is important to note that in each of the cases (12)—(14) the optical transfer function (3) is real. The update operators K and S in (11) are designed to have the same type of symmetry as H and also yield real optical transfer functions $\widetilde{k}(v_1,v_2)$ and $\widetilde{s}(v_1,v_2)$ respectively.

If none of the above symmetry properties holds, the approach described herein can be used after 'squaring down' the system – by applying the conjugated operator H^* to the prediction error in the intermediate calculations. This would make the system in the update symmetric and allow using the same approach to analysis and design.

Computing Fourier transforms of (11) yields

$$\widetilde{u}(n+1) = [1 - \widetilde{k}\cdot\widetilde{h} - \widetilde{s}]\cdot\widetilde{u}(n) + \widetilde{k}\widetilde{y}_b \quad (15)$$

This frequency domain description can be considered as a collection of modal dynamics equations, each mode corresponding to different combination of the spatial frequencies. The dynamics of the modes are independent and the controller design requirements can be considered for each mode separately. This would yield a family of the requirements parameterized by the frequencies v_1 and v_2. The requirements at different frequencies are influenced by a common set of the controller parameters.

Let us consider in more detail how some of the common specification requirements can be expressed in the spatial frequency domain. Consider the following specifications: convergence speed, image restoration performance, and noise sensitivity. These specifications can be extended to include robustness to modeling error, see [9,15].

The update convergence rate is defined by the multiplier at $\widetilde{u}(n)$ in the r.h.s. (15). The update converges to a steady state point as $(r_0)^n$ provided that for all spatial frequencies $0 \le v_1, v_2 \le 2\pi$ the following condition holds

$$\left|1 - \widetilde{k}(v_1,v_2)\widetilde{h}(v_1,v_2) - \widetilde{s}(v_1,v_2)\right| \le r_0 < 1 \quad (16)$$

The remaining specifications are defined for the image $\widetilde{u}(v_1,v_2)$ obtained as a steady state of the system (15). This steady state can be determined by substituting (4) into (15)

$$\widetilde{u}(v_1,v_2) = \frac{\widetilde{k}\widetilde{h}}{\widetilde{k}\widetilde{h} + \widetilde{s}}\widetilde{u}_d(v_1,v_2) + \frac{\widetilde{k}}{\widetilde{k}\widetilde{h} + \widetilde{s}}\widetilde{e}(v_1,v_2) \quad (17)$$

The small steady state prediction error requirement is that in the absence of noise (drop the second term in the r.h.s (17)) the image restoration error should be small, i.e., $\left|\widetilde{u} - \widetilde{u}_d\right| \le u_0$. This should hold for all frequencies inside the spatial bandwidth and can be written in the form

$$\left|\frac{\widetilde{s}(v_1,v_2)}{\widetilde{k}(v_1,v_2)\widetilde{h}(v_1,v_2) + \widetilde{s}(v_1,v_2)}\right| \le u_0, \{v_1,v_2\} \in B, \quad (18)$$

where B is the spatial 'in band' domain and u_0 specifies acceptable relative intensity of the error. This domain can be chosen in different ways. As one example, the domain B is a set of spatial frequencies v_1, v_2, where $\widetilde{h}(v_1,v_2) \ge h_0 > 0$.

The next requirement is that noise amplification in (17) is small. This should hold for all frequencies $0 \le v_1, v_2 \le 2\pi$ and can be presented in the form

$$\left|\frac{\widetilde{k}(v_1,v_2)}{\widetilde{k}(v_1,v_2)\widetilde{h}(v_1,v_2) + \widetilde{s}(v_1,v_2)}\right| \le d_0, \quad (19)$$

The introduced design specifications are reminiscent of the loopshaping design in feedback control, but they also differ in two major respects. First, we are dealing with design of a non-causal spatial loop and, thus, the form of the specification requirements is somewhat different from those encountered in the time domain loop design. In particular, the presence of the smoothing operator S is specific to the spatial design. Second, as will be explained in more detail below, the requirements (16), (18), (19) can be cast in the form of a linear program, which can be solved in a straightforward way. At the same time, in loopshaping of causal feedback there is no straightforward formal solution satisfying all the specifications and it is usually an iterative process involving a designer.

C. Design of feedback operators

The design of localized FIR operators K and S herein closely follows the approach of [9], where feedback design for a distributed control problem is considered. The update (11) is a recursive filter estimating the 2-D input signal u from noisy and blurred system output data. It is well known that estimation problems in linear systems theory are dual to control problems and the same mathematical tools can be applied to both. This is why the distributed control approach of [8] is applicable to the problem in question.

The FIR feedback operators K and S in (11) are assumed to have the same symmetry properties as the blur operator H. In the basic case of a central (2-fold) symmetry, these operators can be presented in the form

$$k(\lambda_1,\lambda_2) = k_0 + \sum k_{mn}\left(\lambda_1^m\lambda_2^n + \lambda_1^{-m}\lambda_2^{-n}\right), \quad (20)$$

$$s(\lambda_1,\lambda_2) = s_0 + \sum s_{mn}\left(\lambda_1^m\lambda_2^n + \lambda_1^{-m}\lambda_2^{-n}\right), \quad (21)$$

where k_0, k_{mn}, s_0, s_{mn} are the design parameters and the summation in (20)—(21) is performed over all indices covering a given vicinity of the zero spatial shift. A rectangular support set of the FIR operators might consists of all indices such that $0<m<N$, $|n|<N$ or $m=0$, $0<n<N$. Corresponding relationships can be written down in the case of a 4-fold or 8-fold symmetry. As an example, in case of 8-fold symmetry the operator K has the form

$$k(\lambda_1,\lambda_2) = k_0 + \sum k_{mn}\left(\lambda_1^m\lambda_2^n + \lambda_1^{-m}\lambda_2^{-n} + \lambda_1^{-m}\lambda_2^n + \lambda_1^m\lambda_2^{-n} + \lambda_1^n\lambda_2^m + \lambda_1^{-n}\lambda_2^{-m} + \lambda_1^{-n}\lambda_2^m + \lambda_1^n\lambda_2^{-m}\right)$$

Collecting all the independent weights of the two FIR operators yields the design parameter vector

$$x = [k_0 \quad \dots \quad k_{mn} \quad \dots \quad s_0 \quad \dots \quad s_{mn} \quad \dots] \qquad (22)$$

This vector should be chosen such that the system transfer functions satisfy requirements (16), (18), (19).

By substituting $\lambda_1 = e^{iv_1}, \lambda_2 = e^{iv_2}$ into (20), (21) and using (12), one can notice that all three optical transfer functions $\tilde{k}(v_1,v_2)$, $\tilde{s}(v_1,v_2)$ and $\tilde{h}(v_1,v_2)$ are real. These functions are also linear in the design parameter vector (22). Another key fact is that the denominator in (20), (21) is always real positive. This follows from (16). The denominator affine in x can be presented in the form

$$\begin{aligned} &\tilde{s}(v_1,v_2) + \tilde{k}(v_1,v_2)\tilde{h}(v_1,v_2) = c^T(v_1,v_2)x \\ &c^T(v_1,v_2)x > 0, \qquad v_1,v_2 \in \Re \end{aligned} \qquad (23)$$

The last inequality follows from the requirement (16) which can be presented in the form

$$1 - r_0 \le c^T(v_1,v_2)x \le 1 + r_0, \qquad 0 < r_0 < 1, \quad (24)$$

As noted in [9], the denominator is positive real because the all operators are symmetric non-causal. This is fundamentally different from (and simpler than) time-domain design of feedback control, where the denominator is not real.

The design constraints (18), (19) - and many other design specifications that one might wish to consider for this problem - can be presented in the form

$$\left|\frac{a^T(v_1,v_2)x + b(v_1,v_2)}{c^T(v_1,v_2)x}\right| \le 1$$

With the real positive denominator, this can be written as

$$-c^T(v_1,v_2)x \le a^T(v_1,v_2)x + b(v_1,v_2) \le c^T(v_1,v_2) \quad (25)$$

For feedback design, these inequalities can be complemented by a requirement that the integrator leakage operator S is possibly small. Small operator S improves the image recovery performance. The latter requirement can be presented in the form

$$|Rx| \le x_0, \qquad x_0 \to \min, \qquad (26)$$

where the matrix R selects the last half of the components of x, ones that contain the weights of the smoothing operator S. The absolute value in (26) is component-wise and x_0 is a scalar. The linear inequalities of the form (25) following from the design constraints (18), (19), as well as (24) can be solved by gridding the spatial frequencies $\{v_1, v_2\}$. By adding (26) and augmenting the design vector x with the parameter x_0 we have a Linear Programming (LP) problem. The LP problem of a large size (for a dense grid in v_1, v_2) can be solved efficiently and reliably with modern interior point solvers. In the example below, LINPROG solver from Matlab Optimization Toolbox was used.

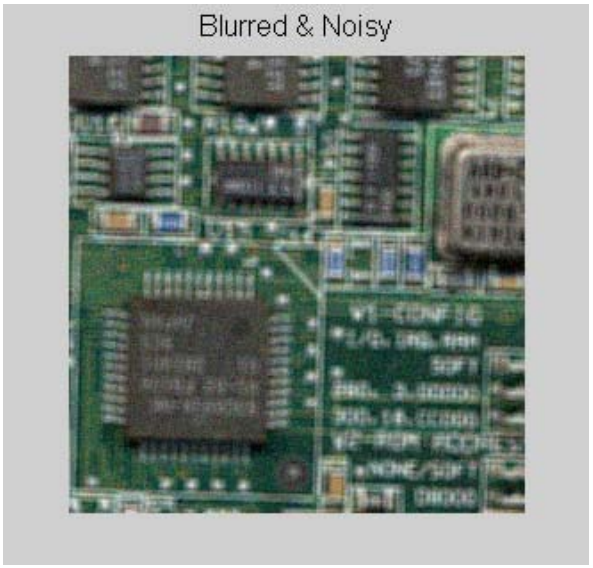

Fig. 1. Blurred image with noise added

IV. Implementation Example

This section describes an example of implementing the update (11) with the operators designed using the LP design approach of the previous section. The original undistorted image u_d used in this example was taken from Matlab Imaging Toolbox demo. This image was distorted with a localized Gaussian blur operator H with a FIR PSF operator with maximal N=3 spatial offset taps on each side for each of thee two spatial dimensions. A random noise e with a maximum magnitude 0.02 was added to the image. The blurred and noisy image y_b is shown in Figure 1.

The feedback operators S and K in (11) have been designed as described in the previous section. The design assumed FIR operators with N=3 maximal spatial offset and 8-fold symmetry (the operator H has 8-fold symmetry).

The following parameters have been used in the design:

- the convergence rate in (16) was chosen as $r_0 = 0.08$
- the recovery error bound in (18) was $u_0 = 0.3$
- the image noise bound in (19) was chosen as $d_0 = 0.08$
- the in-band domain B in (18) was a set of spatial frequencies such that $\tilde{h}(v_1,v_2) \ge h_0$, with $h_0 = 0.185$

The design resulted in the feedback operators K and S illustrated in Figure 2. They have 8-fold symmetry and only a first quadrant (nonnegative offsets) is shown.

The recovered image was obtained by applying 30 steps of the update (11) with designed operators K and S and the known blur operator H to the image y_b. The recovered

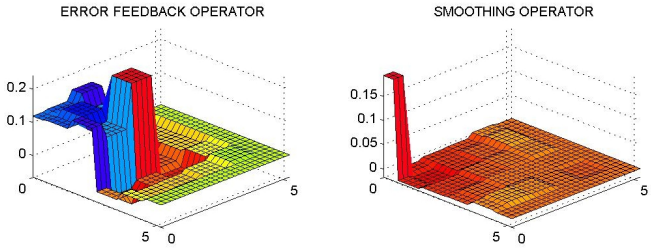

Fig. 2. Designed symmetric update operators K and S (first quadrant only)

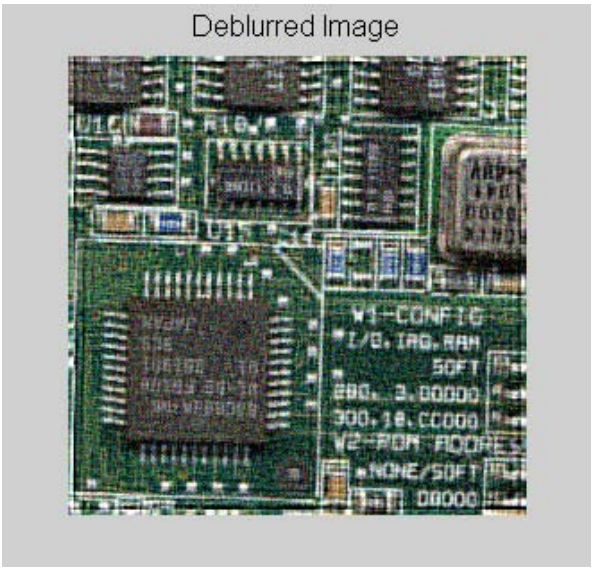

Fig. 3. Recovered (de-blurred) image

image is displayed in Figure 3. As one can see the recovery quality is quite good. The original and blurred images were taken from Matlab Image Processing Toolbox demo illustrating the use of a Lucy-Richardson algorithm. The image in Figure 6 has somewhat better perceived recovery quality than one obtained by the Lucy-Richardson algorithm in the Matlab demo. The fundamental difference however is that if run further unchecked, the update (11) continues (slightly) improving the recovery quality, while continuing the Lucy-Richardson update leads to the restored image deterioration because of the high-frequency noise increase and ringing that become increasingly noticeable. The recovery error at iteration 10 is slightly less than for the proposed algorithm, at iteration 15 it is larger.

Finally, Figure 4 illustrates optical transfer functions in the designed specifications for the update, those in (16), (18), (19). The upper plot shows regularization error transfer function $\tilde{s}(v_1,v_2)/\left[\tilde{s}(v_1,v_2)+\tilde{k}(v_1,v_2)\tilde{h}(v_1,v_2)\right]$. The middle plot shows noise amplification transfer function $\tilde{k}(v_1,v_2)/\left[\tilde{s}(v_1,v_2)+\tilde{k}(v_1,v_2)\tilde{h}(v_1,v_2)\right]$. The lower plot shows the loop gain $\tilde{s}(v_1,v_2)+\tilde{k}(v_1,v_2)\tilde{h}(v_1,v_2)$.

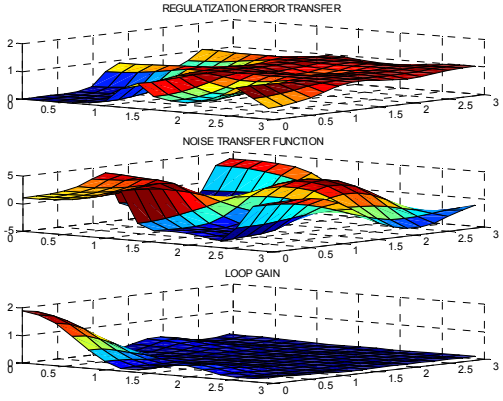

Fig. 4. Loop transfer functions for the designed update

V. Conclusions

The iterative deblurring update analyzed in this work is has performance comparable with other existing updates such as Lucy-Richardson update. It is similarly localized and can be implemented in parallel hardware. The fundamental advantage of the proposed update and the design method is in the proven convergence stability. Most other deblurring update methods start accumulating error if the iterations continue too long. In this work this is recognized as a stability problem in a distributed feedback loop. The stability, convergence, noise amplification and other requirements to the proposed update are explicitly included in the formal specification-based design procedure explained in this paper.

References

[1] Bamieh, B., Paganini, F., and Dahleh, M. "Distributed control of spatially-invariant systems," *IEEE Trans. on Automatic. Contr.*, Vol. 47, No. 7, July 2002, pp. 1091—1107.

[2] Biemond, J., Lagendijk, R.L., and Mersereau, R.M. 'Iterative methods for image deblurring,' *Proceedings of the IEEE*, Vol. 78, No. 5, pp. 856—883, 1990.

[3] D'Andrea, R. and Dullerud, G.E., "Distributed control for spatially interconnected systems," IEEE Trans. on Automatic Control, Vol. 48, No. 9, 2003, pp. 1478—1495.

[4] De Castro, G. A. and Paganini, F. "Convex synthesis of localized controllers for spatially invariant systems," Automatica, Vol. 38, No. 3, 2002, pp. 445—450.

[5] Dudgeon, D.E. An iterative implementation of 2-D digital filters," *IEEE Trans. on Acoustics Speech and Signal Processing*, Vol. ASSP-28, No. 6, 1980, pp. 666—671.

[6] Dudgeon, D.E. and Mersereau, R.M. *Multidimensional Digital Signal Processing*, Prentice-Hall, 1984.

[7] Dudgeon, D.E. and Quatieri, T.F. "Implementation of 2-D digital filters by iterative method," *IEEE Trans. on Acoustics Speech and Signal Processing*, Vol. ASSP-30, No. 3, 1982, pp. 473—487.

[8] Gonzalez, R.C. and Woods, R.E. *Digital image processing*, *2nd ed.,* Upper Saddle River, N.J.: Prentice Hall, 2002.

[9] Gorinevsky, D., Boyd, S., Stein, G. "Optimization-based tuning of low-bandwidth control in spatially distributed systems," *American Control Conf.*, Vol.3, pp.2658—2663, Denver, CO, June 2003.

[10] Gorinevsky, D. and Stein, G. "Structured uncertainty analysis of robust stability for multidimensional array systems," *IEEE Trans. on Automatic Control*, Vol. 48, No. 8, 2003, pp. 1557—1568.

[11] Hunt, B.R. "The application of constrained least squares estimation to image restoration by digital computer," *IEEE Trans. Computers*, Vol. C-22, No. 9, pp. 805—812, 1973.

[12] Katsaggelos, A. "Iterative image restoration algorithms," *Optical Engineering*, Vol 28, No. 7, pp. 735—748, 1989.

[13] Lim, J.S.. *Two-dimensional signal and image processing*, Englewood Cliffs, N.J.: Prentice Hall, 1990.

[14] Starck, J.L., Pantin, E., and Murtagh, F. "Deconvolution in Astronomy: A Review," *Publications of the Astronomical Society of the Pacific*, Vol. 114, pp. 1051–1069, 2002.

[15] Stein, G. and Gorinevsky, D. "Design of surface shape control for large two-dimensional array," *IEEE Trans. on Control Systems Technology*, Vol. 13, No. 3, 2005, pp. 422—433.

[16] Stewart, G., Baker, P., Gorinevsky, D., and Dumont, G. "An experimental demonstration of recent results for spatially distributed control systems," *American Control Conference*, pp. 2216—2221, vol. 3, 2001, Arlington, VA, June 2001.

[17] Tikhonov, A.N. and Arsenin, V.A. *Solutios of ill-posed problems*. W.H. Winston, Washington D.C., 1977.

[18] Gorinevsky, D. and Boyd, S. "Optimization-based design and implementation of multi-dimensional zero-phase IIR filters," *IEEE Trans. on Circuits and Systems – I*, 2005 (to appear).