Combining Frequency and Spatial Domain Information for Fast Interactive Image Noise Removal

Anil N. Hirani, Takashi Totsuka

Sony Corporation

Abstract

Scratches on old films must be removed since these are more notice-able on higher definition and digital televisions. Wires that suspend actors or cars must be carefully erased during post production of special effects shots. Both of these are time consuming tasks but can be addressed by the following image restoration process: given the locations of noisy pixels to be replaced and a prototype image, restore those noisy pixels in a natural way. We call it image noise removal and this paper describes its fast iterative algorithm. Most existing algorithms for removing image noise use either frequency domain information (e.g low pass filtering) or spatial domain infor-mation (e.g median filtering or stochastic texture generation). The few that do combine the two domains place the limitation that the image be band limited and the band limits be known.

Our algorithm works in both spatial and frequency domains without placing the limitations about band limits, making it pos-sible to fully exploit advantages from each domain. While global features and large textures are captured in frequency domain, local continuity and sharpness are maintained in spatial domain. With a judicious choice of operations and domains in which they work, our dual-domain approach can reconstruct many contiguous noisy pixels in areas with large patterns while maintaining continuity of features such as lines. In addition, the image intensity does not have to be uniform. These are significant advantages over exist-ing algorithms. Our algorithm is based on a general framework of projection onto convex sets (POCS). Any image analysis technique that can be described as a closed convex set can be cleanly plugged into the iteration loop of our algorithm. This is another important advantage of our algorithm.

CR Categories: I.3.3 [Computer Graphics]: Picture / Image Gen-eration; Display Algorithms; I.3.6 [Computer Graphics]: Method-ology and Techniques – Interaction techniques; I.4.4 [Image Pro-cessing]: Restoration; I.4.9 [Image Processing]: Applications.

Additional Keywords: scratch and wire removal, projections onto convex sets, POCS.

{`hirani | totsuka`}@av.crl.sony.co.jp
Research Center, Sony Corporation
6-7-35 Kitashinagawa Shinagawa-ku, Tokyo 141, Japan

1 INTRODUCTION

The proliferation of television channels and increasing use of mul-timedia viewing platforms means that older films are likely to see increased use. In addition, higher definition and digital television formats mean that imperfections in old film stock are going to be-come more noticeable. Removal of scratches from old films and photographs is one motivation for this paper.

Another motivation comes from needs of film and video post production. In some special effects scenes in films, actors or objects are suspended from wires. These wires are later removed in post production either by using an optical process or by processing the digitized film. The digital process is much more commonly used now. As a result, increasing the efficiency of tools for digital wire removal has become important. All these factors indicate the need for efficient and accurate tools for removing scratch, wire and other unwanted noise from images.

All these problems can be addressed by the following image restoration process.

> Given *(i)* the locations of noisy pixels and *(ii)* a proto-type (sample) image, restore those noisy pixels in a *nat-ural* way.

By *natural*, we mean that the continuity of intensity and features (e.g., textures, lines) with the surrounding area is maintained. For the scratch removal and the wire removal applications described above, pixels to be restored are those under a scratch or a wire, and the sample image is usually taken from a nearby region. In this paper, we refer to this image restoration process as "image noise removal".

Although image restoration is not a new concept, existing noise removal algorithms have difficulty with noise which *(i)* consists of many contiguous pixels and *(ii)* is in a textured area of image or areas with prominent lines. Note that by *texture* we mean not only small stochastic texture but also small patterns like fabric texture as shown in Fig. 10(b). In addition there can be prominent system-atic lines or lines placed randomly in the image. The brick wall in Fig. 10(a) and the stone wall in Fig. 10(c) are examples. Our algo-rithm for removal of noise is based on the theory of projections onto convex sets. Ours is a fast iterative algorithm that uses the available information from both frequency and spatial domain.

The pixels determined by the algorithm to replace the noise are *(a)* as sharp as the surrounding area *(b)* maintain continuity of prominent lines running across the noise pixels and *(c)* have a tex-ture matching the surrounding texture. While some previous algo-rithms were able to remove such noise from images with stochastic texture or small regularly patterned textures, ours works on those as well as the more difficult cases of systematic or randomly placed prominent lines. To our knowledge it is the first application of POCS for interactive image noise removal. Ours is also the first image noise removal algorithm that combines frequency and spa-tial domain information in an extendible way. It does this by using the clean and well understood formalism of POCS and without re-quiring that the images be band limited. In addition it works even when the noisy pixels are contiguous and numerous. Another key
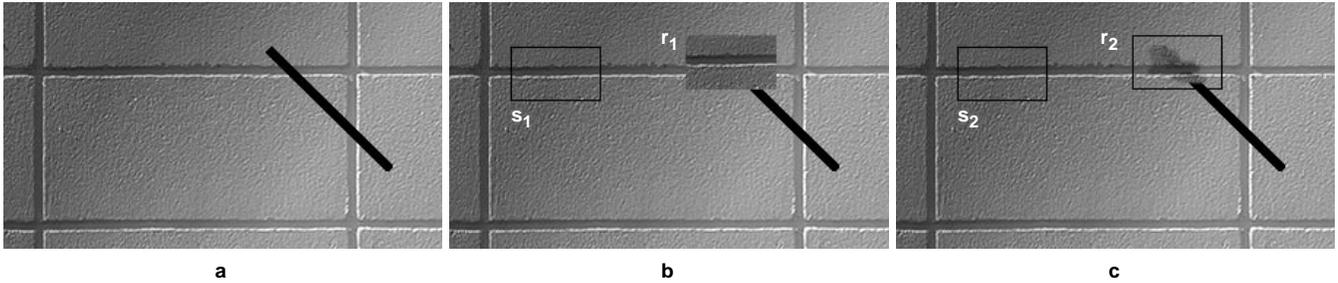
Figure 1: Problems with copying from another area of image. (a) Image with noise (noise is the black diagonal line). (b) The area s_1 has been copied into the area r_1 in an attempt to cover up the noise. Note the alignment and shading mismatch. The horizontal "cement" line appears broken in r_1 now and the area is darker than surrounding. (c) Using a cloning brush the area s_2 has been sampled and copied carefully into r_2 to maintain alignment of the horizontal cement line. But such a tool cannot solve the problems of mismatched intensity. The cloning source is darker than the destination area. Results of order statistics non linear filters like median and results of low pass filters are not shown here. But they do not work on images and types of noise shown here. See text for detail.

advantage of our algorithm is that the image intensity does not have to be uniform across the image.

The rest of the paper is organized as follows. Section 2 summarizes related previous works. Following a brief overview of POCS in section 3, our algorithm is described in section 4. Section 5 shows results of our algorithm and section 6 gives conclusions and possible future directions.

2 PREVIOUS WORK

Previous work on image noise removal can be divided into intra-frame and inter-frame techniques depending on where the information needed for removal comes from. Inter-frame algorithms copy needed pixels from preceding or succeeding frames. They may compensate for motion of object or camera by tracking key points of an image. Inter-frame methods fail when scratches run across many frames (such scratches are common because of the vertical motion of the film through projectors) or when there is too much camera activity. In either of these cases, the needed pixels cannot be found easily in the preceding or succeeding frames.

This paper describes an intra-frame algorithm assuming that other frames do not have the needed pixels. Previous intra-frame methods can be further classified as follows based on which information they use.

1. Use frequency domain information only (e.g low pass filter).

2. Use spatial domain information only

   (a) Median and similar order statistics filters.
   (b) Spatial statistical texture synthesis.
   (c) Cloning by copying pixels.

3. Use spatial and frequency domain information

   (a) Projections onto convex sets for band limited images.
   (b) Matrix algorithms for band limited images.
   (c) Spatial and frequency based statistical texture synthesis.

2.1 Frequency Domain Only

Frequency domain algorithms such as low pass filtering can capture global structure of the image but lose local control (line continuity, sharpness). As a result lines and other details become blurred. Since human visual system is very sensitive to details of an image like those conveyed by the lines, the result is unacceptable for removing noise that consists of many contiguous pixels. We have not shown results of low pass filtering here because such filters perform very poorly for the kind of noise and images shown in Fig. 10 (many contiguous noisy pixels in textured areas or in textured areas with prominent lines).

2.2 Spatial Domain Only

One problem shared by all spatial-only methods is that they have local control and information but do not have any information about the global structure of the image. The limitation to local neighborhood is due to practical computational constraints in some cases. In addition, some of these methods like median filtering etc. are inherently incapable of using the global information meaningfully.

Cloning tools of popular commercial image manipulation programs allow copying from another area of the image using brush like strokes. However, aligning reconstructed lines with existing lines by this method is time consuming and error prone. An even bigger problem is when the image has uneven intensity due to lighting conditions or inter reflections. In such cases, finding the same intensity source area can be difficult.

Fig. 1 demonstrates these problems with copying from another area of the image. Fig. 1(b) shows how the shading as well as the alignment can be different between source (s_1) and destination (r_1) areas. Fig. 1(c) shows how careful use of a manual cloning tool can somewhat ameliorate the alignment problem in this case. But without using the frequency domain information, such a tool can do nothing about shading mismatch as shown in the figure.

A recent survey of median and similar non-linear order statistics filters describes the advantages and shortcomings of these [7]. The problem with order statistics filters is when not enough correct information is available for meaningful order statistics. This is typically the case when numerous contiguous pixels are noisy.

Spatial domain texture synthesis algorithms [5] have shown remarkable results for stochastic type or small regular texture. However, these methods fail when the image has long range structure as in the image of brick wall in Fig. 10(a). The computational cost increases prohibitively for long range image structure because such algorithms use second order statistics.

2.3 Spatial And Frequency Domains

In the case of texture synthesis it is possible to use multi-resolution directional filters and then work with only single order statistics (histograms) as in [4]. This could be considered a spatial and frequency domain algorithm. However this method works only for stochastic texture or small regular texture. In addition [4] is not a noise removal algorithm. It is designed for generating large texture areas from sample images. It cannot be used to generate pixels that maintain continuity of prominent lines crossing noise pixels while retaining the noise free pixels.

Gerchberg-Papoulis and related algorithms [6, 2] are POCS based algorithms that use frequency and spatial domain information. However, they work only for band limited images and the
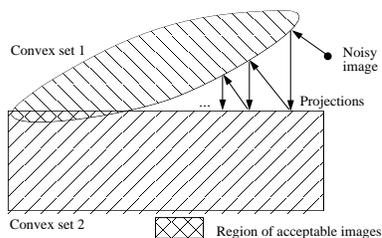
Figure 2: Pictorial representation of POCS. See section 3 for details.

Image with noise   **User paints mask on noise**   **Binary noise mask passed to algorithm**

Figure 3: The creation of noise mask for algorithm A1. Left image shows the actual noise (the dark uneven line running diagonally). Middle image shows the image with a binary mask that user has painted over the noise, to cover the noise. This does not have to be a straight line or rectangle, although it happens to be so in this case. The right most image shows the binary mask that will be passed to the algorithm. The middle image will become the r_0 input of the algorithm. See section 4 and Fig. 4 for details.

band limits must be known. In addition, recent extensions like [10] require expensive calculations of lines that intersect noise pixels. In other recent work [1] shows how to reduce band-limited interpolation and extrapolation problems for finite-dimensional signals to solution of a set of linear equations. Further, they show that the corresponding matrix is positive-definite with a spectral radius less than 1. The authors then derive properties of convergence of algorithms for different types of noises. Another matrix based method is [9]. These methods require that the image be band limited and the limits be known.

3 PROJECTIONS ONTO CONVEX SETS

Papoulis [6] introduced an algorithm for reconstructing band limited signals by alternating between signal and transform domains and applying the constraints of each domain. The constraints are preservation of known pixel values and enforcement of band limits. This approach was later generalized and given a geometric interpretation. A further generalization [11, 8] has come to be known as the method of projections onto convex sets (POCS). It allows the use of *any* information about the image (or any other signal) as long as the information can be represented as a closed convex set. Although the POCS theory was developed in the context of Hilbert spaces, for digital image restoration, it is convenient to restrict our attention to finite dimensional spaces. This space might be, for example, the space of all $M \times N$ complex matrices where the image has M rows and N columns.

Given a set $\mathcal{C}$ in such a space, $x, y \in \mathcal{C}$, we say the $\mathcal{C}$ is convex iff for any $0 \leq \mu \leq 1, \mu x + (1 - \mu)y \in \mathcal{C}$. $\mathcal{C}$ is closed if it contains all its limit points. See [11] for details. We'll use the words *closed convex set* and *constraints* interchangeably since the only constraints we'll be working with will select closed convex sets from a larger set. Projection onto a convex set consists of finding an image satisfying the constraint and "closest" to the image being projected. Intuitively, this can be thought of as making the least possible change to satisfy the constraints. See Fig. 2 for a pictorial representation of POCS.

Repeated projection onto all the convex sets is guaranteed to find an image that satisfies all the constraints if at least one such image exists. See the classic Youla and Webb paper [11] for more details. The advantages of POCS come from the fact that finding a direct projection onto the desired intersection is usually very difficult and expensive, while an efficient projection onto each set is more likely to be found. This is why formulating and solving a problem as POCS can be quite attractive computationally. Note that POCS is a general algorithm, with potential applications in many areas besides image restoration.

4 OUR ALGORITHM

The Fourier transform is an integration over the entire signal. After a transform, many of the essential global features of an image
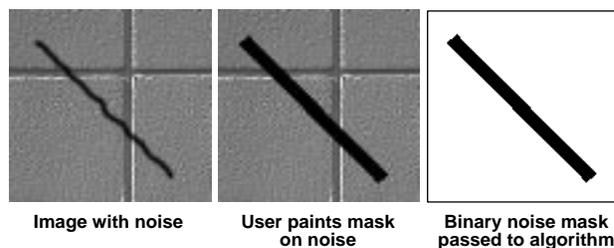
become localized, i.e come closer in the spectrum. These can include repeating patterns, overall image intensity, slow variation in intensity due to inter reflection or shading etc. On the other hand, rapidly varying stochastic texture or sharpness of lines and edges appear scattered in the spectrum. These are features that are localized in the spatial domain.

Clearly there is a need to combine these two for noise removal and texture synthesis. As we will show, POCS is a way of doing this in a clean and extendible fashion. In this section we first describe our basic algorithm A1, which combines the frequency and spatial domains in a POCS framework. Then we show how the use of POCS allows us to easily and cleanly extend A1 to solve important practical problems. The efficiency of the algorithm comes from the fact that each iteration requires fast operations on small subimages, not on the entire image.

4.1 Information Needed

No algorithm can restore an image or generate new texture without information, every algorithm needs some hint. Existing algorithms take a sample subimage (can be the entire image in some cases), which is usually taken from nearby pixels, analyze it, and extract hint information.

Our algorithm is no exception. It also needs some hint. In our case, a neighborhood of the noise (called repair subimage) is selected by the user to provide hint about the local spatial information. A nearby or similar subimage (called sample subimage) is selected by the user to provide a hint for the frequency information. The noise is located by the user creating a binary mask that covers it completely (the mask can be larger than the actual noise). Example of binary noise mask can be seen in Fig. 3. The black line in the noisy images in Fig. 10 can also be thought of as the noise mask covering the actual noise underneath. The algorithm starts with these images as the noisy image input.

This does not necessarily mean we need more information. We use two subimages, one for extracting global features, and one for maintaining local continuity. The algorithm does not place any restriction on choosing the location of the sample subimage. If these pieces of information can be obtained from one place, the sample and the repair subimages can overlap as in several subimages shown in Fig. 6 and 7. Or they can be far apart as in the case of some subimages in the brick wall, the first image in Fig. 6.

4.2 Base Algorithm (A1)

Fig. 4 gives a flowchart of the base algorithm A1. First, the global features are restored. This is best done in frequency domain by cor-
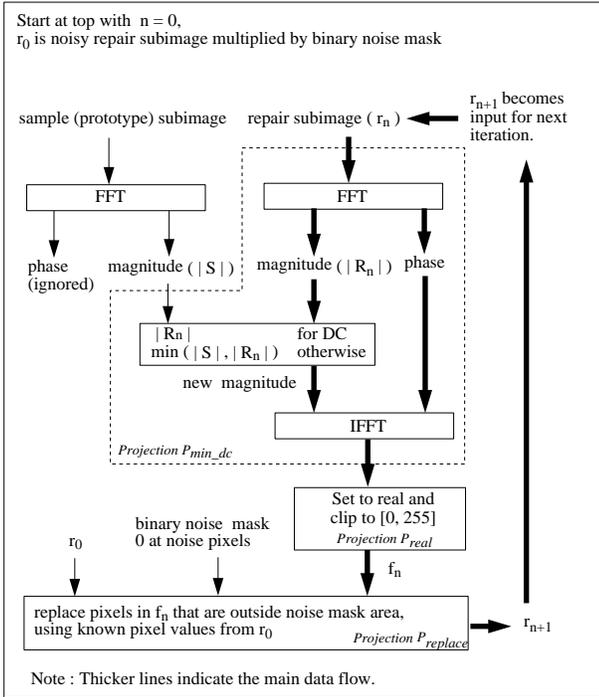
Figure 4: Details of our base algorithm A1. See section 4 for details.

Figure 5: Part of one iteration of A1 in spatial and frequency domains. (a)-(c) in first row show magnitudes of the Fourier transforms; (a) Repair subimage with binary noise mask ; (b) Sample subimage ; (c) Minimum of the first two, except at DC (center of FT) where the value from first one is used. The high energy in (a) due to the noise mask is seen as a diagonal white brightness. This has been considerably reduced in (c). Note that (a) and (b) are similar, except for the high energy due to the noise mask. This is because the repair and sample subimages are approximately translated versions of each other. (d)-(f) in second row show corresponding spatial domain data. See section 4.2.1 for details.

4.2.1 *Using Global Frequency Information*

The first projection operation that we use

$$\mathcal{P}_{min\text{-}dc}(r) = \text{IFFT}(M_S e^{i\,\text{phase}(R)}). \qquad (1)$$

involves M_S which is nearly a MIN operation, hence the name $\mathcal{P}_{min\text{-}dc}$. The M_S in the above equation is

$$M_S(r) = \begin{cases} \min(|R(u,v)|, |S(u,v)|) & \text{if } (u,v) \neq (0,0) \\ |R(0,0)| & \text{if } u=0, v=0 \end{cases}$$
$$(2)$$

Noise in general adds magnitude to the spectrum. Taking MIN effectively reshapes the repair spectrum into the sample spectrum. Our projection, $\mathcal{P}_{min\text{-}dc}$ has this nice property, and it is a projection onto a closed convex set (see below). M_S defined in Eq.2 is a kind of minimum taking operation on $|R(u,v)|$ and $|S(u,v)|$. The only exception is at DC, $u=0, v=0$ where the value of $|R(0,0)|$ is retained. The motivation for not modifying the DC value of the repair-spectrum is that it contains the value of the overall repair subimage intensity.

Also note that the phase is retained in Eq. 1, i.e while reshaping spectrum magnitude we leave the phase of the repair spectrum untouched. It turns out that the phase is reconstructed automatically over several iterations as in the phase reconstruction algorithms used in astronomy and other fields [3]. Phase reconstruction results in the automatic alignment of global features, e.g the alignment of the "cement" line in subimage r_a in Fig. 6. Doing this in frequency domain is easy. In spatial domain, an alignment would have required expensive block matching.

The underlying set

$$\mathcal{C}_{min\text{-}dc} = \{r : |R(u,v)| \le |S(u,v)|, (u,v) \neq (0,0)\}. \qquad (3)$$

is closed and convex and this can be proved similarly to proof on pp. 86 of [11] after making straightforward adjustments to go to the discrete case. In that proof, set $M(\omega_1, \omega_2) = S$ and $\Delta = \{(\omega_1, \omega_2) \neq (0,0)\}$. Note that $\mathcal{P}_{min\text{-}dc}$ is a projection because it makes the least change possible to make its input satisfy $\mathcal{C}_{min\text{-}dc}$.

recting the spectrum magnitude. The first step is to Fourier transform the repair and sample subimages. Since the repair-spectrum is corrupted due to noise, the sample-spectrum is used as a template for improving the repair-spectrum. It is very important to use this sample information correctly and this is where the theory of convex projections is important. The information must be represented as a convex set and an orthogonal projection to this set must be used. In addition this must be an efficient projection.

Several "obvious" ways of using the sample-spectrum are actually incorrect, in that they will yield algorithms that diverge because of non convexity. For example, one may think of using the sample spectrum to replace the repair spectrum completely. The practical objection to this is that good information is thrown away along with the bad. Theoretically too, this is unworkable because replacing the spectrum magnitude is not a projection onto a convex set.

Another plausible improvement might be to use a mixture of the two spectra, perhaps weighted by an α and $1-\alpha$ respectively, where α is chosen by the user. This too leads to a diverging algorithm which is incapable of noise removal. Again the reason is non convexity of the underlying set. Other plausible choices like using the high peaks of the sample spectrum etc. are also incorrect due to the same reason.

Before we start the description of our projections and convex sets, a note about notation. In the following equations r_0 is the starting repair subimage multiplied by the binary noise mask and s is the sample subimage (thus these are real matrices). r is an arbitrary complex matrix. r_0, s and r all have the same dimension. R and S are the Fourier transforms of r and s respectively i.e in a typical implementation $R = \text{FFT}(r)$, $S = \text{FFT}(s)$ where FFT stands for the Fast Fourier Transform operation. IFFT is the inverse FFT.
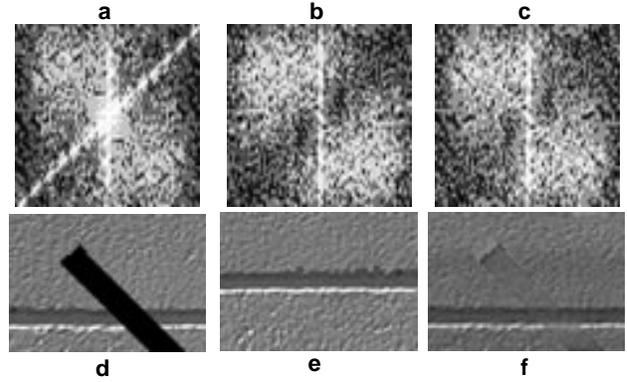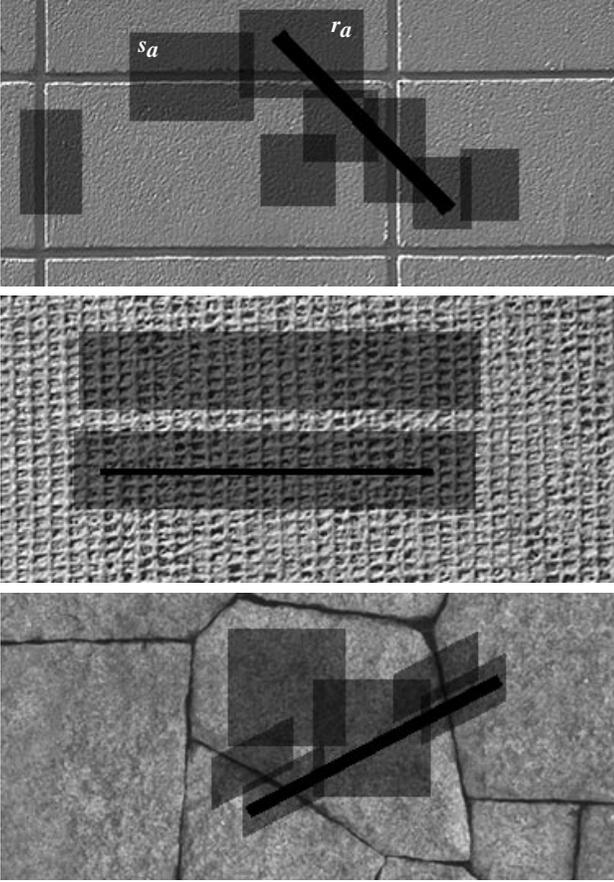
Figure 6: Repair and sample subimages used for examples in Fig. 10(a)-(c) with algorithm A1. Black line is scratch, sample and repair subimages are shown as dark patches. outlines. Prominent lines in sample and repair subimage don't have to be aligned. See e.g the thick horizontal line between bricks in s_a and r_a.

Thus the complete operation $\mathcal{P}_{min\text{-}dc}(r)$ consists of *(i)* taking an FFT of r *(ii)* creating a new spectrum magnitude by taking a minimum of $|R|$ and $|S|$ at all frequencies except DC where $|R(0,0)|$ is retained and and retaining the phase of R and, *(iii)* taking an IFFT using the new magnitude and the phase of R unchanged. See Fig. 4.

As described above, this projection $\mathcal{P}_{min\text{-}dc}$ thus has two purposes – to reshape the spectrum magnitude to match the prototype in order to get the global information correct and to align the prominent global features like prominent lines correctly. See Fig. 5 to see effect of one application of $\mathcal{P}_{min\text{-}dc}$ in frequency and spatial domains (for the purpose of displaying, Fig. 5(f) is actually shown after clipping the output of $\mathcal{P}_{min\text{-}dc}$ to real values between 0 and 255).

4.2.2 *Using Local Spatial Information*

At the end of $\mathcal{P}_{min\text{-}dc}$ (Fig. 4) we are back in spatial domain. The result is now closer to the answer. But since we modified the spectrum magnitude it is possible that after IFFT we now have imaginary component in the image matrix. Some values may also be outside the feasible range of $[0, 255]$. To bring the values back into the feasible range, the values of the spatial domain matrix are made real and clipped to $[0, 255]$. In addition, since the operation $\mathcal{P}_{min\text{-}dc}$ was in frequency domain it affects even the pixels outside of the scratch. These must now be corrected in spatial domain. This is
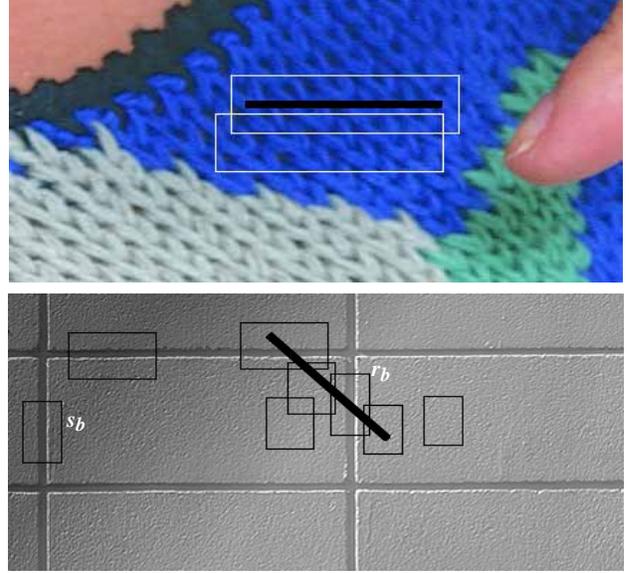

Figure 7: Repair and sample subimages used for examples in Fig. 10 (d) and (e) for algorithm A2 and A3 respectively. Black line is scratch, sample and repair subimages are shown as white or black outlines. For A3 (and to a lesser extent, for all algorithms), the repair and sample subimages can have very different shading. See for example s_b and r_b.

done simply by copying the known pixel values around the noise from the original repair subimage. These two rather simple projections are given below as equations, along with the closed convex sets. Proof of their convexity is simple and can be found in [11].

The convex set corresponding to the clipping to real values in $[0, 255]$ is

$$\mathcal{C}_{real} = \{r : r(j,k) \in \Re, 0 \le r(j,k) \le 255\}. \qquad (4)$$

The corresponding projection $\mathcal{P}_{real}(q)$ clips the input to a real value between 0 and 255.

Let W be the set of coordinate pairs where the binary noise mask is 0, i.e W is the set of locations under the noise mask pixels. The convex set corresponding to known pixel replacement is

$$\mathcal{C}_{rep} = \{r : r(j,k) = r_0(j,k), (j,k) \notin W\}. \qquad (5)$$

Let w be the binary mask which is 0 at noise pixel locations and 1 otherwise. Then the appropriate projection corresponding to the convex set $\mathcal{C}_{rep}$ is

$$\mathcal{P}_{rep}(r) = r(1-w) + r_0 w. \qquad (6)$$

4.2.3 *Iterations*

After applying $\mathcal{P}_{min\text{-}dc}$, $\mathcal{P}_{real}$ and $\mathcal{P}_{rep}$ we come to the end of the first iteration of A1. This process is then repeated. Thus the algorithm A1 can be written as

$$r_0 = \text{initial repair subimage} \times \text{noise mask} \qquad (7)$$
$$r_{n+1} = \mathcal{P}_{rep}\mathcal{P}_{real}\mathcal{P}_{min\text{-}dc}\, r_n. \qquad (8)$$

In the current implementation, the user sets the number of iterations. It is easy to implement other termination criteria. The algorithm is fast because it usually converges in under 10 iterations and each iteration requires 1 FFT, 1 IFFT and copying, all performed on a small neighborhood of the noise and *not* on the entire image. Results of A1 are shown in Fig. 10(a)-(c).
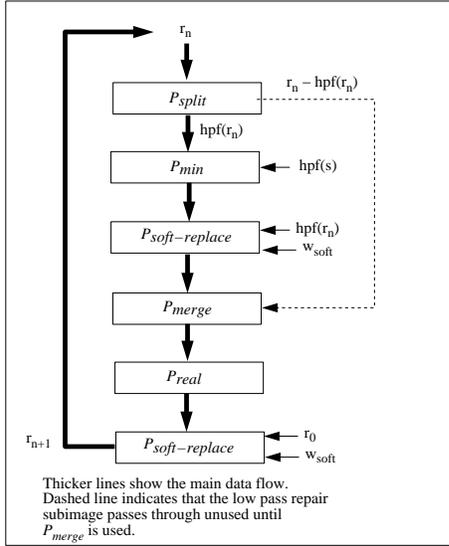
r_n

$r_n - \mathrm{hpf}(r_n)$

P_{split}

$\mathrm{hpf}(r_n)$

P_{min} ← $\mathrm{hpf}(s)$

$P_{soft-replace}$ ← $\mathrm{hpf}(r_n)$
← w_{soft}

P_{merge}

P_{real}

r_{n+1} $P_{soft-replace}$ ← r_0
← w_{soft}

Thicker lines show the main data flow.
Dashed line indicates that the low pass repair
subimage passes through unused until
P_{merge} is used.

Figure 8: Algorithm A3. See section 4.4 for details.

4.3 Soft Scratch Algorithm (A2)

In this and Sec. 4.4 we present two extensions to the basic algorithm A1. Our purpose in doing this is twofold. First, these two extensions solve some practical shortcomings of algorithm A1. Equally important, we show that by working in a POCS framework and using a dual-domain approach, important and substantial extensions can be made fairly easily. Hopefully, these extensions will also serve as guides for someone trying to extend the basic algorithm A1 in other ways.

The continuity of large prominent lines crossing the binary noise mask is generated by $P_{min\text{-}dc}$. But a transition in the local high frequency texture near the noise mask edge might be noticeable to human eye since the mask is sharp edged. It would be useful to use a soft edged mask for the noise to fix this potential problem.

This is easy to do with a slight modification in A1. In the final projection P_{rep} in each iteration of A1, use a soft edged noise mask instead of a binary mask. This new projection that we will call $P_{soft\text{-}rep}$ can be written as

$$P_{soft\text{-}rep}(r) = r(1 - w_{\mathrm{soft}}) + r_0 w_{\mathrm{soft}} \quad (9)$$

where w_{soft} is a soft edged noise mask. The underlying convex set $C_{soft\text{-}rep}$ can be written as

$$C_{soft\text{-}rep} = \{r(j,k) : r_0(j,k) = p(j,k)w_\alpha + q(j,k)(1 - w_\alpha)\} \quad (10)$$

where q is an arbitrary image and w_α is 1 outside binary noise mask, 0 inside binary noise mask and $\alpha(d)$ in soft noise mask edge region, and $0 <= \alpha(d) <= 1$ is a nice smoothly rising function like 1 - gaussian, depending on the distance d from the binary noise mask edge. Thus the algorithm can be now written similar to A1 using $P_{soft\text{-}rep}$ instead of P_{rep}. Showing that $C_{soft\text{-}rep}$ is convex is straightforward using simple algebra. The results of using A2 on a color image by applying it to the r, g, b channels is shown in Fig. 10 (d). More explanation of results is in section 5.

4.4 Split Frequency Algorithm (A3)

Notice that the example images used for A1 and A2 have had nearly uniform shading across the image, as in Fig. 10 (a)-(d). The next extension we describe removes this restriction of uniform shading.

The resulting algorithm is A3, shown in Fig. 8. The results of this algorithm A3 are shown in Fig. 10 (e). A comparison between A1 and A3 on a very unevenly shaded image is shown in Fig. 9. Although it may not be visible in the final printed paper, the circled area shows remnants of noise in the left image on which A1 has been applied. This problem is absent in the result of A3 on right in Fig. 9.

Note that overall variation in shading of an image is a global feature and so we choose the frequency domain to attack this problem. The basic idea is very simple – ignore the shading (which is a very large, global feature) by ignoring the low frequency components. Then, to the high frequency image, apply the projections similar to A2 followed by merging the effect of the shading. Now we go through the algorithm step by step. As shown in Fig. 8, the algorithm can be written as

$$r_0 = \text{initial repair subimage} \times \text{noise mask} \quad (11)$$
$$r_{n+1} = P_{soft\text{-}rep} P_{real} P_{merge} P_{soft\text{-}rep} P_{min} P_{split} \, r_n. \quad (12)$$

The main new projections are P_{split} and P_{merge}. The first of these splits the input image r into two images, a high pass filtered $\mathrm{hpf}(r)$ and a low pass filtered $r - \mathrm{hpf}(r)$. We use a gaussian filter to create $\mathrm{hpf}(r)$ and $r - \mathrm{hpf}(r)$. P_{merge} does the reverse of P_{split} by simply adding the output of previous projections, which is the *processed* high pass filtered component of the repair subimage, with the *unprocessed* low pass filtered repair subimage as shown in Fig. 8.

Note that since the lower frequencies are being ignored during processing we can simplify $P_{min\text{-}dc}$ to P_{min} by simplifying M_S (2) of algorithm A1 to $M_S(r) = \min(|R(u,v)|, |S(u,v)|)$. Thus now we are taking MIN across the entire spectrum, including the DC unlike A1.

After this, a replace operation will be performed. Since we are using a high pass filtered repair subimage, the effect of the noise will be seen outside the noise mask after P_{split}. Therefore when we replace the known pixel values, we should use a better estimate in each iteration. This is done by using the latest $\mathrm{hpf}(r_n)$ instead of $\mathrm{hpf}(r_0)$ as input for $P_{soft\text{-}rep}$. Thus the first $P_{soft\text{-}rep}$ of A3 is similar to equation 9 of A2 except that $\mathrm{hpf}(r_n)$ is used instead of r_0 in equation 9. After merging the result with the low frequency image using P_{merge}, P_{real} is applied which is the same as in A1 or A2. Finally the known values are replaced using the original r_0 just as in A1 or A2. It is easy to show convexity of the underlying sets for P_{merge} and P_{split} using linearity of the Fourier transform.
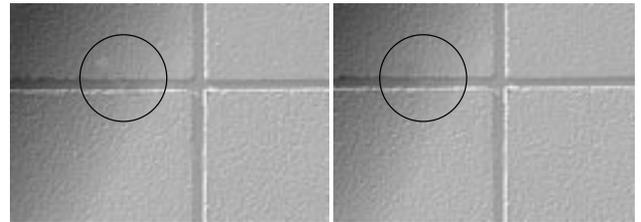

Figure 9: Comparison of A1 and A3 on an image with intensity variation. In the circled area some leftover noise is visible in the A1 result on the left whereas the A3 result on the right is cleaner [the difference may not be obvious in the final printed version]. See section 4.4 for details.

5 RESULTS

Fig. 10 shows noise removal using our algorithms A1, A2 and A3. The images shown have stochastic and regular textures and prominent systematic and random lines. The black line running across

the first image of each group of images in Fig. 10 is the noise. (a)-(c) show the result of algorithm A1, (d) shows result of A2 and (e) shows result of A3.

The first four images, shown as group (a) are – clockwise from top left – image with noise, image after 1, 2 and 10 iterations of our algorithm A1. The other groups shown in (b)-(e) show only the noisy image and the image after 10 iterations of our algorithms.

The sample and repair subimages used for (a)-(c) are shown in Fig. 6 as darkened patches. It is important to note that the formulation of the algorithm makes the selection of subimages easy. No manual alignment of features is necessary during subimage selection. This can be seen in the Fig. 6. Note for example that the subimage r_a has the cement line running towards the bottom of it while that horizontal line is nearly in the middle in the corresponding sample subimage s_a.

The next improvement to A1 is the algorithm A2 which uses a soft edged mask instead of a binary mask. Fig. 10 (d) shows noise removal from a color image using our algorithm A2. A2 is applied to r, g and b channels separately. The subimage selection is shown in Fig. 7.

Our last and most powerful algorithm is A3 which is able to handle images with varying intensity which might have been caused by shading or inter-reflection etc. The result is shown in Fig. 10 (e) and the subimage selection is shown in Fig. 7. Note that for example, in Fig. 7, s_b and r_b are two subimages with very different intensity, one is darker than the other. To some extent variation in intensity is tolerated by all three algorithms, though A3 is best able to deal with that.

See sections 4.2, 4.3 and 4.4 for algorithms A1, A2 and A3 respectively. Although not shown here, removing non-straight contiguous noise, or noise removal from synthetic images with precisely repeating patterns (and no stochastic texture) requires no extra work for our algorithm. In fact synthetic images with exactly repeating patterns are reconstructed perfectly.

The limitations of our approach are that the contents of sample and repair subimages must be approximately translated versions of each other. The can be seen in the subimage selections in Fig. 6 and Fig. 7. If the prominent lines and texture in repair and sample subimages are rotated versions of each other or there is perspective or other distortion between the two, then the algorithm will not work. Thus for example, we can't use a vertical feature as in s_b of Fig. 7 to fix the noise in an area which has a horizontal feature.

6 CONCLUSIONS AND FUTURE WORK

A fast iterative algorithm for image noise removal has been described. While most existing algorithms have worked solely in spatial or frequency domain, our algorithm works in both domains, making it possible to fully exploit the advantages from each domain. Although a few previous algorithms combine frequency and spatial domain information [6, 2], they required the image to be band limited, required that the band limits be known. Our algorithm does not place this limitation.

As shown in the results, with a judicious choice of operations (in terms of constraints and projections) and domains in which the operations work, our dual-domain approach can (1) reconstruct many contiguous noisy pixels, (2) reconstruct textures even when they are large featured, (3) maintain sharpness, (4) maintain continuity of features (e.g., lines) across the noisy region. These advantages make the algorithm very useful in many areas.

Important applications of this algorithm are in the field of film and video post production: for removing wires used in special effects scenes and for restoring old films and photographs that have become scratched.

Our algorithm is based on a general framework of POCS and can be extended in a clean way. Besides the constraints and the projections described in this paper, any image analysis and/or feature extraction techniques that are described in a closed convex set can be plugged into the iteration loop. Also, the choice of domain is not limited to the spatial and frequency domains. For example, one could choose the Wavelet transform if multiresolutional analysis is desired.

One of our motivations for presenting this work was to increase awareness about the general and powerful method of Projections Onto Convex Sets in the graphics community. To us, it appears to be an interesting way of thinking about various problems and until now it has been popular only amongst the image processing community. It is possible to imagine other uses for POCS besides image restoration, by using appropriate convex sets. Examples related to our work could be : restoring missing 3D geometry data acquired by range data acquisition systems, filling occluded information during image based rendering etc.

In the context of image restoration there are a few areas that need attention. An interactive brush implementation in which the sample and the repair subimages are automatically selected based on a brush stroke, would be useful. A multi-frame extension which allows better inter-frame continuity is another important extension. We have found that when we restore multiple frames of a still image in which the wire noise is moving some kind of moving noise is visible when the images are observed in sequence. But when the movie is stopped, the restorations seems good and the noise disappears. A straightforward extension by using 3D Fourier transform using a few frames at a time has not worked. Our color image processing is also rather naive and needs more attention. A study of how the variation in binary noise mask size affects the performance would also be desirable. Finally, what to do about rotation of features or perspective distortion between sample and repair subimages are other areas that need attention. We are currently working on these issues.

References

[1] FERREIRA, P.J.S.G. and PINHO, A.J. "Errorless Restoration Algorithms for Band-Limited Images", *Proc. IEEE Intl. Conf. Image Proc. (ICIP)*, III, 157-161, 1994.

[2] FERREIRA, P.J.S.G. "Interpolation and the Discrete Papoulis-Gerchberg Algorithm", *IEEE Trans. Sig. Proc.*, 42, No. 10, 2596-2606, Oct 1994.

[3] FIENUP, J.R. "Phase Retrieval Algorithms: a Comparison", *Appl. Opt.*, 21, No. 15, 2758-2769, 1982.

[4] HEEGER, D.J. and BERGEN, J.R. "Pyramid-Based Texture Analysis/Synthesis", *Proc. SIGGRAPH 95*, 229-238, 1995.

[5] MALZBENDER, T. and SPACH, S. "A Context Sensitive Texture Nib", *Communicating with Virtual Worlds*, Thalmann, N.M. and D. Thalmann (eds.), Springer-Verlag Tokyo, 1993.

[6] PAPOULIS, A. "A New Algorithm in Spectral Analysis and Band-Limited Extrapolation", *IEEE Trans. Cir. & Sys.*, 22, No. 9, 735-742, 1975.

[7] PITAS, I. and VENETSANOPOULOS, A.N. "Order Statistics in Digital Image Processing", *Proc. IEEE*, 80, No. 12, Dec 1992.

[8] SEZAN, M. I. and STARK, H. "Image Restoration by the Method of Convex Projections: Part 2 – Applications and Numerical Results", *IEEE Trans. Med. Imag.*, 1, No. 2, 95-101, 1982.

[9] STROHMER, T. "On Discrete Band-Limited Signal Extrapolation", In *Mathematical Analysis, Wavelets, and Signal Processing*, Ismail, M. et. al. (eds.), AMS Contemporary Mathematics, 190, 323-337, 1995. Also available from *http://tyche.mat.univie.ac.at*, the home page of NUHAG in University of Vienna.

[10] SUN, H. and KWOK, W. "Concealment of Damaged Block Transform Coded Images Using Projections onto Convex Sets", *IEEE Trans. Image Proc.* , 4, No. 4, April 1995.

[11] YOULA, D.C. and WEBB, H. "Image Restoration by the Method of Convex Projections: Part 1 – Theory", *IEEE Trans. Med. Imag.*, 1, No. 2, 81-94, 1982.
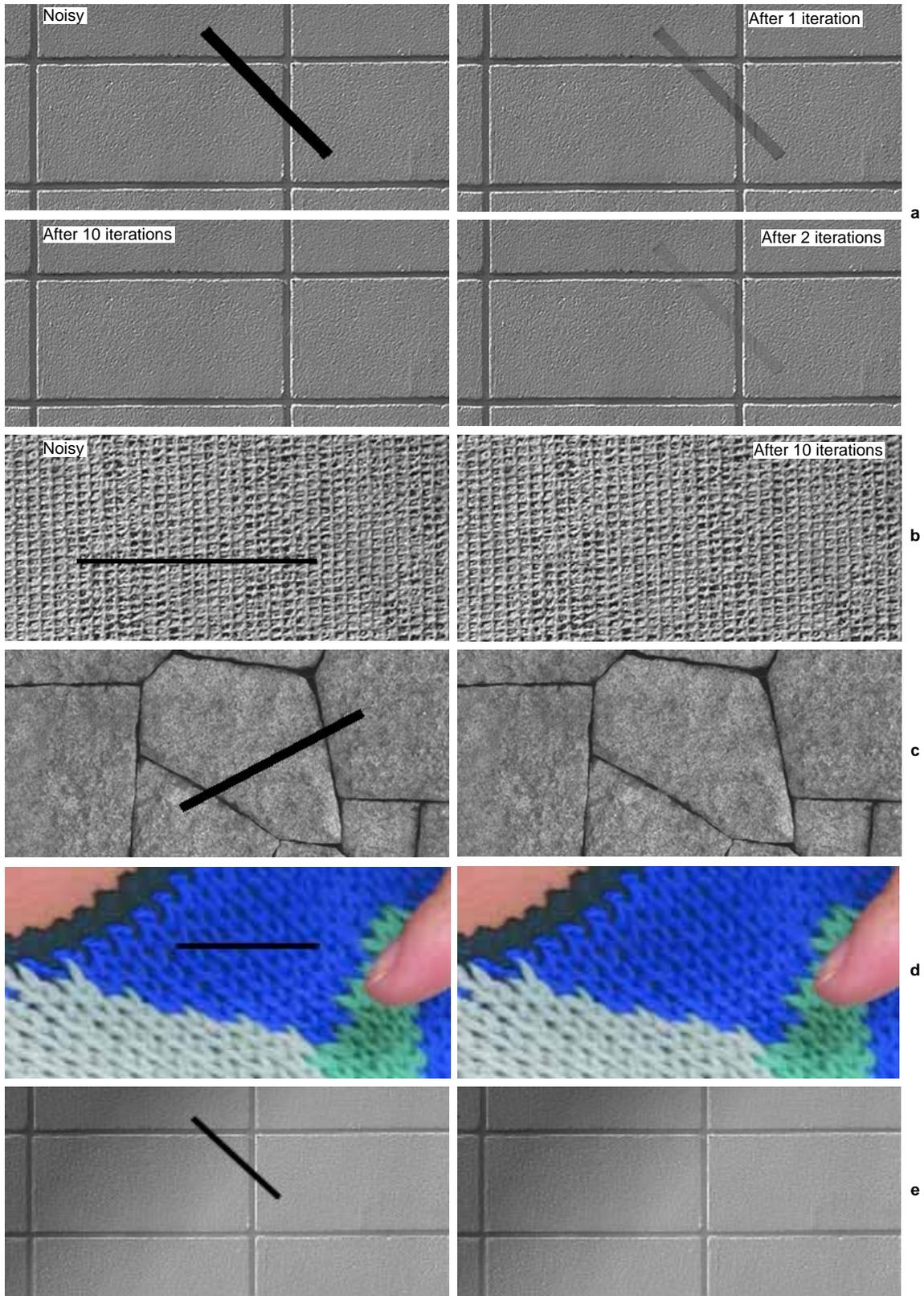
Figure 10: Results of algorithms A1, A2, A3. (a)-(c) Show removal of noise using A1 from textured images with (a) systematic long distance structure (cement lines in brick wall) (b) small regular texture (fabric) and (c) randomly placed prominent lines (stone wall). Images are 377×176, scratches approx. 9, 4 and 8 pixels wide. (d) Shows a simple application of A2 to a color image by applying A2 to each channel. (e) Shows results of A3 on an image with intensity varying across the image. See section 5 for more details and Fig. 6 and Fig. 7 for sample and repair subimages.