

ITERATIVE CONSTRAINED DECONVOLUTION METHODS

APRIL, 2002

Many product and system designs suffer from signal processing problems. The signal of interest is degraded by noise, blur and the presence of other extraneous data. The important data is hidden by noise and less important signals. Separating the data stream into its useful components is essential to product success. Plexar has solved many problems in this arena. The following is one example.

Introduction

This document briefly describes the use of an iterative constrained deconvolution algorithm for solving a deblurring problem in image reconstruction. In the specific instance, we had the projection of a pin inside a water phantom. The aim was to reconstruct the image of the pin and the water phantom using the acquired projection views.

Problem

The projection views of the pin were highly degraded (blurred) by scatter. We found that a straightforward image reconstruction did not give us back the pin at the correct dimension. Thus we had to use some sort of deblurring method to extract the true physical dimension of the pin in the reconstructed image.

Observations/Experimentation

From the experimental scans, we noticed that the extent of the blur of the pin in the projection depended on the distance of the pin from the center of the scan circle. We performed experimental scans to determine the behavior of the blur with distance. We then fit the data to a Gaussian distribution. This depth dependent Gaussian distribution is thus the spatially varying point spread function (PSF), of the imaging system (see Figure 1).

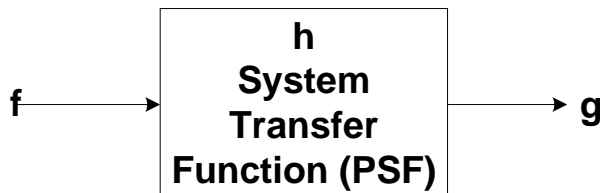


Figure 1: System transfer function of imaging component(s)

Solution

We found that in the presence of noise and other factors, this deblurring task did not lend itself to a straight blind deconvolution, Figure 2.

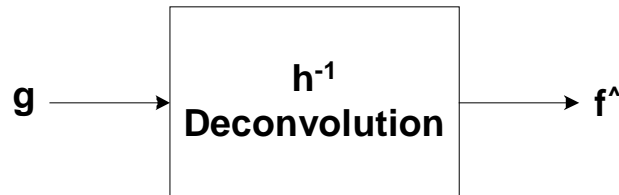


Figure 2: Inverse of the system transfer function that restores the true image

Blind convolution has pitfalls due to any of the following factors:

- (a) Zeros in the system transfer function
- (b) Inaccuracy of the values of h cause larger inaccuracies in the inverted function
- (c) Sparse nature of h
- (d) Computationally intensive due to large matrix size
- (e) Inability to model h accurately

Considering these factors, we chose to use iterative algorithms that tackle the deconvolution problem by iteratively solving for the inverse. The advantage is that these algorithms have certain convergence properties that make them more stable. One disadvantage is that they can be computationally intensive. We used variations of the Van-Cittert algorithm to accomplish our deblurring task.

Van-Cittert Algorithm

One example of an iterative deconvolution algorithm is the Jansson-Van Cittert iterative deconvolution algorithm. The block diagram is shown in Figure 3. The logic of the algorithm is that an initial guess, f^0 , is made of the true image. This initial estimate is operated on by the known system transfer function to give an output image, g^0 . If the initial guess is correct, the difference of the observed image and the output, $(g - g^0)$ should be zero. If the difference is not zero, this difference is weighted by a relaxation factor, α , and added to the initial guess to yield the first iteration estimate, f^1 of the true image. This algorithm is repeated until the error in the output function is below a threshold or until the update is no longer significant. For our purposes, we modified the algorithm slightly.

The “process” box is an illustration of the Van-Cittert Algorithm with the modifications. It can be as simple as a function that transforms from measurement space, g , to the output space, h . It could contain additional constraints to the iterative process like convergence testing.

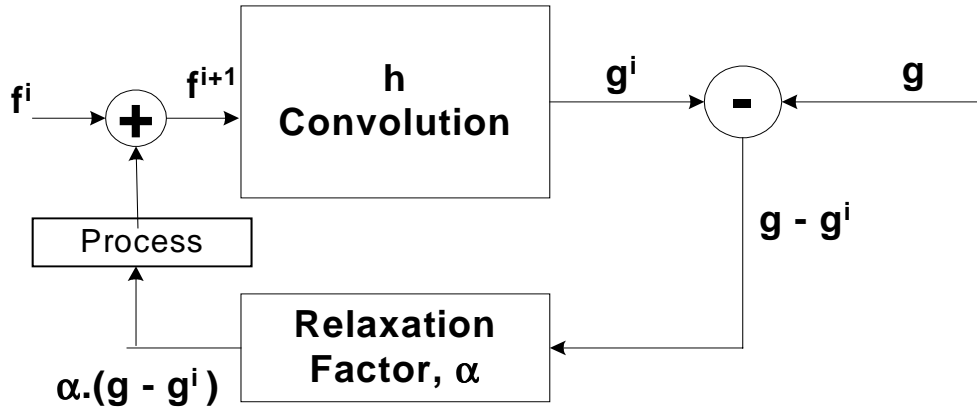


Figure 3: Modified Van-Cittert iterative constrained deconvolution algorithm that iteratively restores the true image

The advantage of such a method is that the process is fairly stable in the presence of noise and error in modeling of h when compared to the non-iterative deconvolution. There are other constraints that can be placed on this iterative algorithm to make it more stable in the presence of noise. In our application we used the “process” functionality to make the system less susceptible to noise and to apply some positivity constraints on the signal .

Before backprojection, the observed projections were iteratively deblurred using the h function that was appropriate for the pixel that was being reconstructed. This deblurring algorithm along with other processing functions gave us back the true physical dimensions of the pin.

Results

Figure 4 shows the behavior of the blurring function, h , with depth. The sharpest signal occurs when the region of scan is close to the surface of the skin.

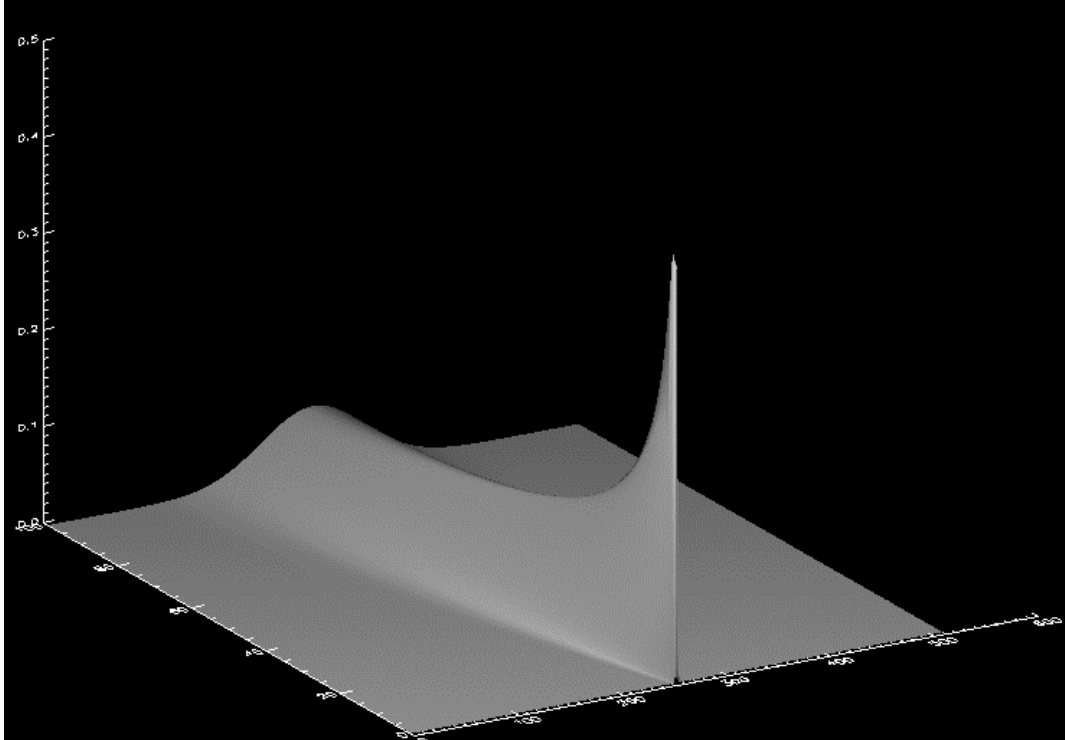


Figure 4: Figure showing the PSF of a signal as a function of depth. The deeper the location of the point, the wider the PSF.

Figure 5 shows the result of a Van-Cittert (modified) deconvolution of the signal shown as a thick line. Deconvolution of this signal at various depths is shown as well. A pixel at the deepest point of the image requires a larger amount of deblurring to counter act the large blurring that it suffers during the formation of the projection data during imaging.

The information in Figure 5 is ordered in a table and can be looked up by pixels based on their depth for further processing, in this case image reconstruction. Figure 6 shows the information in Figure 5 in the form of a shaded surface. Figure 7 shows the output of the filtering process as a function of the iteration number.

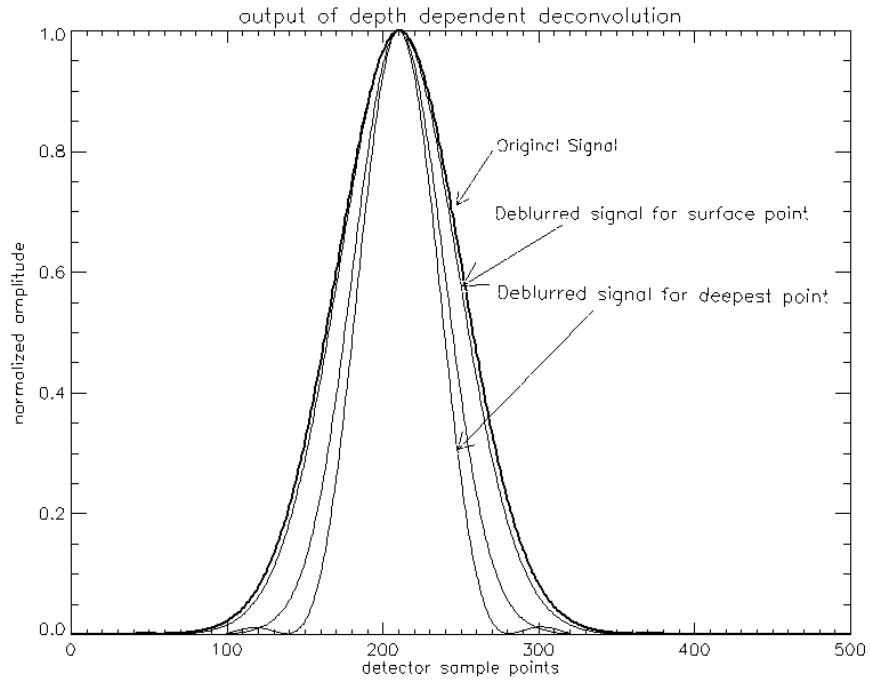


Figure 5: Results of deblurring (normalized to 1.0) of 1D signal (thick line) by various depth dependent PSFs in the modified Van-Cittert algorithm.

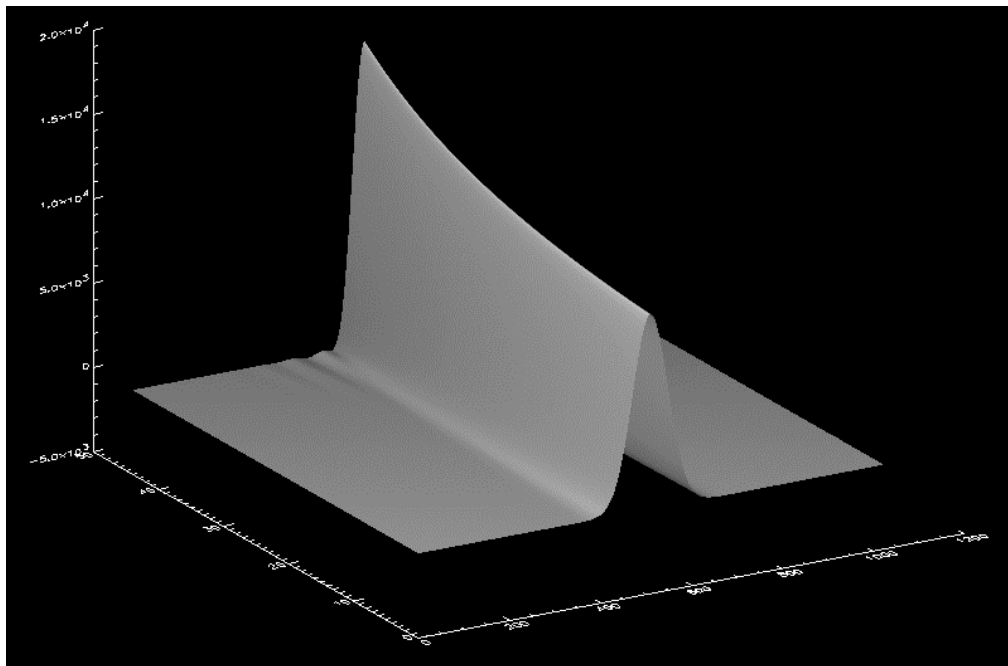


Figure 6: CORRECTION PSF as a function of depth. The deeper the location of the point, the narrower is the CORRECTION PSF.

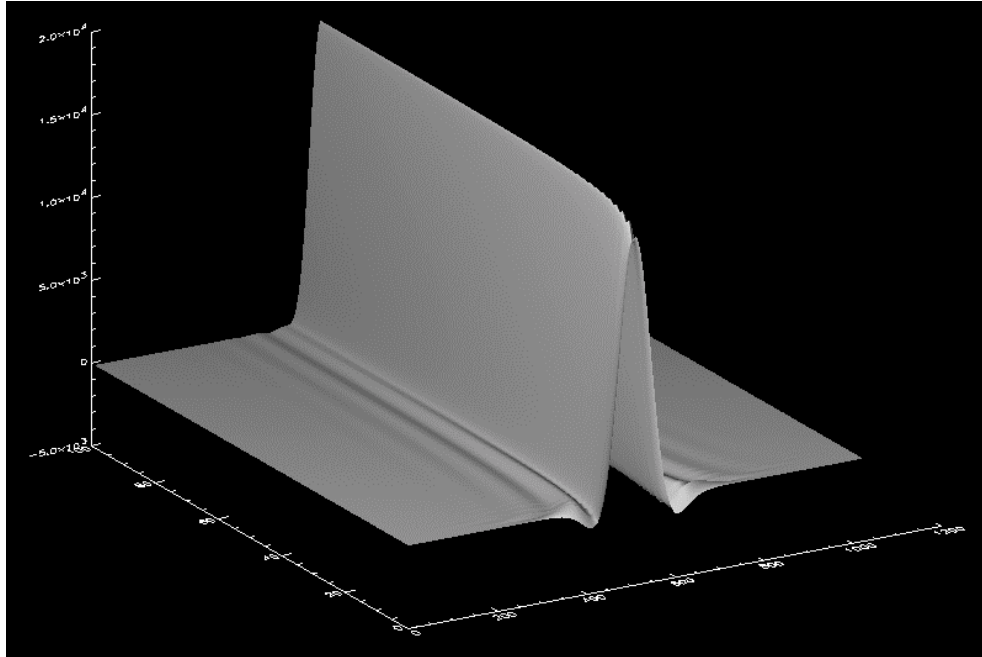


Figure 7: CORRECTION PSF as a function of iteration for a fixed depth. The closest information refers to the CORRECTION PSF in the first few iterations.

About Sastry K.L.A.

Sastry is one of 3 Ph.D.'s on Plexar's staff. He has been with Plexar for 8 years and has resolved innumerable engineering issues for Plexar's clients. He has a Ph.D. in Biomedical Engineering from the University of Akron, an M.S. in Biological Sciences and a B.E. in Electrical & Electronics Engineering.