AN ABSTRACT OF THE THESIS OF

Sharath Kumar Dhamodaran for the degree of Master of Science in Electrical and Computer Engineering presented on April 5,2013.

Title: Pattern Discovery in Noisy Images

Abstract approved: ____

Sinisa Todorovic

The Focused Ion Beam (FIB) tool is a versatile instrument for nano-machining in circuit editing. Circuit editing is one of the most important steps in the design of an electronic circuit on a chip. Circuit editing can be improved by imaging of silicon plates and analyzing the resultant images. However smaller features in FIB produce lower signal-to-noise (S/N) ratio in images, and thus render visual feedback increasingly difficult. For this Master thesis, we focus on analyzing a set of Focused Ion Beam grey valued images. Our goal is to reduce the noise in images, and apply an appropriate filter, for feature extraction. This is expected to allow us to see the significant features of the silicon plates.

The main challenge is that the noise and its level are unknown. Also the significant features of the silicon plate are unknown. Therefore the problem that we face here is discovery of an unknown pattern in images, under unknown noise. We make the assumption that noise is a superposition of a Gaussian noise, Salt-and-Pepper noise and affine perturbations. We seek the significant features using a least squares optimization to solve for the latent affine transformation, and recover the most significant pattern in the images. The improved images enable the tool user to recognize critical features more reliably for circuit editing.

Pattern Discovery in Noisy Images

by

Sharath Kumar Dhamodaran

A THESIS

submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Master of Science

Presented April 5, 2013

Commencement June 2014

Master of Science thesis of Sharath Kumar Dhamodaran presented on April 5, 2013

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Sharath Kumar Dhamodaran, Author

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Dr. Sinisa Todorovic for being such a wonderful mentor. Dr. Sinisa helped me discover that the problem that I was trying to work on was a hard and also an interesting problem. He guided me in the right direction and always made himself available for discussions in spite of his busy schedule. I thank my mentors at Intel Corporation, Teresa Morrison and Haripriya E Prakasam for their support in helping me work on the project. My interactions with them greatly improved my understanding of image processing. I also thank Dr. Huaping Liu and Dr. Xiaoli Fern for agreeing to be on my committee. I was introduced to many concepts in statistics and machine learning through their courses.

I would also like to thank all my friends (you know who you all are ☺) for their help, motivation and encouragement during my graduate study. My deepest gratitude goes to my father Mr. L.Dhamodaran, my mother Mrs. D.Gaja Gowri, my sister D.Suveetha and my other family members for their endless love and support and more importantly, their patience. Finally, I express my heartfelt thanks to Swathi for being such a wonderful and a supportive person who stood with me in all difficulties throughout my masters studies.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

1. Introduction

1.1. Motivation and Background

The Focused Ion Beam (FIB) tool is a versatile instrument for nano-machining in circuit editing. Circuit editing is one of the most important steps in the design of an electronic circuit on a chip. Circuit editing can be improved by imaging of silicon plates and analyzing the resultant images. However smaller features in FIB produce lower signal-to-noise (S/N) ratio in images, and thus render visual feedback increasingly difficult. [28]

Usually, Focused Ion Beam (FIB) images are used for comparing regular patterns on the same site of the plate. However these FIB images are corrupted by a large amount of noise. Also, the imaging process might introduce variations in site, viewpoint, and illumination as illustrated in Figure 1.

The main challenge is that the noise and its level are unknown. Also the significant features of the silicon plate are unknown. Therefore the problem that we face here is discovery of an unknown pattern in images, under unknown noise. The Ion Beam images contain Salt & Pepper noise, as well as some additional noise. This is also common in other coherent imaging systems such as Laser, Scanning Electron Microscope (SEM), acoustic and Synthetic Aperture Radar (SAR) imagery [28]. To reduce this noise, most existing methods make the assumption that the noise can be modeled as combination of Salt-and-Pepper and Gaussian. Given such a noise model, we can apply suitable filters towards noise reduction.

Current Focused Ion Beam techniques use median filter. The limitation of this method is that median filters destroy the true signal, and we might lose significant features. We make the assumption that noise is a superposition of a Gaussian noise, Salt-and-Pepper noise and affine perturbations. We seek the significant features using a least squares optimization to solve for the latent affine transformation, and recover the most significant pattern in the images. Applying median filters of different sizes is the state-of-the-art for this problem. This MS thesis significantly advances the existing methods.

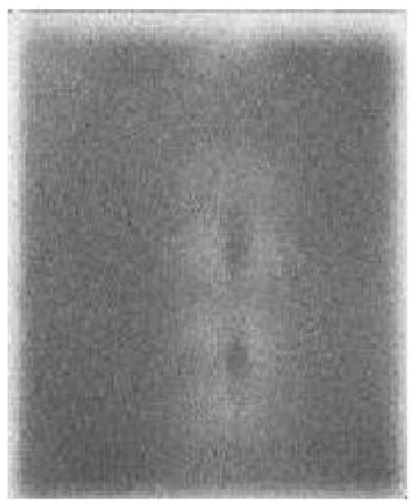


Figure 1:An Image generated from the Focused Ion Beam (FIB) tool

We tried different techniques that are described in detail in chapter 3. After the extensive experimentation, we have adopted the final approach explained in section 1.2

1.2. Overview

This section introduces the final approach in dealing with the problem of unknown noise and unknown features. The following is the sequence of steps that we used

1) Preprocessing
2) Geometric Transformations
3) Least Squares

The first step in our approach is the preprocessing step that includes histogram equalization and image filtering. Histogram equalization is a very common technique that is used in many image processing applications. We experimented with different filters to check if we were able to see the features present in the images. Since we could not eliminate the noise to a greater extent, we adopted a geometric transformation technique on top of the first method. There are many ways by which geometric transformation is done. Some of the common methods are the rotation, reflection, resizing and translation methods. Our images were perturbed with minimal angle rotation and resizing to generate ensembles.

At this stage, a stack of more than 1000 images were generated for each features. We calculated the mean of the resulting stack of images to check if we see any distinct features in the images. The last approach in our discovery of unknown patterns in nosy images was the Least Squares method. We utilized a simple as well as a robust iterative least squares algorithm technique to bring out the significant features in the noisy images. We tested the results with different training images (images that seemed to have different features while looked through a human eye) and were able to obtain significant feature patterns in the images without blurring the image too much.

2. Literature Review

Significant progress in low-level vision has been achieved by algorithms that are based on energy minimization. [44] Typically, the algorithm's output is calculated by minimizing an energy function that is the sum of two terms: a data fidelity term which measure the likelihood of the input image given the output and a prior term which encodes prior assumptions about the output. Examples of tasks that have been tackled using this approach include optical flow estimation [22, 10], stereo vision [11, 12] and image segmentation. An important subclass of these problems is when the output is itself a natural image. This includes problems such as transparency analysis [17], removal of camera blur [13] image denoising and image inpainting [21].

For low-level vision tasks where the output is a natural image, the prior should capture some knowledge about the space of natural images. This space is obviously a tiny fraction of the space of $N \, x \, N$ matrices, but how can we characterize it? Some of the earliest energy-based methods used a quadratic smoothness assumption [15]. Thus the energy was simply the sum of squared local derivative operators. This corresponds to a Gaussian prior on images and would be most appropriate if the distribution of natural images were indeed Gaussian. Unfortunately, images are very non Gaussian.

When derivative-like filters are applied to natural images the distribution of the filter output is highly non Gaussian - it is peaked at zero and has heavy tails [19, 23, 20]. This property is remarkably robust and holds for a wide range of natural scenes. Non Gaussian marginals are also obtained for optical flow & stereo [21, 16]. Thus a Gaussian prior is not appropriate and more recent algorithms typically assume a robust, nonquadratic energy on local derivatives [10, 12]. Ideally, one would like to learn the functional form from a training data. Also, one would like to know whether basing the energy functions on local derivatives is the best thing.

In a seminal paper [27] Zhu and Mumford showed how to address both questions using the principle of maximum likelihood. Denoting by x an image, they defined the probability of an image by means of an energy function that depends on the output of linear filters ω_k applied to the image:

$$Pr(x; \{\omega_k, \psi_k\}) = \frac{1}{Z(\{\omega_k, \psi_k\})} e^{-\Sigma_{i,k} E_k(\omega_{i,k}^T x)} \quad ..(1)$$

$$= \frac{1}{Z(\{\omega_k, \psi_k\})} \prod_{i,k} \psi_k(\omega_{i,k}^T x) \quad ..(2)$$

where ***i*** is an index over image pixels and ***k*** is an index over linear filters. $\omega_{i,k}^T$ is the result of applying the linear filter ω_k to image ***x*** at location ***i***. The ***partition function*** $Z(\{\omega_k, \psi_k\})$ is an explicit normalization constant and is defined by:

$$Z(\{\omega_k, \psi_k\}) = \int_x \prod_{i,k} \psi_k(\omega_{i,k}^T x) dx \quad ..(3)$$

For an arbitrary energy function, the partition function is intractable since it requires integrating over all possible images. Note that equation **2** contains special cases with some well-known priors used in low-level vision. If the filters are just horizontal and vertical derivatives and the energy functions are quadratic, this gives the classical smoothness assumptions. If the filters are horizontal and vertical derivatives while the energy functions are robust norms, this gives the more modern robust smoothness assumptions. Zhu and Mumford proposed learning both the set of filters ω_k and the corresponding energies E_k from data by maximizing the likelihood of the training set. Specifically, they assumed the filters were chosen from a discrete set of oriented derivative like filters while the energy functions could be arbitrarily shaped. For derivatives at the nest scale, the learned potentials were qualitatively similar to the log histograms and peaked at zero. But at the coarser scales the potentials were ***inverted***, they had a minimum at zero, even though the log histograms have a maximum at zero. This inversion effect is more pronounced the coarser the filters.

Despite the progress made by using maximum likelihood to learn energy functions for low-level vision, two significant problems remain. The first problem is that performing the learning is excruciatingly slow. In both [27, 21], learning is performed using gradient ascent -by following the gradient of the log likelihood in equation 2. This gradient includes the gradient of the partition function which is intractable. Zhu and Mumford used Gibbs sampling in order to estimate the gradient of the partition function, and noted that it could take many sweeps of sampling to converge to a suitable gradient. Since sampling needs to be performed before each gradient descent step, learning is extremely slow even when faster sampling techniques are used [26, 25]. Roth and Black used an approximate sampling method called contrastive divergence. [14]. Even with this approximation, they noted that learning is very slow.

A second problem with existing approaches based on maximum likelihood is that it is extremely difficult to actually compare the likelihood for two competing models. Again, this is due to the intractable partition function in equation 2. Even if we wait long enough for a sampling algorithm to give us fair samples from the model, calculating the partition function from a finite number of samples is a difficult problem [18]. Thus, we have no way of currently saying whether the non-intuitive findings represent a local minimum of the optimization, or whether they really give higher likelihood to natural images.

$$Pr(x; \{ \omega_k, \psi_k \}) = \frac{1}{Z(\{ \omega_k, \psi_k \})} e^{-\Sigma_k E_k(\omega_k^T x)} \quad ..(4)$$

$$= \frac{1}{Z(\{ \omega_k, \psi_k \})} \prod_k \psi_k(\omega_k^T x) \quad ..(5)$$

For this model it has been shown that when the potentials are Gaussians, the optimal filters are the ***minor components*** -the principal components of the data with minimal eigenvalue [24]. Thus training in this case is extremely fast and requires just one eigenvector computation on the training set. It has also been shown that in the ***under complete* case** -when the number of filters is smaller than the dimensionality of x, the partition function can be calculated exactly using the singular value decomposition of the vectors ω_k. Unfortunately, neither of these results is directly applicable to the case we are interested in -as mentioned earlier images are highly non-Gaussian so Gaussian potentials are not appropriate. Furthermore, the translation invariance of images would suggest that our prior also needs to be translation invariant as in the FOE model. If we have K filters in the FOE model, then the model is K times over complete.

It would thus be desirable to obtain (1) a fast algorithm for learning good filters and (2) a way to calculate the partition function in the over complete, non-Gaussian case. In this paper we provide both. We derive tractable lower and upper bounds on the partition function based on the Fourier transform of the filters { ω_k}. We also show how to calculate high likelihood filters using iterated PCA ***with no sampling required***. Applying our results to previous models shows that the non-intuitive features are *not* an artifact of the learning process but rather are capturing robust properties of natural images.

The methods used in modelling natural images have limiting assumptions (Type of noise was known and the image shows natural scene). We could not use the method in our images because our images are special & the statistics of natural images do not apply. We have shown our results in the section 2.1.

2.1. Fourier Method

If we compute the Fourier Transform of natural images, then the power spectrum will have long tails. We tested it with our images but our power spectrum does not behave the same. This is clearly shown in the below figure 2.

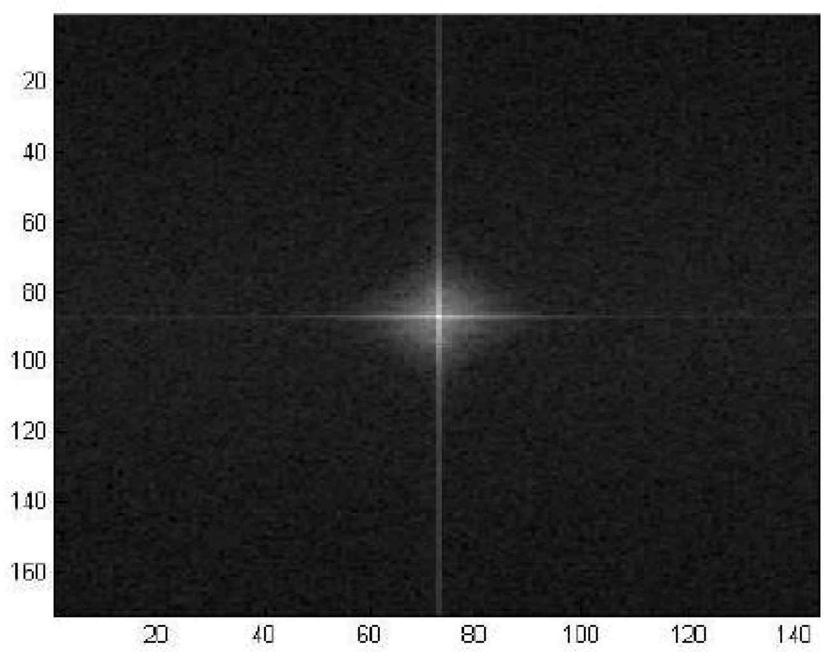


Figure 2: Power Spectrum density of Focused Ion Beam (FIB) images

As you can see, there is a region concentrated around the Fourier frequency zero that has non-zero values. But for natural images, most Fourier values will be non-zero in the entire frequency domain.

3. Attempts toward Image Quality Improvements

In this chapter, we give a brief overview of methods that we initially believed would be suitable, but did not give good results. This chapter motivates our final approach presented in chapter 4.

We tried experimenting the images initially with different techniques to see if we can clearly see the features. Our results were not as we expected it to be since we lost lot of important features/information in the images while trying to reduce noise. The experiments were divided into the following category

1) Noise Characterization
2) Noise Mitigation Analysis
3) Advanced Image Processing

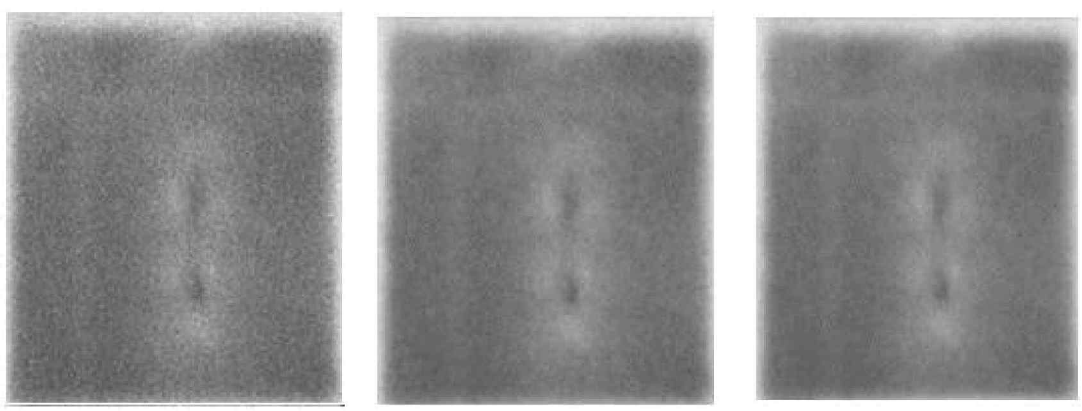
3.1. Noise Characterization

Our first step was to consider the type of noises within the image. We noted that this could be "Salt and pepper" type of noise but there was an additional noise that seemed to make well defined sides very jaggy. Temporally averaged images provide sufficient filled in pixels over time, so that the images are fully defined and are good candidates for determining the types of noise. To characterize the noise, [33] we measured "how noisy" via a signal to noise measurement and subjectively looked at the distribution of the noise and how it responded to various filters. The SNR for the images was found to be very low (SNR <= 5.0). We believed there was some salt and pepper noise since there were some dark specs in light regions and white specs in dark regions. Since it seemed more complicated noise than just Salt & Pepper noise, additional analysis was performed.

3.2. Noise Mitigation Analysis

3.2.1. Image Denoising

Once we have characterized the noise, a useful technique to measure mitigation was to perform denoising. Some of the simplest Noise filtering techniques that can be employed for initial denoising of images were the Gaussian & Median Filtering. The advantage of median filtering is that the value coming back is exactly one of the original pixel intensity values and the disadvantage is that it can offset a nice well defined edge as much as the half-width of the size of the kernel. It will work well in general for Salt and Pepper noise. [32]



(a) (b) (c)

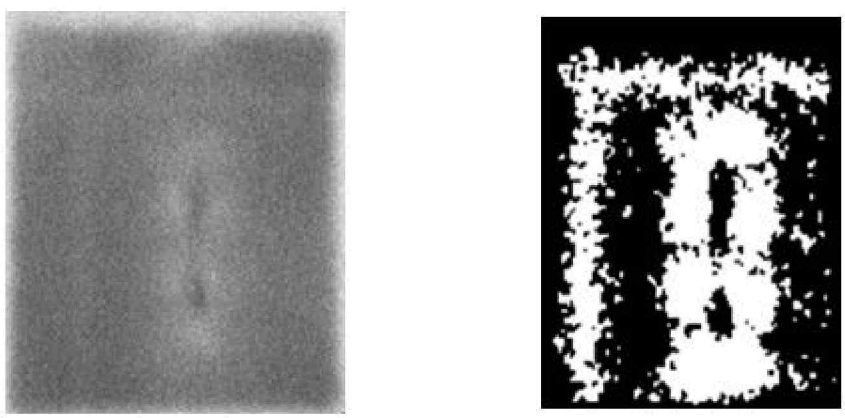
Figure 3: Image Denoising a) Original Image b) Gaussian 3X3 c) Gaussian 5X5

It is desirable to preserve shape so the smaller the kernel the better e.g. 3x3 is better than 5x5. Also, 5x5 is difficult to use on small images which is an undesirable feature.

3.2.2. Binary Image Filtering

Binary Image filtering [42] is one of the popular techniques used in feature extraction. There are several reasons that we opted for binary image processing. Some of the common reasons were smaller memory requirements and faster execution times.

The grey valued images were converted to binary images [29] to exaggerate the features to able to do further image analysis and also have the possibility of binary noise removal.



(a) **(b)**

Figure 4: Binary Image Filtering a) Original Image b) Gaussian 3x3

Notice, how clearly the shapes come out when viewed in the binary images. There are also some disadvantages associated with this type of filtering. The main disadvantage being the loss of the internal details within an image. This is very dangerous because we might lose costly information in an image which in our case might be some unknown features.

3.2.3. Edge Detection & Sharpening

To make it easier to see the exact edge for features, we applied edge detection and then made it brighter with sharpening. This is mainly for the benefit of human verification of features by eye. Some of the edge detection techniques that we used were Canny, Sobel, Prewitt, and Roberts. We found that canny edge detection seemed to do a better job of providing a clear edge. We have shown the results in Figure 5. [30, 31]

In order to tackle the problem, we had 2 different approaches

1) Do the Edge detection directly on the filtered image.
2) Do Sharpening on the filtered image and then apply edge detection on the sharpened image.

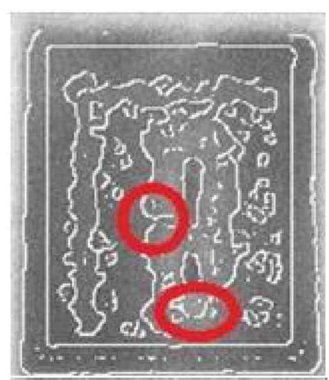
We believed 2 seems to be better than 1 because sharpening not only improved the contrast of the image but also removes extra jagged edges in the features of the image when applied with edge detection. This is indicated by the red circles. Since we were not sure if the features marked with red circles were actual features, we thought it would be ideal to both A and B and determine which was better than the other.

The impact of edge detection and sharpening techniques on the **Gaussian 3X3** filtered image is shown below in Figure 5. The red circles in the sharpened image have some faint features which we believed was closer to ground truth.

(a)

(b)

(c)



(d)

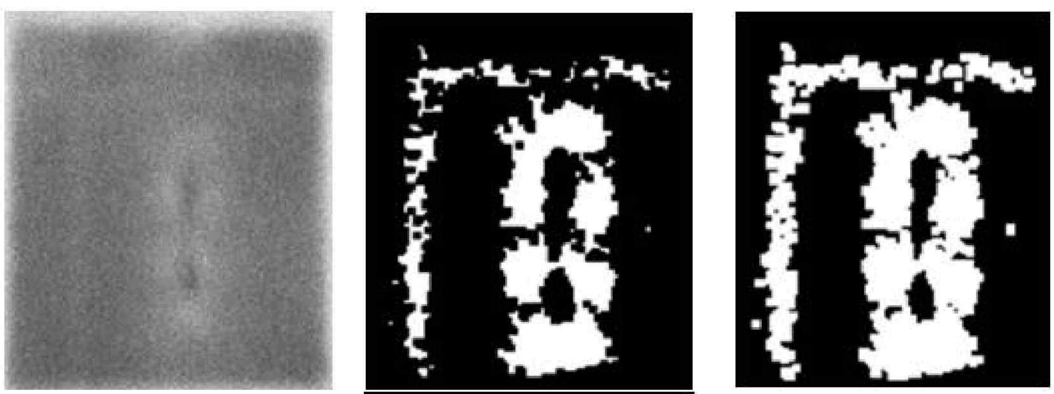
Figure 5: Image Enhancements a) Gaussian 3x3 b) Edge Detection without Sharpening c) Sharpened Image d) Edge detection with Sharpening

We do not show the results of Sobel, Prewitt and Roberts since we wanted to keep it simple in this thesis.

3.3. Advanced Image Processing

Even though we were able to see some kind of feature like appearances in the images after the edge detection & sharpening methods, we wanted to perform a higher or an advanced image processing technique such as morphology. Some of the most common morphological operations are the erosion and dilation of an image. [8, 39]

Morphology also removes significant features. Generally morphological operations are very useful in many image processing but for our problem, it could not do the trick since the noise was unknown and also the features were also unknown.



(a) **(b)** **(c)**

Figure 6: Morphology a) Original Image b) Erosion c) Dilation

4. Our Approach

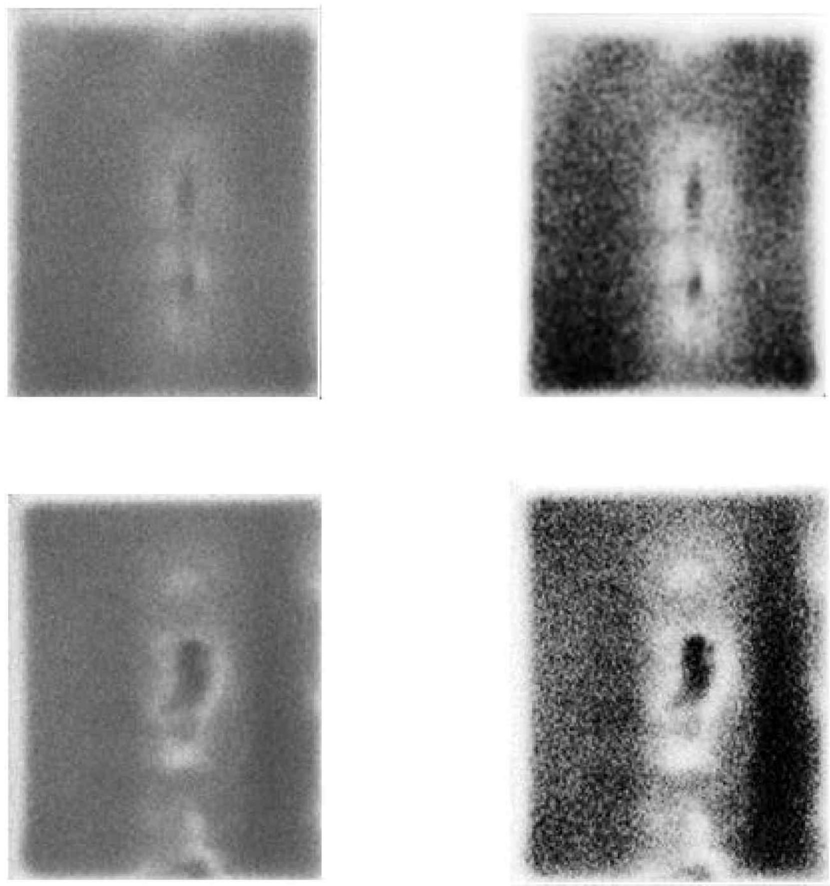
In this chapter, we present our new approach towards pattern discovery in Focused Ion Beam (FIB) images. Since the features in the images were unknown, we had to carefully consider techniques that overcame the disadvantages of our previous methods. We perform the initial preprocessing of images which is then followed by generating ensembles using geometric transformation methods. Once we obtain the stack of images, we adopt a method to calculate the mean of images. Finally, we implement a simple & a robust iterative least squares algorithm technique to better preserve the features in the images and also render visual feedback much easier.

4.1. Preprocessing

We have been given a set of images that are already temporally averaged over a span of time. They have been filtered using the most common filtering technique, median filter. We perform the preprocessing stage by first converting these temporally averaged images to histogram equalized images and we also apply suitable filters to better see the features in the images.

4.1.1. Histogram Equalization

Histogram Equalization is a very powerful technique that is almost always used as a first step in very noisy images. The technique is applied to our different patterns of images generated from the Focused Ion Beam (FIB) tool that are temporally averaged. It is also used in situations where we would want to reveal detail in an image that cannot easily be seen with the naked eye. [5, 40]



(a) **(b)**

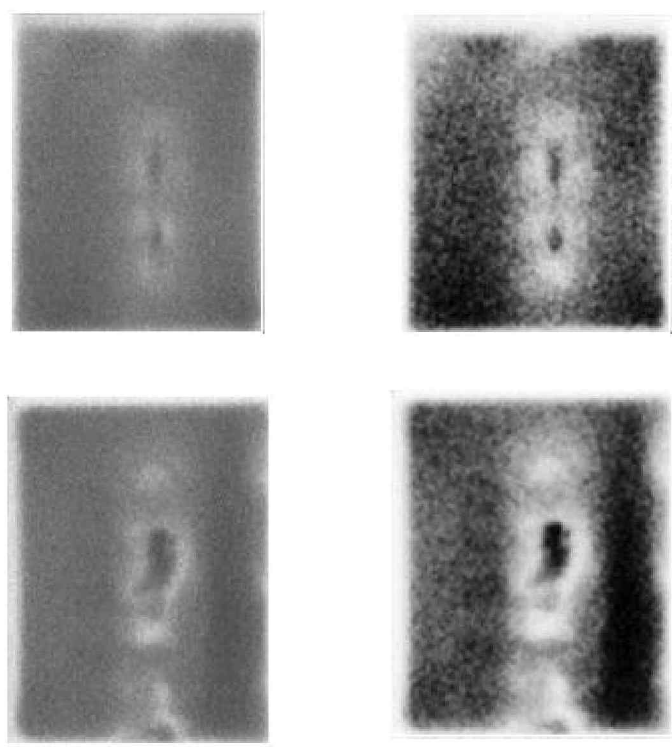
Figure 7: Preprocessing Stage a) Original Image b) Histogram Equalized

4.1.2. Image Filtering

The filtering techniques that were used on our images were average, disk, Gaussian, laplacian, log, median, motion, sobel, unsharp, & wiener filtering.

4.1.2.1. Mean Filtering

The method simply replaces each pixel value in an image with the mean (`average') value of its neighbors, including itself. [32, 34] It is based around a kernel, which represents the shape and size of the neighborhood. Often a 3×3 square kernel is used. 5x5 kernels are used for more severe smoothing.

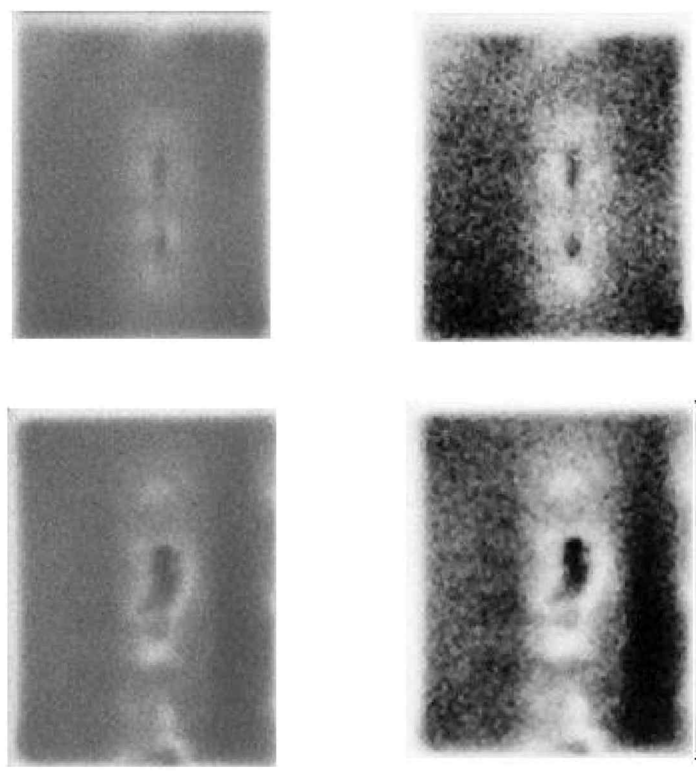


(a) **(b)**

Figure 8: Mean Filtering a) Original Image b) 3X3 Kernel

4.1.2.2. Median Filtering

This method is the state of the art means of filtering our images since we have found that our images contain Salt & Pepper noise. [32, 35] The median filter is much better at preserving sharp edges than the mean filter. A 3×3 kernel was used here.

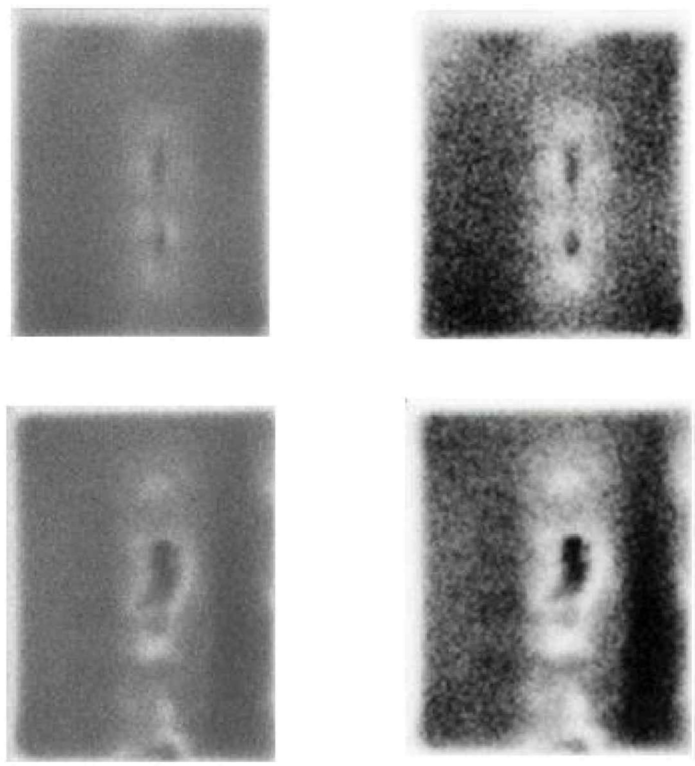


(a) (b)

Figure 9: Median Filtering a) Original Image b) 3X3 Kernel

4.1.2.3. Gaussian Filtering

This method uses a 2-D convolution operator that is used to `blur' images and remove detail and noise. The kernel is bell-shaped. [32, 36] The degree of smoothing is determined by the standard deviation of the Gaussian.

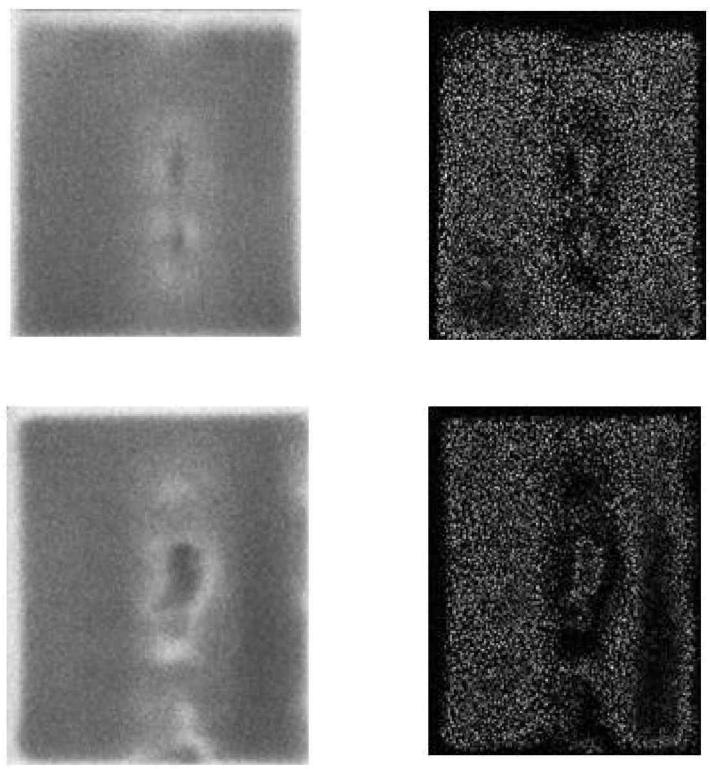


(a) (b)

Figure 10: Gaussian Filtering a) Original Image b) 3X3 Kernel

4.1.2.4. Laplacian Filtering

The Laplacian of an image highlights regions of rapid intensity change. [32, 37] The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise.

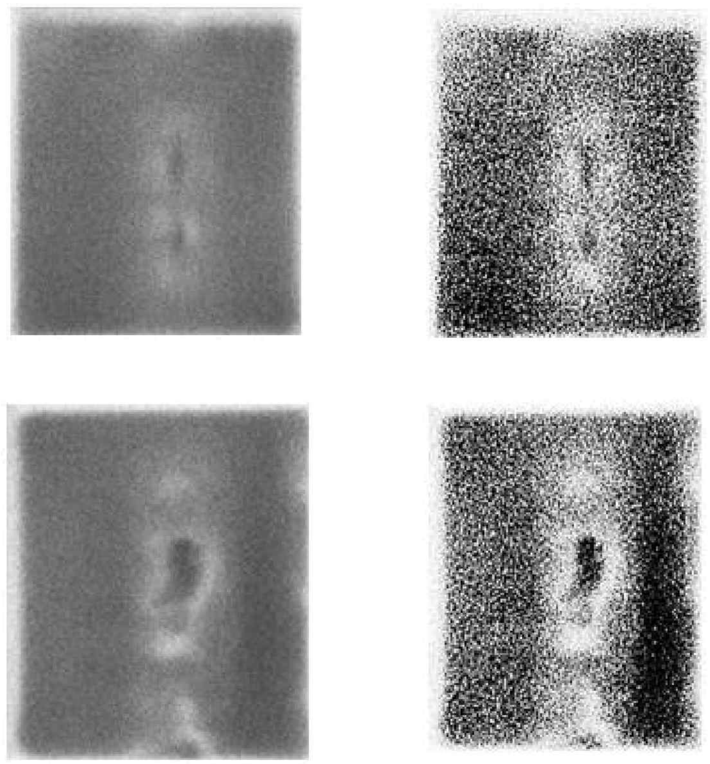


(a) (b)

Figure 11: Laplacian Filtering a) Original Image b) Filtered Output

4.1.2.5. Unsharp Filtering

The Unsharp filter is a simple sharpening operator which derives its name from the fact that it enhances edges in images. [32, 38]



(a) (b)

Figure 12: Unsharp Filtering a) Original Image b) Filtered Output

4.2. Geometric Transformation

To generate the ensembles, a geometric transformation was needed. We utilized the minimal angle rotation and resizing methods to perturbate the image. [7]

4.2.1. Image Rotation

We subjected our images to a minimal angle rotation *(-1,-0.5, 0.5, 1).* We do not show illustrations of these transformations, since they are mostly less visible to human vision and may depict the original image again. [41] [43]

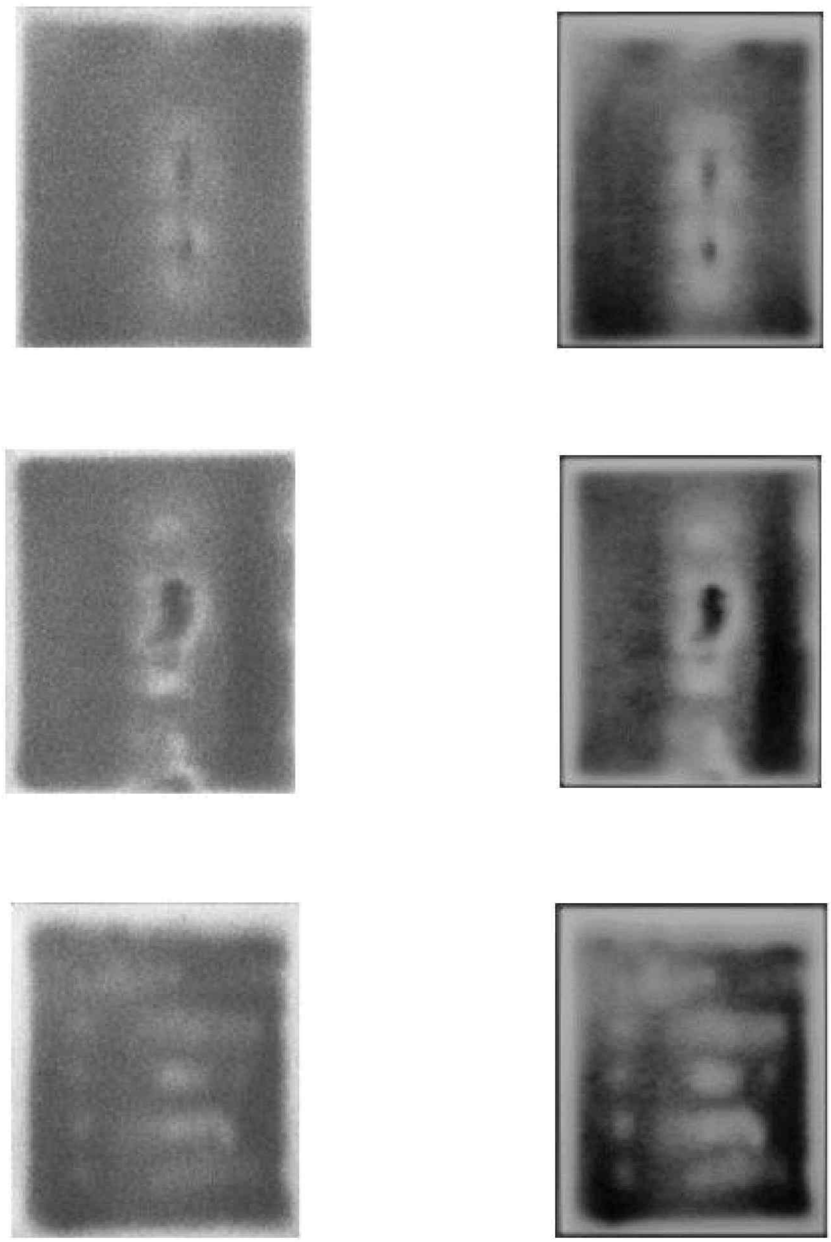
4.2.2. Image Resizing

Since the sizes of our images are 144x172 pixels wide, the resizing is done on different scales to induce perturbation in images. With image rotation & image resizing, we were able to obtain a stack of 1000 images as training images. [41, 43]

4.3. Mean

Finally once the images are rotated and rescaled, a stack of images are presented in a matrix and the mean of the image is calculated to see the distinctive features.
The following were the steps used to find the mean of the stack of images

1) Initialize the variable that is used to get the mean of the image to 0
2) Loop over all the images in the current directory
3) Begin the loop at this stage
4) Resize every image to maintain the same size for all the images
5) Add the variable to the existing stack obtained in Step 4
6) End the loop at this stage
7) Calculate the mean of images.



(a) (b)

Figure 13: Mean Method a) Original Image b) Resultant Average Output

4.4. Least Squares

The final step to the pattern discovery problem was in model fitting and computing the least squares. The problems concerning least squares are generally solved in 2 ways: linear least squares and non-linear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem has a closed-form solution (can be evaluated in a finite number of standard operations). The non-linear problem has no closed-form solution and is usually solved by iterative process. [1, 2, 3, 4]

The model we used in our problem was

$$f_{original} = f_i + \eta$$

We assume that a ground truth pattern $\hat{f}$ generates images f_i with additive noise η

To extract the ground truth image, we want to reduce the noise jointly in all images. Minimizing the function,

$$\sum_i | f - f_i|$$

The Mean of the image can be calculated as

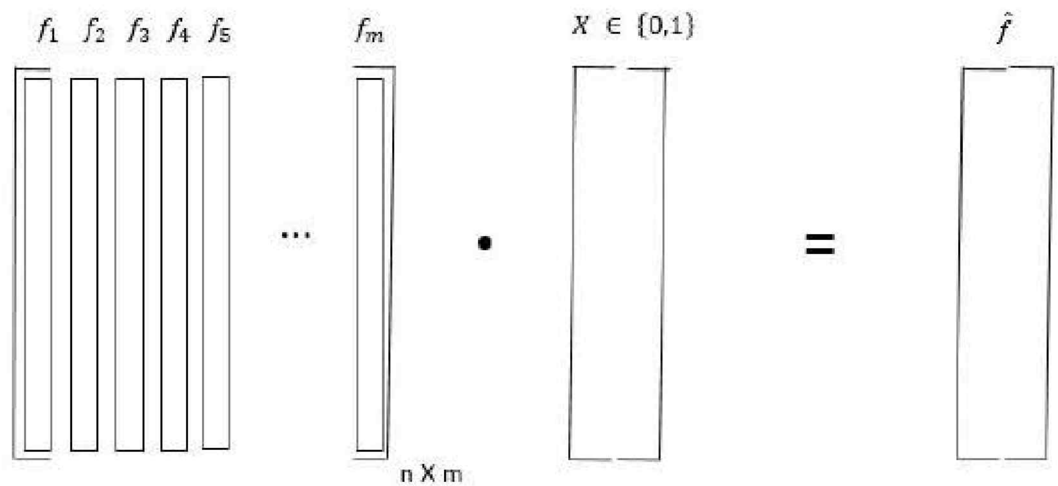
$$f = \frac{1}{N} \sum P_{ij}{}^{-1}(\hat{f}_i - \eta_{ij})$$

where N is the total number of variations and P_{ij}^{-1} is the perturbation operator.

Once the mean of the images is calculated, now we use 2 different approaches towards computing the Least squares method.

1) Simple Iterative Algorithm
2) Robust Iterative Algorithm

The illustration for calculating the least squares is given below.



where,
n = number of pixels
m = number of images

The stacks of the images are stored in a big matrix representing F.

Since, X is unknown here, we use the above 2 methods for calculating X in different setting.

3.4.1. Simple Iterative Algorithm

Solving the equation for the unknown coefficients once again requires writing the function that represents the sum of the squared errors. In this case it is a matrix computation, and can be represented as

$$\min_{X} \| f - (F * X) \|$$

$$L(x) = \sum \varepsilon^2 = \varepsilon^T \varepsilon = (f - Fx)^T (f - Fx)$$

Expansion of the function yields

$$L(x) = f^T f - x^T F^T f - f^T F x - x^T F^T F x$$

$$L(x) = f^T f - 2x^T F^T f - x^T F^T F x$$

since $x^T F^T f$ is a scalar value. Just as in the case of the simple linear model, calculus is used to find the minimum value of $L(x)$

$$\frac{dL}{dx} = -2F^T f + 2F^T F X$$

When solved, this becomes

$$F^{T}FX - F^{T}f$$

which gives us,

$$x = (F^{T}F)F^{-1}f^{T} \qquad ..(1)$$

Therefore, the least squares solution for any model that is linear in its unknown coefficients can be obtained with the above matrix computation.

The following are the steps by which the algorithm works on

Step 1: By the current approach with finding the mean, we obtain $\hat{f}$, which is the mean image

$$f^{(0)} = f^{(0)} = X^{(0)}$$

Step 2: Now our iterative model is given by

$$X^{(t)} = (F^{T}F)^{-1}F^{T}f^{(0)}$$

Step 3: Upon further iteration,

$$X^{(t+1)} = F * X^{(t-1)}$$

Step 4: Repeat the iteration till the error becomes 10^{-6}

$$\left|X^{(t+1)} - X^{(t)}\right|_2^2 < \mathcal{E} = 10^{-6}$$

3.4.2. Robust Iterative Algorithm

Here we just perturb the model with a constant value to search over all reasonable values and also check if this method helps in reducing noise further.

$$\min_{X} \|f - (F * X)\|$$

Constraint: ¥ k, X_k ε [0,1]

The sparse solution is given by $\|X\|^2$

In the robust iterative algorithm method, we introduce a scalar by means of relative weight in our equation (1)

$$\min_{X} \|\hat{f} - (F * X)\| + \lambda * (\|X\|_2^{\,2} - 1)$$

After solving the equation for X using the above sequence, we get

$$x = (F^T F + (\lambda * I))^{-1} F^T f$$

where,

λ = relative weight

I = Identity element

The following are the steps by which the algorithm works on

Step 1: By the current approach with finding the mean, we obtain $\hat{f}$, which is the mean image

$$f^{(0)} = f^{(0)} = X^{(0)}$$

Step 2: The loop begins now with varying values of "λ"

Our iterative model is given by

$$X^{(t)} = (F^T F + (\lambda * I))^{-1} F^T f^{(0)}$$

Step 3: Upon further iteration,

$$X^{(t+1)} = F * X^{(t-1)}$$

Step 4: End the loop

5. Results

In this work we try to discover significant features in newly produced silicon plates. There is no ground truth since even the human supervisors do not know what significant features should look like. Our main goal is to improve the quality of the images so that experts can later study significant features. Therefore our results can be only qualitative.

In the following, we present 5 types of experiments.

1) Simple Processing using Filtering
    1. Mean Filtering
    2. Median
    3. Median + Mean Filtering
2) Direct Iterative Least Squares without Processing
    1. No Median Filtering
    2. No Perturbation
3) Iterative Least Squares
    1. Mean Filtering
    2. Median Filtering
4) Simple Iterative Least Squares
    1. Filtering & Perturbation
5) Robust Iterative Least Squares
    1. Filtering & Perturbation

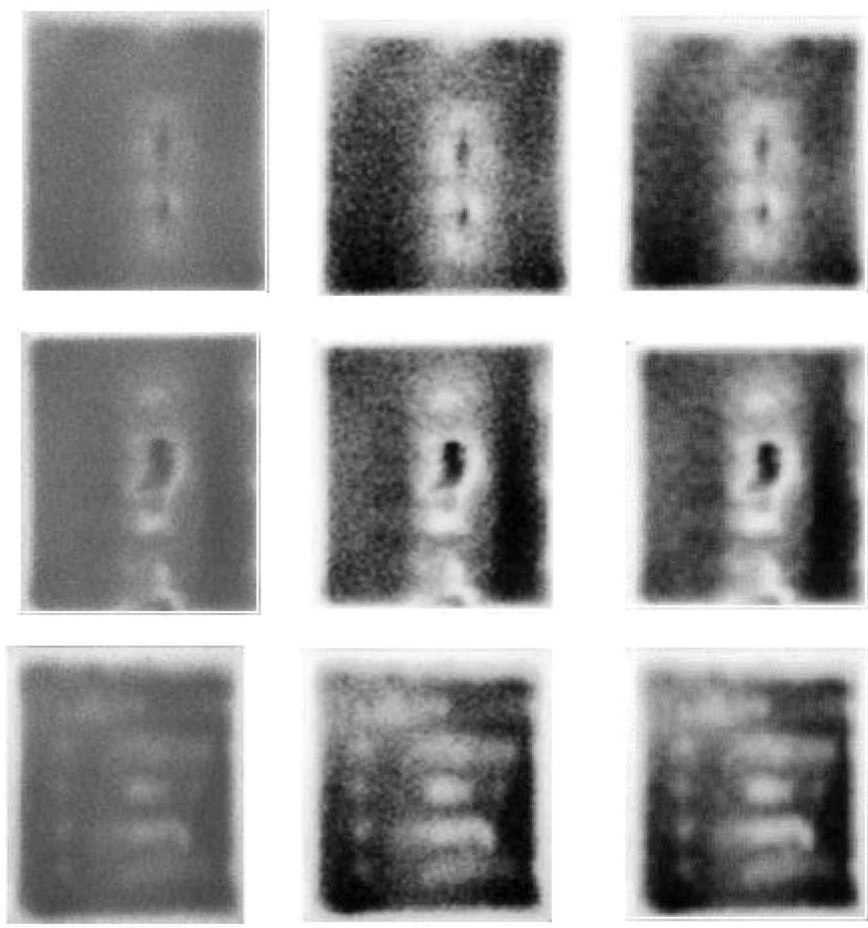
All of these steps test the importance of each step of our approach. The results at each step are shown clearly and we notice a significant improvement of clearly seeing the features.

5.1. Simple Processing using Filtering

The first step in our experiment was to identify the shape of features using basic filtering techniques such as the mean filtering and the median filtering.

5.1.1. Mean Filtering

We used 3 images with distinct features which are temporally averaged over a span of time *t* and then applied mean filtering on it.



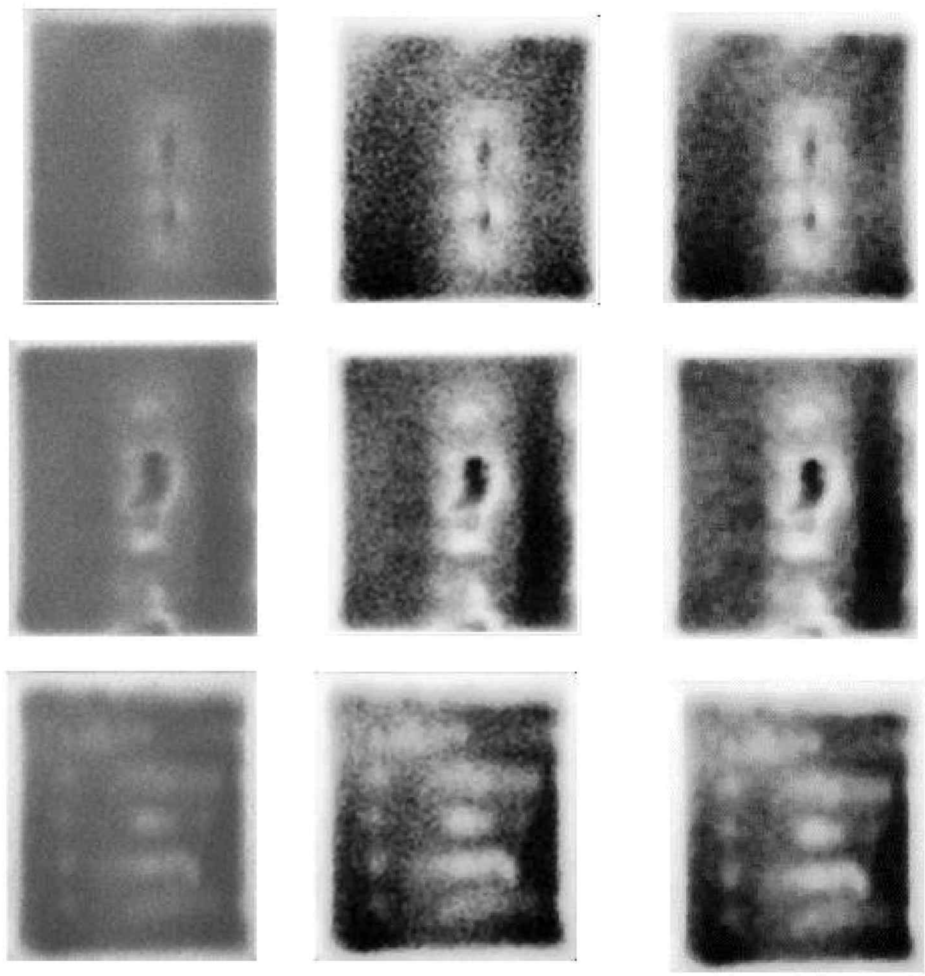
(a) (b) (c)

Figure 14: Mean Filtering a) Original Image b) 3 X 3 Filter c) 5 X 5 Filter

As can be seen in figure 4, mean filtering gives a blurry image for feature discovery. It seems like 5x5 distorts shape more than 3x3 and also difficult to use on small images which is an undesirable feature.

5.1.2. Median Filtering

We used 3 images with distinct features which are temporally averaged over a span of time ***t*** and then applied median filtering on it.



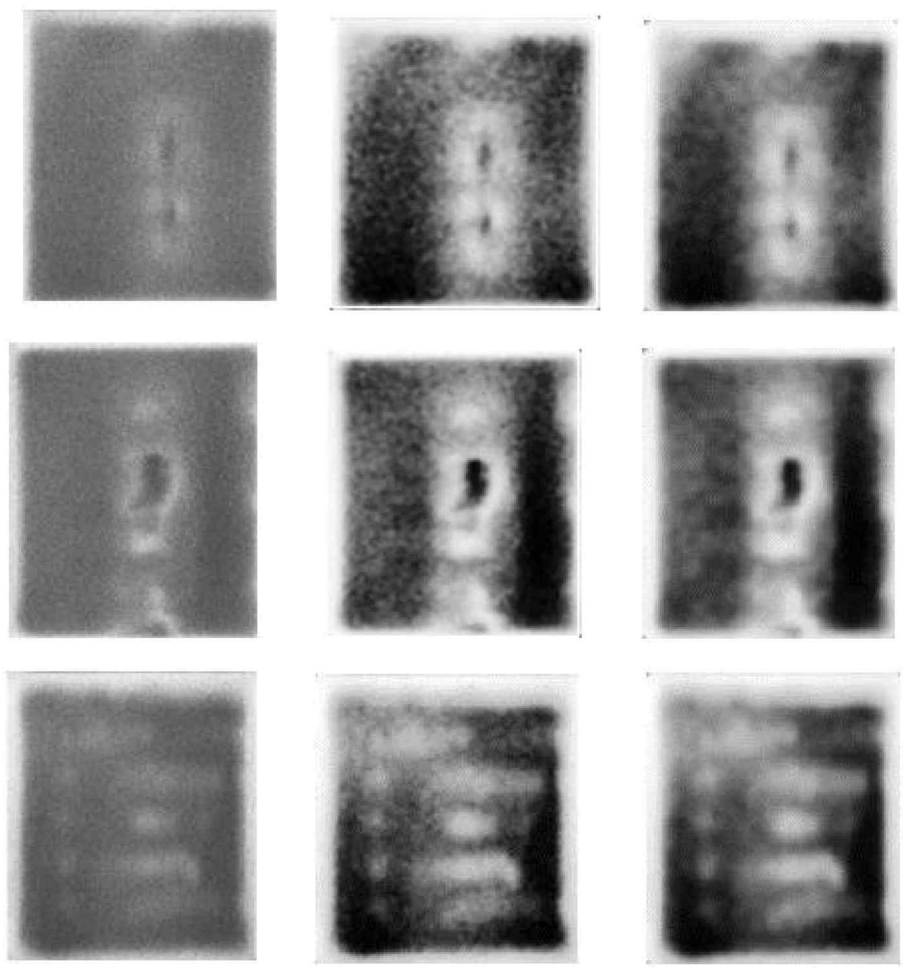
(a) (b) (c)

Figure 15: Median Filtering a) Original Image b) 3 X 3 Filter c) 5 X 5 Filter

The advantage of median filtering is that it works well in general for salt & pepper noise, the most predominant noise in our case. Since it is desirable to preserve shape, the smaller the kernel the better and that is why 3x3 is better than a 5x5.

5.1.3. Median + Mean Filtering

We used 3 images with distinct features which are temporally averaged over a span of time *t* and then applied median as well as mean filtering on it.



(a) (b) (c)

Figure 16: Median + Mean Filtering a) Original Image b) 3 X 3 c) 5 X 5

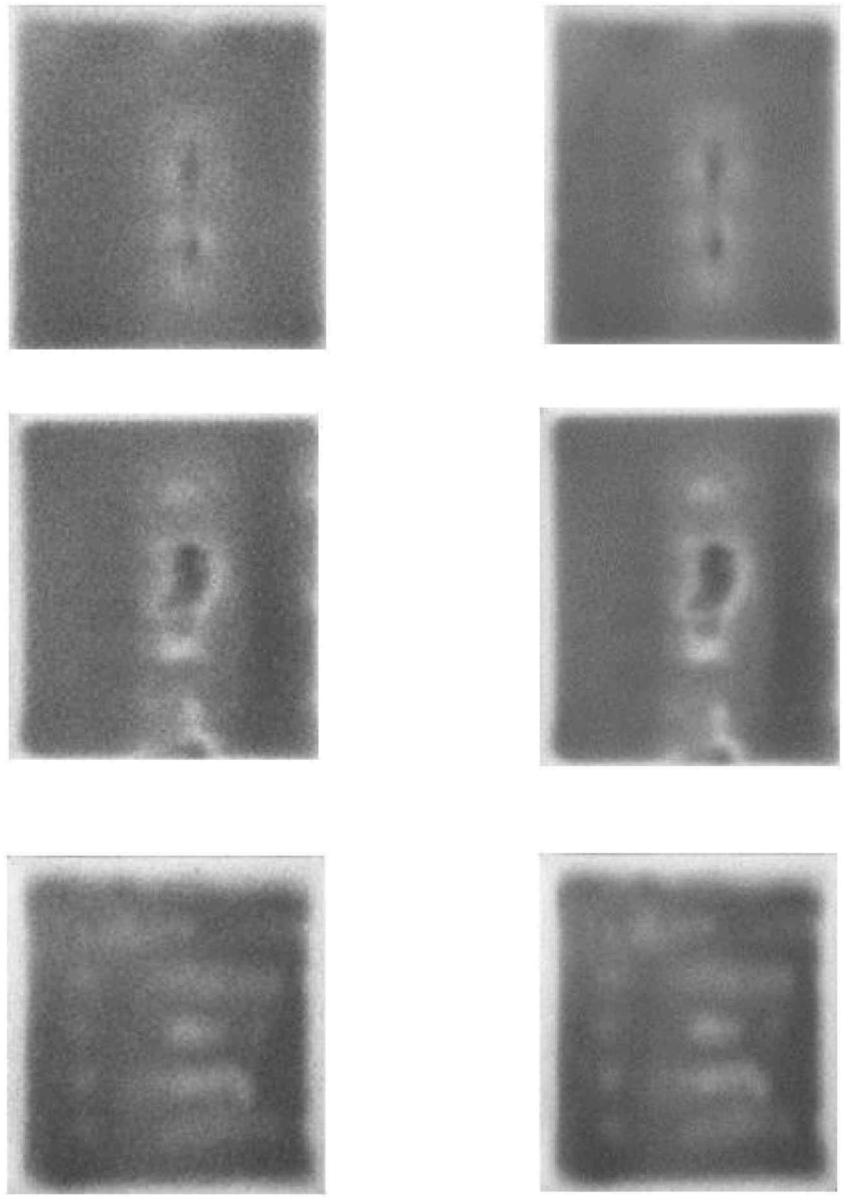
This seems the best combination among the simple filtering technique discussed so far. This is because of the fact the noise in our images is complex and consists of Additive Gaussian Noise and Salt & Pepper Noise.

5.2. Direct Iterative Least Squares

The final step to the pattern discovery problem was in model fitting and computing the least squares. The problems concerning least squares are generally solved in 2 ways: linear least squares and non-linear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem has a closed-form solution (can be evaluated in a finite number of standard operations). The non-linear problem has no closed-form solution and is usually solved by iterative process. [1, 2, 3]

5.2.1. Without preprocessing

We used 3 images with distinct features which are temporally averaged over a span of time ***t***. The least squares model was used on the images without any filtering or perturbation. This was our first step in determining if the method of least squares would be useful for our kind of images where the noise level is unknown and also the patterns in the images are unknown. Since we wanted to ensure that there isn't too much blurring, we did not apply any filters and just utilized the least squares model on the images.

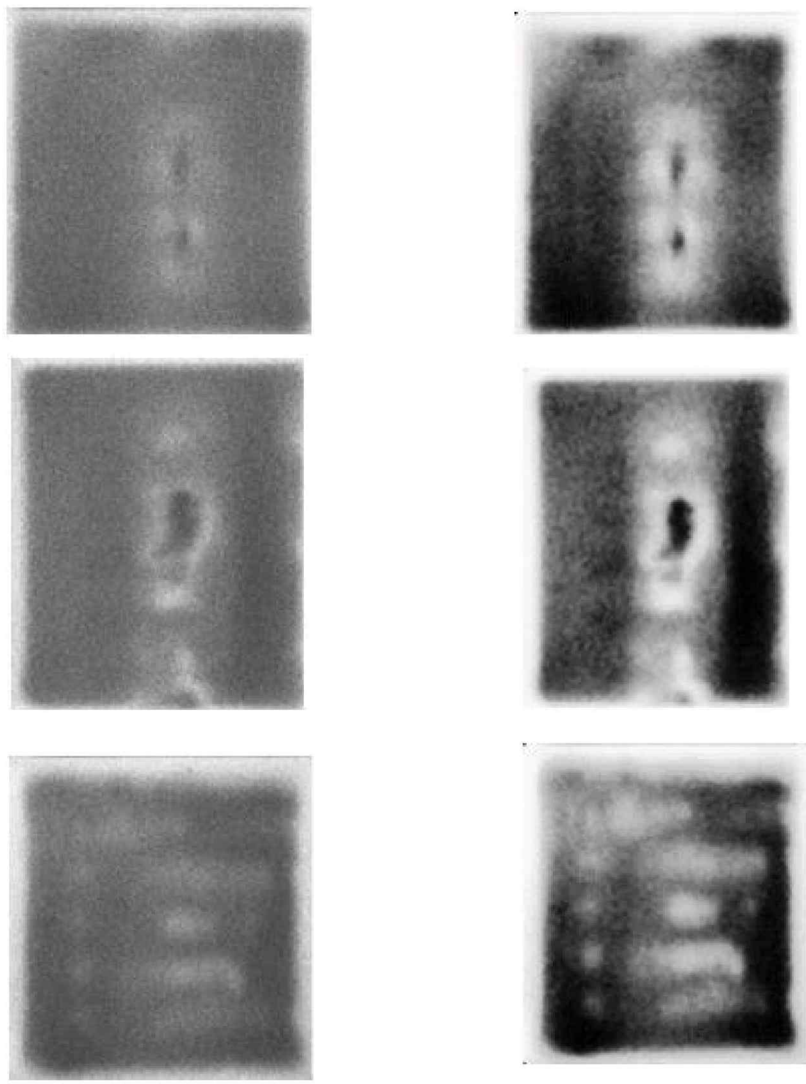


(a) (b)

Figure 17: Three Distinct Features a) Original Image b) Direct Least Squares without Preprocessing

5.2.2. With Median Filtering

We used 3 images with distinct features which are temporally averaged over a span of time *t*. We applied a median filter initially and, the least squares method was used to see if the features are visible to the human eye.

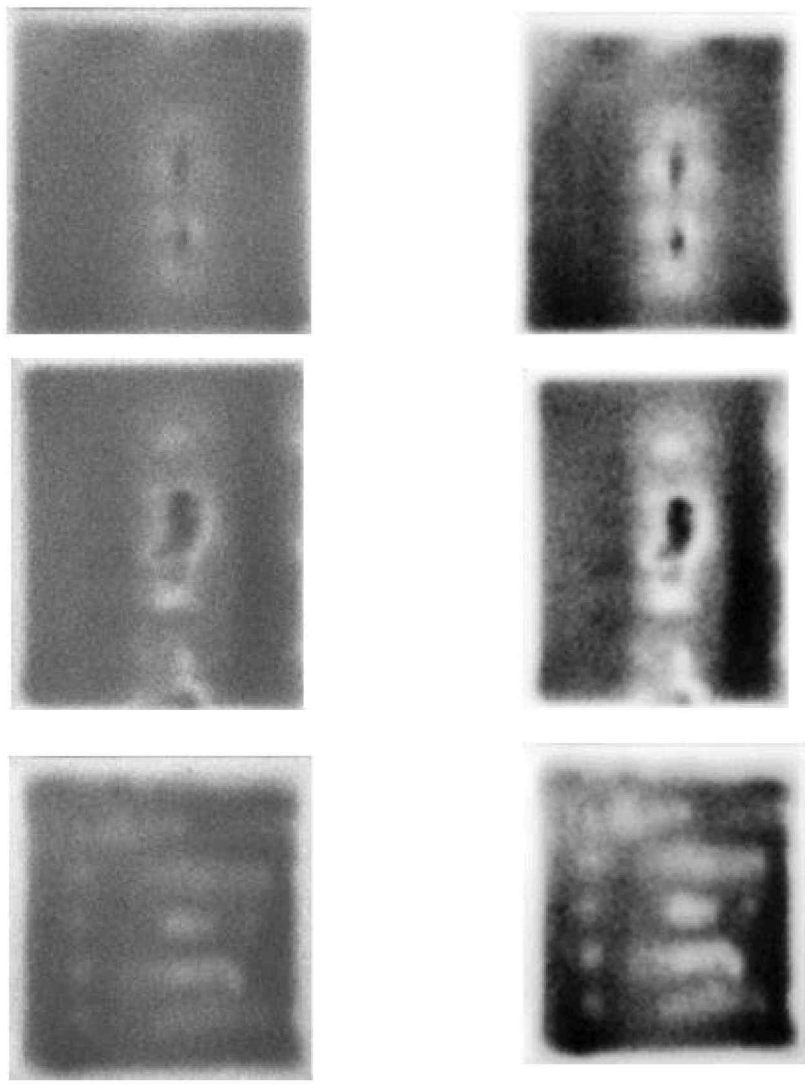


(a) **(b)**

Figure 18: Three Distinct Features a) Original Image b) Direct Least Squares with Median Filtering

5.2.3. With Mean Filtering

We used 3 images with distinct features which are temporally averaged over a span of time *t*. We applied a mean filter initially and, the least squares method was used to see if the features are visible to the human eye.



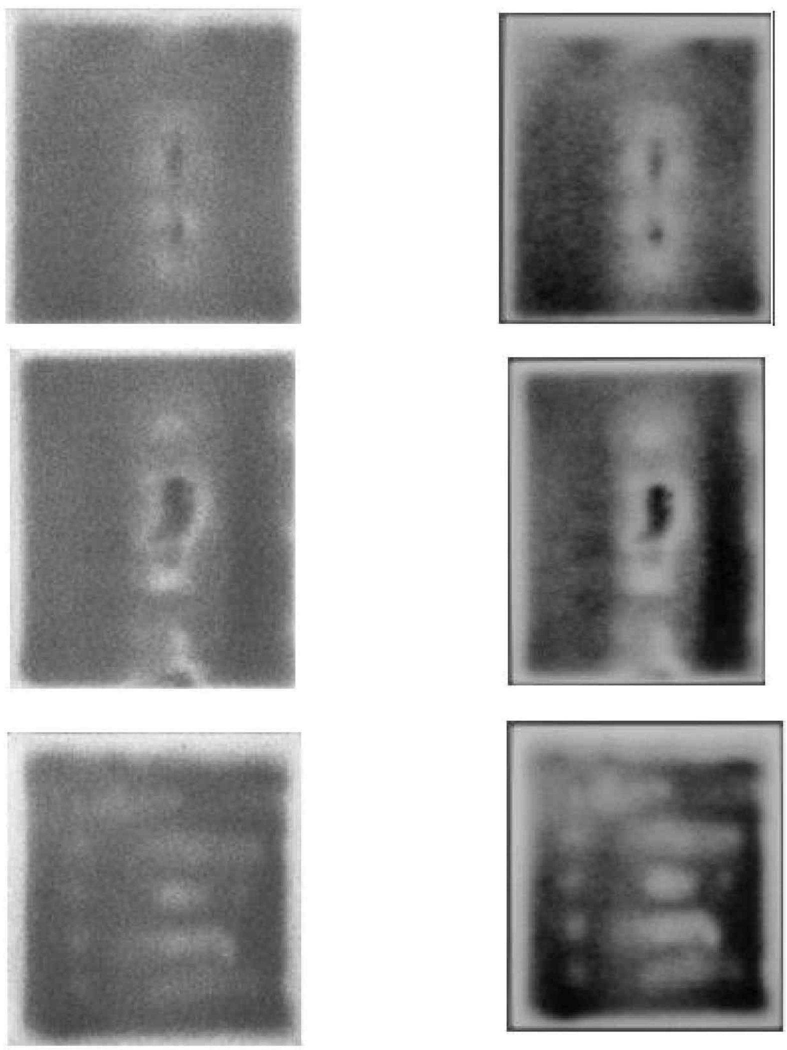
(a) (b)

Figure 19: Three Distinct Features a) Original Image b) Direct Least Squares with Mean Filtering

5.3. Simple Iterative Least Squares Algorithm

5.3.1. Filtering & Perturbation

We used 3 images with distinct features which are temporally averaged over a span of time ***t*.**



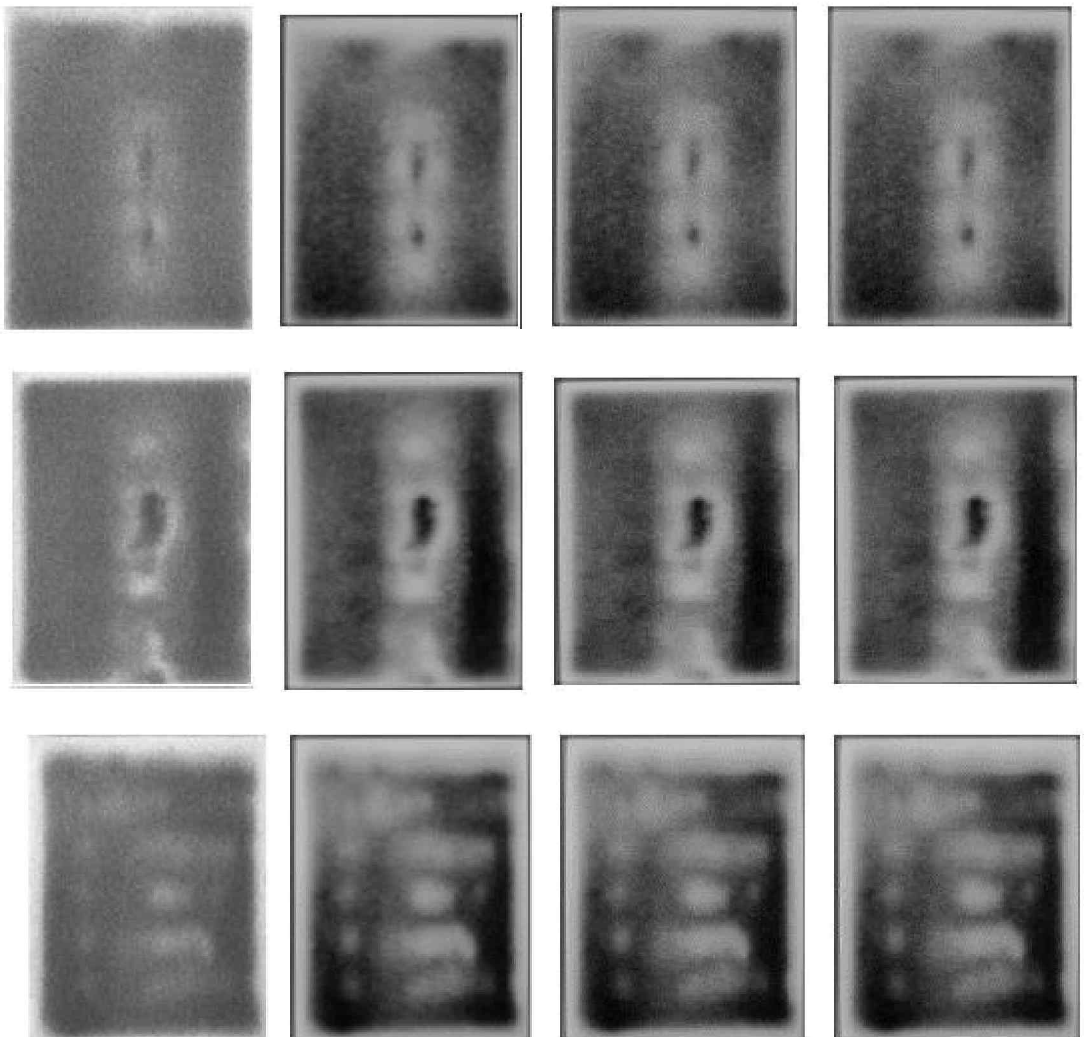
(a) (b)

Figure 20: Three Distinct Features a) Original Image b) Simple Iterative Least Squares

5.4. Robust Iterative Least Squares Algorithm

5.4.1. Filtering & Perturbation

We used 3 images with distinct features which are temporally averaged over a span of time ***t***.



(a) (b) (c) (d)

Figure 21: Three Distinct Features a) Original Image b) Robust Iterative Least Squares with λ=0.7 c) λ=1.8 d) λ=2.7

In comparison with the previous experiments that are done without perturbations, we see that the simple & robust iterative least squares algorithms perform better and the features seem to be retained in the images even with more noise suppression. It seems the FB technique introduces not only additive noise but also some geometric transformations in terms of rotation and translation and resizing. From the above figures, both the simple and robust iterative algorithms behave the same. For simplicity, we recommend the chip inspectors to use the simpler methods.

Feedback form the employer:

This section serves as a human study. This is necessary, since we do not have the quantitative evaluation. This survey consisted of a group of 10 experts from Intel who verified the results. They were unanimous in their judgments and were very happy with the results produced. It helped them discover important features.

Average Responses for my survey

1) Is the quality good on a scale of 1-4?
   - 9/10 people said 4, and one person said 3.
2) What features of interest have you noticed?
   - There was 100% agreement in the particular discovered pattern. Those were namely 8-Shaped, E-Shaped, Y-Shaped patterns.
3) Are the most important notable features preserved on a scale of 1-4?
   - 8/10 people gave 4 and 2 people gave 3.
4) Is there a significant difference in the resulting quality of the images relative to those previously used?
   - There was a 100% agreement that we achieved improvement.
5) Is our approach showing the features accurately corresponding to the CAD layout?
   - There was a 100% agreement that it shows accurately to the CAD layout.

6. Conclusion

Discovery of important patterns in images, which in turn are corrupted by unknown noise is a challenging problem. This problem arises in many applications, including circuit editing using FIB.

We have performed different experiments with the FIB images

- Initially, we tested our images with the mean and the median filtering methods. Since we predicted that some component of the noise contains the Salt & Pepper noise, medina filter was better. But later we combined both the filtering techniques and applied on our images. This seemed to do a better job than the previous 2 techniques.
- Once the simple filtering was done, we implemented a least squares models on the images without any filtering or perturbation. We got better results than the first which just involved filtering.
- After applying a filter mechanism and employing a least squares model on the images, we were able to suppress the noise amount way better than the previous.
- Our final approach in dealing with noisy patterns was the simple iterative & the robust least squares algorithm. Prior to applying our model, our images were subjected to filtering and geometric. At this stage in the process, the level of noise significantly reduced and also preserved very important detail or features in the images were retained. It also looked like both the simple and the robust iterative algorithm strategies showed the same result.

We also performed a user study which included 10 people answering our survey questions. The responses were positive and encouraging.

Response:

Around 90% agreed that our quality was good. Around 80% agreed that important notable features were preserved and not lost in the analysis. There was a 100% agreement in various discovered patterns such as the 8-Shaped, E-Shaped, and Y-Shaped. There was also a 100% agreement that the features were corresponding to the CAD layout and also achieved improvement.

Appreciation from the Employer:

1) "The algorithms for each of your approach will be added to the tool for further processing of images generated from the FIB tool." – Software & Services group at Intel
2) I got the Intel recognition award for my work. – Software & Services Group at Intel.

The job is done and the software is adopted. The work was a success based on the user feedback. It had a great impact on the employer.

7. References

[1] Wikipedia on Least Squares, http://en.wikipedia.org/wiki/Least_squares.

[2] Å. Björck (1996). *Numerical Methods for Least Squares Problems*. SIAM. ISBN 978-0-89871-360-2.

[3] C.R. Rao, H. Toutenburg, A. Fieger, C. Heumann, T. Nittner and S. Scheid (1999). *Linear Models: Least Squares and Alternatives*. Springer Series in Statistics.

[4] T. Kariya and H. Kurata (2004). *Generalized Least Squares*. Wiley.

[5] *Histogram Equalization*, http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm.

[6] Image Rotation, http://homepages.inf.ed.ac.uk/rbf/HIPR2/rotate.htm.

[7] Affine Transformation, http://homepages.inf.ed.ac.uk/rbf/HIPR2/affine.htm.

[8] Morphology Analysis, http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm.

[9] Digital Filtering, http://homepages.inf.ed.ac.uk/rbf/HIPR2/filtops.htm

[10] M. J. Black and P. Anandan. The robust estimation of multiple motions: affine and piecewise smooth fields. Technical Report spl-93-092, Xerox PARC, 1993.

[11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proc. IEEE Conf. Computer. Vision Pattern Recog.* 1999.

[12] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of IEEE CVPR*, pages 261.268, 2004.

[13] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787.794, 2006.

[14] G. E. Hinton and Y. W. Teh. Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of Uncertainty in Artificial Intelligence (UAI-2001)*, 2001.

[15] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.* 7(1.3):185.203, August 1981.

[16] J. Huang, A. B. Lee, and D. Mumford. Statistics of range images. In *CVPR*, pages 1324.1331, 2000

[17] Levin and Y.Weiss. User assisted separation of reflections from a single image using a sparsity prior. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2004.

[18] R. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

[19] B. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607.608, 1996.

[20] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans Image Processing*, 12(11):1338.1351, 2003.

[21] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005

[22] S. Roth and M. J. Black. On the spatial statistics of optical flow. In *ICCV*, pages 42.49, 2005.

[23] E. Simoncelli. Statistical models for images: compression restoration and synthesis. In *Proc Asilomar Conference on Signals, Systems and Computers*, pages 673.678, 1997

[24] C. K. I. Williams and F. V. Agakov. Products of gaussians and probabilistic minor component analysis. *Neural Computation*, 14(5):1169.1182, 2002

[25] S. Zhu and X. Liu. Learning in gibbsian fields: How fast and how accurate can it be? *IEEE Trans on PAMI*, 2002.

[26] S. C. Zhu, X. W. Liu, and Y. N. Wu. Exploring texture ensembles by efficient Markov Chain Monte Carlo. Toward a trichromacy theory of texture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):554. 569, 2000.

[27] S. C. Zhu and D. Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, (11):1236.1250, 1997.

[28] Yongkyoo Choi*, Heecheon Kim , Oscar Han: "Image enhancement technology to get fine defect image for FIB".

[29] Kuntal Gosh, Anirban Roy: "Neuro-visually inspired figure-ground segregation", IEEE trans. ICIP, 2011.

[30] Image filtering & Edge Detection: http://web.media.mit.edu/~maov/classes/vision09/lect/09_Image_Filtering_Edge_Detection_09.pdf

[31] Contrast Enhancement Techniques: http://www.mathworks.com/products/image/examples.html?file=/products/demos/shipping/images/ipexcontrast.html

[32] Filtering: http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter4.pdf

[33] Noise Characterization: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/noise.htm"

[34] Mean Filtering: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm"

[35] Median Filtering: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm"

[36] Gaussian Filtering: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm"

[37] Laplacian Filter: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm"

[38] Unsharp Filter: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm"

[39] Morphology: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm"

[40] Histogram Eq: "http://homepages.inf.ed.ac.uk/rbf/HIPR2/histeq.htm"

[41] Geometric Operators: http://homepages.inf.ed.ac.uk/rbf/HIPR2/geomops.htm

[42] Binary Image Processing: "http://www-ee.uta.edu/Online/Devarajan/ee6358/BinaryImage.pdf".

[43] George Walberg: "Geometric Transformation Techniques for Digital Images: A Survey", Technical Report CUCS-390-88, December 1988.

[44] Yair Weiss, William T.Freeman: "What makes a good model of Natural Images?", IEEE trans. CVPR, Computer Vision and Pattern Recognition, 1063-6919, 2007.