

TV vs H1 Inpainting

Jake Bergquist, u6010393

March 19, 2019

Introduction

For this midterm project we were tasked with formulating and implementing H1 and total variation in painting methods to replace corrupted data from images. In the case of the H1 inpainting we do so by implementing a H1 denoising algorithm which, in short, performs an iterative laplacian blur into the missing area. If we consider the H1 denoising formulation (Equation 1) we see that this function aims to estimate the denoised image, $\hat{I}(x)$ by minimizing the difference between the sampled image ($d(x)$) and the denoised image ($I(x)$) as well as the gradient of the denoised image. By integrating these terms we find that we can simplify to Equation 2 wherein our minimization changes to minimizing the sum of the inner products of $d(x) - I(x)$ with itself plus the inner product of the negative laplacian of $I(x)$ with $I(x)$. From here we get Equation 3 which allows us to solve for the ideal $I(x)$ by setting it equal to zero. By using a finite difference model we can approximate the inverse of the laplacian needed to calculate for this ideal $I(x)$. Now if we consider our inpainting problem instead we also want to impose the condition that we have data missing. The same denoising algorithm can be used to fill in this missing space. The entire H1 denoising algorithm boils down to applying a laplacian blur to the image. If we apply this blur into the areas where data is missing (set the corrupt data to 0) this will begin to interpolate the missing data based on the neighboring data. We can iteratively blur into the missing region while replacing the non missing data with the original measured data. This prevents the image from being overly smoothed. Finally we can perform a final H1 denoise to smooth any noise if desired.

$$\hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int ||d(x) - I(x)||^2 dx + \frac{1}{2\alpha^2} \int ||\nabla I(x)||^2 dx \quad (1)$$

$$\hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \langle d(x) - I(x), d(x) - I(x) \rangle + \frac{1}{2\alpha^2} \langle -\Delta I(x), I(x) \rangle \quad (2)$$

$$-\Delta I(x) + \frac{(I(x) - d(x))}{\sigma^2} = 0 \quad (3)$$

In the case of TV denoising and inpainting we must consider the slightly different primary formulation Equation 4. Notice that the second term is no longer the L2 norm but the L1 norm. Thus we can no longer solve this minimization as simply as we cannot operate in Hilbert space. Thus we must use a dual space formulation for this optimization. Because of the relationship outlined in equation 5 we can rewrite Equation 4 as Equation 6. By differentiating this equation and setting equal to zero we get the dual formulation of our problem in Equation 7. By substituting

7 into 4 and completing the square by adding a constant term $d^2(x)dx$ we get a more simple primal formulation in Equation 8. All of this defines TV denoising, which allows for the preservation of discontinuous edges via the L1 term in the primal form which is not possible in the smoothing done by the L2 term of the H1 denoising. To optimize the TV problem we can either take steps in the primal and dual directions by optimizing over 4 and 7 or we can optimize over the full dual using 8. However, to modify this formulation to account for an image with missing data we have two options. In both we impose a $m(x)$ mask term that is zero where we are missing data and 1 where we are not missing data. By multiplying this mask pointwise across the first term in 4 we account for the fact that there is no contributing data at those locations. However this changes our formulation of 8 significantly. Because we now have zeros in $d(x)$ we can no longer formulate 8 as this would result in division by zero. We can either perform this optimization of inpainting by using the primal dual formulation (4 and 7) (again like in H1, using the interpolation into the missing region, then at each iteration replacing the area outside of the missing data area with the original measured image) or we can attempt to formulate the full dual problem like in 8. In this report I will present results from the former of these, but I will present a formulation for the second. Due to some error along the way I was unable to get the full dual formulation to function.

$$\hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int ||d(x) - I(x)||^2 dx + \frac{1}{\alpha^2} \int ||\nabla I(x)|| dx \quad (4)$$

$$||\nabla I(x)|| = \sqrt{I_x^2(x) + I_y^2(x)} = \operatorname{argmax}_{||w(x)|| < 1} -I(x) \operatorname{div}(w(x)) \quad (5)$$

$$\hat{I}(x) = \operatorname{argmax}_{||w(x)|| < 1} \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int ||d(x) - I(x)||^2 dx - \frac{1}{2\alpha^2} \int I(x) \operatorname{div}(w(x)) dx \quad (6)$$

$$\hat{I}(x) = \sigma^2 \operatorname{div}(w(x)) + \alpha^2 d(x) \quad (7)$$

$$\hat{I}(x) = \operatorname{argmax}_{||w(x)|| < 1} - \int \alpha^2 \operatorname{div}(w(x)) + \frac{d(x)}{\sigma^2} \quad (8)$$

To formulate the full dual TV inpainting we must add a term to the primal problem that fixes our division by zero problem. This is seen in Equation 9 where we add a ϕ weighted term that adds a constant c which prevents a division by zero error. Taking the derivative of 9, setting it to zero, and a bit of rearrangement gives us 10. By defining 11 and 12 we can simplify to 13 as our dual formula. Substituting 13 into 9 we get a very long equation 14. After using the fact that $m(x)^2 = m(x)$, defining ?? and 16 we get 17. The goal at this point is to group terms that are linear in $w(x)$, and those that are quadratic in $w(x)$. By expanding 17 and grouping these terms we get 18 with $F(X), G(X), H(x)$ defined as shown. With this we can define our full dual step as shown in 22. In practice this produces an all zeros or all ones image. Thus there is likely an error involving the multiplication of the mask.

$$\hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int m(x) ||d(x) - I(x)||^2 dx + \alpha \int ||\nabla I(x)|| dx + \frac{1}{2\phi^2} \int ||I(x) - C||^2 dx \quad (9)$$

$$\hat{I}(x) = \frac{\phi^2 m(x) d(x) + \sigma^2 \phi^2 \alpha \operatorname{div}(w(x)) + \sigma^2 C}{\phi^2 m(x) + \sigma^2} \quad (10)$$

$$A(x) = \frac{\phi^2 m(x) d(x) + \sigma^2 C}{\phi^2 m(x) + \sigma^2} \quad (11)$$

$$B(x) = \frac{\sigma^2 \phi^2 \alpha}{\phi^2 m(x) + \sigma^2} \quad (12)$$

$$\hat{I}(x) = A(x) + B(x) \operatorname{div}(w(x)) \quad (13)$$

$$\begin{aligned} \hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int & \|m(x)A(x) + m(x)B(x) \operatorname{div}(w(x)) - m(x)d(x)\|^2 dx, \\ & -\alpha \int A(x) \operatorname{div}(w(x)) + B(x) \operatorname{div}^2(w(x)) dx, \\ & + \frac{1}{2\phi^2} \int \|A(X) + B(x) \operatorname{div}(w(x)) - C\|^2 dx \end{aligned} \quad (14)$$

$$C(x) = m(x)A(x) - m(X)d(x) \quad (15)$$

$$E(x) = A(x) - C \quad (16)$$

$$\begin{aligned} \hat{I}(x) = \operatorname{argmin}_{I(x)} \frac{1}{2\sigma^2} \int & \|C(x) + m(x)B(x) \operatorname{div}(w(x))\|^2 dx, \\ & -\alpha \int A(x) \operatorname{div}(w(x)) + B(x) \operatorname{div}^2(w(x)) dx, \\ & + \frac{1}{2\phi^2} \int \|E(X) + B(x) \operatorname{div}(w(x))\|^2 dx \end{aligned} \quad (17)$$

$$\hat{I}(x) = \operatorname{argmin}_{I(x)} \int F(x) \operatorname{div}^2(w(x)) + \int G(x) \operatorname{div}(w(x)) + H(x) dx \quad (18)$$

$$F(x) = \frac{m(x)B(x)}{2 * \sigma^2} - \alpha B(x) + \frac{B(x)^2}{2\phi^2} \quad (19)$$

$$G(x) = \frac{C(x)m(x)B(x)}{\sigma^2} - \alpha A(x) + \frac{E(x)B(x)}{\phi^2} \quad (20)$$

$$H(x) = \frac{C(x)^2}{2\sigma^2} + \frac{E(x)^2}{2\phi^2} \quad (21)$$

$$w(x)^{k+1} = w(x)^k + \epsilon(2\nabla(F(x) \operatorname{div}(w(x)^k)) + \nabla G(x)) \quad (22)$$

On the other hand using the primal dual formulation is very easy. As stated before, the locations where data is missing are set to zero and the TV denoising is run. After many iterations the TV denoising will start to interpolate into the missing region. At each iteration the region outside of the missing region is returned to its starting state to prevent over filtering of the image. As I explored different alpha and sigma values I found that a sigma of 0.25 and alpha of around 500 produced good results with a small step size (0.01 for primal and dual) and a large number of iterations (1000). However I found those alpha and sigma values were not ideal for other kinds of inpainting scenarios which are shown in results. To this end I thought it would be

nice to try to optimize the selection of alpha and sigma. Given that we have the uncorrupted image (a not realistic scenario) we could easily come up with a cost function to minimize within the missing data mask area as seen in 23. A few days before this assignment was due I was taught by a collaborator about how a simplex optimization algorithm works. By first taking three "random" samples of the parameter space (given that we have two parameters in this case) we can create a triangle of sampled locations. From this we can determine which of these has the maximum energy according to our Eq 23. We can then formulate our new guess to be away from this high energy guess towards the two lower energy guesses. The two lower energy guesses and our new guess form a new triangle and the process repeats, slowly walking towards an energy minimum. This approach does potentially get stuck in local minimum but it has the advantage of not needing to calculate a gradient directly and only needs to sample as it goes. By defining the first triangle as an equilateral triangle in the space with sides equal to some step size ϵ , and the new guess step such that it would make a new equilateral triangle we can easily step through the space. By slowly decreasing the new step distance we can allow the algorithm to more precisely zoom in on the minimum as it goes. Of course there are more elegant ways to implement this, but I was working with a limited time span since I learned about this (about a day and a half). I was only able to implement this for TV, but it could easily be done for H1 as well to optimize for those parameters. Additionally one could imagine how such an algorithm could be implemented with a different energy function that did not require the actual uncorrupted image but rather used some other metric of improvement.

$$\operatorname{argmin}_{\sigma, \alpha} \sum |1 - m(x)|I(x) - |1 - m(x)|d(x) \quad (23)$$

To test these algorithms and methods, the original as well as three other corrupted images were used. Corruption two was done by removing evenly spaced vertical lines from the data. Corruption 3 was made by randomly deleting 20% of the image, and corruption 4 was made by corrupting evenly spaced horizontal lines. For H1 inpainting 1000 iterations were used. For TV inpainting 40 iterations of simplex optimization were used with starting sigma at 0.01, starting alpha at 100, and optimization step at 0.01. The TV inpainting itself was done in 1000 iterations.

Results

The resulting inpainted images for H1 are shown in Figure 1 and for TV in Figure 3. Note how H1 performed well for corruptions 2-4, but not so well on corruption 1. Whereas TV worked well on corruption 1 but less well on 2-4. Figure 4 shows the energy of the TV optimization for sigma and alpha selection for each case. As can be seen all cases decrease in energy through the optimization. Notably corruption 1 has the lowest overall energy, due to the lower area of corrupted pixels. Figure 5 and Figure 6 show the path of the optimization of alpha and sigma selection in the different corruption cases. As can be seen there is a large difference in the overall trajectories between corruption 1 and the other corruptions.

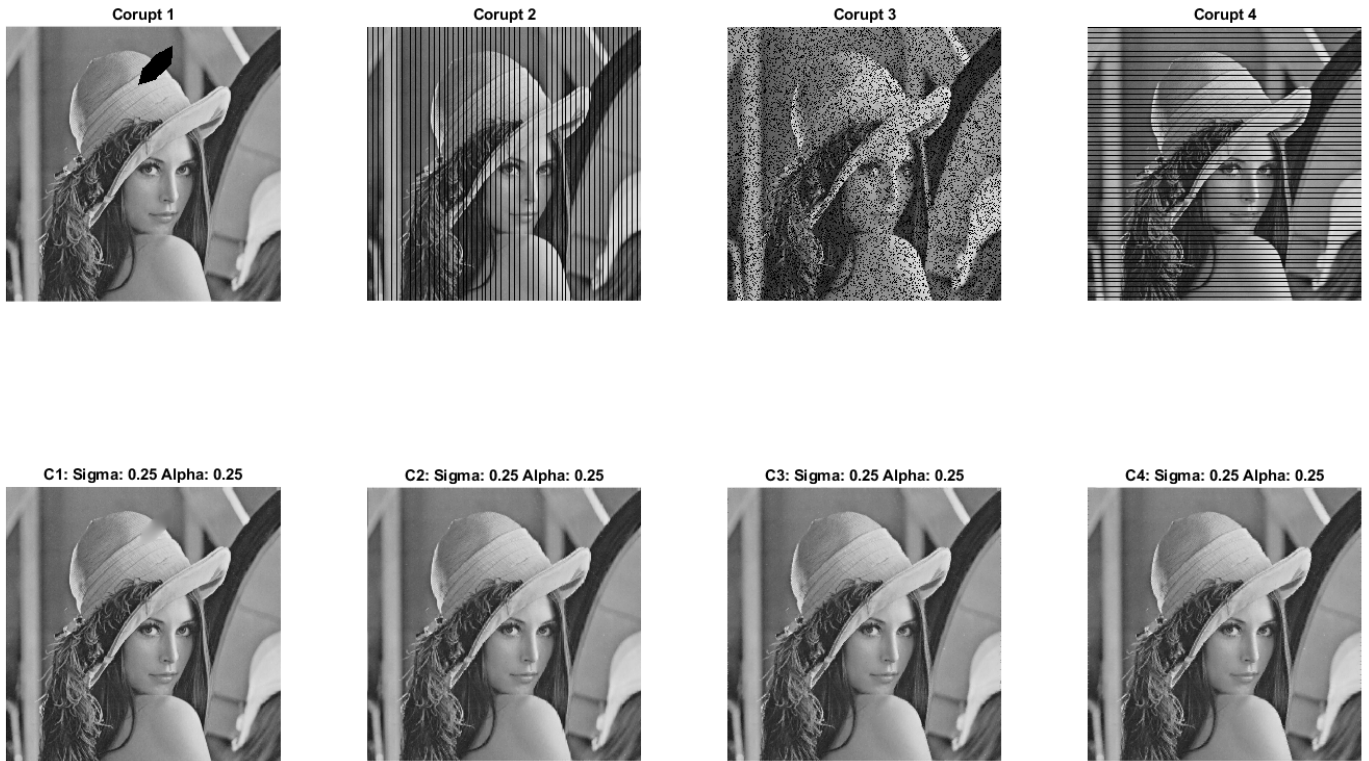


Figure 1: Results from the H1 inpainting. Top row shows the corrupted images. Bottom row shows the H1 inpainted images.

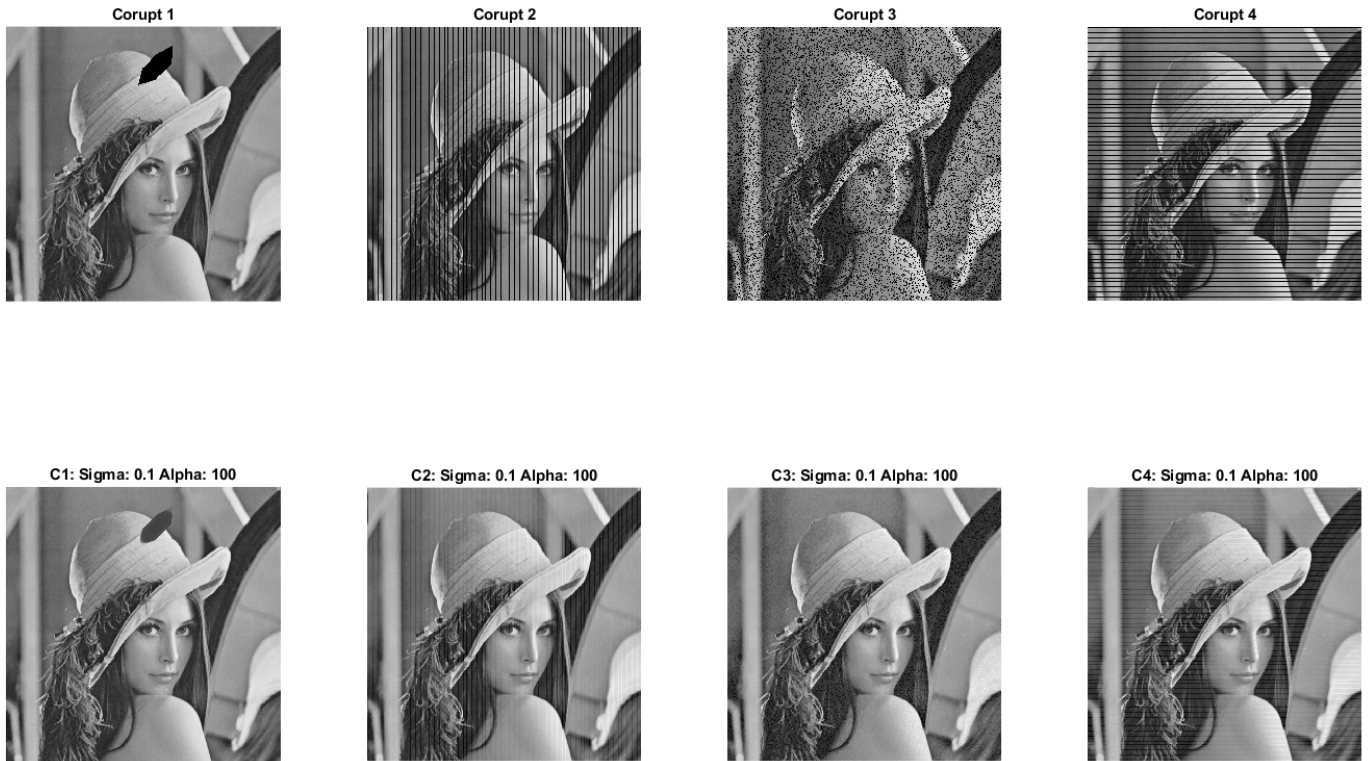


Figure 2: Results from the TV inpainting without optimization. Top row shows the corrupted images. Bottom row shows the H1 inpainted images. Corrupt 1 in red, corrupt 2 in green, corrupt 3 in blue, corrupt 4 in pink

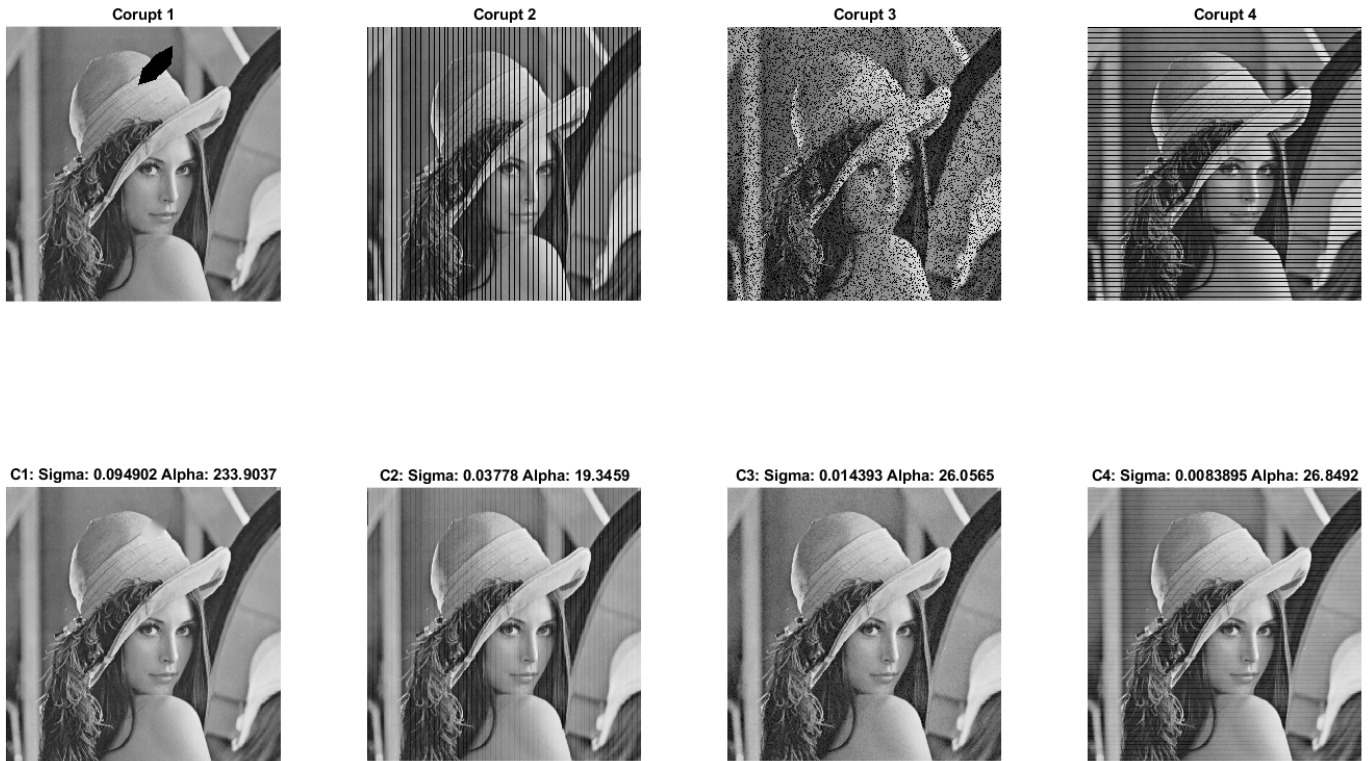


Figure 3: Results from the TV inpainting with optimized sigma and alpha values. Top row shows the corrupted images. Bottom row shows the H1 inpainted images.

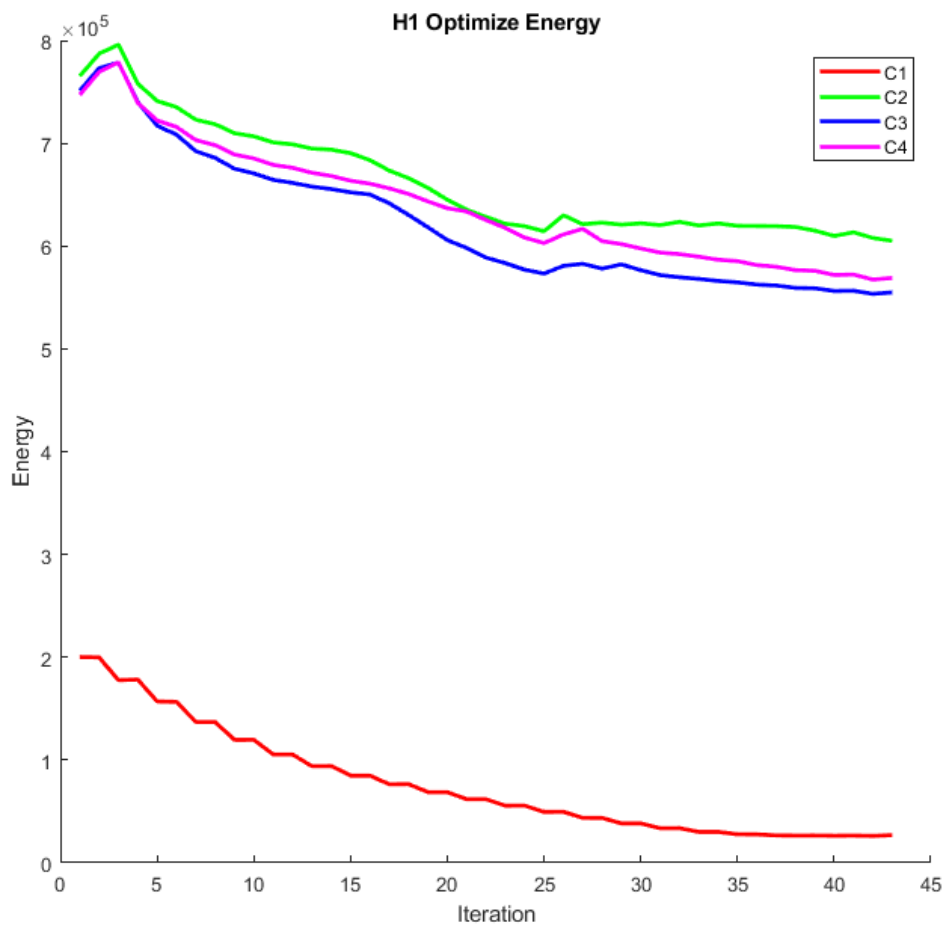


Figure 4: Energy during TV sigma alpha optimization. initial sigma at 0.01, alpha at 100. Guess step size at 0.01, three initial guesses and 40 simplex steps. Corrupt 1 in red, corrupt 2 in green, corrupt 3 in blue, corrupt 4 in pink

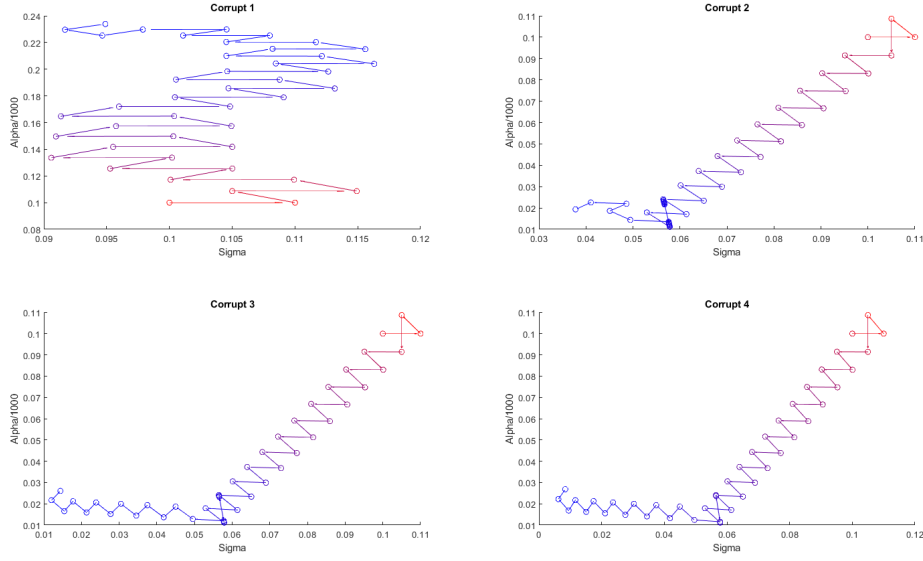


Figure 5: Sigma and alpha values chosen during optimization. Higher energy guesses in red, lower energy in blue. Path of simplex steps shown by vectors.

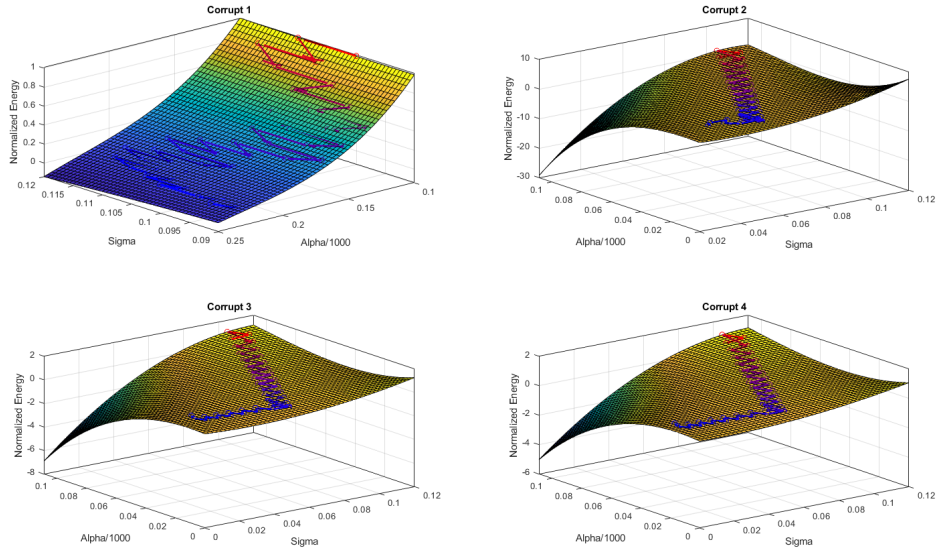


Figure 6: Sigma and alpha values chosen during optimization fit to a surface. Higher energy guesses in red, lower energy in blue. Path of simplex steps shown by vectors.

Discussion

Looking at the results directly from Figure 1 and Figure 3 we can see that there seem to be two classes of corruptions and thus two responses for each method. The first corruption is a

corruption of a local area, which obscures some of the curvature of Lena's hat. The other three corruptions have a more diffuse effect on the image, corrupting small amounts of data but over a wide area (summing to a larger amount of corruption). The H1 method handles the latter cases much better than TV. This is likely because the TV is still allowing these discontinuities in these latter corruptions to persist, as such we still see the remnants of these corruptions in the TV results. The H1 method however fails to do a good job with corruption number one. Particularly it fails to reconstruct the curve of Lena's hat and instead simply blurs into the region with the nearby values. This is, of course expected. And also expectedly the TV is able to reconstruct this curve when alpha and sigma are optimized for.

The idea of there being two classes among these corruptions is further seen when we look at the subsequent visualizations of the TV optimization of sigma and alpha selection. In all cases there are clearly two groups, one with corruption one and one with corruption two - four. In the sigma and alpha selection the major difference can be seen in the trend in alpha values selected. For a large single area missing, a high alpha value is optimal, while with the more spread out corruption a lower alpha is better. Overall it seems that the optimization simplex algorithm I implemented was able to find local minimum of the energy function I defined (EQ 23). When we compare the TV inpaint results with unoptimized sigma and alpha (Figure 2) and the results from the optimized sigma and alpha values (Figure 3) it is apparent mostly in corruption 1. The energy descent however indicates that in all cases the changes to alpha and sigma resulted in better inpainting (Figure 4). Overall I mostly enjoyed this last optimization step as it was 1) self motivated, and 2) the thing I think I understand the best.