

AMATH 515 - OPTIMIZATION
TOTAL VARIATION IMAGE INPAINTING USING SPLIT BREGMAN
MARCH 8, 2024

CATHERINE JOHNSTON

1. MOTIVATION

If a provided digital image is missing pixels in a region, image inpainting is the process of approximating the missing region using the data from the known pixels. A complete image can also have pixels removed in an undesired region and then replaced via inpainting. There are various methods of image inpainting, and the two in particular that we wish to consider in this project are Laplace's equation inpainting (H^1 inpainting) and total variation (TV) inpainting (specifically the split Bregman algorithm), which will both be described in detail in the following sections and then implemented in code to consider various applications. The efficiency (runtime) and accuracy of each method will then be compared. Some example applications of image inpainting are as follows:

- (a) Filling in missing sections of an incomplete image.
- (b) Removing an undesirable feature from an image, such as a stain on cloth.
- (c) Editing a hiking trail map with intersecting trails to remove a trail that has been shut down due to a safety hazard while maintaining continuity of the other trails on the map.

Applications (b) and (c) will be implemented in this project using sample images from the public domain. To simplify the model and its implementation, we consider greyscale images rather than color images.

2. LAPLACE'S EQUATION INPAINTING

A classical approach to inpainting is to apply Laplace's equation in a process also sometimes called H^1 inpainting, which is presented in [GeB]. Consider a greyscale image $g : \Omega \rightarrow \mathbb{R}$, where each $x \in \Omega$ is a (infinitesimally small) pixel in the image and each $g(x) \in \mathbb{R}$ represents the shade of grey of that (infinitesimally small) pixel. This model will of course need to be discretized to be implemented in code. We perform inpainting of an open set of pixels $D \subsetneq \Omega$, not necessarily connected.

Laplace's equation is given by $\Delta u(x) = 0$, where $\Delta = \nabla^2$ is the Laplacian operator. Assuming $g(x) = u(x)$ for all $x \in \Omega \setminus D$, we seek a solution to the Laplace's equation boundary value problem given by

$$\begin{cases} 0 = \Delta u(x), & x \in D, \\ g(x) = u(x), & x \in \partial D, \end{cases}$$

which can be solved for $u(x)$ to determine the inpainted image $u : \Omega \rightarrow \mathbb{R}$. The resulting image as defined above will exactly match the original image $g : \Omega \rightarrow \mathbb{R}$ outside the inpainted region D and will approximate the original image inside the inpainted region.

The name " H^1 inpainting" comes from the use of the H^1 -seminorm of $u(x)$ over D given by

$$\|u\|_{H^1(D)}^2 \doteq \int_D |\nabla u(x)|^2 dx,$$

which is also the L^2 -norm of $\nabla u(x)$ over D . A function $u(x) \in H^1(D)$ if $\|u\|_{H^1(D)} < \infty$. Observe also that $H^1(D) = W^{1,2}(D)$, a Sobolev space of L^2 functions over D . Furthermore, recall Dirichlet's principle that a Laplace's equation inpainting solution in $C^2(D)$ (the set of twice-continuously differentiable functions defined on D) has minimal H^1 -seminorm over D across all functions in $C^2(D)$ satisfying the boundary condition, a result stated in [Ev]. The proof of this fact is derived from Green's identity.

3. TOTAL VARIATION INPAINTING

A more computationally effective method of image inpainting is total variation (TV) inpainting, which is also presented in [GeB]. Instead of the H^1 -seminorm, we use the total variation seminorm of $u(x)$ over Ω , given by

$$\|u\|_{\text{TV}(\Omega)} \doteq \int_{\Omega} |Du(x)| = \sup \left\{ \int_{\Omega} u(x)(\nabla \cdot f(x))dx : f(x) \in C_c^1(\Omega, \mathbb{R}^2)^2, \|f\|_2 \leq 1 \right\},$$

where $C_c^1(\Omega, \mathbb{R}^2)^2$ is the Cartesian product of two spaces of compactly supported continuously differentiable functions from Ω to \mathbb{R}^2 . For smooth $u(x)$, the total variation seminorm can be written as

$$\|u\|_{\text{TV}(\Omega)} = \int_{\Omega} |Du(x)| = \int_{\Omega} |\nabla u(x)|dx,$$

where $u : \Omega \rightarrow \mathbb{R}$ is again the inpainted image. Furthermore, instead of $H^1(D)$, we consider the space of functions of bounded variation (BV) over Ω , where $u(x)$ is in $\text{BV}(\Omega)$ if there exists some Radon measure $Du(x)$ where

$$\int_{\Omega} u(x)(\nabla \cdot f(x))dx = - \int_{\Omega} \langle f(x), Du(x) \rangle$$

for all $f(x) \in C_c^1(\Omega, \mathbb{R}^2)^2$.

To perform total variation inpainting, we seek the function of bounded variation $u(x) \in \text{BV}(\Omega)$ solving the optimization problem

$$\operatorname{argmin}_{u(x) \in \text{BV}(\Omega)} \left[\|u\|_{\text{TV}(\Omega)} + \frac{\lambda}{2} \int_{\Omega \setminus D} (g(x) - u(x))^2 dx \right]$$

for fixed $\lambda > 0$, where we still define open set $D \subsetneq \Omega$. Thus, given original image $g : \Omega \rightarrow \mathbb{R}$, we obtain the inpainted image $u : \Omega \rightarrow \mathbb{R}$, which will exactly match the original image outside the inpainted region D and will approximate the original image inside the inpainted region.

4. IMPLEMENTATION

In this project, total variation image inpainting will be implemented using the split Bregman approach presented in [GeB]. Total variation image inpainting can also be implemented using other algorithms, such as the ADMM algorithm presented in [QGM]. The split Bregman algorithm itself was developed by Goldstein and Osher in [GO].

After discretizing each dimension of the image $g : \Omega \rightarrow \mathbb{R}$ into the N -vector (g_0, \dots, g_{N-1}) , we define its discrete derivative as the forward difference

$$\partial g = \begin{pmatrix} \partial g_0 \\ \vdots \\ \partial g_{N-2} \\ \partial g_{N-1} \end{pmatrix} = \begin{pmatrix} -1 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} g_0 \\ \vdots \\ g_{N-2} \\ g_{N-1} \end{pmatrix}.$$

Since we are working in the two dimensional space $C_c^1(\Omega, \mathbb{R}^2)^2$, we define ∂_x and ∂_y as the discrete derivatives along the dimensions x and y , respectively. Thus, we define the discrete gradient as $\nabla \doteq (\partial_x, \partial_y)^T$. Similarly, we define the discrete divergence $\nabla \cdot \doteq -\nabla^* = -\partial_x^* - \partial_y^*$ and the discrete Laplacian $\Delta \doteq \nabla \cdot \nabla$, where \cdot^* is the adjoint. More particularly, in the image interior, we have the formulas

$$\begin{cases} \nabla \cdot u = u_{ij}^x - u_{(i-1)j}^x + u_{ij}^y - u_{i(j-1)}^y, \\ \Delta u = -4u_{ij} + u_{(i+1)j} + u_{(i-1)j} + u_{i(j+1)} + u_{i(j-1)}, \end{cases}$$

as discussed in [GeA]. Finally, we discretize the total variation seminorm as

$$\|u\|_{\text{TV}(\Omega)} \approx \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\nabla u_{ij}|.$$

To implement Laplace's equation inpainting, we set $u_{ij} = g_{ij}$ for each pixel outside the region D and then perform Gauss-Seidel iterations

$$G_{ij} = \frac{1}{4} [u_{(i+1)j} + u_{(i-1)j} + u_{i(j+1)} + u_{i(j-1)}]$$

away from the boundary, where we then set $u_{ij} = G_{ij}$ for each pixel in the region D . The region D is constructed using an indicator function of all sufficiently white or black pixels. Boundary pixels must be handled case by case, where a more detailed implementation can be seen in our code. We continue to iterate this process until

$$\|u - u_{\text{prev}}\|_2 < \varepsilon$$

for error tolerance $\varepsilon = \|g\|_2 \times 10^{-5}$ as chosen in [GeB].

The split Bregman algorithm for total variation image inpainting solves the constrained optimization problem

$$\begin{cases} \text{argmin}_{d,u} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij}| + \frac{1}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lambda_{ij} (g_{ij} - u_{ij})^2, \\ d = \nabla u. \end{cases}$$

where λ is a pixel-varying parameter chosen to ease computation. Each iteration solves the subproblem

$$\text{argmin}_{d,u} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij}| + \frac{1}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lambda_{ij} (g_{ij} - u_{ij})^2 + \frac{\gamma}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij} - \nabla u_{ij} - b_{ij}|^2$$

for variable b and some parameter $\gamma > 0$, generally a moderate value to ensure fast convergence. We choose $\gamma = 5$ as in [GeB]. This is implemented by alternating between minimizing over d and u for constant u or d , respectively, which is called an alternating direction method.

Since the second term does not depend on d , we minimize over d by solving the optimization problem

$$\operatorname{argmin}_d \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij}| + \frac{\gamma}{2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij} - \nabla u_{ij} - b_{ij}|^2$$

for constant u , which has solution

$$d_{ij} = \frac{\nabla u_{ij} + b_{ij}}{|\nabla u_{ij} + b_{ij}|} \max \left\{ |\nabla u_{ij} + b_{ij}| - \frac{1}{\gamma}, 0 \right\},$$

as provided in [GeA]. Similarly, since the third term does not depend on u , we minimize over u by solving the optimization problem

$$\operatorname{argmin}_u \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lambda_{ij} (g_{ij} - u_{ij})^2 + \gamma \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |d_{ij} - \nabla u_{ij} - b_{ij}|^2$$

for constant d , which yields the solution to the equation

$$\lambda u - \gamma \Delta u = \lambda g - \gamma \nabla \cdot (d - b),$$

which can be approximately solved using one iteration of Gauss-Seidel per iteration of split Bregman. From [GO], we know that (away from the boundary) the solution to the u -subproblem is given by

$$G_{ij} = \frac{\gamma}{\lambda + 4\gamma} \left[u_{(i+1)j} + u_{(i-1)j} + u_{i(j+1)} + u_{i(j-1)} + d_{(i-1)j}^x - d_{ij}^x + d_{i(j-1)}^y - d_{ij}^y - b_{(i-1)j}^x + b_{ij}^x - b_{i(j-1)}^y + b_{ij}^y \right] + \frac{\lambda}{\lambda + 4\gamma} g_{ij},$$

where we then set $u_{ij} = G_{ij}$. Finally, before entering the next iteration, we update $b \mapsto b + \nabla u - d$. We iterate the algorithm until we reach

$$\|u - u_{\text{prev}}\|_2 < \varepsilon$$

for error tolerance $\varepsilon = \|g\|_2 \times 10^{-5}$ as chosen in [GeB].

5. RESULTS

Both Laplace's equation inpainting and total variation inpainting were tested for the following public domain test images, greyscaled using our code:

FIGURE 5.1. An image of a stained strip of cloth, obtained from Creative Commons at https://commons.wikimedia.org/wiki/File:Cloth_strips_with_blood_-_Public_Domain.jpg. Photographed by Lisa Redfern and uploaded in 2014.

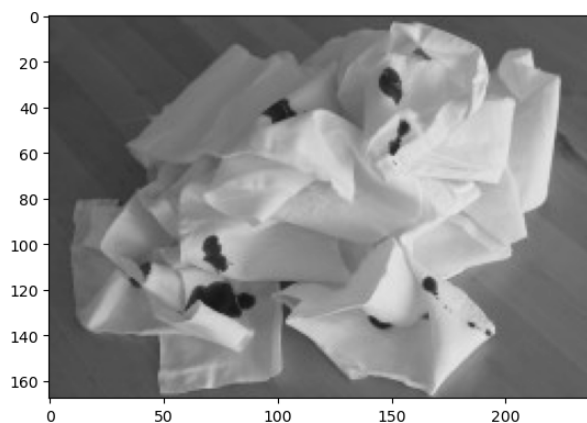
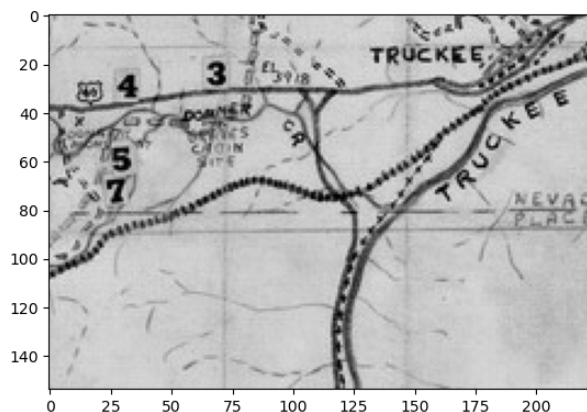


FIGURE 5.2. A cropped version of an image of a 1952 map of Emigrant Trail from P.M. Weddell, obtained from flickr Creative Commons at <https://www.flickr.com/photos/188645737@N03/>. Uploaded by the Donner Summit Historical Society in 2020.



In order to remove the stains from the image of the cloth strips, the image in Figure 5.1 was inpainted in the regions marked below. Similarly, in order to remove a trail from the map of Emigrant Trail, the image in Figure 5.2 was inpainted in the region marked below. Both images are presented below with the inpainted regions marked black and white, respectively:

FIGURE 5.3. Figure 5.1 but with the inpainted regions marked black.

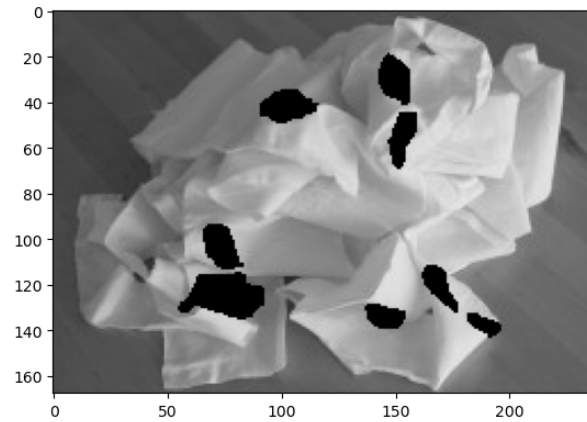
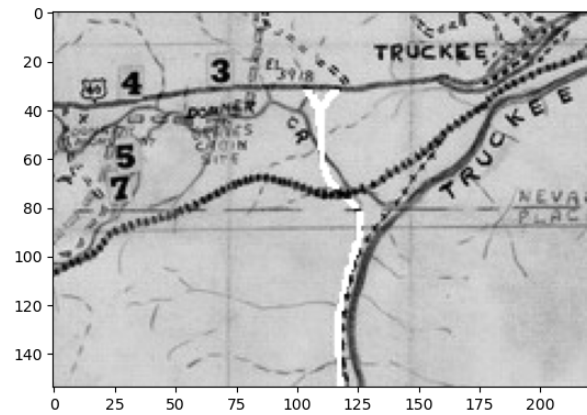


FIGURE 5.4. Figure 5.2 but with the inpainted region marked white.



As the final step prior to implementation, the images were cropped into exact squares using the tool <https://squareanimage.com/>, with the versions of the images actually being implemented below:

FIGURE 5.5. The actually implemented image of the stained cloth, greyscaled and cropped to be square (168×168).

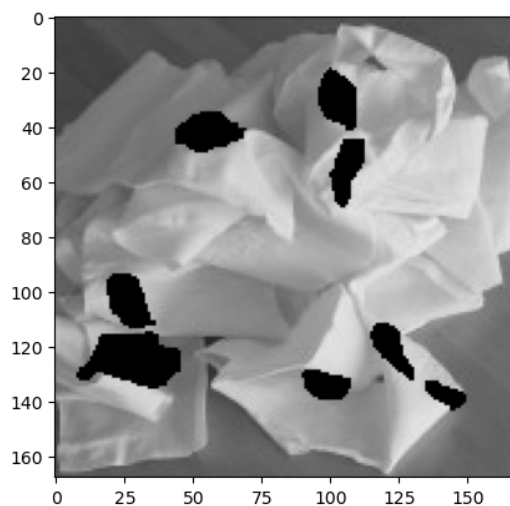
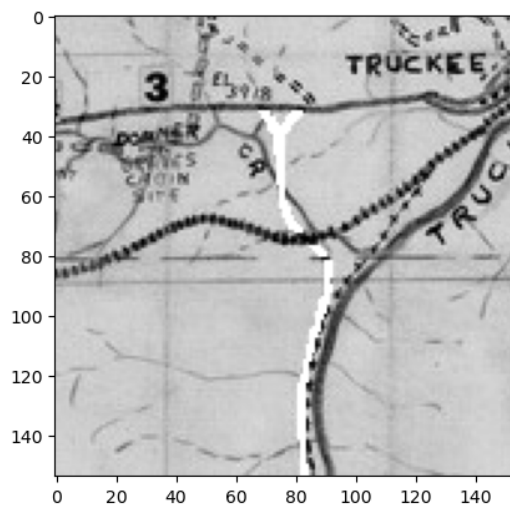


FIGURE 5.6. The actually implemented image of the map of Emigrant Trail, greyscaled and cropped to be square (154×154).



Laplace's equation image inpainting yielded the following results, which were especially good for the image of the stained cloth strip:

FIGURE 5.7. The image of the stained cloth inpainted with Laplace's equation. The stains have now been removed. The runtime to generate this image was 19.241 seconds, using 467 iterations.

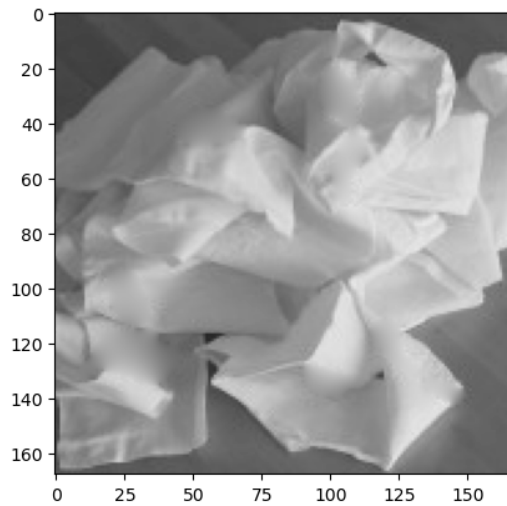
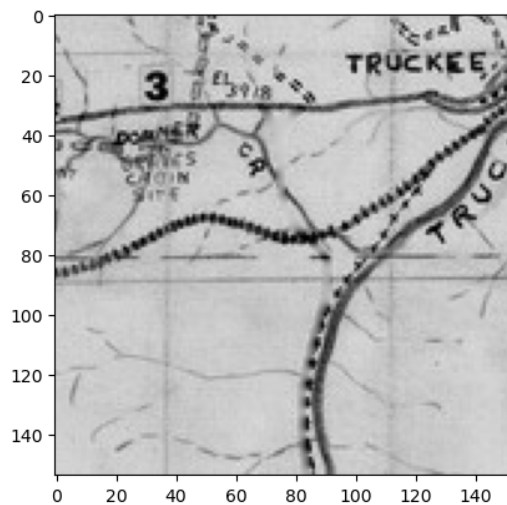


FIGURE 5.8. The image of the map of Emigrant Trail inpainted with Laplace's equation. The trail has now been removed, albeit not as well as the cloth stains. The runtime to generate this image was 1.622 seconds, using 51 iterations.



We encountered several difficulties when implementing total variation image inpainting using the split Bregman approach. First, several details of how to adapt total variation denoising into inpainting were absent from [GeB], the primary source for this project. Second, the contribution of the solution of the d -subproblem canceled out the contribution of the Laplacian of u when solving the u -subproblem using Gauss-Seidel, leading to no pixels being changed each iteration. Regardless, the test images were ran with the same number of iterations to test and compare runtime, although likely the number of iterations would differ if total variation inpainting were implemented correctly. In summary, total variation inpainting took 106.344 seconds to run for the stained cloth and 9.465 seconds to run for the trail map. Around 6 times more time elapsed compared to Laplace's equation inpainting. The code for this project can be found in the Jupyter notebook "Project_Catherine_Johnston.ipynb" and can be accessed using the Github repository <https://github.com/catherinemj/AMATH-515-Project>.

REFERENCES

- [Ev] Evans, L. Partial Differential Equations. *Graduate Studies in Mathematics* **19**, American Mathematical Society, ISBN: 0821807722 (1998).
- [GeA] Getreuer, P. Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. *Image Processing On Line* **2**, 74-95 (2012).
- [GeB] Getreuer, P. Total Variation Inpainting using Split Bregman. *Image Processing On Line* **2**, 147-157 (2012).
- [GO] Goldstein, T. Osher, S. The Split Bregman Method for L1 Regularized Problems. *SIAM Journal on Imaging Sciences* **2**, 323-343 (2009).
- [QGM] Qin, Z. Goldfarb, D. Ma, S. An Alternating Direction Method for Total Variation Denoising. *Optimization Methods and Software* **30**, 594-615 (2014).