Inpainting of Binary Images Using the Cahn-Hilliard Equation

Andrea Bertozzi, Selim Esedoglu, and Alan Gillette*

***Abstract*— Image inpainting is the filling in of missing or damaged regions of images using information from surrounding areas. We outline here the use of a model for inpainting based on the Cahn-Hilliard equation, which allows for fast, efficient inpainting of degraded text, as well as super-resolution of high contrast images.**

***Index Terms*— Image inpainting, super-resolution, binary images, Cahn-Hilliard equation**

EDICS Category: ISR-INTR, ISR-SUPR, FLT-PDEQ

I. Introduction

IMAGE inpainting is the filling in of damaged or missing regions of an image with the use of information from surrounding areas. In its essence, it is a type of interpolation. Its applications include restoration of old paintings by museum artists, and removing scratches from photographs.

The pioneering work of Bertalmio et. al. [1] introduced image inpainting for digital image processing. Their model is based on nonlinear partial differential equations, and is designed to imitate the techniques of museum artists who specialize in restoration. In particular, they focused on the principle that good inpainting algorithms should propagate sharp edges in surrounding areas into the damaged parts that need to be filled in. This can be done, for instance, by connecting contours of constant grayscale image intensity (called isophotes) to each other across the inpainting region (see also Masnou and Morel [2]), so that gray levels at the edge of the the damaged region get extended to the interior continuously. They also impose the direction of the isophotes as a boundary condition at the edge of the inpainting domain.

In subsequent work with Bertozzi [3], they realized that the nonlinear PDE introduced in [1] has intimate connections with two dimensional fluid dynamics through the Naiver-Stokes equation. Indeed, it turns out that the steady state problem originally proposed in [1] is equivalent to the inviscid Euler equations from incompressible flow, in which the image intensity function plays the role of the stream function in the fluid problem. The natural boundary conditions for inpainting are to match the image intensity on the boundary of the inpainting region, and also the direction of the isophote lines ($\nabla^{\perp} I$). For the fluid problem this is effectively a generalized 'no-slip' boundary condition that requires a Navier-Stokes formulation, introducing a diffusion term. This analogy also shows why diffusion is required in the original inpainting problem. In practice nonlinear diffusion (as in Perona-Malik [4], ROF [5]) works very well to avoid blurring of edges in the inpainting.

A different approach to inpainting was proposed by Chan and Shen [6]. They introduced the idea that well-known variational image denoising and segmentation models can be easily adapted to the inpainting task by a simple modification. In particular, these models often include a fidelity term that keeps the solutions close to the given image. By restricting the effects of the fidelity term to the complement of the inpainting region, Chan and Shen showed that very good image completions can be obtained. The principle behind their approach can be summarized as follows: variational denoising and segmentation models all have an underlying notion of what constitutes an image. In the inpainting region, the models of Chan and Shen reconstruct the missing image features by relying on this built-in notion of what constitutes a natural image.

The first model introduced by Chan and Shen used the total variation based image denoising model of Rudin, Osher, and Fatemi [5] for the inpainting purpose. This model can successfully propagate sharp edges into the damaged domain. However, because of a regularization term, the model exacts a penalty on the length of edges, and thus the inpainting model cannot connect contours across very large distances. Another caveat is that this model does not continuously extend the direction of isophotes across the boundary of the inpainting domain. However, for an interesting application of this method, see Kang, Chan, and Soatto [7].

Subsequently, Kang, Chan, and Shen [8] introduced a new variational image inpainting model that addressed the caveats of the total variation based one. Their model is motivated by the work of Nitzberg, Mumford, and Shiota, [9] and includes a new regularization term that penalizes not merely the length of edges in an image, but the integral of the square of curvature along the edge contours. This allows both for isophotes to be connected across large distances, and their directions to be kept continuous across the edge of the inpainting region.

Following in the footsteps of Chan and Shen, Esedoglu and Shen [10] adapted the Mumford-Shah image segmentation model [11] to the inpainting problem for grayscale images. They utilized Ambrosio and Tortorelli's elliptic approximations [12] to the Mumford-Shah functional. Gradient descent for these approximations leads to parabolic equations with a small parameter ε in them; they represent edges in the image by transition regions of thickness ε. These equations have the benefit that highest order derivatives are linear. They can therefore be solved rather quickly. However, like the total variation image denoising model, the Mumford-Shah segmentation model penalizes length of edge contours, and as a result does not allow for the connection of isophotes across large distances in inpainting applications.

In order to improve the utility of the Mumford-Shah model in inpainting, Esedoglu and Shen introduced the Mumford-Shah-Euler image model that, just like the previous work of Kang, Chan, and Shen [8], penalizes the square of the curvature along an edge contour. Following previous work by March [13], they then used a conjecture of De Giorgi [14] to approximate the resulting variational problem by an elliptic one. The resulting gradient descent equations are fourth order, nonlinear parabolic PDEs with a small parameter in them.

More recently, Grossauer and Scherzer [15] have used the complex Ginzburg-Landau equation in a technique for inpainting grayscale images. This method assigns the real part u of a complex quantity $w = u + iv$ to be the grayscale values of the image. The complex quantity w is then forced by their algorithm to reside on a circle of radius 1, centered at the origin, in the complex plane. The complex

December 9, 2005. This work was supported by the National Science Foundation and the intelligence community through the joint ”Approaches to Combat Terrorism” program (NSF Grant AST-0442037). This work is also supported by ONR grant N000140410078 and NSF grants CCF-0321917 and DMS-0410085.

A. Bertozzi and A. Gillette are with the UCLA Mathematics Department, Box 951555, Los Angeles, CA 90095-1555 USA (email: bertozzi@math.ucla.edu; agillett@math.ucla.edu)

S. Esedoglu is with the Department of Mathematics, University of Michigan, 530 Church Street, Ann Arbor, MI 48109-1043 USA (email: esedoglu@umich.edu)

Ginzburg-Landau equation then leads to a coupled system to be solved for u and v, respectively.

II. The Modified Cahn-Hilliard Equation Inpainting Model

Our idea is that a much simpler class of models exist that still has many of the desirable properties of the model introduced in [10], but for which there are very fast computational techniques available. In particular, we show that in the case of high-contrast or binary images, a slightly modified Cahn-Hilliard equation allows us to obtain inpaintings as good as the ones in previous papers, but achieves them much more rapidly. This faster method is a result of both a new simplified PDE model and the use of fast solvers for such a model.

Let $f(\vec{x})$, where $\vec{x} = (x, y)$, be a given image in a domain Ω, and suppose that $D \subset \Omega$ is the inpainting domain. Let $u(\vec{x}, t)$ evolve in time to become a fully inpainted version of $f(\vec{x})$ under the following equation:

$$u_t \ = \ -\Delta(\varepsilon\Delta u - \frac{1}{\varepsilon}W'(u)) + \lambda(\vec{x})(f - u) \tag{1}$$

where

$$\lambda(\vec{x}) \ = \ \begin{cases} 0 & \text{if } \vec{x} \in D, \\ \lambda_0 & \text{if } \vec{x} \in \Omega \setminus D, \end{cases}$$

The function $W(u)$ is a nonlinear potential with wells corresponding to values of u that are taken on by most of the grayscale values. In the examples considered here, we use binary images in which most of the pixels are either exactly black or white. In this binary case, W should have wells at the values $u = 0$ and $u = 1$. In the examples presented in this document we use the function $W(u) = u^2(u-1)^2$, however other functions could be used. We assume that the image function $u(\vec{x}, t)$ takes on grayscale values in a domain Ω. and satisfies periodic boundary conditions on $\partial\Omega$. Alternatively, Neumann boundary conditions could be used, or any boundary conditions for which one can use fast solvers for the equation (see discussion below). Equation (1) is what we will call the *modified* Cahn-Hilliard equation, due to the added fidelity term $\lambda(\vec{x})(f - u)$.

The role of ε in equation (1) is important. In the original Cahn-Hilliard equation, ε serves as a measure of the transition region between two metals in an alloy, after heating and reaching a steady state. Applied to image processing, ε is a measure of the transition region between the two grayscale states – for example between the black and white of printed text.

Another important feature of this new idea is that fast solvers exist for the numerical integration of the Cahn-Hilliard equation and similar diffuse interface equations. To date no such solvers have been applied to these problems in the context of imaging applications, and we believe that this synergistic combination of a simpler PDE-based method and a state-of-the-art fast solver provides significant improvement over the previous state-of-the-art (see section IV).

Here we demonstrate how to implement this idea using a specific fast solver known as *convexity splitting* [16], [17]. However, other fast solvers might be used with good performance. Convexity splitting involves dividing up the energy functional for the equation into two parts – a convex energy plus a concave energy. The part of the Euler-Lagrange equation derived from convex portion is then treated implicitly in the numerical scheme, while the portion derived from the concave part is treated explicitly.

Under the right conditions, convexity splitting for gradient flow-derived equations can allow for an unconditionally gradient stable time-discretization scheme, which means arbitrarily large time steps can be taken. Vollmayr-Lee and Rutenberg [17] have more recently refined the conditions under which stability is applicable.

The new modified Cahn-Hilliard equation is not strictly a gradient flow. The original Cahn-Hilliard equation (equation (1) with $\lambda = 0$) is indeed a gradient flow using an H^{-1} norm for the energy

$$E_1 \ = \ \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}W(u) \ d\vec{x}, \tag{2}$$

while the fidelity term in equation (1) can be derived from a gradient flow under an L^2 norm for the energy

$$E_2 \ = \ \lambda_0 \int_{\Omega\setminus D} (f - u)^2 d\vec{x}. \tag{3}$$

But in total, the modified Cahn-Hilliard equation is neither a gradient flow in H^{-1} nor L^2. However, the idea of convexity splitting, one for the Cahn-Hilliard energy in equation (2) and one for the energy E_2 in equation (3), can still be applied to this problem with good results.

For example, one can split E_1 as

$$E_1 \ = \ E_{11} \ - \ E_{12} \tag{4}$$

where

$$E_{11} = \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{C_1}{2}|u|^2 \ d\vec{x} \tag{5}$$

and

$$E_{12} \ = \ \int_\Omega -\frac{1}{\varepsilon}W(u) + \frac{C_1}{2}|u|^2 \ d\vec{x}. \tag{6}$$

A possible splitting for E_2 is

$$E_2 \ = \ E_{21} \ - \ E_{22} \tag{7}$$

where

$$E_{21} \ = \ \int_{\Omega\setminus D} \frac{C_2}{2}|u|^2 \ d\vec{x} \tag{8}$$

and

$$E_{22} \ = \ \int_{\Omega\setminus D} -\lambda_0(f-u)^2 + \frac{C_2}{2}|u|^2 \ d\vec{x}. \tag{9}$$

For the splittings discussed above, the resulting time-stepping scheme is:

$$\frac{u^{n+1} - u^n}{\Delta t} \ = \ -\nabla_{H^{-1}}(E_{11}^{n+1} - E_{12}^n) \ -\nabla_{L^2}(E_{21}^{n+1} - E_{22}^n) \tag{10}$$

where $\nabla_{H^{-1}}$ and ∇_{L^2} represent gradient descent with respect to the H^{-1} inner product, and L^2 inner product, respectively. This translates to a numerical scheme of the form

$$\begin{aligned} \frac{u^{n+1}(\vec{x}) - u^n(\vec{x})}{\partial t} &+ \varepsilon\Delta^2 u^{n+1}(\vec{x}) - C_1\Delta u^{n+1}(\vec{x}) + C_2 u^{n+1}(\vec{x}) \\ &= \Delta(\frac{1}{\varepsilon}W'(u^n(\vec{x}))) + \lambda(\vec{x})(f(\vec{x}) - u^n(\vec{x})) \\ &\quad - C_1\Delta u^n(\vec{x}) + C_2 u^n(\vec{x}). \end{aligned} \tag{11}$$

The constants C_1 and C_2 are positive, and need to be chosen large enough so that the energies E_{11}, E_{12}, E_{21}, and E_{22} are convex. C_1 should be comparable to $\frac{1}{\varepsilon}$, while C_2 should be comparable to λ_0. Numerical tests have shown that with these choices the scheme

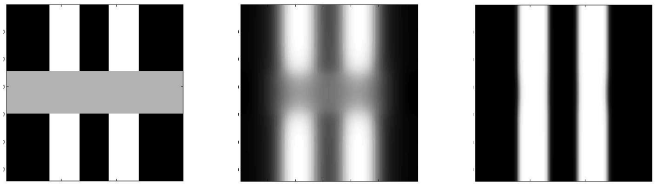
(11) becomes unconditionally stable. Equation (11) is then solved for $u^{n+1}(\vec{x})$, given $u^n(\vec{x})$, by way of a two-dimensional Fast-Fourier-Transform method. We present some examples below in which we indicate precise values of Δt, C_i, λ, and ε used to perform the inpainting.

Finally we mention that one can perform inpainting across larger regions by considering a two-step method. The inpainting is done first with a larger ε, which results in topological reconnection of shapes with edges smeared by diffusion. The second step then uses the results of the first step and continues with a much smaller value of ε in order to sharpen the edge after reconnection. In practice such a two-stage process can result in inpainting of a stripe across a region that is over ten times the width of the stripe, without any a priori knowledge of the location of the stripe.

III. Examples

The modified Cahn-Hilliard equation lends itself particularly well to the inpainting of simple binary shapes, such as stripes and circles. Moreover, its applicability can be extended to achieve inpainting of objects composed of stripes and circles, i.e., roads or text. Below and on the next page, we show several examples of the method and its performance. All examples were performed on a Linux desktop system using a Pentium 4 processor, and programmed in MATLAB.

A. Inpainting of a double stripe

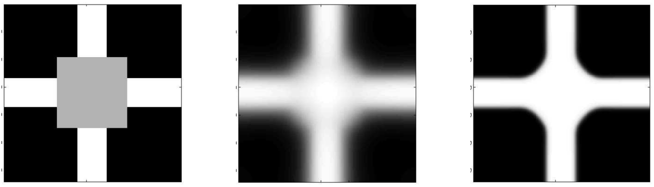


(a) (b) (c)

Fig. 1. (a) Initial data (inpainting region in gray). (b) Intermediate state at $t = 50$. (c) steady state at $t = 700$. (Gap distance is 30 units, Image domain is 128x128).

Above in figure 1, we see the two-step process at work to inpaint two stripes. The gray region in figure 1(a) denotes the inpainting region. We begin running the modified Cahn-Hilliard equation with a large value of ε $(= .8)$, and at $t = 50$ we reach a steady state. We then switch to a small value of ε $(= .01)$, using the result from figure 1(b) as initial data. The final result is reached at $t = 700$ and is shown in figure 1(c). In this test, Δt was set to 1, $\lambda = 50,000$, $C_1 = 300$, and $C_2 = 150,000$.

B. Inpainting of a cross

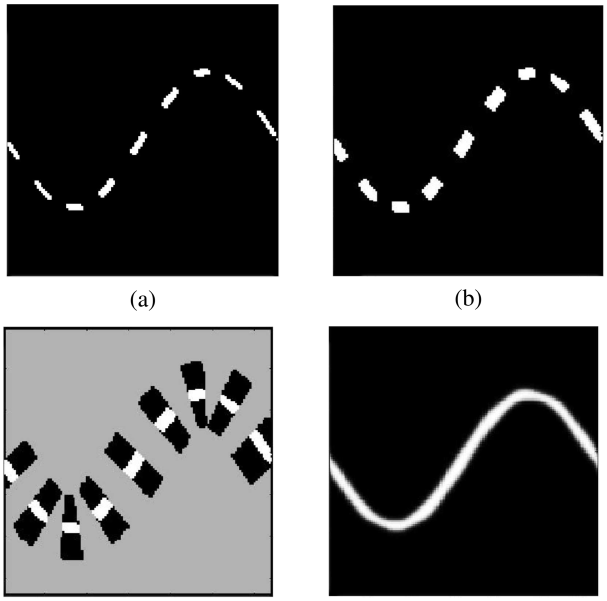


(a) (b) (c)

Fig. 2. (a) Initial data of cross (inpainting region in gray). (b) Intermediate state at $t = 300$. (c) Steady state at $t = 1000$. (Image domain is 128x128, stripe width is 20 units, initial gap distance is 50 units).

In figure 2(a), the gray region denotes the "gap", or region to be inpainted. As with the stripes, the modified Cahn-Hilliard equation is run to steady state for a large value of ε $(= .8)$, resulting in figure 2(b) at $t = 300$. This data is then used as initial data for the modified Cahn-Hilliard equation with ε $(= .01)$ set to a small value. The final result is a completed cross at $t = 1000$. The parameters were set as $\Delta t = 1$, $\lambda = 100,000$, $C_1 = 300$, and $C_2 = 3\lambda$.

C. Inpainting of a Sine Wave



(a) (b)

(c) (d)

Fig. 3. Inpainting a sine wave. (Image domain is 128x128).

Figure 3 shows how the modified Cahn-Hilliard equation may be applied to the inpainting of simple road-like structures. In figure 3(a), an incomplete sine wave is shown. In figure 3(b), the sine wave is artificially "fattened" by expanding each white point's area radially by a factor of 3. This is done in order to give the modified Cahn-Hilliard equation sufficient boundary conditions to do effective inpainting.

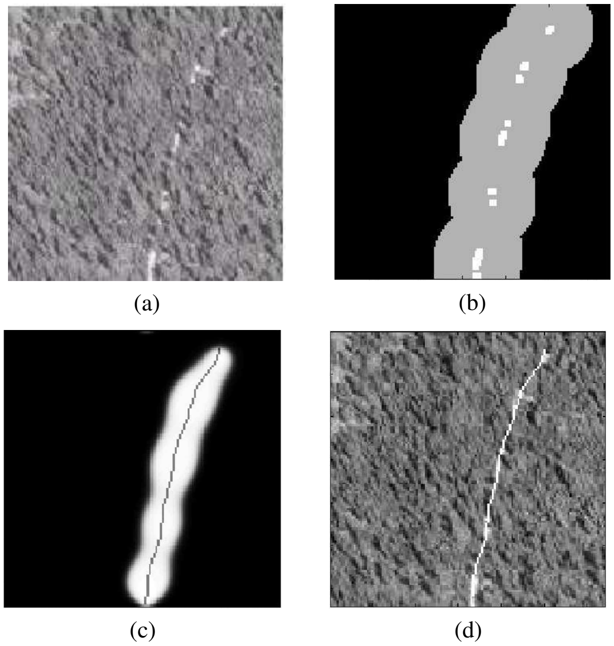
In figure 3(c), the gray area represents the inpainting region. The remaining white and black portions of the image are thus outside the inpainting region, and essentially held fixed in place by the fidelity term of the modified Cahn-Hilliard equation (1). The two-step method was then used to inpaint the sine wave. Figure 3(d) shows the finished result.

The initial value of ε was taken to be .8, and then at $t = 200$ this was switched to a value of $\varepsilon = .01$. The final inpainting result was taken at $t = 4000$ (which corresponds to a time of 24 seconds real processing time). The parameters were set as: $\Delta t = 1$, $C_1 = 300$, and $C_2 = 3\lambda$.

D. Inpainting of a Road

Figure 4(a) shows a satellite image of a road passing through a forest in Washington state. After a simple thresholding of grayscale values, the visible pieces of the road are shown as the white regions in figure 4(b). The gray area in figure 4(b) represents the inpainting region, which was found by creating a circle about each established point of the road, the radius of which was chosen to be the maximum estimated gap length between existing portions of the road.

Note also that each thresholded white point of the road has been expanded in radius, as was done for the sine wave in figure 3(b). In



(a) (b)

(c) (d)

Fig. 4. Inpainting of an obscured road. (Image domain is 128x128).

the original satellite photo, the road actually has an average width of about 1 pixel, making it very difficult to establish meaningful boundary conditions for the inpainting problem.

In figure 4(c), steady state has been reached using the modified Cahn-Hilliard equation, via the aforementioned two-step process. The result in (c) is too thick, but if a centerline extraction is performed, and the resulting centerline overlaid on the initial satellite photo, we arrive at an estimation of the path of the road through the trees as shown in figure 4(d). Note that the result in figure 4(d) does not continue the road to the top of the satellite photo. This is due to a lack of data for the road in that region, as exemplified by figure 4(b).

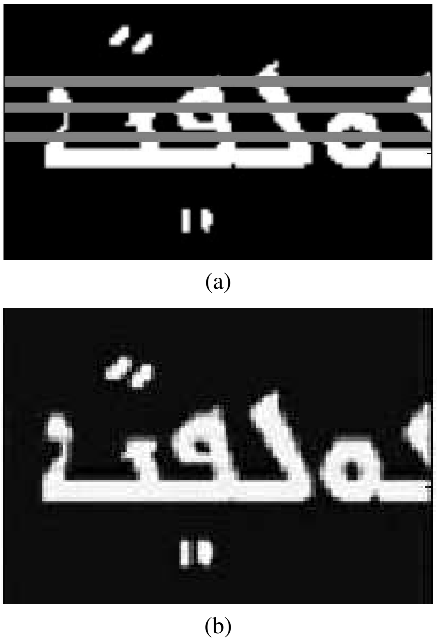
The initial value for ε was .8, which was switched at $t = 100$ to $\varepsilon = .005$. The final result was taken at $t = 500$ (which corresponds to 6 seconds of processor time). The parameters were: $\Delta t = 1$, $\lambda = 1,000,000$, $C_1 = 30,000$, and $C_2 = 3\lambda$.

Much more efficient inpainting, akin to what was accomplished for the sine wave in figure 3 could be done, if a smaller inpainting region could be determined. For example, the reason that figure 4(c) displays such a poor representation of a road is due to the inpainting region literally being too wide (the gray portion of figure 4(b)). If we could come up with an inpainting region similar to that in figure 3(c), much better approximations to the road could be accomplished, possibly negating the need for centerline extraction.

E. Recovery of Text

In figure 5(a), several lines obscure some Arabic writing. Using these obstructing lines as the inpainting region, the modified Cahn-Hilliard two-step scheme can inpaint the occluded parts of the writing. The initial value for ε was .08. At $t = 100$, ε was switched to .01. The program was then run to 1000 time steps. Δt was set to 1, the fidelity constant λ was set to 50,000,000, C_1 was set to 10,000, while C_2 was set to 3λ. The final inpainting result is shown in figure 5(b).

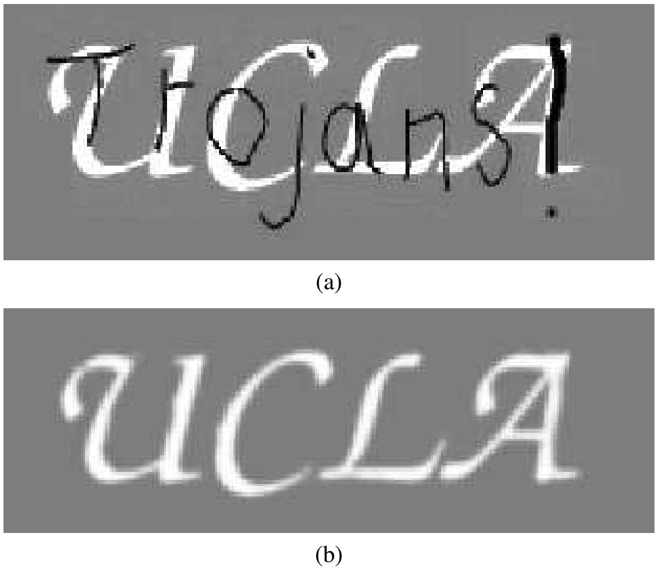
In figure 6(a), graffiti is written over the UCLA logo. Using the graffiti as the inpainting region, the modified Cahn-Hilliard equation inpaints the missing logo parts by the two-step method. Until $t = 50$, a large value of ε $(= .8)$ is used. At $t = 50$, ε is switched to a small value $(= .005)$. The final result in figure 6(b) is the completed logo, looking no worse for wear after its encounter with graffiti. Δt was set to 1, the fidelity constant λ was set to 50,000,000, C_1 was set to 15,000, while C_2 was set to 3λ.



(a)

(b)

Fig. 5. Recovery of damaged text. (Image domain was 128x128).



(a)

(b)

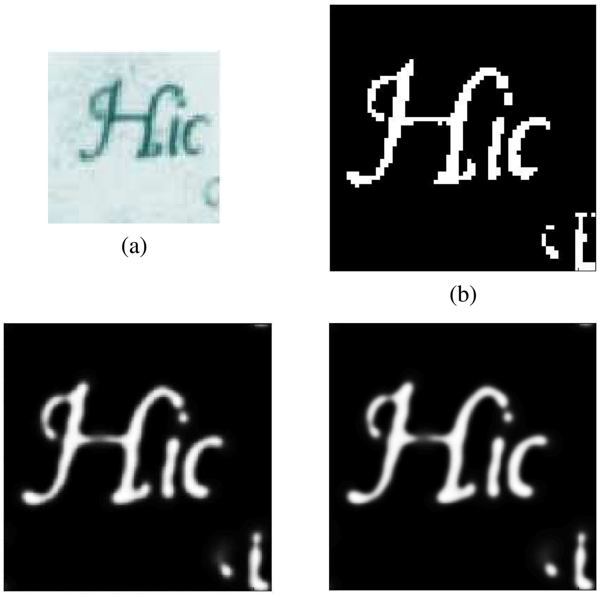
Fig. 6. Recovery of damaged text. (Image domain was 256x256).

F. Super-resolution

The modified Cahn-Hilliard equation can also be used for the purposes of super-resolution of text. Latin writing is shown in figure 7(a), of size 64X64. Figure 7(b) shows the text enlarged by 3X using MATLAB's "nearest-neighbor" algorithm.

First, the white region of figure 7(b) is subsampled to provide initial data for inpainting. Next, the modified Cahn-Hilliard algorithm runs until $t = 40$ using a very large fidelity constant, $\lambda = 50,000,000$, and very small ε (=.005).

After $t = 40$, λ is set equal to zero, and the ordinary Cahn-Hilliard equation is allowed to run on the text. This allows for the smoothing of jagged parts of the text that appeared after magnification (figure 7(b)). Figure 7(c) and 7(d) show the results at $t = 350$ and $t = 450$



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

Fig. 7. Super-resolution of text. Magnification 3X. Original size 64x64.

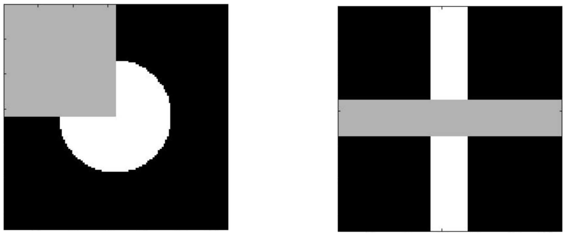
respectively. Throughout this test, C_1 was set to 300, C_2 was set to 150,000,000, and ε was set to .005. This particular test used a constant value of ε.

IV. Comparison with other methods

One of the chief benefits of using the modified Cahn-Hilliard (mCH) equation to do inpainting are the fast numerical techniques available for its solution. To quantitatively determine how much faster this makes the modified Cahn-Hilliard equation than other binary inpainting techniques, a series of comparison tests were run.

The methods we tested against were the Curvature Driven Diffusion (CDD) inpainting model of Chan and Shen [18], the Euler's Elastica (EE) inpainting model of Chan, Kang, and Shen [8], and the Mumford-Shah-Euler (MSE) inpainting model of Esedoglu and Shen [10].

Each method was tested on two examples – inpainting a 3/4 circle, and inpainting a disconnected stripe. All tests were run on the same system used in section III (with the exception that the EE method was programmed in C++).



$\frac{3}{4}$ circle Disconnected stripe

Fig. 8. Inpainting data for comparison tests. Gray color denotes inpainting regions.

A. Graphic Results

Figures 9 and 10 show the performance of each inpainting method on the circle and stripe tests, respectively. As can be seen in figure 9, CDD requires random data to begin inpainting the circle (CDD^2). The EE method fared well on the circle test with zero initial data in the inpainting region (EE^1), but became mired when the test was started with random data there (EE^2).

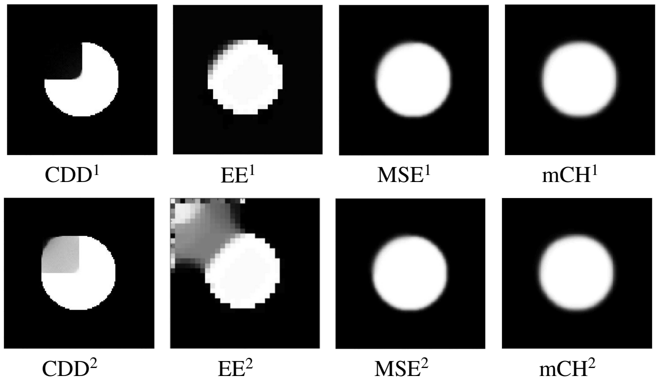


Fig. 9. Results for the circle inpainting test. ^{1} – zero initial data assumed in inpainting region. ^{2} – random initial data assumed in inpainting region.

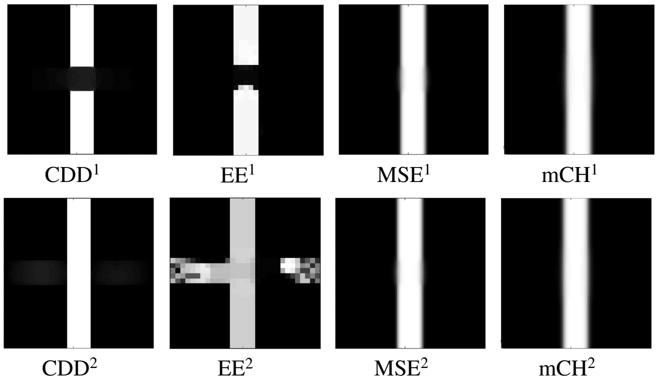


Fig. 10. Results for the stripe inpainting test. ^{1} – zero initial data assumed in inpainting region. ^{2} – random initial data assumed in inpainting region.

The MSE and mCH methods, however, had no strict preference for the initial data in the inpainting region. Results were the same whether random or zero initial data was assumed (MSE ^{1,2}, mCH ^{1,2}).

B. Tabulated Results

Tables I and II show the timing results for each method. These are the correct times for the graphical results shown in figures 9 and 10.

TABLE I
Comparison Tests, Inpainting Region Set to Zero

Method	Inpainting Time (seconds)	
	Circle	Stripe
CDD	>5,400	>5,400
EE*	>18,000	>18,000
MSE	45	24
mCH	24	6

* 30X30 grid used. All others 128X128.

V. Conclusion

We have shown how the Cahn-Hilliard equation can be modified to achieve fast inpainting of binary imagery. This modified Cahn-Hilliard equation can be applied to the inpainting of simple binary shapes, text reparation, road interpolation, and super-resolution. The two-step process we employ, described at the end of section II, allows for effective inpainting across large unknown regions. Although it is

TABLE II
COMPARISON TESTS, INPAINTING REGION SET TO RANDOM DATA

Method	Inpainting Time (seconds)	
	Circle	Stripe
CDD	>270	>270
EE*	>1,800	>1,800
MSE	300	30
mCH	24	5

* 30X30 grid used. All others 128X128.

generally desired for the end-user to specify the inpainting domain, this method can be used for interpolating simple roads and other situations where a user-defined inpainting region is not feasible.

This method assumes zero data in the inpainting region. The two-step process then channels the solution toward the desired steady state in a repeatable process. Although at least one undesirable steady state may be possible mathematically, the method steers away from this by first achieving a very rough but wide-ranging inpainting, and then using this state as initial data for a subsequent inpainting with sharp transitions between white and black regions.

In the context of binary image inpainting, the modified Cahn-Hilliard equation has displayed a considerable decrease in computation time when compared with other inpainting methods. Fast numerical techniques available for the Cahn-Hilliard equation also allow for larger data sets to be processed, greatly aiding the speed of computation.

ACKNOWLEDGMENT

The authors would like to thank the National Geospatial Intelligence Agency for suggesting the road example, and Jean-Michel Morel for useful suggestions regarding super-resolution techniques.

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Siggraph 2000, Computer Graphics Proceedings*, K. Akeley, Ed. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000, pp. 417–424.

[2] S. Masnou and J. Morel, "Level lines based disocclusion," *5th IEEE Int'l Conf. on Image Processing, Chicago, IL*, Oct. 4-7, 1998.

[3] M. Bertalmio, A. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," *IEEE Computer Vision and Pattern Recognition (CVPR), Hawaii*, vol. 1, pp. I355–I362, December 2001.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. **12**, pp. 629–639, 1990.

[5] L. Rudin, S. Osher, and E. Fatemi, "Non linear total variation based noise removal algorithms," *Physica D*, vol. **60**, pp. 259–268, 1992.

[6] T. F. Chan and J. Shen, "Mathematical models of local non-texture inpaintings," *SIAM Journal on Applied Mathematics*, vol. **62**, no. 3, pp. 1019–1043, 2001.

[7] S. Kang, T. Chan, and S. Soatto, "Landmark based inpainting from multiple views," *UCLA Math CAM 02-11, 2002*.

[8] T. Chan, J. Shen, and S. Kang, "Euler's elastica and curvature-based image inpainting," *SIAM Journal on Applied Mathematics*, vol. **63**, no. 2, pp. 564–592, 2002.

[9] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1993, no. 662.

[10] S. Esedoglu and J. Shen, "Digital inpainting based on the Mumford-Shah-Euler image model," *European Journal of Applied Mathematics*, vol. **13**, pp. 353–370, 2002.

[11] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. **42**, pp. 577–685, 1989.

[12] L. Ambrosio and V. M. Tortorelli, "Approximation of functionals depending on jumps by elliptic functionals via gamma convergence," *Communications on Pure and Applied Mathematics*, vol. **43**, pp. 999–1036, 1990.

[13] R. March and M. Dozio, "A variational method for the recovery of smooth boundaries," *Image and Vision Computing*, vol. **15**, pp. 705–712, 1997.

[14] E. De Giorgi, "Some remarks on gamma-convergence and least squares methods," in *Composite Media and Homogenization Theory*, G. D. Maso and G. F. Dell'Antonio, Eds. Birkhauser, 1991, pp. 135–142.

[15] H. Grossauer and O. Scherzer, "Using the complex Ginzburg-Landau equation for digital inpainting in 2d and 3d," *Scale Space Methods in Computer Vision, Lecture Notes in Computer Science 2695*, pp. 225–236, 2003.

[16] D. Eyre, "An unconditionally stable one-step scheme for gradient systems," Unpublished paper, June 9, 1998.

[17] B. P. Vollmayr-Lee and A. D. Rutenberg, "Fast and accurate coarsening simulation with an unconditionally stable time step," *Physical Review E*, vol. **68**, no. 0066703, pp. 1–13, 2003.

[18] T. F. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. **12**, no. 4, pp. 436–449, 2001.

Andrea Bertozzi received A. B., M. A., and Ph.D. degrees in Mathematics from Princeton University in 1987, 1988, and 1991 respectively. She was on the faculty of the University of Chicago from 1991-1995 and on the faculty of Duke University from 1995-2004. During 1995-6 she was Maria Goeppert-Mayer Distinguished Scholar at Argonne National Laboratory. Since 2003 she has been with UCLA as Professor of Mathematics and currently serves as Director of the Program in Computational and Applied Mathematics. Her current research interests include image inpainting, image segmentation, cooperative control of robotic vehicles, swarming, and fluid interfaces.

Prof. Bertozzi is a member of the Society for Industrial and Applied Mathematics, the American Mathematical Society, the American Physical Society, and the Association for Women in Mathematics. She has served as a plenary/distinguished lecturer for both SIAM and AMS and is associate editor for the SIAM journals *Multiscale Modelling and Simulation, Mathematical Analysis*, and *SIAM Review (Survey and Reviews)*. She also serves on the editorial board for *Interfaces and Free Boundaries* and *Applied Mathematics Research Express*. Her past honors include a Sloan Foundation Research Fellowship and the Presidential Career Award for Scientists and Engineers from the Office of Naval Research.

Selim Esedoglu received his Sc.B. in mathematics from Brown University in 1996. Subsequently, he got his M.S. and Ph.D. in mathematics from the Courant Institute, New York University, in 1998 and 2000 respectively. He was a postdoctoral researcher at the Institute for Mathematics and its Applications in Minneapolis between 2000 and 2002, and a CAM Assistant Professor at the mathematics department of UCLA between 2002 and 2005. Currently, he is an Assistant Professor of mathematics at the University of Michigan, Ann Arbor.

Alan Gillette received the B. A. degree in Mathematics and Physics in 1992 from Washington University in St. Louis. From 1993 to 1995, he was employed as Engineer for Alcotek, Inc. of St. Louis. Between 1995 and 1999, he served as Engineer and later as Program Manager for Cemax-Icon, which was a subsidiary of the Eastman Kodak Company. Since 2003, he has served as a Research Assistant in the Applied Mathematics Department, University of California, Los Angeles, where he is currently writing his Ph.D. thesis on image inpainting.