[2] L. Oisel, E. Memin, L. Morin, and L. F. Galpin, "One-dimensional dense disparity estimation for three-dimensional reconstruction," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1107–1119, Sep. 2003.
[3] L. Ziyi and B. E. Shi, "Subpixel resolution binocular visual tracking using analog VLSI vision sensors," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 47, no. 12, pp. 1468–1475, Dec. 2000.
[4] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis, "Face localization and authentication using color and depth images," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 152–168, Feb. 2005.
[5] G. C. DeAngelis, I. Ohzawa, and R. D. Freeman, "Depth is encoded in the visual cortex by a specialized receptive field structure," *Nature*, vol. 11,352, no. 6331, pp. 156–159, 1991.
[6] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.
[7] H. B. Barlow, C. Blakemore, and J. D. Pettigrew, "The neural mechanism of binocular depth discrimination," *J. Physiol.*, vol. 193, pp. 327–342, 1967.
[8] G. C. DeAngelis, B. G. Cumming, and W. T. Newsome, "Cortical area MT and the perception of stereoscopic depth," *Nature*, vol. 394, pp. 677–680, 1998.
[9] G. F. Poggio and T. Poggio, "The analysis of stereopsis," *Annu. Rev. Neurosci.*, vol. 7, pp. 379–412, 1984.
[10] B. A. Draper, J. R. Beveridge, A. P. W. Bohm, C. Ross, and M. Chawathe, "Accelerated image processing on FPGAS," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1543–1551, Dec. 2003.
[11] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 993–1008, Aug. 2003.
[12] D. J. Fleet and A. D. Jepson, "Stability of phase information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1253–1268, Dec. 1993.
[13] B. Porr, B. Nürenberg, and F. A. Wörgöter, "VLSI-Compatible computer vision algorithm for stereoscopic depth analysis in real-time," *Int. J. Comput. Vis.*, vol. 49, no. 1, pp. 39–55, 2002.
[14] A. Darabiha, J. Rose, and W. J. MacLean, "Video-Rate stereo depth measurement on programmable hardware," presented at the Int. Conf. Computer Vision and Pattern Recognition, Madison, WI, Jun. 2003.
[15] F. Solari, S. P. Sabatini, and G. M. Bisio, "Fast technique for phase-based disparity estimation with no explicit calculation of phase," *Elect. Lett.*, vol. 37, no. 23, pp. 1382–1383, 2001.
[16] A. Cozzi, B. Crespi, F. Valentinotti, and F. Wörgötter, "Performance of phase-based algorithms for disparity estimation," *Mach. Vis. Appl.*, vol. 9, pp. 334–340, 1997.
[17] Handel-C Language Reference Manual, Celoxica Company, 2003.
[18] Celoxica Company. [Online]. Available: http://www.celoxica.com
[19] Xilinx Company. [Online]. Available: http://www.xilinx.com
[20] J. Díaz, E. Ros, S. Mota, E. M. Ortigosa, and B. del Pino, "High performance stereo computation architecture," in *Proc. Int. Conf. Field Prog. Logic and Applications*, Tampere, Finland, Aug. 24–26, 2005, pp. 463–468.
[21] J. Valls, M. Kuhlmann, and K. K. Parhi, "Evaluation of CORDIC algorithms for FPGA design," *J. VLSI Signal Process.*, vol. 3, pp. 207–222, Nov. 2002.
[22] M. Gong and R. Yang, "Image-gradient-guided real-time stereo on graphics hardware," in *Proc. 5th Int. Conf. 3-D Digital Imaging and Modeling*, Ottawa, ON, Canada, Jun. 13–16, 2005, pp. 548–555.
[23] S. Forstmann, S. Thüring, Y. Kanou, J. Ohya, and A. Schmitt, "Real time stereo by using dynamic programming," in *Proc. Conf. Computer Vision and Pattern Recognition*, Washington, DC, Jul. 02, 2004, vol. 3, pp. 29–29.
[24] H. Niitsuma and T. Maruyama, "Real-time detection of moving objects," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2004, vol. 3203.
[25] J. Woodfill and B. V. Herzen, "Real-time stereo vision on the parts reconfigurable computer," in *Proc. Conf. IEEE Symp. Field-Programmable Custom Computing Machines*, Napa, CA, Apr. 1997, pp. 201–201.
[26] T. Kanade, "Development of a video-rate stereo machine," in *Proc. ARPA Image Understanding Workshop*, Nov. 1994, pp. 549–558.

Inpainting of Binary Images Using the Cahn–Hilliard Equation

Andrea L. Bertozzi, Selim Esedoḡlu, 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 binary 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*—Binary images, Cahn–Hilliard equation, image inpainting, super-resolution.**

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 imitates the techniques of museum artists who specialize in restoration. They focused on the principle that good inpainting algorithms should propagate sharp edges 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 extend continuously into the interior. 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 method in [1] has intimate connections with 2-D fluid dynamics through the Navier–Stokes equation. Indeed, the steady-state equation 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 NavierStokes formulation, introducing a diffusion term. This analogy also shows why diffusion is required in the original inpainting problem. In practice, nonlinear diffusion [4], [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 adapted to the inpainting task by a simple modification. 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,

Manuscript received December 9, 2005 ; revised June 30, 2006. This work was supported in part by the National Science Foundation (NSF) and the intelligence community through the joint "Approaches to Combat Terrorism" program under Grant AST-0442037, in part by the Office of Naval Research under Grant N000140410078, and in part by NSF Grants ACI-0321917 and DMS-0410085. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thierry Blu.

A. L. Bertozzi and A. Gillette are with the Mathematics Department, University of California, Los Angeles, CA 90095-1555 USA (e-mail: bertozzi@math.ucla.edu; agillett@math.ucla.edu).

S. Esedoḡlu is with the Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1043 USA (e-mail: esedoglu@umich.edu).

Digital Object Identifier 10.1109/TIP.2006.887728

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 builtin notion of what constitutes a natural image. The first model introduced by Chan and Shen used the total variation based image denoising model [5]. 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.

Subsequently, Chan *et al.* [7] 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 *et al.*[8] 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, Esedoḡlu and Shen [9] adapted the Mumford–Shah image segmentation model [10] to the inpainting problem for grayscale images. They utilized Ambrosio and Tortorelli's elliptic approximations [11] 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, Esedoḡlu and Shen introduced the Mumford–Shah–Euler (MSE) image model that, just like the previous work of Kang *et al.* [7], penalizes the square of the curvature along an edge contour. Following previous work by March [12], they then used a conjecture of De Giorgi [13] 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 [14] 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 Ginzburg–Landau equation then leads to a coupled system to be solved for u and v, respectively.

All of the above methods are PDE-based methods. With regard to other kinds of methods for binary inpainting, the closest in spirit are those based on spline continuation of the edges [15], [16]. This approach is very fast for simple regions and the complexity of the algorithm depends on the number of edges involved in the inpainting problem. In contrast, our PDE-based method is $O(NlogN)$, regardless of the complexity of the image, where $N = n \times n$ is the number of pixels in a square region surrounding the inpainting domain. The log correction is due to the use of the FFT for solving an implicit equation.

II. 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 [9], 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:

$$u_t = -\nabla^2 \left(\varepsilon \nabla^2 u - \frac{1}{\varepsilon} W'(u) \right) + \lambda(\vec{x})(f - u) \tag{1}$$

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 (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* [17], [18]. Convexity splitting decomposes the energy functional into two parts—a convex energy plus a concave energy. Then one evolves the gradient flow for the Euler–Lagrange equation using a semi-implicit timestep in which the convex part of the energy is implicit and the concave part explicit. Under the right conditions, this convexity splitting results in an unconditionally stable time-discretization scheme, allowing for arbitrarily large time steps. Vollmayr-Lee and Rutenberg [18] have recently refined the conditions under which stability is applicable for the original Cahn–Hilliard equation.

The new modified Cahn–Hilliard equation is not strictly a gradient flow. The original Cahn–Hilliard 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 (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. For a discussion of Cahn–Hilliard and gradient flows in H^{-1}, see Taylor and Cahn [19]. However, the idea of convexity splitting, one for the Cahn–Hilliard energy in (2) and one for the energy E_2 in (3), can still be applied to this problem with good results.

For example, one can split $E_1 = E_{11} - E_{12}$ where

$$E_{11} = \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{C_1}{2}|u|^2 \, d\vec{x} \quad \text{and} \tag{4}$$

$$E_{12} = \int_\Omega -\frac{1}{\varepsilon}W(u) + \frac{C_1}{2}|u|^2 \, d\vec{x}. \tag{5}$$

Note that $-E_{12}$ is not strictly concave; however, in practice, one can choose C_1 so that it is concave over the range of values of u empirically observed in the simulation. Likewise, we can rewrite $E_2 = E_{21} - E_{22}$ where

$$E_{21} = \int_{\Omega\setminus D} \frac{C_2}{2}|u|^2 \, d\vec{x} \quad \text{and} \tag{6}$$

$$E_{22} = \int_{\Omega\setminus D} -\lambda_0(f-u)^2 + \frac{C_2}{2}|u|^2 \, d\vec{x}. \tag{7}$$

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

$$\frac{u^{n+1}-u^n}{\Delta t} = -\nabla_{H^{-1}}(E_{11}^{n+1} - E_{12}^n) - \nabla_{L^2}(E_{21}^{n+1} - E_{22}^n) \tag{8}$$

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

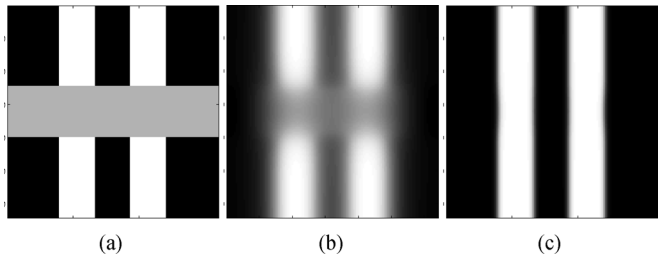
$$\frac{u^{n+1}-u^n}{\Delta t} + \varepsilon\nabla^4 u^{n+1} - C_1\nabla^2 u^{n+1} + C_2 u^{n+1} = \nabla^2\left(\frac{1}{\varepsilon}W'(u^n)\right) + \lambda(\vec{x})(f(\vec{x}) - u^n) - C_1\nabla^2 u^n + C_2 u^n. \tag{9}$$

The constants C_1 and C_2 are positive; they must be large enough so that E_{11}, E_{12}, E_{21}, and E_{22} are convex for the range of u in the simulation. In practice, C_1 is comparable to $1/\varepsilon$, while C_2 is comparable to λ_0. Numerical tests show that with these choices the scheme (9) is unconditionally stable. Equation (9) for u^{n+1} then involves only constants and the Laplace operator applied to the new time level. On a square domain, with periodic or Neumann boundary conditions, one can solve this efficiently using a pseudospectral method based on a fast Fourier transform [20], [21]. The main idea is that the operators applied to u^{n+1} have the same eigenfunctions as the FFT and, thus, are diagonalizable using this decomposition. We present some examples below and state the parameters used for Δt, C_i, λ, and ε.

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.

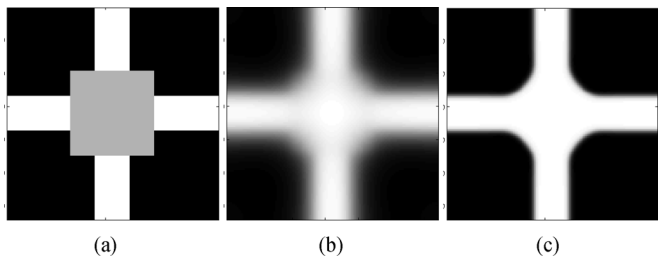
III. Examples

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. We show several examples performed on a Linux desktop system using a Pentium 4 processor, and programmed in MATLAB.



(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 128×128). Total CPU time 11.5 s. $\Delta t = 1$, $\lambda = 50\,000$, $C_1 = 300$, and $C_2 = 150\,000$.



(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 128×128, stripe width is 20 units, initial gap distance is 50 units). Total CPU time 15.6 s. $\Delta t = 1$, $\lambda = 100\,000$, $C_1 = 300$, and $C_2 = 3\lambda$.

A. Inpainting of a Double Stripe

Fig. 1 shows a two-step process. The gray region in Fig. 1(a) denotes the inpainting region. We begin with ε (= .8), and at $t = 50$ we reach a steady state, shown in Fig. 1(b). We then switch to a small value of ε (= .01), using the result from Fig. 1(b) as initial data. The final result at $t = 700$ and is shown in Fig. 1(c).

B. Inpainting of a Cross

In Fig. 2(a), the gray region denotes the inpainting region. As with the stripes, the modified Cahn–Hilliard equation is run to steady state for a large value of ε (= .8), resulting in Fig. 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 $t = 1000$.

C. Inpainting of a Sine Wave

Fig. 3 shows how the modified Cahn–Hilliard equation may be applied to the inpainting of simple road-like structures. In Fig. 3(a), an incomplete sine wave is shown. In Fig. 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 for effective inpainting. In Fig. 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 (1). The two-step method was then used to inpaint the

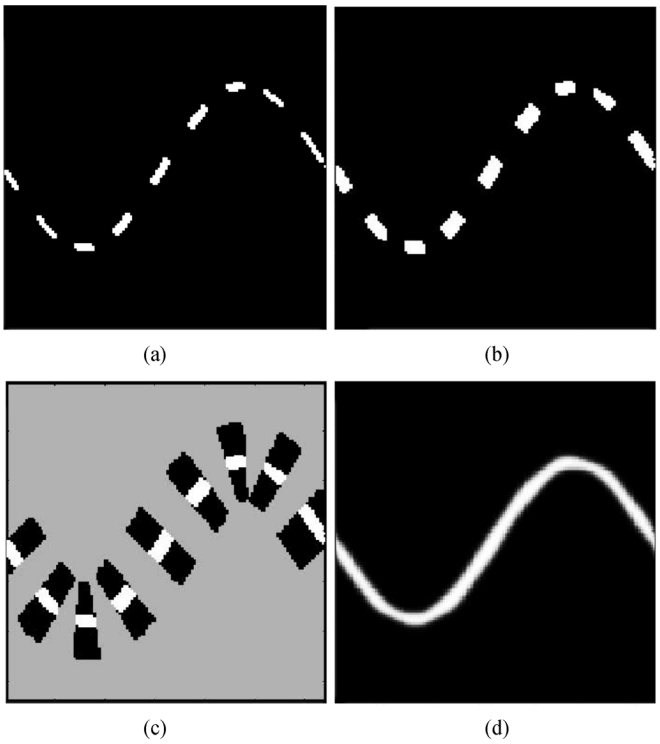


Fig. 3. Inpainting a sine wave. (a) Initial data. (b) Expansion of white points by factor of 3. (c) Gray area denotes inpainting region (d) Final result at $t = 4000$, CPU time of 35.1 s (image domain is 128×128). $\Delta t = 1$, $C_1 = 300$, and $C_2 = 3\lambda$.

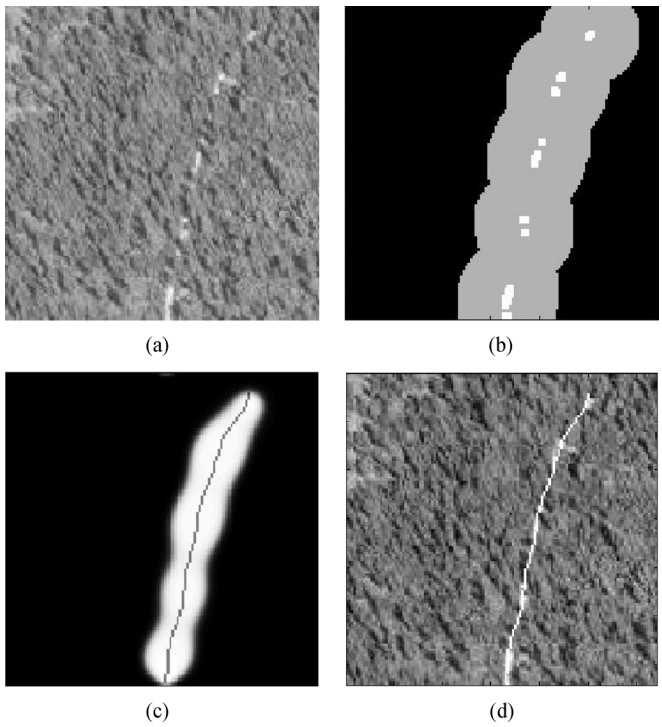


Fig. 4. Inpainting of an obscured road. (a) Original photograph. (b) Thresholding for road pixels, expansion of road pixels by factor of 3. Gray inpainting region is the union of circles of radius R centered on the road pixels where R is the maximum gap between consecutive road pixels. (c) Inpainting result at $t = 500$, 8.5 s of CPU time. $\Delta t = 1$, $\lambda = 1\,000\,000$, $C_1 = 30\,000$, and $C_2 = 3\lambda$. (d) Centerline of inpainting region, superimposed on original photograph (image domain is 128×128).

sine wave. Fig. 3(d) shows the final 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 $t = 4000$.

D. Inpainting of a Road

Fig. 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 Fig. 4(b). The gray area in Fig. 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 that each thresholded white point of the road has been expanded in radius, as for the sine wave in Fig. 3(b). In 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 Fig. 4(c), the 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 the resulting centerline, overlaid on the initial satellite photo, shows, in Fig. 4(d), a close an estimation of the path when overlayed on the original photograph. The initial value for ε was.8 and was switched at $t = 100$ to $\varepsilon = .005$. The final result was taken at $t = 500$. Much more efficient inpainting, as in in Fig. 3, could be done with a more precise estimation of the inpainting region.

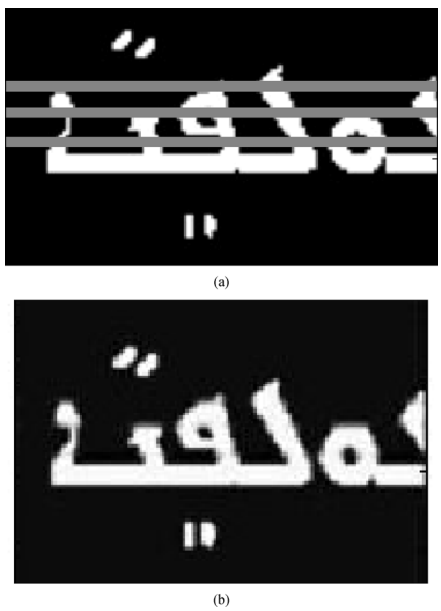
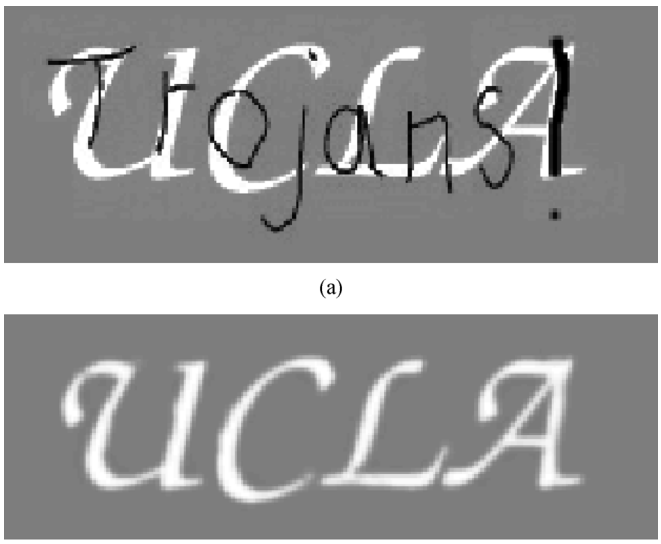


Fig. 5. Recovery of damaged text. (Image domain was 256×256). $\Delta t = 1$, $\lambda = 50\,000\,000$, $C_1 = 10\,000$, $C_2 = 3\lambda$, CPU time is 22.4 s at $t = 200$, shown above.

E. Recovery of Text

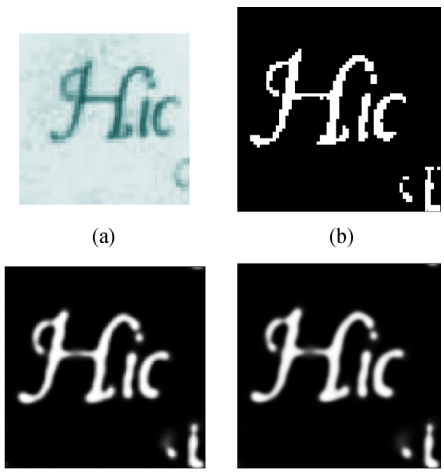
In Fig. 5(a), several lines obscure some Arabic writing. Using these obstructing lines as the inpainting region, the modified Cahn–Hilliard two-step scheme can inpaint the occluded parts of the writing. The initial value for ε was.08. At $t = 100$, ε was switched to.01. The program was then run to 200 time steps and the result is shown in Fig. 5(b).

In Fig. 6(a), graffiti is written over the UCLA logo. Using the graffiti as the inpainting region, We restore the image using the two-step method. Until $t = 100$, a large value of ε ($= .8$) is used. At $t = 100$, ε is switched to a small value ($= .003$). The final result in Fig. 6(b) is the restored logo.



(a) (b)

Fig. 6. Recovery of damaged text. (Image domain is 256×256). $\Delta t = 1$, $\lambda = 50\,000\,000$, $C_1 = 15\,000$, $C_2 = 3\lambda$, CPU time is 13.25 s at $t = 120$, shown above.



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

Fig. 7. Super-resolution of text. Magnification $3\times$. Original size 64×64. Total CPU time is 17.4 s. $C_1 = 300$, $C_2 = 150\,000\,000$, $\Delta t = 1$.

F. Super-Resolution

The modified Cahn–Hilliard equation can also be used for the purposes of super-resolution of text. This application has been studied by other PDE-based methods [3], [22], as well as by many other classes of algorithms (e.g., [23]), and, thus, it is natural to consider here. Latin writing is shown in Fig. 7(a), of size 64×64. Fig. 7(b) shows the text enlarged by $3\times$ using MATLAB's "nearest-neighbor" algorithm. First, the white region of Fig. 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 [Fig. 7(b)]. Fig. 7(c) and (d) show the results at $t = 350$ and $t = 450$, respectively.

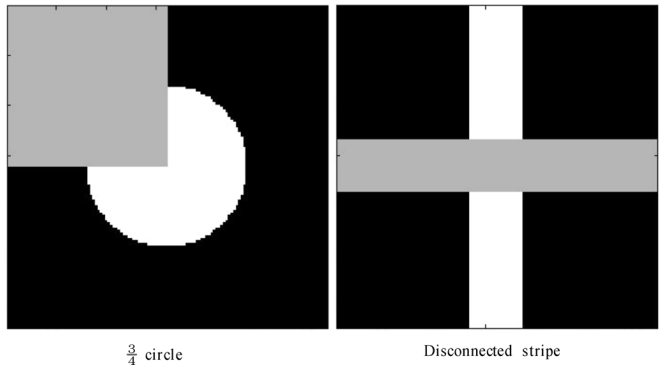


Fig. 8. Inpainting data for comparison tests. Gray color denotes inpainting regions.

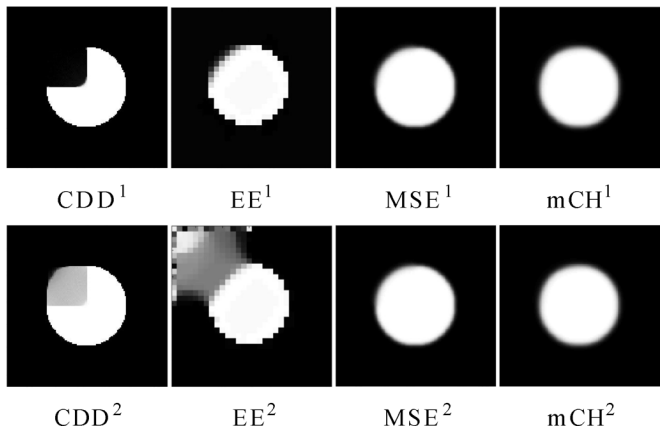


Fig. 9. Results for the circle inpainting test. ^{1} – zero initial data assumed in inpainting region. ^{2} – random initial data assumed in inpainting region.

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 PDE 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 [24], the Euler's elastica (EE) inpainting model of Chan *et al.* [7], and the MSE inpainting model of Esedoğlu and Shen [9]. 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++).

A. Graphic Results

Figs. 9 and 10 show the performance of each inpainting method on the circle and stripe tests, respectively. As can be seen in Fig. 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). 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 ($\mathrm{MSE}^{1,2}$, $\mathrm{mCH}^{1,2}$).

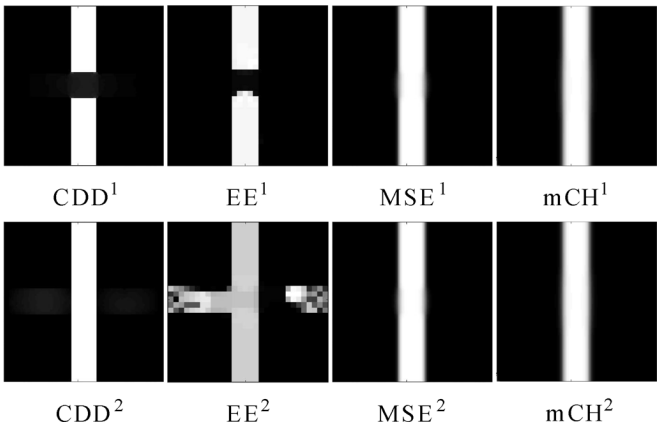


Fig. 10. Results for the stripe inpainting test. [1] – zero initial data assumed in inpainting region. [2] – random initial data assumed in inpainting region.

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.

* 30×30 grid used. All others 128×128.

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.

* 30×30 grid used. All others 128×128.

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 Figs. 9 and 10.

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 works best when the end-user to specifies the inpainting domain, this method can also be used for interpolating simple roads and other situations where a user-defined inpainting region is not feasible. Through a two-step process, the method can inpaint across large gaps in a repeatable way. Although multiple solutions, including broken connections, may be possible mathematically, the method can find a continuous solution by first performing very diffuse but continuous connection, 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 PDE-based inpainting methods. Fast numerical techniques available for the Cahn–Hilliard equation also allow for efficient computation with relatively large datasets.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. Siggraph Computer Graphics*, K. Akeley, Ed., 2000, pp. 417–424.

[2] S. Masnou and J. Morel, "Level lines based disocclusion," presented at the 5th IEEE Int. Conf. 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," in *Proc. IEEE Computer Vision and Pattern Recognition*, Dec. 2001, vol. 1, pp. I355–I362.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

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

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

[7] T. Chan, S. Kang, and J. Shen, "Euler's elastica and curvature-based inpaintings," *SIAM J. Appl. Math.*, vol. 63, no. 2, pp. 564–592, 2002.

[8] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*. New York: Springer-Verlag, 1993, vol. 662, Lecture Notes in Computer Science.

[9] S. Esedoğlu and J. Shen, "Digital inpainting based on the Mumford–Shah–Euler image model," *Eur. J. Appl. Math.*, vol. 13, pp. 353–370, 2002.

[10] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.

[11] L. Ambrosio and V. M. Tortorelli, "Approximation of functionals depending on jumps by elliptic functionals via gamma convergence," *Commun. Pure Appl. Math.*, vol. 43, pp. 999–1036, 1990.

[12] R. March and M. Dozio, "A variational method for the recovery of smooth boundaries," *Image Vis. Comput.*, vol. 15, no. 9, pp. 705–712, 1997.

[13] 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. New York: Birkhauser, 1991, pp. 135–142.

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

[15] G. Schuster, X. Li, and A. K. Katsaggelos, "Spline-based boundary loss concealment," presented at the IEEE Int. Conf. Image Processing, 2003.

[16] L. D. Soares and F. Pereira, "Spatial shape error concealment for object-based image and video coding," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 586–599, Apr. 2004.

[17] D. Eyre, *An Unconditionally Stable One-Step Scheme for Gradient Systems*, Jun. 1998, unpublished.

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

[19] J. Taylor and J. Cahn, "Linking anisotropic sharp and diffuse surface motion laws via gradient flows," *J. Statist. Phys.*, vol. 77, pp. 183–197, 1994.

[20] P. Smereka, "Semi-implicit level set methods for curvature flow and for motion by surface diffusion," *J. Sci. Comput.*, vol. 19, pp. 439–456, 2003.

[21] K. Glasner, "A diffuse interface approach to hele-shaw flow," *Nonlinearity*, vol. 16, pp. 1–18, 2003.

[22] B. D. Morse and D. Schwartzwald, "Image magnification using level-set reconstruction," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 333–340, 2001.

[23] G. Dalley, B. Freeman, and J. Marks, "Single-frame text super-resolution: A Baysian approach," in *Proc. Int. Conf. Image Processing*, 2004, pp. 3295–3298.

[24] T. F. Chan and J. Shen, "Non-texture inpainting by curvature-driven diffusions," *J. Vis. Commun. Image Represent.*, vol. 12, no. 4, pp. 436–449, 2001.