IMAGE INPAINTING-BASED EDGE ENHANCEMENT USING THE EIKONAL EQUATION

Zhaozhong Wang
Image Processing Center, Beihang University
Beijing 100191, China

ABSTRACT
This paper proposes a novel application of image inpainting techniques for the edge enhancement problems in image deblurring and denoising. The edge enhancement effect is achieved by the jumps of pixel values at edge locations resulted from an inpainting process. The process is formulated by the eikonal partial differential equation (PDE) to rule the inpainting priority of pixels in automatically erased regions. The equation is then numerically solved by the fast marching method. A solution of the Laplace’s equation is also embedded in the numerical scheme to assure the smoothness in non-edge locations. Experiments are performed and comparisons are made with other PDE-based enhancement methods to demonstrate the merit of the proposed algorithm.

Index Terms— Image inpainting, edge enhancement, eikonal equation, Laplace’s equation, fast marching method.

1. INTRODUCTION
Image inpainting is a method to fill image regions [1], typically used to repair damaged parts or remove specific objects. This paper suggests a novel application of the inpainting method for image edge enhancement based on a technique of the eikonal partial differential equation (PDE). This work is motivated by observing a defect of existing inpainting algorithms: they might result in jumps of pixel values in the middle of an inpainted region, since pixel values were generally propagated from the region’s boundary to its middle [1] with some degree of error. This work, however, will convert the defect to a merit by making the jumps coincide with image edges, so as to enhance blurry edges.

Compared with typical inpainting methods, this work will show the following features: First, it extends the application of inpainting from simply object removal to edge enhancement in deblurring and denoising problems. Second, it automatically determines which regions should be inpainted, while most of other methods have to inpaint manually masked regions. Finally, it uses the eikonal equation approach, which is seldom used in other PDE-based inpainting methods. There are applications of inpainting techniques [2] to problems like super-resolution reconstruction based on the total variation (TV) model [3], which results in PDEs different from the eikonal equation. The work in this paper may be viewed as a generalization and deeper formulation of our previous work [4] for image magnification using watershed-guided inpainting, since the continuous watershed transformation [5] may be modelled by the eikonal PDE.

There are a number of image enhancement algorithms, wherein the PDE-based algorithms include anisotropic diffusion [6, 7, 8], shock filters [9, 10] and those combining diffusion and shock filters [11]. The well-known bilateral filter [12] is also related to the PDE approach of anisotropic diffusion [13]. Anisotropic diffusion generally cannot guarantee a rigorous edge discontinuity, since there exists a little degree of diffusion across edges. Shock filters assure the edge discontinuity in theory [9], but they might introduce unnatural effects such as jagged edges (see Section 4 for illustration). The proposed PDE method will generate more naturally enhanced images with smaller degree of jags, and theoretically guarantee the edge discontinuity in inpainted regions.

We shall formulate the math of our method in Section 2, using the eikonal equation to assure the edge discontinuity, followed by a Laplace’s equation to assure smoothness of non-edge locations. Section 3 proposes a numerical solution based on a modified fast marching method. Section 4 experimentally compares our algorithm with other PDE-based enhancement methods, and Section 5 concludes the work.

2. THE INPAINTING METHOD FOR EDGE ENHANCEMENT
We aim to apply an inpainting technique to recover sharp edges from blurry images. This technique may also be used as a post-processing step of image denoising for edge enhancement. Our method has the following steps: First it automatically erases a region containing edges; then it fills the region by propagating information from surrounding pixels; the propagated pixels finally meet together to finish the inpainting. Since we need to recover the discontinuity of pixel values at edge location, it is crucial that the propagation process stops at the edge location. This can be guaranteed by using the eikonal equation.

Let \( f : D \subset \mathbb{R}^2 \to \mathbb{R}, f \in C^2(D) \) be a 2D smooth image to be enhanced, and let the function \( h := ||\nabla f|| \) be
the gradient magnitude of $f$. Suppose that each connected erased region $\Omega$ contains only one edge. We define a function $T(x, y)$ as the arrival time function, which gives the inpainting priority of points in $\Omega$ as time goes by.

We then have the following eikonal equation with Dirichlet boundary condition

$$\|\nabla T(x, y)\| = \eta(x, y), \quad (x, y) \in \Omega,$$

$$T(x, y) = \eta_0(x, y), \quad (x, y) \in \partial \Omega,$$

where $\eta_0(x, y) := \|\nabla h(x, y)\|$. It is known [14] that this problem admits a unique local viscosity solution.

A trivial choice of the boundary function $\eta_0(x, y)$ is that $\eta_0(x, y) = 0$ for $(x, y) \in \partial \Omega$. But this may cause the propagated pixels meet each other at locations other than the edge. To avoid this problem, we set

$$\eta_0(x, y) = h(x, y), \quad (x, y) \in \partial \Omega,$$

where the gradient magnitude $h(x, y)$ is generally various with different $(x, y)$ on the boundary $\partial \Omega$, so the propagations started from different boundary pixels are asynchronously. This situation will be naturally handled in the numerical framework of fast marching method.

We define the set of points to be propagated at time $t$ as

$$C(t) = \{(x, y) : T(x, y) = t\}. \quad (3)$$

This set, typically a curve in 2D space, is known as the wave front. We shall inpaint points on the wave front $C(t)$ at time $t$ and make sure that the inpainted non-edge regions are smooth. Thus we use the following Laplace’s equation with Dirichlet boundary condition

$$\nabla^2 s(x, y) = 0, \quad (x, y) \in C(t) \subset \Omega, \quad t > 0,$$

$$s(x, y) = f(x, y), \quad (x, y) \in \partial \Omega,$$

to solve the enhanced 2D image $s : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, which is noted to be a generalized solution since its derivatives do not exist at discontinuity locations.

3. THE NUMERICAL ALGORITHM

There are several algorithms to solve the eikonal equation (1), see the Ref. [15] for a review and comparison, which indicates that the fast marching method (FMM) [16] is preferred. The FMM is also used for image inpainting [17] to maintain continuous isophotes, but our algorithm recovers discontinuous edges and is different from that work.

The 2D numerical approximation of $\|\nabla T\|$ in Eq. (1) that selects the physically correct viscosity solution is given by the following upwind scheme [18]:

$$\max(D^{-x}_{ij}T, -D^{-y}_{ij}T, 0)^2 + \max(D^{+x}_{ij}T, -D^{+y}_{ij}T, 0)^2 = \eta^2_{ij},$$

where $D^{-x}_{ij}$ and $D^{+x}_{ij}$ are respectively the standard backward and forward finite difference schemes at location $(i, j) \in \mathbb{Z}^2$, which is the discretized grid coordinate of the continuous coordinate $(x, y)$. Using first-order finite difference approximations of $\nabla T$ in the $x$-direction

$$D^{-x}_{ij}T = \frac{T_{ij} - T_{i-1,j}}{\Delta x}, \quad D^{+x}_{ij}T = \frac{T_{i+1,j} - T_{ij}}{\Delta x},$$

and similar formulae for the differences in the $y$-direction, Eq. (5) can be written as

$$\sum_{i=1}^{2} \max\left(\frac{T_{ij} - T_{i,j}}{\Delta t}, 0\right)^2 = \eta^2_{ij},$$

where $\Delta_1 = \Delta x$, $\Delta_2 = \Delta y$ are sampling intervals, and

$$T_1 = \min(T_{i-1,j}, T_{i+1,j}), \quad T_2 = \min(T_{i,j-1}, T_{i,j+1}).$$

Eq. (6) can then be easily solved for $T_{ij}$ by algebraic methods. Image gradient used in the computation may be numerically approximated by the central finite difference.

Before describing the FMM to solve the eikonal equation, we should give a numerical scheme for the Laplace’s equation (4), which will be embedded in the FMM. The Laplace’s equation is elliptic and is typically approximated based upon the central difference. By discretizing Eq. (4), its numerical scheme [19] is given as

$$s_{ij} = \frac{1}{4} \left(s_{i+1,j} + s_{i-1,j} + s_{i,j-1} + s_{i,j+1}\right).$$

This equation is generally solved by iteration schemes to update $s_{ij}$ and its neighboring points. To speed up the computation, however, we shall solve the equation without iteration: For a point $s_{ij}$ on the current wave front, we find in its nearest neighbors the points that have been inpainted (tagged as known), then assign to $s_{ij}$ the average value of these inpainted point values. Such a one-pass assignment will not introduce much error since the erased regions among edges are typically thin. In addition, this scheme naturally cooperates with the FMM to form the following compact algorithm:

Initially, all boundary points of each $\Omega$ are tagged as known, whose nearest neighbors in $\Omega$ are computed for the arrival time using Eq. (6) and are tagged as narrow band; other points in $\Omega$ are tagged as far. Then

1. Find the point with minimal arrival time among all narrow-band points, and change its tag to known.
2. Compute its pixel value by averaging its nearest known neighbors’ pixel value.
3. Find its nearest neighbors that are either far or narrow band.
4. Update their arrival times by solving Eq. (6), and tag the far points as narrow band.
5. Go back to Step 1, until the narrow band is empty.
4. EXPERIMENTS AND DISCUSSIONS

To implement the above algorithm, we should first specify methods to automatically erase regions for inpainting. Here we simply detect the erased regions by thresholding the gradient magnitude image \( h(x, y) \), see Fig. 2 for an illustration. Out tests validate that the inpainting algorithm is robust to the thresholding as long as each connected erased region contains only one edge. Other methods to detect the erased regions, such as the dilation of Canny edge [4] can be used as well.

We show experimental results for 1D signal denoising as in Fig. 1. The 1D signal is actually the profile of a scan-line in a 2D image which is enhanced by the 2D algorithm in Section 3. Of course we can simplify the algorithm to process 1D signal directly. Next we show results of 2D images in Figs. 2 and 3 for deblurring and denoising, respectively. For noisy images we first smooth them using an isotropic Gaussian filter, then enhance edges using the proposed algorithm.

Since our algorithm belongs to the class of PDE-based enhancement methods, we compare it with other PDEs, including the edge-enhancement anisotropic diffusion [8] and the complex shock filter [10], see Figs. 2 and 3. The anisotropic diffusion performs relatively weak for enhancement, since it still blurs the pixels across edges in a small degree. The complex shock filter outperforms in robustness than other shock filters such as that in [9], but its enhancement may cause unnatural effects, e.g., in the regions of the top head and the neck of the parrot in Fig. 2(c). The shock filter may also generate jagged edges as illustrated in Fig. 3(c).

Using our method the edges in the erased regions are enhanced sharply, and the entire image looks more natural. Our method also results in some jagged edges, but in a lower degree than the shock filter. Moreover, the proposed algorithm removes artifacts at image boundary, see Figs. 2(c) and (d) for comparison. A drawback of our algorithm is its sensitivity to perturbations in gradient magnitude \( h(x, y) \), which may cause the sharpened edges biased from their desired locations. Table 1 lists the results for a set of images, where the SNR (signal-noise ratio) provides an objective measure, though it does not well reflect subjective judgements.

Both diffusion and shock filters need multiple iteration steps; our algorithm needs only one-pass of computation. In addition, diffusion and shock filters process entire images, but our algorithm only enhances parts of image in erased regions. Specifically, diffusion and shock filters typically have the computational complexity of \( O(kn) \) [9], where \( k \) is the number of iterations and \( n \) the total number of image points. Our algorithm has the computational complexity of \( O(m \log m) \) [16], where \( m \) is the number of points in erased regions, generally much less than the total point number \( n \).

In the above experiments, the iterations for both the diffusion and shock filters are set as \( k = 30 \), and the image size in both Figs. 2 and 3 are of \( n = 200 \times 200 \). For our algorithm, \( m \approx 0.13n \) in Fig. 2 and \( m \approx 0.23n \) in Fig. 3. Using Matlab to test the two images, the diffusion filter consumes 1.26s and 1.31s, the shock filter 4.14s and 4.18s, and our algorithm 1.95s and 3.84s, respectively. Thus for images with general size the computational load of our algorithm is comparable to diffusion and shock filters, but does not show much speedup since the used scheme of FMM requires more numerical treatments in one step of iteration. Table 1 gives the averages of computational loads for more images.

5. CONCLUSION

This work proposes an extension of the original intention of image inpainting: from repairing damaged regions to enhancing blurred edges. The method uses the eikonal equation, a seldom used PDE in the inpainting community, to define the
The proposed algorithm is fast and with the complexity of $O(m \log m)$, where $m$ is the number of points to be inpainted, which is generally much less than the total number of image points. The enhancing performance of the method is comparable to and even better than shock filters and other PDE-based methods. The method automatically detects regions to be inpainted, and is robust to the variation of regions’ shape as long as each connected region contains only one edge. But the algorithm is sensitive to the perturbations in gradient magnitude, so a pre-filtering for the gradient may be required.

### 6. REFERENCES


