Photoshop Healing Brush: a Tool for Seamless Cloning

Todor Georgiev

Adobe Systems,
345 Park Avenue, San Jose, CA 94087, USA.

Abstract. The Healing Brush is a tool first introduced in Photoshop, that achieves seamless removal of defects in images. A similar image processing algorithm, called Poisson Editing, was later proposed in [8]. Our paper presents the theoretical ideas on which Healing is based, as well as some implementational details. Healing is performed by constructing iterative solutions to a fourth order partial differential equation. Its solutions follow the spatial variations between pixels in a sampled area, while at the same time are continuous and have continuous derivatives at the boundary. Poisson Editing is only continuous at the boundary. Also, our exact equation describes cloning of features from one image to another with greater fidelity. Our mathematical understanding of the process is based on viewing the image as a section in a fibred manifold, and minimizing certain natural expression for the energy written in terms of connections. In this we are, in a way, following the line of thought in Gauge Theories in Physics.

1 Introduction

Photographers often need to replace a damaged part of an image by a more 'healthy' looking patch, such as removing dust, scratches, wrinkles. For this purpose there is the Clone Tool in Photoshop and similar tools in other applications. The problem is that as a rule there is no area in the image with exactly the same texture and lighting so that replacement is seamless and natural. Doing this type of work was a difficult job, considered an art by photographers, and sophisticated techniques have been developed [1]. Photoshop 7 introduced the Healing Brush tool [2] as a solution to those problems. But algorithms for seamless 'fill in' had been independently addressed in the literature in different ways, which we will describe next.

1.1 Inpainting

A method called Inpainting [3] is using a combination of second and third order partial differential equation (PDE) for solving it.

This approach has two limitations. First, the reconstructed area is too smooth, which follows from the nature of the PDEs used. More recent methods [4] extract a texture component of the image and use it to fill in the texture component of the selected area, while reconstructing the image component based on PDEs. Second, for a low order PDE the reconstructed image in the selected area may not have smooth behavior at the boundary. In the simple case of second order PDE with Dirichlet boundary condition, there is discontinuity in the slope of the reconstructed function.

1.2 Texture Synthesis

This approach synthesizes texture in the selected area, based on some sort of 'surrounding' texture. The new image is created using appropriately selected individual pixels or an arrangement of many small patches directly taken from the sampled area [5], [6], [7].

Texture synthesis may not be the best approach for relatively smooth images with substantial change in color. There may not be a good area to get texture from – in the sense that it may be hard to achieve continuity if we rely only on fixed sampled pixel values. Also, it takes away human control of what's going on in reconstruction. A more interactive tool might give better chance to copy exactly the feature needed in the damaged area.

1.3 Poisson Editing

A recent paper [8] is much closer to the approach we defined with the Healing Brush in Photoshop 7 [2]. The authors achieve seamless cloning by solving a Poisson equation to fill in the selected area. The right hand side 'source' term in the equation is Laplacian of the sampled image. Dirichlet boundary conditions make the solution continuous at the boundary.

In our experiments with similar techniques we have discovered that continuity at the boundary is not always sufficient. Seamless fill in requires continuity of derivatives, if we insist on quality. One solution of the problem is a fourth order PDE, used for the Healing Brush.

2 Harmonic and Biharmonic Reconstructions

Everywhere in this paper $\triangle$ will denote the Laplacian.

Our first approach to Healing was to reconstruct the image in the defective area as a harmonic function, i.e. solution of the Laplace equation, with Dirichlet boundary conditions.

$$\triangle f = 0 \tag{1}$$

As mentioned above, this solution, even if continuous, has discontinuity in the derivatives at the boundary. Also, harmonic functions appear 'too stiff,' not flexible enough to follow gracefully variations in brightness/color.

We decided to fix this problem by using solutions of the Bilaplace equation with appropriate boundary conditions, known as biharmonic functions.

$$\triangle^2 f = 0 \tag{2}$$

Any harmonic function is at the same time biharmonic, so we are picking a solution from a proper superset of the previous (harmonic) set of functions, which provides us with the greater flexibility.

A simple way to solve Laplace equation in a given area with Dirichlet boundary conditions is to iterate with the following kernel (divided by 4):

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

We need to write only in the selected area, while sampling from outside the boundary is permitted when part of the kernel covers outside pixels. See [9] for faster numerical methods.

The Biharmonic equation can be easily solved by similar iterative methods. For example, the following kernel can be used (divided by 32):

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -2 & 8 & -2 & 0 \\ -1 & 8 & 12 & 8 & -1 \\ 0 & -2 & 8 & -2 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Just as with the Laplace equation, this kernel follows from a finite difference representation for the differential operator (in this case - the Bilaplacian). Conceptually, the idea is to let a diffusion type evolution take place until a stationary state is reached. When doing this we need to take care that the time derivative term comes with appropriate coefficient so that instabilities are avoided. Again, we write only in the selected area, while if needed (i.e. if the kernel is placed close to the boundary) we sample from outside.

Further extensions of this approach to higher order PDEs are possible and we have experimented with up to $\triangle^4 f = 0$. In general, higher order PDEs produce better reconstruction in terms of blending into the surroundings. Intuitively, higher order derivatives are matched at the boundary, together with the pixel value. This can be viewed as almost bringing in of texture from the outside, if you consider Taylor Series expansion of the function about a boundary point - to the inside, and assuming we match the outside derivatives up to certain order.

Another point of view on this effect is that higher order 'poliharmonic' functions are supersets of lower order, so that we have more freedom to choose and we can achieve better interpolation if we match boundary conditions.

For practical purposes, however, going to very high order PDEs is not reasonable because the iterative method gets very slow to converge. Even if results get better, they get much more expensive. That's why we decided to limit ourselves to Biharmonic functions for Photoshop.

3 Fibred Space Approach to Image Processing

As mentioned above, biharmonic and other reconstructions of defective areas in images are too smooth. We need to find appropriate PDEs that have non-smooth, rugged-looking solutions that resemble the fine scale structure of the image. To explain the general idea of the approach we have taken to this, we need to go to fundamentals of human vision.

It is a well known fact that our visual system does change the physical contents (the pixels) of the perceived image. Some of the effects of adaptation to color and lightness are well understood in the general framework of Land's Retinex theory [10]. The influence of the observer on the image (as perceived) probably can not be fully captured in one single theory. But it is clear that what we see in a picture is not simply pixel values. We see those values as transformed by interpretation. As J. Koenderink says in a recent paper [11], we should do image processing 'right' in terms of how images are perceived, taking into account the invariance of the image we see under a group of transformation of the physical image.

Image space (grayscale images) is obviously R^3, because we are dealing with pixels (with coordinates x, y) in the image plane, taking values in z. At this point, most people assume that different pixels can be compared by the value of z. As a general rule, this is wrong. Pixels with higher z do not always appear brighter. Brightness, or 'lightness' [10] depends on the surroundings and other vision-related factors, including higher level human understanding. Change of brightness is not simply change in pixel value. Comparison of pixels and the very concept of derivative needs to be redefined.

The traditional model of image space is Cartesian Product of the image plane and the positive real line of pixel values. This structure gives us two natural projections: For any point in image space we can immediately say which pixel it is, and what the pixel value is – according to the two components of the Cartesian product.

Above we argued that the second projection does not exist apriori. We do not compare pixels by pixel values. We need a model of image space in which changes (derivatives) can be defined independent of pixel values. We propose doing this by replacing the Cartesian product structure of image space with a Fibred Space structure (see also [11]).

3.1 Fibred Space

By definition, Fibred Space (E, π, B) consists of two spaces E and B, and a mapping π, called projection, of the first onto the second (which has lower dimension) [12]. See Figure 2. For each point $p \in B$ there is the so called fibre F_p in E, consisting of points that are sent to p by π (definition of fibre F_p). We can not compare two points from different fibres. This is related to the fact that π has no inverse. There is no distinguished mapping of B into E.

A section in a Fibred Space is a mapping f that sends points in B to E, and has the property $\pi(f(p)) = p$ for any $p \in B$. A grayscale image is a section

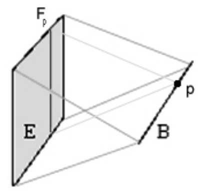

Fig. 1. Fibred space (E, π, B).

in a fibred image space (R^3, π, R^2). We see that apriori there is no projection onto z, no comparison between different pixels, no derivative. As a result, change and derivative at a point in image space is defined not by a vector in the image plane (x, y), as it is with functions, but by a vector in the total space (x, y, z). The direction of that vector is considered as direction of no change and called 'horizontal'.

3.2 Connections

In fibred spaces changes in the section are measured by a connection (instead of derivative). As the name says, connections show how we compare, or transfer pixel values from one fibre to another, in other words - how fibres are 'connected.' In Gauge theories [13], the simplest example of such a field is the Electromagnetic field. Connection ∇_i replaces the gradient ∂_i according to the simple expression

$$\nabla_i = \partial_i - A_i. \tag{3}$$

Any section g can be considered constant (horizontal) relative to appropriately chosen connection A_i such that $(\partial_i - A_i)g = 0$. The solution of this equation is:

$$A_i = \partial_i g/g. \tag{4}$$

For a simple example of the compensating role of connections, assume the zero connection on an image, $\nabla_i = \partial_i f$. If we introduce some shadow, or a change of lighting g, the image becomes gf. At the same time the visual system can potentially adapt to the shadow by using compensating field (4), so that $(\partial_i - A_i)gf = g\partial_i f$.

We see that now gradients change in the same way as the image itself under relighting. As a result, the equations in image processing using derivatives should be invariant under relighting. This is in line with [11] where a certain group of changes is proposed.

4 Healing

We refer to Healing as a method of reconstructing damaged areas by seamless cloning. To define Healing mathematically, let's first consider the energy expression for the Laplace equation:

$$\int (\partial_i f)(\partial_i f) dx dy \tag{5}$$

Minimizing this energy in the area of reconstruction with additional Dirichlet condition for the function at the boundary gives us the Laplace equation (1). In other words, Laplace equation is the Euler-Lagrange equation of an energy expression that is gradient squared of the image f. It answers the question: Which is the surface attached to the boundary, that has minimum sum of the rate of change squared?

4.1 Deriving the Equation

We know that rate of change in a fibred space needs to be defined in terms of a connection A_i. This suggests a new energy expression that should be used in the case of fibred spaces:

$$\int (\partial_i - A_i) f (\partial_i - A_i) f dx dy \tag{6}$$

The Euler-Lagrange equation for this energy is:

$$\triangle f = A^2 f + \partial_i A_i f. \tag{7}$$

Using the expression (horizontality condition) for the connection (4) that makes given area of the image constant, we get:

$$\triangle f = f \triangle g / g. \tag{8}$$

Assuming variations in the images are 'small' relative to the image itself, we can approximate f/g by a smooth function, or even a constant. We simply multiply the right hand side of Poisson equation by a constant which scales variations appropriately to achieve better fidelity. Also, our theoretical approach reveals the deeper meaning of the 'guidance' field in [8], which is related to the connection.

In this section we have derived the Laplace-Beltrami equation related to a connection extracted from another area of the image. The idea of using connections (covariant derivatives) with the appropriate energy expression can be extended to any PDE, second order or higher.

4.2 Healing Brush Algorithm

For the sake of speed and simplicity, we would ignore the details of the rigorous result, and define an easy Healing algorithm:

(1) Use as input the difference function $h_{in} = f - g$.
(2) Reconstruct h_{in} (solving PDE) using difference boundary conditions. Result of reconstruction is h_0.
(3) Define healed image h as $h = h_0 + g$.

The result h of this algorithm is a solution of $\triangle h = \triangle g$ with appropriate boundary conditions.

This algorithm is simpler and faster by solving Laplace instead of Poisson equation. Also, it can immediately be converted into similar Biharmonic or higher order algorithm by using Bilaplace (or other PDE) solver at step (2). As discussed above, it has better smoothness properties at the boundary.

We have implemented this Biharmonic algorithm for the Healing Brush in Photoshop.

4.3 Some Implementational Details

Since solving Bilaplace (and higher order) equation is much slower, we use a Laplace solver to build Harmonic solution as our first approximation. Then we iterate with our Biharmonic kernel starting from that solution, trying to modify it into a true Biharmonic function. Those iterations converge fast for pixels close to the boundary, but are still slow for the inside. The trick is that the inside is already constructed to our reasonable satisfaction, and we don't really need to change it much. The true reason why we had to go Biharmonic was - matching derivatives at the boundary. We can stop short of achieving true Biharmonic solution for the inside of big areas, as long as we are smooth at boundary and at least Harmonic inside.

By direct iterations, as described above, we build a solution of the Bilaplace equation that is smooth at the boundary. It matches pixel values of the inside to the outside, and is in a way similar to solving with Dirichlet boundary conditions, only extended to derivatives. This approach has problems when there is a very different color somewhere at the boundary. This color contaminates the reconstruction inside, and in most cases it is not the right thing to do.

This problem is solved by effectively dropping the Dirichlet continuity at the boundary, based on a new boundary condition similar to Neumann boundary condition. Here is a description in relation to the user interface.

This algorithm is active only when there is a selection in the image. Selection mask is used in Photoshop to define the area that needs to be modified. By definition all brushes write only inside the selection. With the Healing Brush there is something more than that. The user applies it close to the selection boundary and at least partly inside. Pixels inside the selection close to the boundary are treated differently from pixels deep inside:

There are two masks – the selection mask and the mask of the brush. These masks partly overlap. The central pixel for the kernel is covered by both masks. We use the Harmonic and Biharmonic kernel coefficients when reading pixels that are covered by the selection mask. When that non-central pixel is not covered by the selection mask, we do not read it. Instead, we use the pixel value of the central (for the kernel) pixel. The effect is similar to Neumann boundary condition: the reconstructed surface at the boundary is independent of the outside pixel values, and approaches a state of zero derivative normal to the boundary. This avoids contamination.

Masks have values between 0 and 1. Anything that is not 0 is considered selected by the Healing algorithm. In the end the result is interpolated with the old pixel value according to the mask. There are more details on how this is done, but these will be discussed elsewhere.

5 Conclusion and Future Work

The success of our approach in the case of the Healing Brush suggest the following theoretical idea. In certain cases of image processing, treating the image as a section in a fibred space is justified. Then we can use expressions for the energy based on connections (covariant derivatives) in relation to any PDE or other image processing algorithm, and this defines a wide area of research.

References

1. Eismann, K.: Photoshop Restoration and Retouching. Que, Indianopolis (2001)
2. Adobe Systems: Photoshop 7.0 User Guide. Adobe Systems, San Jose (2002)
3. Bertalmio, M., Sapiro, G., Castelles, V., Ballester, C.: Image Inpainting. In: Proceedings of ACM SIGGRAPH 2000, pp. 417-424, ACM Press (2000)
4. Bertalmio, M., Vese, L., Sapiro, G,. Osher, S.: Simultaneous Structure and Texture Image Inpainting. IEEE Transactions on Image Processing, 12(8), pp. 882-889 (2003)
5. Efros, A., Leung, T.: Texture Synthesis by Non-Parametric Sampling. In: Proc. Int. Conf. Computer Vision, pp. 1033-1038 (1999)
6. Wei, L., Levoy, M.: Fast Texture Synthesis using Tree-structured Vector Quantization. In: Proceedings of ACM SIGGRAPH 2000, pp. 479-488, ACM Press (2000)
7. Criminisi, A., Perez, P., Toyama, K.: Object Removal by Exemplar-Based Inpainting In: Proceedings of CVPR 2003, pp. 721-728 vol. 2 (2003)
8. Prez, P., Gangnet, M., Blake, A.: Poisson Image Editing. In: Proceedings of ACM SIGGRAPH 2003, pp. 313-318 (2003)
9. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. Cambridge University Press (1992).
10. Land, E., McCann, J.: Lightness and Retinex Theory. In: Journal of the Optical Society of America, v. 61, no. 1 pp. 1-7 (1971)
11. Koenderink, J., van Doorn, A.: Image Processing Done Right. In: Proceedings of ECCV 2002, pp. 158-172. Springer (2002)
12. Saunders, D.: The Geometry of Jet Bundles. Cambridge University Press (1989)
13. Choquet-Bruhat, Y., DeWitt-Morette, C.: Analysis, Manifolds and Physics. Amsterdam, North-Holland (1982)