



Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Fourier implementation of Poisson image editing

J.-M. Morel^a, A.B. Petro^{b,*}, C. Sbert^{b,1}^a CMLA, ENS Cachan, 61, avenue du Président Wilson, 94235 Cachan cedex, Paris, France^b Universitat de les Illes Balears, Crta. de Valldemossa km 7.5, 07122 Palma de Mallorca, Spain

ARTICLE INFO

Article history:

Received 20 December 2010

Available online 31 October 2011

Communicated by S. Wang

Keywords:

Image editing

Fourier transform

Poisson equation

Local contrast enhancement

ABSTRACT

Poisson editing, introduced in 2003, is becoming a technique with major applications in many different domains of image processing and computer graphics. This letter presents an exact and fast Fourier implementation of the Poisson editing equation proposed in (Pérez et al., 2003). The proposed algorithm can handle well all Poisson editing methods that are currently implemented with finite differences and multigrid methods. But it also authorizes fast complex editing strategies where the edited region is obtained by an algorithm instead of a manual selection. The selected region can therefore have a complex topology without additional computational cost. In this letter the proposed method is applied to a classic local contrast enhancement principle introduced in (Caselles et al., 1999). The manual selection of the dark regions is replaced by a lower threshold and the method becomes fast, efficient, level-line preserving, and interactive. The proposed method can be tried on line on any uploaded image at http://www.ipol.im/pub/demo/lmps_selective_contrast_adjustment/.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The concept of image editing encompasses the operations by which the local content of one or several images is selected and manipulated to create new syncretic images. The simplest such operation is a copy-paste of a part of an image into another. Image stitching, by which several images are fused into a panorama is another example (Levin et al., 2004). Local contrast or color adjustments after selecting manually a part of the image (for example the shadows) is another variant. The inpainting operation, by which an object is removed in the image and replaced by a texture is also a classic editing operation. For many examples of such image editing problems and surveys of the techniques we refer to Pérez et al. (2003), Wang and Cohen (2008), and Bertalmio et al. (2003).

Most image editing operations modify an input image I by manually selecting a region Ω in it. The task is to fill in this region by picking information from the rest of the image $I \setminus \Omega$, or from another image or, in the case of a selective contrast change, form the image in Ω itself. The most recent algorithms to solve this problem are based on partial differential equations. Two types of PDE, parabolic equations and elliptic equations, are the most

widely used. Image inpainting (Bertalmio et al., 2000) is based on parabolic equations. Matting operations (Sun et al., 2004) are based on the Poisson equation. The main challenge in image editing is to avoid suspicious color or texture alterations which would reveal the silhouette $\partial\Omega$ of the image region which has been modified (Chuan et al., 2008; Jia et al., 2006). As amply demonstrated by Pérez et al. (2003), the Poisson equation (1) is an extremely efficient response to this problem:

$$\begin{cases} \Delta u(x) = f(x) & \text{if } x \in \Omega, \\ u(x) = g(x) & \text{if } x \in \partial\Omega, \end{cases} \quad (1)$$

where $\Delta u(x, y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ is the Laplacian of u . The authors of Pérez et al. (2003) proposed to use (1) for “seamless editing”. They rely on psychophysical observations suggesting that our vision perceives the Laplacian of the images rather of the image itself. Thus the image u is satisfactorily specified by giving its Laplacian. On the other hand knowing the Laplacian inside the domain and the image value on the boundary is enough to reconstruct the image, and this is done by the Poisson equation. This work have been widely cited and various improvements have been presented (Levin et al., 2004; Jia et al., 2006; Chuan et al., 2008; Sun et al., 2004).

Although good results can be achieved using Poisson image editing, it is often a time-consuming process for the user because the region to be modified is selected manually and Poisson solvers becomes slow when the domain has a complex topology. The goal of this work is to accelerate the method by using only the Fourier method. There is a wide choice of methods to solve a linear system coming from a discrete elliptic equation. In the family of iterative methods the multigrid method dominate. Their main idea is to

* Corresponding author.

E-mail addresses: morel@cmla.ens-cachan.fr (J.-M. Morel), anablen.petro@uib.es (A.B. Petro), catalina.sbert@uib.es (C. Sbert).¹ The authors were supported by the Ministerio de Ciencia e Innovación under Grant TIN2008-04752, by MISS project of Centre National d'Etudes Spatiales, the Office of Naval Research under Grant N00014-97-1-0839 and by the European Research Council, advanced grant “Twelve labours”.

accelerate the convergence of basic iterative methods like SOR. Multigrid methods can treat arbitrary regions and boundary conditions, but are notoriously slow when the regions have not a simple geometry. Indeed, they proceed by a recursive subsampling of the domain Ω . This is only efficient when Ω is a rectangle with dyadic dimensions (or the union of a few rectangles with dyadic dimensions). Efficient solvers comparable to the multigrid method to solve the Poisson editing equations are examined in (Jeschke et al., 2009) where the authors present a Laplacian solver for minimal surfaces. In (Agarwala, 2007) the author proposes to use quad-trees to improve the efficiency of the technique that constructs seamless composites. Later solutions have also relied on the multigrid method: (Kazhdan and Hoppe, 2008; McCann and Pollard, 2008; Grady et al., 2005; Grady, 2008).

Our goal here is to show that a simple global reformulation of the Poisson editing problem permits to use fast Fourier solvers. A Fourier solver for the Poisson equation is exact, non-iterative and in practice faster than a full multigrid method, even in the case of a rectangular domain.

Since the proposed implementation method solves the Poisson equation over the whole image domain instead of only the unknown region Ω , it will be verified experimentally that the known pixel values are only minimally altered.

The proposed numerical method permits to automatize a local contrast enhancement proposed in (Pérez et al., 2003). The new tool is fast and efficient, requiring no manual selection and allowing one to edit regions with arbitrarily complex topology with no additional cost.

A review of Poisson image editing is presented in Section 2. Section 3 details the new implementation using the Fourier method. A number of possible choices for the guidance vector field is examined. These different choices with experimental results are presented in Section 4. Finally, Section 5 shows some examples of the correctness of the method.

2. Poisson image editing

In this Section, we detail the image Poisson reconstruction from “a guidance vector field”, in the terminology of Pérez et al. (2003). Let R , a closed subset of \mathbb{R}^2 , be the image domain, and let $\Omega \subset R$ be a closed subset with boundary $\partial\Omega$. The problem is to find the image whose gradient field is the closest, in L_2 -norm, to the prescribed “guidance vector field” \mathbf{v} defined in Ω , under given boundary conditions on $\partial\Omega$. The image is known in the rest of the image, $R \setminus \Omega$ (Fig. 1). This problem writes

$$\min_u \int_{\Omega} |\nabla u - \mathbf{v}|^2, \quad \text{with } u|_{\partial\Omega} = f|_{\partial\Omega} \tag{2}$$

and its solution is the unique solution of the Poisson equation with Dirichlet boundary conditions

$$\Delta u = \text{div } \mathbf{v}, \quad \text{with } u|_{\partial\Omega} = f|_{\partial\Omega}. \tag{3}$$

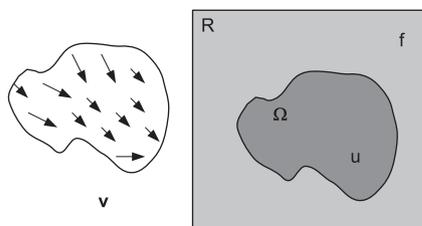


Fig. 1. The unknown function u interpolates f in domain Ω under a guidance vector field \mathbf{v} .

The variational problem (2) can be discretized using a finite difference approach. Using the notation of Pérez et al. (2003), it amounts to solve the following linear equations

$$|N_p|u_p - \sum_{q \in N_p \cap \Omega} u_q = \sum_{q \in N_p \cap \partial\Omega} f_q + \sum_{q \in N_p} v_{pq}, \quad \text{for all } p \in \Omega, \tag{4}$$

where N_p is the 4-connected neighborhood of p and $v_{pq} = \mathbf{v}(\frac{p+q}{2}) \cdot \overrightarrow{pq}$. Since the region Ω can have an arbitrary shape, the authors in (Pérez et al., 2003) propose to solve the linear system (4) using iterative solvers, for example the Gauss–Seidel iteration with overrelaxation or a multigrid method.

3. How to solve the Poisson image editing equations using Fourier transform?

The aim of this work is to solve the above variational problem (2), and, consequently the associated PDE (3), using the Fourier method. To do that the boundary of the domain Ω must coincide with the coordinate lines, which in general is not the case. This can be done by extending the definition domain of the guidance vector field to the whole image domain R , and solving the variational problem over all the whole image domain. The guidance vector field \mathbf{v} defined on Ω is extended to R by (Fig. 2)

$$\mathbf{V} = \begin{cases} \mathbf{v} & \text{over } \Omega \\ \nabla f & \text{otherwise,} \end{cases} \tag{5}$$

where f is the image, which is known over $R \setminus \Omega$. Then the problem is to minimize

$$\min_u \int_R |\nabla u - \mathbf{V}|^2. \tag{6}$$

The minimizer is uniquely determined by the Euler–Lagrange equation

$$\Delta u = \text{div } \mathbf{V}, \quad \text{over } R, \tag{7}$$

with homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{over } \partial R, \tag{8}$$

where \mathbf{n} is the direction orthogonal to the boundary.

By the above result the solution of the minimization problem (6) is the restriction of an even function which is periodic with a period twice the image size. The gradient field of the solution is the vector field that best approximates, in the L^2 -norm, the guidance vector field \mathbf{V} . In $R \setminus \Omega$ the gradient function approximates the gradient of f and in that sense the solution is only an approximation of f . Yet, their Laplacians coincide, which means that they differ only by a very smooth (harmonic) function. In Ω the gradient of the solution approximates the given guidance vector field defined in Ω .

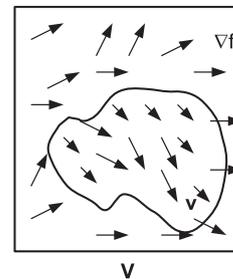


Fig. 2. The guidance vector field \mathbf{V} defined on the image domain.

3.1. Fourier transform method

Since the Eq. (7) has constant coefficients and the boundaries of R coincide with the coordinate lines, we can apply the Fourier transform method to solve the partial differential equation. The Neumann boundary condition is implicitly imposed by extending the original image symmetrically across its sides, so that the extended image, which is four times bigger, becomes symmetric and periodic. Once this is done, we can apply the Fourier transform.

The discrete Fourier transform (DFT) permits to compute directly the Fourier coefficients of a band limited and periodic function u from its samples u_{jl} on a $J \times L$ grid by

$$\widehat{u}_{mn} = \sum_{j=0}^{J-1} \sum_{l=0}^{L-1} u_{jl} e^{-\frac{2\pi ijm}{J}} e^{-\frac{2\pi iln}{L}} \quad (9)$$

with $m = 0, \dots, J-1$ and $n = 0, \dots, L-1$. The Fourier series formula recovers u from its Fourier coefficients by

$$u(x, y) = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \widehat{u}_{mn} e^{\frac{2\pi imx}{J}} e^{\frac{2\pi iny}{L}}. \quad (10)$$

Thus, by a simple differentiation, the Poisson equation $\Delta u = \text{div}(\mathbf{V})$ translates into a relationship between the Fourier coefficients of u and \mathbf{V} :

$$\left(\left(\frac{2\pi m}{J} \right)^2 + \left(\frac{2\pi n}{L} \right)^2 \right) \widehat{u}_{mn} = \frac{2\pi im}{J} \widehat{V}_{1mn} + \frac{2\pi in}{L} \widehat{V}_{2mn}, \quad (11)$$

where $\mathbf{V} = (V_1, V_2)$.

In short, the strategy for solving (7-8) by Fourier technique is:

- Quadruplicate by symmetry the discrete domain and \mathbf{V} .
- Compute the discrete Fourier transforms of V_1 and V_2 .
- Compute the discrete Fourier transform of the solution \widehat{u}_{mn} as

$$\widehat{u}_{mn} = \frac{\frac{2\pi im}{J} \widehat{V}_{1mn} + \frac{2\pi in}{L} \widehat{V}_{2mn}}{\left(\frac{2\pi m}{J} \right)^2 + \left(\frac{2\pi n}{L} \right)^2}. \quad (12)$$

- Obtain the samples u_{jl} of the solution by the inverse discrete Fourier transform.
- Restrict them to the initial domain.

Remark. The Fourier transform comes with some overhead such as padding to the nearest power of 2, but with smart Fourier libraries, like fftw, having integer values that are not powers of 2 is no more a complexity issue. Products of small factors are most efficient, but an $O(n \log n)$ algorithm is used even for prime sizes. Likewise, we mentioned for pedagogic reasons the quadruplication. In fact, again with smart libraries like fftw the quadruplication is implicit and performed directly as a cosine transform.

3.2. Computation of the vector field \mathbf{V}

The guidance vector field \mathbf{V} defined in (5) depends on the gradient of the original image or on the gradient of some source image. To compute the guidance vector field, multigrid methods use finite differences approximating the gradient of the image. Actually the Fourier method permits exact expressions for all derivatives. If the guidance vector field is the gradient of some function f , $\mathbf{V} = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$, then

$$\widehat{V}_{1mn} = -\frac{2\pi im}{J} \widehat{f}_{mn}, \quad \widehat{V}_{2mn} = -\frac{2\pi in}{L} \widehat{f}_{mn}. \quad (13)$$

To solve the Eq. (11) for each choice of the guidance vector field it is therefore necessary to compute the Fourier transform of the images whose gradients compose the vector field.

4. Experimental results

Now we list some of the possible choices for the guidance vector field and show the results of the proposed implementation. In some examples a classic multigrid method would work as well. Yet, for all examples where the structure of Ω is complicated, using the global Fourier strategy simplifies drastically the situation.

4.1. Local illumination changes

Perez et al. use the method of Fattal et al. (2002) to modify smoothly the image dynamic range. The idea is to transform the gradient field of the logarithm of the image to reduce the large gradients and to increase the small ones. The transformed gradient is used to reconstruct the logarithm of the image, by solving the Poisson equation. In fact in (Pérez et al., 2003) the authors select a region Ω , for example an under-exposed region, and transform the gradient in the log-domain by

$$\mathbf{v} = \alpha^\beta |\nabla f|^{-\beta} \nabla f, \quad (14)$$

where $\alpha = 0.2$ times the average gradient norm of f over Ω and $\beta = 0.2$, that is, the guidance vector field is a concave function of the gradient in the log-domain.

The Perez et al. method is relatively complex. Indeed, it depends on two parameters, α, β . The selection of the dark region is manual and only permits to select a few simple image regions. Thus, it is not adapted to treat the dark regions of images with back light or excessively dark shadows. Indeed, in such images the dark regions can be many and have a complicated topology. Thus, they can hardly be selected manually. The alternative proposed here is to select the dark region using a threshold T . $T = 50$ was the default threshold used successfully in all treated examples. Applying this threshold in classic JPEG images to the gray level intensity yields automatically the perceptually dark regions. Over the dark region, the gradient vector of the gray level intensity is amplified by a factor α . Since a stronger factor would amplify too much the noise, the default value was 2.5.

These values ($T = 50, \alpha = 2.5$) are easily explained. First of all most eight bit digital images have their parts with luminance below 50 quite dark and no detail can be really seen on most screens below this value. It is therefore natural to try to enhance contrast in these regions. If such dark regions are small then the method needs not be applied, no image improvement is possible. If instead the dark regions are large, this contrast enhancement can add up valuable visual information. Now the question is: why $\alpha = 2.5$ as default value? Here again, we assume a decent quality image, where the noise standard deviation does not exceed four. Then multiplying the contrast by 2.5 gives back a final noise of standard deviation less than 10, which is the limit where the noise is still tolerable for users.

In short, the default values $T = 50$ and $\alpha = 2.5$ are not universal. They are motivated by the average quality eight bit images and the average sensitivity of visual perception in dark regions. If the method has to become automatic, then the best method is to select the level λ such that the level set $\{x | u(x) \leq \lambda\}$ has its area equal to 25% of the image area. Then $\alpha = 2.5$, as indicated. With these default values for T, α the proposed algorithm becomes fully automatic, and it is anyway easy to change them interactively:

- Select the dark region Ω by the threshold T .
- Define the guidance vector field by

$$\mathbf{V} = \begin{cases} \nabla f & \text{in } R \setminus \Omega, \\ \alpha \nabla f & \text{in } \Omega, \end{cases} \quad (15)$$

where f is the gray level intensity.



Fig. 3. Top left: original image. Top right: zones with intensity below $T = 50$. Bottom: result using guidance vector field (15) with $\alpha = 2.5$.

- Solve the Poisson equation with Neumann boundary conditions using the Fourier transform as explained in the previous section.
- Compute the color channels proportionally using the new gray level intensity.

Figs. 3–5 show results obtained using the proposed algorithm. Note the complexity of the topology dark regions. The method can be tried on line on any uploaded image at http://www.ipol.im/pub/demo/lmps_selective_contrast_adjustment/.

4.2. Other editing operations

The goal of this short section is to illustrate the fact that the proposed Fourier method gives equivalent results to multigrid

methods. In the classic copy-paste problem, the guidance vector field \mathbf{v} is a gradient field from another source image g . The guidance field therefore is

$$\mathbf{V} = \begin{cases} \nabla g, & \text{over } \Omega, \\ \nabla f, & \text{otherwise,} \end{cases} \quad (16)$$

where f be the image which will be kept on $R \setminus \Omega$. Fig. 6 shows the source image g , the original image f and the obtained result.

The same technique also can be used as inpainting method, using as source image the destination image itself, typically to remove an object and to replace it by a texture patch from the same image. In Fig. 7 we can observe an example of this application.

Another application proposed in Perez et al. involves the guidance vector field

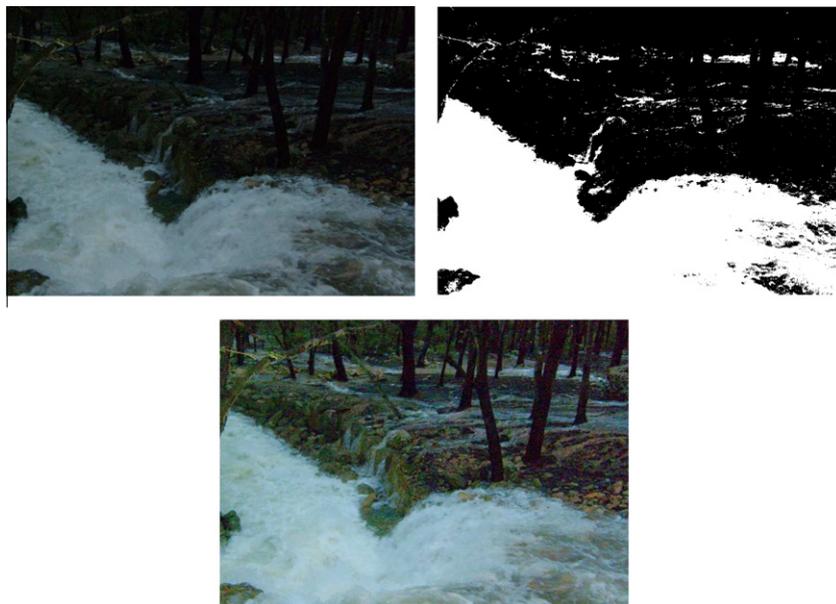


Fig. 4. Top left: original image. Top right: zones with intensity below $T = 50$. Bottom: result using guidance vector field (15) with $\alpha = 2.5$.



Fig. 5. Top left: original image. Top right: zones with intensity below $T = 50$. Bottom: result using guidance vector field (15) with $\alpha = 2.5$.



Fig. 6. Top left: source image. Top right: destination image. Bottom: result.



Fig. 7. Left: original image. Right: the deer is eliminated, using the original image as source and destination image.

$$\mathbf{V} = \begin{cases} \nabla f, & \text{if } |\nabla f| > |\nabla g|, \\ \nabla g, & \text{otherwise.} \end{cases} \quad (17)$$

In this case, two images are mixed into one, thus creating a realistic graffiti (Fig. 8).

5. Correctness of the method

Since the proposed implementation solves the Poisson equation over the whole image domain instead of the selected sub-domain, it is a sanity check to verify that the known pixel values are only



Fig. 8. Left: text image. Middle: texture image. Right: mixing result.

Table 1
Comparison between original image and result of Neumann problem.

Image	MSE	PSNR	MD
Fig. 3	0.00325	71.72	0.1136
Fig. 4	0.00575	69.33	0.055
Fig. 5	4.9e-5	88.89	0.026
Fig. 7	1.15e-5	96.46	0.005

minimally altered. This was done experimentally with the original images of Figs. 4, 6 and 7. On these images we solved the Neumann problem

$$\begin{cases} \Delta u = \Delta f, & \text{in } R, \\ \frac{\partial u}{\partial n} = 0, & \text{in } \partial R \end{cases} \quad (18)$$

and finally we imposed that mean and standard deviation of u remain the same as for f . Table 1 shows the MSE, PSNR, and MD (maximal difference) between u and f . It remains far below the any perceptual difference in these images.

To confirm that outside the selection area the changes introduced are negligible and no Gibbs effect is produced, Fig. 9 shows an experiment where we have pasted a high frequency texture on a low frequency texture. The Poisson editing technique creates a minimal perturbation on the background and no Gibbs effect. By

looking carefully at the boundary of the pasted region, one can, however observe some diffusion from the extrema of the high frequency texture into the low texture region.

6. Discussion

Seamless image editing introduced by Pérez et al. (2003) requires to solve a large linear system on an arbitrary region of the image domain, and becomes slow and difficult if the region is non connected and complex, thus not amenable to multigrid methods. The Fourier approach proposed here is able to solve all image editing tools proposed in (Pérez et al., 2003). The Fourier method is exact, non-iterative, and allows to treat regions with arbitrary topology. Its main advantage with respect to former implementations is that it is no more limited to a manual selection of one or several simple regions; image regions selected automatically by any criterion can be edited at no additional computational cost. There are only two cases where the original numerical method by finite difference and the multigrid decomposition is preferable. The first case is when the edited region is actually small and rectangular, in which case the complexity of a multigrid method, being proportional to the domain size, becomes inferior to the FFT complexity on the whole image. The second case is when it is absolutely required not to alter, even minimally, the colors in the image outside the edited zone. To illustrate the simplicity of the method, a fast and automatic local contrast enhancement algorithm treating complex regions has been tested.

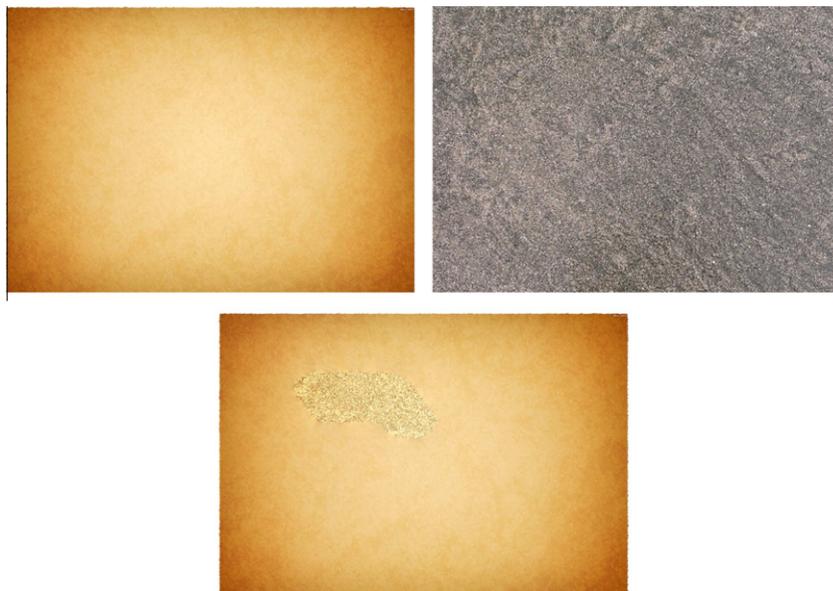


Fig. 9. Top left: low frequency texture. Top right: high frequency texture. Bottom: pasting result.

References

- Agarwala, A., 2007. Efficient gradient-domain compositing using quadtrees. *ACM Trans. Graph.* 26.
- Bertalmio, M., Vese, L., Sapiro, G., Osher, S., 2003. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Processing.* 12, 882–889.
- Bertalmio, M., Sapiro, G., Ballester, C., Caselles, V., 2000. Image inpainting. In: *SIGGRAPH'00*.
- Caselles, V., Lisani, J.L., Morel, J.M., Sapiro, G., 1999. Shape preserving local histogram modification. *IEEE Trans. Image Process.* 8, 220–230.
- Chuan, Q., Shuozhong, W., Xinpeng, Z., 2008. Image editing without color inconsistency using modified poisson equation. In: *Interant. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 397–401.
- Fattal, R., Lischinski, D., Werman, M., 2002. Gradient domain high dynamic range compression. *ACM Trans. Graph.* 21, 242–256.
- Grady, L., 2008. A lattice-preserving multigrid method for solving inhomogeneous poisson equation used in image analysis. In: *Proc. 10th European Conf. on Computer Vision*, pp. 252–264.
- Grady, L., Tasdizen, T., Whitaker, R., 2005. A geometric multigrid approach to solving the 2d inhomogeneous laplace equation with internal dirichlet boundary conditions. In: *IEEE Internat. Conf. on Image Processing*, vol. 2, pp. 642–645.
- Jeschke, S., Cline, D., Wonka, P., 2009. A GPU Laplacian solver for diffusion curves and Poisson image editing. *ACM Trans. Graph.* 28, 1–8.
- Jia, J., Sun, J., Tang, C.K., Shum, H.Y., 2006. Drag-and-drop pasting. *ACM Trans. Graph.* 3, 631–636.
- Kazhdan, M., Hoppe, H., 2008. Streaming multigrid for gradient-domain operations on large images. *ACM Trans. Graph.* 27.
- Levin, A., Zomet, A., Peleg, S., Weiss, Y., 2004. Seamless image stitching in the gradient domain. In: *Proc. Eighth European Conf. on Computer Vision*, vol. 4, pp. 377–389.
- McCann, J., Pollard, N.S., 2008. Real-time gradient-domain painting. *ACM Trans. Graph.* 27.
- Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing. *ACM Trans. Graph. (SIGGRAPH'03)* 22, 313–318.
- Sun, J., Jia, J., Tang, C.K., Shum, H.Y., 2004. Poisson matting. *ACM Trans. Graph.* 23, 315–321.
- Wang, J., Cohen, M.F., 2008. *Image and Video Matting: A survey*. Now Publishers.