COMP0026: Image Processing

Poisson Editing

Lectures will be Recorded

Recap

- Major contributions from Lucas, Tomasi, Kanade
  - Tracking feature points
  - Optical flow
- Key ideas
  - By assuming brightness constancy, truncated Taylor expansion leads to simple and fast patch matching across frames
  - Coarse-to-fine registration

Reference

Poisson Image Editing

Patrick Pérez* Michel Gangnet† Andrew Blake‡
Microsoft Research UK

Abstract

Using generic interpolation machinery based on solving Poisson equations, a variety of novel tools are introduced for seamless editing of image regions. The first set of tools permits the seamless importation of both opaque and transparent source image regions into a destination region. The second set is based on similar mathematical ideas and allows the user to modify the appearance of the image seamlessly, within a selected region. These changes can be arranged to affect the texture, the illumination, and the color of objects lying in the region, or to make tileable a rectangular selection.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering;

Keywords: Interactive image editing, image gradient, guided interpolation, Poisson equation, seamless cloning, selection editing

1 Introduction

Image editing tasks concern either global changes (color/intensity corrections, filters, deformations) or local changes confined to a selection. Here we are interested in achieving local changes, ones that are restricted to a region manually selected, in a seamless and effortless manner. The extent of the changes ranges from slight distortions to complete replacement by novel content. Classic tools to achieve that include image filters confined to a selection, for slight changes, and interactive cut-and-paste with cloning tools for complete replacements. With these classic tools, changes in the selected regions result in visible seams, which can be only partly hidden, subsequently, by feathering along the border of the selected region.

We propose here a generic machinery from which different tools for seamless editing and cloning of a selection region can be derived. The mathematical tool at the heart of the approach is the Poisson partial differential equation with Dirichlet boundary conditions which specifies the Laplacian of an unknown function over the domain of interest, along with the unknown function values over the boundary of the domain. The motivation is twofold.

First, it is well known to psychologists [Land and McCann 1971] that slow gradients of intensity, which are suppressed by the Laplacian operator, can be superimposed on an image with barely noticeable effect. Conversely, the second-order variations extracted by the Laplacian operator are the most significant perceptually.

Secondly, a scalar function on a bounded domain is uniquely defined by its values on the boundary and its Laplacian in the interior. The Poisson equation therefore has a unique solution and this leads to a sound algorithm.

So, given methods for crafting the Laplacian of an unknown function over some domain, and its boundary conditions, the Poisson equation can be solved numerically to achieve seamless filling of that domain. This can be replicated independently in each of the channels of a color image. Solving the Poisson equation also has an alternative interpretation as a minimization problem: it computes the function whose gradient is the closest, in the L_2-norm, to some prescribed vector field — the *guidance* vector field — under given boundary conditions. In that way, the reconstructed function interpolates the boundary conditions inwards, while following the spatial variations of the guidance field as closely as possible. Section 2 details this guided interpolation.

We will examine a number of possible choices for the guidance vector field. We show in particular that this interpolation machinery leverages classic cloning tools, both in terms of ease of use and capabilities. The resulting cloning allows the user to remove and add objects seamlessly. By mixing suitably the gradient of the source image with that of the destination image, it also becomes possible to add transparent objects convincingly. Furthermore, objects with complex outlines including holes can be added automatically without the need for painstaking cutting out. These different cloning facilities are presented in Section 3.

As shown in Section 4, the same machinery can also be used to modify the appearance of an image within a restricted domain, while avoiding visible discontinuities on the domain boundary. In particular, the color, the texture, or the illumination of an object can easily be modified without any need for precise delineation of object boundaries. Also, a rectangular image region can be made seamlessly tileable.

Related work The Poisson equation has been used extensively in computer vision. It arises naturally as a necessary condition in the solution of certain variational problems. In the specific context of image editing applications three previous pieces of work are related to the use of the Poisson equation proposed here.

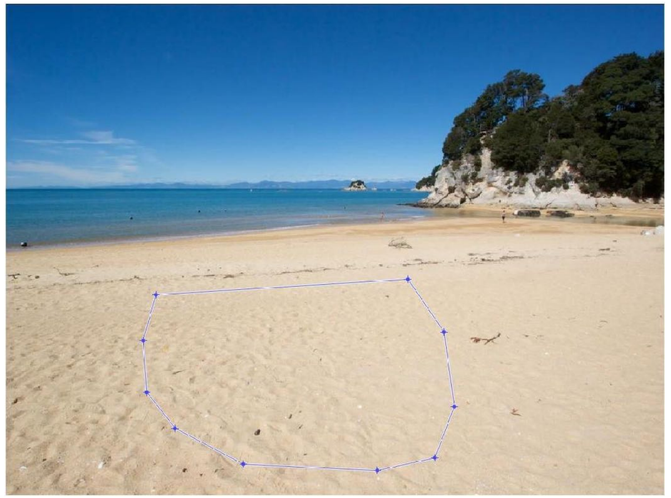
In [Fattal et al. 2002], the gradient field of a High Dynamic Range (HDR) image is rescaled non-linearly, producing a vector field that is no longer a gradient field. A new image is then obtained by solving a Poisson equation with the divergence of this vector field as right-hand-side and under Neumann boundary conditions specifying that the value of the gradient of the new image in the direction normal to the boundary is zero. In contrast, the method we are proposing here can be applied to arbitrary patches selected from an image, not just to the entire image. In order to do this, Neumann boundary conditions on a rectangular outline must be replaced by Dirichlet conditions on an arbitrary outline. A further generalization is to extend the range of nonlinear operations applied to gradients, to include maximum operations and suppression of small gradients, both of which have useful editing functions.

In [Elder and Goldberg 2001], a system is introduced to edit an image via a sparse set of its edge elements (edgels). To suppress an

*e-mail: pperez@microsoft.com
†e-mail:mgangnet@microsoft.com
‡e-mail:ablake@microsoft.com

Goal

Direct cloning

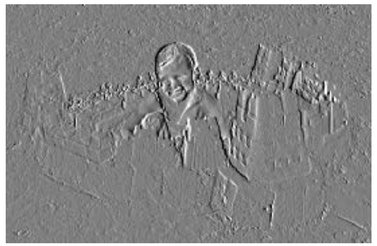


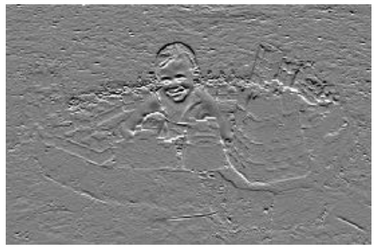
Why does it look wrong?

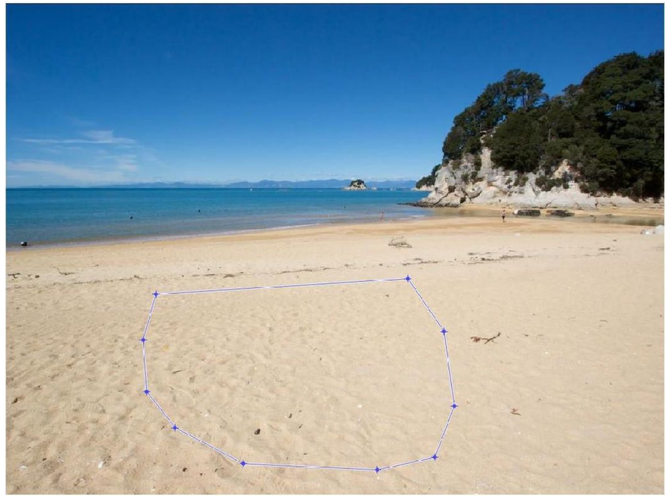
- Land and McCann, 1971

- For the human eye
  - Low gradients
    - Hard to notice difference between pixels
  - Second order variations (Laplace)
    - We notice them a lot

Seamless Cloning







Basic Approach

Basic Approach

$$\nabla I_{x,y} = \begin{bmatrix} I_{x+1,y} - I_{x,y} \\ I_{x,y+1} - I_{x,y} \end{bmatrix}$$

Basic Approach

$$\nabla I_{x,y} = \begin{bmatrix} I_{x+1,y} - I_{x,y} \\ I_{x,y+1} - I_{x,y} \end{bmatrix}$$

$$G_{x,y} = \nabla I_{x,y} \cdot \frac{f(\|\nabla I_{x,y}\|)}{\|\nabla I_{x,y}\|}$$

Basic Approach

$$\nabla I_{x,y} = \begin{bmatrix} I_{x+1,y} - I_{x,y} \\ I_{x,y+1} - I_{x,y} \end{bmatrix}$$

$$G_{x,y} = \nabla I_{x,y} \cdot \frac{f(\|\nabla I_{x,y}\|)}{\|\nabla I_{x,y}\|}$$

$$\arg\min_I \sum_{x,y} [(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y})^2 + (I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})^2]$$

Basic Approach

$$\frac{\partial \mathcal{L}}{\partial I_{x,y}} = -2(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y}) - 2(I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})$$
$$+2(I_{x,y} - I_{x-1,y} - G^{(x)}_{x-1,y}) + 2(I_{x,y} - I_{x,y-1} - G^{(y)}_{x,y-1})$$

Basic Approach

$$\frac{\partial \mathcal{L}}{\partial I_{x,y}} = -2(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y}) - 2(I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})$$
$$+2(I_{x,y} - I_{x-1,y} - G^{(x)}_{x-1,y}) + 2(I_{x,y} - I_{x,y-1} - G^{(y)}_{x,y-1})$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

Basic Approach

$$\frac{\partial \mathcal{L}}{\partial I_{x,y}} = -2(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y}) - 2(I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})$$
$$+2(I_{x,y} - I_{x-1,y} - G^{(x)}_{x-1,y}) + 2(I_{x,y} - I_{x,y-1} - G^{(y)}_{x,y-1})$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

Basic Approach

$$\frac{\partial \mathcal{L}}{\partial I_{x,y}} = -2(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y}) - 2(I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})$$
$$+2(I_{x,y} - I_{x-1,y} - G^{(x)}_{x-1,y}) + 2(I_{x,y} - I_{x,y-1} - G^{(y)}_{x,y-1})$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

Basic Approach

$$\frac{\partial \mathcal{L}}{\partial I_{x,y}} = -2(I_{x+1,y} - I_{x,y} - G^{(x)}_{x,y}) - 2(I_{x,y+1} - I_{x,y} - G^{(y)}_{x,y})$$
$$+2(I_{x,y} - I_{x-1,y} - G^{(x)}_{x-1,y}) + 2(I_{x,y} - I_{x,y-1} - G^{(y)}_{x,y-1})$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

Basic Approach

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

$$I_{x+1,y} + I_{x,y+1} + I_{x-1,y} + I_{x,y-1} - 4I_{x,y} = G^{(x)}_{x,y} - G^{(x)}_{x-1,y} + G^{(y)}_{x,y} - G^{(y)}_{x,y-1}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * I = [-1 \;\; 1 \;\; 0] * G^{(x)} + [-1 \;\; 1 \;\; 0] * G^{(y)}$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

Basic Approach

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

$$A\mathbf{I} = \mathbf{b}$$

Basic Approach

$$\Delta I = \nabla^2 I = \mathrm{div} G$$

$$[0 \;\; 1 \;\; 0 \;\; 1 \;\; -4 \;\; 1 \;\; 0 \;\; 1 \;\; 0] \cdot I = [-1 \;\; 1 \;\; 0] \cdot G^{(x)} + [-1 \;\; 1 \;\; 0] \cdot G^{(y)}$$

$$A\mathbf{I} = \mathbf{b}$$

$$A = \begin{bmatrix} -2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -3 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -2 \end{bmatrix}$$

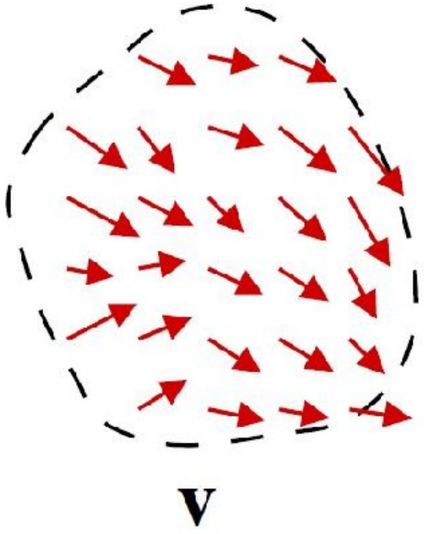
Formulation

- For the discrete case of an image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$

$$\begin{aligned} \nabla f(x,y) &= (\nabla_x f(x,y) + \nabla_y f(x,y)) \\ &= (f(x+1,y) - f(x,y), f(x,y+1) - f(x,y)) \end{aligned}$$

$$\begin{aligned} \mathrm{div}(\nabla f(x,y)) &= \mathrm{div}(\nabla_x f(x,y), \nabla_y f(x,y)) \\ &= \nabla_x^2 f(x,y) + \nabla_y^2 f(x,y)) \\ &= \Delta f(x,y) \\ &= 4f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1) \\ &= 4f(x,y) - \sum_{f(i,j) \in \mathcal{N}(x,y).} f(i,j) \end{aligned}$$

Problem Formulation



Guidance vector field

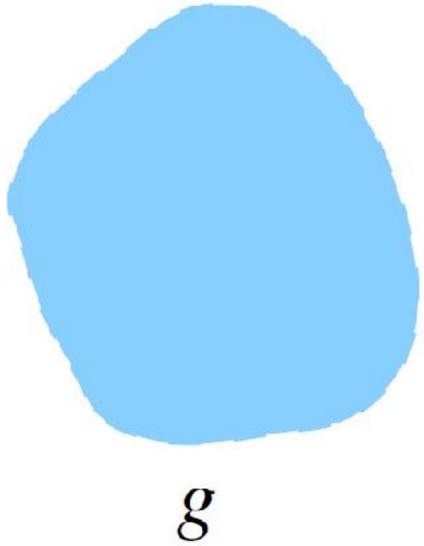
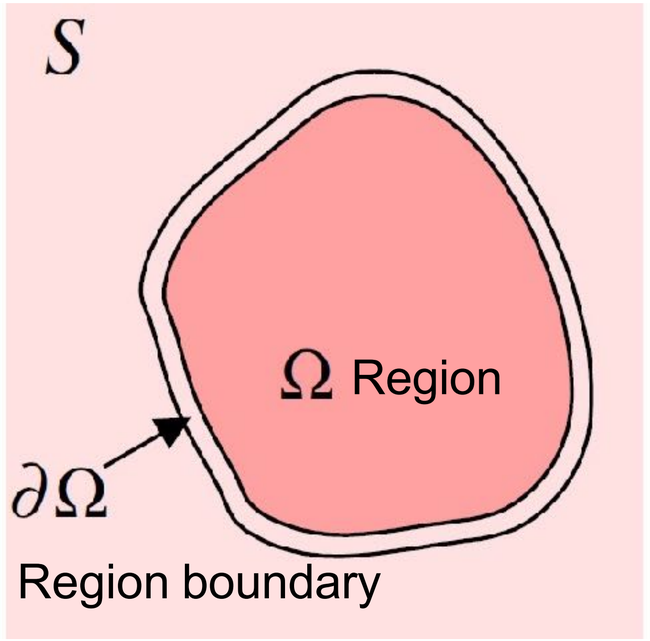
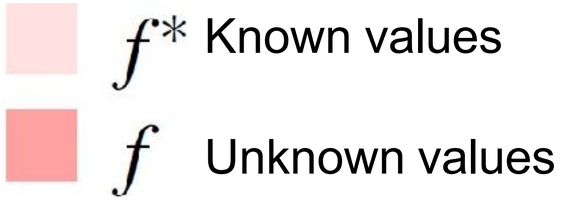


Image source region

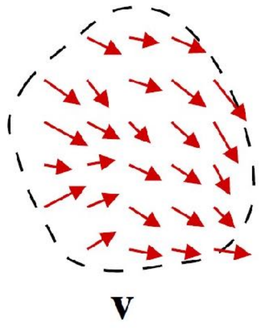


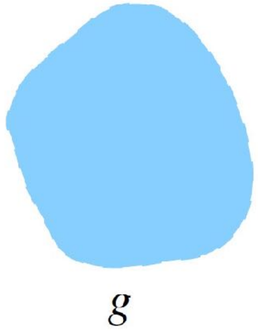
Destination image

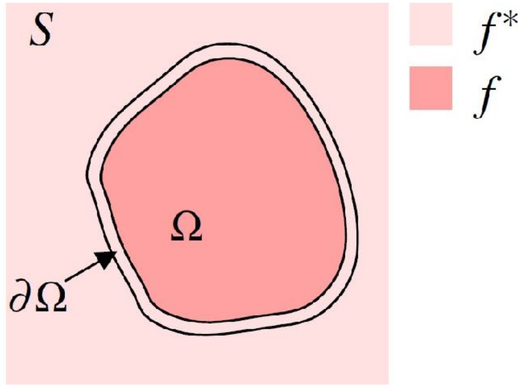


$$\Delta f = \text{div}\mathbf{v} \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Formulation







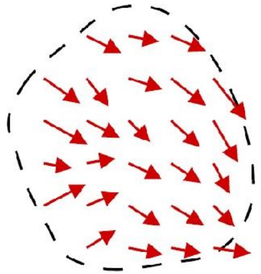
$$\Delta f = \text{div}\mathbf{v} \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

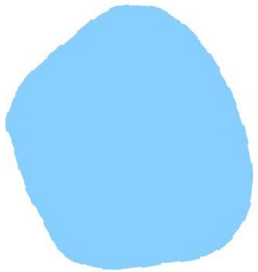
$$\Delta f_p = |\mathcal{N}_p| f_p - \sum_{q \in \mathcal{N}_p \cap \Omega} f_q - \sum_{q \in \mathcal{N}_p \cap \Omega} f_q^*$$

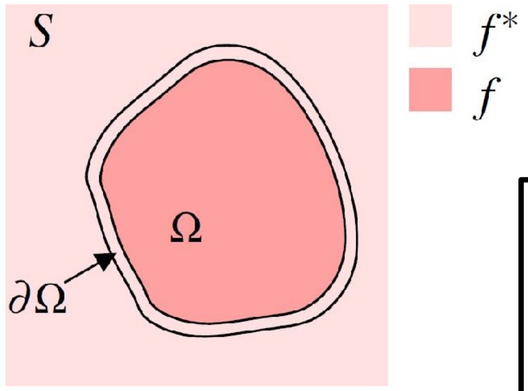
$$\text{div}\mathbf{v} = \sum_{q \in \mathcal{N}_p} v_{pq} \qquad (v_{pq} = g_p - g_q)$$

$$\boxed{|\mathcal{N}_p| f_p - \sum_{q \in \mathcal{N}_p \cap \Omega} f_q = \sum_{q \in \mathcal{N}_p \cap \partial\Omega} f_q^* + \sum_{q \in \mathcal{N}_p} v_{pq}}$$

Poisson Equation







$$\Delta f = \mathrm{div}\mathbf{v} \text{ with } f|\partial\Omega = f^*|\partial\Omega$$

$$|\mathcal{N}_p| f_p - \sum_{q \in \mathcal{N}_p \cap \Omega} f_q = \sum_{q \in \mathcal{N}_p \cap \partial\Omega} f_q^* + \sum_{q \in \mathcal{N}_p} v_{pq}$$

$$\mathbf{Ax} = \mathrm{b} \rightarrow \begin{pmatrix} |\mathcal{N}_{p_1}| & a_{p_1 p_2} & a_{p_1 p_3} & \cdots & a_{p_1 p_{|\Omega|}} \\ a_{p_2 p_1} & |\mathcal{N}_{p_2}| & a_{p_2 p_3} & \cdots & a_{p_2 p_{|\Omega|}} \\ a_{p_3 p_1} & a_{p_3 p_2} & |\mathcal{N}_{p_3}| & \cdots & a_{p_3 p_{|\Omega|}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{p_{|\Omega|} p_1} & a_{p_{|\Omega|} p_2} & a_{p_{|\Omega|} p_3} & \cdots & |\mathcal{N}_{p_{|\Omega|}}| \end{pmatrix} \begin{pmatrix} f_{p_1} \\ f_{p_2} \\ f_{p_3} \\ \vdots \\ f_{p_{|\Omega|}} \end{pmatrix} = \begin{pmatrix} b_{p_1} \\ b_{p_2} \\ b_{p_3} \\ \vdots \\ b_{p_{|\Omega|}} \end{pmatrix}$$

$$a_{p_i p_j} = \begin{cases} -1, & p_j \in \mathcal{N}_{p_i} \cap \Omega \\ 0, & otherwise \end{cases}$$

$$b_{p_i} = \sum_{q \in \mathcal{N}_{p_i} \cap \partial\Omega} f_q^* + \sum_{q \in \mathcal{N}_{p_i}} v_{p_i q}$$

Seamless Cloning

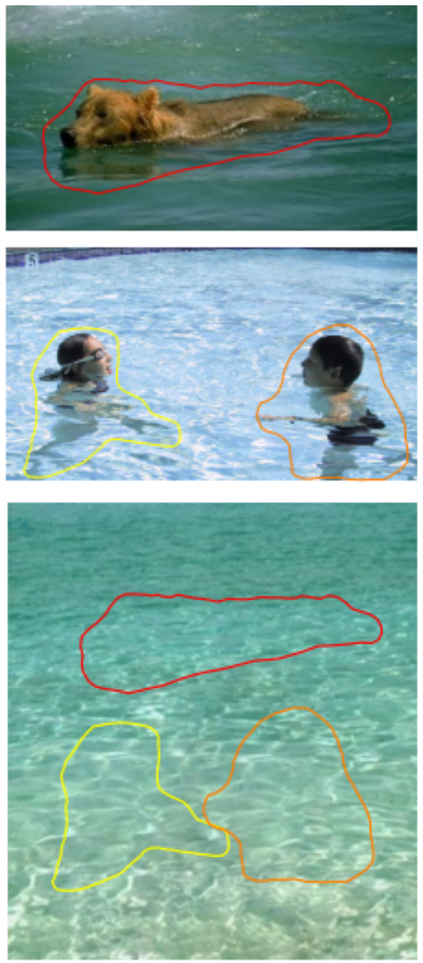
$$\mathbf{v} = \nabla g$$

$$\Delta f = \Delta g \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Examples

sources | destinations | cloning | seamless cloning

Examples



sources/destinations

cloning

seamless cloning

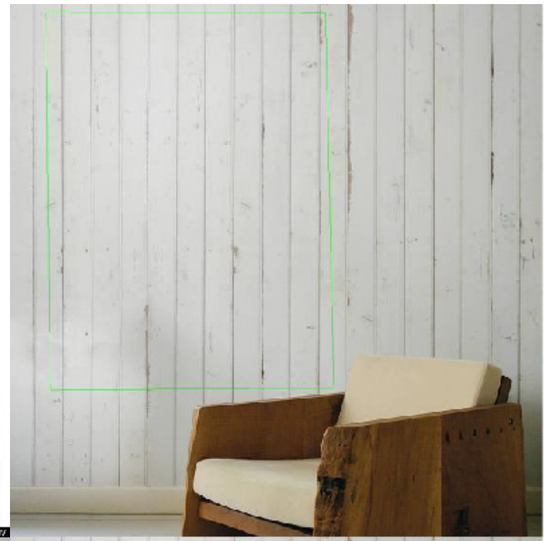
Model and Solve SLE in Matlab

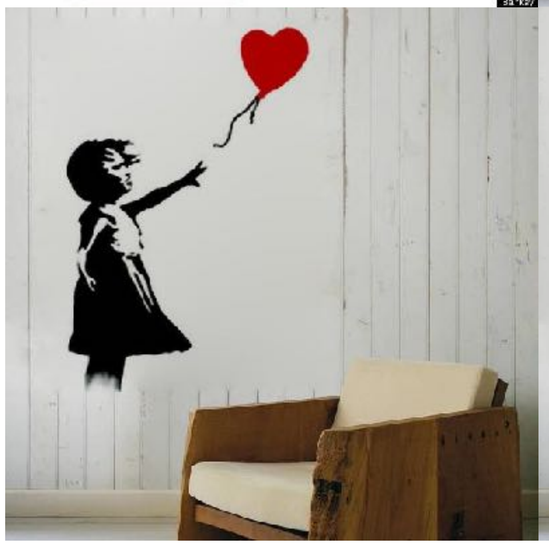
- Matrix $\mathbf{A}$ is a really sparse matrix
  - At most, five non-zero elements per row
  - Define using the function sparse
- Solve SLE
  - Use backslash operator: x=A\b
  - Returns the least square solution

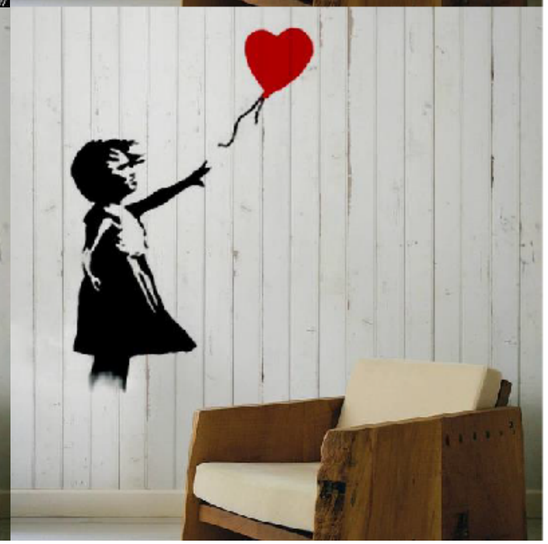
Guidance Vector Field

- **Different effects depending on chosen v_{pq}:**
- $v_{pq} = 0$
- $v_{pq} = g_p - g_q$
- $$v_{pq} = \begin{cases} f_p^* - f_q^*, & \text{if } |f_p - f_q| > |g_p - g_q| \\ g_p - g_q, & \text{otherwise.} \end{cases}$$

Have fun!







Reference

P. Perez, M. Gangnet, and A. Blake. 2003. *Poisson Image Editing.* SIGGRAPH ’03, pp. 313–318.