MA3111: Mathematical Image Processing
Image Inpainting

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 320317, Taiwan

First version: October 08, 2021/Last updated: November 25, 2024

Outline of “image inpainting”

In this lecture, we will introduce image inpainting

- *using the variational method (VM) with split Bregman and*
- *using sparse representation and dictionary learning (SRDL).*

The material of this lecture is based on

- (VM) P. Getreuer, Total variation inpainting using split Bregman, *Image Processing On Line,* 2 (2012), pp. 147-157.
- (SRDL) Y. Sharon, J. Wright, and Y. Ma, Computation and relaxation of conditions for equivalence between ℓ^1 and ℓ^0 minimization, *UIUC Technical Report UILU-ENG-07-2008, 2007.*

Image inpainting:
https://www.nvidia.com/research/inpainting/index.html

Matlab codes: http://brendt.wohlberg.net/software/SPORCO/

Basic ideas of image inpainting

- Given an image where a specified region is unknown, *image inpainting or image completion* is the problem of inferring the image content in this region (無中生有).
- Image inpainting is an interpolation problem, filling the unknown region with a condition to agree with the known image on the boundary.
- A classical solution for such an interpolation is to solve Laplace's equation. However, *Laplace's equation is usually unsatisfactory for images since it is overly smooth.*

  It cannot recover a step edge passing through the region.

H^1 inpainting (Laplace's equation)

- Let $f : \overline{\Omega} \to \mathbb{R}$ be a given grayscale image and let $D \subset \Omega$ be an open set representing the region to be inpainted. In other words, *it is supposed that f is known in $\overline{\Omega} \setminus D$ and unknown in D.*
- The inpainting solution by Laplace's equation is to solve

$$\begin{cases} -\Delta u = 0 & \text{in } D, \\ u = f & \text{on } \partial D, \end{cases}$$

and $u = f$ in $\overline{\Omega} \setminus D$.

- By Dirichlet's principle, if the Laplace inpainting solution u is in $C^2(D)$, then u is the minimizer of the Dirichlet energy:

$$E[v] := \int_D \left(\frac{1}{2} |\nabla v|^2 - v \times 0 \right) d\boldsymbol{x},$$

for all $v \in C^2(D)$ satisfying the boundary condition $u = f$ on ∂D.

- Note that $-\Delta u = 0$ in D is the Euler-Lagrange equation of the energy functional $E[v]$.

The bounded variation space $BV(\Omega)$

Let Ω be an open subset of $\mathbb{R}^2$. The space of functions of bounded variation $BV(\Omega)$ is defined as the space of real-valued function $u \in L^1(\Omega)$ such that the total variation is finite, i.e.,

$$BV(\Omega) = \{u \in L^1(\Omega) : \|u\|_{TV(\Omega)} < \infty\},$$

where the semi-norm $\|u\|_{TV(\Omega)}$ is defined as

$$\|u\|_{TV(\Omega)} := \sup\left\{ \int_\Omega u(\nabla \cdot \varphi)\, d\boldsymbol{x} : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{(L^\infty(\Omega))^2} \le 1 \right\}.$$

- *$BV(\Omega)$ is a Banach space with the norm,*

$$\|u\|_{BV(\Omega)} = \|u\|_{L^1(\Omega)} + \|u\|_{TV(\Omega)}.$$

- *If u is smooth, then $\|u\|_{TV(\Omega)} = \int_\Omega |\nabla u|\, d\boldsymbol{x}$.*

The ROF total-variation model for image denoising

Let $f : \overline{\Omega} \subset \mathbb{R}^2 \to \mathbb{R}$ be a given noisy image. Rudin, Osher, and Fatemi (*Physica D, 1992*) proposed the model for image denoising:

$$\min_{u \in BV(\Omega) \cap L^2(\Omega)} \Big\{ \underbrace{\|u\|_{TV(\Omega)}}_{\textit{regularizer}} + \frac{\lambda}{2} \underbrace{\int_\Omega \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 \, d\boldsymbol{x}}_{\textit{data fidelity}} \Big\},$$

where $\lambda > 0$ is a tuning parameter which controls the regularization strength. Notice that

- A smaller value of λ will lead to a more regular solution.
- The space of functions with bounded variation help remove spurious oscillations (noise) and preserve sharp signals (edges).
- The *TV* term allows the solution to have discontinuities.

The total variation model for image inpainting

- Let $f : \overline{\Omega} \to \mathbb{R}$ be a given grayscale image and let $D \subset \Omega$ be an open set representing the region to be inpainted. In other words, *it is supposed that f is known in $\overline{\Omega} \setminus D$ and unknown in D.*
- The *TV* inpainting method is to find the *BV* function u that solves the minimization problem:

$$\min_{u \in BV(\Omega) \cap L^2(\Omega)} \Big\{ \underbrace{\|u\|_{TV(\Omega)}}_{regularizer} + \underbrace{\frac{\lambda}{2} \int_{\Omega \setminus D} \big(u(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 \, d\boldsymbol{x}}_{data\ fidelity} \Big\},$$

  where $\lambda > 0$ is a regularization parameter.
- Under suitable assumptions, minimizers u exist but are generally not unique.

The *TV* inpainting model may be viewed as denoising

- Inpainting may be viewed as denoising with a *spatially-varying regularization strength* $\lambda(\boldsymbol{x}) \geq 0$,

  $$\min_{u \in BV(\Omega) \cap L^2(\Omega)} \left\{ \|u\|_{TV(\Omega)} + \frac{1}{2} \int_\Omega \lambda(\boldsymbol{x}) \left(u(\boldsymbol{x}) - f(\boldsymbol{x}) \right)^2 d\boldsymbol{x} \right\},$$

  where $\lambda(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in D$ and $\lambda(\boldsymbol{x}) > 0$ for $\boldsymbol{x} \in \Omega \setminus D$.

- For $\boldsymbol{x} \in D$ where $\lambda(\boldsymbol{x}) = 0$, the value $f(\boldsymbol{x})$ is unused and $u(\boldsymbol{x})$ is only influenced by the $\|u\|_{TV(\Omega)}$ term. Outside of D, the model performs TV-regularized denoising and $\lambda|_{\Omega \setminus D}$ specifies the denoising strength.

- This denoising behavior may be desirable when it is difficult to specify the inpainting domain accurately. *By setting λ to a very large value, the denoising effect is limited so that the image remains nearly unchanged outside of D.*

Discretization of the ROF model (matrix indices)

Total variation is approximated by $\|u\|_{TV(\Omega)} \approx h^2 \sum_{i=1}^{N} \sum_{j=1}^{N} |\nabla u_{i,j}|$, where the discrete gradient operator as $\nabla u_{i,j} = (\nabla_x^+ u_{i,j}, \nabla_y^+ u_{i,j})^\top$,

$$\nabla_x^+ u_{i,j} = \begin{cases} \dfrac{u_{i,j+1} - u_{i,j}}{h}, \ 1 \le j \le N-1, \\ 0, \ j = N; \end{cases} \quad \nabla_y^+ u_{i,j} = \begin{cases} \dfrac{u_{i+1,j} - u_{i,j}}{h}, \ 1 \le i \le N-1, \\ 0, \ i = N. \end{cases}$$

Applying the operator splitting technique, we obtain the constrained approximate minimization of the TV model:

$$\min_{d,\, u} \left(\sum_{i,j} |d_{i,j}| + \frac{1}{2} \sum_{i,j} \lambda_{i,j} (f_{i,j} - u_{i,j})^2 \right) \quad \text{subject to } d_{i,j} = \nabla u_{i,j}.$$

Introducing a penalty parameter $\gamma > 0$, we obtain the unconstrained minimization problem:

$$\min_{d,\, u} \left(\sum_{i,j} |d_{i,j}| + \frac{1}{2} \sum_{i,j} \lambda_{i,j} (f_{i,j} - u_{i,j})^2 + \frac{\gamma}{2} \sum_{i,j} |d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2 \right),$$

where b is an auxiliary variable related to the Bregman iteration.

An alternating direction approach: Split Bregman method

Goldstein and Osher (2009) proposed to solve the above-mentioned problem by an alternating direction approach: (see Getreuer 2012)

u**-subproblem:** With d fixed, we solve

$$u^{k+1} = \arg\min_u \left(\frac{1}{2} \sum_{i,j} \lambda_{i,j} (f_{i,j} - u_{i,j})^2 + \frac{\gamma}{2} \sum_{i,j} |d^k_{i,j} - \nabla u_{i,j} - b^k_{i,j}|^2 \right).$$

The optimal u satisfies a discrete screened Poisson equation,

$$\lambda(u - f) + \gamma \nabla \cdot (\nabla u - d + b) = 0,$$

or equivalently,

$$\lambda u - \gamma \Delta u = \lambda f - \gamma \nabla \cdot (d - b),$$

$\nabla \cdot$ and Δ are the discrete divergence and Laplacian, respectively.

It can be viewed as the EL equation of the minimization problem:

$$\min_u \frac{1}{2} \int_\Omega \lambda (f - u)^2 \, d\mathbf{x} + \frac{\gamma}{2} \int_\Omega |d - \nabla u - b|^2 \, d\mathbf{x}.$$

The discrete screened Poisson equation

The discrete screened Poisson equation

$$\lambda u - \gamma \Delta u = \lambda f - \gamma \nabla \cdot (d - b),$$

which is a symmetric and diagonally dominant linear system, may be solved for u in the Fourier domain or by the iterative matrix techniques such as the Gauss-Seidel iterative method:

$$(\lambda_{i,j} + 4\gamma) u_{i,j}^{k+1} = c_{i,j}^{k} + \gamma \left(u_{i-1,j}^{k+1} + u_{i+1,j}^{k} + u_{i,j-1}^{k+1} + u_{i,j+1}^{k} \right),$$

where

$$c_{i,j}^{k} := \big(\lambda f - \gamma \nabla \cdot (d - b)\big)_{i,j}^{k}.$$

d-subproblem

d-subproblem: With u fixed, we solve

$$d^{k+1} = \arg\min_{d} \left(\sum_{i,j} |d_{i,j}| + \frac{\gamma}{2} \sum_{i,j} |d_{i,j} - \nabla u_{i,j}^{k+1} - b_{i,j}^{k}|^2 \right),$$

which has a closed-form solution,

$$d_{i,j}^{k+1} = \frac{\nabla u_{i,j}^{k+1} + b_{i,j}^{k}}{|\nabla u_{i,j}^{k+1} + b_{i,j}^{k}|} \max\left\{ |\nabla u_{i,j}^{k+1} + b_{i,j}^{k}| - \frac{1}{\gamma},\ 0 \right\}.$$

Note: The solution of d-subproblem can be found componentwisely. For each (i, j), the minimizer is given below:

$$\begin{aligned} \arg\min_{\boldsymbol{x} \in \mathbb{R}^2} \left\{ |\boldsymbol{x}| + \frac{\gamma}{2} |\boldsymbol{x} - \boldsymbol{y}|^2 \right\} &= \begin{cases} \frac{\boldsymbol{y}}{|\boldsymbol{y}|}\left(|\boldsymbol{y}| - \frac{1}{\gamma}\right), & |\boldsymbol{y}| > \frac{1}{\gamma} \\ \boldsymbol{0}, & |\boldsymbol{y}| \le \frac{1}{\gamma} \end{cases} \\ &= \frac{\boldsymbol{y}}{|\boldsymbol{y}|} \max\left\{ |\boldsymbol{y}| - \frac{1}{\gamma},\ 0 \right\}. \end{aligned}$$

Updating b and selecting γ

- **Updating b:** The auxiliary variable b is initialized to zero and updated as $b^{k+1} = b^k + \nabla u^{k+1} - d^{k+1}$.
- **Selecting γ:** A good choice of γ is one for which both d and u subproblems converge quickly and are numerically well-conditioned.
  - In d subproblem, the shrinking effect is more dramatic when γ is small.
  - In u subproblem, the effect of Δ and $\nabla\cdot$ increase when γ gets larger. It is also ill-conditioned in the limit $\gamma \to \infty$.

Therefore, γ should be neither extremely large nor small for good convergence.

The split Bregman algorithm

The split Bregman algorithm:

initialize $u = f, d = b = \mathbf{0}$

while $\|u_{current} - u_{previous}\|_2 > tolerance$ *do*

solve the u-subproblem

solve the d-subproblem

$b = b + \nabla u - d$

The default parameter values are: *tolerance* $= \|f\|_2/10^5$ and $\gamma = 5$.

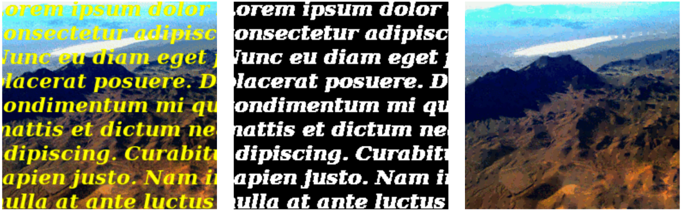
Color images (RGB channels): The vectorial TV (VTV) is used in place of TV,

$$\|u\|_{VTV(\Omega)} := \int_\Omega \Big(\sum_{i \in \text{channels}} |\nabla u_i(\boldsymbol{x})|^2 \Big)^{1/2} d\boldsymbol{x}.$$

The grayscale algorithm can be extended directly to VTV-regularized image inpainting.

Example: text removal, $\lambda = 10^4$

Input | D | TV Inpainting

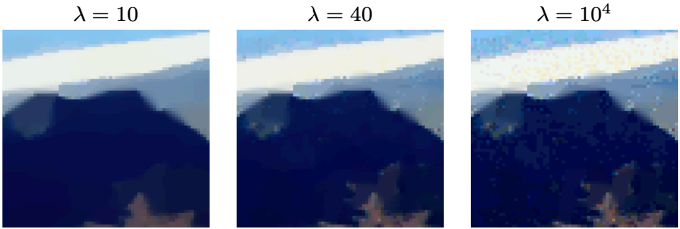


- TV inpainting (and inpainting in general) is most successful when the inpainting domain is thin.
- A good feature of TV inpainting is that it reconstructs edges rather than smoothing them.

Input | TV Inpainting

Example: effect of λ

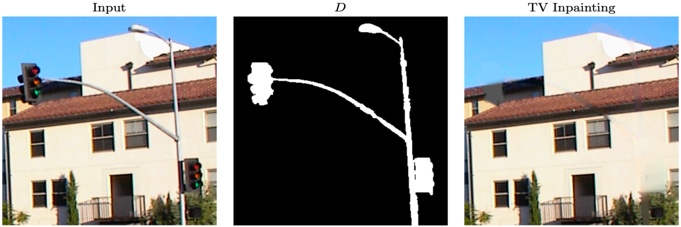
- Outside of the inpainting region D, the TV inpainting model denoises the image. The denoising strength is controlled by the value of λ, where *a smaller value implies stronger denoising.*
- In the figure below, the previous experiment is repeated with three different values of λ.



Example: object removal

In this example, *TV* inpainting is used to attempt to remove a lamppost from an image ($\lambda = 250$).

The result is reasonable over the pole where D is thin, but is poor over the signal where it is thicker.



Basic idea of the sparse representation

- The goal of sparse representation is to find *a sparse coefficient vector* $z = (z_1, z_2, \cdots, z_n)^\top \in \mathbb{R}^n$ (only a few of components of z is nonzero) such that a given signal vector $x \in \mathbb{R}^m$ is a linear combination of a few columns of *a dictionary* $D = [d_1, d_2, \cdots, d_n] \in \mathbb{R}^{m \times n}$, i.e.,

$$x = z_1 d_1 + z_2 d_2 + \cdots + z_n d_n = Dz.$$

- Typically, we use a over-completed dictionary to deal with sparse representation problem. Therefore, there does not exist a unique z such that $x = Dz$.

The sparse representation problem

- The sparse representation problem can be modeled as the following optimization problem:

$$z^* = \arg\min_{z} \|z\|_0 \quad \text{subject to} \quad x = Dz,$$

where $\|z\|_0 := \#\{i : z_i \neq 0\}$. We call $\|z\|_0$ the ℓ^0 "norm"' of z, even though ℓ^0 is not really a norm, since $\|\alpha z\|_0 \neq |\alpha| \|z\|_0$.

- We can relax it into the following unconstrained optimization problem:

$$z^* = \arg\min_{z} \Big\{ \underbrace{\frac{1}{2}\|x - Dz\|_2^2}_{\textit{data fidelity}} + \underbrace{\lambda \|z\|_0}_{\textit{regularizer}} \Big\},$$

where $\lambda > 0$ is a penalty parameter which controls the balance between the data fidelity term and the regularization term.

The sparse representation problem with ℓ^1-norm

- The above optimization problem is an *NP*-hard problem, and thus it is inefficient to solve it when n is large.
- In [Donoho, CPAM 2006-2], if $\boldsymbol{z}$ is sufficiently sparse, then $\|\boldsymbol{z}\|_1$ is a good approximation to $\|\boldsymbol{z}\|_0$.
- From now on, we mainly consider the following

  Sparse representation problem: Given a signal vector $\boldsymbol{x} \in \mathbb{R}^m$ and a dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{m \times n}$, we seek a coefficient vector $\boldsymbol{z}^* \in \mathbb{R}^n$ such that

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \left\{ \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \right\}, \qquad \lambda > 0.$$

Basic idea of the dictionary learning problem

- In the sparse representation problem, the solution of interest $\boldsymbol{z}^*$ is the coefficient vector of a linear combination of over-complete basis elements (columns) from a given dictionary $\boldsymbol{D}$ under some sparsity constraint. Therefore, it is typically accompanied by a dictionary learning mechanism.
- We are going to study a more general problem. The dictionary $\boldsymbol{D}$ is unknown and needed to be sought together with the sparse solution $\boldsymbol{z}$.

The goal of sparse dictionary learning is to learn a dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{m \times n}$ from a given dataset of signals $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^m$, together with finding the sparse coefficient vectors $\{\boldsymbol{z}_i\}_{i=1}^N \subset \mathbb{R}^n$, such that $\boldsymbol{x}_i \approx \boldsymbol{D}\boldsymbol{z}_i$, $\forall i = 1, 2, \cdots, N$.

The sparse dictionary learning problem

Sparse dictionary learning (SDL) problem:

Let $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^m$ *be a given dataset of signals. We seek a dictionary matrix* $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{m\times n}$ *together with the sparse coefficient vectors* $\{\boldsymbol{z}_i\}_{i=1}^N \subset \mathbb{R}^n$ *that solve the minimization problem:*

$$\min_{\boldsymbol{D},\{\boldsymbol{z}_i\}} \left\{ \frac{1}{2} \sum_{i=1}^N \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{z}_i\|_2^2 + \lambda \sum_{i=1}^N \|\boldsymbol{z}_i\|_1 \right\}$$

$$\text{subject to } \|\boldsymbol{d}_k\|_2 \le 1,\ \forall\, 1 \le k \le n,$$

where $\lambda > 0$ *is a penalty parameter.*

Note: To prevent the columns of $\boldsymbol{D}$ being arbitrarily large, we impose the constraints on them, since $\boldsymbol{z}_i$ could be arbitrarily small.

Problem formulation in a more compact form

To simplify the formulation of the SDL problem, we define

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{m \times N}, \qquad \boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{n \times N}.$$

Then the SDL problem can be posed as follows: *Given a training data matrix $\boldsymbol{X}$, find a dictionary matrix $\boldsymbol{D}$ and a coefficient matrix $\boldsymbol{Z}$ such that*

$$\min_{\boldsymbol{D},\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_{1,1} \right)$$

$$\text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n.$$

In the compact form, $\|\cdot\|_F$ denotes the Frobenius norm defined as follows: for a matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N] \in \mathbb{R}^{m \times N}$,

$$\|\boldsymbol{A}\|_F^2 := \sum_{i=1}^{N} \|\boldsymbol{a}_i\|_2^2$$

and $\|\boldsymbol{Z}\|_{1,1}$ is the $L_{1,1}$-norm which is defined as

$$\|\boldsymbol{Z}\|_{1,1} := \sum_{i=1}^{N} \|\boldsymbol{z}_i\|_1.$$

Image inpainting by SR & DL: step 1

Image inpainting aims to fill in the missing pixels. Below we first consider the grayscale image $\boldsymbol{I}$.

Step 1: Given a corrupted image $\boldsymbol{I}$, we divide the image domain into two disjoint regions, the target region Ω_t and the source region Ω_s, $\Omega_t := \{(i,j)|$ the pixel (i,j) need to be inpainted$\}$, Ω_s is the set of the remaining pixels (i,j). Then $\Omega = \Omega_s \cup \Omega_t$. We assume Ω_t is thin.

- Define the mask image $\boldsymbol{M}$ whose size is the same as image $\boldsymbol{I}$,

$$\boldsymbol{M}(i,j) = \begin{cases} 0, & (i,j) \in \Omega_t, \\ 1, & (i,j) \in \Omega_s. \end{cases}$$

- Given $k \in \mathbb{N}$, $k \gg 1$, we extract N patches (submatrices) $\{\boldsymbol{x}_i\}_{i=1}^N$ all with size $k \times k$ from the image (matrix) $\boldsymbol{I}$ and then define the corresponding submatrices $\{\boldsymbol{m}_i\}_{i=1}^N$ from the mask matrix $\boldsymbol{M}$.
- According to each $\boldsymbol{m}_i$, we divide the collection of patches $\{\boldsymbol{x}_i\}_{i=1}^N$ into two parts, $\{\boldsymbol{x}_{s,i}\}_{i=1}^{N_1}$ and $\{\boldsymbol{x}_{t,i}\}_{i=1}^{N_2}$. If all entries in $\boldsymbol{m}_i$ are 1, then the corresponding $\boldsymbol{x}_i \in \{\boldsymbol{x}_{s,i}\}_{i=1}^{N_1}$; otherwise, $\boldsymbol{x}_i \in \{\boldsymbol{x}_{t,i}\}_{i=1}^{N_2}$.

Image inpainting by SR & DL: step 2

Training a dictionary with the patches (submatrices) $\{x_{s,i}\}_{i=1}^{N_1}$:

- Let $m = k^2$. From now on, we rearrange each submatrix as a long column vector with m components in the natural way.
- Fixed $n \in \mathbb{N}$, we train a dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{m\times n}$ with $\boldsymbol{X} = [\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, \cdots, \boldsymbol{x}_{s,N_1}] \in \mathbb{R}^{m\times N_1}$. That is, we solve the following constrained minimization problem:

$$\underset{\boldsymbol{D},\boldsymbol{Z}}{\arg\min}\left\{\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{D}\boldsymbol{Z}\|_F^2 + \lambda\|\boldsymbol{Z}\|_{1,1}\right\}$$

$$\text{subject to } \|\boldsymbol{d}_k\|_2 \leq 1, \ \forall\, 1 \leq k \leq n.$$

Image inpainting by SR & DL: step 3

First, note that some pixel values in $\boldsymbol{x}_{t,i}$ are missing and needed to be inpainted! Next, we are going to find the sparse representations of the vectors in $\{\boldsymbol{x}_{t,i}\}_{i=1}^{N_2}$.

- Let $\mathcal{M} : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ be the function defined by

$$\mathcal{M}\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_m \end{pmatrix}.$$

- For $i = 1, 2, \cdots, N_2$, we seek the sparse representation of $\boldsymbol{x}_{t,i} \in \mathbb{R}^m$ by solving

$$\boldsymbol{z}_i^* = \arg\min_{\boldsymbol{z}_i} \left\{ \frac{1}{2} \|\mathcal{M}(\boldsymbol{m}_{t,i})(\boldsymbol{x}_{t,i} - \boldsymbol{D}\boldsymbol{z}_i)\|_2^2 + \lambda \|\mathcal{M}(\boldsymbol{m}_{t,i})\boldsymbol{z}_i\|_1 \right\},$$

  where $\boldsymbol{m}_{t,i}$ is the corresponding mask vector of $\boldsymbol{x}_{t,i}$.

Image inpainting by SR & DL: step 4

The final step is using the sparse representation vectors to inpaint the missing-pixel vectors $\boldsymbol{x}_{t,i}$:

- We replace $\boldsymbol{x}_{t,i}$ by $\boldsymbol{x}^*_{t,i}$ through the use of $\boldsymbol{D}\boldsymbol{z}^*_i$. More specifically, we define

  $$\boldsymbol{x}^*_{t,i} = \mathcal{M}(\boldsymbol{m}_{t,i})\boldsymbol{x}_{t,i} + (\mathcal{I} - \mathcal{M}(\boldsymbol{m}_{t,i}))\boldsymbol{D}\boldsymbol{z}^*_i,$$

  where $\mathcal{I}$ is the $m \times m$ identity matrix.

- The image can be reconstructed by $\{\boldsymbol{x}_{s,i}\}_{i=1}^{N_1}$ and $\{\boldsymbol{x}^*_{t,i}\}_{i=1}^{N_2}$.

Remark: For a color image, we can decompose it to RGB channels, each is a grayscale image. We use these three images to construct a common dictionary $\boldsymbol{D}$ and then use $\boldsymbol{D}$ to inpaint each channel.

Example 1: image inpainting using SR

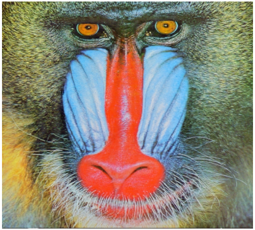
corrupted image

inpainted image

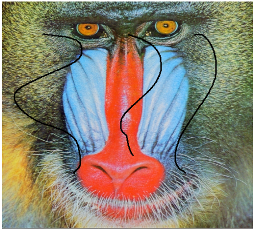
$\lambda = 0.2$ for DL and $\lambda = 0.1$ for SR.

Example 2: image inpainting using SR

corrupted image



inpainted image



corrupted image

inpainted image

References

1. D. L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution, *Communications on Pure and Applied Mathematics, 59 (2006), pp. 797-829.*
2. D. L. Donoho, For most large underdetermined systems of equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution, *Communications on Pure and Applied Mathematics,* 59 (2006), pp. 907-934.
3. V. Fedorov, G. Facciolo, P. Arias, Variational framework for non-local inpainting, *Image Processing On Line,* 5 (2015), pp. 362-386.
4. P. Getreuer, Total variation inpainting using split Bregman, *Image Processing On Line,* 2 (2012), pp. 147-157.
5. Y. Sharon, J. Wright, and Y. Ma, Computation and relaxation of conditions for equivalence between ℓ^1 and ℓ^0 minimization, *UIUC Technical Report UILU-ENG-07-2008, 2007.*