Inpainting with Sparse Linear Combinations of Exemplars: Software Implementation Details

Brendt Wohlberg brendt@lanl.gov

Abstract—**A new exemplar-based inpainting algorithm that represents the region to be inpainted as a sparse linear combination of blocks extracted from the image being inpainted is introduced in [1]. While the high-level concepts are relatively simple, the technicalities of implementation are not. These issues are discussed here in greater detail.**

I. Introduction

Exemplar based methods are becoming increasingly popular for problems such as denoising [2], [3], superresolution [4], [5], [6], texture synthesis [7], and inpainting [8], [9]. The common theme of these methods is the use of a set of actual image blocks, extracted either from the image being restored, or from a separate training set of representative images, as an image model. In the case of inpainting, the approach is usually to progressively replace missing regions with the best matching parts of the same image, carefully choosing the order in which the missing region is filled to minimize artifacts [9]. Instead, we have proposed an inpainting method that represents missing regions as sparse linear combinations of other regions in the same image [1].

II. Sparse Linear Combinations of Exemplars

The image model of the proposed approach is that each block of the restored image should be a sparse linear combination of other image blocks, either from known regions (i.e. not intersecting the inpainting region) of the image being restored, or from a separate training image set. These image blocks are overlapped to reduce blocking artifacts, and, more importantly, to enable information from the exterior of the inpainting region to propagate to blocks entirely within the interior, as illustrated in Fig. 1.

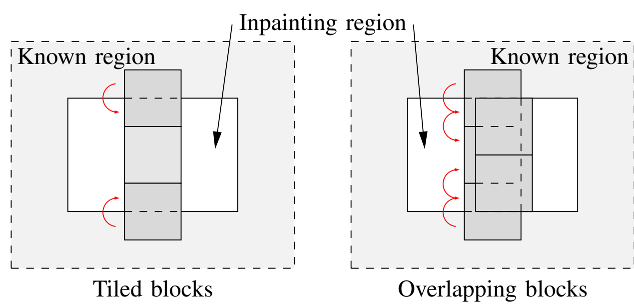

Fig. 1. Image blocks which cross the boundary of the inpainting region can be chosen as the best match to the image model subject to the constraint that they match the known image pixels outside the inpainting region, but in a tiled block structure, blocks interior to the inpainting region are unconstrained. In an overlapping block structure, an additional constraint on the mismatch between overlapping blocks allows the constraint from the known pixels to propagate to interior blocks.

Within this framework, the solution is computed by minimizing a global functional which

1) penalizes the ℓ^1 norm of the linear combination coefficients to encourage a sparse, low complexity, solution,
2) constrains (or penalizes) the mismatch between solution blocks and known pixels, and
3) constrains (or penalizes) the mismatch between overlapping parts of different solution blocks.

In order to discuss this approach in more detail, we need to establish some notation. Denote the image to be inpainted by vector $\mathbf{s}$, the inpainting region mask[1] by $\mathbf{r}$, and the inpainted result by $\mathbf{u}$. To reduce the complexity of computing the mismatch between overlapping parts of different blocks, the blocks are arranged in indexed grids, with the overlap being produced by an offset of the entire grid, as indicated in Fig. 2. This structure allows the total block overlap mismatch to be computed as the mismatch between the grids, without having to track overlapping parts of individual blocks. Since each grid does not cover the entire image, we define mask $\mathbf{g}_k$ for the region of the image covered by grid k. Each block is indexed by the number of its grid and its number within that grid, block k, l being the l^{th} block in grid k. The number of grids is N_g, the number of blocks in grid k is N_{b_k}, and the number of pixels in a block is N_p, and the number of pixels in image $\mathbf{s}$ is N_s.

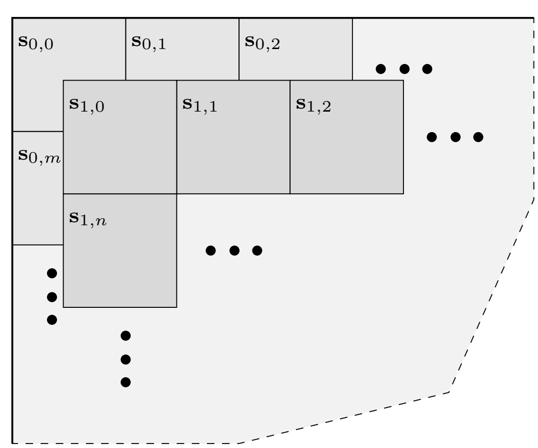

Fig. 2. Structure of overlapping block grids.

[1]This mask is an image/vector taking on the value 0 in the region of the image to be inpainted, and 1 elsewhere. Since this is really the complement of the inpainting region mask, it would perhaps be better described as the "known-region mask".

The following definitions allow linear-algebraic expressions for the functional penalties/constraints outlined at the beginning of this section:

$B_{k,l}$ "Extracts" block k,l of an image to a vector with the same number of elements as the block.

$B_{k,l}^T$ "Inserts" a block-sized vector as block k,l in a zero-valued image.

$R_{k,l}$ Apply the inpainting region mask for block k,l (i.e. zero out the pixels to be inpainted in block k,l). $R_{k,l} = \mathrm{diag}(B_{k,l}\mathbf{r})$.

G_k Apply grid k mask to an image (i.e. zero out the pixels not in grid k). $G_k = \mathrm{diag}(\mathbf{g}_k)$.

$\mathbf{s}_{k,l}$ Block k,l of the image to be inpainted. $\mathbf{s}_{k,l} = B_{k,l}\mathbf{s}$.

$\tilde{\mathbf{s}}_{k,l}$ Block k,l, with unknown (i.e. to be inpainted) region zeroed out, of the image to be inpainted. $\tilde{\mathbf{s}}_{k,l} = R_{k,l}B_{k,l}\mathbf{s}$.

$\Phi_{k,l}$ The dictionary for block k,l.

$\boldsymbol{\alpha}_{k,l}$ The dictionary coefficients for block k,l.

$\mathbf{u}_{k,l}$ The reconstruction of block k,l from its dictionary coefficients. $\mathbf{u}_{k,l} = \Phi_{k,l}\boldsymbol{\alpha}_{k,l}$.

Note that we can write

$$\mathbf{g}_k = \sum_l B_{k,l}^T B_{k,l} \left(1\; 1 \ldots\right)^T \; .$$

Now, for each block $\mathbf{u}_{k,l}$, we wish to minimize or constrain (either equal to zero, or less than some upper bound) the following terms:

1) Solution sparsity. This term is computed as the ℓ^1 norm of the coefficient vector for block k,l:
$$\|\boldsymbol{\alpha}_{k,l}\|_1$$

2) Mismatch with known pixels in $\mathbf{s}_{k,l}$. This term is the ℓ^2 norm of the difference between block k,l of the image to be inpainted and the corresponding reconstruction from $\boldsymbol{\alpha}_{k,l}$, after zeroing out of unknown pixel values by operator $R_{k,l}$:
$$\frac{1}{2}\left\|R_{k,l}\Phi_{k,l}\boldsymbol{\alpha}_{k,l} - R_{k,l}B_{k,l}\mathbf{s}\right\|_2^2$$

3) Mismatch in overlap with block grid $m \neq k$. This term measures the extent to which block k,l agrees with overlapping blocks in grid m. Writing the reconstruction of grid m as $\sum_n B_{m,n}^T \Phi_{m,n}\boldsymbol{\alpha}_{m,n}$, the part of that grid overlapping with block k,l is extracted by operator $B_{k,l}$, giving the difference between the reconstruction of block k,l an the overlapping part of grid m as $\Phi_{k,l}\boldsymbol{\alpha}_{k,l} - B_{k,l}\sum_n B_{m,n}^T \Phi_{m,n}\boldsymbol{\alpha}_{m,n}$. The final overlap mismatch term is the ℓ^2 norm of this difference, after applying operator $\mathrm{diag}(B_{k,l}\mathbf{g}_m)$ to zero out any elements of block k,l which are not in grid m:

$$\frac{1}{2}\left\|\mathrm{diag}(B_{k,l}\mathbf{g}_m)\left(\Phi_{k,l}\boldsymbol{\alpha}_{k,l} - B_{k,l}\sum_n B_{m,n}^T\Phi_{m,n}\boldsymbol{\alpha}_{m,n}\right)\right\|_2^2$$

The next step is to join these per-block penalties/constraints into combined terms for each grid, which is aided by the following compound definitions

$$\boldsymbol{\alpha}_k = \begin{pmatrix} \boldsymbol{\alpha}_{k,0} \\ \boldsymbol{\alpha}_{k,1} \\ \vdots \end{pmatrix} \quad \tilde{\mathbf{s}}_k = \begin{pmatrix} \tilde{\mathbf{s}}_{k,0} \\ \tilde{\mathbf{s}}_{k,1} \\ \vdots \end{pmatrix}$$

$$R_k = \begin{pmatrix} R_{k,0} & 0 & \cdots \\ 0 & R_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \Phi_k = \begin{pmatrix} \Phi_{k,0} & 0 & \cdots \\ 0 & \Phi_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$B_k = \begin{pmatrix} B_{k,0} & 0 & \cdots \\ 0 & B_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \; .$$

We also define

$$C_k = \left(B_{k,0}^T \; B_{k,1}^T \; \ldots\right),$$

so that

$$\sum_l B_{k,l}^T \Phi_{k,l}\boldsymbol{\alpha}_{k,l} = C_k\Phi_k\boldsymbol{\alpha}_k,$$

allowing us to express write combined terms for each grid k

1) Solution sparsity
$$\|\boldsymbol{\alpha}_k\|_1$$

2) Mismatch with $\tilde{\mathbf{s}}_k$
$$\frac{1}{2}\|R_k\Phi_k\boldsymbol{\alpha}_k - \tilde{\mathbf{s}}_k\|_2^2$$

3) Mismatch between overlapping parts of block grids k and m
$$\frac{1}{2}\|G_mC_k\Phi_k\boldsymbol{\alpha}_k - G_kC_m\Phi_m\boldsymbol{\alpha}_m\|_2^2$$

Finally, we need to compine these combined terms for each grid (or grid-pair) into global terms for all grids (or grid pairs), which is aided by the following definitions

$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}_{0,0} \\ \boldsymbol{\alpha}_{0,1} \\ \vdots \\ \boldsymbol{\alpha}_{1,0} \\ \boldsymbol{\alpha}_{1,1} \\ \vdots \end{pmatrix} \quad \tilde{\mathbf{s}} = \begin{pmatrix} \tilde{\mathbf{s}}_0 \\ \tilde{\mathbf{s}}_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{s}}_{0,0} \\ \tilde{\mathbf{s}}_{0,1} \\ \vdots \\ \tilde{\mathbf{s}}_{1,0} \\ \tilde{\mathbf{s}}_{1,1} \\ \vdots \end{pmatrix}$$

$$R = \begin{pmatrix} R_0 & 0 & \cdots \\ 0 & R_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \Phi = \begin{pmatrix} \Phi_0 & 0 & \cdots \\ 0 & \Phi_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$B = \begin{pmatrix} B_0 & 0 & \cdots \\ 0 & B_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad C = \begin{pmatrix} C_0 & 0 & \cdots \\ 0 & C_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$G = \begin{pmatrix} -G_0 & G_1 & 0 & \cdots & 0 \\ 0 & -G_1 & G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -G_{N_G-2} & G_{N_G-1} \\ G_0 & 0 & \cdots & 0 & -G_{N_G-1} \end{pmatrix}$$

Our global penalties/constraints may then be written as

1) Solution sparsity $\|\boldsymbol{\alpha}\|_1$
2) Mismatch with known pixels in $\mathbf{s}$
$$\frac{1}{2}\|R\Phi\boldsymbol{\alpha} - \tilde{\mathbf{s}}\|_2^2$$
3) Mismatch between all pairs 0-1, 1-2, etc. of overlapping block grids
$$\frac{1}{2}\|GC\Phi\boldsymbol{\alpha}\|_2^2$$

The two most obvious ways of combining these distinct goals are the unconstrained problem

$$\min \|\boldsymbol{\alpha}\|_1 + \frac{\gamma_0}{2}\|R\Phi\boldsymbol{\alpha} - \tilde{\mathbf{s}}\|_2^2 + \frac{\gamma_1}{2}\|GC\Phi\boldsymbol{\alpha}\|_2^2$$

with weights γ_0 and γ_1, and the constrained problem

$$\min \|\boldsymbol{\alpha}\|_1 \text{ s.t. } \|R\Phi\boldsymbol{\alpha} - \tilde{\mathbf{s}}\|_2 \leq \sigma_0 \text{ and } \|GC\Phi\boldsymbol{\alpha}\|_2 \leq \sigma_1$$

with upper bounds σ_0 and σ_1. For the experiments reported here, we solve the former problem using an IRLS approach to minimise the functional

$$\|A\boldsymbol{\alpha} - \mathbf{b}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1,$$

where

$$A = \begin{pmatrix} \sqrt{\gamma_0} R\Phi \\ \sqrt{\gamma_1} GC\Phi \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \sqrt{\gamma_0}\,\tilde{\mathbf{s}} \\ \mathbf{0} \end{pmatrix}.$$

In minimising this functional, it is also useful to note that

$$A^T = \begin{pmatrix} \sqrt{\gamma_0}\Phi^T R^T & \sqrt{\gamma_1}\Phi^T C^T G^T \end{pmatrix}$$

and

$$G^T = \begin{pmatrix} -G_0 & 0 & 0 & \cdots & G_0 \\ G_1 & -G_1 & 0 & \cdots & 0 \\ 0 & G_2 & -G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -G_{N_G-2} & 0 \\ 0 & 0 & \cdots & G_{N_G-1} & -G_{N_G-1} \end{pmatrix}.$$

The final restored image (or image region, since the computational domain can be restricted to a small region around the inpainting region) is obtained by averaging the overlapping block grids to obtain a single value for each image pixel.

III. Efficient Formulation

While the conceptually simplest approach, the above formulation exhibits two distinct disadvantages

- Computational inefficiency: the expression $R\Phi\boldsymbol{\alpha} - \tilde{\mathbf{s}}$ has dimensionality $N_p \sum_{k=0}^{N_g-1} N_{b_k}$ and the expression $GC\Phi\boldsymbol{\alpha}$ has dimensionality $N_s N_g$, so that even an in-principle small problem involving a small inpainting region requires computational resources on the scale of the entire image.
- Numerical issues: As a result of the masking operators R_k and G_k, the operators involved in minimization of the final functional have zero rows, which can lead to ill-conditioning and convergence problems.

Here we describe a more efficient (but more conceptually complicated) formulation of the problem which involves expressions of significantly lower dimensionality. Define the following operators

$B_{k,l}$	"Extract" block k, l from an image
$B_{k,l}^T$	"Insert" block k, l into a zero image.
$R_{k,l}$	"Extract" known part of block k, l.
$R_{k,l}^T$	"Insert" known part of block k, l into a zero block.
$Q_{k,l}$	"Extract" block k, l from vector representing unknown region of image, inserting zeros where block does not intersect inpainting region. The result is a zero-vector if block k, l has no intersection with the inpainting region.
$Q_{k,l}^T$	"Insert" block k, l into zero vector representing inpainting region. The result is a zero-vector if block k, l has no intersection with the inpainting region.
G_k	Apply a mask to the inpainting region vector, zeroing out any pixels which are not in grid k. $G_k = \mathrm{diag}\left(\sum_l Q_{k,l}^T Q_{k,l} \,(1\,1\ldots)^T\right)$.

Note that, in this formulation, operator $R_{k,l}$ maps block k, l to a vector which excludes any unknown (to be inpainted) pixels in the block, as opposed to the previous definition of $R_{k,l}$ which zeroed these unknown pixels but retained the original block dimensionality.

We also define

$$\boldsymbol{\alpha}_k = \begin{pmatrix} \boldsymbol{\alpha}_{k,0} \\ \boldsymbol{\alpha}_{k,1} \\ \vdots \end{pmatrix} \quad \Phi_k = \begin{pmatrix} \Phi_{k,0} & 0 & \cdots \\ 0 & \Phi_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$R_k = \begin{pmatrix} R_{k,0} & 0 & \cdots \\ 0 & R_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad B_k = \begin{pmatrix} B_{k,0} & 0 & \cdots \\ 0 & B_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

and

$$Q_k^T = \begin{pmatrix} Q_{k,0}^T \; Q_{k,1}^T \; \ldots \end{pmatrix},$$

allowing us to express penalty/constraint 2) for grid k as

$$\frac{1}{2}\|R_k \Phi_k \boldsymbol{\alpha}_k - R_k B_k \mathbf{s}\|_2^2$$

and penalty/constraint 3) for grids k and m as

$$\frac{1}{2}\left\| G_m \sum_l Q_{k,l}^T \Phi_{k,l}\boldsymbol{\alpha}_{k,l} - G_k \sum_n Q_{m,n}^T \Phi_{m,n}\boldsymbol{\alpha}_{m,n} \right\|_2^2 = \frac{1}{2}\left\| G_m Q_k^T \Phi_k \boldsymbol{\alpha}_k - G_k Q_m^T \Phi_m \boldsymbol{\alpha}_m \right\|_2^2$$

Now, defining

$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha}_0 \\ \boldsymbol{\alpha}_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha}_{0,0} \\ \boldsymbol{\alpha}_{0,1} \\ \vdots \\ \boldsymbol{\alpha}_{1,0} \\ \boldsymbol{\alpha}_{1,1} \\ \vdots \end{pmatrix}$$

$$R = \begin{pmatrix} R_0 & 0 & \cdots \\ 0 & R_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \Phi = \begin{pmatrix} \Phi_0 & 0 & \cdots \\ 0 & \Phi_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$B = \left(\begin{array}{ccc} B_0 & 0 & \cdots \\ 0 & B_1 & \cdots \\ \vdots & \vdots & \ddots \end{array} \right) \quad Q^T = \left(\begin{array}{ccc} Q_0^T & 0 & \cdots \\ 0 & Q_1^T & \cdots \\ \vdots & \vdots & \ddots \end{array} \right)$$

$$G = \left(\begin{array}{cccccc} -G_0 & G_1 & 0 & & \cdots & 0 \\ 0 & -G_1 & G_2 & & \cdots & 0 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & 0 & \cdots & & -G_{N_G-2} & G_{N_G-1} \\ G_0 & 0 & \cdots & & 0 & -G_{N_G-1} \end{array} \right)$$

we can express penalty/constraint 2) over all grids as

$$\frac{1}{2} \| R \Phi \boldsymbol{\alpha} - R B \mathbf{s} \|_2^2$$

and penalty/constraint 3) for grid pairs 0-1, 1-2, etc. as

$$\frac{1}{2} \left\| G Q^T \Phi \boldsymbol{\alpha} \right\|_2^2 \; .$$

In this formulation, penalty/constraint 2) has somewhat lower dimensionality than in the original formulation, and the associated operator in the minimisation problem no longer has zero rows, and penalty/constraint 3) has a much lower dimensionality, being equal to the dimensionality of the inpainting region multiplied by N_g.

Using these definitions, we have

$$A = \left(\begin{array}{c} \sqrt{\gamma_0} R \Phi \\ \sqrt{\gamma_1} G Q^T \Phi \end{array} \right) \quad \mathbf{b} = \left(\begin{array}{c} \sqrt{\gamma_0} R B \mathbf{s} \\ \mathbf{0} \end{array} \right)$$

and

$$A^T = \left(\begin{array}{cc} \sqrt{\gamma_0} \Phi^T R^T & \sqrt{\gamma_1} \Phi^T Q G^T \end{array} \right) \; .$$

References

[1] Brendt Wohlberg, “Inpainting with sparse linear combinations of exemplars,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 689–692.

[2] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, “Image denoising by non-local averaging,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, USA, March 2005, vol. 2, pp. 25–28.

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering.,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 3, March 2002.

[5] Simon Baker and Takeo Kanade, “Limits on super-resolution and how to break them,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1167–1183, 2002.

[6] Michael Elad and Dmitry Datsenko, “Example-based regularization deployed to super-resolution reconstruction of a single image,” *The Computer Journal*, 2007, DOI: 10.1093/comjnl/bxm008.

[7] Alexei A Efros and Thomas K. Leung, “Texture synthesis by non-parametric sampling,” in *IEEE International Conference on Computer Vision (ICCV)*, Kerkyra, Corfu, Greece, September 1999, pp. 1033–1038.

[8] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun, “Fragment-based image completion,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 303–312, 2003.

[9] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.