Sequential mask peeling inpainting algorithm

Filippini Luca Teodoro* Porvazník Michal† Trujić Miloš‡

Abstract

Image inpainting is relatively young field of research, nonetheless a large variety of techniques has been developed. The aim of the inpainting is to restore images which suffered an information loss and to reconstruct the missing parts in order for the reconstructed image to look natural. Many different approaches of the image inpainting can be found in the literature, ranging from sparse decomposition of signals over dictionaries to *Partial Differential Equations* (PDEs). In this paper we present an algorithm which combines the idea of signal diffusion through the boundary of damaged regions together with decomposing overlapping image patches using overcomplete dictionaries.

We compare our algorithm with standard dictionary based approaches, namely *Matching Pursuit* and *Orthogonal Matching Pursuit*, to demonstrate its superior ability to reconstruct images.

1 Introduction

The process of filling the missing regions of an image from the surroundings is known as image inpainting. Although a young field of research, it goes way back to renaissance and art restoration, also known as *retouching*. Image inpainting has various applications such as restoration of old images or removing unwanted elements from the image. In digital domain it is also present in telecommunication, where the need was to fill-in the image blocks that had been damaged during data transmission. The main idea behind the image inpainting is to modify the damaged or unwanted regions so as to appear undetectable to the neutral observer.

Throughout the years many techniques have been developed [9]. In particular we discuss two prominent approaches: patch-based sparse representations over dictionaries and PDE based approaches. In this paper we will present an algorithm which combines some aspects of both approaches to surpass traditional dictionary based methods.

1.1 Sparse Coding

Sparse Coding follows an idea of using an overcomplete dictionary matrix $\mathbf{D} \in \mathbb{R}^{n\times K}$ consisting of K atoms $\{\mathbf{d}_i\}_{i=1}^{K}$ as its columns. For each vectorised image patch $\mathbf{x} \in \mathbb{R}^n$, we assume that the signal $\mathbf{x}$ can be represented as a sparse linear combination of the atoms from $\mathbf{D}$. The signal is represented as $\mathbf{x} \approx \mathbf{Dz}$ where the vector $\mathbf{z} \in \mathbb{R}^K$ corresponds to the representation coefficients of the signal $\mathbf{x}$. Assuming we have a fixed dictionary allowing such decomposition, we only need to find the coding vector $\mathbf{z}$. This leads to the following two formulations of an optimisation problem:

$$\min_{\mathbf{z}} \|\mathbf{z}\|_0 \qquad \text{subject to } \|\mathbf{x}-\mathbf{Dz}\|_2 \le \varepsilon, \text{ or} \tag{1}$$

$$\min_{\mathbf{z}} \|\mathbf{x}-\mathbf{Dz}\|_2 \qquad \text{subject to } \|\mathbf{z}\|_0 \le L \tag{2}$$

for a given ε or L. The above optimisation problem is known to be NP-hard due to its non-convexity [4]. Therefore, a greedy approximation algorithm is employed. The most standard version found in the literature is the *Matching Pursuit* (MP) [6] which at each stage selects the atom from the dictionary maximising the inner product with the residual signal, until maximal number of coefficients is used or the energy of residual is sufficiently small. *Orthogonal Matching Pursuit* (OMP) [10] extends MP by projecting the signal onto the subspace spanned by chosen atoms to get the coefficients at each stage. More advanced algorithms such as *Basis Pursuit* and *Focal Under-determined System Solver* have been used to obtain slightly better results at higher computational cost [5, 8].

A great deal of research has been dedicated to finding suitable dictionaries. Although standard dictionaries such as the overcomplete *Discrete Cosine Transform* or *Haar Wavelet Transform* can be used to obtain satisfactory results, algorithms that learn a dictionary from a set of images have lead to improvements. Among the most popular of these is the K-SVD dictionary learning algorithm [1], which we used in our implementation as well.

1.2 Partial Differential Equation based approaches

In image inpainting we are given an area Ω to be restored. Various PDE based methods have been developed for this task and we do not discuss particular models here. Instead we focus on an idea most of them share - diffusion process that propagates the signal from the mask boundary $\partial\Omega$ inside Ω [7]. At each stage $\partial\Omega$ (boundary pixel layer) is inpainted using information from its neighbourhood outside Ω and removed from the mask. This process is iterated until the mask is empty. The iterative nature of these algorithms results in runtime of several minutes, nevertheless these methods yield visually very impressive results. Even when the unknown regions are large they produce visually plausible reconstructions, which is not the case with dictionary based methods. The reason for this is presumably the iterative inward propagation of signal.

In this paper we present an algorithm that couples this idea with a dictionary based approach further boosted by use of overlapping image patches that are weighted based on their *signal-to-noise ratio* to reconstruct the image. Furthermore, our methods mostly focus on the black and white images, with a possible extension to fully coloured RGB images, which we will not discuss here.

The paper is organised as follows. In Section 2 we describe our algorithms in detail. Section 3 is devoted to

*Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland. E-mail: fteodoro@student.ethz.ch

†Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland.E-mail: pmichal@student.ethz.ch

‡Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland.E-mail: mtrujic@student.ethz.ch

experimental results, followed by discussion of other approaches that we tried in Section 4. In Section 5 we conclude this paper.

2 Our method

In this section we describe our algorithm in detail. To this end, we first explain basic dictionary based inpainting process.

2.1 Standard dictionary based inpainting algorithm

Given a grayscale image and a binary mask of the same size, we first extract vectorised signals from $\sqrt{n} \times \sqrt{n}$ patches from both the image and the mask. Extracted patches can be chosen to be either disjoint or overlapping, in which case some care needs to be taken when image is reconstructed. This leads to matrices $\mathbf{X}$ and $\mathbf{M} \in \mathbb{R}^{n \times N}$, where $\mathbf{X}$ stores the masked image, $\mathbf{M}$ the binary mask, and N is the number of signals extracted from the image. At this stage a pursuit algorithm is employed to approximate each column $\mathbf{x}^{(i)}$ of $\mathbf{X}$, however the mask needs to be applied to dictionary atoms to prevent the approximation algorithm to learn the damaged part of the image. Mathematically, we let the pursuit algorithm solve:

$$\mathbf{z}^{(i)} = \arg\min_{\mathbf{z}} \|\mathbf{x}^{(i)} - (\mathbf{D} \circ \mathbf{M}^{(i)})\mathbf{z}\|_2 \qquad \text{s.t. } \|\mathbf{z}\|_0 \leq L$$

where $\mathbf{M}^{(i)}$ is the matrix obtained by appending K copies of the i-th column of $\mathbf{M}$ to match the dimension of $\mathbf{D}$, and the $\circ$ operator denotes element-wise multiplication. Since $\mathbf{M}$ is a binary matrix with zeroes wherever the image is damaged, the pursuit algorithm will only fit to the known pixels. Once the pursuit algorithm recovers the coding matrix $\mathbf{Z}$, we reconstruct the the signal matrix $\widetilde{\mathbf{X}} = \mathbf{DZ}$ using the full, unmasked dictionary. It only remains to transform the vectorised signals back to image patches.

The underlying assumption of this approach is that patches of natural images belong to a restricted set of signals, such that whenever we can sparsely approximate the known pixels, the unknown pixels will form a visually convincing continuation of the known part.

We now describe our two extensions that build on the above algorithm.

2.2 Iterative mask peeling

As mentioned in the introduction, the idea behind this method is gradual propagation of signal from the boundary of the mask to its interior. At first, number of masked neighbours is precomputed for each pixel in the mask and is updated after each iteration. Furthermore, the above inpainting algorithm is employed to approximate the masked pixels, after which the boundary of the mask is removed (peeled), and the removed pixels are considered as known. This process is repeated until the mask has been sufficiently reduced. The pixels that belong to the boundary are chosen to be the masked pixels having less than a predetermined number of neighbours.

This extension leads to improved results over the algorithm described above, however each iteration performs at least as much computation as the former, leading to significantly increased runtime.

2.3 Combining overlapping patches

The reason for extracting overlapping patches is to avoid creating patchy artefacts to which human vision is perceptive. When disjoint patches are extracted from the image the reconstruction is straightforward, as each pixel belongs to exactly one patch. However, once we extract overlapping patches, some pixels will be covered by several patches and therefore an approach to combine these needs to be developed. Although taking the mean of pixel values over patches that cover it can be used, we found that weighting contributions of different pixels by proportion of known pixels within their respective patches leads to better results. This technique is motivated by the intuition that approximation of a missing pixel is better in a patch with higher signal-to-noise ratio. Formally, let $p := (i, j)$ denote a pixel location and let $v(p)$ denote its value in $[0, 1]$ range. For every patch P define a weight:

$$w(P) := \#\text{ of known pixels in } P.$$

Define by $v(P, p)$ the value of pixel in patch P corresponding to p. Furthermore, let $\mathcal{S}(p)$ be the set of all patches covering p. Then we compute the value of pixel p as follows:

$$v(p) = \frac{\sum_{P \in \mathcal{S}(p)} w(P) v(P, p)}{\sum_{P \in \mathcal{S}(p)} w(P)}.$$

This approach yields improved reconstruction both visually and as measured by the mean squared error, however the runtime grows linearly with the number of extracted patches, as the pursuit algorithm is ran for each signal separately. We now provide the pseudo-code of our algorithm.

Algorithm 1 Sequential mask peeling algorithm

Input: $\mathbf{I}, \mathbf{mask} \in \mathbb{R}^{p \times q}, \mathbf{D} \in \mathbb{R}^{n \times K}, T, C \in \mathbb{R}, L \in \mathbb{N}$
Output: $\tilde{\mathbf{I}} \in \mathbb{R}^{p \times q}$

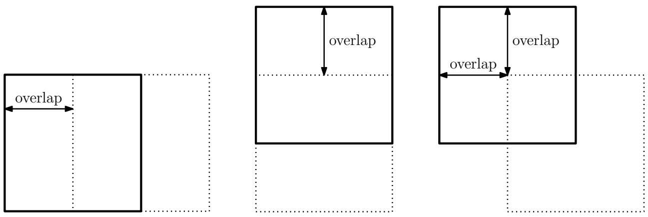
$\tilde{\mathbf{I}} \leftarrow \mathbf{I}$
$\mathbf{\Gamma} \leftarrow$ number of missing pixels in the 8-pixel neighbourhood of each pixel
while (# of zero elements in **mask** $> T$) **do**
  $\mathbf{X} \leftarrow$ extract patches from $\tilde{\mathbf{I}}$
  $\mathbf{M} \leftarrow$ extract patches from **mask**
  $\mathbf{Z} \leftarrow OMP(\mathbf{X}, \mathbf{M}, \mathbf{D}, L)$
  $\partial\Omega \leftarrow$ set of pixels (i, j) s.t. $\mathbf{\Gamma}_{i,j} < C$
  $\mathbf{mask}(\partial\Omega) \leftarrow 1$
  update the neighbourhood matrix $\mathbf{\Gamma}$
  $\tilde{\mathbf{X}} \leftarrow \mathbf{DZ}$
  $\mathbf{I}' \leftarrow \text{reconstruct}(\tilde{\mathbf{X}})$
  $\mathbf{I}'(\mathbf{mask} \neq 0) = \tilde{\mathbf{I}}(\mathbf{mask} \neq 0)$
  $\tilde{\mathbf{I}} = \mathbf{I}'$
end while
return $\tilde{\mathbf{I}}$

The parameters T, C, and L (sparsity constraint) need to be set upfront and varying them leads to different results. In Section 3 we will provide the values used to obtain our results.

3 Results

In this section we present the results obtained with our algorithm and compare it with the MP and OMP. Initially we used overcomplete Discrete Cosine Transform and Haar Wavelet Transform dictionaries. However, applying the K-SVD algorithm [1] to a set of 6144 non-overlapping

16 × 16 patches extracted from 6 images, led to a dictionary which produced better results. In our K-SVD implementation we used OMP to solve the problem (2) with the sparsity constraint $L = 5$ and the number of atoms $K = 512$. Subsequently, as suggested by [2], we pruned the dictionary by removing one atom from every pair of atoms having correlation above 0.8. The values of the parameters chosen for Algorithm 1 were the following: $T = \frac{1}{10} \cdot (\#$ of 0-elements in the original mask$)$, $C = 6$, and $L = 10$. One of the key features of our algorithm is the extent to which the patches overlap. The parameter *overlap* represents horizontal or vertical distance between two neighbouring patches, as seen in the figure below:



(a) Horizontal **(b)** Vertical **(c)** Combined

We now present the results obtained on three different mask types. Figures 2, 4, and 6 were obtained using set of 6 images different from the training set used for the K-SVD dictionary learning algorithm.

3.1 Text mask

The first result we turn to is one of the most common examples found in the literature. We look at the mask of missing pixels in form of text that spans across the image. In the plot below we show how our method of extracting weighted overlapping patches produces significant improvement, even when applied to the basic versions of MP and OMP algorithms. On top of this, it can be seen that the introduction of mask peeling in our algorithm produces slightly better results as well.

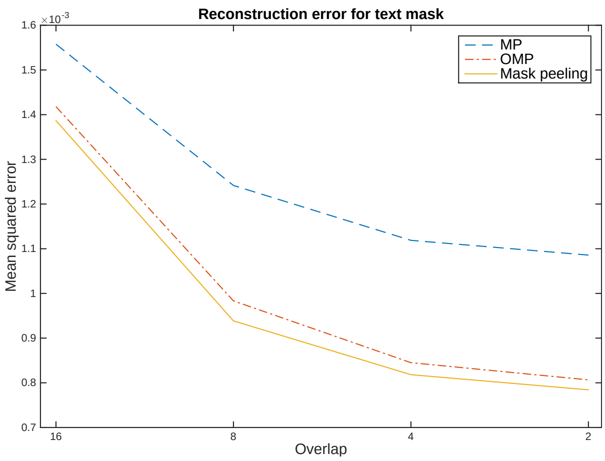


Figure 2: Comparison of MP, OMP, and mask peeling, all three using the idea of weighted overlapping patches of size 16 × 16. Note that the overlap of 16 stands for no overlap.

Moreover, we show the visual results obtained by our algorithm on the standard Lena image covered with the text mask.

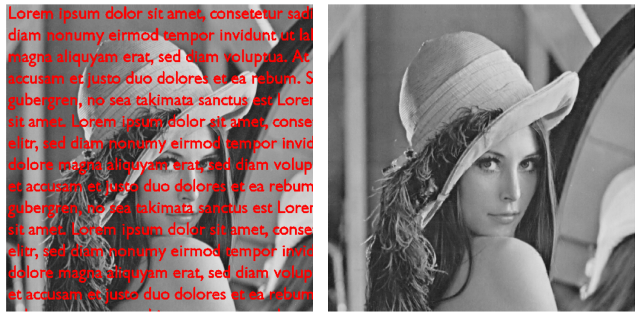


Figure 3: Removal of text mask done by our algorithm. Note that the red text is shown for a clearer presentation, the missing pixels were originally black.

3.2 Artificial scratches mask

As a second example of the inpainting technique we show an image with the mask created as artificial scratches. This problem is also commonly addressed in the literature, and can be viewed as a problem of removing folding marks from scans of old photographs or incidental drawings across the image. The figure below is analogous to Figure 2.

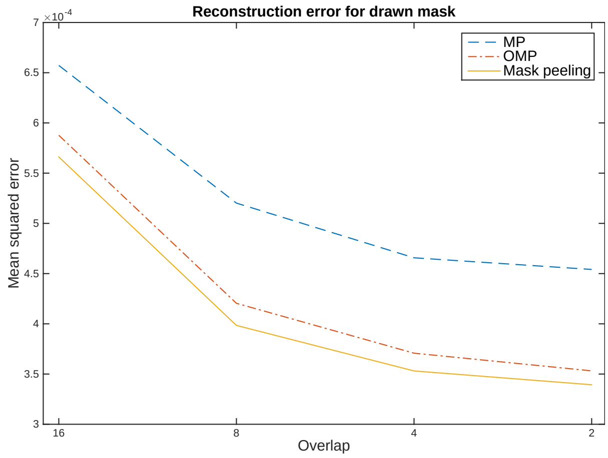


Figure 4: Comparison of MP, OMP, and mask peeling, all three using the idea of weighted overlapping patches of size 16 × 16. Note that the overlap of 16 stands for no overlap.

The visual results obtained can be seen on the Lena image with the artificial hand-produced mask representing scratching marks.

Figure 5: Removal of the artificial scratches mask done by our algorithm. Note that the red marks are shown for a clearer presentation, the missing pixels were originally black.

3.3 Random missing pixels mask

The last result we are going to present is the application of our algorithm to a noisy image. A random mask with increasing level of pepper noise had been applied to the image after which we ran the usual MP and OMP algorithms on non-overlapping patches (overlap = 16). We compared this to our full combined approach of mask peeling and overlapping patches (overlap = 2). Since randomness was involved, we ran all three algorithms 9 times per image, and plotted the mean of the obtained results. Furthermore, we show the error bars of one standard deviation, from which it can be seen that our results are stable under noise realisations.

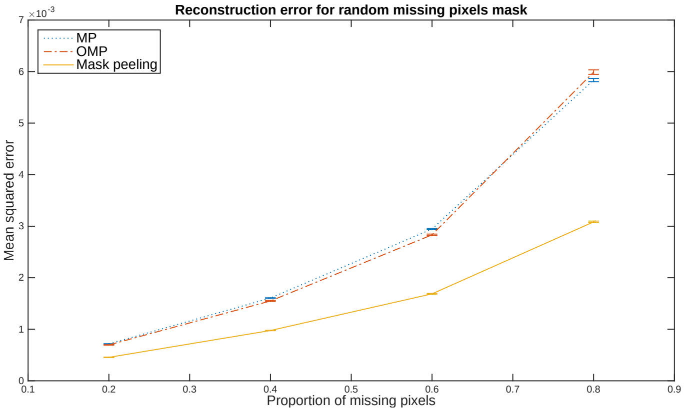


Figure 6: Comparison of MP, OMP, and our approach on different levels of random pepper noise.

Figure 7: Results obtained by our algorithm on the Lena image covered with 80% of random pepper noise.

4 Discussion

In this section we will comment on the results followed by a brief discussion of strengths and weakness of our approach.

As seen in Figures 2, 4, when applied to text or artificial scratches mask, our algorithm outperforms the two baselines. Specifically, the introduction of the overlapping patches is responsible for most of the improvement, however the iterative mask peeling slightly improves this further for every value of overlap. In Figure 6 we compare the baselines with our full combined approach of mask peeling and overlapping patches, observing almost two-fold improvement at every noise level.

In all three cases our algorithm produces visually satisfactory results, nevertheless in Figure 7, with 80% of random pepper noise, we begin to observe blurriness.

Although, our algorithm surpasses the baselines in reconstruction error it comes with a significantly higher computational cost. The results were produced using a Dual-core 2.5GHz IntelCore-i5 with 8GB DDR3 RAM. Without using the overlap, the runtimes of MP, OMP, and mask peeling respectively were: 1.52s, 1.42s, and 4.51s. However, using overlap = 2 increases the number of signals by a factor of 62 resulting in the runtime of 264.33s.

Finally, the motivation of mask peeling was to overcome the inability of dictionary based inpainting techniques to reconstruct images with large damaged blocks, e.g. removing a large black square. This idea was driven by the fact that similar PDE based signal propagation schemes are able to reconstruct such images [3]. However, our algorithm could not produce satisfactory results. The reason behind this is that along with the signal some proportion of the noise was passed on to the boundary layer at each iteration, thus resulting in the noise being subsequently reproduced in the next boundary layer. Whenever the number of layers becomes large enough, as is the case with large damaged blocks, this phenomenon leads to the noise dominating the signal in the interior of the mask. Further research is needed to overcome this issue.

5 Conclusion

In this paper we presented a novel algorithm combining the core ideas from two major approaches to image inpainting, namely sparse representation over overcomplete dictionaries with gradual inward propagation of signal, as used in PDE approaches. Although we demonstrated superior image restoration abilities of this approach when compared to Matching Pursuit and Orthogonal Matching Pursuit, we conclude that the high computational cost of this algorithm outweights the benefits of the restoration capability.

References

[1] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD: Design of dictionaries for sparse representation*. Proceedings of SPARS 5 (2005), 9-12.

[2] M. Aharon, *Overcomplete dictionaries for sparse representation of signals*. PhD thesis, Israel Institute of Technology, (2006).

[3] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*. Image Processing, IEEE Transactions on 12.8 (2003): 882-889.

[4] G. Davis, S. Mallat, and M. Avellaneda, *Adaptive greedy approximations*, Journal of Constructive Approximation 13.1 (1997): 57-98.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*. SIAM journal on scientific computing 20.1 (1998): 33-61.

[6] S. Mallat and Z. Zhang. *Matching pursuits with time-frequency dictionaries*. Signal Processing, IEEE Transactions on 41.12 (1993): 3397-3415.

[7] M. M. Oliveira, B. Bowen, R. McKenna, and Y. Chang, *Fast digital image inpainting*. Proceedings of the International Conference on Visualization, Imaging and Image Processing (2001): 261-266.

[8] B. D. Rao and K. Kreutz-Delgado, *An affine scaling methodology for best basis selection*. Signal Processing, IEEE Transaction on 47.1 (1999): 187-200.

[9] S. Ravi, P. Pasupathi, S. Muthukumar, and N. Krishnan, *Image in-painting techniques - a survey and analysis*, Innovations in Information Technology (IIT), 2013 9th International Conference on (2013): 36-41.

[10] J. A. Tropp, *Greed is good: Algorithmic results for sparse approximation*. Information Theory, IEEE Transactions on 50.10 (2004): 2231-2242.