SPARSITY BASED POISSON INPAINTING

Raja Giryes and Michael Elad

Department of Computer Science, The Technion - Israel Institute of Technology Haifa, 32000, Israel

ABSTRACT

Poisson noise appears in various imaging applications, such as low-light photography, medical imaging and space imaging. In many cases we may have occlusions in the received image in addition to the noise. Thus, the problem of Poisson denoising turns to be a Poisson inpainting one in which we need both to remove the noise and recover the values in the occluded locations. In this work we extend a recent novel Poisson denoising method for the task of image inpainting. To the best of our knowledge this is the first work that deals with the problem of Poisson inpainting.

Index Terms— Sparse Approximation, Poisson Denoising, Inpainting, Dictionary Learning, Greedy Methods

1. INTRODUCTION

Poisson noise appears in many applications such as night vision, computed tomography (CT), fluorescence microscopy, astrophysics and spectral imaging. In many of these applications occlusions occur in addition to the noise. Thus, in addition to the task of noise removal there is a need to recover the values of the missing entries. The problem we have to solve is the one of Poisson inpainting, which is a combination of Poisson denoising and image inpainitng.

Let $\mathbf{x}$ be the original clean image (represented as a column-stacked vector). In the standard Poisson denoising problem (no missing pixels) we are given a Poisson noisy image $\mathbf{y}$ which is a Poisson distributed random vector with mean and variance equal to $\mathbf{x}$. Many schemes for recovering $\mathbf{x}$ from $\mathbf{y}$ exist [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Some rely on variance stabilizing transformations such as Anscombe [12] and Fisz [13], that approximately convert the Poisson denoising problem into a Gaussian one, for which plenty of methods exist (e.g. [14, 15]). Other methods rely directly on the noise statistics for recovering the original image. These are more effective in very strong noise cases, where the stabilizing transformations become much less effective [1, 2].

In the inpainting problem we have a diagonal masking matrix $\mathbf{M}$ that contains zeros and ones on its diagonal. The ones and zeros define valid and invalid entries in the measured image $\mathbf{y}$. The existing inpainting schemes assume that the known entries in $\mathbf{y}$ contain either the corresponding values in $\mathbf{x}$ or a Gaussian noisy version of them [16, 17, 18, 19].

In this work we treat the task of Poisson inpainting that combines both the Poisson denoising and inpainting problems. We treat the case where the measurements have very low SNR, which corresponds to small peak (maximal intensity) value in the original image. We extend our recently proposed sparsity poisson denoising algorithm (SPDA) [2], which achieves state-of-the-art denoising performance in this setup, for the task of Poisson inpainting. To the best of our knowledge, this is the first attempt to handle this problem.

This paper is organized as follows. Section 2 presents our new Poisson inpainting method. Due to space limits we focus only on the parts that differ from SPDA. Section 3 contains some experimental results and Section 4 concludes the paper.

2. THE POISSON INPAINTING ALGORITHM

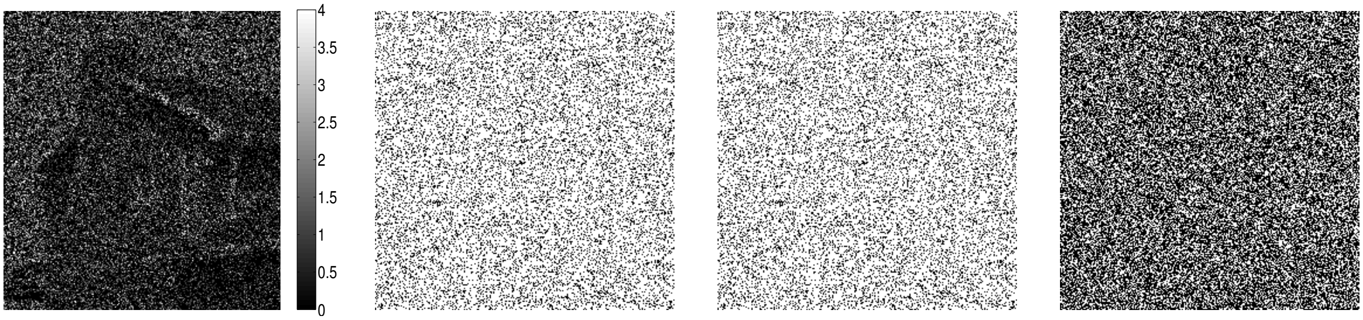
Before presenting our extension to SPDA [2], we start with a brief description of this method. As it is a patch based strategy, it extracts overlapping patches from the noisy image and processes each of them using the assumption that each has a sparse representation under a given dictionary $\mathbf{D}$. The algorithm is iterative and in each iteration it gets a new recovery for each patch, by decoding its representation under the dictionary, and then updates the dictionary by a dictionary learning technique from [20]. The recovered image is a result of returning each reconstructed patch to the place it was taken from and averaging the pixels that fall in the same place.

Our Poisson inpainting algorithm consists of the following four steps: (i) Patch grouping; (ii) Sparse coding; (iii) Poisson noise estimation; and (iv) Dictionary learning. For the last step we utilize the same learning technique as in SPDA. The reader may refer to [2] for more details. We turn to explain the first three steps.

2.1. Patch Grouping

In SPDA the reconstruction strategy relies on decoding the representation of each patch. However, decoding each patch alone is not likely to give a good recovery since the patches are very noisy as can be seen in Fig. 1(a). For this reason

R. Giryes thanks the Azrieli Foundation for the Azrieli Fellowship. This research was supported by European Communitys FP7- ERC program, grant agreement no. 320649.



(a) Noisy Image (b) 20% Missing Pixels Mask (c) 40% Missing Pixels Mask (d) 60% Missing Pixels Mask

Fig. 1. Noisy *house* image with peak = 2 and masks for 20%, 40% and 60% missing pixels (black denotes the missing pixels).

SPDA uses the assumption that similar patches can be represented by the same dictionary atoms, i.e., have the same support in their representation. Noisy patches that correspond to similar patches in the original image can be recovered by requiring them to have the same support. With this joint sparsity constraint each of them is likely to be recovered better.

In order to use the above assumption a patch clustering method is required. SPDA uses a greedy method for this purpose [2]. Though being suboptimal, it is fast and seems to be sufficient for our needs. It sequentially divides the patches into groups by selecting each time a random patch and adding to its group a constant number of patches which are closest to it. The distance between patches is calculated by applying a Gaussian blur on the noisy image and then using the euclidean distance. Alternatively, if an estimate for the image is available then it can be used for the distance calculation.

For adapting the above method to patches with missing pixels, we replace each missing value with a weighted average of its surrounding pixels before applying the Gaussian blur. As neighbors of missing pixels can also be missing, we initialize their values with zero and repeat the update process several times till we get convergence. Note that these estimated values serve only for calculating the distances between patches in the clustering process.

2.2. Sparse Coding

Given a group of noisy patches $\{\mathbf{q}_1, \dots, \mathbf{q}_l\}$ we would like to recover their sparse representation under the dictionary $\mathbf{D}$. By maximizing the log-likelihood of the Poisson distribution and using the assumption that each patch $\mathbf{p}_i$ of the original image has a sparse representation $\boldsymbol{\alpha}_i$ under the dictionary $\mathbf{D}$ in an exponential model $\mathbf{p}_i = \exp(\mathbf{D}\boldsymbol{\alpha}_i)$, we get the following minimizing problem in the denoising case [1, 2]:

$$\min_{\boldsymbol{\alpha}_i} \mathbf{1}^* \exp(\mathbf{D}\boldsymbol{\alpha}_i) - \mathbf{q}_i^* \mathbf{D}\boldsymbol{\alpha}_i \quad s.t. \quad \|\boldsymbol{\alpha}_i\|_0 \le k. \quad (1)$$

Notice that with joint sparsity the minimization should be done over all the patches in the group with the restriction that all the representations have the same support.

Algorithm 1 Poisson Inpainting Greedy Algorithm

Require: $k, \mathbf{M}, \mathbf{D} \in \mathbb{R}^{d\times n}, \{\mathbf{q}_1, \dots, \mathbf{q}_l\}$ where $\mathbf{q}_i \in \mathbb{R}^d$ is a Poisson distributed vector with mean and variance $\exp(\mathbf{D}\boldsymbol{\alpha}_i)$ at the locations that the values in $\mathbf{q}_i$ are valid ($\mathbf{M}$ has one on its diagonal) and with unknown value at the other locations, and k is the maximal cardinality of $\boldsymbol{\alpha}_i$. All representations $\boldsymbol{\alpha}_i$ are assumed to have the same support. Optional parameter: Estimates of the true image patches $\{\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_l\}$.

Ensure: $\hat{\mathbf{p}}_i = \exp(\mathbf{D}\hat{\boldsymbol{\alpha}}_i)$ an estimate for $\exp(\mathbf{D}\boldsymbol{\alpha}_i)$.

```
Begin Algorithm:
-Initialize the support T^0 = ∅ and set t = 0.
-Form {M_1, ..., M_l}: the submatrices of M that correspond to the patches {q_1, ..., q_l} respectively.
while t < k do
    -Update iteration counter: t = t + 1.
    -Set initial objective value: v_o = inf.
    for j = 1 : n do
        -Check atom j: T̃^t = T^{t-1} ∪ {j}.
        -Calculate current objective value: v_c =
        min_{α_1,...,α_l} Σ_{i=1}^l 1* exp(D_{T̃^t} α_i) − q_i* M_i* D_{T̃^t} α_i
        if v_o > v_c then
            -Update selection: j^t = j and v_o = v_c.
        end if
    end for
    -Update the support: T^t = T^{t-1} ∪ {j^t}.
    -Update representation estimate: [α̂_1^t, ..., α̂_l^t] =
    argmin_{α_1,...,α_l,j} Σ_{i=1}^l 1* exp(D_{T^t} α_i) −
    q_i* M_i* D_{T^t} α_i.
    if {p̃_1, ..., p̃_l} are given then
        -Estimate error: e_t = Σ_{i=1}^l ||exp(D α̂_i^t) − p̃_i||_2^2.
        if t > 1 and e_t > e_{t-1} then
            -Set t = t − 1 and break (exit while and return the
            result of the previous iteration).
        end if
    end if
end while
-Form the final estimate p̂_i = exp(D α̂_i^t), 1 ≤ i ≤ l.
```

In inpainting the decoding can be made only using the valid information. Notice that for each entry in $\mathbf{q}_i$ there is a corresponding row in the dictionary $\mathbf{D}$. Thus, for adapting (1) to inpainting we should remove the rows related to the missing entries at least from the second element. We can remove them also from the first one but we decide not to do so. Denoting by $\mathbf{M}_i$ the submatrix of $\mathbf{M}$ that corresponds to $\mathbf{q}_i$ we get the following minimization problem:

$$\min_{\boldsymbol{\alpha}_i} \mathbf{1}^* \exp(\mathbf{D}\boldsymbol{\alpha}_i) - \mathbf{q}_i^* \mathbf{M}_i^* \mathbf{D}\boldsymbol{\alpha}_i \quad s.t. \quad \|\boldsymbol{\alpha}_i\|_0 \leq k. \tag{2}$$

Note that this minimization problem is likely to be NP-hard and thus an approximation strategy is needed. In [2] a greedy algorithm has been proposed for approximating (1) with the joint sparsity assumption. A modified version of that algorithm for the inpainting minimization problem in (2) is presented in Algorithm 1. The output of this algorithm provides us with an initial estimate for each patch and therefore (by averaging) for the whole image also.

Fig. 2. Test images used in this paper. From left to right: Flag, House, Peppers and Ridges.

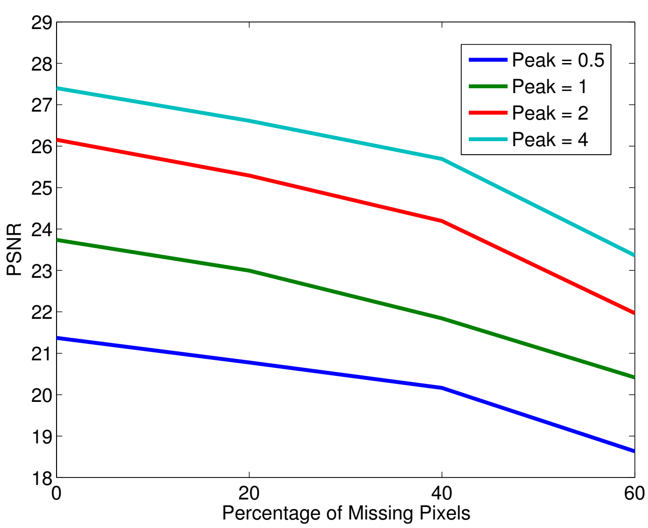


Fig. 3. Average PSNR as function of the percentage of missing pixels for different peak values.

2.3. Poisson Noise Estimation

As we have modified the sparse coding scheme for the purpose of inpainting we can do the same for the dictionary learning strategy used in [2]. However, We choose another route: instead of adapting the dictionary learning stage for inpainting, we modify the measurements to fit as an input for the dictionary update step. Having the initial image recovery, we replace each unknown pixel in the noisy image with a noisy pixel generated using the noise statistics and the given image recovery. This provides us with a noisy image for which we can apply any regular Poisson denoising algorithm.

We use this strategy within the dictionary learning process of SPDA. For the first dictionary update step we use the output of the sparse coding (Section 2.2). As each dictionary learning stage provides us with a new image estimate, we keep generating new approximations for the noisy image at each iteration.

3. EXPERIMENTS

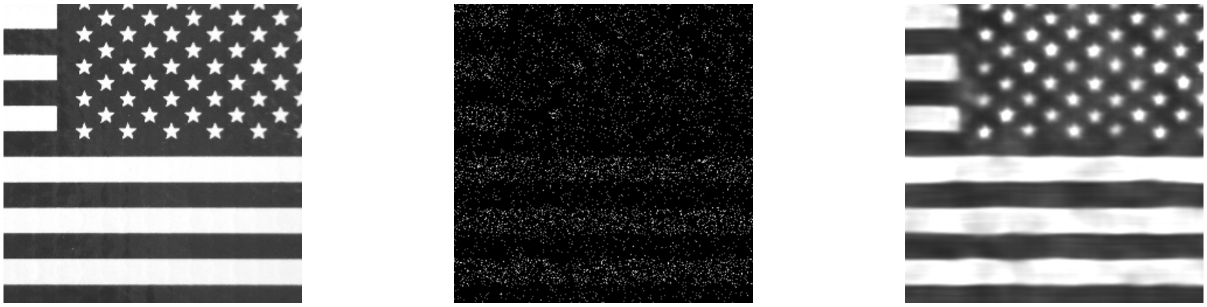
In this section we test the performance of the proposed inpainting scheme. It uses the same parameters as SPDA in [2]. We start by getting an initial recovery using the sparse coding method for inpainting. Then we use five dictionary learning steps, where in each step we replace the missing values in the measurements with their noisy estimated values generated using the most recent recovery result of the algorithm. We recluster all the patches based on the final recovered image and repeat the whole process again with the new groups.

We apply the new Poisson inpainting algorithm on the four test images in Fig. 2. We test four peak values $(0.5, 1, 2, 4)$ with 20%, 40% and 60% missing pixels. We select the locations of the missing pixels randomly. Figure 1 presents examples for patterns of missing pixels. We compare also to the case of 0% missing pixels (regular Poisson denoising).

Figure 3 presents the average PSNR achieved for recovering *flag, house, peppers* and *ridges* as a function of the percentage of missing entries. Four graphs are displayed, each for a different peak value. The visual recovery result can be seen in Figs. 4 and 5 for *flag* with peak 0.5 and *house* with peak 2. It can be seen that for every 20% of missing pixels we lose $1dB$ on average. However, the decrease in performance is not linear. For 60% missing pixels we lose $3dB$ on average while for 20% we lose much less than $1dB$.

4. CONCLUSION

This work proposes a novel Poisson inpainting scheme that relies on a recent state-of-the-art Poisson denoising method [2]. To the best of our knowledge this is the first work that treats the Poisson inpainting problem. In our experiments we have assumed that the locations of the missing pixels are drawn randomly. However, in real-world applications it is more likely that these locations will have a structured pattern. This is left to a future research. We believe that this paper opens a large room for research on Poisson inpainting.



(a) Original Image (b) Noisy Image (c) 0% Missing Recovery, PSNR =19.07

(d) 20% Missing Recovery, PSNR =18.59 (e) 40% Missing Recovery, PSNR =17.87 (f) 60% Missing Recovery, PSNR =15.47

Fig. 4. Poisson Inpainting of *flag* image with 0, 20, 40 and 60 percent missing pixels and peak = 0.5.

(a) Original Image (b) Noisy Image (c) 0% Missing Recovery, PSNR =24.87

(d) 20% Missing Recovery, PSNR =23.86 (e) 40% Missing Recovery, PSNR =22.83 (f) 60% Missing Recovery, PSNR =21.02

Fig. 5. Poisson Inpainting of *house* image with 0, 20, 40 and 60 percent missing pixels and peak = 2.

5. REFERENCES

[1] Joseph Salmon, Zachary Harmany, Charles-Alban Deledalle, and Rebecca Willett, "Poisson noise reduction with non-local PCA," *Journal of Mathematical Imaging and Vision*, pp. 1–16, 2013.

[2] Raja Giryes and Michael Elad, "Sparsity based poisson denoising with dictionary learning," *CoRR*, vol. abs/1309.4306, 2013.

[3] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B. Sibarita, and J. Salamero, "Patch-based nonlocal functional for denoising fluorescence microscopy image sequences," *IEEE Trans. on Med. Imag.*, vol. 29, no. 2, pp. 442–454, Feb. 2010.

[4] M. Makitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Trans. on Image Proces.*, vol. 20, no. 1, pp. 99–109, Jan. 2011.

[5] Bo Zhang, J.M. Fadili, and J.L. Starck, "Wavelets, ridgelets, and curvelets for poisson noise removal," *IEEE Trans. on Image Processing*, vol. 17, no. 7, pp. 1093–1108, July 2008.

[6] C.-A. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: Unsupervised non local means for poisson noise," in *IEEE International Conference on Image Processing (ICIP)*, Sept. 2010, pp. 801 –804.

[7] F.-X. Dupe and S. Anthoine, "A greedy approach to sparse Poisson denoising," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2013*, Sept 2013, pp. 1–6.

[8] Xiaoqun Zhang, Yujie Lu, and Tony Chan, "A novel sparsity reconstruction method from Poisson data for 3D bioluminescence tomography," *J. Sci. Comput.*, vol. 50, no. 3, pp. 519–535, Mar. 2012.

[9] M. A T Figueiredo and J.M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec 2010.

[10] E. Gil-Rodrigo, J. Portilla, D. Miraut, and R. Suarez-Mesa, "Efficient joint poisson-gauss restoration using multi-frame l2-relaxed-l0 analysis-based sparsity," in *IEEE International Conference on Image Processing (ICIP)*, Sept. 2011, pp. 1385–1388.

[11] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian, "Deblurring of Poissonian images using BM3D frames," *Proc. SPIE*, vol. 8138, pp. 813 812–813 812–7, 2011.

[12] F. J. Anscombe, "The transformation of Poisson, Binomial and negative-Binomial data," *Biometrika*, vol. 35, no. 3-4, pp. 246–254, 1948.

[13] M. Fisz, "The limiting distribution of a function of two independent random variables and its statistical application," *Colloquium Mathematicum*, vol. 3, pp. 138–146, 1955.

[14] K.. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *ICCV, 2009*, 2009, pp. 2272–2279.

[16] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2000, SIGGRAPH '00, pp. 417–424, ACM Press/Addison-Wesley Publishing Co.

[17] Antonio Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Proces.*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.

[18] Guoshen Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity," *IEEE Trans. on Image Processing*, vol. 21, no. 5, pp. 2481 –2499, may 2012.

[19] I. Ram, M. Elad, and I. Cohen, "Image processing using smooth ordering of its patches," *IEEE Trans. on Image Proces.*, vol. 22, no. 7, pp. 2764–2774, July 2013.

[20] L.N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 79–82, Jan. 2013.