A Simple Compressive Sampling Mode and the Recovery of Nature Images Based on Pixel Value Substitution

Wenping Shao, Lin Ni [1]

Abstract: Compressive Sampling (CS) is a new technique for information acquisition and processing. In this paper we propose a new algorithm based on Pixel Value Substitution (PVS) to reconstruct the nature images when the model of compressive sampling is very simple. By indirectly utilizing the fact that usually the gradient of a natural image is sparse, we divide the image into many small blocks, and for each block, we use only one typical value to represent all the pixels' values of that block. And thus we can construct a new matrix with full column rank from the Gaussian measurement matrix, and get a new system of equations that can be solved by the least square method which is also used in the greedy algorithms. And through analyzing we find out the statistical feature of the reconstructed signal, and the factors that influence the reconstruction quality, which tell us that, in order to get the most appropriate value for each pixel, the reconstruction needs to be repeated as is shown in the concrete steps of the algorithm, and also the block's size should be appropriate. At last, experimental results are given to demonstrate the viewpoints in this paper and they show that, in addition to improving the quality, PVS can also significantly reduce the time consumption.

Keywords: Compressive Sensing • Pixel Value Substitution • Gaussian Measurement Matrix • Natural Images.

[1] W. Shao, L. Ni (✉)
Department of Electronic Engineering and Information Science, University of Science and Technology of China,
Jinzhai Road, Hefei, Anhui Province, P.R.China,
e-mail: nilin@ustc.edu.cn

1 introduction

Traditional signal processing is based on the Nyquist sampling theorem, which means the signal can be accurately recovered when the sampling frequency is at least two times of the signal frequency. In recent years, a new signal processing technique has appeared, which is called Compressed Sensing (CS) [1, 2]. And by using the CS theory, signal sampling frequency can be greatly reduced. By using the fact that many natural signals are sparse in some transformed domains, the basic idea of CS is that we can transform the signal into two incoherent domains [3]. If the original signal is *x*, the transformation operator is *f*, and observation operator is *g*, then the transformed signals will be *f(x)* (which is sparse) and *g(x)*. CS theory tell us we can choose just a small part of the elements set of *g(x)* to recover *f(x)* and *x* accurately, and the choice is arbitrary [4], which means we can use any part of the set, premising that the number of elements is sufficient. There are many specific methods based on CS theory, and main types are greedy algorithms [5, 6, 7] and l_1 convex optimization [2].

Today, signal and image processing technology has been able to process an image in various ways, but in some occasions, when collecting images information or when observing images, the equipment can not match the needs of complex processing, and so that the amount of captured data should be small, and the sampling mode should be simple. This is closely related with the CS theory, and it's necessary to research how to recover an image when it was sampled in some very simple ways. Suppose that *x* is the original image signal, and we observe *x* directly by using a Gaussian measurement matrix *A*, and then we can get the observed signal:

$$y = Ax \ . \tag{1}$$

This is a very simple sampling mode, and this article propose an algorithm about how to recover *x* from *y* in this condition.

2 Pixel Value Substitution

In this section we propose the PVS algorithm, discuss the reconstruction quality, and show the concrete steps of the algorithm.

2.1 Principle

We can see from (1) that, it's an underdetermined system of equations, and if we know nothing about *x*, then we cannot get the solution of these equations. But fortunately, natural images do have some rules, and one rule is that the gradient of a natural image is sparse [8], which means most pixels of the gradient image have the values that are close to zero. And then we know that, usually, values of adjacent pixels are almost the same, and we can use just one value to represent them.

We know that, if we multiply a pixel value of x by a column of A, then we can get a part of y.so we can rewrite the observation equations on the following form:

$$y = Ax = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} I_{ij} = \sum x_{ij} I_{ij} . \tag{2}$$

Here, $A \in R^{M\times N}$, $y \in R^{M\times 1}$, and $m \times n = N$. And x_{ij} denote the pixel in the ith row and jth column of an $m \times n$ image x, and I_{ij} denote a column of the measurement matrix A, which correspond to x_{ij}.

Now, we divide the image x into T small blocks, and the rth block is denoted by D_r. Assume that D_r has k_r pixels, and then:

$$k_1 + k_2 + ... + k_T = N .$$

We know that，usually, all pixel values of D_r are almost the same ,thus:

$$y = \sum x_{ij} I_{ij} = \sum_{r=1}^{T} (\sum_{i,j\in D_r} x_{ij}^{(r)} I_{ij}^{(r)}) \tag{3}$$

$$\approx \sum_{r=1}^{T} [\alpha_r (\sum_{i,j\in D_r} I_{ij}^{(r)})] = \sum_{r=1}^{T} \alpha_r \Psi_r = \Psi\alpha \tag{4}$$

$$= \sum_{r=1}^{T} \sqrt{k_r}\,\alpha_r (\sum_{i,j\in D_r} \frac{I_{ij}^{(r)}}{\sqrt{k_r}}) = \sum_{r=1}^{T} \beta_r I_r = B\beta . \tag{5}$$

Here, $x_{ij}^{(r)}$ denote the pixel in the ith row and jth column of x, and it belongs to D_r; $B \in R^{M\times T}$, $\alpha, \beta \in R^{T\times 1}$; and α_r represent a typical value that can substitute for each $x_{ij}^{(r)}$,and obviously, α_r is close to them; and β is the normalization of α .

And (3), (4), and (5) tell us that we can reduce the number of unknown parameters by substituting one value for several values. According to the knowledge of random distribution, I_r is a random vector, and elements of I_r and I_{ij} have the same distributional parameters. So we know that B is a Gaussian matrix too. And according to the Marchenko-Pastur law [9], it's easy to know that:

$$T \le M \Rightarrow \text{Rank}(B) = T . \tag{6}$$

And (6) tell us if T is less than M, then B would be a full column rank matrix. Therefore, we can use the least-square method to solve (5), and can get the solutions:

$$\hat{\beta} = \arg\min_{\beta} \| y - B\beta \|_2 \ . \tag{7}$$

Then we can get $\hat{\alpha}$ and the reconstructed image signal $\hat{x}$:

$$\hat{x}_{ij}^{(r)} = \hat{\alpha}_r = \hat{\beta}_r \big/ \sqrt{k_r} \ . \tag{8}$$

2.2 Analysis of the Reconstruction Quality

Assume that we have got $\hat{x}$, and now we analyze the difference between $\hat{x}$ and *x*, which is also called residual Δx .We know that:

$$\|\Delta x\|_2^2 = < x - \hat{x}, x - \hat{x} > = \sum_{r=1}^{T} (\sum_{i,j \in D_r} (x_{ij}^{(r)} - \hat{\alpha}_r)^2)$$

$$\geq \sum_{r=1}^{T} \{ \frac{1}{k_r^2} [\sum_{i,j} (k_r x_{ij}^{(r)})^2 - (\sum_{i,j} x_{ij}^{(r)})^2] \} . \tag{9}$$

And equalities hold when and only when:

$$\hat{\alpha}_r = \frac{1}{k_r} \sum_{i,j \in D(r)} x_{ij}^{(r)}, \ r = 1,2,...,T \ . \tag{10}$$

In other words, when $\hat{\alpha}_r$ is the average value of D_r, $< \Delta x, \Delta x >$ has a minimum energy, and under the condition (10), we use $\hat{\beta}^*$, $\hat{\alpha}^*$, $\hat{x}^*$ to rewrite $\hat{\beta}$, $\hat{\alpha}$, $\hat{x}$.

On the other hand, if $\hat{x}$ is the original signal, then the observation signal will be $\hat{y} = A\hat{x}$, and the reconstructed signal will exactly be $\hat{x}$. But if the observation signal is *y*, then it will be almost impossible that the reconstructed signal is *x.* So we need to know the difference between y and $\hat{y}$. Let Δy denotes the difference, and we know that:

$$\Delta y = y - \hat{y} = \sum_{r=1}^{T} (\sum_{i,j \in D(r)} (x_{ij}^{(r)} - \hat{\alpha}_r) I_{ij}^{(r)}) = \sum_{r=1}^{T} (\sum_{i,j \in D(r)} \Delta x_{ij}^{(r)} I_{ij}^{(r)}) = \sum \Delta x_{ij} I_{ij} \ . \tag{11}$$

So Δy can be considered as a random vector, and according to knowledge of chi-square distribution, when *A* is a large dimensional matrix, there is an approximate proportional relationship between $< \Delta y, \Delta y >$ and $< \Delta x, \Delta x >$. So in the statistical sense, when $\hat{x} = \hat{x}^*$, $< \Delta y, \Delta y >$ reaches the minimum value, or the difference between y and $\hat{y}$ is the smallest. We know that if the difference of observation signals is smaller, then the difference of reconstructed signals will be smaller too. So $\hat{x}^*$ is the nearest signal to $\hat{x}$ which is recovered from *y*, or the reconstructed signal tends to be $\hat{x}^*$.

Let a denote the average value of a small block, b denote the value of a pixel in the center of this block, and c denote the value of a pixel on the edge of this block, and then usually the difference between a and b is smaller than the difference between a and c. So in the center of a block, $\hat{x}^*$ and x have a relatively high degree of similarity, while on the edge, have a relatively low degree of similarity. So in order to get the most appropriate value for each pixel, we can repeat the dividing and the reconstructing for several times, and each time use a different way to divide the image, as is shown in 2.4.

The conclusion that the reconstructed signal tends to be $\hat{x}^*$ is correct only in the statistical sense, which means we can find from the results of large number of experiments that, generally speaking, $\hat{x}^*$ is most close to x. But it is not true in a specific experiment, because Δy also leads to a reconstructed signal. We know that:

$$y = A\hat{x}^* + A(x - \hat{x}^*) = A\hat{x}^* + e\ . \tag{12}$$

So y can be divided into two parts: one part is $A\hat{x}^*$ and (7) tell us we can get the reconstructed signal $\hat{x}^*$ from this part; the other part is e, and also, we can get the reconstructed signal δ, which is something we don't expect to see because it can cause some uncertain interference. According to (7), we define:

$$\Delta\hat{\beta} = \arg\min_{\beta} \|e - B\beta\|_2\ . \tag{13}$$

And so we can know that the value of $<e,e>$ or $<\Delta\hat{\beta},\Delta\hat{\beta}>$ should be as small as possible. Divide e into two parts: $e^P = B\Delta\hat{\beta}$, and $e^V = e - B\Delta\hat{\beta}$, then, according to the related knowledge of matrix theory, we can know that $e^P \perp e^V$, and :

$$<e,e> = <e^P,e^P> + <e^V,e^V>\ .$$

And if we define:

$$\lambda = \frac{<e^P,e^P>}{<e^V,e^V>}, \tag{14}$$

then, λ should also be as small as possible. For the $M \times T$ matrix B, according to the knowledge of subspace projection, λ increases when M decreases and T maintains constant, or when M maintains constant and T increases.

Now we can get some factors that affect the quality of the reconstructed signal. The first one is the image's own features, such as the smoothness of the image, and they will affect e. The second one is the compression ratio M / N which is also called q. If other conditions remain unchanged, increasing q is equivalent to increasing M while letting T remain unchanged, and so it is equivalent to decreasing λ which can improve the quality. And q also affects the upper limit of T, because the derivation (6) should be tenable. The third factor then, is T. Decreasing T is equivalent to decreasing λ and this could help to improve the quality. However, a smaller T means a larger average size of the blocks, and usually $< e, e >$ will be lager too, and this will reduce the quality. So T should be an appropriate value. Of the three factors, the first two are usually unalterable, while the last one is what we can take advantage of.

2.3 Comparison with the Greedy Algorithms

Actually, this algorithm is similar to the greedy algorithms, such as OMP, ROMP, etc, because both of the two kinds use some particular column vectors called key column vectors to solve problem. We can see from (3)(5)(12) that:

$$y = A\hat{x}^* + A(x - \hat{x}^*) = B\hat{\beta}^* + A\Delta\hat{x}^* = [B, A]w = Cw \,. \tag{15}$$

And it's easy to know that usually:

$$w_i >> w_j \quad , i = 1,2,...,T; \;\; j = T+1, T+2,..., T+N \,. \tag{16}$$

So, the column vectors of B are more important than the column vectors of A, and they are the key column vectors. The difference, then, is that in the PVS algorithm we can construct all of these vectors directly and quickly, while in the greedy algorithms we obtain them by using iterative methods which are usually time-consuming jobs. And the experimental results will show that the PVS algorithm can save much time.

2.4 Concrete Steps

In order to get a better result, we need to repeat the dividing and reconstructing. Usually, we divide the image into many small blocks and each non-edge block is of the same size called the basic size. Assume that the size of the original image is $m \times n$ ($m \times n = N$), the basic size is $s \times t$, and the size of the block at the top-left end of the image is $u1 \times u2$, and meanwhile define:

$$v1 = 1 + \mathrm{mod}(m - u1 - 1, s) \,, v2 = 1 + \mathrm{mod}(n - u2 - 1, t) \,. \tag{17}$$

$$K = 2 + (m - u1 - v1)/s \,, L = 2 + (n - u2 - v2)/t \,. \tag{18}$$

Then, it's easy to know that the $m \times n$ image can be divided into $T = K \times L$ blocks, and the block at lower right end of the image has $v1 \times v2$ pixels. For the rth block D_r, we have know that each $x_{ij}^{(r)}$ has a different correlation degree with α_r, so we can construct a relative weight coefficient matrix ρ with the size $m \times n$, and $\rho_{ij}^{(r)}$ represents the relative weight coefficient between $x_{ij}^{(r)}$ and

α_r. And also we can construct a total weight coefficient matrix p to accumulate ρ, because the reconstruction will be repeat for $s \times t$ times, and each repetition needs a different ρ.

The following is the concrete steps:

```
Algorithm: Pixel Value Substitution (PVS)
Input: $y, A, m, n, s, t$.
Initialization: $u1 = s$, $u2 = t$, $p = 0$, $\hat{x} = 0$.
Steps: (i)  Divide the image according to (17)(18);
(ii)  Construct $B$ according to (3)(4)(5)(i); and set $\rho$ according to (i);
(iii)  Calculate $\hat{\beta}$ and $\hat{\alpha}$ according to (7)(8);
(iv)  $\hat{x}_{ij}^{(r)} = \hat{x}_{ij}^{(r)} + \rho_{ij}^{(r)} \alpha_r$;  $p = p + \rho$;
(v)  $u1 = u1 - 1$; if $u1 > 0$, go to (i); else $u1 = s$;
(vi)  $u2 = u2 - 1$; if $u2 > 0$, go to (i); else end.
Output: $\hat{x}_{ij} = \hat{x}_{ij} / p_{ij}$ and $\hat{x}$ is the output.
```

3 Experiments, Results and Discussion

In order to show the effect of the block's size on the quality of the reconstructed image, in the experiments we choose three different basic sizes including 5×5, 3×3, and 2×2 for the dividing of the image, and so we can get 3 specific methods of the PVS algorithm called 5×5 PVS, 3×3 PVS, and 2×2 PVS. Of course we should make sure that $T \le M$.

Now we discuss how to set ρ. For the 5×5 or 3×3 PVS, we know that usually, D_r has a center pixel, and if we use Ω to denote all the center pixels mentioned above, and all the edge pixels of the while image, then we could have:

$$\rho_{ij}^{(r)} = \begin{cases} 1, & x_{ij}^{(r)} \in \Omega \\ 0, & x_{ij}^{(r)} \notin \Omega \end{cases}.$$

And for the 2×2 PVS, then, always we have:

$$\rho = 1.$$

Take the 512×512 gray LENA test image as the original image, and in order to increase the speed, it is divided into 256 sub-images [10], and the size of each sub-image is 32×32. These sub-images are processed independently.

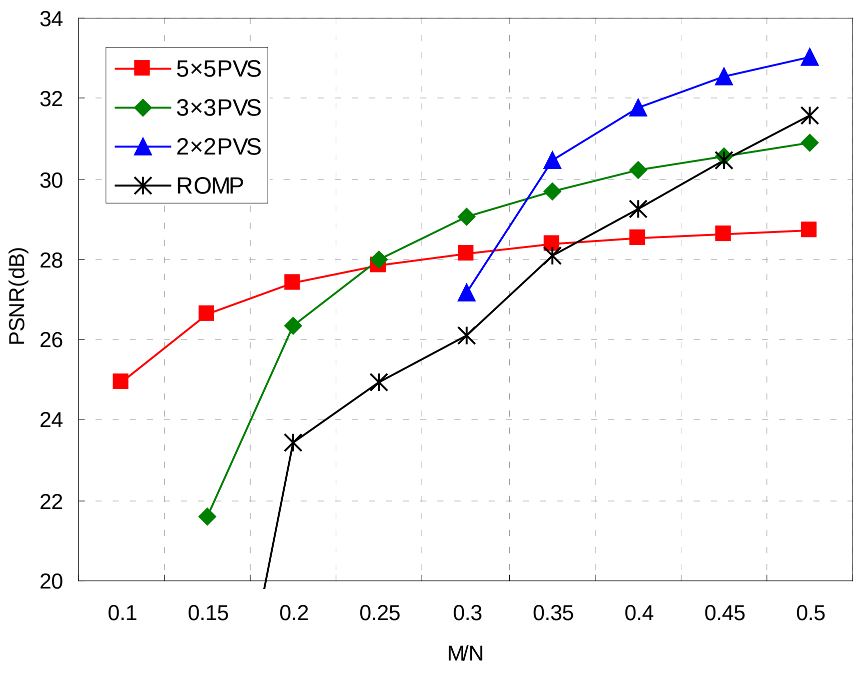

Fig. 1 The PSNR of each method. *Red*, *green*, *blue* and *black curves* denote respectively the PSNR of the 4 methods as are shown

As a contrast, the ROMP algorithm is used too, and the following is the specific process. First, 5 level wavelet transform is applied to the original image, and so we can get the sparse image. Then, the sparse image is divided into 256 sub-images, and the size of each sub-image is 512×2, which means each sub-image also contains 1024 pixels. And they are processed independently too.

Fig 1 shows the peak signal to noise ratio (PSNR) of each of the four methods when q varies from 0.1 to 0.5. We can see from Fig 1 that, for the PVS algorithm, a proper basic size or T can help to improve the PSNR and the quality of the reconstructed image, if the others conditions keep constant. The experimental results also show that a proper T should satisfy $T / M < 0.8$. Again, we can see from Fig 1 that, whatever the value of q is, the best method is not ROMP but one of the 3 PVS methods, which means that compared with ROMP, PVS can get a better result when the basic size is the most proper one. And one reason is that, what the ROMP algorithm reconstructs directly is not the original image itself but the sparse image transformed from it, and just one pixel of this sparse image may have a great influence on the whole original image.

Fig 2 shows the original image and the reconstructed images, and we can see from Fig 2(c) that subjective visual effect is good.

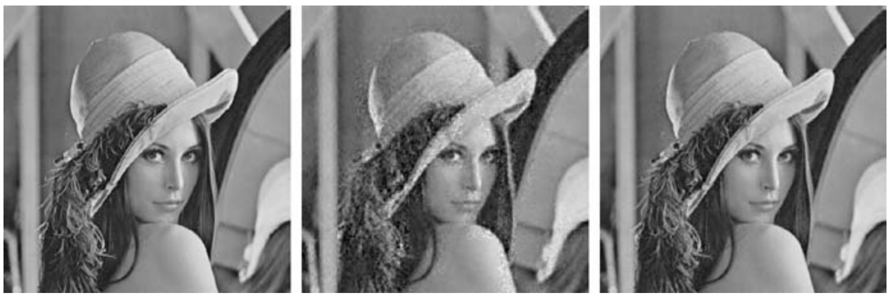

Fig. 2 The original image and the reconstructed images based on PVS. (a) denotes the 512×512 LENA test image. (b) denotes the reconstructed image when the basic size is 5×5 and M/N=0.1, and here PSNR=24.92. (c) denotes the reconstructed image when the basic size is 2×2 and M/N=0.5, and here PSNR=33.03

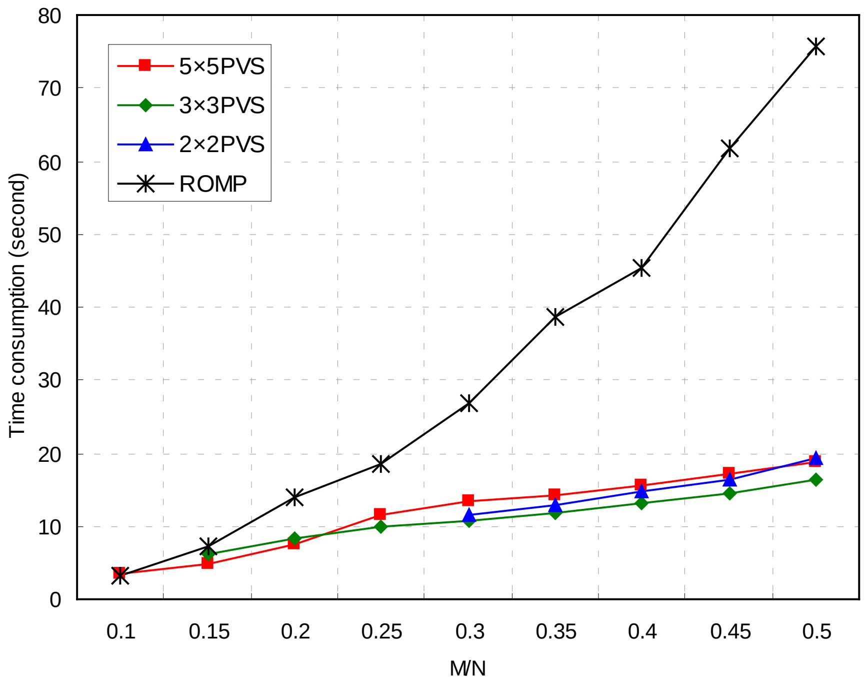

Fig. 3 The time consumption of each method. *Red*, *green*, *blue* and *black curves* denote respectively the time consumption of the 4 methods as are shown

Fig 3 shows the time consumption of the four methods. We can see that compared with ROMP, PVS can significantly reduce the time consumption. In fact, ROMP needs many times of iteration to search those key columns, and each iteration needs much more time than the previous one. But PVS is different, because each repetition needs the same time which is in inverse proportion to the number of the repetitions. So, the time consumption of PVS is usually much lower than that of ROMP, which means PVS can save much time.

3 Conclusion

In this paper, we have discussed the reconstruction of the nature images when the sampling model is very simple. We used a new way to solve the problem, and proposed the PVS algorithm including its concrete steps. We also analyzed the reconstruction quality and the similarities and differences of the two algorithms. The experimental results told us it’s a feasible method and in some degree is better than the greedy algorithms. However, we do have some future work to do. For example, we can use blocks of different sizes to divide the image, or we can propose an adaptive selection method so that the algorithm can automatically choose the most suitable basic size.

References

1. Candès, E. J. (2006). Compressive sampling. In Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures (pp. 1433-1452)
2. Donoho, D. L. (2006). Compressed sensing. Information Theory, IEEE Transactions on, 52(4), 1289-1306.
3. Candes, E., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. Inverse problems, 23(3), 969
4. Candès, E. J., & Wakin, M. B. (2008). An introduction to compressive sampling. Signal Processing Magazine, IEEE, 25(2), 21-30
5. Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. Signal Processing, IEEE Transactions on, 41(12), 3397-3415
6. Tropp, J. A., & Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. Information Theory, IEEE Transactions on, 53(12), 4655-4666
7. Needell, D., & Vershynin, R. (2009). Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. Foundations of computational mathematics, 9(3), 317-334
8. Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1), 259-268
9. Bai, Z. D. (1999). Methodologies in spectral analysis of large-dimensional random matrices, a review. Statist. Sinica, 9(3), 611-677
10. Gan, L. (2007, July). Block compressed sensing of natural images. In Digital Signal Processing, 2007 15th International Conference on (pp. 403-406). IEEE