A spectral algorithm for removing salt and pepper from images

Olivier Rioul
Dept. COM – URA CNRS 820
Ecole Nationale Supérieure des Télécommunications (ENST Paris)
46, rue Barrault, 75634 PARIS Cedex 13, France
Email: rioul@com.enst.fr

ABSTRACT

This paper presents an elegant solution to the impulse noise cancellation problem for images. The approach taken here is based on the theory of error-correcting (BCH) codes. Impulses are considered as "errors". The process is in two steps: First, find the error locations in the image; then, correct the error values.

A first pass is made on the lines of the image, a second pass is then performed on the columns. Next the same procedure is repeated. In most cases, impulse noise is completely removed after a few iterations are performed.

1. STATEMENT OF THE PROBLEM

This paper presents an elegant solution to the impulse noise cancellation problem for images. This problem can be formulated as follows.

Figure 1. Initial image: 128×128 portion of LENA 256×256.

We are given an original image such as the one of fig. 1. In our example, the gray level resolution is 8 bits, i.e., the pixel amplitude range is $[0 \cdots 255]$. Fig. 2 shows the image corrupted by 5% impulse noise – a pixel is damaged with probability = 0.05. The damaged pixel value was taken randomly in the range $[0 \cdots 255]$; since it sometimes looks bright in dark regions of the image, and sometimes dark in bright regions, the visual aspect of this type of noise is that of "salt and pepper". The problem addressed here is to remove salt and pepper, using the important side information that most of the spectral content of the initial image is confined into low frequency components.

Figure 2. Salt and pepper on LENA (5% impulse noise).

2. SOME CLASSICAL SOLUTIONS

Some of the usual techniques for combatting impulse noise are illustrated in figs. 3–5. Linear lowpass filtering (fig. 3) can remove "out-of-band" noise components, but "in-band" noise is still present, resulting in visible distortion. This suggests that the process of removing impulse noise should be non-linear.

A classical method, similar to clipping in the amplitude domain, consists of applying a (nonlinear) median filter to the lines (or the columns) of the image. In fig. 4, each output pixel is the median value of the 3 neighbor pixels in the cor-

0-7803-3629-1/96/$5.00 @ 1996 IEEE

responding corrupted line. [Here the image was processed line by line, in accordance to all algorithms described in this paper. In practice a 3×3 2-D median filter is used.] The process is fast, but many original pixel values are affected, and the resulting distortion is still important.

Figure 3. Reconstructed image: low-pass filtered.

Figure 4. Reconstructed image: median filtered (line by line, 3 taps).

Another interesting approach is to apply a modified version of block truncation coding (BTC) [1]. Each 4×4 block in the image is quantized into a two-level signal: the threshold is taken as the median value of the block, and the two quantized values are the upper and lower median of the corresponding input pixel values. Therefore, unless the block is highly corrupted, impulse noise is removed from it. Note that BTC also performs data compression—as it was first designed for; in fig. 5, blocking effects are visible.

Figure 5. Reconstructed image: BTC coded (4×4 blocks).

3. CHARACTERISTICS OF THE APPROACH TAKEN HERE

The approach taken in this paper is based on the theory of error-correcting codes. Impulses are considered as "errors". The process is in two steps: First, find the error locations in the image; then, correct the error values.

Our approach differs from the ones mentionned above in several respects:

- Because of the two-step (localisation + correction) procedure, only those pixels classified as damaged by the algorithm will be affected. Therefore, provided all errors can be suitably corrected, the reconstructed image will practically not be distorted at all.
- Out-of-band components of the impulse noise are used as a signature to locate impulses to be removed. Once impulses are located, both in-band and out-of-band components can be subtracted out.
- A procedure that applies on local regions is more likely to fail when a local region is highly corrupted (burst errors). Here the process is global: typically it can be applied to entire lines of the image. Failure results only when the total number of impulses in one line is too large. This, for suitable choice of parameters in the algorithm, is a fairly improbable event.

We note that the recent method proposed by Mitra and Yu [2] adopts the same "localization + correction" strategy. They use the output of a quadratic filter (or some other more sophisticated method) to detect impulses, and then estimate the true value of the corrupted pixel by means of a local mean operator. Even when each step is linear, their combination yields a highly nonlinear procedure, just like the method proposed here. Their method seems fast and satisfactory for the first two points listed above, but not for the third (failure in case of burst errors).

4. ORIGIN AND INITIAL DESCRIPTION

We shall describe our method in the framework of BCH codes, whose definition and properties can be investigated using the discrete Fourier transform (DFT). Blahut gives an excellent tutorial review of such "BCH techniques" in [5].

The original theory of BCH and related codes was developped in the sixties for channel coding problems and takes place in the algebraic framework of finite fields. Later in the early eighties, Wolf [3] and Marshall [4] mentionned that these techniques can be applied to impulse noise cancellation, when properly revisited in the field of real or complex numbers. Wolf gave an example of a single-impulse correction. More generally, the classical BCH decoding procedure, as reviewed in [5], can correct multiple errors in a 1D signal.

A short description of the BCH algorithm is as follows. The original signal $x_0, x_1, \cdots, x_{N-1}$ of length N has DFT $X_k = \sum_i x_i W^{-ik}$, $k = 0, \cdots, N-1$, where $W = \exp(2j\pi/N)$. We assume, for the moment, that at least $2t$ consecutive DFT components of x are equal to zero, e.g., the high-frequency components: $X_k = 0$ for $k \in A = \{N/2 - t, \cdots, N/2, \cdots, N/2 + t\}$ (when N is even). The corrupted signal is y. We define the error signal to be $e_i = y_i - x_i$. The number of errors ν is the number of indexes i such that $e_i \neq 0$. It is assumed that $\nu \leq t$: at most t errors occur, and the algorithm will be able to locate and correct them.

Rewrite equation $y = x + e$ in the frequency domain: $Y_k = X_k + E_k$. Since X_k is zero for $k \in A$, we compute "out-of-band" components of the impulse noise $Y_k = E_k$, $k \in A$ (usually referred to as the "syndrome"). These are used as a "signature" of the impulses to be removed.

First error locations are found by computing the *error-locator polynomial* $\Lambda(z) = \prod_{e_i \neq 0}(1 - W^i z^{-1})$ of degree $\nu \leq t$. It turns out [5] that the coefficients of $\Lambda(z)$ can be found by solving a Toeplitz system made out of the $Y_k = E_k$, $k \in A$. This system is of the same form as the Yule-Walker equations of order t, and can be solved using various well-known algorithms. The error locations are then found by searching those locations i for which $\Lambda(W^i)$ vanishes. In practice we test for $|\Lambda(W^i)| < \epsilon$ to take computational noise into account.

Finally error values $e_i \neq 0$ are determined such that E_k is equal to the high-frequency components Y_k computed for $k \in A$. This gives a system of $2t$ linear equations to be solved for the $\leq t$ values $e_i \neq 0$. Several algorithms are described in [5]. The reconstructed signal is then $\hat{x}_i = y_i - e_i$. If $\nu > t$, the algorithm fails, but failure can be detected (e.g. the errors values found are not real but complex).

5. IMPROVED ALGORITHM AND SIMULATIONS

The BCH algorithm strongly relies upon the assumption that at least $2t$ consecutive DFT components of the original signal (to be recovered) are equal to zero. In practice, these components will not be exactly equal to zero. Unfortunately, the BCH algorithm soon breaks down in this case and has to be modified.

Let us now see how the modified algorithm works when applied to the 'salt and pepper' problem. The input signal is a line (or a column) of the image, of length N. In the initial algorithm, a large value of t is chosen (about $N/4$) but we require correction of only ν errors where ν is much smaller than t. Only the best candidates i for which $\Lambda(W^i) \approx 0$ are retained as possible error locations. The error values are then determined such that $\sum_{k \in A} |E_k - Y_k|^2$ is minimized by a least square method applied to an over-determined system of equations.

In practice, the algorithm attributes a small value to e_i when i does not, in fact, correspond to an impulse. Hence such false alarms can be avoided. Sometimes a few impulses are not detected as such, but the other ones are generally corrected.

Failure results when too many errors are present or when the initial high-frequency spectral components are not negligible. Our modified algorithm often fails or does not correct all errors even when the percentage of impulse noise in the image is relatively small (about 3%).

However, a very simple loop procedure can be used to achieve our goal: A first pass is made on the lines of the image, a second pass is then performed on the columns. Next the same procedure is repeated. Even when only a few im-

Figure 6. Iterative denoising of LENA (Fig. 2) using the spectral algorithm: (a) after decoding lines, then (b) columns then (c) lines again.

pulses are suitably corrected in the beginning of the algorithm, such correction greatly reduces the task of the following step, which is performed in the other direction in the image. The next step will in turn be able to correct more errors, and so on. Fig. 6 shows simulation results for 5% impulse noise: three iterations (lines/columns/lines) were actually needed in this case.

6. CONCLUSION

In this paper, we have described a nonlinear "localization & correction" procedure for removing impulse noise from images, where lines and columns are processed sequentially. The theory is based on error-correcting BCH codes defined over the field of real numbers. It is interesting to realize how apparently independent areas (coding theory and image processing) can merge in this situation.

Classical methods for impulse noise cancellation rely on the assumption that pixels are highly correlated in local regions of the image. Our approach is original in that the assumption under which impulse noise can be removed is a "global" one (zeros in the spectrum of entire lines or columns). As a perspective, we may envision that this kind of global approach can be successfully applied for the recovery of entire "blocks" in an image, in which blocks are initially missing, e.g. due to transmission errors.

REFERENCES

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Comm.*, vol. COM-27, no. 9, Sept. 1979, pp. 1335–1342.

[2] S. K. Mitra and T.-H. Yu, "A new nonlinear algorithm for the removal of impulse noise from highly corrupted images," in *Proc. 1994 IEEE Int. Symp. Circuits and Systems*, London,vol. 3, pp. 17–20.

[3] J. K. Wolf, "Redundancy, the discrete Fourier transform, and impulse noise cancellation," *IEEE Trans. Comm.*, vol. COM-31, no. 3, March 1983, pp. 458–461.

[4] T. G. Marshall Jr., "Coding of real-number sequences for error correction: A digital signal processing problem," *IEEE J. Select. Areas Comm.*, vol 2, no. 2, Mar. 1984, pp. 381–392.

[5] R. E. Blahut, *Algebraic methods for signal processing and communications coding*, Signal Processing and Digital Filtering, C. S. Burrus ed., Springer Verlag: New York, 1992.