MML Based Noise Cleaning of Images

Torsten Seemann, Peter Tischer
Department of Computer Science, Monash University
Clayton, Victoria, Australia, 3168

Abstract

Noise cleaning is the process of removing unwanted noise from an image. The problem with existing techniques is that they do not preserve two-dimensional image structure, and/or they require parameters to be supplied by the user. In this paper we describe an algorithm called MNC which explicitly segments the local neighbourhood of the pixel to be cleaned, and filters using only those pixels in the same segment. Additionally, the Minimum Message Length principle is used to decide on what is the best segmentation. We show MNC to be a good performer, with little structure loss and no special parameters.

1 Introduction

Noise cleaning is the process of removing noise from a signal. Types of noise include additive noise, multiplicative noise, and impulse noise, with the additive variety being a good model for the noise found in most digital images. Noise cleaning is often considered a recovery process — an original image I has been corrupted by some independently distributed additive noise ξ, resulting in $I' = I + \xi$. The aim is to compute the best possible estimate $\hat{I}$ of I using only I'.

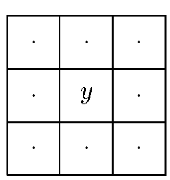

Figure 1: Local 3×3 neighbourhood of y

There exist many different noise cleaning techniques [1, 2]. Typically a small local set of pixels spatially enclosing the pixel y to be cleaned are used for filtering. A square $m \times m$ neighbourhood $\mathcal{M}$ (m odd) centered on y is a common choice, with $m \in \{3, 5, 7, 9\}$ being typical. Figure 1 gives an example for $m = 3$.

Simple linear averaging or smoothing masks such as the box operator (arithmetic mean of $\mathcal{M}$) tend to smooth edges and blur the image, resulting in a loss of image structure. Non-linear filters such as the median [3] and weighted-median [4] filters will preserve certain structural configurations, but fail on others. More complex approaches such as the sigma [5], and k-nearest neighbour [6] filters attempt to preserve some structure, but require parameters to be supplied by the user.

In this paper we describe a noise-cleaning technique (called MNC – "MML Noise Cleaning"). MNC endeavours to preserve image structure by explicitly segmenting $\mathcal{M}$ and filtering only using those pixels in the same segment as y. Additionally, the Minimum Message Length (MML) [7, 8, 9, 10] information theoretic measure is used to decide on the best segmentation to use for each pixel, thus removing the need for arbitrary parameters.

2 Local Segmentation

MNC considers noise cleaning to be a two part process. First, y and its local neighbourhood are *segmented* into one or more regions. This leaves us with two types of pixels — those which are in the same segment as y, and those which are not. We propose that only those in the same segment be used in the filtering process. Techniques such as mean and median filtering can now be used, as the operations will no longer overlap any segment boundaries, thus preserving the two-dimensional image structure.

The problem of which segmentation to use can be broken into two steps — producing good candidate segmentations, and deciding which of the candidates is the best.

2.1 Neighbourhood Size

For the rest of this paper, we will assume that only a local 3×3 neighbourhood $\mathcal{M}$ will be used to compute $\hat{I}$. However, MNC can be extended to any $m > 3$.

A reasonable assumption would be that most 3×3 blocks can be modelled by either one or two segments, representing flat and edge regions. Additionally, within a segment, we assume that a single representative value (RV) like the mean or median is sufficient to describe the grey-level properties of the segment. These assumptions are utilised to good effect by the many varieties of the lossy image compression technique Block Truncation Coding (BTC) [11]. BTC uses only local pixel information and is well known for its edge-preserving properties.

2.2 One Segment Models

Clearly, a one segment model (1SM) is fully defined by its RV, as the 9 pixels in $\mathcal{M}$ take the value of the RV. If we are dealing with pixels having z bits per pixel, there are 2^z possible 1SMs. However, if a given block $\mathcal{M}$ is well described by a 1SM, we expect the RV to be around the mean, median, or midpoint (average of the highest and lowest pixels in the block). In MNC, we consider only these three.

2.3 Two Segment Models

A two segment model (2SM) is described by a 3×3 bitmap and 2 RVs. There are $2^{m^2} - 2$ possible bitmaps, with the first and second segments having 2^z and $2^z - 1$ possible RVs respectively. This results in approximately $2^{m^2 z^2}$ possible 2SMs.

For MNC, we can reduce this search space significantly. We expect a large proportion of true two segment regions to have histograms which are bimodal in nature. This suggests a simple thresholding technique like that used in BTC would be sufficient to segment a small region. In fact, as there are only m^2 pixels in $\mathcal{M}$, only $m^2 - 1$ thresholds need be considered, and even fewer if $\mathcal{M}$ contains duplicate pixels. Additionally, the optimal RVs for the two segments are also expected to be close to the mean. In MNC, only the means of the two segments are considered as potential RVs.

3 An MML Approach

Given a set of candidate segmentations, the problem remains as to which of them is the *best* segmentation. A well-grounded approach which takes the *complexity* of the model into consideration, is the MML approach.

MML [7, 8, 9, 10] is is an information theoretic approach to inductive inference, similar to MDL [10] and Bayesianism. Let us assume we have some measurements $X = \{x_1, x_2, \ldots\}$ from the real world, and a set of models $M = \{m_1, m_2, \ldots\}$ with which we attempt to explain X.

MML can be used to assess the quality of each $m_i \in M$. This is done by constructing a description of (a) the model m_i, and (b) the data X in terms of the model m_i. The description, or message, takes the form of an efficient losslessly encoded binary string S. The length of S is measured in bits, and is called the code length. *The MML principle states the the best model for the data is the one which has the shortest overall code length.* We do not actually need to construct each S explicitly, only measure how long it would be.

For our application, the models M consist of segment maps and RVs for the segments. The data X are the residuals resulting from the difference between the given model m_i and the actual pixels.

3.1 Probability Estimation

The message takes the form of a series of events E. Each event E consists of n possible outcomes e_1 to e_n. In MNC, sets of frequency counters $f(e_i)$ are kept for each event, one counter per outcome. These are used to determine a probability $Pr(e_i)$ for each outcome using the formula

$$Pr(e_i) = \frac{1 + f(e_i)}{n + \sum_1^n f(e_i)} \tag{1}$$

The code length $CL(e_i)$ for a given outcome e_i is then

$$CL(e_i) = -\log_2 Pr(e_i) \;\; \text{bits} \tag{2}$$

If an event E consists of k discrete possibilities assumed equally likely, then it clearly follows that

$$CL(E) = -\log_2 k \;\; \text{bits} \tag{3}$$

The overall code length is simply the sum of all the individual code lengths for each component event in the message.

4 Message Format

All the messages have the following overall structure:

I. A binary event stating whether this is a 1SM or a 2SM.

II. A series of events describing the segmentation of $\mathcal{M}$. These will differ depending on the result of Part I.

III. The encoding of the residuals – the differences between the actual pixel values (from I') and those represented in the model from Part II.

We will now describe the details for Parts II,III for the 1SM and 2SM cases separately.

4.1 One Segment Events

Part II:

1. The RV value for the whole segment. Assumed uniformly distributed on $[0, 2^z - 1]$.

Part III:

2. The residuals for the segment. The frequency counts used are separate from those used in the 2SMs.

4.2 Two Segment Events

Part II:

1. The number of pixels k $(1 \leq k \leq 8)$ in the `0` segment (the segment with the lower RV).
2. A binary bitmap consisting of k `0`s and $(9-k)$ `1`s. As we have already transmitted the value of k, there are only $\binom{9}{k}$ possible bitmaps.
3. The RV for the `0` segment, assumed uniformly distributed on $[0, 2^z - 1]$.
4. The RV for the `1` segment. As the previous RV (say x) is known to be the lower of the two, this RV is assumed uniformly distributed on $[x+1, 2^z - 1]$.

Part III:

5. The residuals for the `0` segment.
6. The residuals for the `1` segment. Separate frequency counts are used for these and the previous residuals.

5 Prior Distributions

Each event of the segment descriptions in Section 4 consists of some set of possibilities (eg. Step 1 for the 2SM case has 8 possible outcomes). MML requires that each component have some prior probability distribution.

We expected the distribution of the residuals to be highly non-uniform, so the priors for them were initially estimated from the images by using the relative frequencies of the residuals from a 4-neighbour averaging filter. For all other events, a uniform prior distribution was used.

We then took an iterative approach. The first iteration used the priors just described. The posterior (actual observed) probabilities of all the events after the first iteration were then used as the priors for the second (and final) iteration.

6 MNC Algorithm

For each pixel y in the image to be cleaned:

1. Gather the local neighbourhood $\mathcal{M}$ of y.
2. Construct messages for one segment models of $\mathcal{M}$. We used used the mean, median, and midpoint of $\mathcal{M}$.
3. Construct messages for two segment models of $\mathcal{M}$. We used 8 binary thresholds.
4. Find the message S with the shortest message length.
5. Replace y the current pixel with the RV assigned to it by S.

Pixels on the edge of the image which do not have a full 3×3 neighbourhood are simply replaced by the average of as many pixels from its 8-neighbourhood as possible.

7 Results

For our experiments, we used the $512 \times 512 \times 8$ bit synthetic image `shapes` of Figure 2. It was chosen because it contains a variety of smooth and edge regions, including sections which are not modelled particularly well by only one or two segments. We then added Gaussian noise[1] with $\mu = 0$ and $\sigma^2 = 256$, resulting in Figure 3.

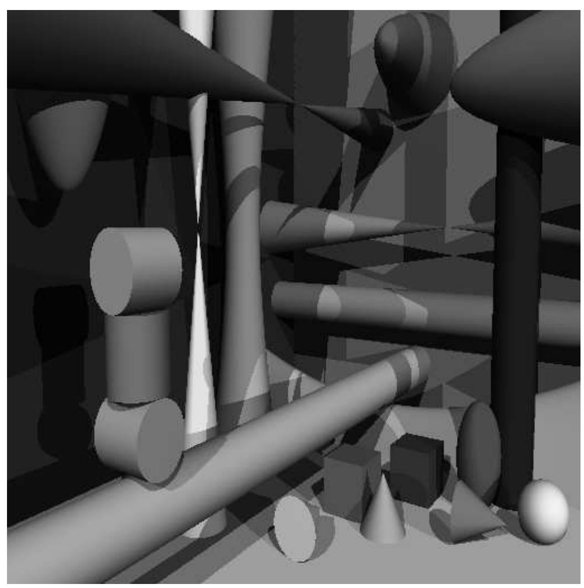

Figure 2: Original `shapes` test image

[1] If the addition of the noise resulted in a pixel taking on an illegal value, it was clipped to fall in the range $[0, 2^z - 1]$.

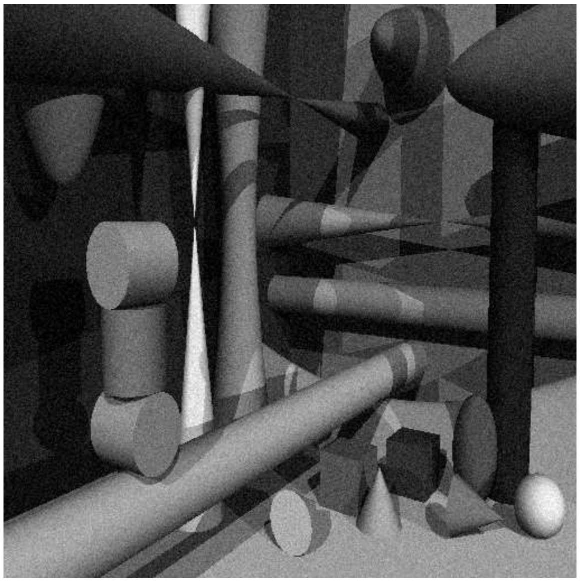

Figure 3: Image `shapes` with noise $\sim N(0, 256)$

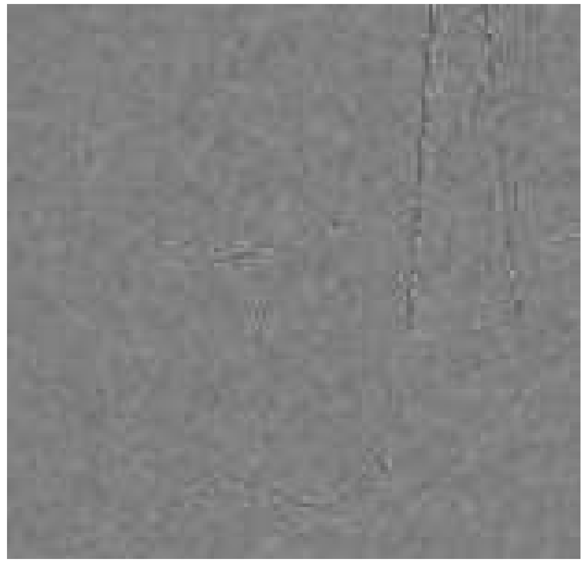

Figure 4: Close-up of MNC residuals

The PSNR was taken as $-\log_{10}(\text{MSE}/x_{max}^2)$ where x_{max} is the maximum pixel value 255. MNC is the technique described in this paper. The Box and Gauss filters are the smoothing masks $\frac{1}{9}\{1,1,1,\ \ 1,1,1,\ \ 1,1,1\}$ and $\frac{1}{16}\{1,2,1,\ \ 2,4,2,\ \ 1,2,1\}$ respectively. The w-median entry is a weighted median filter, similar to the plain median filter except that the middle pixel y is included 3 times. For kNN (k-Nearest Neighbour) we used $k = 6$ as it gave the best results, and Lee's sigma filter was over a 3×3 neighbourhood.

Algo-rithm	PSNR (dB)	RMSE	Filt. Image?	Resid. Struct?
MNC	33.43	5.43	sharp	(little)
Box	31.46	6.82	blurred	much
Gauss	32.14	6.31	blurred	much
median	32.57	6.00	sharp	some
w-median	31.80	6.55	sharp	little
kNN	32.19	6.27	sharp	some
sigma	30.87	7.30	ok	some

Table 1: Noise cleaning performance

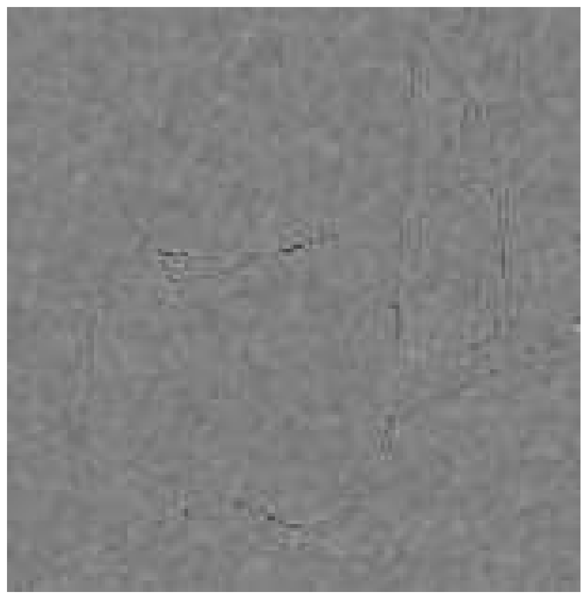

Figure 5: Close-up of Median residuals

8 Discussion

The quantitative results in Table 1 show MNC to be a good performer, beating the weighted median and k-Nearest Neighbour algorithms by about 0.8 dB. However, a single scalar such as PSNR does not fully capture noise cleaning performance. Table 1 also gives our qualitative examination of the filtered images and the amount of structure in the residual image (difference between the noise-free original and the filtered image).

As expected the simple smoothing masks resulted in blurred edges, leaving much structure in the residuals. The sigma filter did not perform as well as expected; it would probably do better with a larger neighbourhood size. The weighted median image had better structure preserving properties than the plain median, but performed slightly worse in terms of PSNR. This highlights the problem of choosing the weights for an arbitrary image.

The median and k-Nearest Neighbour filters came closest to MNC's performance. Figures 4 and 5 show a close-up of the residuals for the MNC and median filters near the peak of the tall upright white cone on the left of the `shapes` image. Mid-grey represents zero error. We can see that MNC had difficulty with the cone's left edge. This is because it was really a 3 segment region, which MNC tries to approximate with 2 segments.

Overall, MNC was the most successful algorithm. It produced a sharp output image with little structure loss, had a high PSNR, and did not require any special parameters to be supplied by the user.

It is possible to extend MNC to larger window sizes, however, the 1-or-2 segment assumption breaks down at larger scales. An efficient search mechanism for good candidate 3 and 4 segment regions is required. If one produces a bitmap with `0`s for when MNC chose a 1SMs and `1`s for 2SMs, the resulting image is an edge map. This suggests that MNC may be used for global image segmentation by combining the results of many noise-insensitive local segmentation operations.

References

[1] R. M. Haralick, L. G. Shapiro, "Computer and Robot Vision", Vol. 1, Addison-Wesley, 1992.

[2] G. A. Mastin, "Adaptive Filters for Digital Image Noise Smoothing: An Evaluation", *Computer Vision, Graphics, and Image Processing*, Vol. 31, 1985, pp. 103–121.

[3] J. W. Tukey, "Exploratory Data Analysis", Addison-Wesley, 1971.

[4] D. R. K. Bownrigg, "The Weighted Median Filter", *Communications of the ACM*, Vol. 27, 1984, pp. 807–818.

[5] J-S. Lee, "Digital Image Smoothing and the Sigma Filter", *Computer Vision, Graphics, and Image Processing*, Vol. 24, 1983, pp. 255–269.

[6] L. Davis, A. Rosenfeld, "Noise Cleaning by Iterated Local Averaging", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, 1978, pp. 1006-1010.

[7] C. S. Wallace, D. M. Boulton, "An Information Measure for Classification", *Computer Journal*, No. 11, 1968, pp. 185–194.

[8] C. S. Wallace, P. R. Freeman, "Estimation and Inference by Compact Coding", *J. R. Statist. Soc. B*, 49(3), 1987, pp. 240–265.

[9] J. J. Oliver, D. J. Hand, "Introduction to Minimum Coding Inference", Technical Report 94/205, Monash University, 1994.

[10] J. J. Oliver, R. A. Baxter, "MDL and MML : Similarities and Differences", Technical Report 94/207 (Amended January 1995), Monash University, 1995.

[11] P. Fränti, O. Nevalainen, T. Kaukoranta, "Compression of Digital Images by Block Truncation Coding: A Survey", *Computer Journal*, Vol. 37, No. 4, 1994, pp. 308–332.