Structure Preserving Noise Filtering of Images using Explicit Local Segmentation

Torsten Seemann and Peter Tischer
School of Computer Science and Software Engineering, Monash University
Clayton, Melbourne, Victoria, Australia, 3168
torsten@csse.monash.edu.au, pet@csse.monash.edu.au

Abstract

The trend in modern image noise filtering algorithms has been toward structure preservation by using only those neighbouring pixels which are similar to the current pixel in some way. In this paper we introduce a technique call FUELS (Filtering Using Explicit Local Segmentation) which explicitly segments the $m \times m$ *region encompassing the current pixel and filters using only those pixels from the same segment. By exploiting mask overlap an effective mask size of* $(2m-1) \times (2m-1)$ *is obtained, as well as robustness in regions which do not fit the image model. The algorithm can be iterated, and our results show FUELS to outperform existing algorithms both quantitatively and qualitatively.*

1. Introduction

Noise filtering [1, 2] is the process of removing unwanted noise from an image and is often a preprocessing stage in image analysis. Thus we desire a filter to preserve any image structure such as edges and texture.

Traditional linear filters such as the box and Gaussian [1] filters remove additive noise but indiscriminately blur edges. Order statistics approaches such as the median [3] and weighted median [4] filters can preserve some structures but fail on others. The trend in modern non-linear filters has been towards "adaptive local neighbourhood selection" [5]. That is, only neighbouring pixels similar to the current pixel (using some measure) are used for filtering. This philosophy germinated from techniques such as the Sigma [6], k-Nearest Neighbour [7], and Selected Neighbourhood Averaging [8] filters.

In Section 2 we describe the operation of our FUELS algorithm and Section 3 will give a small numerical example. Results are presented in Section 4 and conclusions and future work are discussed in Section 5.

2. The FUELS Algorithm

Let us assume we are using an $m \times m$ (m odd) mask containing $M = m^2$ pixels centered on the pixel y to be filtered and that each pixel in the image has been corrupted by i.i.d. additive noise. The FUELS algorithm consists of four main steps.

2.1. Estimation of Noise Variance

This step is done once at the start of the process. Let the "local variance" be the sample variance of a small $m \times m$ region. We can form an estimate $\hat{\sigma}_\eta^2$ of the "true" global noise variance σ_η^2 as the average of as many local variances as we can obtain from the image. The regions we use to compute the local variances should not contain any edges, as these will overestimate σ_η^2.

To ensure this, we first *roughly* classify each pixel as either being in a flat or edge region by using horizontal and vertical 3×3 Sobel edge detector masks [1]. Only regions with a centre pixel having a Sobel gradient magnitude less than 16 were used in the variance estimation.

2.2. Local Segmentation

In typical images we have found that most small blocks of pixels are simple in structure. The majority are flat or planar, suggesting their M pixels probably belong to the same segment. The rest are edge or texture regions having a simple bimodal distribution of pixels (not necessarily symmetric) which we can consider a two segment region.

To prevent blurring of edges we would like to average only those pixels which are in the *same segment* as y. Let $x_1 \cdots x_M$ be the pixels in the local neighbourhood of y, and $x_{(1)} \cdots x_{(M)}$ be the same but in ascending order. We must first decide whether to model this region as one or two segments. FUELS uses a confidence interval approach such that if the dynamic range $\Delta x = |x_{(M)} - x_{(1)}|$ of the block is too large we treat it as two segments:

$$\text{No. segments} = \begin{cases} 1 & \text{if} \quad \Delta x \leq 6\hat{\sigma}_\eta \\ 2 & \text{otherwise} \end{cases} \tag{1}$$

If we conclude that the region consists of a single segment we can use *all* M pixels to compute a robust estimate μ of y. FUELS uses the sample mean which is a good estimate in the presence of additive noise:

$$\mu = \bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{2}$$

Segmenting a small $m \times m$ region of pixels is reasonably simple. We expect a bimodal pixel histogram so a thresholding approach is appropriate. The lossy image compression technique Block Truncation Coding [9] has a long research history and is concerned with efficiently choosing a good binarizing threshold t.

We use the DRT algorithm [10] which divides the pixels into two groups by setting $t = (2\bar{x} + 3x_{(1)} + 3x_{(M)})/8$. Chan shows that this t is almost always equivalent to using the minimum mean squared error threshold.

If q is the number of x_i's with a value less than t, then the smoothed pixels estimates for the the "Low" and "High" segments are:

$$\mu_L = \frac{1}{q}\sum_{x_i < t} x_i \qquad\qquad \mu_H = \frac{1}{M-q}\sum_{x_i \geq t} x_i \tag{3}$$

2.3. Combining Pixel Estimates

At each filtering step we are computing pixel estimates (either μ, or one of μ_L/μ_H) for *every* pixel within the $m \times m$ mask surrounding y, not just y itself. The fact that the masks overlap means that each pixel is actually included in M separate masks. We can improve our estimate of y by not just using the estimate obtained when y was at the centre of a mask, but by taking an *average* of all M estimates.

This has two main effects. Firstly, the mask is effectively of size $(2m-1) \times (2m-1)$ but without the extra computation and modelling otherwise required. Secondly, the filter is more robust to erroneous estimates caused by outlier pixels or regions which were not well modelled by just one or two segments.

2.4. Iteration

It is possible to iterate the FUELS algorithm by letting the filtered output image become the input to the next iteration. We have found that up to 2 or 3 iterations can improve results for very noisy data ($\sigma \geq 15$). This has the effect of increasing the effective size of the mask, which is beneficial in large smooth areas. However the improvement in RMSE is usually accompanied by some loss of structure and a larger worst case error.

3. An Example

We give a numerical example of the FUELS algorithm with $m = 3$ on the 25 pixel region in Figure 1. We wish to filter the centre pixel currently having value 13. We assume $\hat{\sigma}_\eta = 2$. There are nine overlapping 3×3 blocks A–I in this region also designated in Figure 1.

13	8	11	9	38
12	11	10	10	37
10	9	**13**	14	35
9	11	12	32	34
11	10	31	37	33

A	B	C
D	E	F
G	H	I

Figure 1. Example overlapping 3×3 blocks

Region	Δx	Segs	$\bar{x}$	t	μ_L	μ_H	$\hat{y}$
A	5	1	10	-	10	-	10
B	6	1	10	-	10	-	10
C	29	2	19	22	11	36	11
D	4	1	10	-	10	-	10
E	23	2	13	18	11	32	11
F	27	2	21	22	11	34	11
G	22	2	12	18	10	31	10
H	28	2	18	21	11	33	11
I	25	2	26	24	13	33	13
						Average estimate	11

Table 1. Example block calculations

Table 1 lists the calculations for the nine 3×3 blocks. For this example all values have been rounded to integers, but normally full precision is kept. We can see that the algorithm has successfully segmented the pixels into the two expected regions for each 3×3 block. The centre pixel $y = 13$ was covered by 9 blocks each giving an estimate $\hat{y}$ in the last column of Table 1. The final filtered pixel value is the average of that column giving 11. This average was not contaminated by the nearby pixels having values ≥ 31. In the next Section we apply this algorithm to a full image.

4. Results

For our experiments we used a $512 \times 512 \times 8$ bit synthetic image[1] which consists of smooth planar regions and edges of varying contrast, some of which are not modelled particularly well by only one or two segments. The image

[1] ftp://ftp.cs.monash.edu.au/users/torsten/icpr98/shapes.pgm

was then corrupted with additive white Gaussian noise having $\mu = 0$ and $\sigma = 16$. In Table 2 we describe the terms that are used to describe the filtering results in Table 3.

Term	*Description*
m	The side length of the square mask used
N	Number of times the algorithm was iterated
RMSE	Root mean squared error (L_2 norm)
PSNR	Peak signal to noise ratio (dB) $-\log_{10}(\text{MSE}/255^2)$
MAE	Mean absolute error (L_1 norm)
WCAE	Worst case absolute error (L_∞ norm)
FI	Qualitative examination of the filtered image
RS	Subjective exam. of residual structure: 1=good,5=bad
FUELS	The algorithm described in this paper
Box	Smoothing mask $\frac{1}{9}\{1,1,1\ \ 1,1,1\ \ 1,1,1\}$
Gauss	Smoothing mask $\frac{1}{16}\{1,2,1\ \ 2,4,2\ \ 1,2,1\}$
Med	Standard median filter [3]
WMed	Weighted median filter [4] (y occurs 3 times)
kNN	k-Nearest Neighbour [7] ($k = 6$ was best)
Sigma	Lee's sigma filter [6]

Table 2. Description of terms used in Table 3

Algo.	m	N	RMSE	PSNR	MAE	WCAE	FI	RS
FUELS	3	1	5.3	33.6	3.8	77	sharp	1
	3	2	4.9	34.4	3.3	83	sharp	2
	3	3	4.8	34.4	3.2	86	sharp	2
Box	3	1	7.6	30.5	5.2	126	blurry	3
	3	2	7.2	31.0	4.5	124	blurry	4
Gauss	3	1	7.3	30.8	5.4	95	ok	3
	3	2	7.1	31.2	4.7	117	blurry	4
Med	3	1	7.5	30.7	5.7	175	sharp	1
	3	2	6.4	32.0	4.6	175	sharp	2
	3	3	6.1	32.4	4.2	178	sharp	3
	5	1	6.8	31.5	4.3	198	blurry	4
WMed	3	1	8.3	29.8	6.5	166	sharp	1
	3	2	7.2	30.9	5.5	174	sharp	1
kNN	3	1	7.6	30.5	5.9	123	good	1
	3	2	6.5	31.9	4.8	140	good	2
	3	3	6.4	32.0	4.5	149	ok	2
Sigma	3	1	9.2	28.9	6.6	97	sharp	2
	3	2	8.4	29.7	5.4	118	ok	3
	5	1	8.0	30.1	5.6	111	ok	4
	7	1	8.3	29.8	5.6	105	blurry	5

Table 3. Filtering results for `shapes`

5. Conclusions

The FUELS algorithm is the best performer on the `shapes` image in terms of PSNR, being 2.0dB ahead of its nearest rival, the median filter. It produced sharp output images with little structure present in the residual image, and had the lowest WCAE of all the filters. The median filter produced sharp outputs because it tends to alter rather than blur image structure. The weighted median filter has better structure preserving properties, but at the expense of not being able to clean smoother areas as effectively. As expected the simple Box and Gauss filters blurred edges resulting in large structure losses.

The algorithms closer to FUELS in methodology, such as kNN and Sigma did reasonably well in terms of preserving structure, but still blurred many edges. The reason for this is that their edge-sensor component is fixed in some way over the whole image, whereas FUELS adaptively segments each local image block.

The performance of FUELS is somewhat dependent on the estimate of the noise variance, especially on extremely edgy images. The Sobel threshold of 16 could be lowered to make the variance estimate more conservative. Currently, FUELS *averages* the estimates $\hat{y}_1 \cdots \hat{y}_M$ for the current pixel. One could use a median operator instead to improve robustness, especially in the presence of impulse noise. FUELS can also be extended to larger mask sizes. We found $m = 5$ to work reasonably well, but the assumption that a 25 pixel block can be modelled using just 1 or 2 segments begins to become less valid.

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] M. W. Hansen, W. E. Higgins, "Image Enhancement using Watershed-Based Maximum Homogeneity Filtering", *IEEE 2nd International Conference on Image Processing*, Vol. II, pp482–485, 1995.

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

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

[8] R. M. Haralick, L. Watson, "A Facet Model for Image Data", *Computer Graphics and Image Processing*, Vol. 15, pp113–129, 1981.

[9] P. Fränti, O. Nevalainen, T. Kaukoranta, "Compression of Digital Images by Block Truncation Coding: A Survey", *Computer Journal*, Vol. 37, No. 4, pp308–332, 1994.

[10] K. W. Chan, K. L. Chan, "Optimized Adaptive AMBTC", *IEEE 1st International Conference on Image Processing*, Vol. II, pp371–374, 1994.