NONLINEAR SMOOTHING FILTER USING ADAPTIVE RADIAL CLUSTERING

Irene Yu-Hua Gu
Dept. of Signals and Systems,
Chalmers University of Technology, Sweden
irenegu@s2.chalmers.se

Vasile Gui
Dept. of Electronics and Communications,
Polytechnic University of Timisoara, Romania
gui@hermes.ee.utt.ro

ABSTRACT

A novel adaptive nonlinear filter is proposed aimed at smoothing homogenous regions while maintaining image structures. The filter can be utilized as a pre-processing tool in image segmentation and edge estimation for improving the results. Several special features are introduced to the filter, including using local adaptive radial clustering and pixel filtering to exclude the influence of outliers and to maintain image structures; using steepest-ascent method to iteratively update pixels to the nearest clusters obtained by mean-shift; and introducing highly parallel processing by using random seed samples and their associated data blocks which enables fast processing and the global optimum solution of the nonlinear filter. Experiments were done on images of various complexities, and good results were obtained. Evaluations of the filter were also done in terms of edge preserving and image segmentation.

1. INTRODUCTION

The importance of edges in visual perception and computer vision is widely recognized. Edge enhancement is also useful in pre-processing of image segmentation where edge information can be used to obtain partitions corresponding to real objects, or meaningful parts of the objects in the image. Edge enhancement and image smoothing are often conflicting demands that cannot be well addressed in the framework of linear filtering. Since linear smoothing filters may result in edge blurring, it is often desirable to use nonlinear filters. Many nonlinear smoothing techniques have been investigated [1-5]. A commonly exploited idea is to weight pixels according to their confidence of being representative for the estimate. The difference often arises from the way the confidence is obtained. In order statistic L-filter, weighted ordered intensities in the neighbourhood region are used to modify the pixel value. In edge-preserving smoothing filter [10], a set of neighbourhood regions of the same central pixel is defined and the region having the smallest intensity variance is used to modify the centre pixel value. In nonlinear diffusion [4,9], weights are inversely proportional to the magnitude of the intensity gradient at the currently processed location. Sharp boundaries separating homogeneous regions are produced, however, the computational cost is high. Offset filtering [3] offers a fast alternative where the centre of filter kernel is placed away from the assumed edges with an offset depending on local image geometry. Another alternative is to use adaptive non-local filtering [6], where estimation of offset vector field is separated from the actual image filtering by means of pixel permutation. Good results were reported however shaded areas were often split into several flat regions separated by artificial boundaries. It is desirable that shaded regions be treated differently from blurred edges. Another problem in many clustering-based segmentation approaches is that pixels are first mapped to a feature space, followed by clustering in the feature space globally [3,7]. This may cause deviations of clusters due to grouping pixels from different neighbourhood regions. To alleviate this problem, it is desirable that clustering be performed on a local basis.

Motivated by the above, we propose a novel edge-preserving nonlinear smoothing filter based on adaptive local clustering. The filter is implemented as L parallel processes to data blocks associated with randomly selected seed samples. Due to random selection, no pixel or block is particularly favoured. This results in an equivalence of series and parallel processing, and enables the convergence of the filter to its optimum solution. The processing is local and is constrained to a limited set of pixel intensities so that outliers are excluded in clustering and pixel modification (filtering). Further, iterative pixel filtering uses steepest-ascent method guided by mean-shift clustering theory. This step also prevents pixel intensity drifting towards the cluster values in shaded regions.

2. IMAGE MODELING AND NONLINEAR SMOOTHING

Let the observed image *f(x,y)* be modelled as the superposition of three parts: the 'homogeneous' part $g_1(x,y)$, the structural part (e.g. edges) $g_2(x,y)$, and the 'noise' part *n(x,y)*,

$$f(x,y) = g_1(x,y) + g_2(x,y) + n(x,y) \qquad (1)$$

The nonlinear smoothing under consideration is to find the 'best' estimate (under a selected criterion), such that the resulted image $\hat{f}(x,y)$ is the edge preserving smoothing of the original

$$\hat{f}(x,y) \triangleq h(x,y) \odot f(x,y) \approx h(x,y) * g_1(x,y) + k g_2(x,y) \qquad (2)$$

where *h(x,y)* is a smoothing filter, and $\odot$ is an operator. To the structured image part, the desired output from the filter is $k\,g_2(r)$, i.e., the filter maintains image structures however allows a scale difference. To the homogeneous image part, the desired output of the filter is a local region *i* dependent constant $h(x,y) * g_1(x,y) = c_i$. Obviously, this is associated with a nonlinear filtering problem.

Let us consider the following criterion function on a local image window $D_{x,y}$ centred at *(x,y)*,

$$e(x,y) = \sum_{x',y' \in D_{x,y}} \left| \hat{f}(x,y) - f(x',y') \right|^m \quad (3)$$

The best estimation is to find $\hat{f}(x,y)$ and the operator $\odot$, such that $e(x,y)$ in (3) is minimized. For analysis purpose, let us first consider several special cases of linear filter, where $\odot$ in (2) is a convolution. For m = 2, the estimate $\hat{f}(x,y)$ is associated with the arithmetic mean of the samples within the windows $D_{x,y}$. The filter is a LS estimator which works well for images with homogeneous intensities, however, edges are blurred. For m = 1, the estimate is the median of the samples within the windows. Unlike the mean filter, the median filter does less blur to the edges and is more robust. Further, increasing m leads to an increased influence of outliers on the estimated values, causing an increased deviation to the desired filter. Obviously, linear filters cannot obtain the desired solution. Since the outliers are likely to be associated with image structures for noise-free images (or noisy images with high SNRs), we propose a novel nonlinear smoothing filter based on adaptive radial clustering. Some special features introduced to this filter are: (a) The filter avoids using and modifying outliers in local regions in order to maintain the structures in the image (detailed in Sections 5 and 6); (b) The filtering is an iterative process using the steepest-ascent method on an adaptively selected radial set of local data. Under the general framework of mean-shift theory, this guarantees that the output pixel values are shifted along the gradient direction towards the nearest homogeneous region. The clustering is performed locally on an adaptively selected radial set of data to prevent the deviation of cluster prototypes (in Sections 5 and 6); (c) A highly parallel filtering process is introduced, where seed samples, randomly re-generated for image partitions, are associated with L data blocks. This enables fast processing and the global optimum of the nonlinear filter (detailed in Section 4).

3. SYSTEM DESCRIPTION

The proposed algorithm for nonlinear smoothing filter can be subdivided into: (a) Finding seed samples for L-parallel processing. This is done by randomly choosing seed samples for image partitions. (b) L-parallel processing is then applied to local regions. For each data block, local adaptive radial clustering method is introduced (b.1) to find the cluster prototypes and (b.2) to modify pixel values subsequently towards the desired outputs. The number of local clusters is set to be adaptive according to the dynamic range of pixel intensities. To avoid influence of outlier pixels, clustering and filtering are only applied to pixels whose intensities are within a radial distance to the estimated cluster prototypes. The details are described in the following sections.

4. SELECTING SEEDS FOR PARALLEL PROCESSING

In order to facilitate parallel processing, random seed samples are generated for each image partition. This step is introduced for fast parallel processing and achieving global optimum of the nonlinear filter. The image is first partitioned into small and fixed blocks, whose centers are shifted according to the seed samples. Assuming the size of the block is W by W, the shift $[s_x \;\; s_y]^T$ in block location is determined by the seed samples, randomly generated from a 2D uniform distribution [1 W]. Seed samples are then associated with rectangular data blocks B_{ij}, with up-left corner coordinators

$$b_{i,j} = [s_x + (j-1)W \;\; s_y + (i-1)W]^T, \;\; 1 \le i, j \le M \quad (4)$$

where *(i,j)* are indices of the block, and $L=MxM$. Seed samples are re-generated for new image partitions. Therefore, no pixel, or block is particularly favoured. This is necessary to guarantee the equivalence of parallel processing and the conventional serial processing (i.e., one pixel is processed at each time).

5. STEEPEST-ASCENT METHOD FOR PIXEL FILTERING AND ITS ASSOCIATION TO MEAN-SHIFT

Apart from eliminating the influence of outliers through adaptive radial clustering (described in Section 6), a novel pixel modification (filtering) algorithm based on the steepest-ascent method is introduced. This stage is applied to guide pixel modification along the gradient direction towards the nearest cluster. It also prevents pixel intensity drifting towards the cluster prototypes in shaded regions. Within the intensity set $[c_1 - r \;\; c_2 + r]$, a pixel value is iterated towards the nearest local cluster prototype c_l using

$$f^{(n+1)}(x,y) = f^{(n)}(x,y) + a\,(c_l - f^{(n)}(x,y)) \quad (5)$$

where n is the iteration number, a is small positive number (set to be slowly decreasing with iterations) which controls the convergence speed and the steady state performance. This filtering method can be interpreted under the theoretical framework of mean-shift [8]. Mean-shift is an efficient nonparametric method for estimating the gradient of density in multivariate distributions. Let $\boldsymbol{z}$ be the m dimensional feature vector, $S_{\boldsymbol{z}}$ a small spherical window of radius r centred at $\boldsymbol{z}$, and $p(\boldsymbol{z})$ the estimated probability density function (pdf). The gradient of pdf in the window centre is shown to be proportional to the difference between the local mean and the centre value within the window, i.e., $E\{z \mid z \in S_z\} - z = \frac{r^2}{m+2} \frac{\nabla p(z)}{p(z)}$. The mean-shift always points to the direction of the maximum increase in density. This can be used to locate local high density regions in a feature space by means of the steepest-ascent algorithm, and to subsequently modify pixel values along that direction. In our case, the feature space used for pixel modification is the intensity histogram of pixels in the data block. Given an initial pixel from the histogram, the algorithm finds the closest high density region to this point within the radius of the searching window. The choice of the radius is important: A smaller radius can better eliminate the influence of outliers to the estimates at the expense of an increased risk of finding too 'local' extrema. In the next section, a new version of mean-shift clustering using adaptive radius is proposed.

6. ADAPTIVE LOCAL RADIAL CLUSTERING AND FILTERING

According to (2), the filter should have little influence to $g_2(x,y)$. The filter should leave some outliers untouched to preserve image structures (e.g. edges and lines), while filtering out other outliers (e.g. transition pixels from blurred edges). In order to obtain a selective behaviour in the presence of different kinds of outliers, the following three-stage process is proposed.

6.1. Determining the Number of Local Clusters

The number of local clusters is chosen adaptively according to the dynamic range of the intensity histogram. The intensity histogram in each block is first computed. The actual number of local clusters is then decided according to the histogram. For each block, let the maximum and minimum intensity values be $f_{\max}$ and $f_{\min}$. If $f_{\max} - f_{\min} < r_0$, where r_0 is a threshold, then the region is considered as homogeneous or low contrast, and the number of clusters is set to one. Otherwise, a maximum number of clusters K is chosen. In our experiments, the maximum number of clusters is set to 2 due to the small block size used.

6.2. Estimating Local Cluster Prototypes

Local cluster prototypes are initially determined by K-nearest neighbourhood method. The median of absolute differences (MAD) within each cluster is then computed, which is a robust estimator of local variance. Mean-shift iterations with radius $r_i = MAD_i$ are then carried out, starting from the initial clusters c_l. This leads to a so-called 'partial local segmentation'. The role of this stage is to define local partitions, so that pixels with different labels are modified differently. Further, each cluster is obtained from a high confidence subset of pixels. Note that the confidence is adaptively defined through the variable radius r_i.

6.3. Pixel Filtering

After initial clustering, the iteration process is only applied to pixels within the intensities of $[c_1 - r \;\; c_2 + r]$ in the block, according to the following distance measure

$$d_l(x,y) = \left| f(x,y) - c_l \right| / MAD_l \qquad l = 1,2 \qquad (6)$$

where MAD_l is the mean absolute difference in the lth cluster. If $d_1(x,y) \geq d_2(x,y)$, then pixel $f(x,y)$ is iterated using (5) towards c_2, otherwise towards c_1. Pixels outside the range $[c_1 - r \;\; c_2 + r]$ are excluded from filtering. A favourable effect is that pixels from outside this range, possibly belonging to undetected small clusters, are not affected. Hence, cluster validation problem is alleviated.

The proposed algorithm is briefly summarized in Table 1.

Initialise: iteration number n=0;
(a). Generate seeds and the associated blocks; Set n=n+1;
(b). L-parallel processing for each data block:
Compute intensity histogram;
Determine the number of clusters, initialise the prototypes;
Iteratively update prototypes by adaptive r mean-shift;
Iteratively modify pixels according to (5) and (6);
(c). Go back to (a), until either no block has been modified, or n exceeds the maximum iteration number $n > n_p$.

Table 1. Algorithm for nonlinear smoothing filter

7. RESULTS AND EVALUATIONS

Experimental were performed on a set of images with various degree of complexities, and good results were obtained. As an example, Fig.1 shows an original 'flower' image and the image resulted from the proposed filter. The parameters used in the experiments were W=9, r_0=9 and n_p=20. It is observed that homogeneous regions are smoothed and edges are well preserved. It is also observed that the amount of details preserved is mainly dependent on the selection of window W in the filter, as shown in the example of Fig.2. Larger W leads to less details, however, most visible structures in images were found to be remarkably stable with respect to W. Another interesting property is that increasing W has little effect on the shape of the structures, e.g., no visible corner rounding. The results also showed that there is a good accuracy in edge localization, which were partly contributed by the removing of outliers.

To evaluate the edge preserving property of the filter, comparisons were performed by applying a simple edge detection algorithm both to the original and filtered images. Examples are included in Fig.3. The edge detection method consisted of 2nd-order image derivatives (i.e., Laplacian of Gaussian) followed by thresholding. The results showed that the proposed filter indeed performs well in terms of edge preserving smoothing.

To verify the benefit of the filter to image segmentation, experiments were conducted by first using the proposed filter followed by a very simple segmentation method. As an example, Fig.4 shows the segmentation results with and without applying the proposed filter. All results showed that texture regions were well segmented after applying the proposed filter (e.g. (4a) and (4c), although there were a few spurious regions, partially due to the defect in our boundary extraction algorithm. Further, we observed that the proposed filter indeed produced partial image segmentation, especially for less complex images and in the high contrast regions. Preliminary comparisons to [9] indicated that the proposed method has obtained equal or improved results.

Further, it is observed that the program is fast. The executing time was 25 Sec. for a 256×256 'Lena' image on a single 133

MHz 486 PC. The processing time was slightly increased with increased W.

8. CONCLUSIONS

A novel nonlinear smoothing filter is presented. Using the image model and the criterion of the filter, the newly introduced filter strategies, which include constrained radial intensity range in clustering, steepest-ascent method for iterative pixel filtering and parallel re-generated random blocks, are shown to be effective in eliminating the influence of outliers, modifying pixels along the mean-shift towards the closest cluster, and enabling highly parallel processing. Our experimental results have also shown that the method has indeed generated good image smoothing meanwhile well preserving or enhancing the image structures. Therefore, the proposed method can be used for edge-preserving image smoothing. In addition, it may be used as pre-processing step for improving the results of image segmentation and edge estimation. The method is only suitable for high SNR images.

REFERENCES

[1] J. Sera, Image Analysis and Mathematical Morphology, Academic Press, New York, 1982.

[2] P. Salembier, L. Torres, F. Meyer and C. Gu, "Region-based video coding using mathematical morphology", Proceeding of the IEEE, Vol. 83, pp 843-856, 1995.

[3] R.N. Dave and R. Krishnapuram, "Robust clustering methods: a unified view", IEEE Trans. Fuzzy Systems, Vol. 5, No. 2, pp 270-293, 1997.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion", IEEE Trans. PAMI, Vol. 12, No. 7, pp 629-639, 1990.

[5] M.Nitzberg, D.Mumford, T. Shiota, Filtering, Segmentation and Depth, Springer-Verlag Berlin Heidelberg, 1993.

[6] B. Fischl and E. L. Schwartz, "Adaptive nonlocal filtering: a fast alternative to anisotropic diffusion for image enhancement", IEEE Trans. PAMI, Vol. 21, No.1, pp 42-48, 1999.

[7] J-P.Wang, "Stochastic relaxation on partitions with connected components and its application to image segmentation", IEEE Trans. PAMI, Vol. 20, No. 6, pp 619-636, 1996.

[8] Y. Cheng, "Mean shift, mode seeking and clustering", IEEE Trans. PAMI, Vol.17, pp790-799, 1995.

[9] D. Comaniciu, P. Meer: "Mean shift analysis and applications", in Proc. 7th Intern. Conf. Computer Vision, Greece, pp1197-1203, 1999.

[10] P.Bakker, L.J.van Vliet, P.W.Verbeek, "Edge preserving orientation adaptive filtering", in Proc. 5th Annual Conf. Advanced School for Computing and Imaging, Delft, The Netherlands, pp 207-213, June,1999.

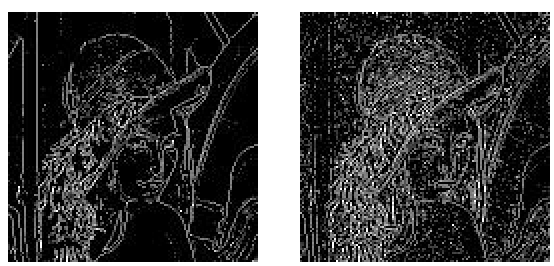
(1.a) original image (1.b) filtered image

Fig.1 Original and the filtered images.

(2.a) edges from (1.b) (2.b) edges from (1.a)

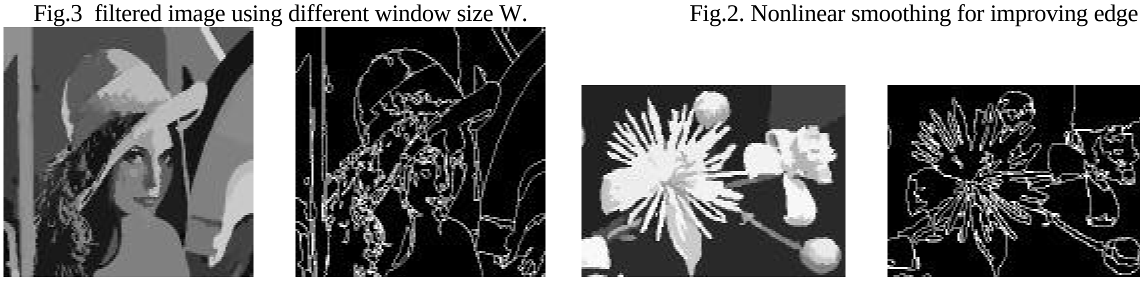
(3.a) filtered (W=15) (3.b) filtered image (W=33)

Fig.3 filtered image using different window size W.



(2.c) edges from filtered image (2.d) edges from original image

Fig.2. Nonlinear smoothing for improving edge detection.



(4.a) segmented image (4.b) boundaries from (4.a) (4.c) segmented image (4.d) boundaries from (4.c)

Fig.4 Nonlinear filtering followed by image segmentation and boundary extraction