

Image Filtering Driven by Level Curves

Ajit Rajwade, Arunava Banerjee and Anand Rangarajan

Department of CISE, University of Florida, Gainesville, USA
(E-mail: {avr,arunava,anand}@cise.ufl.edu)

Abstract. This paper presents an approach to image filtering that is driven by the properties of the iso-valued level curves of the image and their relationship with one another. We explore the relationship of our algorithm to existing probabilistically driven filtering methods such as those based on kernel density estimation, local-mode finding and mean-shift. Extensive experimental results on filtering gray-scale images, color images, gray-scale video and chromaticity fields are presented. In contrast to existing probabilistic methods, in our approach, the selection of the parameter that prevents diffusion across the edge is robustly decoupled from the smoothing of the density itself. Furthermore, our method is observed to produce better filtering results for the same settings of parameters for the filter window size and the edge definition.

1 Introduction

Filtering of images has been one of the most fundamental problems studied in low-level vision and signal processing. Over the past decades, several techniques for data filtering have been proposed with impressive results on practical applications in image processing. As straightforward image smoothing is known to blur across significant image structures, several anisotropic approaches to image smoothing have been developed using partial differential equations (PDEs) with stopping terms to control image diffusion in different directions [1]. The PDE-based approaches have been extended to filtering of color images [2] and chromaticity vector fields [3]. Other popular approaches to image filtering include adaptive smoothing [4] and kernel density estimation based algorithms [5]. All these methods produce some sort of weighted average over an image neighborhood for the purpose of data smoothing, where the weights are obtained from the difference between the intensity values of the central pixel and the pixels in the neighborhood, or from the pixel gradient magnitudes. Beyond this, techniques such as bilateral filtering [6] produce a weighted combination that is also influenced by the relative location of the central pixel and the neighborhood pixels. The highly popular mean-shift procedure [7], [8] is grounded in similar ideas as bilateral filtering, with the addition that the neighborhood around a pixel is allowed to change dynamically until a convergence criterion is met. The authors prove that this convergence criterion is equivalent to finding the mode of a local density built jointly on the spatial parameters (image *domain*) and the intensity parameters (image *range*).

In this paper, we present a new approach to data filtering that is rooted in simple yet elegant geometric intuitions. At the core of our theory is the representation of an image as a function that is at least C_0 continuous everywhere. A key property of the image level sets is used to drive the diffusion process, which we then incorporate in a framework of dynamic neighborhoods *à la* mean-shift. We demonstrate the relationship of our method to many of the existing filtering techniques such as those driven by kernel density estimation. The efficacy of our approach is supported with extensive experimental results. To the best of our knowledge, *ours is the first attempt to explicitly utilize image geometry (in terms of its level curves)* for this particular application.

This paper is organized as follows. Section 2 presents the key theoretical framework. Section 3 presents extensions to our theory. In section 4, we present the relationship between our method and mean-shift. Extensive experimental results are presented in section 5, and we present further discussions and conclusions in section 6.

2 Theory

Consider an image over a discrete domain $\Omega = \{1, \dots, H\} \times \{1, \dots, W\}$ where the intensity of each discrete location (x, y) is given by $I(x, y)$. Moreover consider a neighborhood $\mathcal{N}(x_i, y_i)$ around the pixel (x_i, y_i) . It is well-known that a simple averaging of all intensity values in $\mathcal{N}(x_i, y_i)$ will blur edges, so a weighted combination is calculated, where the weight of the j^{th} pixel is given by $w^{(1)}(x_j, y_j) = g(|I(x_i, y_i) - I(x_j, y_j)|)$ for a non-increasing function $g(\cdot)$ to facilitate anisotropic diffusion, with common examples being $g(z) = e^{-\frac{z^2}{\sigma^2}}$ or $g(z) = \frac{\sigma^2}{\sigma^2 + z^2}$, or their truncated versions. This approach is akin to the kernel density estimation (KDE) approach proposed in [5], where the filtered value of the central pixel is calculated as:

$$\hat{I}(x_i, y_i) = \frac{\sum_{(x_j, y_j) \in \mathcal{N}(x_i, y_i)} I(x_j, y_j) K(I(x_j, y_j) - I(x_i, y_i); W_r)}{\sum_{(x_j, y_j) \in \mathcal{N}(x_i, y_i)} K(I(x_j, y_j) - I(x_i, y_i); W_r)}. \quad (1)$$

Here the kernel K centered at $I(x_i, y_i)$ (and parameterized by W_r) is related to the function g and determines the weights. The major limitations of the kernel based approach to anisotropic diffusion are that the entire procedure is sensitive to the parameter W_r and the size of the neighborhood, and might suffer from a small-sample size problem. Furthermore, in a discrete implementation, for any neighborhood size larger than 3×3 , the procedure depends only on the actual pixel values and does not account for any gradient information, whereas in a filtering application, it is desirable to place greater importance on those regions of the neighborhood where the gradient values are lower.

Now consider that the image is treated as a continuous function $I(x, y)$ of the spatial variables, by interpolating in between the pixel values. The earlier

discrete average is replaced by the following continuous average to update the value at (x_i, y_i) :

$$\hat{I}(x_i, y_i) = \frac{\int \int_{\mathcal{N}(x_i, y_i)} I(x, y) g(|I(x, y) - I(x_i, y_i)|) dx dy}{\int \int_{\mathcal{N}(x_i, y_i)} g(|I(x, y) - I(x_i, y_i)|) dx dy}. \quad (2)$$

The above formula is usually not available in closed form. We now show a principled approximation to this formula, by resorting to geometric intuition. Imagine a contour map of this image, with multiple iso-intensity level curves $C_m = \{(x, y) | I(x, y) = \alpha_m\}$ (referred to henceforth as ‘level curves’) separated by an intensity spacing of Δ . Consider a portion of this contour map in a small neighborhood centered around the point (x_i, y_i) (see Figure 1(a)). Those regions where the level curves (separated by a fixed intensity spacing) are closely packed together correspond to the higher-gradient regions of the neighborhood, whereas in lower-gradient regions of the image, the level curves lie far away from one another. Now as seen in Figure 1(a), this contour map induces a tessellation of the neighborhood into some K facets, where each facet corresponds to a region in between two level curves of intensity α_m and $\alpha_m + \Delta$, bounded by the rim of the neighborhood. Let the area a_k of the k^{th} facet of this tessellation be denoted as a_k . Now, if we make Δ sufficiently small, we can regard even the facets from high-gradient regions as having constant intensity value $I_k = \alpha_m$. This now leads to the following weighted average in which the weighting function has a very clean geometric interpretation, unlike the arbitrary choice for $w^{(1)}$ in the previous technique:

$$\hat{I}(x_i, y_i) = \frac{\sum_{k=1}^K a_k I_k g(|I_k - I(x_i, y_i)|)}{\sum_{k=1}^K a_k g(|I_k - I(x_i, y_i)|)}. \quad (3)$$

As the number of facets is typically much larger than the number of pixels, and given the fact that the facets have arisen from a locally smooth interpolation method to obtain a continuous function from the original digital pixel values, we now have a more robust average than that provided by Equation 1. To introduce anisotropy, we still require the stopping term $g(|I_k - I(x_i, y_i)|)$ to prevent smearing across the edge, just as in Equation 1.

Equation 2 essentially performs an integration of the intensity function over the domain $\mathcal{N}(x_i, y_i)$. If we now perform a change of variables transforming the integral on (x, y) to an integral over the range of the image, we obtain the

expression

$$\begin{aligned} \hat{I}(x_i, y_i) &= \frac{\int \int_{\mathcal{N}(x_i, y_i)} I(x, y) w^{(1)}(x, y) dx dy}{\int \int_{\mathcal{N}(x_i, y_i)} w^{(1)}(x, y) dx dy} = \frac{\int_{q=q_1}^{q=q_2} \int_{\mathcal{C}(q)} \frac{qg(|q - I(x_i, y_i)|)}{|\nabla I|} dl dq}{\int_{q=q_1}^{q=q_2} \int_{\mathcal{C}(q)} \frac{g(|q - I(x_i, y_i)|)}{|\nabla I|} dl dq} \\ &= \frac{\lim_{\Delta \rightarrow 0} \sum_{\alpha=q_1}^{q_2} \int_{q=\alpha}^{\alpha+\Delta} \int_{\mathcal{C}(q)} \frac{qg(|q - I(x_i, y_i)|)}{|\nabla I|} dl dq}{\lim_{\Delta \rightarrow 0} \sum_{\alpha=q_1}^{q_2} \int_{q=\alpha}^{\alpha+\Delta} \int_{\mathcal{C}(q)} \frac{g(|q - I(x_i, y_i)|)}{|\nabla I|} dl dq} \quad (4) \end{aligned}$$

where $\mathcal{C}(q) = \mathcal{N}(x_i, y_i) \cap f^{-1}(q)$, $q_1 = \inf\{I(x, y) | (x, y) \in \mathcal{N}(x_i, y_i)\}$, $q_2 = \sup\{I(x, y) | (x, y) \in \mathcal{N}(x_i, y_i)\}$ and l stands for a tangent along the curve $f^{-1}(q)$. This approach is inspired by the smooth co-area formula for regular functions [9] which is given as

$$\int_{\Omega} \phi(u) |\nabla u| dx dy = \int_{-\infty}^{+\infty} \text{Length}(\gamma_q) \phi(q) dq \quad (5)$$

where γ_q is the level set of u at the intensity q and $\phi(u)$ represents a function of u . Note that the term $\int_{q=\alpha}^{\alpha+\Delta} \int_{\mathcal{C}(q)} \frac{dl dq}{|\nabla I|}$ in Equation 4 actually represents the area in $\mathcal{N}(x_i, y_i)$ that is trapped between two contours whose intensity value differs by Δ . Previous work from [10] and [11] considers this quantity when normalized by $|\Omega|$ to be actually equal to the probability that the intensity value lies in the range $[\alpha, \alpha + \Delta]$. Bearing this in mind, Equation 3 now acquires the following *probabilistic* interpretation:

$$\hat{I}(x_i, y_i) = \frac{\sum_{\alpha=q_1}^{q_2} \Pr(\alpha < I < \alpha + \Delta | \mathcal{N}) \alpha g(|\alpha - I(x_i, y_i)|)}{\sum_{\alpha=q_1}^{q_2} \Pr(\alpha < I < \alpha + \Delta | \mathcal{N}) g(|\alpha - I(x_i, y_i)|)} \quad (6)$$

As $\Delta \rightarrow 0$, this produces an increasingly better approximation to Equation 2.

It should be pointed out that there exist methods such as adaptive filtering [4], [12] in which the weights in Equation 1 are obtained as $w^{(2)}(x_j, y_j) = g(|\nabla I(x_j, y_j)|)$. These methods place more importance on the lower-gradient *pixels* of the neighborhood, but do not exploit level curve relationships in the way we do, and the choice of the weighting function does not have the geometric interpretation that exists in our technique. There also exists an extension to the standard neighborhood filter in Equation 1 reported in [13], which performs a weighted least squares polynomial fit to the intensity values (of the pixels) in the neighborhood of a location (x, y) . The value of this polynomial at (x, y) is then considered to be the smoothed intensity value. This technique differs from

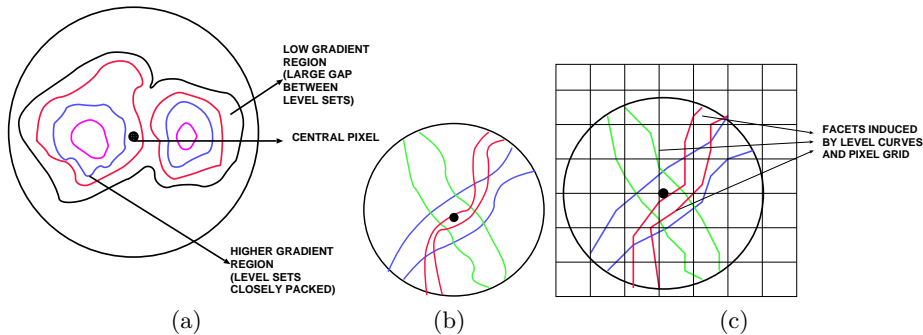


Fig. 1. (a) An image contour map with high and low gradient regions in a neighborhood around a pixel (dark dot). (b) A contour map of an RGB image in a neighborhood. The red, green and blue contours correspond to contours of the R,G,B channels respectively. The tessellation induced by the above level-curve pairs contains 19 facets. (c) A tessellation induced by RGB level curve pairs and the square pixel grid.

the one we present here in two fundamental ways. Unlike our method, it does not use areas between level sets as weights to explicitly perform a weighted averaging. Secondly as proved in [13], its limiting behavior when $W_r \rightarrow 0$ and $|\mathcal{N}(x, y)| \rightarrow 0$ resembles the geometric heat equation with a linear polynomial, and resembles higher order PDEs when the degree of the polynomial is increased. Our method is the true continuous form of the KDE-based filter from Equation 1. This KDE-based filter limits to the Perona-Malik equation, as proved in [13].

3 Extensions of Our Theory

3.1 Color Images

We now extend our technique to color (RGB) images. Consider a color image defined as $I(x, y) = (R(x, y), G(x, y), B(x, y)) : \Omega \rightarrow \mathcal{R}^3$ where $\Omega \subset \mathcal{R}^2$. In color images, there is no concept of a single iso-contour with constant values of all three channels. Hence it is more sensible to consider an overlay of the individual iso-contours of the R, G and B channels. The facets are now induced by a tessellation involving the intersection of three iso-contour sets within a neighborhood, as shown in Figure 1(b). Each facet represents those portions of the neighborhood for which $\alpha_R < R(x, y) < \alpha_R + \Delta_R, \alpha_G < G(x, y) < \alpha_G + \Delta_G, \alpha_B < B(x, y) < \alpha_B + \Delta_B$. The probabilistic interpretation for the update on the R,G,B values is as follows

$$\hat{R}(x_i, y_i), \hat{G}(x_i, y_i), \hat{B}(x_i, y_i) = \frac{\sum_{\beta} \Pr[\beta < (R, G, B) < \beta + \Delta | \mathcal{N}] \beta g(R, G, B)}{\sum_{\beta} \Pr[\beta < (R, G, B) < \beta + \Delta | \mathcal{N}] g(R, G, B)}$$

where $\beta = (\alpha_R, \alpha_G, \alpha_B)$, $\Delta = (\Delta_R, \Delta_G, \Delta_B)$ and $g(R, G, B) = g(|R - R(x_i, y_i)| + |G - G(x_i, y_i)| + |B - B(x_i, y_i)|)$. Note that in this case, $I(x, y)$ is a function from a subset of \mathcal{R}^2 to \mathcal{R}^3 , and hence the three-dimensional joint *density* is ill-defined in the sense that it is defined strictly on a 2D subspace of \mathcal{R}^3 . However given that the implementation considers joint cumulative interval measures, this does not pose any problem in a practical implementation. We wish to emphasize that the averaging of the R,G,B values is performed in a strictly coupled manner, all affected by the *joint* cumulative interval measure.

3.2 Chromaticity Fields

Previous research on filtering chromaticity noise (which affects only the direction and not the magnitude of the RGB values at image pixels) includes the work in [3] using PDEs specially tuned for unit-vector data, and the work in [5] (page 142) using kernel density estimation for directional data. The more recent work on chromaticity filtering in [14] actually treats chromaticity vectors as points on a Grassmann manifold $\mathcal{G}_{1,3}$ as opposed to treating them as points on \mathcal{S}^2 , which is the approach presented here and in [5] and [3].

We extend our theory from the previous section to unit vector data and incorporate it in a mean-shift framework for smoothing. Let $I(x, y) : \Omega \rightarrow \mathcal{R}^3$ be the original RGB image, and let $J(x, y) : \Omega \rightarrow \mathcal{S}^2$ be the corresponding field of chromaticity vectors. A possible approach would involve interpolating the chromaticity vectors by means of commonly used spherical interpolants to create a continuous function, followed by tracing the level curves of the individual unit-vector components $\mathbf{v}(x, y) = (v_1(x, y), v_2(x, y), v_3(x, y))$ and computing their intersection. However for ease of implementation for this particular application, we resorted to a different strategy. If the intensity intervals $\Delta = (\Delta_R, \Delta_G, \Delta_B)$ are chosen to be fine enough, then each facet induced by a tessellation that uses the level curves of the R, G and B channel values, can be regarded as having a constant color value, and hence the chromaticity vector values within that facet can be regarded as (almost) constant. Therefore it is possible to use just the R,G,B level curves for the task of chromaticity smoothing as well. The update equation is very similar to Equation 7 with the R,G,B vectors replaced by their unit normalized versions. However as the averaging process does not preserve the unit norm, the averaged vector needs to be renormalized to produce the spherical weighted mean.

3.3 Gray-scale Video

For the purpose of this application, the video is treated as a single 3D signal (volume). The extension in this case is quite straightforward, with the areas between level curves being replaced by volumes between the level surfaces at nearby intensities. However we take into account the causality factor in defining the temporal component of the neighborhood around a pixel, by performing the averaging at each pixel over frames only from the past.

4 Level Curve Based Filtering in a Mean-shift Framework

All the above techniques are based on an averaging operation over only the image intensities (i.e. in the range domain). On the other hand, techniques such as bilateral filtering [6] or local mode-finding [15] combine both range and spatial domain, thus using weights of the form $w_j = g^{(s)}((x_i - x_j)^2 + (y_i - y_j)^2)g^{(r)}(|I(x_i, y_i) - I(x_j, y_j)|)$ in Equation 1, where $g^{(s)}$ and $g^{(r)}$ affect the spatial and range kernels respectively. The mean-shift framework [8] is based on similar principles, but changes the filter window dynamically for several iterations until it finds a local mode of the joint density of the spatial and range parameters, estimated using kernels based on the functions $g^{(r)}$ and $g^{(s)}$. Our level curve based approach fits easily into this framework with the addition of a spatial kernel. One way to do this would be to consider the image as a surface embedded in 3D (a Monge patch), as done in [16], and compute areas of patches in 3D for the probability values. However such an approach may not necessarily favor the lower gradient areas of the image. Instead we adopt another method wherein we assume two additional functions of x and y , namely $X(x, y) = x$ and $Y(x, y) = y$. We compute the joint probabilities for a range of values of the joint variable (X, Y, I) by drawing local level sets and computing areas in 2D. Assuming a uniform spatial kernel for $g^{(s)}$ within a radius W_s and a rectangular kernel on the intensity for $g^{(r)}$ with threshold value W_r (though our core theory is unaffected by other choices), we now perform the averaging update on the vector $(X(x, y), Y(x, y), I(x, y))$, as opposed to merely on $I(x, y)$ as was done in Equation 6. This is given as:

$$(X(x_i, y_i), Y(x_i, y_i), \hat{I}(x_i, y_i)) = \frac{\sum_{k=1}^K (x_k, y_k, I_k) a_k g^{(r)}(|I_k - I(x_i, y_i)|)}{\sum_{k=1}^K a_k g^{(r)}(|I_k - I(x_i, y_i)|)}. \quad (7)$$

In the above equation (x_k, y_k) stands for a representative point (say, the centroid) of the k^{th} facet of the induced tessellation¹, and K is the total number of facets within the specified spatial radius. Note that the area of the k^{th} facet, i.e. a_k , can also be interpreted as the joint probability for the event $\tilde{x} < X(x, y) < \tilde{x} + \Delta_x, \tilde{y} < Y(x, y) < \tilde{y} + \Delta_y, \alpha < I(x, y) < \alpha + \Delta$, if we assume a uniform distribution over the spatial variables x and y . Here Δ is the usual intensity binwidth, (Δ_x, Δ_y) are the pixel dimensions, and (\tilde{x}, \tilde{y}) is a pixel grid-point. The main difference between our approach and all the aforementioned range-spatial domain approaches is the fact that we naturally incorporate a weight in favor of the lower-gradient areas of the filter neighborhood. Hence the mean-shift vector in our case will have a stronger tendency to move towards the region of the neighborhood where the local intensity change is as low as possible (even if a uniform spatial kernel is used). Moreover just like conventional mean shift,

¹ The notion of the centroid will become clearer in Section 5.

our iterative procedure is guaranteed to converge to a mode of the local density in a finite number of steps, by exploiting the fact that the weights at each point (i.e. the areas of the facets) are positive. Hence Theorem 5 of [7] can be readily invoked. This is because in Equation 7, the threshold function $g^{(r)}$ for the intensity is the rectangular kernel, and hence the corresponding update formula is equivalent to one with a weighted rectangular kernel, with the weights being determined by the areas of the facets.

A major advantage of our technique is that the parameter Δ can be set to as small a value as desired (as it just means that more and more level curves are being used), and the interpolation gives rise to a robust average. This is especially useful in the case of small neighborhood sizes, as the intensity quantization is now no more limited by the number of available pixels. In conventional mean-shift, the proper choice of bandwidth is a highly critical issue, as very few samples are available for the local density estimate. Though variable bandwidth procedures for mean-shift algorithms have been developed extensively, they themselves require either the tuning of other parameters using rules of thumb, or else some expensive exhaustive searches for the automatic determination of the bandwidth [17], [18]. Although our method does require the selection of W_s and W_r , the filtering results are less sensitive to the choice of these parameters in our method than in standard mean shift.

5 Experimental Results

In this section we present experimental results to compare the performance of our algorithm in a mean shift framework w.r.t. conventional kernel-based mean shift, as well as to two recent algorithms that are closely related to mean-shift: UINTA [19] and NL-Means [20]. For our algorithm, we obtain a continuous function approximation to the digital image, by means of piecewise linear interpolants fit to a triple of intensity values in half-pixels of the image (in principle, we could have used any other smooth interpolant). The corresponding level sets for such a function are also very easy to trace, as they are just segments within each half-pixel. The level sets induce a polygonal tessellation. We choose to split the polygons by the square pixel boundaries as well as the pixel diagonals that delineate the half-pixel boundaries, thereby convexifying all the polygons that were initially non-convex (see Figure 1(c)). Each polygon in the tessellation can now be characterized by the x, y coordinates of its centroid, the intensity value of the image at the centroid, and the area of the polygon. Thus, if the intensity value at grid location x_i, y_i is to be smoothed, we choose a window of spatial radius W_s and intensity radius W_r around $(x_i, y_i, I(x_i, y_i))$, over which the averaging is performed. In other words, the averaging is performed only over those locations x, y for which $(x - x_i)^2 + (y - y_i)^2 < W_s^2$ and $|I(x, y) - I(x_i, y_i)| < W_r$. We would like to point out that *though the interpolant used for creating the continuous image representation is indeed isotropic in nature, this still does not make our filtering algorithm isotropic*. This is because polygonal regions, whose intensity value does not satisfy the constraint $|I(x, y) - I(x_i, y_i)| < W_r$, do not contribute



Fig. 2. Leftmost column: original images, Second from left: degraded images with zero mean Gaussian noise of std. dev. 0.003, Second from right: results obtained by our algorithm, and rightmost column: mean shift with Gaussian kernel (right column). Both both methods, $W_s = W_r = 3$. *VIEWED BEST when ZOOMED in the pdf file.*

to the averaging process (see the stopping term in Equation 3), and hence the contribution from pixels with very different intensity values will be nullified.

5.1 Gray-scale Images

We ran our filtering algorithm over four arbitrarily chosen images from the popular Berkeley image dataset ², and the Lena image. To all these images, zero mean Gaussian noise of variance 0.003 (per unit gray-scale range) was added. The filtering was performed using $W_s = W_r = 3$ for our algorithm and compared to mean-shift using Gaussian and Epanechnikov kernels with the same parameter. Our method produced superior filtering results to conventional mean shift with both Gaussian and Epanechnikov kernels. The results for our method, for Gaussian kernel mean shift and for UINTA are displayed in Figure 2. The visually superior appearance was confirmed objectively with mean squared error (MSE) values in Table 1. *It should be noted that the aim was to compare our method to*

² <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

Image	M1	M2	M3	M_{UINTA}	M_{NL}	MSE
1	110.95	176.57	151.27	280.7	130.7	181.27
2	53.85	170.18	106.32	95.43	127.48	193.5
3	106.64	185.15	148.379	121.3	147.41	191.76
4	113.8	184.77	153.577	127.4	147.98	190
Lena	78.42	184.16	128.04	101.5	125.38	194.82

Table 1. MSE for filtered images using (M1) = Our method with $W_s = W_r = 3$, using (M2) = Mean shift with Gaussian kernels with $W_s = W_r = 3$ and (M3) = Mean shift with Gaussian kernels with $W_s = W_r = 5$, M_{UINTA} = MSE with UINTA method with neighborhood radius 9, smoothing parameter $h = 10$ (similar to W_r), 1000 samples for density estimate and 30 iterations per pixel, and M_{NL} = MSE with NL-means with search window size 18×18 , neighborhood size 5×5 , and smoothing parameter $h = 5$ (similar to W_r). MSE = mean-squared error in the corrupted image. Intensity scale is from 0 to 255.

standard mean shift for the exact same setting of the parameters W_r and W_s , as they have the same meaning in all these algorithms. Although increasing the value of W_r will provide more samples for averaging, this will allow more and more intensity values to leak across edges. Moreover, in Table 1, we also compare our method to NL-means [20] and UINTA [19], again for similar parameter settings. Further empirical results with our algorithm (using $W_s = W_r = 5$) were obtained on Lansel’s benchmark dataset [21]. The dataset contains noisy versions of 13 different images. Each noisy image is obtained from one of three noise models: additive Gaussian, Poisson, and multiplicative noise model, for one of five different values of the noise standard deviation $\sigma \in \{\frac{5}{255}, \frac{10}{255}, \frac{15}{255}, \frac{20}{255}, \frac{25}{255}\}$, leading to a total of 195 images. Despite the fact that we did not tweak any parameters depending on the noise model (we chose $W_r = W_s = 5$), we produced excellent denoising results. The average MSE and MSSIM (an image quality metric defined in [21]) are shown in the plots in Figure 3. We have also displayed the denoised versions of a fingerprint image from this dataset under three different values of σ for additive noise in Figure 3.

5.2 Color Images

Similar experiments were run on colored versions of the same four images from the Berkeley dataset. The original images were degraded by zero mean Gaussian noise of variance 0.003 (per unit intensity range), added independently to the R,G,B channels. For our method, independent interpolation was performed on each channel and the joint densities were computed as described in the previous sections. Level sets at intensity gaps of $\Delta_R = \Delta_G = \Delta_B = 1$ were traced in every half pixel. Experimental results were compared with conventional mean shift using a Gaussian kernel. The parameters chosen for both algorithms were $W_s = W_r = 6$. Despite the documented advantages of color spaces such as Lab [5], all experiments were performed in the R,G,B space for the sake of simplicity,

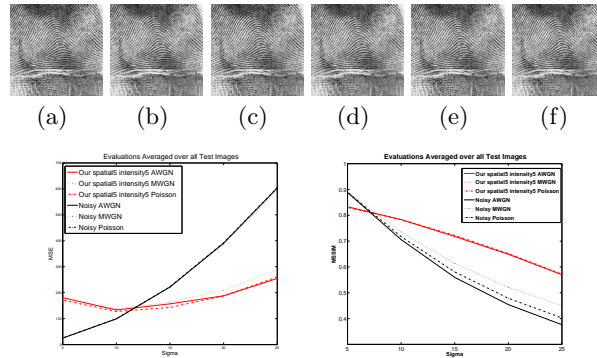


Fig. 3. First row: (a), (c) and (e): Fingerprint image subjected to additive Gaussian noise of std. dev. $\sigma = \frac{5}{255}$, $\frac{10}{255}$ and $\frac{15}{255}$ respectively. (b), (d) and (f): Denoised versions of (a), (c) and (e) respectively. Second row: A plot of the performance of our algorithm on the Lansel dataset, averaged over all images from each noise model (Additive Gaussian (AWGN), multiplicative Gaussian (MWGN) and Poisson) and over all five σ values, using MSE (left) and MSSIM (right) as the metric. *VIEWED BEST when ZOOMED in the pdf file (in color).*

and also because many well-known color de-noising techniques operate in this space [2]. As seen in Figure 4 and Table 2, our method produced better results than Gaussian kernel mean shift for the chosen parameter values.

5.3 Experiments with Chromaticity Vectors and Video

Two color images were synthetically corrupted with chromaticity noise altering just the direction of the color-triple vector. These images are shown in Figure 5. These images were filtered using our method and Gaussian kernel mean shift with a spatial window of size $W_s = 4$ and a chromaticity threshold of $W_r = 0.1$ radians. Note that in this case, the distance between two chromaticity vectors \mathbf{v}_1 and \mathbf{v}_2 is defined to be the length of the arc between the two vectors along the great circle joining them, which turns out to be $\theta = \cos^{-1} \mathbf{v}_1^T \mathbf{v}_2$. The specific expression for the joint spatial-chromaticity density using the Gaussian kernel was $e^{-\frac{(x-x_i)^2+(y-y_i)^2}{2W_s^2}} e^{-\frac{\theta^2}{2W_r^2}}$. The filtered images using both methods are shown in Figure 5. Despite the visual similarity of the output, our method produced a mean-squared error of 378 and 980.8, as opposed to 534.9 and 1030.7 for Gaussian kernel mean shift.

We also performed an experiment on video de-noising using the David sequence³. The first 100 frames from the sequence were extracted and artificially degraded with zero mean Gaussian noise of variance 0.006. Two frames of the corrupted and de-noised (using our method) sequence are shown in Figure 6, as

³ obtained from <http://www.cs.utoronto.ca/~dross/ivt/>



Fig. 4. Left column: original images, second from left: Degraded images with zero mean Gaussian noise of std. dev. 0.003, second from right: results by our algorithm (left column) and Rightmost: mean shift with Gaussian kernel (right column). Both both methods, $W_s = W_r = 6$. *Viewed best when ZOOMED in the pdf file (in color).*

also a temporal slice through the entire video sequence (for the tenth row of each frame). For this experiment, the value of Δ was set to 8 in our method.

Image	M1	M2	M3	MSE
1	319.88	496.7	547.9	572.54
2	354.76	488.7	543.4	568.69
3	129.12	422.79	525.48	584.24
4	306.14	477.25	526.8	547.9

Table 2. MSE for filtered images using (M1) = Our method with $W_s = W_r = 6$, using (M2) = Mean shift with Gaussian kernels with $W_s = W_r = 6$ and (M3) = Mean shift with Epanechnikov kernels with $W_s = W_r = 6$. MSE = mean-squared error in the corrupted image. Intensity scale is from 0 to 255 for each channel.

6 Discussion

We have presented a new method for image denoising, whose principle is rooted in the notion that the lower-gradient portions of an image inside a neighborhood around a pixel should contribute more to the smoothing process. The geometry of the image level sets (and the fact that the spatial distance between level sets is inversely proportional to the gradient magnitudes) is the driving force behind our algorithm. We have linked our approach to existing probability-density based approaches, and our method has the advantage of robust decoupling of the edge definition parameter from the density estimate. In some sense, our method

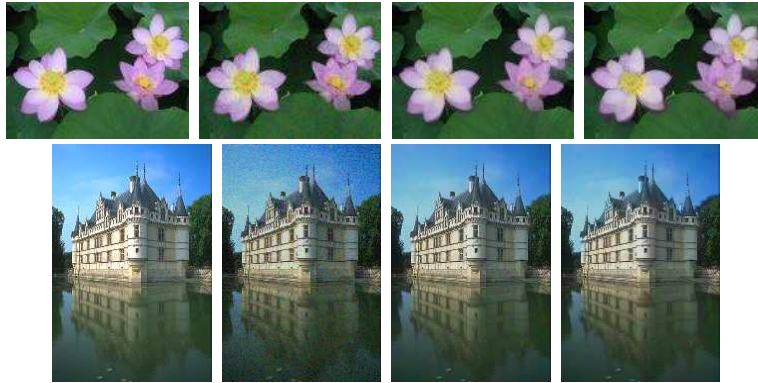


Fig. 5. Two images and their corrupted versions obtained by adding chromaticity noise (first and second columns respectively). Results obtained by filtering with our method (third column), and with Gaussian mean shift (fourth column). *Viewed best when ZOOMED in the pdf file (in color).*

can be viewed as a continuous version of mean-shift. It should be noted that a modification to standard mean-shift based on simple image up-sampling using interpolation will be an approximation to our area-based method (given the same interpolant). We have performed extensive experiments on gray-scale and color images, chromaticity fields and video sequences. To the best of our knowledge, ours is the *first piece of work on denoising which explicitly incorporates the relationship between image level curves and uses local interpolation between pixel values* in order to perform filtering. Future work will involve a more detailed investigation into the relationship between our work and that in [16], by computing the areas of the contributing regions with explicit treatment of the



Fig. 6. First two images: frames from the corrupted sequence. Third and fourth: images filtered by our algorithm. Fifth and sixth images: a slice through the tenth row of the corrupted and filtered video sequences. The images are numbered left to right, top to bottom.

image $I(x, y)$ as a surface embedded in 3D. Secondly, we also plan to develop topologically inspired criteria to automate the choice of the spatial neighborhood and the parameter W_r for controlling the anisotropic smoothing.

References

1. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on PAMI* **12**(7) (1990) 629–639
2. Tschumperle, D., Deriche, R.: Vector-valued image regularization with pdes : A common framework for different applications. *IEEE Trans. on PAMI* **27**(4) (2005) 506–517
3. Tang, B., Sapiro, G.: Color image enhancement via chromaticity diffusion. *IEEE Trans. on Image Processing* **10** (1999) 701–707
4. Saint-Marc, P., Chen, J., Medioni, G.: Adaptive smoothing: a general tool for early vision. *IEEE Trans. on PAMI* **13**(6) (1991) 514–520
5. Plataniotis, K., Venetsanopoulos, A.: Color image processing and applications. Springer-Verlag New York (2000)
6. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *ICCV*. (1998) 839–846
7. Cheng, Y.: Mean shift, mode seeking and clustering. *IEEE Trans. on PAMI* **17**(8) (1995) 790–799
8. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Trans. on PAMI* **24**(5) (2002) 603–619
9. Chan, T., Shen, J.: *Image Processing and Analysis: Variational, PDE, wavelets, and stochastic methods*. SIAM (2005)
10. Hadjidemetriou, E., Grossberg, M., Nayar, S.: Histogram preserving image transformations. *IJCV* **45**(1) (2001) 5–23
11. Rajwade, A., Banerjee, A., Rangarajan, A.: Probability density estimation using isocontours and isosurfaces: applications to information-theoretic image registration. *IEEE Trans. on PAMI* **31**(3) (2009) 475–491
12. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *IVC* **22** (2004) 73–81
13. Buades, A., Coll, B., Morel, J.M.: Neighborhood filters and PDEs. *Numerische Mathematik* **105**(1) (2006) 1–34
14. Subbarao, R., Meer, P.: Discontinuity preserving filtering over analytic manifolds. In: *CVPR*. (2007)
15. van de Weijer, J., van den Bloomgard, R.: Local mode filtering. In: *CVPR*. Volume 2. (2001) 428–436
16. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. *IEEE Trans. on Image Processing* **7** (1998) 310–318
17. Comaniciu, D.: An algorithm for data-driven bandwidth selection. *IEEE Trans. on PAMI* **25** (2003) 281–288
18. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: *ICCV*. (2001) 438–445
19. Awate, S., Whitaker, R.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. on PAMI* **28**(3) (2006) 364–376
20. Buades, A., Coll, B., Morel, J.M.: Nonlocal image and movie denoising. *IJCV* **76**(2) (2008) 123–139
21. Lansel, S.: About DenoiseLab. Available from <http://www.stanford.edu/~slansel/DenoiseLab/documentation.htm>