

Non-Local Means Denoising

fdmw97

Algorithm Description

The Non-Local Means algorithm takes an image v consisting of I pixels. Then for each pixel $i \in I$ produces a new value for i that is the result of a weighted average of all the pixels in the image. This is described by the formula

$$NL[u](x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |u(x+.) - u(y+.)|^2)(\theta)}{h^2}} u(y) dy$$

where $x \in \Omega$, $C(x) = \int_{\Omega} e^{-\frac{(G_a * |u(x+.) - u(z+.)|^2)(\theta)}{h^2}} dz$ is a normalizing constant [1] In this formula h serves as an adjustable parameter to change the filtering of the image. From the above formula it can be seen that this algorithm weights the pixels in the image such that a pixel is only modified by pixels situated in gaussian neighbourhoods similar to that of x . For the algorithm to be sensibly implemented the above formula must be reformulated into a discrete algorithm. Examples of these are discussed below.

Algorithm Implementations

Pixelwise Implementation: The first discrete implementation I will be discussing is the Pixelwise Implementation. This implementation is the most faithful to the original formula given for the algorithm since it calculates the denoised value for a pixel p one pixel at a time. The algorithm proposed in [2] is as follows. $u = (u_1, u_2, u_3)$ is a colour image, if a pixel p is selected its new value is calculated by first selecting pixels q in a large search area centered on p , this must be bounded though to ensure an acceptable runtime. Patches of a fixed size centered around p and q are then compared by taking the squared Euclidean distance between them, the shorter the distance the more similar they are. This distance is then used with an exponential kernel and a normalising constant to produce the weight associated with pixel q . Once this has been done for all pixels in the search space around p the new value for p is calculated as the weighted average of the pixels q . After this has been repeated for all pixels p in the image the new values for p can be combined to form the denoised image.

Patchwise Implementation The patchwise implementation denoises the image in patches rather than 1 pixel at a time. Its implementation according to [2] is as follows. The algorithm takes image $u = (u_1, u_2, u_3)$ and a patch $B(p, f)$ centered around p and of size $(2f + 1) \times (2f + 1)$. This patch is then compared to patches centered around pixels q , located in a large search area around p , in the same way as they are compared in the pixelwise implementation. The difference is that in the patchwise implementation the weighted average of these patch distances are used to calculate the new value of every p in the patch $B(p, f)$. This saves time as less patches are compared overall.

Comparison Pixelwise is the easier of the 2 implementations discussed in [2] to implement. However it is less efficient as it recalculates the weight of multiple patches each time they are used for a pixel. On the other hand patchwise calculates these weights and then uses them on all the pixels in a particular patch straight away rather than having to

calculate them again later. It is mentioned in [2] that another advantage of patchwise is the higher gain in peak signal-to-noise ratio and fewer noise oscillations at edges in the image. It is worth noting that although patchwise is more efficient it offers no improvement for preserving details in the image than pixelwise [2].

Effect of Parameters on Non-Local Means

Adjusting the value of h had a significant effect on the denoising of the image as can be seen below.



Fig 1. Top to bottom left to right: original, noisy, $h=3$, $h=5$, $h=7$, $h=15$

As can be seen as h is increased the denoising effect becomes more pronounced. However as h grows higher more and more features and details in the image are lost as a cost to gain greater denoising. Varying the size of the template window(tw) (size of the patch compared with the neighbourhood of p) reduces noise in similar parts of the image but does so at the cost of smoothing the edges between those parts of the image causing a loss in texture. This can be seen below



Fig 2. Top to bottom left to right: original, noisy, $tw=3$, $tw=7$, $tw=15$, $tw=21$

Varying the size of the search window around p had little to no effect on the result of denoising the image. With the exception of periodic images where a larger size gives much better results [6].

Comparison to other algorithms

When compared to other denoising algorithms Non-local means performs very favourably. A comparison is carried out in [3]. This comparison found that when compared to algorithms like Gaussian filter, Non-Local means maintains the structure of the original image far better keeping more details and features like hard edges.



Fig 3. Left to right: original, noisy, Non-Local Means, Gaussian Filter

The above shows a blurring of edges when Gaussian filter is used instead of Non-Local means as well as worse denoising in the flat zones of the image. The comparison [3] also found that a major strength of Non-Local means is its ability to denoise periodic images without losing the periodic pattern. The runtime for Non-Local means is a problem compared to other algorithms for denoising as found in [4] when it was compared to median filters. Non-Local means can also produce adhesion artefacts [5], this paper describes how if a pixel in a flat zone (area of similar valued pixels) is near a hard edge it will dominate the value of pixels on the other side of the edge creating an artifact. Non-Linear means tendency to over smooth textured parts of an image is also described in [5] although this weakness can be reduced by taking an average of the denoised image and the noisy image at the end of the process to restore texture [5].

Extensions to Non-Local Means

Gaussian Filter

References

1. Antoni Buades, and Jean-Michel Morel. *A non-local algorithm for image denoising*. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02 (CVPR '05). IEEE Computer Society, USA, 60–65.
2. Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. *Non-Local Means Denoising*. Image Processing Online.
3. Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. *On image denoising methods*. CMLA Preprint.
4. Subhojit Sarker, Shalini Chowdhury, Samanwita Laha and Debika Dey. *USE OF NON-LOCAL MEANS FILTER TO DENOISE IMAGE CORRUPTED BY SALT AND PEPPER NOISE*. Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.2, April 2012 .
5. Coll, Bartomeu & Morel, Jean-Michel. *Image and movie denoising by nonlocal means*. International Journal of Computer Vision - IJCV.

6. Salmon, J. *On Two Parameters for Denoising With Non-Local Means*. IEEE Signal Processing Letters
7. Sachin Chachada, Byung Tae Oh, Namgook Cho, San A. Phong, Daniel Manchala and C.-C.Jay Kuo. *Extension of Non-Local Means (NLM) Algorithm with Gaussian Filtering for Highly Noisy Images*. 2011 Visual Communications and Image Processing (VCIP), Tainan, 2011, pp. 1-4.