

Software Methodologies COMP2231 2019/2020 Image processing assignment

clvp22

January 13, 2020

1 D1

1.1 Introduction

The non-local means denoising algorithm is an effective algorithm for removing noise from a desired image. The purpose of denoising algorithms is to recover the original image from a noisy image; we can add gaussian white noise to an image, and then apply our denoising algorithm to evaluate its effectiveness by comparing our result to the original image. The wide class of denoising algorithms share the same basic principle, that denoising an image can be achieved by averaging; some algorithms perform averaging locally, by minimizing variation in the pixel neighbourhood, using calculus, or by frequency analysis and wavelet thresholding methods. The non-local means algorithm that will be described below also uses averaging. However a given pixel's estimated value is calculated by taking a mean of the values of all pixels that have similar gaussian neighbourhoods to that pixel, in other words the algorithm does not restrict itself to the neighbourhood of some given pixel, but instead uses the neighbourhoods of similar pixels elsewhere in the image.

1.2 NL-means algorithm

We are given some noisy image $v = \{v(p) | p \in P\}$ which consists of P pixels. Our algorithm works by evaluating every single pixel p in P in turn. For each p in P we calculate some value $NL[v](p)$ and we reconstruct our denoised image using these values as we go along. $NL[v](p)$, for a pixel p is computed as a weighted average of all the pixels in the image v , it is given by,

$$NL[v](p) = \sum_{q \in P} w(p, q) v(q),$$

where $w(p, q)$ is a list of weights denoting the similarity between the neighbourhoods of pixel p and q , and $0 \leq w(p, q) \leq 1$ and $\sum_q w(p, q) = 1$ hold. The similarity between the neighbourhoods of pixel p and q is calculated using intensity gray level vectors $v(N_p)$ and $v(N_q)$, these vectors represent a square neighbourhood of fixed size $|N|$ (typically 7×7) centered around pixel p and q respectively. The similarity between the two neighbourhoods is then calculated using the formula $\|v(N_p) - v(N_q)\|_{2, \alpha}^2$, where $\alpha > 0$ is the standard deviation of the gaussian kernel, and the formula itself is effectively the squared euclidean distance between the two vectors, with a gaussian kernel applied to it. The pixels with similar gaussian neighbourhoods to p have larger weights than those that don't, the weights are precisely defined as follows,

$$w(p, q) = \frac{1}{Z(p)} e^{-\frac{\|v(N_p) - v(N_q)\|_{2, \alpha}^2}{h^2}},$$

where $Z(p)$ normalizes the weights. And the parameter h is used to control the amount of filtering performed by the algorithm, as it controls the exponential function and by extension the weights function. The NL-means algorithm is special because it compares the geometry of the neighbourhoods at p and q not just the grey levels at p and q . This means if the two pixels p and q that have very similar grey levels, and neighbourhoods that are structurally different, we will get a small weight $w(p, q)$ that is very close to zero.

2 D2

2.1 Pixelwise implementation

An example of a pixelwise implementation for this algorithm is given by the non-local means [1,2] filter, which for a colour image $v = (v_1, v_2, v_3)$ and pixel p is given by,

$$NL[v_i](p) = \frac{1}{Z(p)} \sum_{q \in B_p} v_i(q) w(p, q),$$

$$Z(p) = \sum_{q \in B_p} w(p, q),$$

where v_i represents some colour channel $i = 1, 2, 3$, $Z(p)$ is our normalizing agent again, and B_p is a neighbourhood of fixed size $|B|$ (typically 21×21) centered around p . Notice how this implementation of the non-local means algorithm differs from the original description; our pixel p is not compared to every other pixel in the image, it is instead compared to every other pixel in B_p , a local fixed size neighbourhood around p . As a result this algorithm limits our research zone dramatically in the interests of computation time. Typically our research zone B_p for any given pixel p has a size of 21×21 , but for larger σ values, or in other words more gaussian noise, we use a 35×35 region. The weights $w(p, q)$ are calculated as before, as the squared euclidean distance between the two vectors N_p and N_q , which represent fixed sized neighbourhoods around pixels p and q respectively. For clarity, $|N|$ tends to be less than $|B|$, with N_p typically being a 7×7 square neighbourhood centered around p . Again,

$$w(p, q) = e^{-\frac{\|v(N_p) - v(N_q)\|_{2,\alpha}^2}{h^2}},$$

notice that the $Z(p)$ normalizing function is omitted for simplicity and has been moved outside the summation. After the procedure each pixel p and its colour channels are fixed with their new values, giving us a denoised image.

2.2 Patchwise implementation

The patchwise implementation differs from the pixelwise implementation, in that it denoises patches of pixels as opposed to a single pixel at a time. The algorithm denoises some colour image $v = (v_1, v_2, v_3)$ and a particular fixed size square region $B = B_p$ (centered at pixel p) as follows,

$$NL[v_i](B) = \frac{1}{Z} \sum_{N=N_q \in B_p} v_i(N) w(B, N), \text{ and } Z = \sum_{N=N_q \in B_p} w(B, N),$$

where again v_i represents some colour channel $i = 1, 2, 3$, Z is our normalizing agent, B_p is a neighbourhood of fixed size $|B|$ centered around p , and N_q is a neighbourhood of fixed sized $|N|$ centered around q ; where $|B|$ is typically greater than $|N|$ as before. Our weights function $w(N_p, N_q)$ is formulated the same as before. However, with this implementation we dispose of $|N|^2$ possible estimates for each pixel. These estimated can be averaged at the end for each pixel location to construct the resulting denoised image,

$$v_i(p) = \frac{1}{|N|^2} \sum_{N=N_q | q \in N_p} N_{i,p}.$$

The patchwise implementation improves on the pixelwise implementation, giving larger noise reduction. However, as a result we fail to preserve the fine details and the quality of the picture.

3 D3

3.1 Parameters

There are three parameters that we can discuss from the above implementations of the NL-means algorithm; $|N|$ the size of the comparison vectors, $|B|$ the size of the research zone around any given pixel, and h the parameter which controls the extent of filtering performed by the algorithm. The interesting thing is, that all of these parameters need to be adjusted for the value of σ , the standard deviation of the applied gaussian noise. As σ increases, the level of noise increases, so we need larger N vectors for more robust comparisons. We also need larger research neighbourhoods, or B regions, to increase the noise removal by finding more similar pixels. As a result we need to modify $h = k\sigma$ by decreasing k to deal with the larger comparison vectors. The ideal values as suggested in [2] can be found in Table 1 and Table 2 below.

| Grey | | | | Color | | | |
|---------------------|-------------|------------|--------------|---------------------|-------------|------------|--------------|
| σ | Comp. Patch | Res. Block | h | σ | Comp. Patch | Res. Block | h |
| $0 < \sigma < 15$ | 3 x 3 | 21 x 21 | 0.40σ | $0 < \sigma < 25$ | 3 x 3 | 21 x 21 | 0.55σ |
| $15 < \sigma < 30$ | 5 x 5 | 21 x 21 | 0.40σ | $25 < \sigma < 55$ | 5 x 5 | 35 x 35 | 0.40σ |
| $30 < \sigma < 45$ | 7 x 7 | 35 x 35 | 0.35σ | $55 < \sigma < 100$ | 7 x 7 | 35 x 35 | 0.35σ |
| $45 < \sigma < 75$ | 9 x 9 | 35 x 35 | 0.35σ | | | | |
| $75 < \sigma < 100$ | 11 x 11 | 35 x 35 | 0.30σ | | | | |

3.2 Examples

4 D4

4.1 Strengths and limitations

We can evaluate the NL-means algorithm in a mathematical context. The Conditional expectation theorem outlined in more detail in [1] tells us that the NL-means algorithm corrects a noisy image v instead of separating noise from the true image. Furthermore, [1] goes on to tell us that if we assume gaussian white noise has been added to the image, we can show NL-means minimizes the mean square error (euclidean distance between the original and restored image) and is theoretically optimal in this respect.

A more relevant way to evaluate the NL-means algorithm is experimentally. We can evaluate the effectiveness of it under three criteria, method noise, the visual quality of the denoised image and the mean square error. As before we will use the less computationally expensive non-local means [1,2] filter for our comparison with other denoising methods. We will use a comparison vector of size $|N| = 7 \times 7$ and a research neighbourhood of size $|B| = 21 \times 21$. Giving us a complexity of $49 \times 441 \times N^2$ for N^2 pixels.

Firstly, method noise works by evaluating how a denoising algorithm affects a non noisy image; Let v be an image and D_h be a denoising operator that depends on some parameter h , then the method noise is defined as follows,

$$v - D_h v.$$

Denoising algorithms should ideally not change a non noisy image, so the method noise should be small. If we visualise the method noise, a good performing algorithm should produce an image that just looks like noise with little to no structure. Figure 4 visualises method noise results from some common denoising algorithms,

We can see from Figure 4 that gauss filtering performs the worst, and imitates the laplacian edge detection filter. Anisotropic filtering and total variation perform slightly better. Neighbourhood filtering performs well and NL-means performs the best. This illustrates that NL-means adapts well to the geometry of the image, and the more naive algorithms don't.

The improved visual quality of a denoised image can only be judged by the human eye. Figure 5 shows us how some common denosing algorithms perform on the same image, with added gaussian white noise of standard deviation σ . The idea is to evaluate the results based on how well they restore the original image, their visual quality and the accuracy of the reconstructed edges, texture and details.

Notice how the blurred sections and structures of the restored images coincide with the structures in the method noise. NL-means performs well again, this is because with natural images we can find many similar pixels in flat areas, lines or edges. Lines and edges are clearly where the local denoising algorithms struggle. Neighbourhood filtering performs well again but doesn't appear to remove all the noise.

Our final criteria, mean square error can be calculated mathematically and visualised in a table. The mean square error is simply the euclidean distance between the original image and the restored image after some noise has been added to it. Tab 3 illustrates the mean square error values calculated from the results of our denosing algorithms.

| | dice.png | | | | |
|-------|-----------------|-----------------------|-----------------|-------------------------|-----------------|
| | Gauss filtering | Anisotropic filtering | Total variation | Neighbourhood filtering | non-local means |
| Error | 120 | 114 | 110 | 129 | 68 |

A small mean square error suggests our denoising algorithms is effective, because it tells us our restored image is close to the original image. It becomes fairly obvious non-local means performs the best in restoring the original image. The mean square error is definitely the most objective of our criteria, but it tells us nothing about the actual quality of the restored image, so the other criteria are certainly needed.

However, the limitations that come with the non-local means algorithm are still relevant. The main issue with the non-local means algorithm is that it is computationally expensive in comparison to the other denoising algorithms. In our actual implementation we fail to realise the original description of the non-local means algorithm, where the neighbourhood of every pixel is compared to the neighbourhood of every other pixel in the image. This means that the previous statements about optimality and the points made in [1] do not necessarily hold for our implementation.

5 D5

5.1 Modifications and extensions

6 D6

6.1 Applications

References

- [1] Antoni Buades, Bartomeu Coll, and J.M. Morel. "A non-local algorithm for image denoising", *IEEE Computer Vision and Pattern Recognition 2005, Vol 2*, <http://dx.doi.org/10.1109/CVPR.2005.38>, 2005.
- [2] Antoni Buades, Bartomeu Coll, and J.M. Morel. "Non-Local Means Denoising", *Image Processing On Line*, http://dx.doi.org/10.5201/ipol.2011.bcm_nlm, 2011.