Universitat De Girona

Medical Image Segmentation and its Applications

Image Pre-processing

Students:
Ali Berrada
Ama Katseena Yawson

Supervisor:
Robert Martí

October 28, 2018

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Image pre-processing is a essential step in computer vision and image processing. It can be defined as an enhancement of an image data with the aim of suppressing or removing unwanted features for further processing. In recent times, this area of image processing has extended to various fields of science and technology [1]. Examples of such transforms include filtering, colour transforms, scaling, histogram generation and many others. Since the primary objective of the image pre-processing tool is to remove unwanted distortions, a prior knowledge about the image is needed. The prior knowledge can be conditions under which the image was obtained, the imaging modality used and the type of distortion present. For example, in an ultrasound imaging modality the common distortion present is the speckle noise and hence before processing such images, it is important to tackle this problem with an image pre-processing techniques. In this lab report, we analyzed the effect of the parameters used in bias field correction and anisotropic diffusion. Each of this pre-processing techniques are explained into details in the next sections.

2 Bias Field

According to Juntu and et al [2], bias field can be defined as a low frequency signal which produces unwanted signals in MRI images due intensity inhomogeneities of the MRI machine. The visual effect of this distortion is that it blurs the image and minimize the high frequency details of the image such as edges, contours and variations in the intensity values of image pixels as shown in Figure 1. Another effect of this inhomogeneity is that it degrades segmentation and classification algorithms and hence bias field correction is needed.

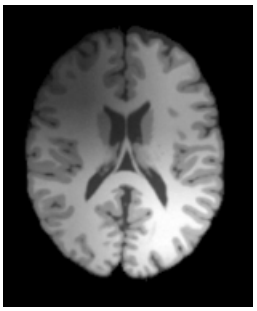

Figure 1: A brain image corrupted with bias field

The correction is done by first estimating the bias field present in the image and then dividing the bias corrupted image by the estimated bias field. There exist two forms of bias field correction techniques which includes the prospective and retrospective methods. The main difference between these two techniques is that, the prospective method try to solve the inhomogeneities in the acquisition procedure while the retrospective does the same in the data after the acquisition procedure. Also the prospective method is unable to account for inhomogeneities caused by the patient during acquisition and hence the widely used method is the retrospective which has the ability to account for both cases. In this lab work, the algorithm we used in removing the bias field from the provided images was the Multiplicative intrinsic component optimization (Mico) algorithm. A summary of how the algorithm operates is explained below.

2.1 Mico Algorithm

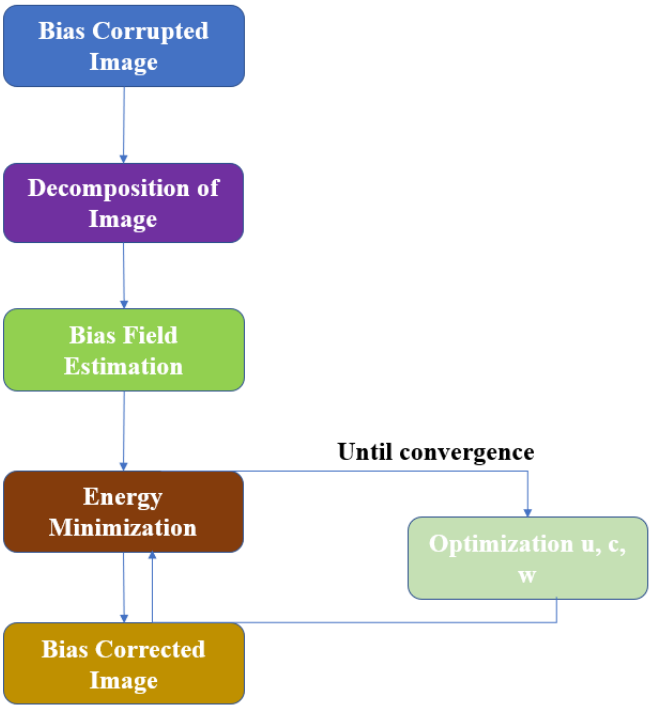

Figure 2: The structure of the Mico algorithm

Based on the algorithm of Li and et al [3], Mico is an energy minimization algorithm which performs both bias field estimation and segmentation. The equation used in

implementing this algorithm is shown in Equation (1). The algorithm focuses on the multiplicative part of equation which includes the true image and bias field.

$$I(x) = b(x)J(x) + n(x) \tag{1}$$

where $b(x)$ is the bias field, $J(x)$ is the true image which is approximately equal to a constant c for all points in the same tissue and $n(x)$ is an additive noise with zero means.

In their implementation, a prior knowledge of the bias field and the true image was required to make the problem solvable. The physical properties for the bias field and the true image were the smoothly varying property of the bias field b and the piecewise constant property of the true image J respectively. To ensure the correct usage of the physical properties, appropriate mathematical representations and descriptions were used. The bias field was given as a linear combination of a set of smooth basis functions $g1, \cdots, gM$(M = Number of basis function) which maintains the smoothly varying property of the bias field. The bias field was estimated using Equation (2).

$$b(x) = w^T G(x). \tag{2}$$

where w^T is a column vector of the optimal weight coefficient and $G(x)$ is the column vector of the basis function.

The above equation is used in the energy minimization step to estimate the bias field. Using the physical property of the true image, the piecewise constant property of the true image J is determined using the concept of membership function. This involves a prior knowledge about the number of regions in the image. To allow for soft segmentation, the fuzzy membership function u was used. Conversely to the membership function which takes either 0 or 1, this takes values between 0 and 1 which implies that the same pixels can be assigned to more than one region. The true image can then be estimated using Equation (3).

$$J(x) = \sum_{i}^{N} c_i u_i(x) \tag{3}$$

where N is the number of regions to be segmented in the image, u is the fuzzy membership and c is the constant that characterize each region. The output of the segmentation part is given the column vector of the fuzzy membership function. The bias field corrected image is found by dividing the original image by the bias field as shown in the Equation(4) below.

$$bc = I/b \tag{4}$$

where bc is the bias corrected image.

The proceeding step is the optimization of the energy function until a certain number of threshold is reached. The optimization of the b and J was done by reducing the energy function F as shown in Equation (5) with respect to u, c and w.

$$F(u, c, w) = \int_{\Omega} \sum_{i=1}^{N} \left| I(x) - w^T G(x) c_i \right|^2 u_i^q(x) dx \tag{5}$$

where q is a fuzzier which helps to achieve soft segmentation of fuzzy segmentation.

The summary of the Mico algorithm is that it takes as an input bias corrupted image and decomposes it into bias field and true image. After the decomposition stage, the bias field is estimated by optimizing parameters of the energy function such u , c and b until there is convergence as shown in Figure 2.

2.2 Implementation

The implementation was made using Matlab R2018b. The brain data given were in the nii format. The braindata **t1_icbm_normal_1mm_pn0_rf20.nii** and **t1_icbm_normal_1mm_pn0_rf40.nii** were bias corrupted 3D volume. Using the NIFTI tools, we loaded the image using the load_nii function. After this step, we read all the 2D images from the 3D volume iteratively and avoided empty images. Some modification were made in the original code to obtain the desired results. We then analyzed the effect of the parameters such as q (fuzzier) and $iterNum$ (number of iteration) on the bias field correction both quantitatively and qualitatively. The value of $q = 2$ and $iterNum = 30$ was kept constant in all the different experiments made. For the qualitative evaluation, we adopted the visual inspection technique by comparing the original image with bias corrected image. In the quantitative evaluation, we used the sum of squared difference metrics to measure the similarity metric after bias field correction. The ssd metric was saved in a text file called Quantitative Evaluation_rf20.txt and Quantitative Evaluation_rf40.txt. All the 2D images before and after bias field correction were saved in a folder called OriginalImages and BiasCorrectedImages respectively. Based on the output obtained from the 2D images, the same parameters were used as input parameters for the 3D volume. In the Mico 3D implementation, we modified the code in order to visualize the bias corrected images.

2.3 Results and Discussion

2.3.1 2D Results

After applying the Mico algorithm on brain 3D volume, the results obtained for some of images are displayed and discussed in this section.

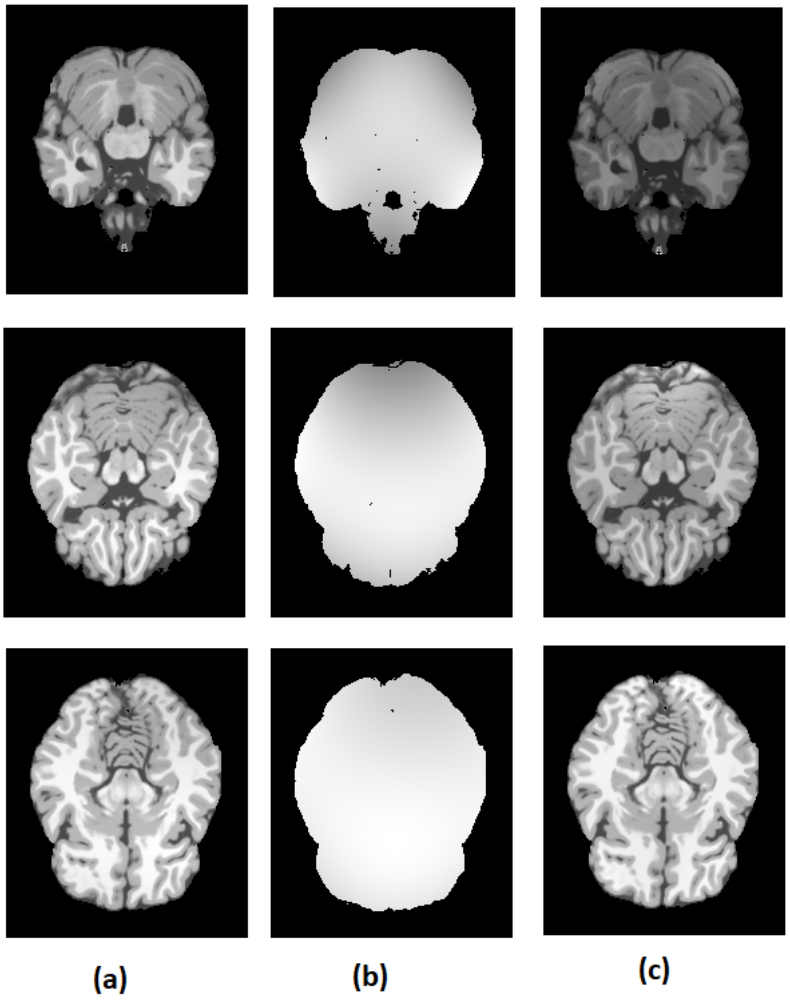

Figure 3: Results showing the Original images, Bias fields and Bias corrected images

In Figure 3, the first column with the label **(a)** represents the original images with bias field, the second column with the label **(b)** represent the corresponding bias field and the last column with the label **(c)** represents the corresponding bias free images. The images shown in Figure 3**(a)** are 2D images taken from the 3D volume. Based on the qualitative evaluation of this set of images (Figure 3), it is evident from the bias

field that there are variations in intensity values as some intensities are increasing towards the middle and sides. To correct this intensity inhomogeneities, Equation (4) was used in correcting the original image to obtain homogeneous intensity in the image as shown Figure 3**(c)**. By comparing the original images in column **(a)** and **(c)** of Figure 3, it can be seen that the images in column **(c)** have equal illumination when compared with that of the column **(a)**. Hence from the qualitative point of view, we can say that the bias field associated with these 2D images were corrected because of the homogeneous illumination visualized after the correction. The quantitative results measured for the images shown in Figure 3 are shown in Table 1 below.

Image Label	**SSD**
1(a)	0.015225
2(a)	0.005868
3(a)	0.000744

Table 1: Quantitative Evaluation of Images shown in Figure 3.

For the quantitative evaluation, we measured the similarity metrics between the bias corrupted images and bias corrected images. The similarity metric used in this case was the sum of squared difference (SSD) using Equation (6).

$$SSD(A,B) = \frac{1}{N}\sqrt{\sum_{x=0}^{X}\sum_{y=0}^{Y}(A(x,y) - B(x,y))^2} \tag{6}$$

where $SSD(A,B)$ is the sum of squared difference measured between the two images, N is the number of elements in the original image, $A(x,y)$ is image $A's$ intensity value at location (x,y), $B(x,y)$ is image $B's$ intensity value at location (x,y).

Based on the results obtained as shown in Table 1, there are some difference between the original images and the bias corrected images. From the SSD results, image with label 1(a) has a higher value relative to 1(b) and 1(c). This implies that 1(a) was more corrupted by the bias field than the others.

2.3.2 3D Results

After experimenting with the different parameters in the 2D implementation, we transposed the parameter values to the 3D algorithm. The code generates 2 NifTi volumes, one for the result of bias correction and the other for the segmentation result. The code also displays for the user a single 2D slice along with its segmentation

result. Some simple additions were made to be able to display also the computed bias field of the same slice and save the whole bias field volume as NifTi file 4. Sample outputs are shown in Figure 5 for "t1_icbm_normal_1mm_pn0_rf40.nii" and in Figure 6 for "t1_icbm_normal_1mm_pn0_rf20.nii".

```
100 -      figure;
101 -      subplot(1,3,1), imshow(Img3D(:,:,N_slc)',[]);
102 -      title("Image");
103 -      subplot(1,3,2), imshow((b(:,:,N_slc).*ROI(:,:,N_slc))',[]);
104 -      title("Bias field");
105 -      subplot(1,3,3), imshow(img_bc(x1:x2,y1:y2,N_slc+z1)',[]);
106 -      title("Bias corrected");
107 -      b=make_nii(b);
108 -      save_nii(b,[str,'_bf.nii']);  % save bias field
```

Figure 4: Code snippet for displaying the 2D bias field of a sample and saving the bias field volume into file.

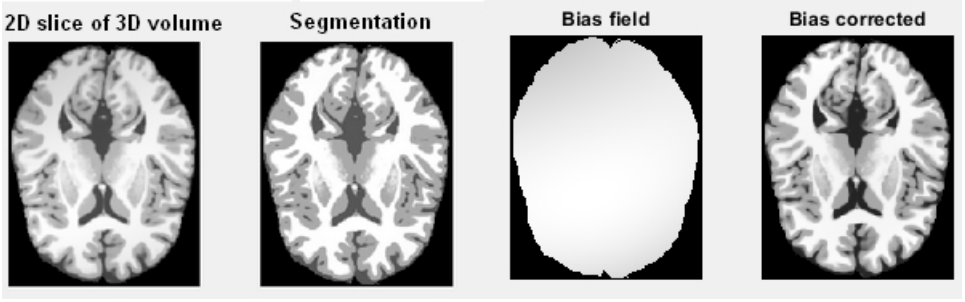

Figure 5: Results showing the Original images, Bias fields and Bias corrected images of a 2D slice of the 3D volume processed by the MICO_3D code

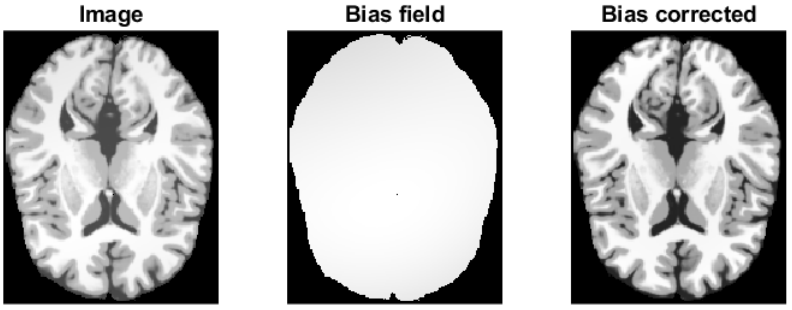

Figure 6: Results showing the Original images, Bias fields and Bias corrected images using the 3D volume

3 Noise suppression

Noise can be defined as random changes of image intensity and are usually seen as grains on the image. There are several types of noise which require different noise removal techniques. In this section we discuss about the anisotropic and isotropic noise removal techniques.

3.1 Anisotropic Diffusion

Anisotropic diffusion is an edge-preserving image denoising method. The idea is to apply smoothing to an image but reduce the effect of the smoothing when near an edge, following an adaptive coefficient called the diffusion coefficient. The anisotropic diffusion equation is defined in (7). Perona and Malik pioneered the idea of anisotropic diffusion and proposed 2 diffusion equations: the first called Leclerc favours high-contrast edge lines over low-contrast ones (9) and the second is called Lorentz favours large regions over smaller ones (10).

$$I_t(x, y, t) = \Delta(c(x, y, t)\Delta I(x, y, t)) \tag{7}$$

where $I_t(x, y, t)$ is the anisotropic diffusion, $c(x, y, t)$ is gradient descriptor and $\Delta I(x, y, t)$ is the gradient of the image.

$$c(x, y, t) = g(|\Delta I(x, y, t)|) \tag{8}$$

$$g(\Delta I) = e^{\frac{-|\Delta I|^2}{k^2}} \tag{9}$$

where k is kappa.

$$g(\Delta I) = e^{\frac{1}{1+\frac{|\Delta I|^2}{k^2}}} \tag{10}$$

3.2 Results

To evaluate the effect of changing parameters on the smoothing result, a combination of different iteration numbers, diffusion constants (kappa) and diffusion equations were tested, and the results obtained in each case are displayed and commented below.

3.2.1 Low-noise Image

Figure 7 shows a single slice from a volume image with little noise along with the blurring effect of using a circular averaging filter of radius 3. We can notice the overall smoothness in the blurred image, but this came at the cost of losing to some extent high-frequency details such as the original image's sharp edges which might be of great importance for analysis or segmentation.

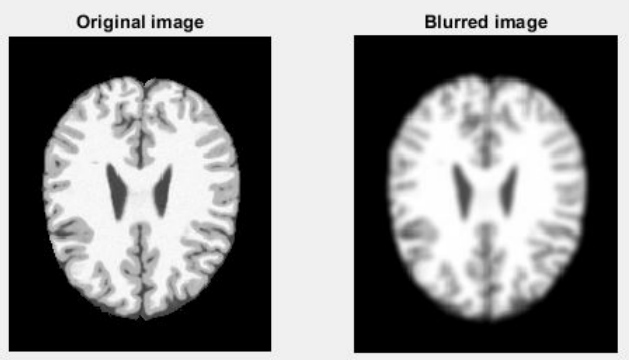

Figure 7: Low-noise image with convolution with a circular filter

Figure 8 and Figure 9 respectively show the results obtained using (Equation 9) and (Equation 10) experimenting with different kappa values (10, 20, 50 and 100) and 2 iteration sizes (50 and 100). In Figure 8, it is not easy to visually spot the differences when the diffusion constant value is 50 or below; however, subtle details can be noticed such as small edge shapes shrinking (or blurring, practically speaking) when the kappa value is 50 compared to them being relatively well preserved at lower kappa values. These differences explain how the dissimilarity error (i.e. how different are the original image and its denoised image) is larger when the kappa increases. From the error also, we know that by fixing the kappa and changing the number of iterations there is more diffusion happening but not so substantially. Finally, as the Equation 1 theoretically promised to prioritize the conserving of edges (and by that the regions delimited by those edges) with high contrast over low contrast, we can see how, when the kappa value was 100, the CSF at the center is well defined while the grey and white matters are averaging, and their separating lines became blurred. If that specific region is of interest for segmentation, for instance, then the anisotropic diffusion using the first diffusion equation and a large value of the diffusion constant may be a very useful pre-processing step. In Figure 3, where the second equation was used, the wider area which is the white matter is having more influence or diffusion over the smaller CSF. The contour of the CSF regions are already blurred with a kappa value of 50. Meanwhile, with a kappa value of 20 we

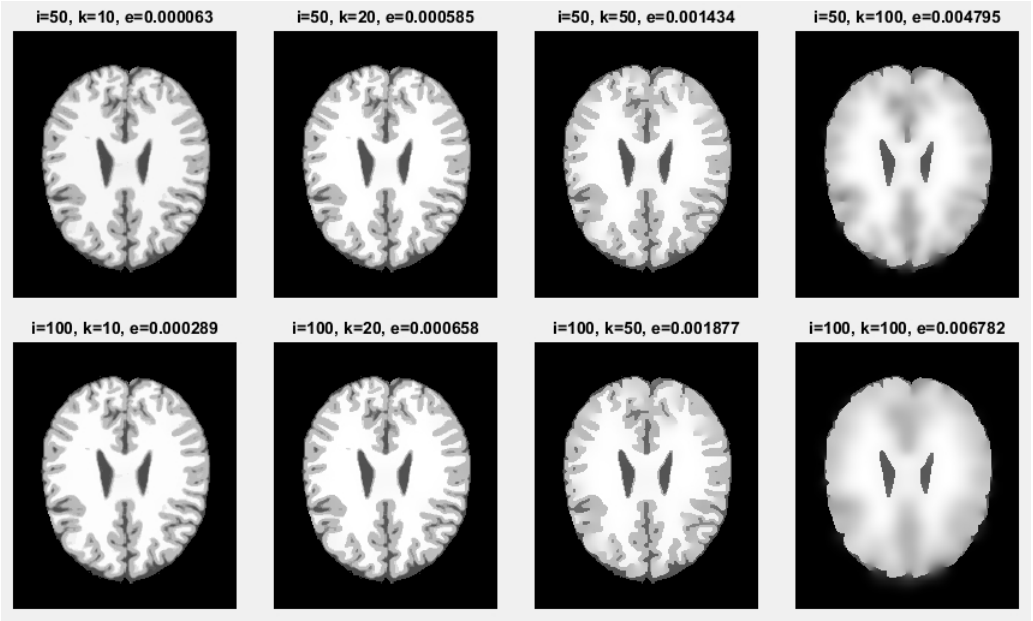

Figure 8: Leclerc gradient descriptor for low noise images

can see that the grey matter is winning over the white matter vessels near the skull as they are surrounded by the foremost.

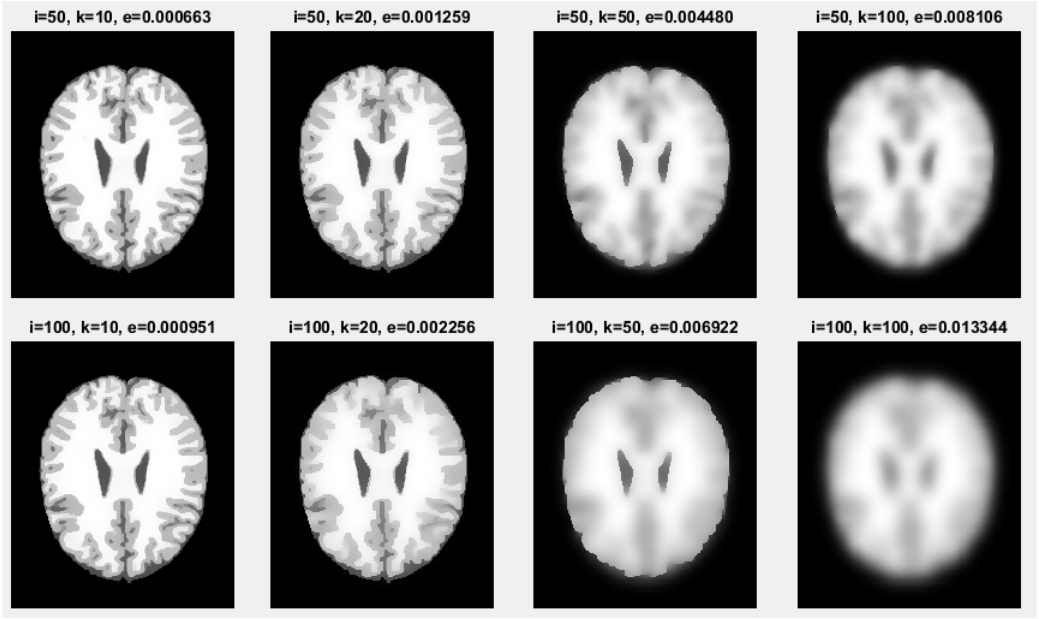

Figure 9: Lorentz gradient descriptor for low noise images

3.2.2 High-noise image

Figure 10 shows a single slice from a volume image affected by a relatively higher noise than in the previous case, along with the blurring effect of using a circular averaging filter of radius 3.

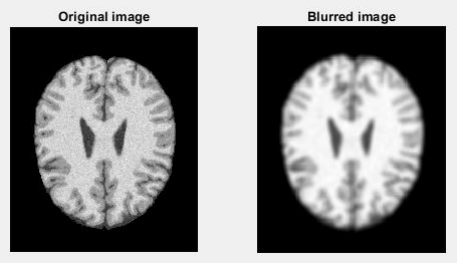

Figure 10: High-noise image with its convolution with a circular filter

Figure 11 shows the results obtained with Equation (9). We notice that a kappa of 50 does a good job in smoothing regions while keeping them distinct, however some small but subtle details are lost such as certain parts the white matter near the skull. A kappa value of 20 does relatively less on the smoothing but keeps more details. Also, with the same kappa value but double the number of iterations, the result provides better smoothing. Therefore, we tested with a higher number of iterations and increased the kappa to 30 and the result qualitatively looks more balanced (Figure 12).

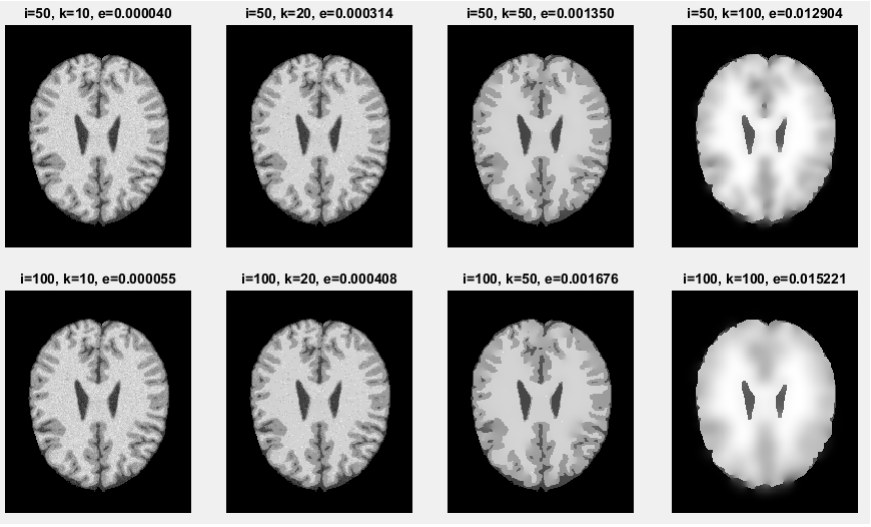

Figure 11: Leclerc Gradient descriptor for High noise images

Figure 12: Better qualitative result

Figure 13 shows the result obtained with the Lorentz diffusion equation. This equation worked best for this case since the diffusion is not affected by the noise pixels within the regions (the regions influence the noisy pixels – and not the opposite – since the latter are sparse). By using a small diffusion constant like 10, the algorithm preserves the edge lines as well.

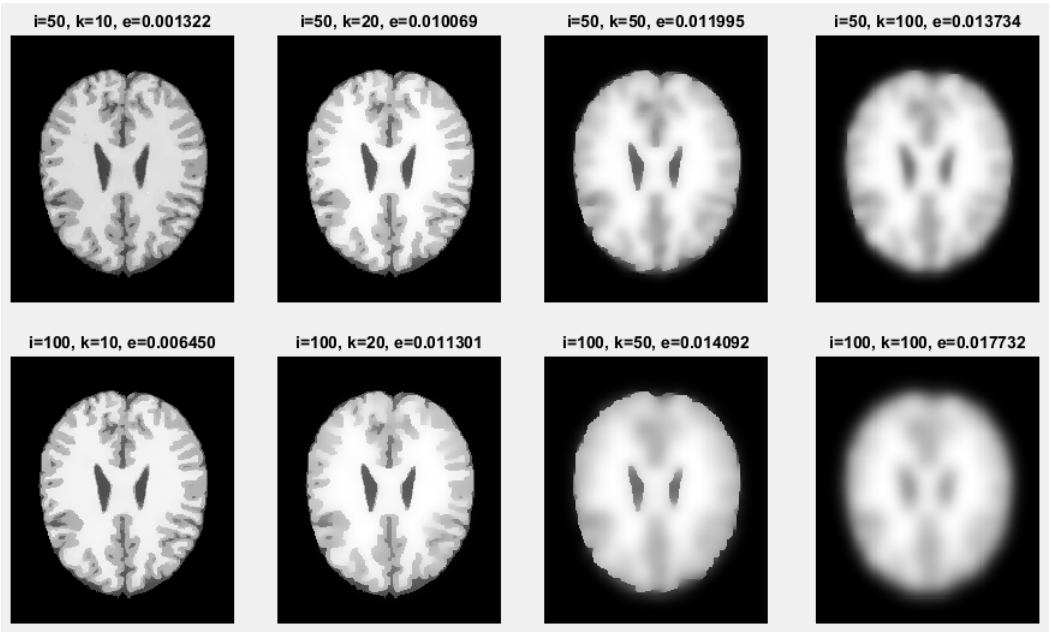

Figure 13: Lorentz gradient descriptor for High noise images

3.3 Isotropic Diffusion

With the anisotropic diffusion, the diffusion at each direction is done differently following a different control coefficient for each direction. If these coefficients are uniform, then the diffusion is isotropic. In other words, the smoothing in the isotropic approach does not depend of direction or the characteristics of the image content, and therefore the technique is not edge-preserving. The mean and Gaussian blurs are 2 popular examples applying isotropic diffusion since their filters operate the same way on every pixel. Implementation-wise, the diffusion coefficient component *c(x, y, t)* in (7) (which in the case of anisotropic diffusion is computed for the 4 directions relative to the 4 directional gradients) is set to a constant; this small change results in the isotropic diffusion shown in Figure 14 and Figure 15 which respectively correspond to the low-noise and high-noise images.

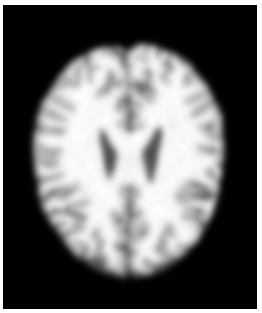

Figure 14: Isotropic diffusion from brain data **pn1_rf0.nii**

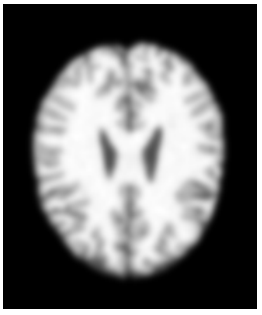

Figure 15: Isotropic diffusion from brain data **pn5_rf0.nii**

4 Conclusion

Image pre-processing is a essential step in computer vision and image processing. In this exercise, we studied the various effect of some image pre-processing techniques. We investigated the effect of the parameters for both bias field correction and anisotropic diffusion using a 3D volume brain data. For the bias field, we noticed some changes in illumination after the bias fields were removed for the qualitative analysis. In the quantitative analysis, we measured the similarity metric using the SSD. Regarding the anisotropic diffusion, we compared the obtained output to that of gaussian blurring and the isotropic.

References

[1] P.Srimathi B. Chitradevi. An overview on image processing techniques. http://www.rroij.com/open-access/an-overview-on-image-processing-techniques.php?aid=47175. Accessed: 2018-10-25.

[2] Jaber Juntu, Jan Sijbers, Dirk Van Dyck, and Jan Gielen. Bias field correction for mri images. In *Computer Recognition Systems*, pages 543–551. Springer, 2005.

[3] Chunming Li, John C Gore, and Christos Davatzikos. Multiplicative intrinsic component optimization (mico) for mri bias field estimation and tissue segmentation. *Magnetic resonance imaging*, 32(7):913–923, 2014.