3D Mumford-Shah segmentation of mice cerebellum area

Maïtine Bergounioux, Laurent Delsol

3D Mumford-Shah segmentation of mice cerebellum area

Maïtine Bergounioux * Sandra Même† Frédéric Szeremeta†
Laurent Delsol* JeanClaude Belœil †

July 3, 2013

Abstract

The aim of this paper is to present a complete methodology to perform the segmentation of a stack of MRI images. More precisely, we deal with (3D) of MRI images of mice brain. We first develop an automatic selection of the cerbellum area. Unadapted segmentation methods are mentionned. At last, we decided to use a second order decomposition method as a processing tool to perform denoising that does not erode contours.We end the segmentation a 3D Mumford-Shah method and present some results on the cerebellum volume estimates.

1 Introduction

The aim of this paper is to present a complete methodology to perform the segmentation of a stack of MRI images. More precisely, we deal with (3D) of MRI images of mice brain. Magnetic Resonance Imaging (MRI) is a non invasive imaging technique which can be used for diagnostic or longitudinal therapeutic purposes. Proton MRI is well adapted to study soft material such as cerebral tissue in animal models of human neuropathologies. However MRI studies on small rodents are challenging due to the small sizes of their brains. The recent use of high fields for MRI instruments enables the increase of the spatial resolution, the improvement of the sensitivity of the technique and to optimize the signal-to-noise ratio. Nevertheless, the increase in the magnetic field increases the sensitivity to the effects of magnetic susceptibility and reduces the natural contrast between different tissues. Thus, one of the research objectives in MRI today is to propose new methodological, technological and instrumental developments to improve the contrast, sharpness, speed and spatial resolution at high field. Manganese has paramagnetic properties leading to an enhancement of the MRI signal. In this work, MR experiments were performed using manganese as contrast agent in order to identify the consequences of the aneuploidy associated with human chromosome 21 in mouse models on the development of the central nervous system. Some mice are trisomic and some are healthy. The goal of the segmentation process is

*Université d'Orléans, Labo. MAPMO, CNRS, UMR 6628, Fédération Denis Poisson, FR 2964, Bat. Math., BP 6759, 45067 Orléans cedex 2, France , maitine.bergounioux@univ-orleans.fr, laurent.delsol@univ-orleans.fr

†CNRS Orléans -Laboratoire CBM, UPR 4301, CNRS Rue Charles Sadron, 45071 Cedex 2 Orléans, France - sandra.meme@cnrs-orleans.fr,frederic.szeremeta@cnrs-orleans.fr, jean-claude.beloeil@cnrs-orleans.fr.

to estimate the volume of the cerebellum without killing the animal. A previous work has been done in [2] but authors were concerned with the brain segmentation which is a different issue. Indeed, brain contours are sharper and usual segmentation techniques as a split and merge method for example are relevant. The conclusion of the quoted paper was that there is no difference between of the brain volumes of trisomic and non trisomic mice. Therefore, we focus on the cerebellum : the challenge is higher since the 3D cerebellum stack is about $76 \times 50 \times 48$ (the original whole head size stack was $341 \times 110 \times 110$. As often in MRI, images are awfully undersampled. In addition, contours are not sharp any longer so that we have to look for dedicated methods.

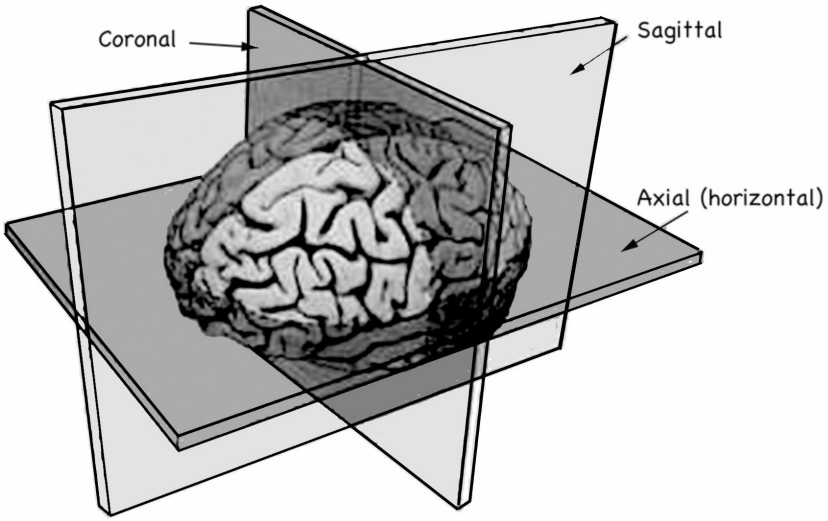


Figure 1: 3D stack different views [11]

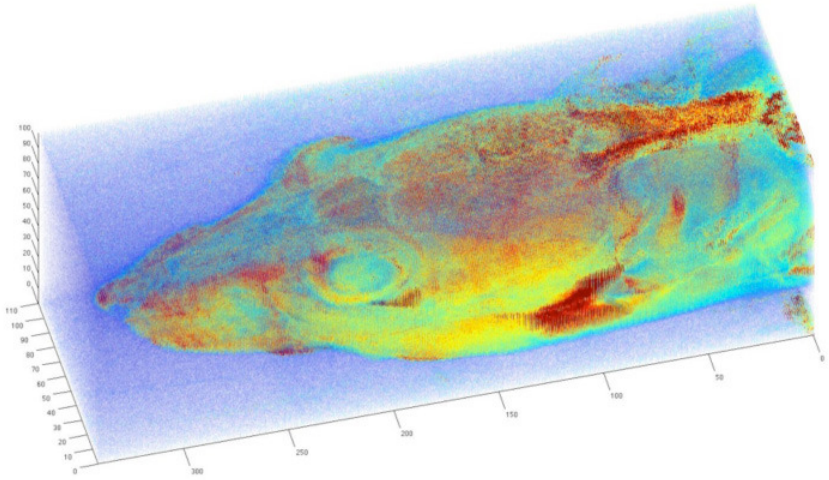


Figure 2: 3D view

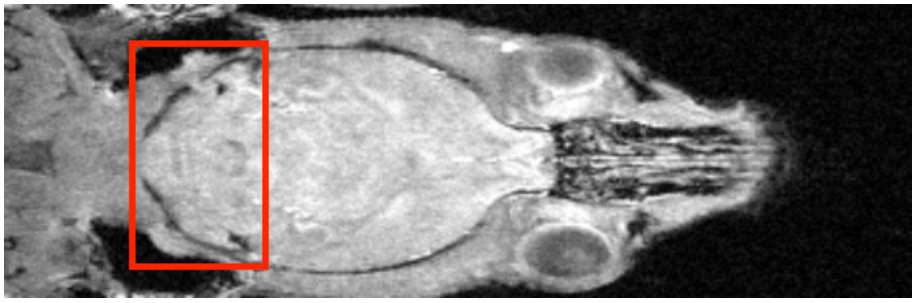


Figure 3: Axial view - slice 70

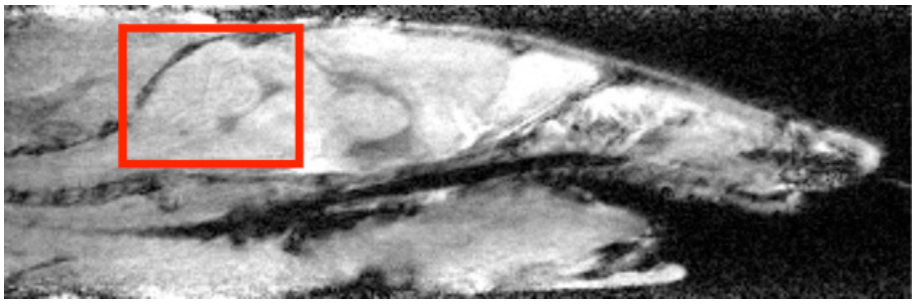


Figure 4: Sagittal view - slice 53

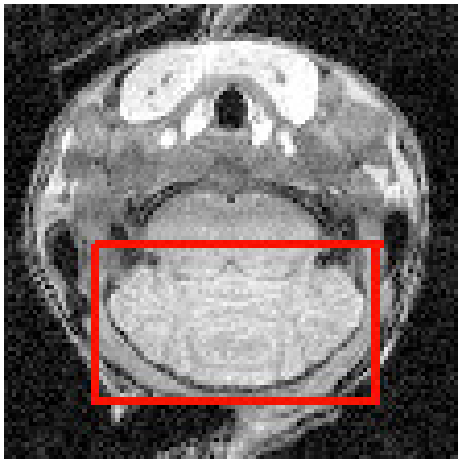


Figure 5: Coronal view - slice 103

The paper is organized as follows: we first describe an automatic procedure to extract the cerebellum area from the mouse head image : more precisely we define a 3D smaller region of interest. Then we present methods that do not work and the method we finally decided to use. Last section is devoted to the analysis of numerical results.

2 Preprocessing: automatic selection of the cerebellum area

We have many stacks to consider (here 14 mice) and we would like to perform an automatic selection of the cerebellum area. In what follows we describe the generic methodology.

We decide to perform the same analysis with respect to sagittal, coronal and axial views. We present the method on the axial (top) views. The goal is to create a 2D thumbnail that will contains all the 2D images of the stack.

- We first extract the wider slice which is here the number 70. We may allow close slices since we shall add some "security" pixels around the area.
- The (15) slices number 70 are contrasted and histograms are equalized.
- Then we perform a denoising process: we have used a variational filter based on the total variation that perform a good denoising while preserving contours: more precisely we have to compute the solution to

$$\min \sum_{i,j} (\bar{u}_{i,j} - u_{i,j})^2 + \lambda \sum_{i,j} |\nabla u_{i,j}|, u \in \mathbb{R}^N \times \mathbb{R}^M \ ,$$

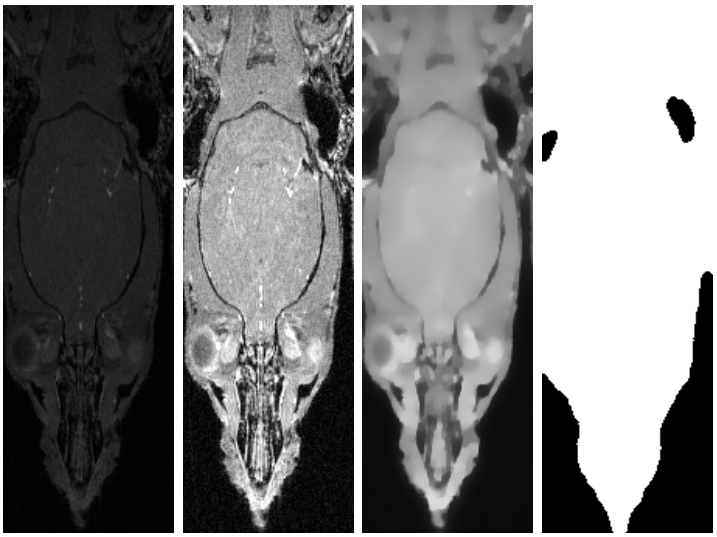
where $\bar{u}$ is the noisy image $\nabla u_{i,j} = (\nabla_1 u_{i,j}, \nabla_2 u_{i,j})$ is a discrete gradient and

$$|\nabla u_{i,j}| = \sqrt{(\nabla_1 u_{i,j})^2 + (\nabla_2 u_{i,j})^2} \ .$$

For more details, one can refer to [1, 13, 14, 8]. We have used it with $\lambda = 50$.

- Then we use an EM (Expectation-maximisation) algorithm to identify the gaussian parts of the histograms of filtered images. We recover two gaussian parts whose parameters are the mean value $\mu_1 \leqslant \mu_2$ and the standard deviation σ_i for $i = 1, 2$.
- We perform a thresholding with $\mu_1 + 2\sigma_1$ as a threshold.
- We count the white pixels line by line: the first line where this number rapidly change sis the beginning of the mouse muzzle.

Next figure illustrates the process



(a) Original (b) Equalized (c) Denoised - $\lambda = 50$ (d) Thresholded

Figure 6: Axial slice number 70

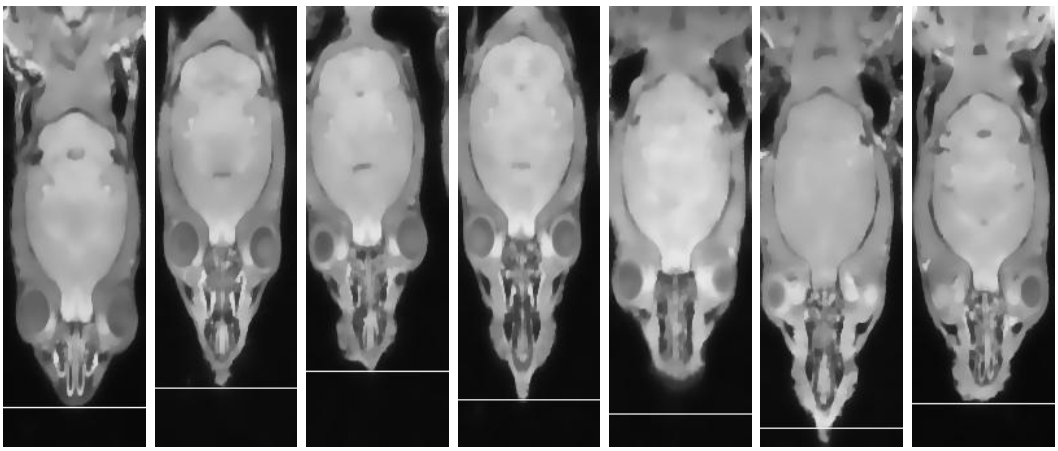
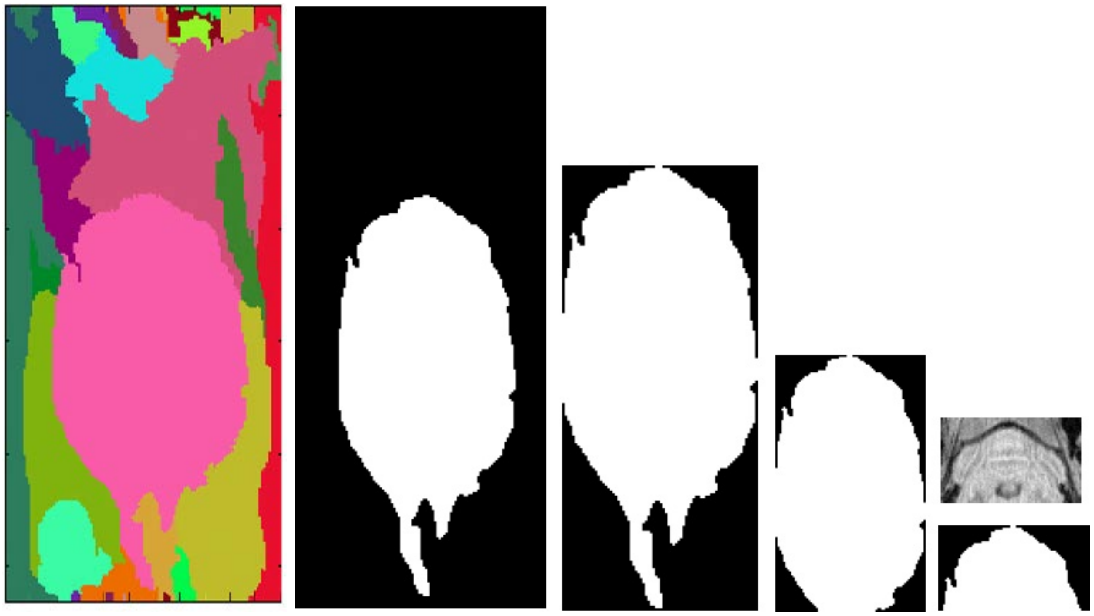


Figure 7: Slice 70 - automatic detection of the muzzle

Then, we remove the muzzle and use a split and merge segmentation method to isolate the brain area. Finally, anatomical information allows to extract the cerebellum area (first 40% pixels from the top):



(a) Brain location with split and merge

(b) Extraction process after thresholding

Figure 8: Different steps for ROI extraction

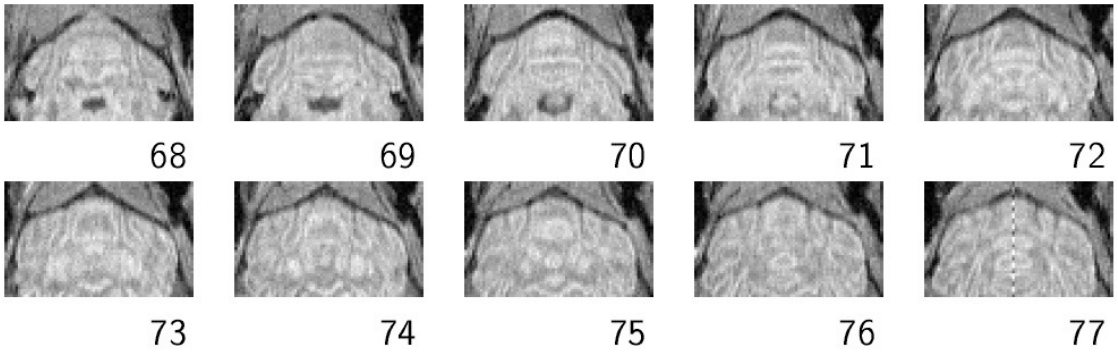


Figure 9: Cerebellum area for slices close to 70

We perform the same work for the two other views (with slice 53 for the sagittal view and slice 103 for the coronal view). We finally adjust the three 2D thumbnails we obtain to get a 3D- box that contains the cerebellum for any stack. Note that we have validated the results with manual selection of the box.

3 Segmentation

Once we have isolated the 3D region of interest we perform a segmentation to find the cerebellum contours. Once we get them, we may compute the volume by counting the voxels inside. We tried first 2D segmentation slice by slice and counted the pixels to recover the 3D volume. However, this method is slow and depends on the view we consider. There are many errors and the results are not very good.

So we decided to perform a 3D segmentation. We have tried may techniques during the 2D segmentation. We briefly report on techniques that are not satisfactory.

3.1 2D segmentation : methods that do not work

As the images are very poor sampled the classical techniques (gradient, Laplacian, high pass filter) do not work and we have to look for more sophisticated methods. We first tried the use of active contours (snake) methods as in [2]. A basic Fast-Marching Method (FFM) to solve the eikonal equation is not satisfying and we have used a method by Forcadel et al [10]. They have proposed a generalized FFM that deal with propagation speed with non constant sign. This method includes Chan-Vese [9] model. Unfortunately, results were not satisfying : the method is too sensitive to the parameters and to the initial contour.

Therefore, we have tried to use a active contour method with Gradient Vector Flow as in [15, 16, 7]. However, though the results seem corrects theye are quite dependent on the initial guess. Moreover, this method is quite sensitive to the choice of parameters and it is impossible to get a robust stopping criterion. The results we present where stopped manually. The itinial contour is chosen as many small circles.

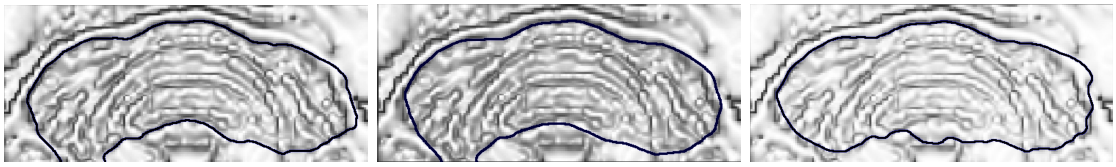
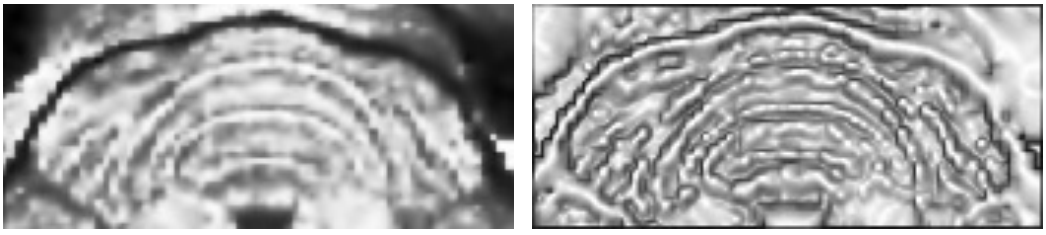


Figure 10: GVF contours for different parameters

This last technique has not been tested on the original one that involved too many contours. We used the Mumford-Shah model (see next section) as a preprocessing tool.



(a) Thumbnail (b) Preprocessing

Figure 11: Preprocessing with Mumford-Shah method

3.2 Second order denoising process

First we have to perform denoising.We have to choose a method that preserves the contours. So we decided to use a second order variational model that provides a filtered image whose contours are not degraded and no additional contour (staircasing effect) appears [5, 6]. We briefly recall the method. We use a classical finite difference scheme to compute the gradient, the divergence and the second order derivative of images. In what follows the image size is $N_1 \times N_2 \times N_3$. The generic component of $u \in X := \mathbb{R}^{N_1 \times N_2 \times N_3}$ is $u_{i,j,k}$. The formulas we used to compute these quantites are given in appendix. We consider the following functional $F : X \to \mathbb{R}^+$ defined as

$$F(v) = \frac{1}{2} \sum_{i,j,k} (\bar{u}_{i,j,k} - v_{i,j,k})^2 + \lambda TV^2(v)$$

where $\bar{u} \in Y$ is the image to be denoised, $\lambda > 0$,

$$TV^2(v) := \sum_{i,j,k} \|(Hv)_{i,j,k}\|_{\mathbb{R}^9} \ ,$$

and for every $W = (W_i)_{1 \leqslant i \leqslant 9} \in \mathbb{R}^9$

$$\|W\|_{\mathbb{R}^9} := \sqrt{\sum_{i=1}^{9} (W_i)^2} \ .$$

We look for a solution to the optimization problem:

$$\inf_{v \in X} F(v) \tag{3.1}$$

We us the following algorithm as [5]:

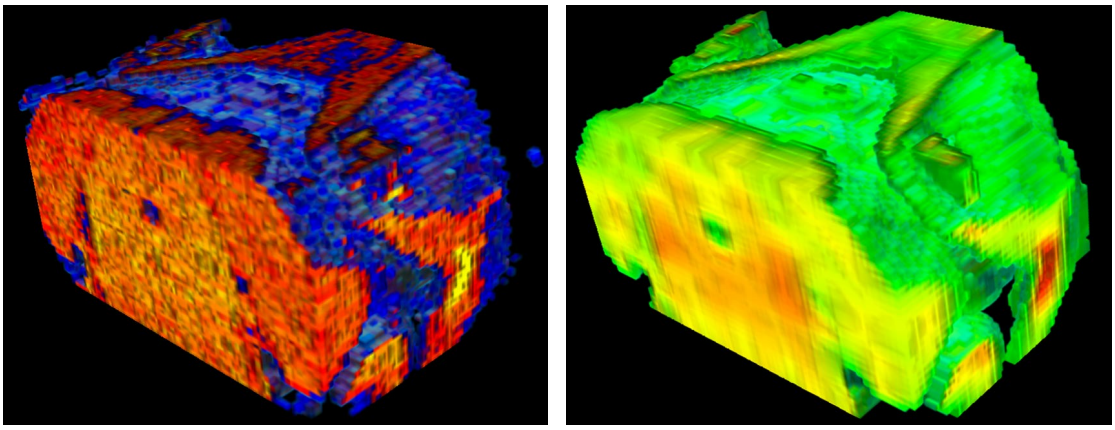
Choose $\tau > 0$

1. Let $p^0 = 0, n = 0$.
2. Suppose p^n is known, we compute p^{n+1} as follows:

$$p^n_{i,j,k} = p^{n+1}_{i,j,k} + \tau \left[(H \left[H^* p - \frac{\bar{u}}{\lambda} \right])_{i,j,k} + \left\| (H \left[H^* p^n - \frac{\bar{u}}{\lambda} \right])_{i,j,k} \right\|_{\mathbb{R}^9} p^{n+1}_{i,j,k} \right]$$

which implies:

$$p^{n+1}_{i,j,k} = \frac{p^n_{i,j,k} - \tau (H \left[H^* p^n - \frac{\bar{u}}{\lambda} \right])_{i,j,k}}{1 + \tau \left\| (H \left[H^* p^n - \frac{\bar{u}}{\lambda} \right])_{i,j,k} \right\|_{\mathbb{R}^9}}$$



(a) Original 3D image (b) Filtered 3D image

Figure 12: Denoising with second-order variational model - $\lambda = 20$

3.3 Approximated Mumford Shah method

We mentionned above that we used the Mumford-Shah [12] method as a preprocessing treatment. In fact, it an alternative to snakes methods. We used an approximate version as in [3]. This method is simple to use but it depends on many parameters that we have to tune. We first recall what the Ambrosio-Tortorelli approximation for the Mumford-Shah model is. We present the method in the general n-dimensional framework. We used it for 2D and 3D images.

Let Ω be a bounded open domain of $\mathbb{R}^n$, $n = 2, 3$ and a function $u_o : \Omega \to [0, 1]$ that represents the normalized image to segment. We look for the contours as the compact set of jumps of u_0 and a smooth approximation to u_0 outside Γ that we call u. We look for a pair (Γ, u) that minimizes the **Mumford-Shah functional**

$$J_{MS}(K, u) = \int_{\Omega\setminus\Gamma} |\nabla u|^2 \, dx + \alpha \int_{\Omega\setminus\Gamma} |u - u_o|^2 \, dx + \beta \, \ell(\Gamma) \ , \tag{3.2}$$

where $\ell(\Gamma)$ is the perimeter of Γ, α and β are scaling and contrast parameters. From the practical point of view the Munford-Shah functional has to be approximated. We have chosen the Ambrosio-Tortorelli approach [3] which is easy to implement. We introduce an auxiliary function v that is a approximate function to the characteristic function $1 - \chi_\Gamma$ of $\Omega\setminus\Gamma$ (equal to 0 on Γ and 1 outside). For every $\varepsilon > 0$ set

$$F_\varepsilon(u, \Gamma = \alpha \int_\Omega (u - u_o)^2 dx + \beta \int_\Omega v^2 |\nabla u|^2 dx + \int_\Omega \left(\varepsilon |\nabla v|^2 + \frac{1}{4\varepsilon}(v - 1)^2 \right) dx \ .$$

The minimum of F_ε is computed by solving the system:

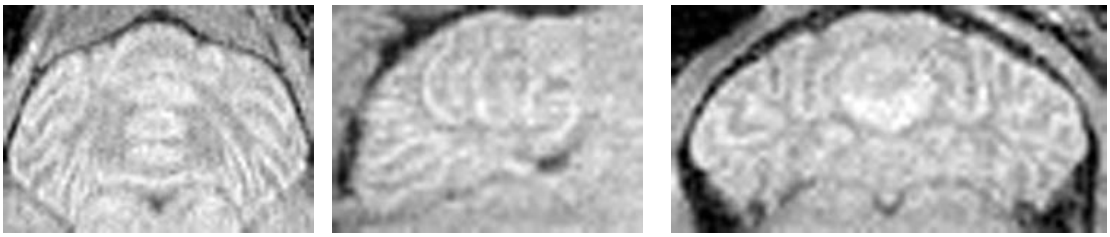
$$\left\{ \begin{array}{rcl} \dfrac{\partial u}{\partial t} & = & -2\alpha(u - u_o) + 2\beta \text{ div } (v^2 \nabla u) \quad \text{in } \Omega \ , \\ \dfrac{\partial v}{\partial t} & = & 2\varepsilon \Delta v - \frac{1}{2\varepsilon}(v - 1) - 2\beta |\nabla u|^2 v \quad \text{in } \Omega \ , \end{array} \right.$$

with usual homogeneous Neuman boundary conditions (normal derivative equal to 0 on the boudary). We have used a time explicit Euler scheme and a classical finite difference spatial discretization. This part can be improved of course.

4 Numerical results

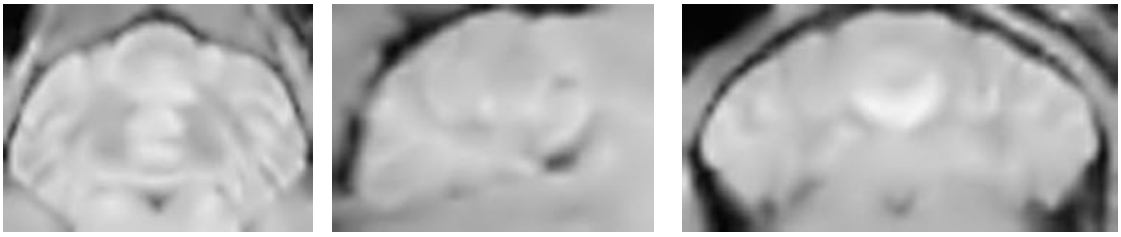
4.1 2D approach

We first have performed 2D segmentation of every slice of 3D stack using the coronal, sagittal and axial views respectively. We have computed the 3D volume by counting the pixels inside the obtained contours for each slice.



(a) Axial view - slide 18 (b) Sagittal view - slice 30 (c) Coronal view - slice 25

Figure 13: Original slices



(a) Axial view - slide 18 (b) Sagittal view - slice 30 (c) Coronal view - slice 25

Figure 14: Denoised slices - $\lambda = 10$

(a) Axial view - slide 18 (b) Sagittal view - slice 30 (c) Coronal view - slice 25

Figure 15: Mumford-Shah segmentation - $\alpha = 1,\ \beta = 0.5,\ \varepsilon = 0.001$

As the Mumford-Shah model depends on parameters $(\alpha, \beta, \varepsilon)$ we performed tests with different values

$$\alpha, \beta \in \{0.1, 1, 5, 10, 20\}, \quad \varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}\} ,$$

a maximal number of iterations set to $N_{max} = 10000$ and a time step $\delta t = 10^{-4}$. Then we chose the median volume among the results. We report the results below:

Mouse	H/T	Axial	Sagittal	Coronal	Mean :Axial Sagittal	Mean : Sagittal Coronal	Mean : Coronal Axial	Mean 3 views
CN2	H	28560	31040	27710	29800	29375	28135	29103
CW3	H	40560	43160	39590	41860	41375	40075	41103
CX2	H	35270	274200	34650	36345	36035	34960	35780
CY1	H	41980	40150	39540	41065	39845	40760	40557
CZ3	H	32100	33450	31350	32775	32400	31725	32300
s79092	H	57130	60160	56680	58645	58420	56905	57990
s79103	H	45220	46060	45450	45640	45755	45335	45577
s79220	H	30820	35400	29210	33110	32305	30015	31810
s79224	H	34160	36240	33500	35200	34870	33830	34633
s79102	T	23060	27510	23650	25285	25580	23355	24740
s79105	T	32220	36060	32670	34140	34365	32445	33650
s79106	T	34040	41890	36040	37965	38965	35040	37323
s79219	T	25470	30200	26580	27835	28390	26025	27417
s79225	T	25660	30460	26120	28060	28290	25890	27413

Table 1: Volume of cerebellum for healthy (H) and trisomic (T) mice

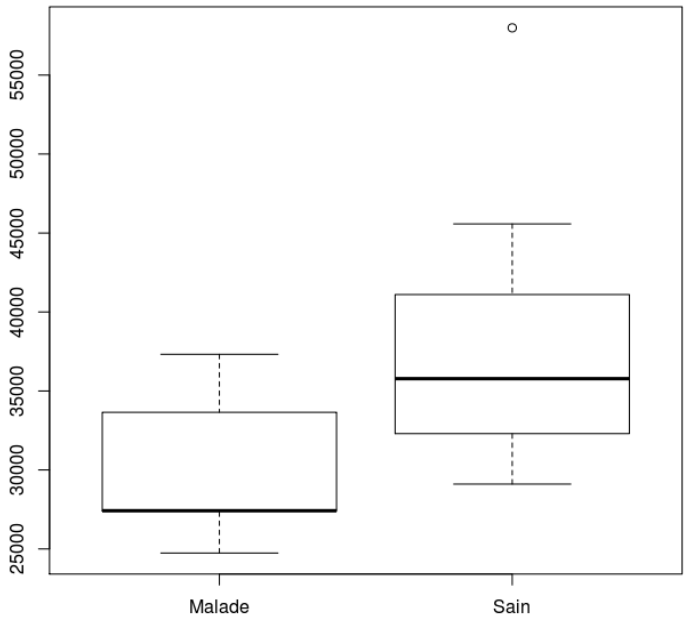
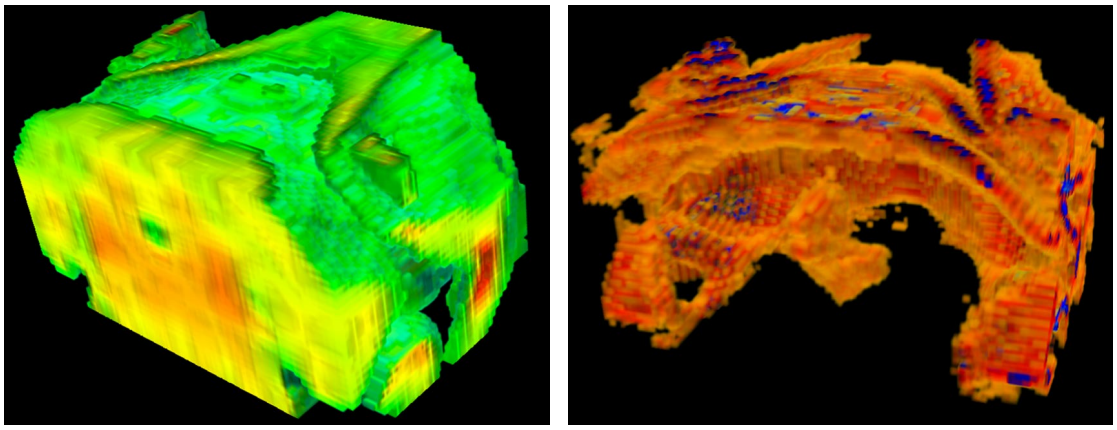


Figure 16: Volumes comparison

A short statistical study is finally carried out on (average) estimated cerebellum volumes. On our sample, healthy mouses seem to have a greater estimated cerebellum volume than trisomic ones (for instance in terms of median, first and third quartiles, see Figure 16). This difference is proved to be significant from a Mann-Whitney test (whose p-value=0.02997). This is enough to conclude healthy mouses tend (stochastically) to have a greater cerebellum volume than trisomic ones (with a risk of 0.03 of being wrong).

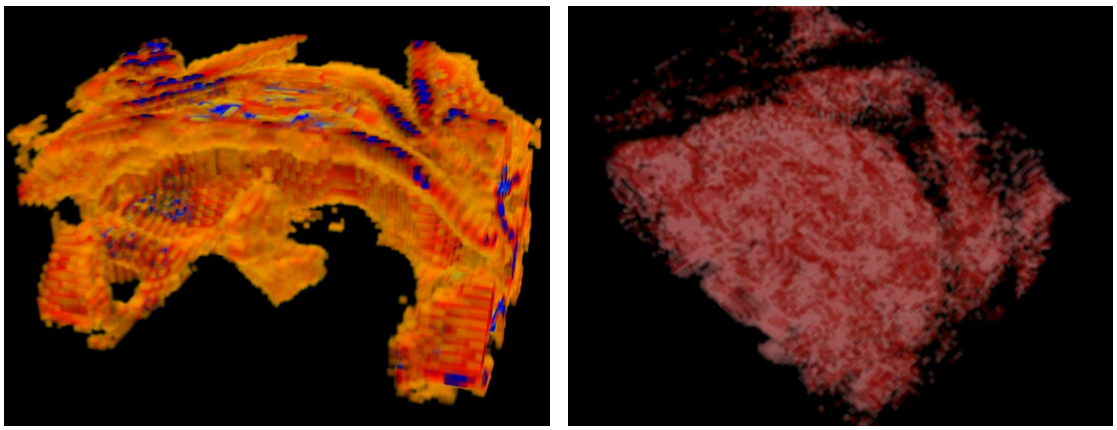
4.2 3D approach

We have noticed that the best choice was $\varepsilon = 0.1$ and that $\alpha = 1$ and $\beta = 0.1$.



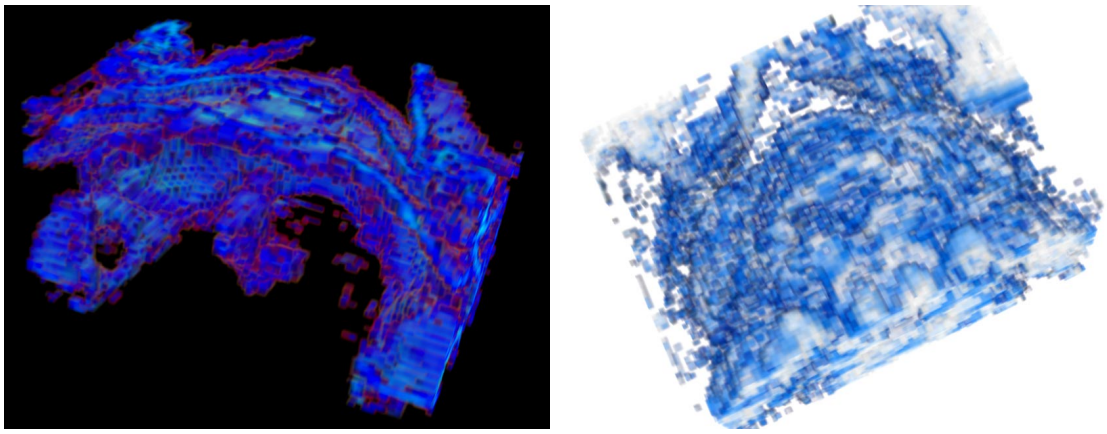
(a) Filtered 3D image - $\lambda = 20$ (b) Contours with $\alpha = 1,\ \beta = 0.01,\ \varepsilon = 0.1$

Figure 17: 3D Segmentation with approximate Mumford-Shah model



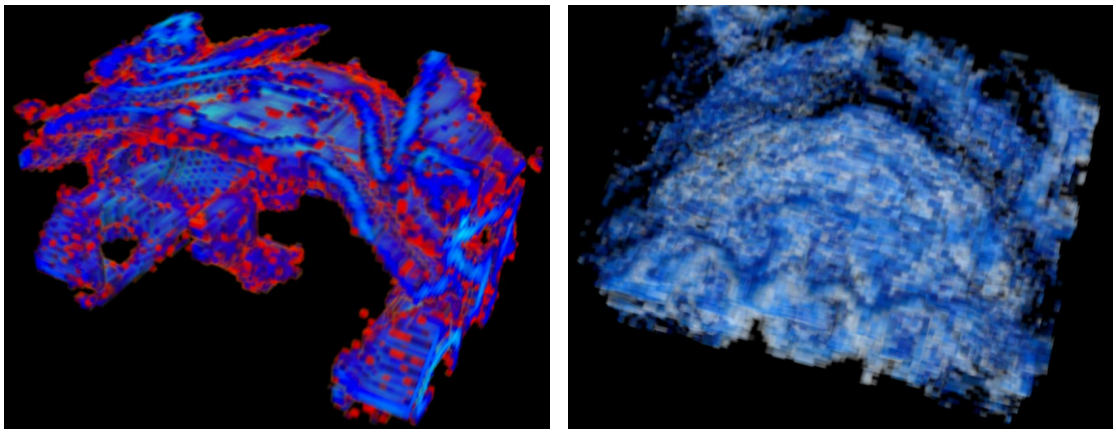
(a) Outside (b) Inside

Figure 18: 3D Segmentation with $\alpha = 1,\ \beta = 0.01,\ \varepsilon = 0.1$ - $\lambda = 20$



(a) Outside (b) Inside

Figure 19: 3D Segmentation with $\alpha = 1,\ \beta = 0.1,\ \varepsilon = 0.1$ - $\lambda = 10$



(a) Outside (b) Inside

Figure 20: 3D Segmentation with $\alpha = 1,\ \beta = 0.1,\ \varepsilon = 0.1$ - $\lambda = 20$

Once we have obtained the contours, we perform a thresholding and count the voxels from the center of the box to the contours in order to get the volume estimate. We compute the 3D volume by counting the voxels inside the obtained contours.

5 Conclusion

The complete process gives promising results. In particular, the hypothesis that trisomic mice have a smaller cerebellum than healthy ones can be validated. Therefore, the methodology can be used for other pathologies: the use of MRI imaging implies that mice are not sacrified.

However, the method is not robust enough since the segmentation is not completely satisfactory. We actually develop (statistical) region methods that should be more precise even in the case where images are not of high quality. This will be adressed in a future work.

Appendix

Gradient

The discrete gradient of an image $u \in Y$ is computed with a forward finite difference scheme:

$$(\nabla u_{i,j,k}) = (\nabla u^1_{i,j,k}, \nabla u^2_{i,j,k}, \nabla u^3_{i,j,k})$$

where

$$\nabla u^1_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k}, \ i < N_1 \\ 0, \ i = N_1 \end{cases}$$

$$\nabla u^2_{i,j,k} = \begin{cases} u_{i,j+1,k} - u_{i,j,k}, \ j < N_2 \\ 0, \ j = N_2 \end{cases}$$

$$\nabla u^3_{i,j,k} = \begin{cases} u_{i,j,k+1} - u_{i,j,k}, \ k < N_3 \\ 0, \ k = N_3 \end{cases}$$

Second order derivative

The second order derivative of $v \in Y$ is computed as :

$$\begin{aligned}(Hv)_{i,j,k} =& (Hv^{11}_{i,j,k}, Hv^{12}_{i,j,k}, Hv^{13}_{i,j,k}, Hv^{21,}_{i,j,k} \\ & Hv^{22}_{i,j,k}, Hv^{23}_{i,j,k}, Hv^{31}_{i,j,k}, Hv^{32}_{i,j,k}, Hv^{33}_{i,j,k}).\end{aligned}$$

For every $i = 1, ..., N_1$, $j = 1, ..., N_2$ and $k = 1, ..., N_3$,

$$(Hv)^{11}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} + v_{i-1,j,k} & 1 < i < N_1 \\ v_{i+1,j,k} - v_{i,j,k}, & i = 1 \\ v_{i,j,k} - v_{i-1,j,k}, & i = N_1 \end{cases}$$

$$(Hv)^{12}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i-1,j+1,k} + v_{i-1,j,k}, \\ \qquad 1 < i \leqslant N_1, \ 1 \leqslant j < N_2 \\ 0, \quad j = N_2 \\ 0, \quad i = 1 \end{cases}$$

$$(Hv)^{22}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} + v_{i,j-1,k}, \ 1 < j < N_2 \\ v_{i,j+1,k} - v_{i,j,k}, \ j = 1 \\ v_{i,j,k} - v_{i,j-1,k}, \ j = N_2 \end{cases}$$

$$(Hv)^{23}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} - v_{i,j-1,k+1} + v_{i,j-1,k}, \\ \qquad 1 < j \leqslant N_2, \ \ 1 \leqslant k < N_3 \\ 0, \ j = 1 \\ 0, \ k = N_3 \end{cases}$$

$$(Hv)^{31}_{i,j,k} = \begin{cases} v_{i+1,j,k} - v_{i,j,k} - v_{i+1,j,k-1} + v_{i,j,k-1}, \\ \qquad\qquad 1 < k \leqslant N_3,\ 1 \leqslant i < N_1 \\ 0,\ k = 1 \\ 0,\ i = N_1 \end{cases}$$

$$(Hv)^{32}_{i,j,k} = \begin{cases} v_{i,j+1,k} - v_{i,j,k} - v_{i+,j+1,k-1} + v_{i,j,k-1}, \\ \qquad\qquad 1 \leqslant j < N,\ 1 < k \leqslant N_3 \\ 0,\ \ j = N_2 \\ 0,\ \ k = 1 \end{cases}$$

$$(Hv)^{33}_{i,j,k} = \begin{cases} v_{i,j,k+1} - v_{i,j,k} + v_{i,j,k-1},\ 1 < k < N_3 \\ v_{i,j,k+1} - v_{i,j,k},\ k = 1 \\ v_{i,j,k} - v_{i,j,k-1},\ k = N_3 \end{cases}$$

Adjoint of the second order derivative

The adjoint of $H^* : X^9 \to X$ is defined for every $p = (p^{11}, p^{12}, p^{13}, p^{21}, p^{22}, p^{23}, p^{31}, p^{32}, p^{33}) \in X^9$, as

$$\begin{aligned}(H^*p)_{i,j,k} = {} & \sigma^{11}_{i,j,k} + \sigma^{12}_{i,j,k} + \sigma^{13}_{i,j,k} + \sigma^{21}_{i,j,k} + \sigma^{22}_{i,j,k} \\ & + \sigma^{23}_{i,j,k} + \sigma^{31}_{i,j,k} + \sigma^{32}_{i,j,k} + \sigma^{33}_{i,j,k}\end{aligned}$$

where

$$\sigma^{11}_{i,j,k} = \begin{cases} p^{11}_{i+1,j,k} - 2p^{11}_{i,j,k} + p^{11}_{i-1,j,k},\ 1 < i < N_1 \\ p^{11}_{i+1,j,k} - p^{11}_{i,j,k},\ i = 1 \\ p^{11}_{i-1,j,k} - p^{11}_{i,j,k},\ i = N_1 \end{cases}$$

$$\sigma^{22}_{i,j,k} = \begin{cases} p^{22}_{i,j+1,k} - 2p^{22}_{i,j,k} + p^{22}_{i,j-1,k},\ 1 < j < N_2 \\ p^{22}_{i,j+1,k} - p^{22}_{i,j,k},\ j = 1 \\ p^{22}_{i,j-1,k} - p^{22}_{i,j,k},\ j = N_2 \end{cases}$$

$$\sigma^{33}_{i,j,k} = \begin{cases} p^{33}_{i,j,k+1} - 2p^{33}_{i,j,k} + p^{33}_{i,j,k-1},\ 1 < k < N_3 \\ p^{33}_{i,j,k+1} - p^{33}_{i,j,k},\ k = 1 \\ p^{33}_{i,j,k-1} - p^{33}_{i,j,k},\ k = N_3 \end{cases}$$

$$\sigma^{12}_{i,j,k} = \begin{cases} p^{12}_{i+1,j,k},\ i = 1,\ j = 1 \\ -p^{12}_{i+1,j-1,k},\ i = 1,\ j = N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i+1,j-1,k},\ \ i = 1,\ 1 < j < N_2 \\ -p^{12}_{i,j,k},\ i = N_1,\ j = 1 \\ p^{12}_{i,j-1,k},\ i = N_1,\ j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k},\ \ i = N_1,\ 1 < j < N_2 \\ p^{12}_{i+1,j,k} - p^{12}_{i,j,k},\ \ 1 < i < N_1,\ j = 1 \\ p^{12}_{i,j-1,k} - p^{12}_{i+1,j-1,k},\ \ 1 < i < N_1,\ j = N_2 \\ p^{12}_{i,j-1,k} - p^{12}_{i,j,k} - p^{12}_{i+1,j-1,k} + p^{12}_{i+1,j,k}, \\ \qquad\qquad 1 < i < N_1,\ 1 < j < N_2 \end{cases}$$

$$\sigma^{13}_{i,j,k} = \begin{cases} p^{13}_{i+1,j,k}, \quad i=1,\ k=1 \\ -p^{13}_{i+1,j,k-1}, \quad i=1,\ k=N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i+1,j,k-1}, \quad i=1,\ 1<j<N_3 \\ -p^{13}_{i,j,k}, \quad i=N_1,\ k=1 \\ p^{13}_{i,j,k-1}, \quad i=N_1,\ k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k}, \quad i=N_1,\ 1<k<N_3 \\ p^{13}_{i+1,j,k} - p^{13}_{i,j,k}, \quad 1<i<N_1,\ k=1 \\ p^{13}_{i,j,k-1} - p^{13}_{i+1,j,k-1}, \quad 1<i<N_1,\ k=N_3 \\ p^{13}_{i,j,k-1} - p^{13}_{i,j,k} - p^{13}_{i+1,j,k-1} + p^{13}_{i+1,j,k}, \\ \qquad\qquad 1<i<N_1,\ 1<k<N_3 \end{cases}$$

$$\sigma^{21}_{i,j,k} = \begin{cases} p^{21}_{i,j+1,k}, \quad j=1,\ i=1 \\ -p^{21}_{i-1,j+1,k}, \quad j=1,\ i=N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i-1,j+1,k}, \quad j=1,\ 1<i<N_1 \\ -p^{21}_{i,j,k}, \quad j=N_2,\ i=1 \\ p^{21}_{i-1,j,k}, \quad j=N_2,\ i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k}, \quad j=N_2,\ 1<i<N_1 \\ p^{21}_{i,j+1,k} - p^{21}_{i,j,k}, \quad 1<j<N_2,\ i=1 \\ p^{21}_{i-1,j,k} - p^{21}_{i-1,j+1,k}, \quad 1<j<N_2,\ i=N_1 \\ p^{21}_{i-1,j,k} - p^{21}_{i,j,k} - p^{21}_{i-1,j+1,k} + p^{21}_{i,j+1,k}, \\ \qquad\qquad 1<j<N_2,\ 1<i<N_1 \end{cases}$$

$$\sigma^{23}_{i,j,k} = \begin{cases} p^{23}_{i,j+1,k}, \quad j=1,\ k=1 \\ -p^{23}_{i,j+1,k-1}, \quad j=1,\ k=N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j+1,k-1}, \quad j=1,\ 1<k<N_3 \\ -p^{23}_{i,j,k}, \quad j=N_2,\ k=1 \\ p^{23}_{i,j,k-1}, \quad j=N_2,\ k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k}, \quad j=N_2,\ 1<k<N_3 \\ p^{23}_{i,j+1,k} - p^{23}_{i,j,k}, \quad 1<j<N_2\ , k=1 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j+1,k-1}, \quad 1<j<N_2,\ k=N_3 \\ p^{23}_{i,j,k-1} - p^{23}_{i,j,k} - p^{23}_{i,j+1,k-1} + p^{23}_{i,j+1,k}, \\ \qquad\qquad 1<j<N_2,\ 1<k<N_3 \end{cases}$$

$$\sigma^{31}_{i,j,k} = \begin{cases} p^{31}_{i,j,k+1}, \quad k=1,\ i=1 \\ -p^{31}_{i-1,j,k+1}, \quad k=1,\ i=N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i-1,j,k+1}, \quad k=1,\ 1<i<N_1 \\ -p^{31}_{i,j,k}, \quad k=N_3,\ i=1 \\ p^{31}_{i-1,j,k}, \quad k=N_3,\ i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k}, \quad k=N_3,\ 1<i<N_1 \\ p^{31}_{i,j,k+1} - p^{31}_{i,j,k}, \quad 1<k<N_3,\ i=1 \\ p^{31}_{i-1,j,k} - p^{31}_{i-1,j,k+1}, \quad 1<k<N_3,\ i=N_1 \\ p^{31}_{i-1,j,k} - p^{31}_{i,j,k} - p^{31}_{i-1,j,k+1} + p^{31,}_{i,j,k+1}, \\ \qquad\qquad 1<k<N_3, 1<i<N_1 \end{cases}$$

$$\sigma^{32}_{i,j,k} = \begin{cases} p^{32}_{i,j,k+1}, & k=1,\ 1=1 \\ -p^{32}_{i,j-1,k+1}, & k=1,\ j=N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j-1,k+1}, & k=1,\ 1<j<N_2 \\ -p^{32}_{i,j,k}, & k=N_3,\ j=1 \\ p^{32}_{i,j-1,k}, & k=N_3,\ j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k}, & k=N_3,\ 1<j<N_2 \\ p^{32}_{i,j,k+1} - p^{32}_{i,j,k}, & 1<k<N_3,\ j=1 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j-1,k+1}, & 1<k<N_3,\ j=N_2 \\ p^{32}_{i,j-1,k} - p^{32}_{i,j,k} - p^{32}_{i,j-1,k+1} + p^{32}_{i,j,k+1}, & \\ & 1<k<N_3, 1<j<N_2 \end{cases}$$

Divergence

Finally the 3D divergence is computed with a backward scheme: for every $(p^1, p^2, p^3) \in Y$

$$\begin{aligned} (\text{div } p)_{i,j,k} = & \begin{cases} p^1_{i,j,k} - p^1_{i-1,j,k} & \text{if } 1<i<N_1 \\ p^1_{i,j,k} & \text{if } i=1 \\ -p^1_{i-1,j,k} & \text{if } i=N_1 \end{cases} \\ & + \begin{cases} p^2_{i,j,k} - p^2_{i,j-1,k} & \text{if } 1<j<N_2 \\ p^2_{i,j,k} & \text{if } j=1 \\ -p^2_{i,j-1,k} & \text{if } j=N_2 \end{cases} \\ & + \begin{cases} p^3_{i,j,k} - p^3_{i,j,k-1} & \text{if } 1<jk<N_3 \\ p^3_{i,j,k} & \text{if } k=1 \\ -p^3_{i,j,k-1} & \text{if } k=N_3 \end{cases} \end{aligned}$$

References

[1] G. Aubert & P. Kornprobst, Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations, Applied Mathematical Sciences 147, Springer Verlag (2006).

[2] A.Almhdie, P. Lopes-Pereira, S. Meme, C.Colombier, V. Brault, F. Szeremeta, B.Doan, R. Ledee, R. Harba, Y. Herault, J. -C. Beloeil & C. Leger, Chan-Vese based method to segment mouse brain MRI images: Application to cerebral malformation analysis in trisomy 21,17th European Signal Processing Conference, EUSIPCO 2009 24-28 August 2009.

[3] L. Ambrosio & V.M. Tortorelli, Approximation of functionals via-convergence, Commun. Pure Appl. Math. 13 (1990) 999–1036.

[4] T. Chan, S. Esedoglu, F. Park , A. Yip, Recent Developments in total Variation Image Restoration, CAM Report 05-01, Department of Mathematics, UCLA 2004.

[5] M. Bergounioux & M.P. Tran, A second order model for 3D texture extraction, Mathematical Image Processing. Springer proceedings in Mathematics 5, 2011.

[6] M. Bergounioux & L. Piffet , A second-order model for image denoising and/or texture extraction, Set Valued and Variational Analysis, Vol. 18, 3-4, pp. 277-306, 2010

[7] M. Bergounioux & L. Guillot, Existence and uniqueness results for the gradient vector flow and geodesic active contours mixed model Comunications on Pure and Applied Analysis, Volume 8, Number 4, July 2009

[8] A. Chambolle, An algorithm for total variation minimization and applications, Journal of Mathematical Imaging and Vision, 20 89–97 (2004)

[9] T. Chan & L. Vese, Active Contours Without Edges, IEEE Transactions on Image Processing, Vol. 10, no 2, 2001.

[10] N. Forcadel, C. Gout & C. Le Guyader, Generalized Fast Marching Method: Applications to Image Segmentation, Numerical Algorithms.

[11] O. Houdé, B.Mazoyer & N. Tzourio-Mazoyer, Cerveau et psychologie: Introduction à l'imagerie cérébrale anatomique et fonctionnelle, PUF, 2002.

[12] D. Mumford & J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (1989) 577–685

[13] S. Osher, E. Fatemi & L. Rudin, Nonlinear total variation based noise removal algorithms, Physica D 60, 259–268 (1992)

[14] S. J. Osher & L. Vese, Image denoising and decomposition with total variation minimization and oscillatory functions. Special issue on mathematics and image analysis, J. Math. Imaging Vision, 20, no. 1-2, 7–18 (2004)

[15] C. Xu & J. L. Prince, Gradient Vector Flow: A new external force for snakes, IEEE Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'97), (1997), 66–71.

[16] C. Xu & J. L. Prince, Snakes, Shapes, and Gradient Vector Flow, IEEE Transactions on image processing, **7** (1998), 359–369.