2.2 Creaseness operator

Antonio López, a member of our group, has studied for his PhD dissertation the differential operators described in this section [72]. He has compared the performance of various classical creaseness operators, and studies the theoretical reason of their lack of continuity. Following, we synthesise his results for the sake of completeness, but for further information regarding particular points, the cited bibliography should be consulted.

Today's literature defines a crease (ridge or valley) in many different ways [10]. Intuitively, a crease can be defined:

1. paths where water gathers downhill when it rains in a landscape (turn the landscape upside-down for a ridge) .

2. points with sharp drops on both hand sides. .

3. when walking uphill, points where you make a sharp turn. .

Before giving the proper definitions, we need to introduce some notation.

We define a discrete image as the sampling of a d-dimensional continuous function:

$$L : \Omega \subset \mathbb{R}^d \rightarrow \Gamma \subset \mathbb{R} \tag{2.1}$$

the level set for a constant l consists of the set of points

$$\mathcal{S}_l = \{\mathbf{x} \in \Omega | L(\mathbf{x}) = l\} \tag{2.2}$$

Figure 2.5 shows the graphical interpretation for the $d = 2$. For the 3–D case, the level sets will be surfaces having the same intensity level. A related family of curves are the slope lines, which follow the gradient and are orthogonal to the level lines.

Operators over a function are defined as usual:

$$\nabla = (\partial/\partial x^1, \ldots, \partial/\partial x^d) \qquad \text{(Gradient)} \tag{2.3}$$

$$\nabla\nabla = (\nabla^{\mathrm{t}} \cdot \nabla) \qquad \text{(Hessian)} \tag{2.4}$$

The derivative of L along the vector $\mathbf{v} = (v^1, \ldots, v^d)^{\mathrm{t}}$:

$$L_{\mathbf{v}} = \nabla L \cdot (\mathbf{v}/\|\mathbf{v}\|) \tag{2.5}$$

And the second-order derivative of L along $\mathbf{v}$ and $\mathbf{w} = (w^1, \ldots, w^d)^{\mathrm{t}}$:

$$L_{\mathbf{vw}} = (\mathbf{v}^{\mathrm{t}}/\|\mathbf{v}\|) \cdot \nabla\nabla L \cdot (\mathbf{w}/\|\mathbf{w}\|) \tag{2.6}$$

Derivatives along the axis of coordinates x^i are annotated as L_{x^i}. In 2–D, useful definitions are $\mathbf{w} = (L_x, L_y)^{\mathrm{t}}$ and $\mathbf{v} = (L_y, -L_x)^{\mathrm{t}}$.

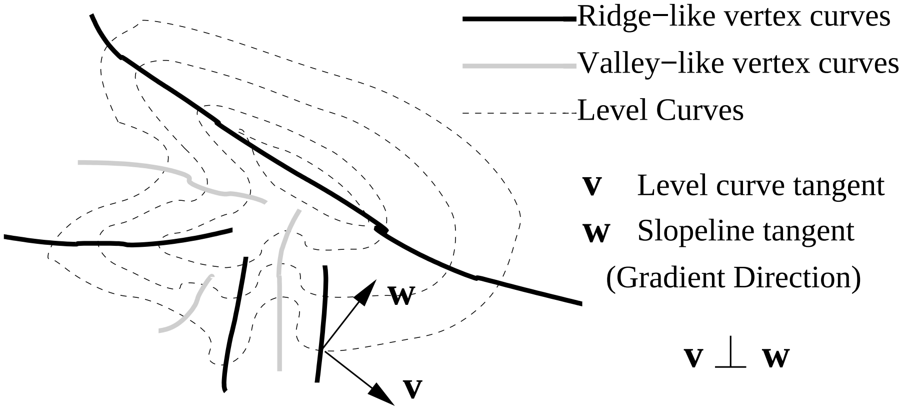

Figure 2.5: Graphical interpretation of terms related to level curves: ridges, valleys as lines following the minimum and maximum curvature, and the v and w vectors at any point.

2.2.1 Three classical definitions of creases

Definition 1: paths where water gathers downhill when it rains in a landscape.

This definition, due to R. Rothe, was saved from oblivion in [36, 37, 38]. It can be written as "parts of slope-lines where other slope-lines converge". Slope-lines are solutions of the ordinary differential equation:

$$\nabla_{\perp} I \cdot \mathrm{d}x = 0 \tag{2.7}$$

Given an initial point, the slope-line including a point will be the set of consecutive points solution of equation 2.7, which is done for instance by means of Runge-Kutta (as implemented in [80]). That can be seen as tracing the path of a rain drop as it runs down-hill. López in [51] studied the algorithmic implementation compared to other operators, but results where not satisfactory.

The main drawback is that the operator is not local, which means that it cannot be described as a function of a neighbourhood. That is to say, many initial points are needed to properly trace the crests, and that has a very high computational cost.

We extensively studied the response of this operator for registration purposes during the initial stages of the research; the registration achieved with this scheme some results [44]. In that paper, we compared its response to other operators, and also started preliminary registrations. However, the final accuracy was limited because the segmentation presented gaps were it was not expected, similarly to those of the $L_{\mathbf{vv}}$operator, explained in the following pages.

The first year of my research was dedicated to study this operator, and the first two publications [44, 52] were based on those results. Afterwards, another creaseness operator compared favourably to Rothe's, so it was replaced in the registration algorithm. See an example with the response for a heart gammagraphic image in figure 2.6.

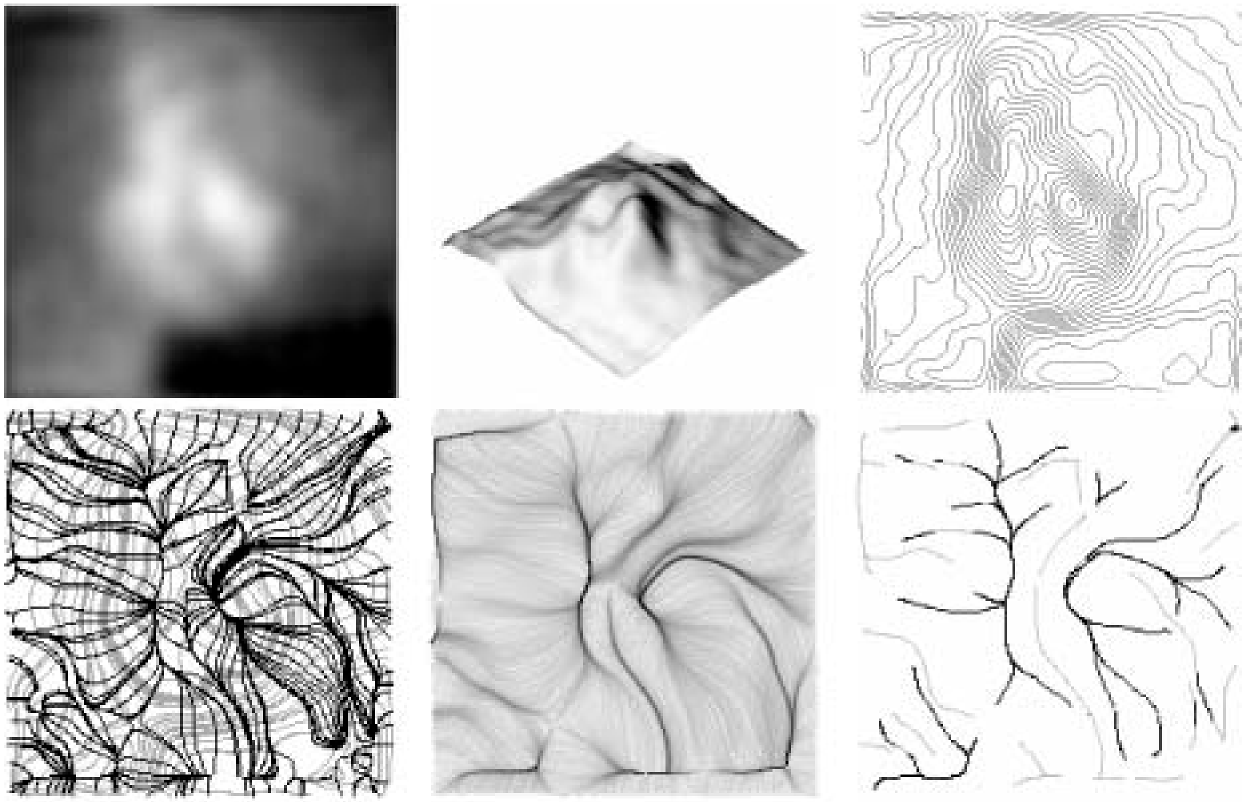

Figure 2.6: From top to bottom and left to right: (a) Heart gammagraphic image. (b) Image seen as a landscape. (c) Level curves. (d) Some slopelines (black) over the level curves (grey). Observe how slopelines crowd along special slopeline segments. (e) Accumulation image where darker grey-levels denote greater accumulation. (f) Selected special slopelines segments: drainage pattern (grey) and inverse drainage pattern (dark).

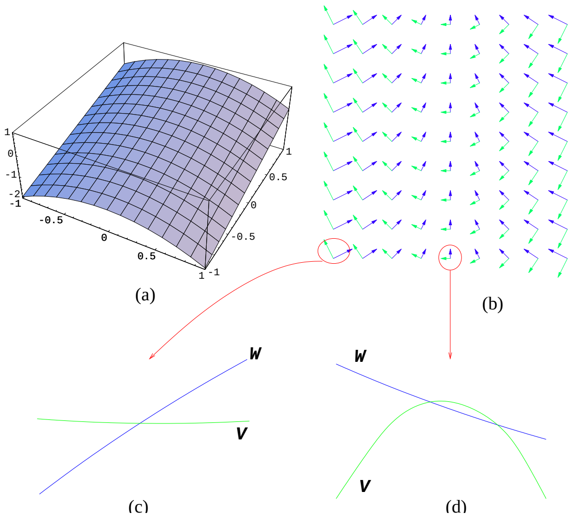

Figure 2.7: Graphical interpretation of the $L_{\mathbf{vv}}$ operator: a) shows the shape function with a crest line; b) at each point the direction of the gradient w in dark and the vector perpendicular v in grey. c) shows the profile following each vector at non-crest and crest (d) points.

Definition 2: points with sharp drops on right and on the left hand side.

We will refer to figure 2.7 to explain the mathematical definition. At any point of an image seen as a landscape, w points uphill, and v is perpendicular to w. If we look at the function profile for a section following v, the shape will be different depending on the neighbouring values: for point at (c), it will be flat , but for a crease point (d) it will clearly show a parabola.

Therefore $L_{\mathbf{vv}}$, which is the second derivative of the function along $\mathbf{v}$, will take high values for valleys and low values for ridges. The explicit formula for $L_{\mathbf{vv}}$ for $d = 2$ [56] is:

$$L_{\mathbf{vv}} = \frac{L_y^2 L_{xx} + L_x^2 L_{yy} - 2L_x L_y L_{xy}}{L_x^2 + L_y^2} \tag{2.8}$$

Definition number 3: when walking uphill, points where you need to make a sharp turn.

This definition recalls that of the level-curve curvature κ: following a level curve, the points where the curvature is higher (lower) determine the valleys (ridge). It has the following analytical expression:

$$\kappa = (2L_x L_y L_{xy} - L_y^2 L_{xx} - L_x^2 L_{yy})(L_x^2 + L_y^2)^{-\frac{3}{2}} \ , \tag{2.9}$$

Comparing equations 2.8 and 2.9, we see that $L_{\mathbf{vv}}$ is κ multiplied by the gradient magnitude $L_{\mathbf{w}}$:

$$\kappa = -L_{\mathbf{vv}}/L_{\mathbf{w}} = -L_{\mathbf{vv}} L_{\mathbf{w}}^{\alpha}, \qquad \alpha = -1 \tag{2.10}$$

For α taking values other than -1, $\alpha \in [-1, 0]$, κ is normalised in several degrees with respect to the gradient magnitude.

2.2.2 New creaseness operators

The differential operators κ and its reduced form $L_{\mathbf{vv}}$ are in fact the particular expression for 2–D of the curvature of the level-curves, which, in 3–D, is termed mean curvature κ_{M} of the level surfaces. In the d–dimensional case we will call it *level-set extrinsic curvature*, LSEC .

The general formula for LSEC uses Einstein summation convention:

$$\kappa_d = (L_\alpha L_\beta L_{\alpha\beta} - L_\alpha L_\alpha L_{\beta\beta})(L_\gamma L_\gamma)^{-\frac{3}{2}}, \quad \alpha, \beta, \gamma \in \mathcal{X}_d \ , \tag{2.11}$$

where $\mathcal{X}_d$ is the d-dimensional (local) coordinate system $\{x^1, \ldots, x^d\}$.

In 3–d we have level surfaces and the straightforward extension of κ is two times the Mean curvature κ_{M} of the level surfaces:

$$\begin{aligned}\kappa_{\mathrm{M}} = \Big[&2(L_x L_y L_{xy} + L_x L_z L_{xz} + L_y L_z L_{yz}) - L_x^2(L_{yy} + L_{zz}) \\ &-L_y^2(L_{xx} + L_{zz}) - L_z^2(L_{xx} + L_{yy})\Big]\ 2(L_x^2 + L_y^2 + L_z^2)^{-\frac{3}{2}}\end{aligned} \tag{2.12}$$

However, the way LSEC is usually computed —directly discretizing Eq. (2.12) or the equivalent expression in the 2–d case for κ— gives rise to a number of discontinuities : there appear gaps at places where we would not expect any reduction of creaseness because they are at the centre of elongated objects and creaseness is a measure of medialness for grey–level elongated objects.

We have checked that this happens independently of the scheme of discretizing derivatives and even when the gradient magnitude is far away from the machine's zero, this is, at pixels where Eq. (2.12) is well-defined.

To avoid discontinuities we present an alternative way of computing creaseness from the level set extrinsic curvature. For a d–dimensional vector field $\mathbf{u} : \mathbb{R}^d \to \mathbb{R}^d, \mathbf{u}(\mathbf{x}) = (u^1(\mathbf{x}), ..., u^d(\mathbf{x}))^{\mathrm{t}}$, the divergence measures the degree of parallelism of its integral lines :

$$\mathrm{div}(\mathbf{u}) = \sum_{i=1}^{d} \frac{\partial u^i}{\partial x^i} \tag{2.13}$$

Now, if we define the normalised gradient vector field of $L : \mathbb{R}^d \to \mathbb{R}$ as

$$\bar{\mathbf{w}} = \begin{cases} \mathbf{w}/\|\mathbf{w}\| & \text{if} \quad \|\mathbf{w}\| > 0 \\ \mathbf{0}_d & \text{if} \quad \|\mathbf{w}\| = 0 \end{cases} \tag{2.14}$$

it can be shown that:

$$\kappa_d = -\mathrm{div}(\bar{\mathbf{w}}) \tag{2.15}$$

For $d = 3$ Eqs. (2.12) and (2.15) are equivalent in the continuous domain. However, and this is the key point, the same approximation using derivatives substituted

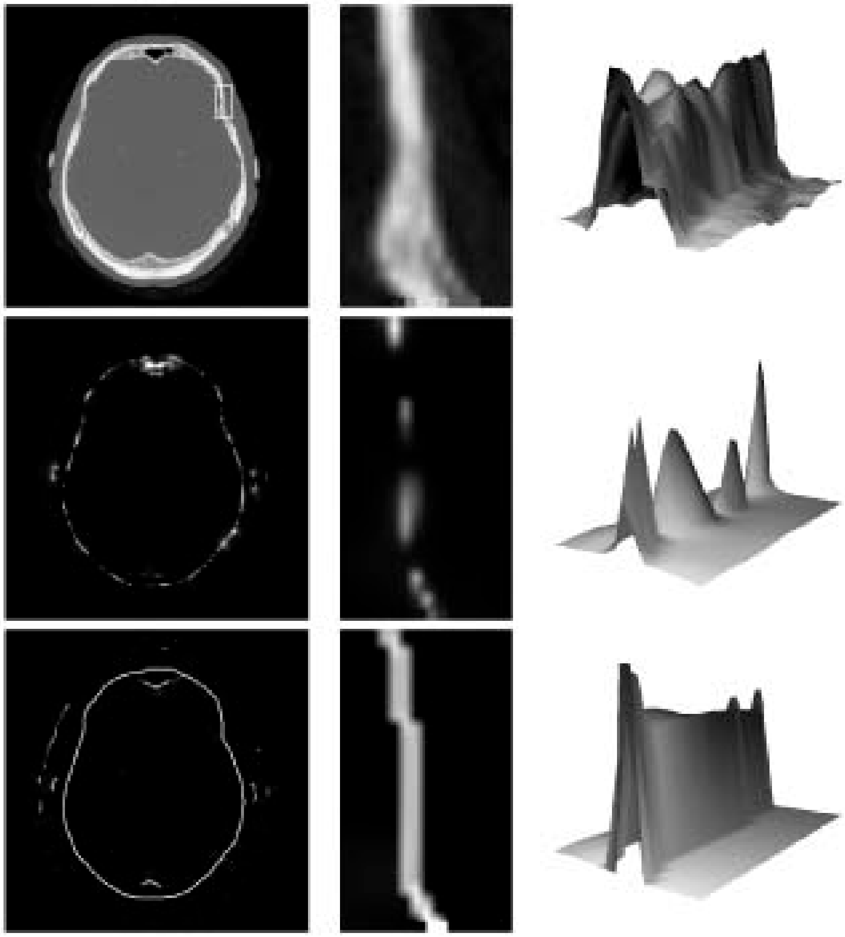

Figure 2.8: Comparison of the operators: on the left row, original CT image with the creaseness images for $L_{\mathbf{vv}}$ and $\mathcal{C}\tilde{\kappa}_d$. The middle column zooms the window extracted from each image at the location show in the CT image. It can be seen that although the bone in the CT image is consistent ridge, gaps appear in the Lvv operator. The cause can be seen if the window is depicted as a landscape (third column): gaps correspond to local minima along the crease, i.e., saddle points. Our operator $\mathcal{C}\tilde{\kappa}_d$ solves this problem.

into Eqs. (2.12) and (2.15) gives rise to very different results, namely, Eq. (2.15) avoids the gaps that Eq. (2.12) produces on creases.

For example, in the case of ridge-like saddle points, as those in figure 2.8, κ not only decreases but also suffers a change of sign barrier on the path of the expected ridge-like curve . The reason is that the neighbourhood of the saddle point is composed mainly of concave zones ($\kappa < 0$) but the sub-pixel ridge-like curve runs through convex zones ($\kappa > 0$) missed by the discretization of Eq. (2.12).

In order to obtain derivatives of a discrete image L in a well–posed manner we have approximated image derivatives by finite centred differences of the Gaussian–smoothed image instead of convolution with Gaussian derivatives of a given standard deviation σ_{D}. For each pixel, neighbour derivatives are calculated 'on the fly' as needed and stored in buffers for the next one. This approach is necessary because otherwise the algorithm would require 10 float images simultaneously. Thus, Eq. (2.15) produces similar results being computationally less time and memory consuming.

The creaseness measure κ_d can still be improved by pre–filtering the image gradient vector field in order to increase the degree of attraction/repulsion at ridge–like/valley–like creases, which is what κ_d is actually measuring. This can be done by the structure tensor analysis ([52], [53]):

1. Compute the gradient vector field $\mathbf{w}$ and the structure tensor field $\mathbf{M}$
$$\mathbf{M}(\mathbf{x};\sigma_{\text{I}}) = G(\mathbf{x};\sigma_{\text{I}}) * (\mathbf{w}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x})^{\text{t}}) \tag{2.16}$$
being $*$ the element–wise convolution of matrix $\mathbf{w}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x})^{\text{t}}$ with the Gaussian window $G(\mathbf{x};\sigma_{\text{I}})$. The standard deviation σ_{I} controls the size of the averaging window.

2. Perform the eigenvalue analysis of $\mathbf{M}$. The normalised eigenvector $\mathbf{w}'$ corresponding to the highest eigenvalue gives the predominant gradient orientation. In the structure tensor analysis, opposite directions are equally treated. Thus, to recover the direction we put $\mathbf{w}'$ in the same quadrant in 2–d, or octant in 3–d, as $\mathbf{w}$. Then, we obtain the new vector field $\tilde{\mathbf{w}}$ and the creaseness measure $\tilde{\kappa}_d$:
$$\tilde{\mathbf{w}} = \text{sign}(\mathbf{w}'^{\text{t}}\mathbf{w})\mathbf{w}' \tag{2.17}$$
$$\tilde{\kappa}_d = -\text{div}(\tilde{\mathbf{w}}) \tag{2.18}$$
The discretization of $\tilde{\kappa}_d$ results from approximating partial derivatives of Eq. (2.13) by finite centred differences of the $\tilde{\mathbf{w}}$ vector field. When the smallest neighbourhood is taken, namely, the 6 pixels neighbouring a pixel (i,j,k), the following expression is obtained:
$$\begin{aligned}\tilde{\kappa}_d[i,j,k] \;=\; & \tilde{w}^1[i+1,j,k] - \tilde{w}^1[i-1,j,k] + \tilde{w}^2[i,j+1,k] - \\ & \tilde{w}^2[i,j-1,k] + \tilde{w}^3[i,j,k+1] - \tilde{w}^3[i,j,k-1]\end{aligned} \tag{2.19}$$

3. Compute a confidence measure $\mathcal{C} \in [0,1]$ to discard creaseness at isotropic areas. As similarity of the eigenvalues $(\lambda_1, \ldots, \lambda_d)$ of the structure tensor implies isotropy we can base the computation of $\mathcal{C}$ on a normalised measure based on their difference. A suitable choice is:
$$\mathcal{C}(\mathbf{x}) = 1 - \exp\left(-\frac{\left(\sum_{i=1}^{d}\sum_{j=i+1}^{d}(\lambda_i(\mathbf{x}) - \lambda_j(\mathbf{x}))^2\right)^2}{2c^2}\right) \tag{2.20}$$
where threshold c is experimentally chosen. In this way, $\mathcal{C}\tilde{\kappa}_d$ has a lower response than $\tilde{\kappa}_d$ at isotropic regions.

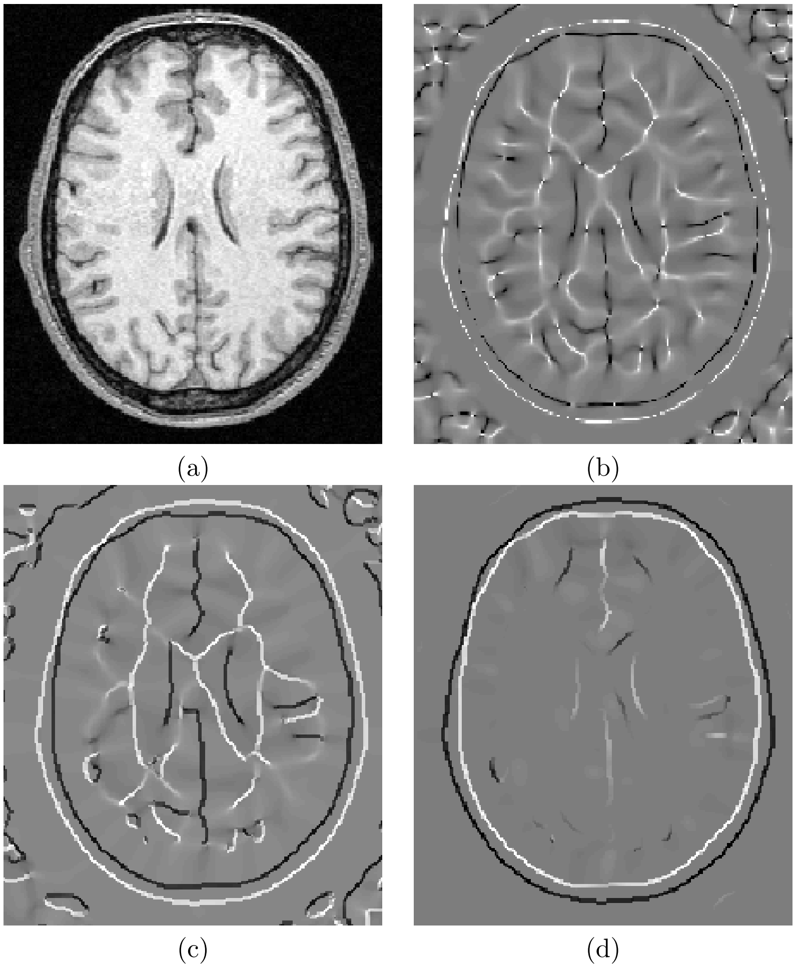

Figure 2.9: From left to right: 3–D creaseness measures of the MR slice in (a) for $\sigma_{\mathrm{D}} = 2.0$ mm: (b) κ, (c) $\tilde{\kappa}$ for $\sigma_{\mathrm{I}} = 2.0$ mm, (d) $\mathcal{C}\tilde{\kappa}$ for $c = 10000$. The gaps in (b) do not appear in (c).

We have tested several methods to compute the eigenvalues. If we only need the highest, the fastest method is Powell's applied to an optimised expression from [21]. If we choose to implement step 3, the fastest is singular value decomposition as implemented in [80]. Figure 2.9 compares κ, $\tilde{\kappa}$ and $\mathcal{C}\tilde{\kappa}$ on an MR volume.