Creaseness from Level Set Extrinsic Curvature

Antonio M. López, Felipe Lumbreras, Joan Serrat

Computer Vision Center and Departament d'Informàtica, Campus UAB,
Edifici O, 08193–Bellaterra (Cerdanyola), Barcelona, Spain.
tel: +34 3 5812561, Fax: +34 3 5811670
antonio@cvc.uab.es

Abstract. Creases are a type of ridge/valley–like structures of a d dimensional image, characterized by local conditions. As creases tend to be at the center of anisotropic grey–level shapes, creaseness can be considered as a type of medialness. Among the several crease definitions, one of the most important is based on the extrema of the level set curvatures. In 2–d it is used the curvature of the level curves of the image landscape, however, the way it is usually computed produces a discontinuous creaseness measure. The same problem arises in 3–d with its straightforward extension and with other related creaseness measures. In this paper, we first present an alternative method of computing the level curve curvature that avoids the discontinuities. Next, we propose the Mean curvature of the level surfaces as creaseness measure of 3–d images, computed by the same method. Finally, we propose a natural extension of our first alternative method in order to enhance the creaseness measure.

Keywords: creaseness, level set curvatures, divergence, structure tensor.

1 Introduction

Ridge/valley–like structures (lines, surfaces, etc) of a d dimensional image, tend to be at the center of anisotropic grey–level objects. Therefore they are useful skeleton–like descriptors. As can be seen in [8] it is possible to characterize ridge/valley–like structures attending to the shape of the image landscape or to its hydrology, giving rise to different mathematical definitions. In the literature we can find a number of local characterizations [2]. We term here as *creases* the ridge/valley–like structures which are characterized locally. Creases have been proposed [5, 2] as a type of medial axis for grey–level objects as the skull in CT and MR images, vessels in arteriographies, roads in aerial photographs, fingerprints, etc.

One of the most useful definitions of creases, due to its invariance properties [2] and its relationship with the shape descriptor called Intensity Axis of Symmetry [5], is the one that is based on the level set curvatures. Given a function $L : \mathbf{R}^d \rightarrow \mathbf{R}$, a level set consists of the points $\mathcal{S}_g = \{\mathbf{x}/L(\mathbf{x}) = g\}$. Different values of g give a sequence of sets $\mathcal{S}_g$ which are the level sets of L. For $d = 2$ the graph of L can be thought as a landscape. Then, the level sets are the level curves

that appear in cartographic maps. In this case, creases are defined through the level curve curvature κ. Negative minima of κ along the level curve tangent direction $\mathbf{v}$, level by level, are valley–like curves and positive maxima are ridge–like curves. We refer here to this type of ridge–like and valley–like curves as *vertex curves*. They can be characterized by the local test

$$e = \nabla\kappa \cdot \mathbf{v} = 0 \tag{1}$$

where $\nabla e \cdot \mathbf{v} < 0$ and $\kappa > 0$ means ridge–like and $\nabla e \cdot \mathbf{v} > 0$ and $\kappa < 0$ means valley–like (Fig. 1).

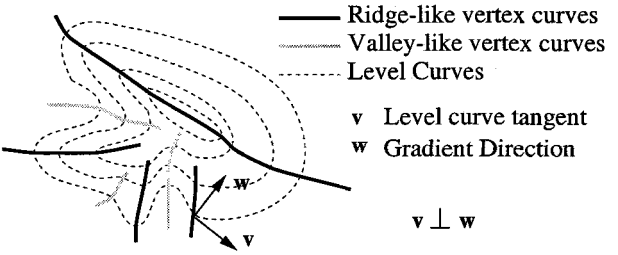

Fig. 1. Vertex curves.

The extension of κ to 3–d is two times the Mean curvature κ_{M} of the level surfaces (see [13] p. 98 and [4] p. 337), which is an extrinsic differential geometric entity. Therefore, from now on we will refer to this extension for d dimensional images as the level set extrinsic curvature (LSEC) κ_d. In [2] we can find the generalization of Eq. (1) to extract r dimensional creases from d dimensional images, by the analysis of the principal curvatures of the level sets.

The computation of (1) or its generalization involves up to fourth order derivatives and the evaluation of complex expressions [5, 2, 10, 14]. However, in many cases, as for elongated structures in 2–d and 3–d images, and plate–like structures in 3–d images, the extrema of curvature are so high that we can circumvent the problem by computing κ or κ_{M} as a creaseness measure and then applying a thresholding. In some applications the creaseness measure itself is sufficient (e.g. as feature for registration of CT/MR brain images [3]). The computation of these creaseness measures needs just up to second order derivatives and their equations are computationally much more cheaper than expressions derived from (1).

In this paper we propose the level curve curvature κ in 2–d, the Mean curvature of the level surfaces κ_{M} in 3–d and, in general, LSEC κ_d as creaseness measures. The level curve curvature has been already proposed as a creaseness measure. However, the traditional way of computing κ [3, 9] gives raise to two problems. Firstly, it produces an extremely large dynamic range of values, but having only a few points with curvature at the upper and lower bounds, which we will call *outliers*. This makes creaseness to differ from medialness since these

outliers are not 'more in the center' than other points with a high, but not outlier, creaseness value. Secondly it produces a discontinuous measure: gaps appear at places where we don't expect any reduction of creaseness for being at the center of some grey–level object.

In Sect. 2 we analyze the problems of computing LSEC by the traditional formula. In Sect. 3 we propose an alternative method to overcome them. In Sect. 4 we go a step further and propose a technique to enhance LSEC creaseness by incorporating the structure tensor analysis in the creaseness measure. As far as we know, it is the first time this technique is employed to compute a creaseness measure. In Sect. 5 we address the main computational aspects. In Sect. 6 we compare the output of our operators with existing creaseness measures. Finally, Sect. 7 summarizes the main conclusions.

2 LSEC Based on the Image Scalar Field

For $L : \mathbf{R}^d \rightarrow \mathbf{R}$ running on $\{x^1, ..., x^d\}$ coordinates, κ generalizes to κ_d according to tensorial calculus as (see [13] p. 98 and [4] p. 337):

$$\kappa_d = (L_\alpha L_\beta L_{\alpha\beta} - L_\alpha L_\alpha L_{\beta\beta})(L_\gamma L_\gamma)^{-\frac{3}{2}}, \quad \alpha, \beta, \gamma \in \{x^1, ..., x^d\} \tag{2}$$

where $L_\alpha = \partial L / \partial \alpha$, $L_{\alpha\beta} = \partial^2 L / \partial \alpha \partial \beta$ and the Einstein summation convention is used. Then, for $d = 2$ and using Cartesian coordinates we obtain the level curve curvature as:

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

and for $d = 3$, two times the Mean curvature of the level surfaces:

$$\begin{aligned} 2\kappa_M = \Big(&2(L_x L_y L_{xy} + L_x L_z L_{xz} + L_y L_z L_{yz}) - L_x^2(L_{yy} + L_{zz}) \\ &- L_y^2(L_{xx} + L_{zz}) - L_z^2(L_{xx} + L_{yy})\Big)(L_x^2 + L_y^2 + L_z^2)^{-\frac{3}{2}} \end{aligned} \tag{4}$$

In 2—d, if we travel along the center of anisotropic structures we go up and down on the relief, passing through maxima, saddles and minima (e.g. Fig. **6(a)**). We have found that computation of κ according to Eq. (3), produces gaps and outliers (Fig. 2) on the creaseness measure around this type of critical points (Fig. **3(c)**), as well as on the center of elongated grey–level objects having a short dynamic range along it. We have checked [7] that this happens independently of the scheme of discretizing derivatives and even when the gradient magnitude $L_{\mathbf{w}} = (L_\gamma L_\gamma)^{1/2}$, for $\mathbf{w} = (L_{x^1}, \ldots, L_{x^d})^{\mathrm{t}}$ being the gradient vector, is far away from the zero of the machine, this is, at pixels where Eq. (3) is well–defined. For example, in the case of ridge–like saddle points, as those in Fig. **6(a)**, κ not only goes down but also suffers from a 'change of sign barrier' on the path of the expected ridge–like curve (Fig. **6(c)**). The reason is that the neighborhood of the saddle point is composed mainly of concave zones ($\kappa < 0$) but the sub–pixel ridge–like curve runs through convex zones ($\kappa > 0$) that discretization of Eq. (3) is not able to detect.

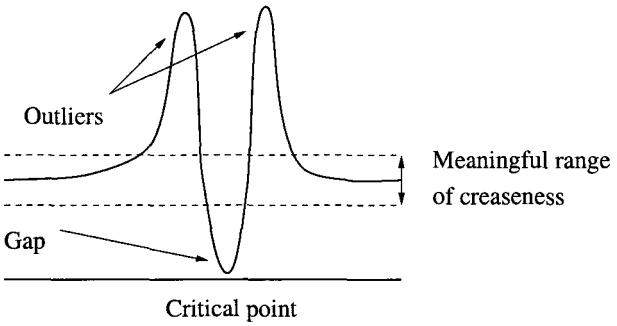

Fig. 2. Profile of κ along a crease which has a critical point on 'its way'.

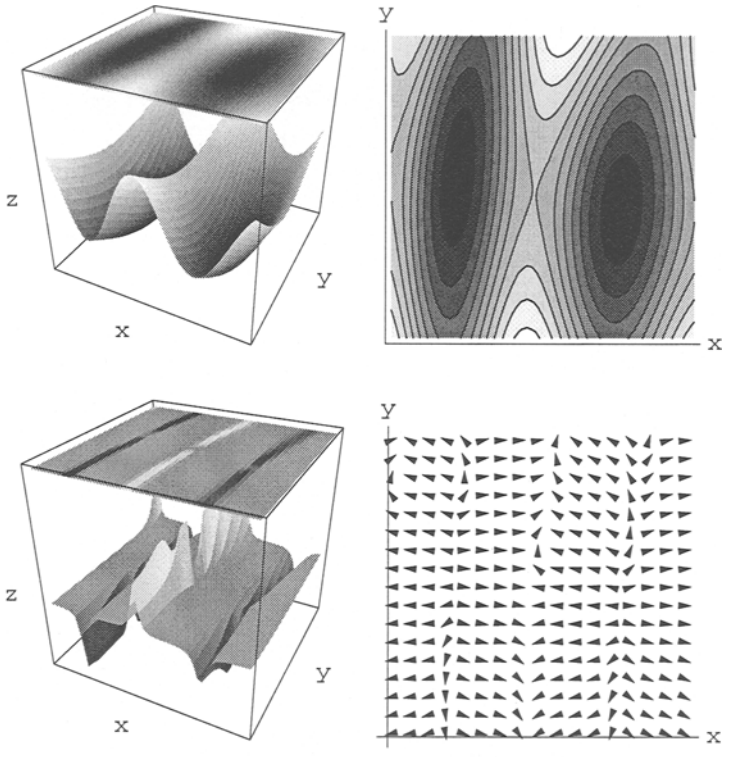

Fig. 3. From top to bottom and left to right: **(a)** Relief with two valley–like regions and a ridge–like one. **(b)** Relief's level curves. **(c)** κ. **(d)** Normalized gradient vector field.

Notice that gaps can not be distinguished locally from points that actually have low creaseness. This affects the use of the creaseness measure itself and the extraction of creases by thresholding the creaseness, since *crease finders* have to decide heuristically how to follow when they reach those discontinuities. To solve the problem of outliers we can use the cut–off transform

$$T(I(\mathbf{x}), g) = \begin{cases} g \text{ if } I(\mathbf{x}) > g \\ -g \text{ if } I(\mathbf{x}) < -g \\ I(\mathbf{x}) \quad \text{Otherwise} \end{cases} \tag{5}$$

for an experimentally given $g > 0$. If g is high we do not see clearly the creaseness measure, if g is low we get a thick response.

The 3–d operator κ_{M} computed according to Eq. (4) has analogous problems to κ in 2–d, as well as other more expensive 3–d creaseness measures as the ridgeness measure $L_{\mathbf{pp}}$ (second directional derivative of L along the direction $\mathbf{p}$ that minimizes the normal curvature of the level surfaces) or the valleyness measure $L_{\mathbf{qq}}$ (in this case $\mathbf{q}$ maximizes the normal curvature) [3, 9].

3 LSEC Based on the Image Gradient Vector Field

To avoid the discretization problems of Eq. (2) when using LSEC as creaseness, we present an alternative way of computing this measure. For the sake of simplicity, consider the 2–d case. Equation (3) is the result of applying the implicit function theorem to a level curve defined as $L(\mathbf{x}) = g$, in order to define its local derivatives and then its curvature κ (see [13] p. 99). Another geometric relationship defines κ through the slopelines, this is, the lines integrating the gradient vector field $\mathbf{w}$, therefore, orthogonal to the level curves. Due to the orthogonality, when level curves are parallel straight lines, slopelines are also parallel and straight, and when the level curves bend, the slopelines diverge (Fig. 4). Therefore, it is clear that there is a connection between curvature of the level curves and the degree of parallelism of the slopelines. In vector calculus we have the divergence operator [12] which measures this degree of parallelism. The divergence of a d dimensional vector field $\mathbf{u} : \mathbf{R}^d \to \mathbf{R}^d, \mathbf{u}(\mathbf{x}) = (u^1(\mathbf{x}), ..., u^d(\mathbf{x}))^{\mathrm{t}}$ is defined as

$$\mathrm{div}(\mathbf{u}) = \sum_{i=1}^{d} \frac{\partial u^i}{\partial x^i} \ . \tag{6}$$

Now, if we denote by $\mathbf{0}_d$ the d dimensional zero vector, then we can define $\bar{\mathbf{w}}$, the normalized gradient vector field of $L : \mathbf{R}^d \to \mathbf{R}$, as

$$\bar{\mathbf{w}} = \begin{cases} \mathbf{w}/\|\mathbf{w}\| \text{ if } \|\mathbf{w}\| > 0 \\ \mathbf{0}_d \quad \text{if } \|\mathbf{w}\| = 0 \end{cases} \tag{7}$$

and then it can be shown that

$$\kappa_d = -\mathrm{div}(\bar{\mathbf{w}}) \ . \tag{8}$$

Equations (2) and (8) are equivalent in the continuous. However, and this is one of the key points of this paper, the discretization of each equation gives different results, namely, Eq. (8) avoids the gaps and outliers that Eq. (2) produces on creases. Thus, from now on, for $d = 2$ we denote by κ and $\bar{\kappa}$ the discrete versions of LSEC according to Eqs. (2) and (8), respectively. Symbols $\kappa_{\mathrm{M}} = \kappa_d/2$ and $\bar{\kappa}_{\mathrm{M}} = \bar{\kappa}_d/2$ for $d = 3$ and κ_d and $\bar{\kappa}_d$ for $d > 3$ will be used analogously.

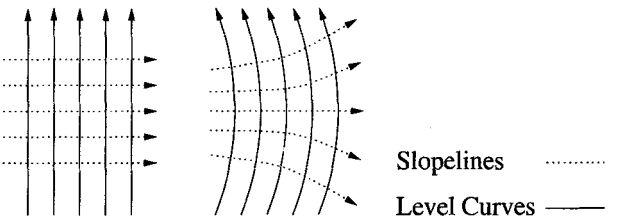

Fig. 4. The divergence of the slopelines depends on the curvature of the level curves.

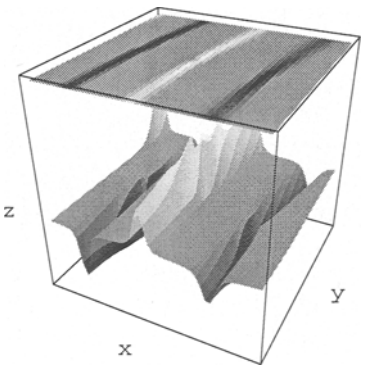

Fig. 5. $\bar{\kappa}$ computed from Fig. 3(**d**).

If we look at the gradient vector field $\bar{\mathbf{w}}$ along a crease we appreciate as dominant the effect of attraction/repulsion of $\bar{\mathbf{w}}$ even in the presence of critical points (Fig. 3(**d**)). This motivated us to compute[1] the level curve curvature from

[1] The definition of κ in terms of the divergence of $\bar{\mathbf{w}}$ is used in other fields of computer vision as in non–linear scale–space. However this is done just to work with a compact notation, but κ is implemented by following Eq. (3) since it gives rise to a most straightforward discretization than using Eq. (8).

the normalized image gradient vector field instead of from the image scalar field L. In Fig. 5 we see the result of computing $\bar{\kappa}$. Notice how gaps are not present anymore, compared with Fig. 3(**c**). Furthermore, the dynamic range of $\bar{\kappa}_d$ is better than that of κ_d. Actually, it can be shown [7] that, for a d dimensional image, if we use centered finite differences to take derivatives, then the values of $\bar{\kappa}_d$ run on $[-d, d]$. The value is $-d$ at d dimensional minima and d at d dimensional maxima. The values of κ_d in the discrete case can also be bounded although having a much broader dynamic range that makes easier the existence of outliers. At Fig. 6 it is also analyzed the behavior of κ and $\bar{\kappa}$ around ridge–like saddle points in a synthetic image.

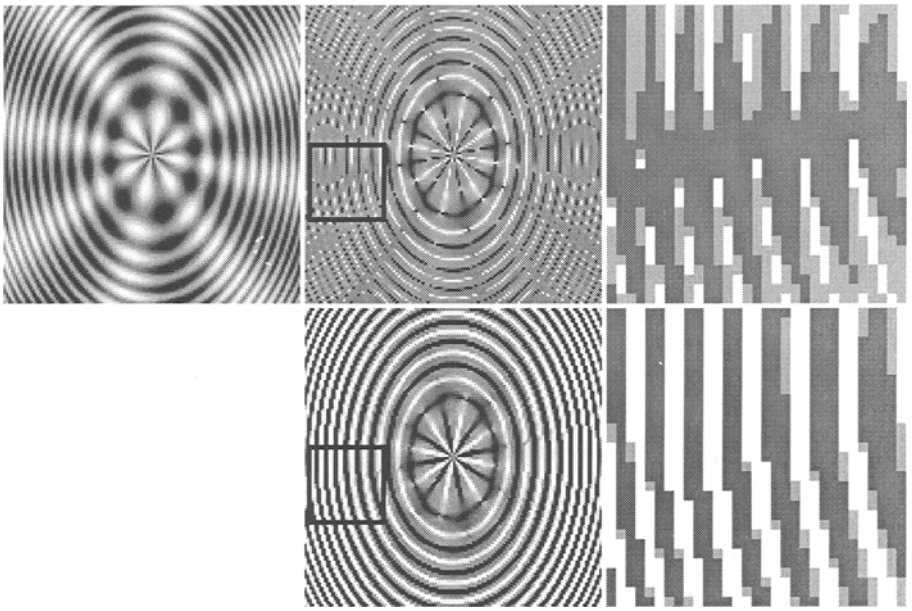

Fig. 6. From left to right: (**a**)$L(x, y) = \sin(60x^2 + 30y^2) + \sin(8 \arctan(y/x))$ sampled in $[-1, 1] \times [-1, 1]$ at a resolution of 128×128 pixels. (**b**) $T(\kappa, 1.0)$ with a region of interest (ROI). (**c**). Zoom of the ROI. White lines: ridge–like creases after thresholding of κ, pixels where $\kappa > 0$ have been seted to lighter grey and $\kappa < 0$ to darker. (**d**) $\bar{\kappa}$. (**e**) Zoom of the previous ROI.

4 LSEC Based on the Image Structure Tensor Field

Once we have established $\bar{\kappa}_d$ as a good creaseness measure, we can go further and enhance it by modifying the gradient vector field of the image previously to apply the divergence operator. We want to filter the gradient vector field in such a way that the configurations of 7(**a**) approach those of Fig. 7(**b**) since

then attraction/repulsion and therefore creaseness, will be higher. At the same time, the qualitative behavior of the gradient vector field at regions where there is neither attraction nor repulsion, must remain unchanged. This filtering can be carried out in a natural way through the *structure tensor*, which is a well–known tool for analyzing oriented textures [1, 6, 11]. Moreover, without extra computational cost, we get a coarse measure of anisotropy that will allow us to attenuate the creaseness measure at zones in which we are not interested, like in flat regions.

Let's fix ideas in 2–d. To compute the dominant orientation, it is sufficient to analyze the behavior of the image gradient in a given neighborhood. We assume that, within a neighborhood of size σ_I centered at a given point $\mathbf{x}$, namely $\mathcal{N}(\mathbf{x}; \sigma_I)$, there is at most a single dominant orientation. Notice that the gradient of a function points towards the direction of maximum change, and the dominant orientation is perpendicular to this direction since anisotropy appears as similar grey values along one orientation and considerable fluctuations perpendicularly.

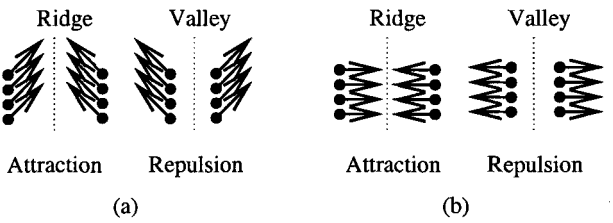

Fig. 7. Attraction and repulsion of vectors in **(b)** are higher than in **(a)**.

The structure tensor is an operator represented by the following symmetric and semi-positive definite $d \times d$ matrix

$$\mathbf{M}(\mathbf{x}; \sigma_I) = \mathcal{N}(\mathbf{x}; \sigma_I) * (\mathbf{w}(\mathbf{x})^t \mathbf{w}(\mathbf{x})) \tag{9}$$

where the convolution '*' of the matrix $(\mathbf{w}(\mathbf{x})^t \mathbf{w}(\mathbf{x}))$ with the window $\mathcal{N}(\mathbf{x}; \sigma_I)$ is element–wise. A suitable choice for the window is a d dimensional Gaussian, i.e. $\mathcal{N}(\mathbf{x}; \sigma_I) = G(\mathbf{x}; \sigma_I)$. This choice implies that neighbors are weighted as a function of their distance.

The dominant gradient orientation is given by the eigenvector which corresponds to the highest eigenvalue of $\mathbf{M}$. However, we have assumed that every point has a preferred orientation. To verify this assumption we introduce a normalized confidence measure: to each orientation we associate a real value $\mathcal{C} \in [0, 1]$ which can be computed from the eigenvalues of the structure tensor. Similarity of the eigenvalues of the structure tensor implies isotropy and, as a result, $\mathcal{C}$ should be close to zero. Therefore, a logical choice consists of testing

whether the sum of quadratic differences in eigenvalues, namely

$$\lambda_{\triangle}(\mathbf{x};\sigma_{\mathrm{I}}) = \sum_{i=1}^{d}\sum_{j=i+1}^{d}(\lambda_i(\mathbf{x};\sigma_{\mathrm{I}}) - \lambda_j(\mathbf{x};\sigma_{\mathrm{I}}))^2, \tag{10}$$

exceeds a predefined threshold c characteristic for $\lambda_{\triangle}$ in the structure we want to enhance. A suitable function is

$$\mathcal{C}(\mathbf{x};\sigma_{\mathrm{I}}) = 1 - e^{-(\lambda_{\triangle}(\mathbf{x};\sigma_{\mathrm{I}}))^2/2c^2} \ . \tag{11}$$

Now we can obtain an enhanced creaseness measure by the following steps:

1. Compute the gradient vector field $\mathbf{w}$ and the tensor field $\mathbf{M}$.

2. Perform the eigenvalue analysis of $\mathbf{M}$. The normalized 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, than $\mathbf{w}$. Then, we obtain the new vector field

$$\tilde{\mathbf{w}} = \mathrm{sign}(\mathbf{w}'^{\mathrm{t}}\mathbf{w})\mathbf{w}' \tag{12}$$

   where the function $\mathrm{sign}(x)$ returns $+1$ if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$. In this way, attraction/repulsion of vectors is reinforced.

3. Compute the creaseness measure

$$\tilde{\kappa}_d = -\mathrm{div}(\tilde{\mathbf{w}}) \ . \tag{13}$$

4. Compute the confidence measure $\mathcal{C}$, tuning the constant c in Eq. (11) to reduce creaseness in the structures we are not interested, in order to take $\tilde{\kappa}_d\mathcal{C}$ as the final creaseness measure.

5 Computational Aspects

To obtain derivatives of a discrete image L in a well–posed manner [13, 4], we use the centered finite differences (CFDs) of a Gaussian smoothed version of the image:

$$L_{\alpha}(\mathbf{x};\sigma_{\mathrm{D}}) \approx \Delta_{\alpha}(L(\mathbf{x}) * G(\mathbf{x};\sigma_{\mathrm{D}})), \quad \alpha \in \{x^1, ..., x^d\} \tag{14}$$

where σ_{D} stands for the standard deviation of the Gaussian and Δ_{α} for the CFD along the α axis. Then, a method to calculate both κ and $\bar{\kappa}$ in 2–d, and κ_{M} and $\bar{\kappa}_{\mathrm{M}}$ in 3–d consists of computing the set of image derivatives and then applying the respective equations. In this case, $\bar{\kappa}$ and $\bar{\kappa}_{\mathrm{M}}$ require less memory and operations than κ and κ_{M} as can be seen in Table 1. However, in 3–d it is convenient to write an algorithm that scans the image voxel by voxel computing the respective expression. The reason is saving memory: by first computing all

Table 1. Number of operations at each pixel/voxel of a 2–d/3–d image. Addition and subtraction are assumed as equivalent, product and division too. The 1–d 1st order CFDs account as 1 addition plus a division and 2nd order CFDs as 3 additions and 1 division.

	κ	$\bar{\kappa}$	κ_M	$\bar{\kappa}_M$
Maximum number of images simultaneously in memory	6	5	9	7
Additions and subtractions	15	6	33	10
Products and divisions	8	4	13	6
Square roots	1	1	1	1
Divisions by a constant	5	4	9	6

the image derivatives involved in $\bar{\kappa}_M$ we need simultaneously 7 float 3–d images (see Table 1 and more details in [7]) which could mean a lot of memory. We have adopted the pixel/voxel scanning approach to minimize memory requirements and therefore disk throughput.

When running pixel/voxel by pixel/voxel computing $\bar{\kappa}$ or $\bar{\kappa}_M$ we have to buffer values to avoid the repetition of calculations [7]. In practice this makes $\bar{\kappa}$ and $\bar{\kappa}_M$ slightly more time consuming than κ and κ_M. Yet the difference is small as can be seen in Table 2. Computation of $\tilde{\kappa}_d$ consumes much more resources than κ_d and $\bar{\kappa}_d$ due to both the averaging of vectors and the eigensystem analysis.

Table 2. Time consuming in a 200 MHz Pentium Pro PC with 64MB of RAM memory under Linux OS. The swapping was needed only for the 250 × 250 × 180 image. All the images are float since the operators are applied after a Gaussian smoothing, which was implemented by separable spatial convolutions but without exploiting Gaussian symmetry.

Image dimensions	Gaussian smoothing ($\sigma_D = 4.0$)	κ	$\bar{\kappa}$	$\tilde{\kappa}$ ($\sigma_I = 4.0$)
256 × 256	0.3 s	0.058 s	0.072 s	1.2 s
512 × 512	1.3 s	0.24 s	0.28 s	5.3 s
		κ_M	$\bar{\kappa}_M$	$\bar{\kappa}_M$ ($\sigma_I = 4.0$)
128 × 128 × 84	50 s	1.8 s	2.1 s	50 s
250 × 250 × 180	360 s	23 s	23.3 s	720 s

6 Results

Results are presented within the framework that motivated our research in creaseness operators. It consists of the automatic registration of 3–d CT and MR brain volumes from the same patient as in [3]. This can be achieved by

registering the surface that runs along the center of the skull from both images, which appears as a ridge–like structure in the CT image and a valley–like structure in the MR. The idea is to base the registration on a search through the space of the six 3–d rotation and translation parameters, using the correlation of the MR valleyness and the CT ridgeness to assess the goodness of each transform. Therefore we need continuous creaseness measures, with the creaseness in the skull bone enhanced.

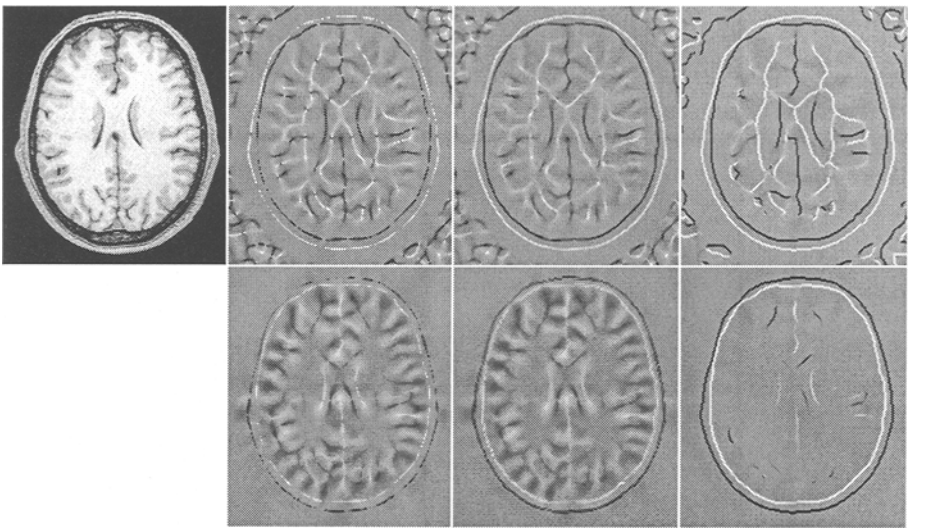

Fig. 8. From top to bottom and left to right: **(a)** 256×256 MR slice. The black fringe (valley–like structure) is the skull bone. In all cases we work with a Gaussian smoothed version of this image, for $\sigma_{\mathrm{D}} = 4.0$ pixels. **(b)** $T(\kappa, 1.5)$ (see the cut–off transform T in Eq. (5)). **(c)** $\bar{\kappa}$. **(d)** $\tilde{\kappa}$ for $\sigma_{\mathrm{I}} = 4.0$ pixels. **(e)** The well–known [3] operator $L_{\mathbf{vv}} = -\kappa L_{\mathbf{w}}$. **(f)** $-\bar{\kappa} L_{\mathbf{w}}$. **(g)** $-\tilde{\kappa} \mathcal{C}$ for $c = 700$.

For 2–d data we obtain the results shown in Figs. 8 and 9. It is clearly distinguished that κ produces a number of gaps along the skull while $\bar{\kappa}$ and $\tilde{\kappa}$ give a continuous response. The $\tilde{\kappa}$ measure is more contrasted than $\bar{\kappa}$ and, when combined with the corresponding confidence measure $\mathcal{C}$, we can almost isolate creaseness on the skull (white response in Fig. 8**(g)** and black in 9**(g)**).

As the application is on 3–d data sets, the operators that we are actually using are $\bar{\kappa}_{\mathrm{M}}$ and $\tilde{\kappa}_{\mathrm{M}}$. Traditional operators such as $L_{\mathbf{pp}}$ and $L_{\mathbf{qq}}$ also have problems of continuity as can be seen in [3] at p. 386, where the authors used the same dataset. In Fig. 10 we can see the CT ridgeness and MR valleyness based on $\bar{\kappa}_{\mathrm{M}}$. Notice how $\bar{\kappa}_{\mathrm{M}}$ is quite continuous along the skull. However, response is also present within the brain. It is difficult to get rid of this response multiplying by the gradient magnitude $L_{\mathbf{w}}$ since we have then the same effect than in the 2–d case, this is, creaseness along the skull loss continuity as in Figs. 8**(f)** and

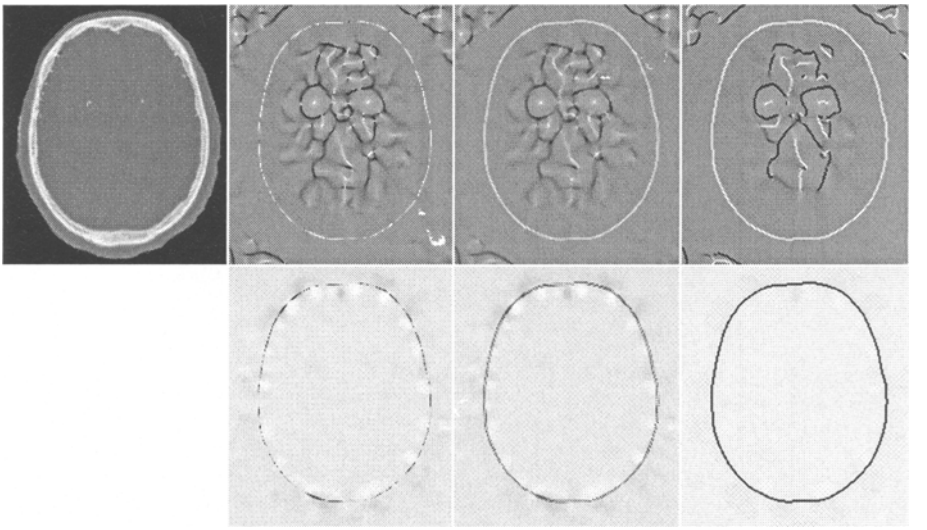

Fig. 9. From top to bottom and left to right: **(a)** 256×256 CT slice. The white fringe (ridge–like structure) is the skull bone. **(b)**–**(g)** have the same meaning and share the parameters of Fig. 8.

9(f) because along the center of the skull, the gradient magnitude is lower than in the skull boundary. In Fig. 11 we see the results based on the $\tilde{\kappa}_M$ operator multiplied by the confidence measure $\mathcal{C}$. As in the 2–d case, the $\tilde{\kappa}_M$ operator itself gives a more contrasted creaseness measure as well as more homogeneous along the crease, and by its combination with $\mathcal{C}$ we get rid of the creaseness response in the background and within the brain.

7 Discussion

The level set extrinsic curvature (LSEC) is a creaseness measure that acts as a medialness measure for grey–level objects. In this paper, we first identified the problem of outliers and, mainly, gaps when computing this creaseness measure by classical schemes. Afterwards we have proposed an alternative method of computing LSEC that overcomes the mentioned problems. Moreover, the use of LSEC in 3–d ($\bar{\kappa}_M$) as a creaseness measure is itself another key point of this paper since previous works went to 3–d on the basis of most computationally expensive principal curvature analysis of the level surfaces. Moreover, these measures suffer also from the problem of gaps while $\bar{\kappa}_M$ does not. We have also proposed a new version of LSEC where the structure tensor analysis has been adapted from oriented texture analysis to enhance LSEC creaseness, which combined with the associated confidence measure gives a very clean creaseness measure along interesting objects. Results have been shown in the context of a real application,

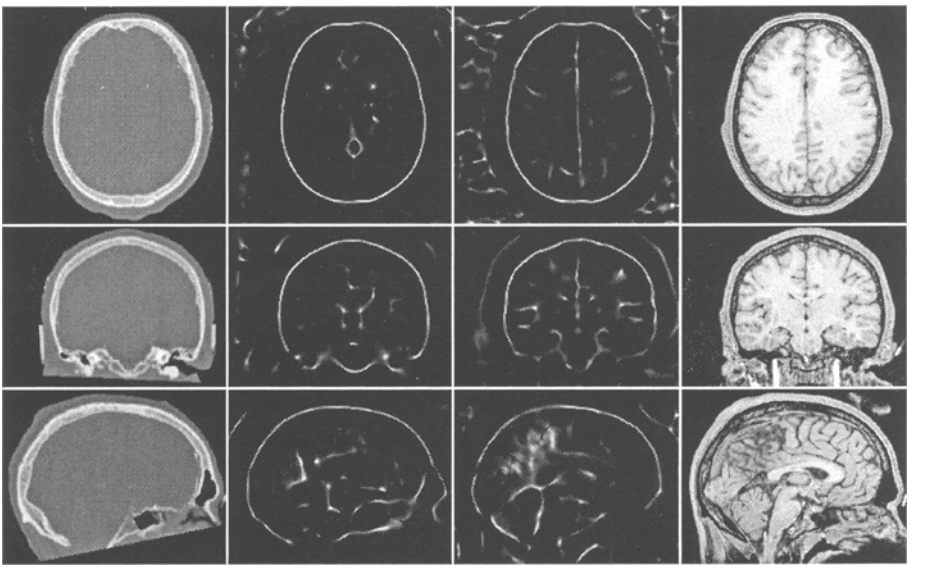

Fig. 10. Columns from left to right: **(a)** Transversal, coronal and sagital slices of a $250 \times 250 \times 180$ CT image with cubic voxels. **(b)** $\bar{\kappa}_{\mathrm{M}}$ of the CT image after a Gaussian smoothing of $\sigma_{\mathrm{D}} = 4.0$ voxels. **(c)** $\bar{\kappa}_{\mathrm{M}}$ of the MR image at the same scale. **(d)** Same slices of a $250 \times 250 \times 180$ MR image with cubic voxels.

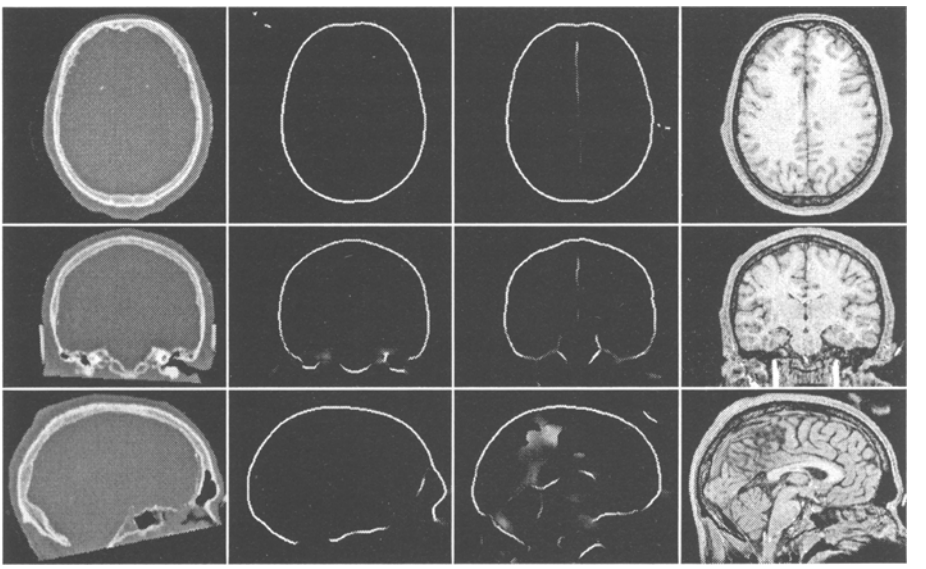

Fig. 11. All the columns have the same meaning than in Fig. 10 but showing the results of $\tilde{\kappa}_{\mathrm{M}}\mathcal{C}$ for $\sigma_{\mathrm{D}} = \sigma_{\mathrm{I}} = 4.0$ voxels and $c = 1000$.

namely, the registration of CT and MR brain volumes, where our operators have proved to give an excellent output.

Acknowledgments

This research has been partially funded by Spanish CICYT projects TIC97–1134–C02–02 and TAP96–0629–C04–03. The authors acknowledge also Dr. Petra van den Elsen from the 3D–CVG at the Utrecht University for providing us two 3–d CT and MR datasets.

References

1. J. Bigun, G. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.
2. D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4:353–373, 1994.
3. P. van den Elsen, J. Maintz, E-J. Pol, and M. Viergever. Automatic registration of CT and MR brain images using correlation of geometrical features. *IEEE Trans. on Medical Imaging*, 14:384–396, 1995.
4. L. Florack, B. ter Haar Romeny, J. Koenderink, and M. Viergever. Cartesian differential invariants in scale–space. *Journal of Mathematical Imaging and Vision*, 3:327–348, 1993.
5. J. Gauch and S. Pizer. Multiresolution analysis of ridges and valleys in grey–scale images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:635–646, 1993.
6. B. Jahne. *Spatio–temporal image processing*, volume 751 of *Lecture Notes in Computer Science*, chapter 8, pages 143–152. Springer–Verlag, 1993.
7. A. López, F. Lumbreras, and J. Serrat. Efficient computing of local creaseness. Technical Report 15, Computer Vision Center, campus of the UAB. Spain., 1997.
8. A. López and J. Serrat. Ridge/valley–like structures: creases, separatrices and drainage patterns. Technical Report 21, Computer Vision Center, campus of the UAB. Spain., 1997.
9. J. Maintz, P. van den Elsen, and M. Viergever. Evaluation of ridge seeking operators for multimodality medical image matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:353–365, 1996.
10. O. Monga and S. Benayoun. Using partial derivatives of 3d images to extract typical surface features. *Computer Vision and Image Understanding*, 61:171–189, 1995.
11. W. Niessen, A. López, W. Van Enk, P. Van Roermund, B. ter Haar Romeny, and M. Viergever. In vivo analysis of trabecular bone architecture. In J. S. Duncan and G. Gindi, editors, *Information Processing and Medical Imaging*, volume 1230 of *Lecture Notes in Computer Science*, pages 435–440, 1997.
12. H. M. Schey. *DIV, GRAD, CURL and all that.* W. W. Norton & Company, 1973.
13. B. ter Haar Romeny and L. Florack. A multiscale geometric model of human vision. In W. Hendee and P. Well, editors, *The Perception of Visual Information*, pages 73–114. Springer–Verlag, 1993.
14. J. P. Thirion and A. Gourdon. Computing the differential characteristics of isointensity surfaces. *Computer Vision and Image Understanding*, 61:190–202, 1995.