Chapter 4

Geometric Features

4.1 Introduction

Features in a sample of subjects are frequently required. This could be for statistical analysis or simply to classify and process a geometry. Although this project is aimed at performing elastic surface registration to deform a generic shape into a target geometry, a better result is expected if a more rigorous approach is implemented for common feature registration. For that reason, the extraction of surface and feature information on a computational domain is investigated in this chapter.

Focus is first given to the local structure tensor. This is mainly used in inhomogeneous mesh coarsening and smoothing. This is done to best preserve the features in a given mesh during this operation. The detection of feature points and crest lines on a surface mesh is then investigated from methods that employ differential geometry concepts. Thresholding is also discussed. This is done after feature extraction to filter out insignificant feature lines.

Likely feature surfaces can be extracted from the local structure tensor analysis. If a local smooth surface approximation or discrete differential geometry operators are applied to extract feature lines, focus could be given to only these areas. In doing so far less false crest lines are already noticeable, reducing the need for additional thresholding and also speeding up computation if the local structure tensor result is already available.

After focusing on feature extraction in this chapter, later chapters will use the extracted features to determine feature correspondence during registration. Feature recognition could aid in matching selected features while disregarding others. Fea-

tures that can't be matched due to possible topological[1] inconsistencies could also be automatically disregarded.

4.2 Local Structure Tensor

Given a tessellated surface mesh, the unit normal of each triangle can be obtained with very little effort. The accepted standard dictates triangle connectivity defined in an anti-clockwise manner for positive normal direction, often referred to as the right hand rule. Keeping this standard in mind, a triangle's unit normal vector may be calculated.

Two vectors are first obtained using the three vertices that make up the triangle. Both vectors have their origin at one of the triangle vertices and are in the direction of the other two respectively. The normalised result of the cross product between these vectors is the triangle unit normal.

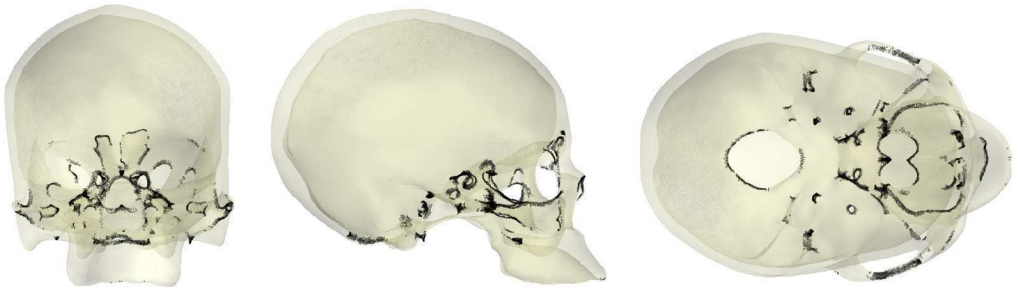
When aimed at recognising localised surface features, the change in surface gradient in a specific area is inspected. Surface gradient information can be captured by the local structure tensor [71]. In order to determine the local structure tensor at a specific vertex $\mathbf{p}$, given by

$$LST(\mathbf{p}) = \sum_{i \in N-\text{neighbours}} \begin{bmatrix} n_x n_x & n_x n_y & n_x n_z \\ n_y n_x & n_y n_y & n_y n_z \\ n_z n_x & n_z n_y & n_z n_z \end{bmatrix}_i , \tag{4.1}$$

the normal of each i^{th} neighbouring vertex, $\begin{pmatrix} n_x & n_y & n_z \end{pmatrix}_i$ also needs calculation. This is done by defining the vertex unit normal vector as the weighted average of the normals of all its incident triangles. The weight attributed to each triangle is determined using the distance from the vertex to the triangle centroid and triangle area. The unit normal of each triangle is multiplied with the area of the triangle and divided by the distance to the triangle centroid. The result obtained after adding all weighted triangle normals incident to the vertex is normalised to approximate the vertex unit normal.

The local structure tensor is defined as the summation of the matrices corresponding to the neighbours of a local vertex. This simple summation allows for

[1] Two structures are topologically similar if one can be completely morphed into the other and one-to-one correspondence remains continuous.



(a) (b) (c)

Figure 4.1: Points on skull geometry where $\lambda_1 < 5 \times \lambda_2$. This condition represents spheres, saddles, ridges and valleys within a certain degree. (a) Frontal, (b) side and (c) bottom view.

(a) (b) (c)

Figure 4.2: Points on skull geometry where $\lambda_1 < 100 \times \lambda_2$. This condition represents spheres, saddles, ridges and valleys within a certain degree. (a) Frontal, (b) side and (c) bottom view.

neighbour contributions to be evaluated in arbitrary order. It is also worth noting that the neighbours considered may be more than only those incident to the local vertex. Additional layers of neighbouring nodes may also be included depending on the size of local features compared to the size of the elements of the tessellated surface mesh. Filtering out the effect of surface noise by using additional layers of neighbouring vertices also proves helpful.

4.2.1 Feature Classification

The eigenvalues and eigenvectors of the local structure tensor indicates the overall distribution of vectors in a local vertex neighbourhood [71]. The features are categorised locally into different types namely spheres and saddles, lines and planes.

Considering the evaluation of features within a surface mesh, let the eigenvalues

(a) (b) (c)

Figure 4.3: Points on skull geometry where $\lambda_1 > 2'000 \times \lambda_2$. This condition represents planes to a certain degree. (a) Frontal, (b) side and (c) bottom view.

(a) (b) (c)

Figure 4.4: Points on skull geometry where $\lambda_1 > 20'000 \times \lambda_2$. This condition represents planes to a certain degree. (a) Frontal, (b) side and (c) bottom view.

of the local structure tensor be λ_1, λ_2 and λ_3 with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The following cases may be considered:

- Spheres and saddles: $\lambda_1 \approx \lambda_2 \approx \lambda_3 > 0$.
- Ridges and valleys: $\lambda_1 \approx \lambda_2 \gg \lambda_3 \approx 0$.
- Planes: $\lambda_1 \gg \lambda_2 \approx \lambda_3 \approx 0$.

Figures 4.1 through 4.4 show results that could be extracted after performing a local structure tensor analysis on the unstructured surface mesh representation of a human skull. For the intended study the intricate details and structures within the nasal cavity and sinuses aren't of statistical importance. In these areas unwanted detail was removed. After then simplifying and smoothing the mesh, vertex normals were determined and 3 layers of neighbouring nodes were used to extract local feature information.

Figures 4.1 and 4.2 show various thresholds of areas that could be considered feature rich. These areas could be included into the classification for spheres, saddles, ridges and valleys. Vertices are displayed here where the eigenvalues satisfy the conditions $\lambda_1 < 5 \times \lambda_2$ and $\lambda_1 < 100 \times \lambda_2$ respectively.

The third eigenvalue $\lambda_3 \leq \lambda_2$ is only required to differentiate spheres and saddles from ridges and valleys. Very few of the features in a human skull geometry can be classified as spherical or saddle points compared to the quantity of ridges and valleys. The intended use of the local structure tensor is to simply classify areas as feature rich or flat. Because of this, no distinction is made between spheres, saddles, ridges and valleys in Figures 4.1 and 4.2.

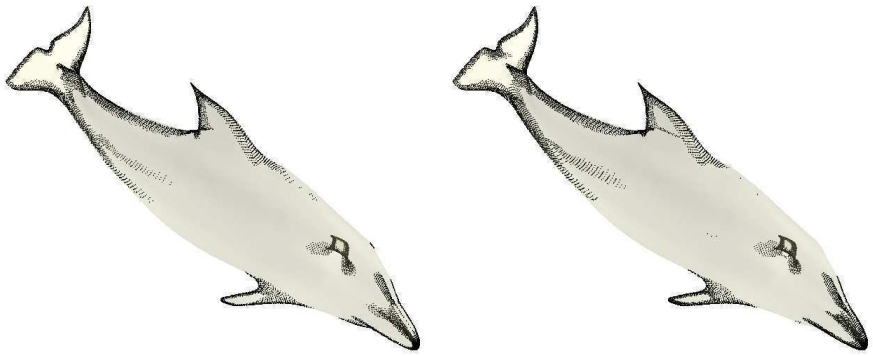
All nodes satisfying the conditions $\lambda_1 > 2'000 \times \lambda_2$ and $\lambda_1 > 20'000 \times \lambda_2$ are shown in Figures 4.3 and 4.4. Here, areas are highlighted to illustrate what could be considered flat or featureless, showing the third case mentioned above for planes.

4.2.2 Spatial Search Speed-up

Because evaluating mesh connectivity is imperative when using N-ring neighbours, it is often possible and beneficial to rather use a user specified amount of nearest neighbouring points. For each mesh node, the spatial domain could simply be evaluated with a nearest neighbour search algorithm for the specified number of closest neighbours. This eliminates the need for mesh connectivity information beyond determining unit normal orientation, producing a significant speed-up in local structure calculation.

A drawback to this proposed method is that it is possible to pick up nodes on opposing surfaces when only the spatial node distribution is evaluated. This happens when different surfaces or parts of a surface are close to one another, preventing successful use of this procedure. The intricate details of a human brain geometry with all it's folds for example could pose a problem in certain areas. This could therefore result in describing areas as feature rich that are in actual fact smooth but simply close to opposing surfaces. Depending on the intended use of the local structure tensor analysis, it is up to the user's discretion to determine the required accuracy.

In case these problems occur in the geometry, only the neighbouring unit normals that correlate to a certain degree could be used in evaluating the local structure tensor. Alternatively, the N-ring procedure should be employed when significant



(a) (b)

Figure 4.5: Points for $\lambda_1 < 50 \times \lambda_2$ on a refined dolphin model using (a) 3-ring neighbourhood (b) 15 nearest neighbours

differences between the local and nearest neighbour unit normals are a possibility.

In Figure 4.5, the points satisfying $\lambda_1 < 50 \times \lambda_2$ is shown on a dolphin geometry. This condition combines the points that are closely related to spheres, saddles, ridges and valleys. The local structure tensor procedure is first done using a 3-ring neighbourhood and then a 15 nearest neighbour search using the k-d tree spatial search algorithm. Significant speed-up is obtained with very little difference in the final reported areas of interest.

The obtained area of interest depends on the support used. If a 5-ring neighbourhood is used, or 20 nearest neighbours for example, the reported area would be larger while smaller support would result in picking up very localised feature areas.

4.3 Differential Geometry Surface Information

For a smooth oriented surface $\mathcal{S}$, the maximal and minimal curvatures are such that $\kappa_{\max} \geq \kappa_{\min}$ with corresponding principal directions $\boldsymbol{\varpi}_{\max}$ and $\boldsymbol{\varpi}_{\min}$. The curvature derivatives along these directions are then $\tau = \partial\kappa/\partial\boldsymbol{\varpi}$, defined locally in the neighbourhood of non-umbilical points [45].

Points on a surface where $\kappa_{\max} = \kappa_{\min}$ are called umbilical points. Principal directions are not defined at these points, making it impossible to determine curvature derivatives. Closure of these points where either one of the curvature derivatives vanishes are however required to form ridges and valleys.

Concave and convex crest lines on a smooth surface are dual with respect to surface normal orientation. With a normal defined consistently, these lines could

be extracted and classified as ridges and valleys on the surface. Concave crest lines are expected where points on the line satisfy the conditions: [45]

$$\tau_{\max} = 0 \qquad \partial\tau_{\max}/\partial\boldsymbol{\varpi}_{\max} < 0 \qquad \kappa_{\max} > |\kappa_{\min}|\,, \tag{4.2}$$

while points on a convex crest line satisfy the following conditions:

$$\tau_{\min} = 0 \qquad \partial\tau_{\min}/\partial\boldsymbol{\varpi}_{\min} < 0 \qquad \kappa_{\min} < -|\kappa_{\max}|\,. \tag{4.3}$$

4.3.1 Application to a Discretised Surface

Numerous techniques have been proposed to extract curvature information from a discretised surface representation. Some techniques approximate an implicit smooth surface either globally [21, 45] or locally [21, 37, 70]. Other techniques approximate surface information through the application of discrete curvature operators [33, 40]. For the purpose of this report, focus is given to the application of an approximate smooth local implicit surface.

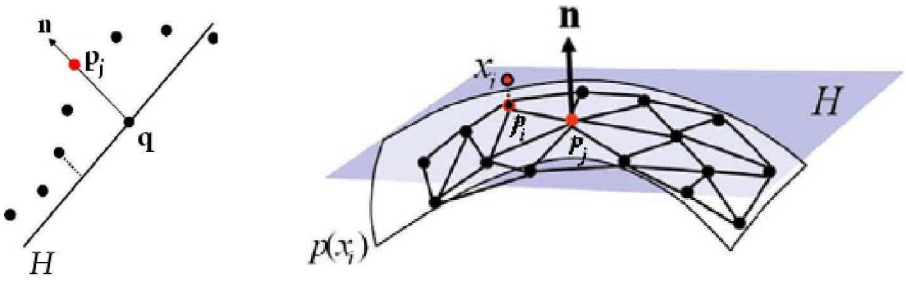
Given a surface in the implicit form $\mathcal{F}(\mathbf{x}) = 0$, $\mathbf{x} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}$, the principal curvatures and associated curvature directions can be obtained at a specific point on the surface from an eigen-analysis on $\nabla\mathbf{n}$. The unit normal $\mathbf{n}$ at this point on surface $\mathcal{F}$ is taken as $\nabla\mathcal{F}/|\nabla\mathcal{F}|$ [21, 37, 40, 45, 70]. This means that the eigenvalues of matrix $\nabla^2\mathcal{F}/|\nabla\mathcal{F}|$ would be the principal curvatures while the associated eigenvectors are the principal curvature directions.

The curvature derivative or extremality coefficient for a given principal curvature and associated curvature direction is given by [70]

$$\tau = \frac{\partial\kappa}{\partial\boldsymbol{\varpi}} = \frac{\mathcal{F}_{ijk}\varpi_i\varpi_j\varpi_k + 3\kappa\mathcal{F}_{ij}\varpi_i n_j}{|\nabla\mathcal{F}|}. \tag{4.4}$$

In this equation, summation over repeated indices are implied with $\mathcal{F}_{ij}$ and $\mathcal{F}_{ijk}$ indicating the second and third partial derivatives of $\mathcal{F}(\mathbf{x})$. Choosing a local coordinate frame $\mathbf{u} \times \mathbf{v} = \mathbf{w}$ in such a way that the origin is situated on the surface and $\mathbf{w} = \mathbf{n}$, the surface approximation could be given in the form of a bi-variate polynomial

$$p(u,v) = \frac{1}{2}\left(b_0u^2 + 2b_1uv + b_2v^2\right) + \frac{1}{6}\left(c_0u^3 + 3c_1u^2v + 3c_2uv^2 + c_3v^3\right) + \ldots \tag{4.5}$$



(a) (b)

Figure 4.6: (a) The MLS projection. Given a point $\mathbf{p}_j$, the local reference plane H and a point $\mathbf{q}$ on H is first determined. An MLS surface is then defined implicitly by the MLS projection as the set of points that project on to themselves [37]. (b) The estimation of a modified MLS approximation to the mesh [37].

In this local coordinate frame, Equation (4.4) is simplified to become [70]

$$\tau = \frac{\partial \kappa}{\partial \boldsymbol{\varpi}} = \mathcal{F}_{ijk}\varpi_i\varpi_j\varpi_k. \tag{4.6}$$

A moving least squares approximation is chosen to construct a local smooth surface approximation. The required surface information is approximated and then applied to the original mesh nodes. This is discussed in the following subsection while procedures followed to extract feature points using shape index and extracting ridge and valley lines are discussed thereafter.

4.3.2 Enhanced Moving Least Squares Approximation

A moving least squares (MLS) surface is implicitly defined as the set of points that project on to themselves under the MLS projection. The procedure to approximate this surface involves first determining an appropriate local reference plane and then fitting a bi-variate polynomial to points projected on this reference plane in a least squares manner [37, 70].

Generating a smooth local surface approximation

Given an unstructured surface mesh $\mathcal{M}$ approximating the smooth surface $\mathcal{S}$ in $\mathbb{R}^3$, let $\mathbf{p}_j$ be a point on the mesh with $\{\mathbf{p}_i\}_{i\in N-\text{neighbours}}$ the set of points in its N-ring neighbourhood. Only the neighbours whose normals don't make obtuse

angles with that of the center node are selected and kept in the $N-$neighbours set [70]. Alternatively, S. Kim and C. Kim [37] suggest fitting a weighted least squares approximation with a Gaussian weighting function that lends higher support to point within in the 1-ring neighbourhood of $\mathbf{p}_j$.

The following general procedure is followed to approximate a bi-variate polynomial surface for curvature, curvature derivatives and principal curvature direction estimation [37, 70]:

- The vertex unit normal $\mathbf{n}_j$ for each point $\mathbf{p}_j$ in the mesh is determined as the weighted average of the neighbouring face normals. The weighting is done in the same way explained under the local structure tensor section, using triangle areas and the distance from the point to the incident triangle centroids.

- A local reference plane $H = \{\mathbf{p}_j | \langle \mathbf{n}, \mathbf{p}_j \rangle = 0, \mathbf{p}_j \in \mathbb{R}^3\}$ is determined through the point $\mathbf{p}_j$ with the same unit normal $\mathbf{n} = \mathbf{n}_j$. In Figure 4.6 (a) this is illustrated should point $\mathbf{q}$ be moved to be exactly point $\mathbf{p}_j$.

- The orthogonal projections $\{\mathbf{x}_i\}_{i \in N-\text{neighbours}}$ and $\{f_i\}_{i \in N-\text{neighbours}}$ perpendicular distance of the neighbouring points onto and from the reference plane are determined.

- A local orthonormal coordinate system is set up with origin at $\mathbf{p}_j$. Using the plane normal direction as $\mathbf{w}$, the local reference coordinate directions are determined. The rotation required from global direction $\mathbf{z}$ to $\mathbf{w}$ is first determined and then applied to the global coordinate system. This is done in such a way that the local coordinate system is consistently defined for all points on the mesh.

- Orthogonal projections of the N-ring neighbours, $\mathbf{x}_i$ can then be rewritten in terms of its components in the two directions $\mathbf{u}$ and $\mathbf{v}$ in the plane.

- A local smooth surface approximation is finally made as a third degree polynomial $p(\mathbf{x}_i)$ that minimises the least squares error

$$\sum_{i \in N-\text{neighbours}} (p(\mathbf{x}_i) - f_i)^2 . \tag{4.7}$$

An example of an estimated MLS surface constructed by fitting a polynomial through neighbouring points in a least squares manner is visible in Figure 4.6 (b).

The same speed up can be obtained as in subsection 4.2.2 by applying a spatial nearest neighbour search algorithm. Without the need to evaluate mesh connectivity, a simple search for N nearest neighbours is applied. Again only those neighbours with sufficient similarity between unit normals are used in fitting the local least squares approximation. This step is already a requirement in doing the enhanced least squares fit.

Estimating curvatures and their derivatives

After fitting a local bi-variate third degree polynomial approximated surface $z = p(\mathbf{x}_i)$, the principal curvatures, curvature derivatives and principal curvature directions in the local coordinate frame can be obtained. This is done by applying the principles of differential geometry to the resulting implicit surface.

The principal curvatures and associated curvature directions can be obtained from an eigen-analysis on $\nabla\mathbf{n}$ at point $\mathbf{p}_i$ of a surface in the form of Equation (4.5). Noting that in this particular case point $\mathbf{p}_i$ is at the local origin, the solution to the eigenvalue problem

$$\begin{bmatrix} b_0 & b_1 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} \varpi_1 \\ \varpi_2 \end{bmatrix} = \kappa \begin{bmatrix} \varpi_1 \\ \varpi_2 \end{bmatrix}, \tag{4.8}$$

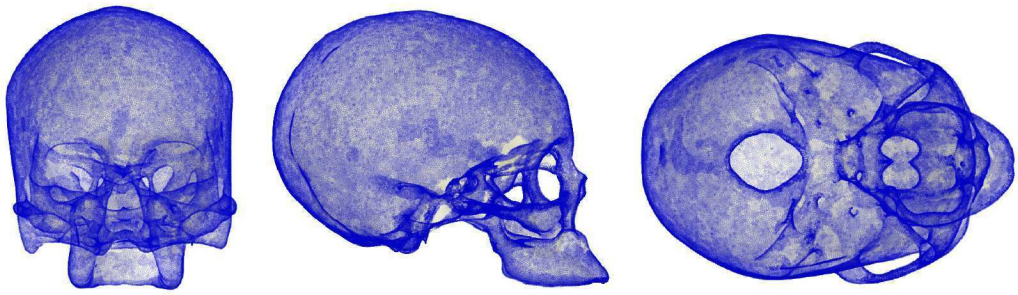
produces the principal curvatures $\kappa_{\max}$ and $\kappa_{\min}$ with associated principal curvature directions $\boldsymbol{\varpi}_{\max}$ and $\boldsymbol{\varpi}_{\min}$ in the local coordinate frame. The curvature derivatives can then be calculated from Equation (4.6) as [70]

$$\tau = \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \end{bmatrix}^T \begin{bmatrix} c_0 & c_1 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \varpi_1 \\ \varpi_2 \end{bmatrix}. \tag{4.9}$$

Converting the principal curvature directions into the global coordinate frame is also done. This is required to help determine the presence of ridge and valley nodes as well as to connect these nodes into lines.

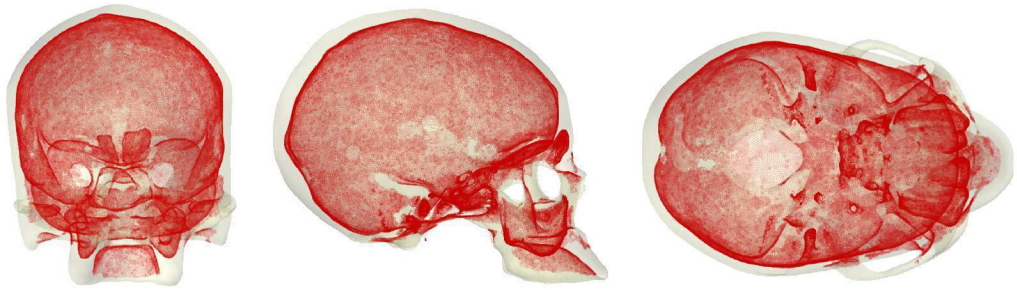
The principal curvatures determined at each point may be used to classify areas and points on the geometry. The MLS procedure is performed on the skull geometry used to generate Figures 4.1 through 4.4. Curvature information is extracted and used to display areas on the geometry that are concave and convex. Figure 4.7 represents the concave areas and Figure 4.8 the convex areas of the skull geometry.

Figures 4.9 through 4.11 illustrate ridge and valley areas on the skull geometry. Only points within some degree of the average curvature evaluated is displayed.



(a) (b) (c)

Figure 4.7: Points on skull geometry where $\kappa_{\max} > |\kappa_{\min}|$. (a) Frontal, (b) side and (c) bottom view.



(a) (b) (c)

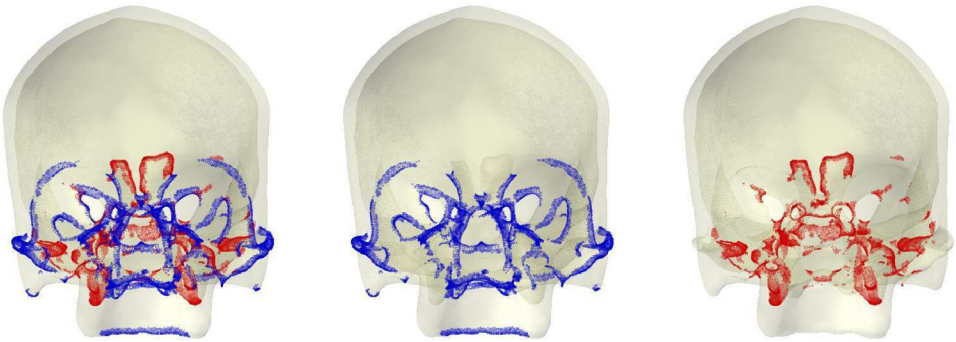
Figure 4.8: Points on skull geometry where $\kappa_{\min} < -|\kappa_{\max}|$. (a) Frontal, (b) side and (c) bottom view.

When using the local structure tensor to evaluate and find feature rich areas on a geometry it is impossible to differentiate between ridges and valleys. Obtaining differential geometry information on the other hand allows one to make this distinction.

4.3.3 Shape Index Feature Points

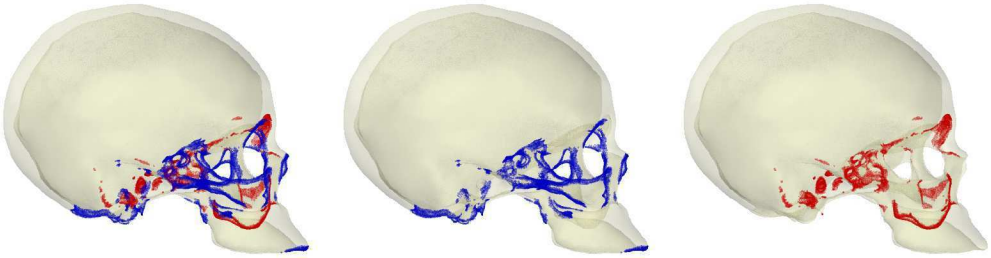
Feature points can be automatically extracted from differential geometry information such as that obtained after performing the MLS procedure. These feature points are defined in areas where large shape variation can be obtained by a calculated shape index.

The shape index S_i is a quantitative measure of the shape of a surface point $\mathbf{p}_i$



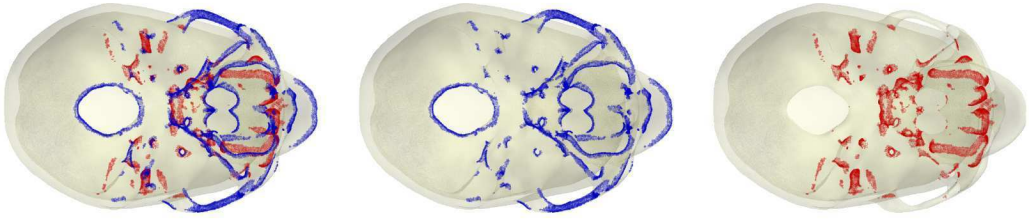
(a) (b) (c)

Figure 4.9: Frontal view highlighting points on skull geometry where $\kappa_{\max} > 5 \times$ average($\kappa_{\max}$) in blue and $\kappa_{\min} < 5 \times$ average($\kappa_{\min}$) in red. (a) Ridges and valleys split up in (b) to show only ridges and (c) only valleys.



(a) (b) (c)

Figure 4.10: Side view highlighting points on skull geometry where $\kappa_{\max} > 5 \times$ average($\kappa_{\max}$) in blue and $\kappa_{\min} < 5 \times$ average($\kappa_{\min}$) in red. (a) Ridges and valleys split up in (b) to show only ridges and (c) only valleys.



(a) (b) (c)

Figure 4.11: Lower view highlighting points on skull geometry where $\kappa_{\max} > 5 \times$ average($\kappa_{\max}$) in blue and $\kappa_{\min} < 5 \times$ average($\kappa_{\min}$) in red. (a) Ridges and valleys split up in (b) to show only ridges and (c) only valleys.

determined by [22]

$$S_i(\mathbf{p}_i) = \frac{1}{2} - \frac{1}{\pi}\tan^{-1}\left(\frac{\kappa_{\max}(\mathbf{p}_i) + \kappa_{\min}(\mathbf{p}_i)}{\kappa_{\max}(\mathbf{p}_i) - \kappa_{\min}(\mathbf{p}_i)}\right). \tag{4.10}$$

All shapes are mapped into the interval $[0, 1]$ with this definition. Larger shape index values represent convex surfaces while smaller values represent concave surfaces. These shape index values capture the characteristics of shape objects and can be used for feature point extraction. Following the work done by Chen *et al.* [22], a window is drawn on the surface around a candidate feature point $\mathbf{p}_i$ by including points in a sphere of radius r_i. This point is then marked as a feature point if the shape index $S_i(\mathbf{p}_i)$ satisfies:

- $S_i(\mathbf{p}_i)$ =max of shape indexes and $S_i(\mathbf{p}_i) \geq (1+\alpha) \times \mu$ or
- $S_i(\mathbf{p}_i)$ =min of shape indexes and $S_i(\mathbf{p}_i) \leq (1-\beta) \times \mu$

where $\mu = \frac{1}{N_r}\sum_{j=1}^{N_r} S_i(\mathbf{p}_j)$ and $0 \leq \alpha, \beta \leq 1$. The parameters α and β control the selection of feature points and N_r is the number of points in the local window.

The shape index of a dolphin geometry is given in Figure 4.12. The shape index of the skull geometry is then given in Figure 4.13 with the corresponding feature points automatically extracted for a radius $r_i = 10$, $\alpha = 0.1$ and $\beta = 0.1$ in Figure 4.14.

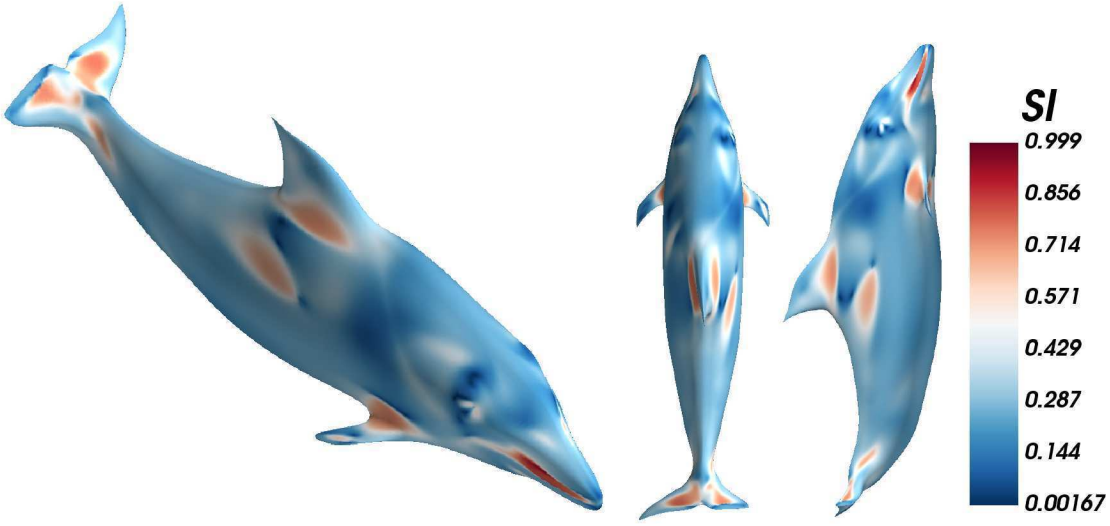
4.3.4 Ridges and Valleys

The ridge and valley nodes are found after obtaining curvature information at each mesh vertex. The mesh vertices that approximate ridge and valley nodes can be found by either finding the zero crossing of extremality coefficients or using only principal curvatures and directions. The nodes are then connected into lines following the directions of principal curvature.

An outline of how to obtain and differentiate between crest nodes is derived from the requirements on concave and convex crest lines presented in Equations (4.2) and (4.3).

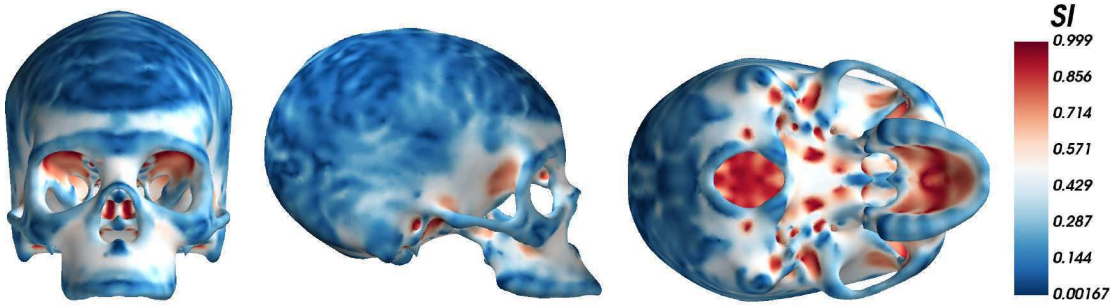
Determining ridge nodes:

- For each node $\mathbf{p}_j$ satisfying $\kappa_{\max} > |\kappa_{\min}|$ the immediate neighbours are inspected.



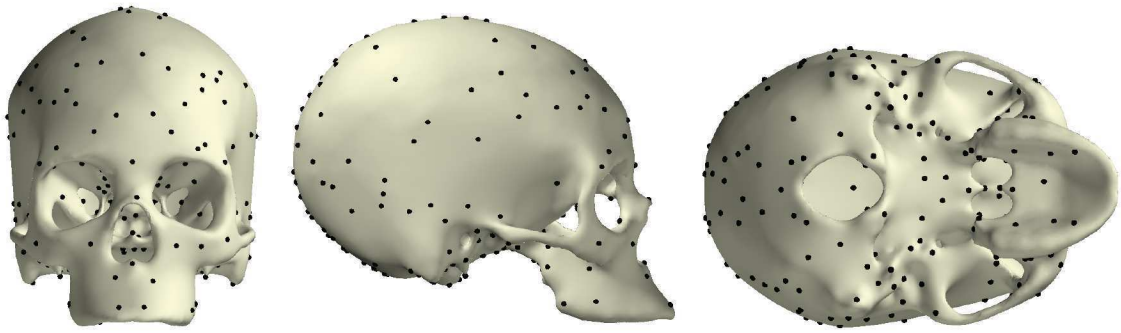
(a) (b) (c)

Figure 4.12: Shape index of a dolphin geometry after using a 5-ring neighbourhood for MLS surface fitting. (a) Isometric (b) top and (c) side view.



(a) (b) (c)

Figure 4.13: Shape index of the skull geometry after using a 3-ring neighbourhood for MLS surface fitting. (a) Frontal, (b) side and (c) bottom view.



(a) (b) (c)

Figure 4.14: Feature points automatically extracted from skull geometry for radius $r_i = 10$ and $\alpha = 0.1, \beta = 0.1$. (a) Frontal, (b) side and (c) bottom view.

- The vector distance to the neighbours are determined.
- Only the neighbours who lie more in direction $\boldsymbol{\varpi}_{\max}$ than $\boldsymbol{\varpi}_{\min}$ are added to the neighbour set I.
- If only principal curvatures and directions are used:
  - $\mathbf{p}_j$ is considered a possible ridge node if for all $i \in I : \kappa^j_{\max} > \kappa^i_{\max}$ is satisfied.
- If the zero crossing of extremality coefficients is required:
  - $\forall i \in I$: If $\boldsymbol{\varpi}^j_{\min} \bullet \boldsymbol{\varpi}^i_{\min} < 0$, the sign of the relevant extremality coefficient is reversed so $\tau^i_{\max} = -\tau^i_{\max}$.
  - $\mathbf{p}_j$ is considered a possible ridge node if for any $i \in I : \tau^j_{\max} \times \tau^i_{\max} < 0$ and $\kappa^j_{\max} > \kappa^i_{\max}$ is satisfied.

Determining valley nodes:

- For each node $\mathbf{p}_j$ satisfying $\kappa_{\min} < -|\kappa_{\max}|$ the immediate neighbours are inspected.
- The vector distance to the neighbours are determined.
- Only the neighbours who lie more in direction $\boldsymbol{\varpi}_{\min}$ than $\boldsymbol{\varpi}_{\max}$ are added to the neighbour set I.

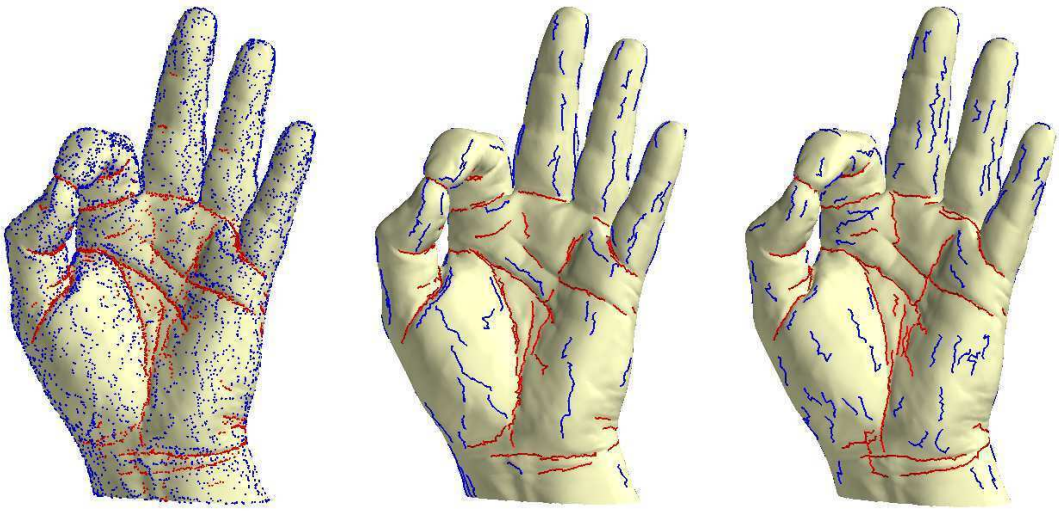
- If only principal curvatures and directions are used:
  - $\mathbf{p}_j$ is considered a possible valley node if for all $i \in I : \kappa^j_{\min} < \kappa^i_{\min}$ is satisfied.
- If the zero crossing of extremality coefficients is required:
  - $\forall i \in I$: If $\varpi^j_{\max} \bullet \varpi^i_{\max} < 0$, the sign of the relevant extremality coefficient is reversed so $\tau^i_{\min} = -\tau^i_{\min}$.
  - $\mathbf{p}_j$ is considered a possible valley node if for any $i \in I : \tau^j_{\min} \times \tau^i_{\min} < 0$ and $\kappa^j_{\min} < \kappa^i_{\min}$ is satisfied.

The procedure of finding the zero crossing of curvature derivatives is more mathematically sound when determining crest nodes. This being the case, it would be the preferred method in determining whether a mesh node is likely to be part of a feature line. As in the smooth surface case, the sign of principal curvature directions are not uniquely defined. This is because curvature lines follow line fields and not vector fields [33].

From Equations (4.2) and (4.3) the zero crossing of extremality coefficients are required. These zero crossings of τ depend on a consistent choice for the sign of the principal directions between neighbours. Should the orientation of a neighbouring principal direction be reversed, the sign of the corresponding τ as determined from Equation (4.9) also needs changing.

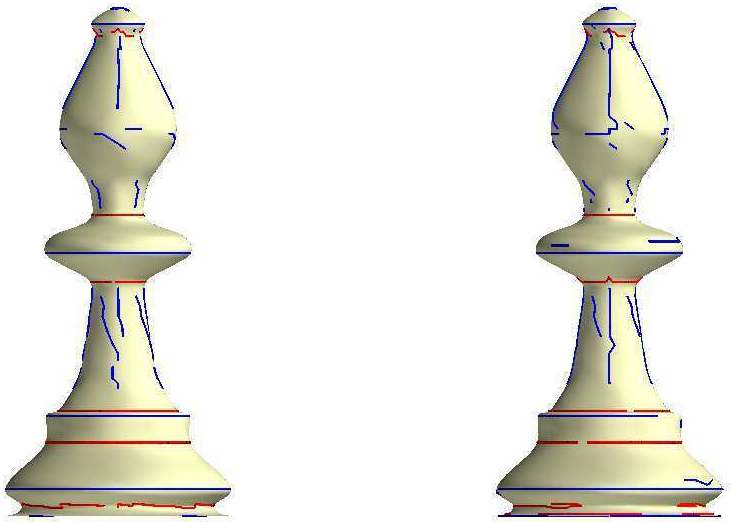
Due to higher order terms present in the polynomial surface approximation, the resulting curvature derivatives are very sensitive to surface discretisation. The different approaches possible to extract curvature lines by either principal curvatures or the zero crossing of extremality coefficients is visible in the results of Figures 4.15 through 4.18. Although using principal curvatures seem to detect less false crest lines, it is still somewhat prone to report these. Both methods however recover the main lines of interest in a similar fashion and after appropriate thresholding should result in approximately the same reported features .

As the detection of crest lines is sensitive to surface discretisation, subdivided faces without sufficient smoothing could for example recover the original mesh edges as features. This can be seen in the seemingly repeated lines or parts thereof on the refined dolphin geometry of Figure 4.18 and false crest nodes on the refined trim-star geometry of Figure 4.19 (a). As with other spurious or false lines, most



(a) (b) (c)

Figure 4.15: Crest nodes and lines on a hand geometry. (a) Possible ridge (blue) and valley (red) points obtained by using only principal curvatures and derivatives along with (b) the lines after connecting points in the relevant principal direction. (c) The equivalent ridge and valley lines obtained by using curvature derivative zero crossing procedure. For visual clarity only lines with more than 8 segments are displayed.



(a) (b)

Figure 4.16: Crest lines on a refined and smoothed bishop geometry. (a) Ridge (blue) and valley (red) lines obtained by using only principal curvatures and directions. (b) The equivalent ridge and valley lines obtained by using curvature derivative zero crossing procedure.

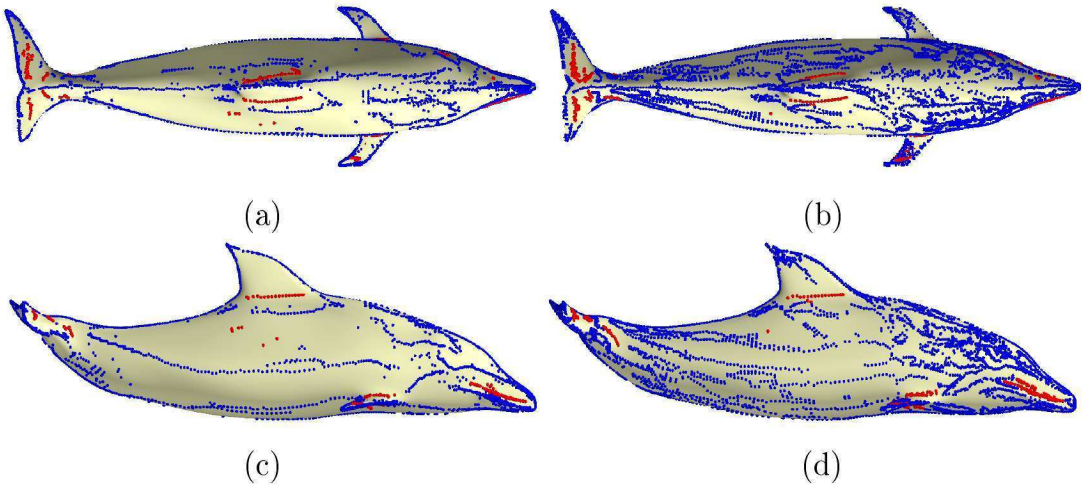


Figure 4.17: Crest nodes on the refined version of a dolphin geometry. (a) and (c) Ridge (blue) and valley (red) nodes obtained by using only principal curvatures and directions. (b) and (d) The equivalent ridge and valley nodes obtained by using curvature derivative zero crossing procedure.

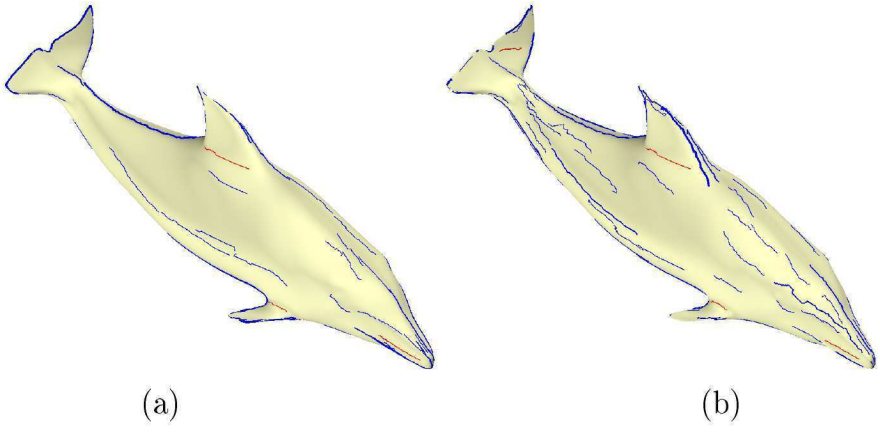
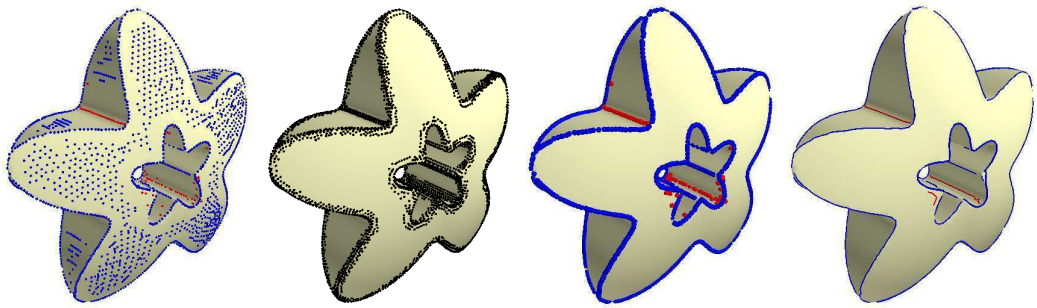


Figure 4.18: Crest lines on the refined version of a dolphin geometry. (a) Ridge (blue) and valley (red) lines obtained by using only principal curvatures and directions. (b) The equivalent ridge and valley lines obtained by using curvature derivative zero crossing procedure. Lines have been pruned and line width scaled according to threshold calculated with Equation (4.11) and a minimum threshold value of $T_h = 10$.



(a) (b) (c) (d)

Figure 4.19: Extracted ridge (blue) and valley (red) lines on the refined trim-star geometry. (a) Extracted crest nodes with presence of false edges already evident on the smoother areas of the geometry. (b) Allowable crest nodes after filtering out those that don't satisfy the local structure tensor condition $\lambda_1 < 10 \times \lambda_2$. (c) The extracted crest nodes that satisfy the local structure tensor filtering condition in (b). (d) The ridge and valley lines constructed using only the filtered crest nodes. In this figure some spurious or false lines are still present indicating that thresholding might still be required. These false lines are likely picked up due to the local discretisation.

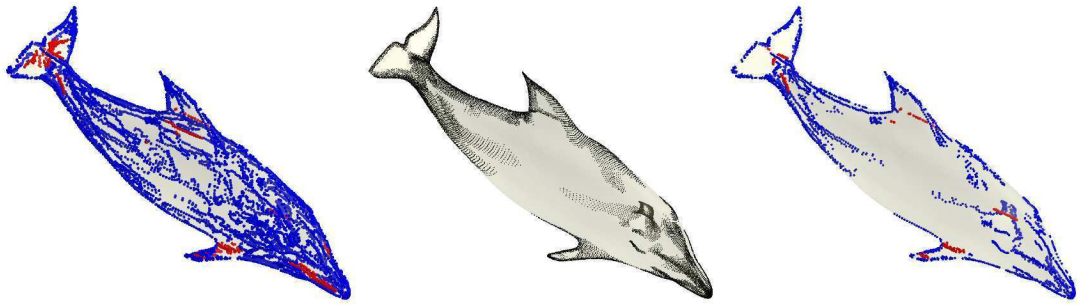
of these lines on the dolphin geometry could be pruned. Unfortunately those with sufficient curvature or those that form part of a line with sufficient threshold are still reported when using the implemented method.

Figures 4.19 and 4.20 illustrate the use of the local structure tensor information to filter ridge and valley nodes.

Connecting nodes into lines

To obtain ridge and valley lines the extracted crest nodes are connected in the direction of principal curvature. Ridges follow the direction of minimum curvature while valleys follow the maximum curvature directions. When connecting the ridge and valley lines, it is again important to note that the orientation of the principal direction is irrelevant. Starting at nodes with maximum curvature:

- For each extracted crest node, a search is performed in both positive and negative orientation of the principal curvature direction.
- Neighbouring crest nodes found in these directions are then considered possible line segment partners.
- A connection is made to these neighbouring nodes.



(a) (b) (c)

Figure 4.20: Filtering crest nodes on the refined version of a dolphin geometry. (a) Ridge (blue) and valley (red) nodes extracted using 100 nearest neighbour search with extremality zero crossing procedure. (b) Points for feature areas satisfying $\lambda_1 < 100 \times \lambda_2$ resulting from 3-ring neighbourhood local structure evaluation. (c) Ridge and valley nodes remaining after the filter procedure.

  - If more than one possible neighbour is detected in a given principal direction orientation, a connection is made to the neighbour who lies more in this direction.

- If any remaining neighbours have been detected as possible crest nodes, these are pruned.

Thresholding

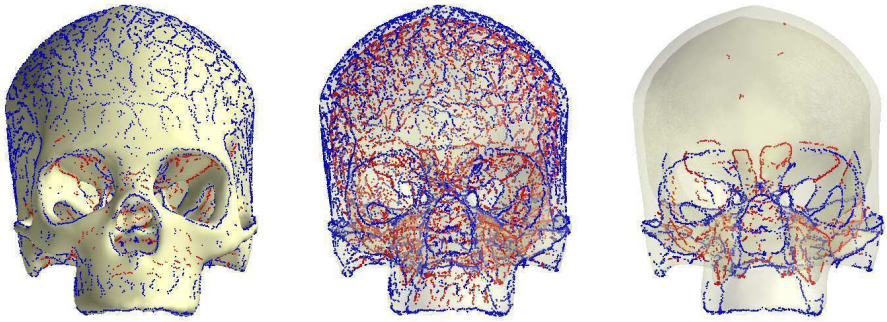
Thresholding is used to get rid of insignificant or false reported ridge and valley lines. The criterion selected and implemented in this report first assigns a single value to every line. This value could be obtained by integrating a curvature or extremality dependent function along the line.

The chosen method implemented involves determining a line threshold as [45]

$$T_h = \int \kappa ds \approx \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{p}_{i+1} - \mathbf{p}_i\| \left(\kappa_{i+1} + \kappa_i\right). \tag{4.11}$$

This calculation uses the trapezoid approximation of the integral. Here, the length of each line segment is multiplied with the average line curvature. For ridges, the κ value is taken as $\kappa_{\max}$ while $-\kappa_{\min}$ is used when determining valley line threshold values.

All of the lines with a threshold value below a user specified scalar value are pruned. In addition, the use of the local structure tensor is also seen as an easy



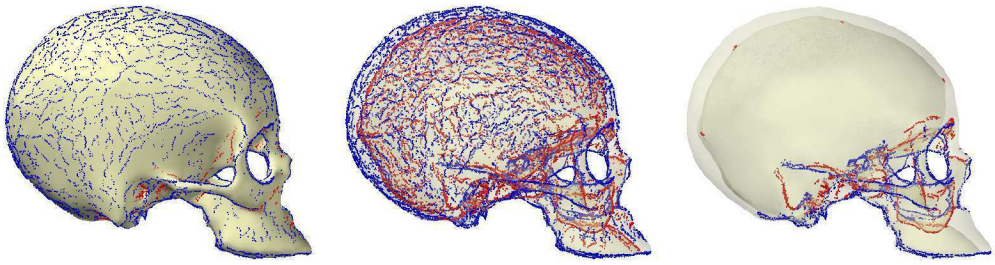
(a) (b) (c)

Figure 4.21: Frontal view of the skull geometry with extracted ridge nodes in blue and valley nodes in red. (a) and (b) show all of the crest nodes on the geometry and (c) contains only nodes that satisfy the local structure tensor condition $\lambda_1 < 50 \times \lambda_2$.

method to avoid unnecessary computation. If the results of a local structure tensor analysis is available or easily determined, the areas containing possible feature curves could first be extracted. Fitting a local smooth surface approximation and determining curvatures, curvature directions and derivatives can then be done on only the filtered areas. Alternatively it could be applied to only preserve crest nodes within a user specified feature rich area.

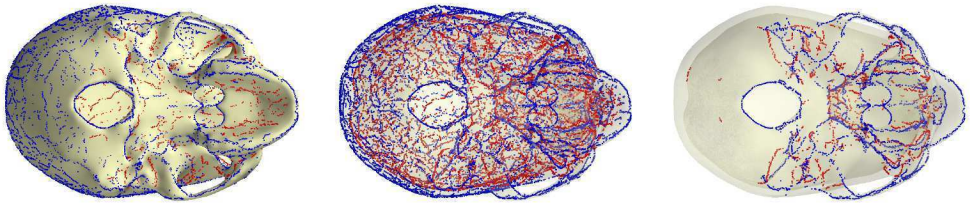
In Figure 4.19 the effect of local structure tensor filtering is visible. Here the entire method of approximating a local smooth surface and extracting feature lines was first applied. It was then filtered to only keep nodes where $\lambda_1 < 10 \times \lambda_2$. The resulting feature lines require less effort to determine an appropriate threshold value.

The possibility of using the local structure tensor result as a pre-curvature estimation step is also considered. Figures 4.21 through 4.23 show crest nodes extracted from a skull geometry. The first case was run on the entire skull model, finding 10'977 ridge and 8'170 valley nodes. A local structure tensor filter was then applied to find nodes where $\lambda_1 < 50 \times \lambda_2$. This is implemented using a nearest neighbour search for 30 of the closest nodes. The MLS approximated implicit surface and crest node extraction procedure was then done on only the 36'617 nodes satisfying the local structure filter instead of the 191'957 nodes of the original model geometry. The resulting 2'936 ridge and 1'708 valley nodes are connected into lines without the need for much additional thresholding. The final thresholded lines with more than 4 line segments are displayed in Figures 4.24 and 4.25.



(a) (b) (c)

Figure 4.22: Side view of the skull geometry with extracted ridge nodes in blue and valley nodes in red. (a) and (b) show all of the crest nodes on the geometry and (c) contains only nodes that satisfy the local structure tensor condition $\lambda_1 < 50 \times \lambda_2$.



(a) (b) (c)

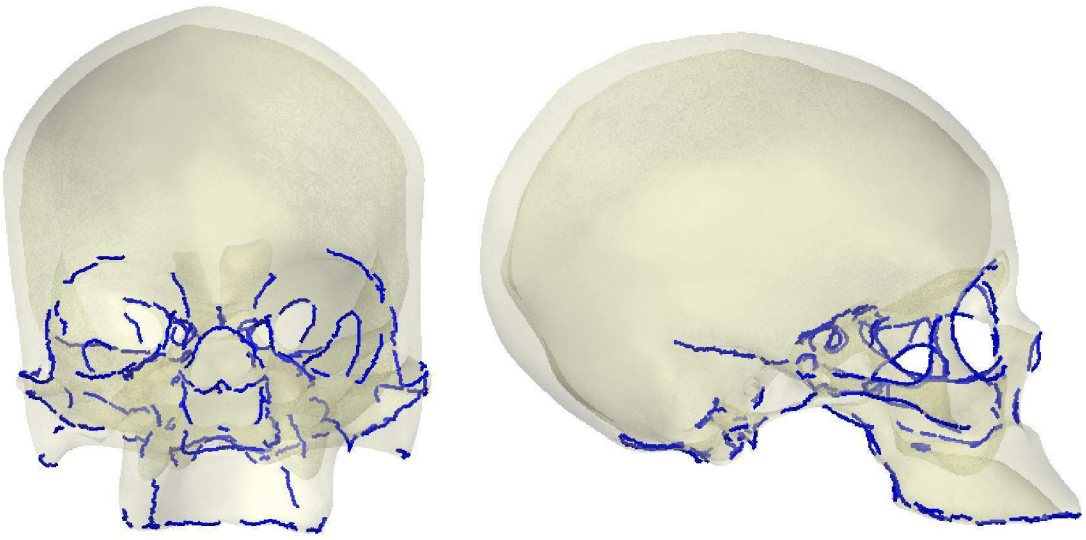
Figure 4.23: Lower view of the skull geometry with extracted ridge nodes in blue and valley nodes in red. (a) and (b) show all of the crest nodes on the geometry and (c) contains only nodes that satisfy the local structure tensor condition $\lambda_1 < 50 \times \lambda_2$.

4.4 The Selection of Matching Features

The information obtained from a local structure tensor analysis can be used to first evaluate the existence of specific features in a given geometry. This would be the first step to determine something akin to feature covariance between like geometries.

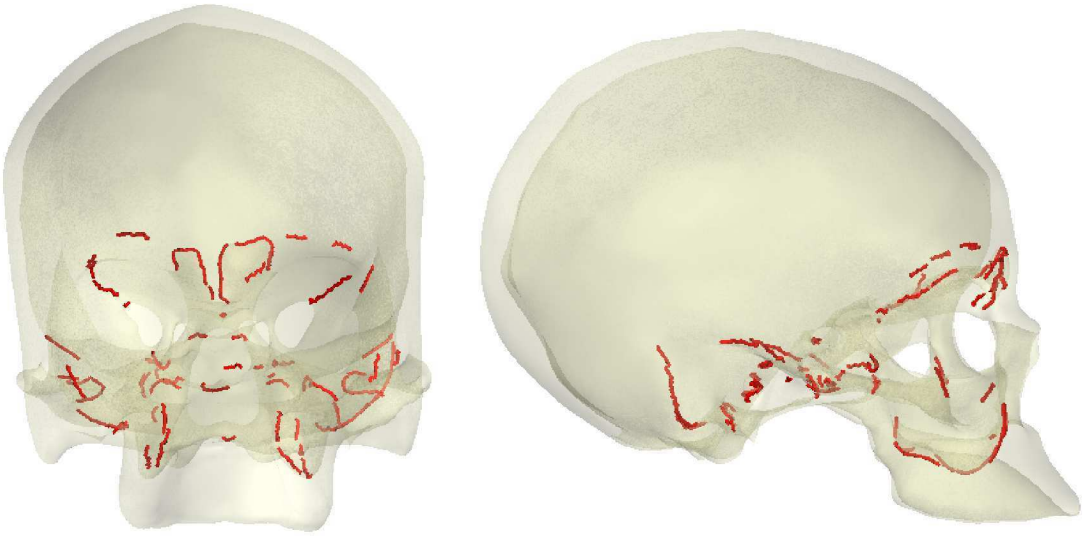
Where closely similar geometries need registering, problems would occur for instances where there is a difference in the topology or if the features overlap as in the case with creating a symmetric smoothed skull geometry in Figure 3.11.

Differences in topology can arise from the geometry itself such as a bullet wound in one skull and no equivalent trauma on the other or even skull degradation. It could also arise as a result of post-processing when surface representations are constructed from MRI or Computed Tomography (CT) scan data. A requirement to completely register a reference skull onto the target geometry would therefore almost certainly develop the need to alter the connectivity of the reference mesh. This would then destroy the required one-to-one consistent mapping between all



(a) (b)

Figure 4.24: Thresholded ridge lines on the skull geometry after first applying the filter $\lambda_1 < 50 \times \lambda_2$ and then thresholding lines to $T_h = 200$. (a) Frontal and (b) lateral view. Only lines with more than 4 line segments are displayed.



(a) (b)

Figure 4.25: Thresholded valley lines on the skull geometry after first applying the filter $\lambda_1 < 50 \times \lambda_2$ and then thresholding lines to $T_h = 200$. (a) Frontal and (b) lateral view. Only lines with more than 4 line segments are displayed.

registered geometries in the sample.

To overcome this problem, a registration routine is required where only selected features are matched. The extraction of feature lines makes it possible to better quantify and describe features within a given geometry. These lines could be used to determine similarities between objects.

If a line or line segment has no equivalent in another geometry of interest, it could be discarded after registration. In this way only matched features would be used in eventually determining the deformation required to morph one geometry into another.