



Report on assessment of MSSTs for 3 D Mathing Application

Somchaipeng, Kerawit; Sporning, Jon; Kreiborg, Sven; Johansen, Peter

Publication date:
2004

Document version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Somchaipeng, K., Sporning, J., Kreiborg, S., & Johansen, P. (2004). *Report on assessment of MSSTs for 3 D Mathing Application*.

Deep Structure, Singularities, and Computer Vision

DSSCV



IST-2001-35443

Deliverable

Deliverable No.:	21
Title:	Report on assessment of MSSTs for 3D Matching Application
Author(s):	K. Somchaipeng, J. Sporning, S. Kreiborg, and P. Johansen
Institution:	3D-Lab, School of Dentistry, Dept. of Pediatric Dentistry, University of Copenhagen
Classification:	Public
Date:	20 September 2004

Report on Assessment of MSSTs for 3D Matching Application

K. Somchaipeng^{1,2}, J. Sparring², S. Kreiborg¹, and P. Johansen²

¹3D-Lab, School of Dentistry, Dept. of Pediatric Dentistry, University of Copenhagen Nørre Alle 20, DK-2200 Copenhagen N, Denmark

²Dept. of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100, Copenhagen N, Denmark

Abstract

We consider images as manifolds embedded in a hybrid high dimensional space of coordinates and features. Images are partitioned into segments based on the energy functional and mathematical landmarks. The nesting of image segments in scale-space is used to construct image hierarchies called Multi-Scale Singularity Trees (MSSTs). Two kinds of mathematical landmarks are proposed namely extremal points and saddle points.

We study the stability of the Extrema-Based MSSTs extracted from a series of smoothly changed generated images. The observed MSST transitions can be categorized into three categories: changes of positions, changes of relations, and changes of connections. The stability comparison between Extrema-Based MSSTs and Saddle-Based MSSTs is also briefly presented.

We point out the equivalence between MSSTs and Multi-Scale Singularity Strings (MSSSs) and propose two sets of edit operations on MSSSs, one for Extrema-Based MSSSs and one for Saddle-Based MSSSs. Simple examples of editing MSSSs are also presented.

We describe a medical image database of volumetric CT scans of the craniofacial area and manually segmented teeth. The database is aimed at evaluating the image matching algorithm based on MSSTs.

1 Gaussian Scale-Space

The $N + 1$ dimensional Gaussian scale-space, $L : \mathbb{R}^{N+1} \rightarrow \mathbb{R}$, of an N dimensional image, $I : \mathbb{R}^N \rightarrow \mathbb{R}$, is an ordered stack of images, where each image is a blurred version of the former [7, 20, 9]. The blurring is performed according to the diffusion equation,

$$\partial_t L = \nabla^2 L, \quad (1)$$

where $\partial_t L$ is the first partial-derivative of the image in the scale direction t , and ∇^2 is the Laplacian operator, which in three dimensions reads $\partial_x^2 + \partial_y^2 + \partial_z^2$.

The Gaussian kernel is the Green's function of the heat diffusion equation, i.e.

$$L(\cdot; t) = I(\cdot) \otimes g(\cdot; t), \quad (2)$$

$$g(x; t) = \frac{1}{(4\pi t)^{N/2}} e^{-x^T x / (4t)}, \quad (3)$$

where $L(\cdot, t)$ is the image at scale t , $I(\cdot)$ is the original image, \otimes is the convolution operator, $g(\cdot; t)$ is the Gaussian kernel at scale t , N is the image dimensionality, and $t = \sigma^2/2$, using σ as the standard

deviation of the Gaussian kernel. The *Gaussian scale-space* is henceforth called the scale-space in this article.

The information in scale-space is logarithmically degraded, the scale-parameter is therefore often sampled exponentially using $\sigma = \sigma_0 e^T$. Since differentiation commutes with convolution and the Gaussian kernel is infinitely differentiable, differentiation of images in scale-spaces is conveniently computed,

$$\partial_{x^n} L(\cdot; t) = \partial_{x^n} (I(\cdot) \otimes g(\cdot; t)) = I(\cdot) \otimes \partial_{x^n} g(\cdot; t). \quad (4)$$

Alternative implementations of the scale-space are multiplication in the Fourier Domain, finite differencing schemes for solving the heat diffusion equation, additive operator splitting[19], and recursive implementation[6, 18].

Each method has its advantages and disadvantages, we prefer the spatial convolution, since it is guaranteed not to introduce spurious extrema in homogeneous regions at low scales. Typical border conditions are Dirichlet, Cyclic repetition, and Neumann boundaries. We use Dirichlet boundaries, where the image is extended with zero values in all directions according to the size of the convolution kernels.

Although the dimensionality of the constructed scale-space is one higher than the dimensionality of the original image, *critical points*, in the image at each scale are always points. A critical point is e.g. an extremum, $\partial_x L = \partial_y L = 0$. Critical points are classified by the eigenvalues of the Hessian matrix, the matrix of all second derivatives, computed at that point. Critical points with all positive eigenvalues are minima, critical points with all negative eigenvalues are maxima, and critical points with a mixture of both negative and positive eigenvalues are saddles.

As we increase the scale parameter, the critical points move smoothly forming *critical paths*. Along scale, critical points meet and annihilate or are created. Such events are called *catastrophic events*, and the points where they occur are called *catastrophe points*. The collection of events at all scales is called the *deep structure* of the image.

The notion of genericity is used to disregard events that are not likely to occur for typical images, i.e. generic events are stable under slight perturbation of the image. There are only two types of generic catastrophe events in scale-space namely pairwise *creation events* and pairwise *annihilation events*[5]. It has further been shown that generic catastrophe events only involves pairs of critical points where one and only one eigenvalue of the Hessian matrix changes its sign, e.g. the annihilation of a minimum (+, +, +) and a saddle (+, +, -). A detailed discussion of a robust method for extracting critical paths and catastrophe points from scale-spaces can be found in [15].

2 Multi-Scale Singularity Trees

There are a few methods for constructing hierarchies from the deep structure of two-dimensional images proposed in the literatures so far [11, 10, 13]. To the best of our knowledge, an attempt to construct hierarchies from the deep structure of three-dimensional images is first proposed in [12] followed by [15]. The latter scheme, which will be extended in this Section, produces rooted ordered binary trees called Multi-Scale Singularity Trees (MSSTs) with catastrophe points as nodes. It only considers the linking of the mathematical landmarks which present at the first scale image. Creation catastrophes are ignored

2.1 Energy Functional and Energy Partitions

Let $\Omega \subset \mathbb{R}^N$ be a compact connected domain and define $I : \Omega \rightarrow \mathbb{R}^+$ to be an image, $\vec{e} \in \Omega$ as a landmark, and $\vec{x} \in \Omega$ as a point in the domain. Consider a set of continuous functions $\gamma : [0, P] \rightarrow \Omega$

for which $\gamma(0) = \vec{e}$ and $\gamma(P) = \vec{x}$, $\gamma \in \Gamma_{\vec{e}\vec{x}}$, where $\Gamma_{\vec{e}\vec{x}}$ is the set of all possible paths from a landmark \vec{e} to a point \vec{x} , and γ is parameterized using Euclidean arc-length. The energy $E_{\vec{e}}(\vec{x})$ with respect to an extremum \vec{e} calculated at \vec{x} is defined as

$$E_{\vec{e}}(x) = \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \frac{1}{\sqrt{\alpha}} \int_0^P \sqrt{\left| \frac{\partial \gamma(p)}{\partial p} \right|^2 + \alpha \left| \frac{\partial I(\gamma(p))}{\partial p} \right|^2} dp. \quad (5)$$

Considering images as manifolds embedded in a high dimensional space. An N dimensional intensity image becomes an N dimensional manifold embedded in a hybrid $N + 1$ dimensional space of coordinates and the features¹, the “space-feature”[8]. For two dimensional images, it is commonly known as the height plot of the image. The energy defined at a point associated with a landmark can then be thought of as the minimum of the distances travelling up and down from the landmark to that point. When the coefficient α is set to infinity, the energy is also known as the path variation, a generalization of the total variation[3].

Let $\mathcal{E} \subset \Omega$ be the set of all landmarks in the image. An image segment or an energy partition S_i associated with a landmark $\vec{e}_i \in \mathcal{E}$ is defined as the set of all points in the images, where the energy $E_{\vec{e}_i}(\vec{x})$ is minimal,

$$S_i = \{\vec{x} \in \Omega | E_{\vec{e}_i}(\vec{x}) < E_{\vec{e}_j}(\vec{x}), \forall \vec{e}_j \in \mathcal{E}, i \neq j\}. \quad (6)$$

An approximation of the energy map $M_i : \Omega \rightarrow \mathbb{R}^+$, which defines the energy at every point in the image associated with a landmark \vec{e}_i , can be efficiently calculated using the *Fast Marching Methods*[14].

The tessellation of the image segments obtained depends on the selection of the landmarks and the energy functional. Mathematical landmarks like extremal points seem to be a natural selection since they represent physically the image content, significant features in the image contain at least one such points, and they can be easily and automatically detected. The selection of extremal points as landmarks leads to Extrema-Based MSSTs.

Another candidate for landmarks are the saddles points, which leads to Saddle-Based MSSTs. We show later in Section 3. that the transition of Saddle-Based MSSTs are simpler than those of Extrema-Based MSSTs.

2.2 Building Scale-Space Hierarchies

MSSTs are obtained by linking catastrophe points based on the nesting of image segments in scale-space. Resulting MSSTs are always rooted ordered binary tree. For simplicity, we will begin by describing the building procedures of Extrema-Based MSSTs and then move to Saddle-Based MSSTs at the end of the Section.

2.2.1 Extrema-Based MSSTs

Each node of the tree consists of three parts, the *leftport*, the *body*, and the *rightport*. The leftport represents the image segment that immediately covers the image segment represented by the rightport, which disappears after the catastrophe event mentioned in the body. Assuming that critical paths and catastrophe points in the scale-space image are already and correctly extracted, the tree building algorithm is as follows:

The algorithm starts by setting the root of the tree as $BC_{\infty}E_{last}$, where B denotes border of the image, E_i denotes the last extremum points in scale, and C_{∞} denotes the virtual catastrophe at scale infinity where the border and the last extremum points virtually annihilate.

¹In this case the only feature is the intensity or the zeroth jet space. The energy functional can be easily defined for higher order jet space images, color images, or locally orderless images with scale-space histograms to handle textures[17], if a metric in the feature space is given.

At the highest catastrophe point C_{top} in scale, calculate an energy map with respect to the catastrophe C_{top} , create a node $E_{cover}C_{top}E_{ann}$, where E_{cover} is the extremum which its image segment covers the catastrophe C_{top} , and E_{ann} is the extremum that annihilates with a saddle at the catastrophe C_{top} . Link the new created node as the left child of the root if E_{cover} equals B or as the right child of the root if E_{ann} equals E_{last} .

Determining the covering image segment is easy since it is the image segment belonging to the extremum or the border with minimal energy. We simply choose the extremum or the border that has the lowest value in the energy map with respect to the catastrophe.

Repeat the same procedure on the next highest catastrophe C_{next} to the lowest one but instead of linking the new node as the right child or the left child of the root, link the new node as the left child of a node in the tree, which does not have its left child and its leftport equals the leftport of the new node, or link the new node as the right child of a node in the tree, which does not have its right child and its rightport equals the leftport of the new node. There will be only one possible linking point exists. An example of Extrema-Based MSSTs together with the deep structure it represents are shown in Figure 1(a).

2.2.2 Saddle-Based MSSTs

A similar procedure is applied for constructing Saddle-Based MSSTs. The algorithm starts with setting the root of the tree as $C_{top}S_{top}$, where the leftport is set to *null*, C_{top} denotes the highest catastrophe in scale, and S_{top} denotes the saddle point that annihilates with an extremum at C_{top} . The root will have only the right child which is $S_{top}C_{next}S_{ann}$, where C_{next} denote the next highest catastrophe point, and S_{ann} denotes the saddle point that annihilates with an extremum at the catastrophe C_{next} .

Go down the scale to the next highest catastrophe and repeat the procedure described in Section 2.2.1. considering saddle points as landmarks instead of extrema. An example of Saddle-Based MSSTs is shown together with the deep structure it represents in Figure 1(b).

2.3 Discussion

We prefer our tree building method because unlike all other proposed methods, which produce hard-linked trees: only the best linking point is suggested, our method produces soft-linked trees in the sense that all possible linking points are suggested where linking points with the lowest energy is selected. The added information is very useful for estimating the costs of transitions.

If the energies associated with a catastrophe point calculates at two different landmarks are slightly different, it implies that the link could then be easily switched with small perturbation on the image and the cost of this transition should be small accordingly. On the other hand if the energy difference is large, it then can be assumed that the link is rather stable and a high cost should be paid for the transition. The cost of that transition can be estimated proportionally to the difference of the two energies.

3 Stability of MSSTs

To empirically study the stability of MSSTs, a series of generated images of three stationary and one moving Gaussian blobs is used in the experiment. All images in the series have the same number of extrema, i.e. a minimum surrounded by four maxima. Smooth changes of the images are forced by smoothly changing the parameters used for generating the images. For each test image a scale-space is computed and the Extrema-Based MSST is constructed. Changes in the constructed MSSTs between neighbouring images are observed and carefully classified into categories.

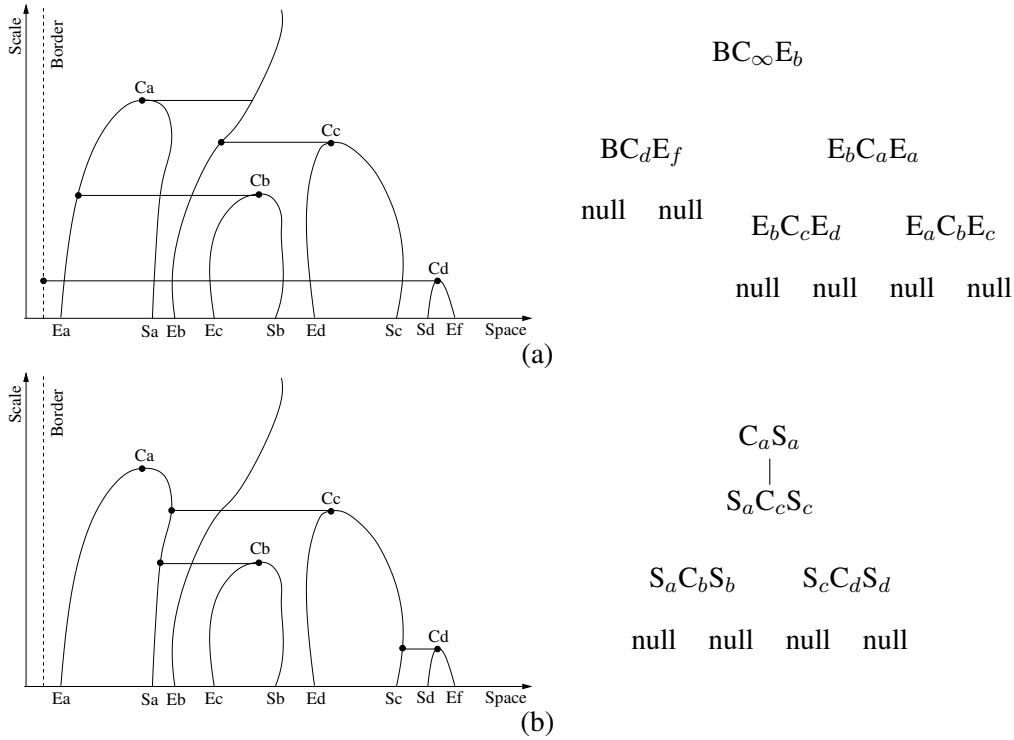


Figure 1: The deep structure with its corresponding Extrema-Based MSST(a) and the deep structure with its corresponding Saddle-Based MSST(b)

Three categories of transitions are observed in the experiment, i.e. *changes of positions*, *changes of connections*, and *changes of relations*. Changes of positions are the movements of the extrema and catastrophe points, changes of connections swap the connection between two extremum-catastrophe pairs, and changes of relations are the changes of the nesting of the energy partitions in scales. Interested readers can find more detail on the experiment in [16].

Figure 2. shows the before and after images of the observed three transition categories, with the critical paths and the linkings superimposed on the images.

3.1 Discussion

In practice, some of the the observed three transition categories might occur at the same time. In general, changes of positions occur most frequently. Changes of connections and changes of relations usually accompanied by changes of positions.

Since the process of constructing Saddle-Based MSSTs is only interested in saddle points and their paths in scale-space, the changes of connections, which are the connections between catastrophe points and extremal points, do not effect their structures at all. There are then only two kinds of transitions for Saddle-Based MSSTs: *changes of positions* and *changes of relations*. Therefore we expect Saddle-Based MSSTs to be more stable than Extrema-Based MSSTs.

4 Multi-Scale Singularity Strings

Due to redundancies in the tree structure and the data stored in the nodes of the tree, MSSTs can be converted to string structures simply by concatenating nodes of the tree in order according to the

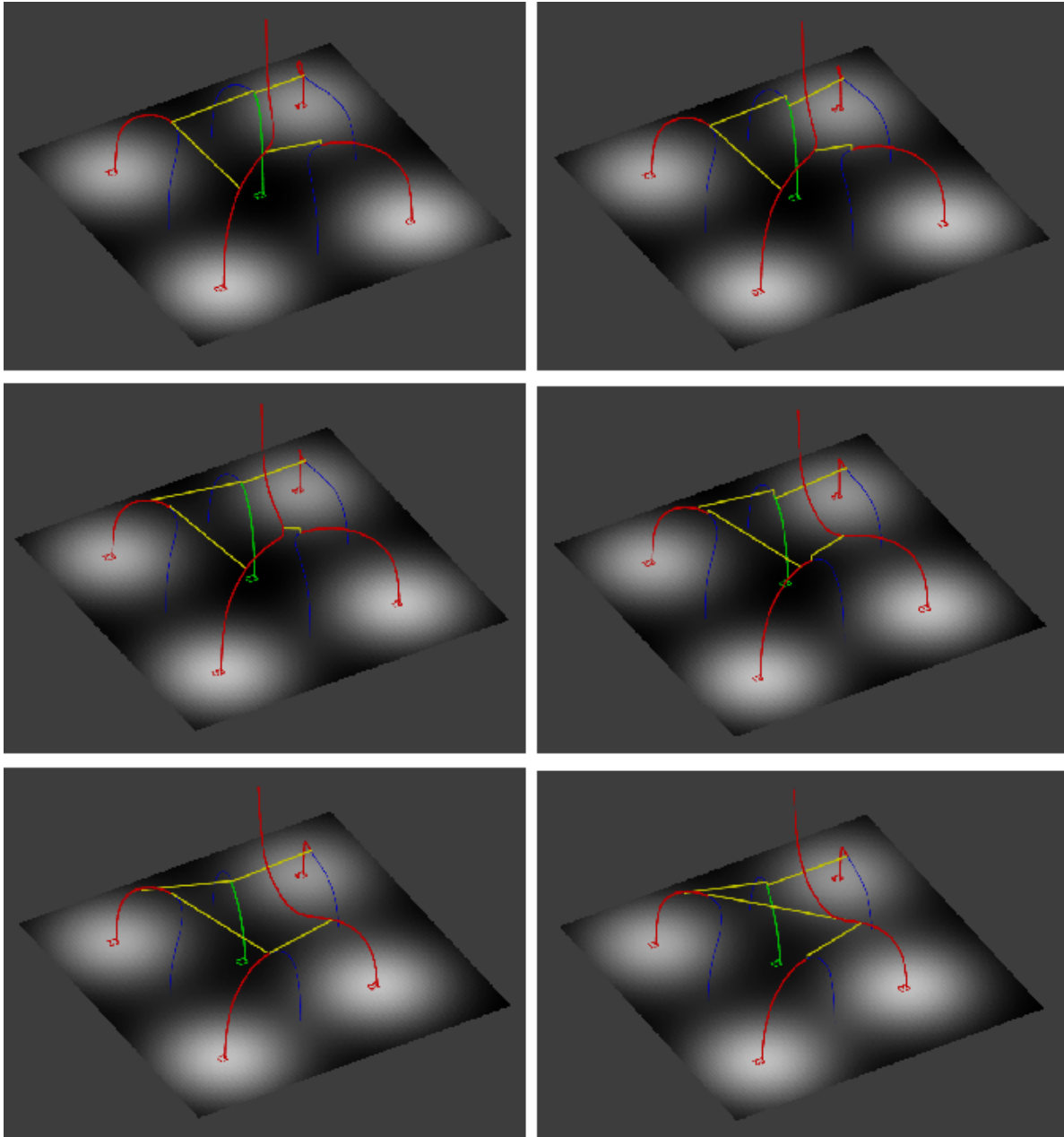


Figure 2: The three observed transitions of Extrema-Based MSSTs shown as images before (left column) and after (right column) the three categories of transitions: changes of positions (top row), changes of connections (middle row) and changes of relations (bottom row). The critical paths and the linkings are superimposed on the images: saddle paths in blue, maximum paths in red, minimum paths in green, and the linkings in yellow.

$$\begin{aligned}
& [BC_\infty E_b][E_b C_a E_a][E_b C_c E_d][E_a C_b E_c][BC_d E_f] \\
& \quad \text{(a)} \\
& [C_a S_a][S_a C_c S_c][S_a C_b S_b][S_c C_d S_d] \\
& \quad \text{(b)}
\end{aligned}$$

Figure 3: The Extrema-Based MSSS(a) and the Saddle-Based MSSS(b) equivalent to their corresponding MSSTs in Figure 1(a) and 1(b).

scales of the catastrophes. The resulting strings are called Multi-Scale Singularity Strings (MSSSs). The information encoded in the structure of the tree is completely kept in the string. An MSST can be converted to its equivalent MSSS and vice versa without loss of information.

The Extrema-Based MSSS and the Saddle-Based MSSS equivalent to the Extrema-Based MSST and the Saddle-Based MSST in Figure 1(a) and 1(b) are shown in Figure 3 (a) and (b), respectively.

5 Matching MSSTs

We would like to solve the problem of image matching by matching their pre-computed MSSTs. The similarity between two images can be estimated as the minimum total cost of a series of edit operations that transforms the MSST of one image to the MSST of the other. This problem is commonly known as the Tree Edit Distance (TED) problem[4].

Because MSSTs can be equivalently represented by their equivalent MSSSs and edit operations on string are simpler and easier to be defined than those on tree, we choose to solve the problem of matching MSSTs by matching their equivalent MSSSs. Due to the nature of causality in scale-space there is a right-to-left dependency between nodes in the MSSSs. Unlike normal character strings, MSSSs are supposed to be edited from right to left.

In order to avoid insertion operations, which is hard to define, two MSSSs S_A and S_B are edited to a common MSSS S_C . Given two MSSSs S_A and S_B , the edit distance $edit(S_A \rightarrow S_B)$ is the sum of the total cost of a series of edit operations $\{o_1, o_2, \dots, o_n\}$ that changes S_A to S_C and the total cost of a series of delete operation $\{d_1, d_2, \dots, d_m\}$ that change S_B to S_C .

$$edit(S_A \rightarrow S_B) = \sum_{i=1}^n cost(o_i) + \sum_{j=1}^m cost(d_j) \quad (7)$$

The cost of the delete operation can be defined proportionally to the stability measures of that catastrophe points. To make the distance symmetric, the distance between S_A and S_B , $distance(S_A, S_B)$, is defined as the sum of $edit(S_A \rightarrow S_B)$ and $edit(S_B \rightarrow S_A)$.

$$distance(S_A, S_B) = edit(S_A \rightarrow S_B) + edit(S_B \rightarrow S_A) \quad (8)$$

In order to be able to define costs for all possible transitions full energy matrices are needed, however with only energy maps constructed with respect to catastrophe points, as described in the building process of MSSTs in Section 2.2., we can only obtain values in the lower triangle of the matrix. We propose to calculate energy maps with respect to landmarks at the first-scale image. This way, full energy matrices can be obtained.

5.1 Editing Extrema-Based MSSSs

Each node in the Extrema-Based MSSS contains only symbols. The content of each symbol, which is what actually to be matched, must be stored in the following tables and matrices.

1. Extremum Table (E-Table)

The table containing the positions and features of all extrema in the first-scale image.

$$T_E = \begin{bmatrix} E_0 & : & position_0, features_0 \\ E_1 & : & position_1, features_1 \\ \vdots & : & \vdots \\ E_n & : & position_n, features_n \end{bmatrix}$$

2. Catastrophe Table (C-Table)

The table containing the positions, features, and stabilities of all catastrophe points in the scale-space image.

$$T_C = \begin{bmatrix} C_1 & : & position_1, scale_1, features_1, stability_1 \\ C_2 & : & position_2, scale_2, features_2, stability_2 \\ \vdots & : & \vdots \\ C_n & : & position_n, scale_3, features_n, stability_n \end{bmatrix}$$

3. Extremum-Extremum Energy Matrix (E-E Matrix)

The matrix containing the energy of the minimal path between all extrema in the first-scale image.

$$M_{EE} = \begin{bmatrix} 0 & E_0E_1 & \cdots & E_0E_n \\ E_1E_0 & 0 & \cdots & E_1E_n \\ \vdots & \vdots & \ddots & \vdots \\ E_nE_0 & E_nE_1 & \cdots & 0 \end{bmatrix}$$

4. Extremum-Catastrophe Energy Matrix (E-C Matrix)

The matrix containing the energy of the minimal paths between all catastrophe points in the scale-space image and all extrema in the first-scale image.

$$M_{CE} = \begin{bmatrix} C_1E_0 & C_1E_1 & \cdots & C_1E_n \\ C_2E_0 & C_2E_1 & \cdots & C_1E_n \\ \vdots & \vdots & \ddots & \vdots \\ C_nE_0 & C_nE_1 & \cdots & C_nE_n \end{bmatrix}$$

These tables and matrices are part of the Extrema-Based MSSS and are given to the matching algorithm. The cost of each edit operation is derived from these tables and matrices.

5.1.1 Edit Operations on Extrema-Based MSSSs

Five edit operations make the complete set of edit operations on Extrema-Based MSSSs. The five edit operations are as follows.

1. Extremum-Editing $E(E_i \rightarrow [pos, feat])$

Changing of the position and features of the extrema E_i stored in the E-Table.

$$\text{cost}(E(E_i \rightarrow [pos, feat])) = w_e | E_i - [pos, feat] |$$

2. Catastrophe-Editing $C(C_i \rightarrow [pos, feat])$

Changing of the position and features of the catastrophe C_i stored in the C-Table. If the ordering of catastrophe points in scale changes, MSSS must be changed accordingly. Illegal relations, if exist, must be properly updated.

$$\text{cost}(C(C_i \rightarrow [pos, feat])) = w_c | C_i - [pos, feat] |$$

3. Left-Relabeling $L(E_{left} \rightarrow E_{new})$

Changing of the leftport from E_{left} to E_{new} which corresponds to changing of the nesting relations from being nested inside E_{left} to E_{new} .

$$\text{cost}(L(E_{left} \rightarrow E_{new})) = w_l | E_{right}E_{new} - E_{right}E_{left} |$$

4. Right-Relabeling $R(E_{right} \rightarrow E_{new})$

Changing of the rightport from E_{right} to E_{new} which corresponds to changing of the connection from C_{cat} being connected to E_{right} to E_{new} . Illegal connections and relations, if exist, must be properly updated.

$$\text{cost}(R(E_{right} \rightarrow E_{new})) = w_r | C_{cat}E_{right} - C_{cat}E_{new} |$$

5. Deletion $D(E_{left}C_{cat}E_{right} \rightarrow \phi)$

Deleting a node $E_{left}C_{cat}E_{right}$ from the MSSS. Illegal relations, if exist, must be properly updated.

$$\text{cost}(D(E_{left}E_{right}C_{cat})) = w_{dstability}(C_{cat})$$

5.1.2 Simple Example

We consider a simple case of editing two Extrema-Based MSSSs S_A and S_B extracted from two images I_A and I_B with two extrema respectively. The extrema in the two images are located at different positions and have different intensity values. The deep structures of S_A and S_B are shown in Figure 4(a) and Figure 4(b), respectively.

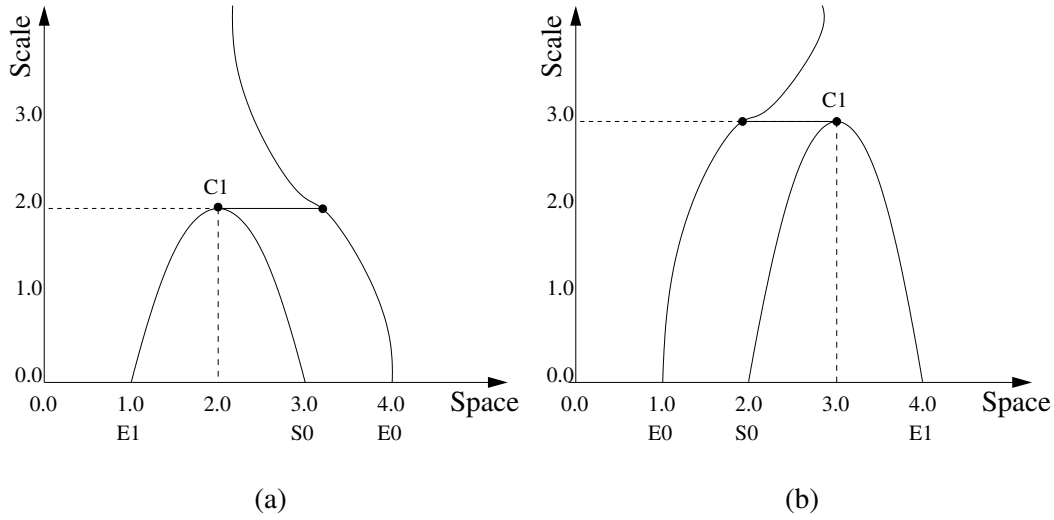
Let S_A be an Extrema-Based MSSS consisting of two nodes: $[BC_\infty E_0][E_0 C_1 E_1]$ with

$$T_E = \begin{bmatrix} E_0 & : & pos = 4 \\ E_1 & : & pos = 1 \end{bmatrix},$$

$$T_C = [C_1 \quad : \quad pos = 2, scale = 2],$$

and S_B be another Extrema-Based MSSS also consisting of two nodes: $[BC_\infty E_0][E_0 C_1 E_1]$ with

$$T_E = \begin{bmatrix} E_0 & : & pos = 1 \\ E_1 & : & pos = 4 \end{bmatrix},$$


 Figure 4: The deep structure of S_A and S_B

$$T_C = [C_1 : pos = 3, scale = 3] .$$

We would like to find a series of edit operations that change S_A to S_B through a common MSSS S_C . The series of the edit operations that changes S_A to S_C , where S_C is the common MSSS, consists of two edit operations²

$$C(C_1 \rightarrow [pos = 3, scale = 3])R(E_1 \rightarrow E_0) .$$

The series of delete operation that changes S_B to S_C is empty and the common string S_C equals S_B . The edit cost $edit(S_A \rightarrow S_B)$ is then

$$edit(S_A \rightarrow S_B) = cost(C(C_1 \rightarrow [pos = 3, scale = 3])) + cost(R(E_1 \rightarrow E_0)) .$$

The first operation corrects the position and scale of the catastrophe C_1 , the second operation changes the connection of C_1 from being connected to E_1 to being connected to E_0 . Because E_0 is no longer present at the scale of C_1 , the leftport E_0 is then no longer valid and must be updated to E_1 . The S_A after applying the edit operations is $[BC_\infty E_0][E_1 C_1 E_0]$ with

$$T_E = \begin{bmatrix} E_0 & : & pos = 4 \\ E_1 & : & pos = 1 \end{bmatrix} ,$$

$$T_C = [C_1 : pos = 3, scale = 3] ,$$

which is now equivalent to S_B .

5.2 Saddles-Based MSSSs

Similar to Extrema-Based MSSSs, each node of the Saddle-Based MSSS contains only symbols. The content of each symbol, which is what we actually want to match, are stored in the following tables and matrices.

²Assuming that this series of edit operation give the lowest total cost.

1. Saddle Table (S-Table)

The table containing the positions and features of all saddles in the first-scale image.

$$T_S = \begin{bmatrix} S_1 & : & position_0, features_0 \\ S_2 & : & position_1, features_1 \\ \vdots & : & \vdots \\ S_n & : & position_n, features_n \end{bmatrix}$$

2. Catastrophe Table (C-Table)

The table containing the positions, features, and stabilities of all catastrophe points in the scale-space image.

$$T_C = \begin{bmatrix} C_1 & : & position_1, scale_1, features_1, stability_1 \\ C_2 & : & position_2, scale_2, features_2, stability_2 \\ \vdots & : & \vdots \\ C_n & : & position_n, scale_3, features_n, stability_n \end{bmatrix}$$

3. Saddle-Saddle Energy Matrix (S-S Matrix)

The matrix containing the energy of the minimal paths between all saddles in the first-scale image.

$$E_{SS} = \begin{bmatrix} 0 & S_1 S_2 & \cdots & S_1 S_n \\ S_2 S_1 & 0 & \cdots & S_2 S_n \\ \vdots & \vdots & \ddots & \vdots \\ S_n S_1 & S_n S_2 & \cdots & 0 \end{bmatrix}$$

5.2.1 Edit Operations on Saddles-Based MSSSs

Because the connections between catastrophe points and extrema in scale-space can change but the connections between saddles and catastrophes are fixed, edit operations on Saddle-Based MSSSs are simpler. There are only three edit operations needed as follows.

 1. Saddle-Catastrophe-Editing $SC(S_{right} \rightarrow [pos_s, feat_s], C_{cat} \rightarrow [pos_c, feat_c])$

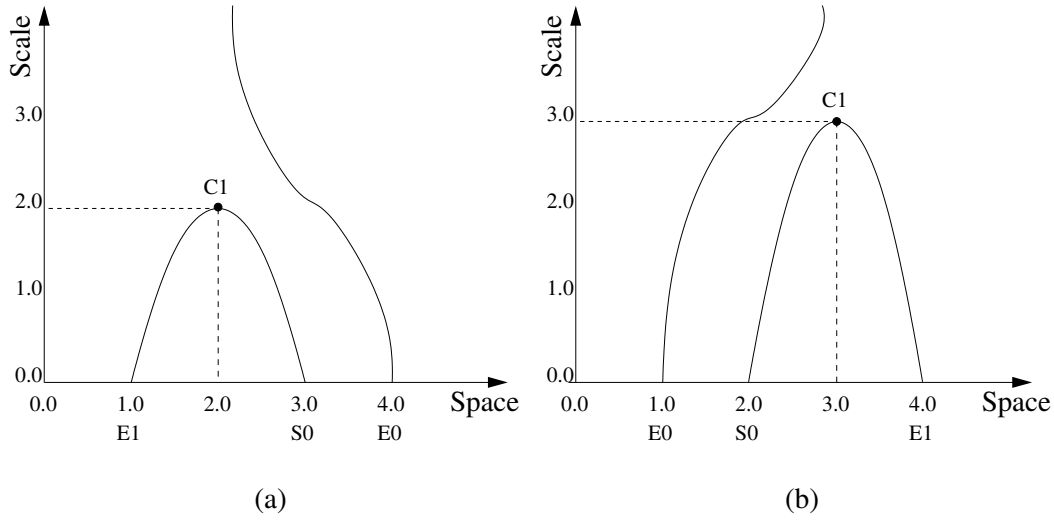
Changing of the position and features of the saddle S_{right} stored in the S-Table and the catastrophe C_{cat} stored in the C-Table. If the ordering of catastrophe points changes, update the MSSS accordingly. Illegal relations, if exist, must be properly updated.

$$cost(SC(S_{right} \rightarrow [pos_s, feat_s], C_{cat} \rightarrow [pos_c, feat_c])) = w_{sc} | S_{right} - [pos_s, feat_s] | + w_c | C_{cat} - [pos_c, feat_c] |$$

 2. Left-Relabeling $L(S_{left} \rightarrow S_{new})$

Changing of the leftport from S_{left} to S_{new} which corresponds to changing of the nesting relations from S_{right} being nested inside S_{left} to S_{new} .

$$cost = w_l | S_{right} S_{left} - S_{right} S_{new} |$$


 Figure 5: The deep structures of S_A and S_B

 3. Deletion $D(S_{left}C_{cat}S_{right} \rightarrow \phi)$

Deleting a node $S_{left}C_{cat}S_{right}$ from the MSSS. Illegal relations, if exist, must be properly updated.

$$\text{cost}(D(E_{left}E_{right}C_{cat})) = w_{dstability}(C_{cat})$$

5.2.2 Simple Example

We consider the image matching of the two images as described in Section 5.1.2 but using Saddle-Based MSSSs instead. The deep structures of the Saddle-Based MSSSs S_A and S_B are shown in Figure 5(a) and Figure 5(b), respectively.

Let S_A be a Saddle-Based MSSS consisting of one node: $[C_1S_1]$ with

$$T_S = [S_1 \quad : \quad pos = 3],$$

$$T_C = [C_1 \quad : \quad pos = 2, scale = 2],$$

and S_B be another Saddle-Based MSSS consisting of one node: $[C_1S_1]$ with

$$T_S = [S_1 \quad : \quad pos = 2],$$

$$T_C = [C_1 \quad : \quad pos = 3, scale = 3].$$

The series of the edit operation that transform S_A to S_B through a common MSSS S_C is

$$SC(C_1 \rightarrow [pos = 3, scale = 3], S_1 \rightarrow [pos = 2])$$

The series of delete operation is empty and the common MSSS S_C equals S_B . The edit cost $\text{edit}(S_A \rightarrow S_B)$ is then

$$\text{edit}(S_A \rightarrow S_B) = \text{cost}(SC(C_1 \rightarrow [pos = 3, scale = 3], S_1 \rightarrow [pos = 2]))$$

The only operation needed to edit S_A to S_B is to corrects the position and scale of the catastrophe C_1 and the position of the saddle S_1 . The S_A after applying the edit operations is $[C_1 S_1]$ with

$$T_S = [S_1 \quad : \quad pos = 2],$$
$$T_C = [C_1 \quad : \quad pos = 3, scale = 3].$$

which is now equivalent to S_B .

5.3 Discussion

We propose to calculate the edit distance as a linear combination of the transition cost, where the coefficients of the linear combination is estimated as a best fit with hand classified images. This estimation has not yet been performed since the edit distance algorithm has not yet been completed.

We suggest that the coefficients are all first set to one and then adjusted according to the discrepancy between the matching results and the hand classified images until the best fit is obtained.

6 Medical Image Database

This Section describes an image database of the craniofacial area. The database consists of four three-dimensional volumetric CT images obtained from four patients with various abnormalities and 113 masks of the teeth segmented manually. The database is aimed at assessing the performance of the image matching algorithm based on the Multi-Scale Singularity Trees (MSSTs). The performance of the matching algorithm can be measured by the rate of correct matching between teeth of the same kind at different poses and locations.

6.1 CT Images and Tooth Masks

The image database consists of four three-dimensional volumetric CT images obtained from four patients with various abnormalities and 113 masks of the teeth segmented manually. The teeth are segmented using the $\nabla Vision$ [2], the multi-scale interactive segmentation tool, installed at the 3D-Lab. $\nabla Vision$ is a joint development between researchers from the IT University of Copenhagen and the University of Copenhagen. The program has been develop under the project "Computing Natural Shape" and as a project in the now closed company Generic Vision.

All images are stored in the ANALYZETM 7.5 file format[1] with big-endian encoding. According to the ANALYZETM 7.5 file format, two separate files with the same name but different endings are used for an image, the header ended with `.hdr` and the raw data ended with `.img`. The CT images are stored in different directories namely `p3`, `p4`, `p5`, and `p6` together with the segmented tooth masks named with number. The $\nabla Vision$ project files ended with `.prj` can also be found in the corresponding directories.

The volumetric craniofacial CT images use a 32-bit real number for a voxel. The segmented tooth masks use an 8-bit integer for a voxel, the value zero for the background and the value 254 for the object. The properties of the images and the number of the masks of segmented teeth for each CT image are listed in Table 1. Figure 6. shows the surfaces of the segmented teeth of the four CT images.

Name	Dimension	Size(mm)	Number of Segmented Teeth
p3	$399 \times 232 \times 107$	$0.37 \times 0.37 \times 0.80$	44
p4	$302 \times 196 \times 84$	$0.49 \times 0.49 \times 1.00$	32
p5	$305 \times 199 \times 160$	$0.49 \times 0.49 \times 0.60$	8
p6	$288 \times 238 \times 73$	$0.47 \times 0.47 \times 1.00$	29
Total Number of Segmented Teeth			113

Table 1: Properties of the CT images in the database

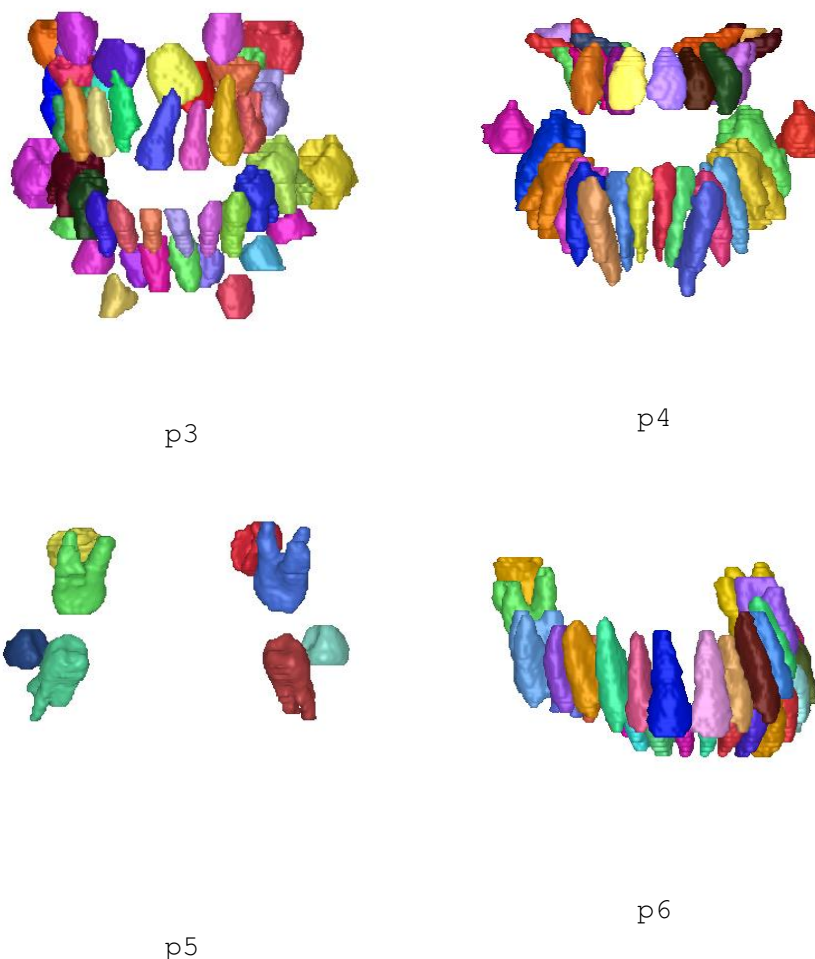


Figure 6: Surfaces of the segmented teeth of the four craniofacial CT images

6.2 Computational Task

The database is aimed at assessing the performance of the image matching algorithm based on the Multi-Scale Singularity Trees (MSSTs)[15, 16]. The medical task interested is the detection and location of teeth in CT/MR images, which is useful for a number of interesting problems such as studying the statistics of tooth shapes, studying the growth pattern, and analyzing the configuration of teeth for possible corrections.

For each segmented tooth, an MSST is computed. The similarity between two segmented teeth is measured as the total cost of edit operations that transform one MSST to the other. This problem is commonly known as the Tree Edit Distance (TED) problem[4].

In order to find the best match, the enquiry tooth have to be matched against all other teeth in the database. Fortunately, the MSSTs of all the segmented teeth can be pre-computed and later reused in the matching process. As a sub-goal, if the matching algorithm permits, MSSTs of teeth will be used to match into the MSSTs computed from the craniofacial CT images of a larger area to evaluate the possibility of sub-object extractions.

6.3 Assessment

The set of plastic teeth prepared by human experts can be used as the ground truth. The performance of the matching algorithm can be measured by the rate of the correct matching between teeth of the same kind at different poses and locations. The matching results will include the best match and all other matches ranked according to the distance or similarity.

7 Concluding Remarks

The selection of landmarks and the energy functional define the tessellation of energy partitions. The nesting of energy partitions in scale-space creates hierarchies which are captured and presented using tree structure called Multi-Scale Singularity Trees (MSSTs). MSSTs capture the deep structure of images. MSST transitions observed experimentally from a series of generated images are categorized into three categories: changes of positions, changes of relations, and changes of connections. We have not yet clearly proved the completeness of the obtained transitions.

MSSTs and introduced MSSSs are equivalent. Both structures topologically represent the deep structure of images. It's however easier and more intuitive to study and define edit operations on MSSSs since their transitions are simpler[16]. We present the two set of edit operations together with their cost estimates each on Extrema-Based MSSSs and Saddles-Based MSSSs. Since the edit operations on Saddles-Based MSSSs are simpler, we prefer the Saddles-Based MSSSs as a multi-scale representation of images suitable for image matching applications.

Creation events and annihilation events between pairs of critical points that cannot be tracked down to the first-scale image are ignored. These catastrophic events, however generic, are not frequently occurred. Including these catastrophic events in MSSTs/MSSSs and defining edit operations that can handle these events might improve the matching results.

MSSSs are preferable for image matching applications however one of the advantages of MSSTs over MSSSs is the hierarchical grouping of energy partitions suggested by its sub trees. We believe that exploiting this advantage might be useful for sub-object extraction, e.g. matching MSSS of a sub-object with MSSS equivalent to a sub tree cut from the MSST of the whole image.

References

- [1] *ANALYZETM 7.5 File Format*. Biomedical Imaging Resource, Mayo Foundation.
- [2] *∇Vision*. Generic Vision.
- [3] P. A. Arbelaez and L. D. Cohen. "The Extrema Edges". In *Scale Space 2003, LNCS 2695*, pages 180–195, 2003.
- [4] P. Bille. Report on Known Algorithm for Tree Matching. Technical report, Deliverable No.5, DSSCV, IST-2001-35443, 21. March 2003.
- [5] James Damon. Local Morse theory for Gaussian blurred functions. In Jon Sporring, Mads Nielsen, Luc Florack, and Peter Johansen, editors, *Gaussian Scale-Space Theory*, chapter 11, pages 147–163. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [6] R. Deriche. Recursively Implementing the Gaussian and its Derivatives. In V. Srinivasan, Ong Sim Heng, and Ang Yew Hock, editors, *Proceedings of the 2nd Singapore International Conference on Image Processing*, pages 263–267. World Scientific, Singapore, 1992.
- [7] T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrotechnical Laboratory*, 26:368–388, 1962. (in Japanese).
- [8] R. Kimmel, N. Sochen, and R. Malladi. From High Energy Physics to Low Level Vision. In *Scale-Space Theory in Computer Vision*, Utrecht, The Netherlands, 1997.
- [9] J. J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363–370, 1984.
- [10] Arjan Kuijper. *The deep structure of Gaussian scale space images*. PhD thesis, Image Sciences Institute, Institute of Information and Computing Sciences, Faculty of Mathematics and Computer Science, Utrecht University, 2002.
- [11] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(6):529–541, 1990.
- [12] O.F.Olsen and M.Nielsen. Multi-scale gradient magnitude watershed segmentation. In A. Del Bimbo, editor, *ICIAP '97 - 9th Int. Conf. on Image Analysis and Processing*, volume 1310 of *Lectures Notes in Computer Science*, pages 6–13. Springer, September 1997.
- [13] B. Platel. Multiscale Hierarchical Segmentation. Technical Report BMT-Report no. 2002-04, Department of BioMedical Engineering, Technical Universitet of Eindhoven, 2002.
- [14] J. A. Sethian. Fast Marching Methods. *SIAM Review*, 41(2):199–235, 1999.
- [15] K. Somchaipeng, J Sporring, S. Kreiborg, and P. Johansen. Software for Extracting Multi-Scale Singularity Trees. Technical report, Deliverable No.8, DSSCV, IST-2001-35443, 15. September 2003.
- [16] K. Somchaipeng, J Sporring, S. Kreiborg, and P. Johansen. Some Transitions of Extrema-Based Multi-Scale Singularity Trees. Technical report, DSAGM'04, Dept. of Computer Science, University of Copenhagen (DIKU), August 2004.

REFERENCES

- [17] B. van Ginneken and B. M. ter Haar Romeny. Applications of locally orderless images". In *Scale Space 1999*, 1999.
- [18] L.J. van Vliet, I.T. Young, and P.W. Verbeek. Recursive Gaussian Derivative Filters. In *Proceedings of the 14th International Conference on Pattern Recognition ICPR'98*, volume 1, pages 509–514., Brisbane, Australia, 1998. IEEE Computer Society Press.
- [19] J. Weickert, K. J. Zuiderveld, B. M. ter Haar Romeny, and W. J. Niessen. Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering. In *Proceedings of the 4th International Conference on Image Processing*, volume 3, pages 396–399, Santa Barbara, CA, USA, October 26–29 1997. IEEE Computer Society Press.
- [20] A. P. Witkin. Scale–space filtering. In *Proc. 8th Int. Joint Conf. on Artificial Intelligence (IJCAI '83)*, volume 2, pages 1019–1022, Karlsruhe, Germany, August 1983.