Multi-Scale Singularity Trees

(MSSTs)

Kerawit Somchaipeng

This thesis was set in LaTeX by the author.

Multi-Scale Singularity Trees

(MSSTs)

Kerawit Somchaipeng

This is a Ph.D. thesis from the Danish Ph.D. degree in Computer Science at The Department of Computer Science, University of Copenhagen, Denmark. Supervisors have been Jon Sporring, Peter Johansen, and Sven Kreiborg, and the advisory board consisted of ...

Copenhagen, Date

Contents

Preface

Three years in Copenhagen were, so far, undoubtedly the greatest and the most demanding period of my life. The opportunity to study and work in this beautiful city is priceless and the experience was far beyond my initial expectation.

First of all, I would like to thank Prof. Jon Sporring who gave me this life-time opportunity. The thesis could not possibly be done without his supportive advices and constructive comments. His kindness and patience during extensive discussions and extended working hours are also highly appreciated. I would also like to thank Prof. Peter Johansen and Prof. Sven Kreiborg for their kindness, support, and the excellent working environments they provided.

I would like to give my thanks to all the members of the DSSCV Project especially Prof. Ole Fogh Olsen, Prof. Arjan Kuijper, Frans Kanters, Bram Platel, Philip Bille, Prof. Luc Florack, and Prof. Mads

Nielsen for their invaluable time and patience during many inspiring and productive discussions. I thank everybody at the 3D-Lab and the Image Group for their beautiful friendship and heartfelt hospitality, especially Prof. Kenny Erleben, who co-authored in some of the publications. My life in Copenhagen would be much harder without help from Thai people who study and live in Copenhagen. I owe you, my friends.

Finally, I want to give my special thanks to my mother and father for their unconditional love, support, and understanding.

The research described in this thesis was carried out at the 3D-Lab, School of Dentistry, Dept. of Pediatric Dentistry, University of Copenhagen and is part of the DSSCV project supported by the IST Programme of the European Union (IST-2001-35443).

The following is the list of my publications published during the Ph.D. study. These publications are what the thesis is built on. Comments, suggestions, and questions from the reviewers of these publications are greatly appreciated.

Papers in Conference Proceedings

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, Transitions of Multi-Scale Singularity Trees, In *Proceedings of the International Workshop on Deep Structure, Singularity, and Computer Vision 2005*, Maastrict, The Netherlands, June 2005.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, Multi-Scale Singularity Trees: Soft-Linked Scale-Space Hierarchies, In *Proceedings of the 5th International Conference on Scale-Space 2005*, pp 97 - 106, Hofgeismar, Germany, April 2005.

- K. Somchaipeng, K. Erleben, and J. Sporring, A Multi-Scale Singularity Bounding Volume Hierarchy, In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'05)*, pp 179 - 186, Plzen, Czech Republic, January 2005.

Technical Reports

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Performance Evaluation of Tree Object Matching*, Technical report, DSSCV project, IST-2002-35443, September 2005.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Report on Matching 3D Image Structures by their MSSTs in a Given Application*, Technical report, DSSCV project, IST-2002-35443, Deliverable No. 33, September 2005.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Report on Assessment of MSSTs for 3D Matching Application*,

Technical report, DSSCV project, IST-2002-35443, Deliverable No. 21, September 2004.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Some Transitions of Extrema-Based Multi-Scale Singularity Trees*, In S. I. Olsen, editor, DSAGM 2004, pp 48-55, Technical report No. 04-10, Dept. of Computer Science (DIKU), August 2004.

- K. Somchaipeng, K. Erleben, and J. Sporring, *A Multi-Scale Singularity Bounding Volume Hierarchy*, Technical report No 04-08, Dept. of Computer Science (DIKU), 2004.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Software for Extracting 3D MSSTs*, Technical report, DSSCV project, IST-2002-35443, Deliverable No. 8, September 2003.

Others

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Some Transitions of Extrema-Based Multi-Scale Singularity Trees*, The DTU Visiondays'04, Copenhagen, Denmark, June 2004.

- K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen, *Image Matching using MSSTs*, Pre-Seminar in connection with MICCAI 2006, IT University of Copenhagen, Copenhagen, Denmark, August 2005.

Kerawit Somchaipeng

Chapter 1

Introduction

What we call seeing a thing clearly, is
only seeing enough of it to make out
what it is; this point of intelligibility
varying in distance for different
magnitudes and kind of things, while
appointed quantity of mystery remains
nearly the same for all.
J. Ruskin

The concept of scales is applied everywhere, explicitly or implicitly. Images of the same scene might look completely different when measured at different scales. A famous painting by Salvador Dali[1] titled

[1]Salvador Dali (1904-1989) is considered as the greatest artist of the surrealist

"Gala Contemplating the Mediterranean Sea which at Twenty Meters becomes a Portrait of Abraham Lincoln"[2], painted in 1976, vividly demonstrates the significance of scales. The painting of a naked woman gazing out the window suddenly becomes a portrait of Abraham Lincoln when it is placed at a certain distance from the viewer. Without relevant prior knowledge, we do not know what scale is better than or should be preferred to another, i.e. it is not possible to argue conclusively that a naked woman is more likely, what Dali actually wanted us to see, than the portrait of Abraham Lincoln. One might be able to give plausible reasons why one is to be prefered. However, that is only possible because he or she has already put some prior knowledge into action. In fact, a naked woman or a portrait of Abraham Lincoln is not any more nor less superior than every thing we perceive at scales in between, and they should be equally and similarly treated. Without prior knowledge about the scene, all scales are equally important. Pushing the discussion to the extreme, one can argue that we never see anything clearly, ever! The striking fact is made obvious in an Internet interactive movie "Secret Worlds: The Universe Within"[3]

In practice, the range of scales is limited by the field of view (the *outer scale*) and the resolution of the measuring device (the *inner scale*). For digital images the outer scale is the size of the image and the inner scale is the size of the pixel. In western literature, one of the first systematic ways of dealing with multi-scale is the *image pyramids* [40], which is a stack of images of the same scene with decreasing resolutions. The idea was later formalized and became know as the *Gaussian scale-space* [29, 98, 39, 84][4]. Gaussian scale-space is a stack of images where each image is a blurred version of the former. As the name suggests, the blurring is done by convolving the original image with Gaussian kernels of increasing widths. Several different sets of basic constraints, the so-called scale-space axioms, have been proposed and all suggested the Gaussian as the unique blurring kernel family [95, 49][5]. The Gaussian kernels and their derivatives have also been found to resemble the receptive fields in the retina at the very earliest stages of human visual system [51, 47, 50].

When an image is subjected to Gaussian smoothing, shapes in the image change, and their distinct details merge and sometimes split. The splitting of details in blurred images might at-first seem counter-intuitive. However it does occur in daily life if one pays a little attention. One classic example is the splitting of a leaf from the branch

art movement and one of the greatest masters of art of the twentieth century.

[2]The painting is part of the permanent collection at the Salvador Dali Museum in St. Petersburg and is one of the most reproduced painting by Salvador Dali.

[3]http://micro.magnet.fsu.edu/primer/java/scienceopticsu/powersof10/

[4]Because [29] was published in 1962 in Japanese, it was unknown to the western literature until 1999 [96].

[5]Alternative scale-space representations including Poisson scale-space and α-scale-space, which are related to Gaussian scale-space via a one-parameter class of operationally well-defined intermediate representations generated by a fractional power of (minus) the spatial Laplace operator. They are discussed in [14]

of a tree when it is observed from a certain distance. The relation of image structures at different scales is refereed to as the deep structure of the image [39, 42], which will be discussed extensively in the thesis. Singular points in scale-space, where pairs of critical points merge or split, are called catastrophe points. For one dimensional signals, it has been proved that the original signals can be completely reconstructed from the positions of these catastrophes [32, 31]. For images of higher dimensionality e.g. 2, the problem is much more complicated and perfect reconstruction has not yet been found [33, 34, 30].

A few methods for constructing image hierarchies representing the deep structures of images in scale-space have been described in the literatures [98, 46, 42, 64, 60]. In this thesis, we propose and later experimentally evaluate a novel scale-space image hierarchies called Multi-Scale Singularity Trees (MSSTs). MSSTs are powerful and unique because of their simple binary structure and straightforward construction method, and most importantly, their soft-linked nature, where all connections in MSSTs are accompanied by their strengths. Using the matching algorithm developed specifically for MSSTs to fully exploit their unique properties, the usefulness and performance of MSSTs are further shown experimentally.

1.1 The DSSCV Project

The Deep Structures, Singularities, and Computer Vision (DSSCV) project was a multi-organization project supported by the IST Programme of the European Union (IST-2001-35443). The project was started in October 2002 and ended in September 2005. Its main purpose is to develop, both in theory and in practice, sophisticated image and shape representations, and efficient algorithms for solving specific computer vision tasks.

The members of the project consisted of several experts in the fields, post-docs and Ph.D. students from four institutes located in three European countries: Denmark, United Kingdom, and The Netherlands. Using singularity theory, scale-space theory, and algorithmics, the project attacked the computer vision problems from different yet related angles.

The author's specific jobs in the DSSCV project has been to develop a tree structure image representation called Multi-Scale Singularity Trees (MSSTs) that represent the deep structures of images, a matching algorithm based on the MSSTs, and to evaluate the performance of the matching method on specific image databases. We emphasize that the goal has not been to find a fast algorithm, but to quantitatively study the accuracy and demonstrate the advantages of MSSTs in image matching.

1.2 A Guide to the Rest of the Thesis

This thesis consists of a collection of selected articles the author published during his Ph.D. study. Published articles are updated with new developments and corrections. Each article is rearranged according to the format of the thesis, and is included in the thesis as a chapter. The author decided to keep each chapter readable on its own. This means that there might be some overlaps between each chapter, e.g. multiple introductions to Gaussian scale-space. In this thesis, ther term "scale-space" is used loosely to mean "Gaussian scale-space".

The thesis is organized in two parts. The first part of the thesis mainly concerns the theoretical aspects of the Gaussian scale-space and the proposed Multi-Scale Singularity Trees. The second part of the thesis exploits the concepts introduced in the first part in applications.

Part I contains two chapters where: Chapter 2 formally introduces the Multi-Scale Singularity Trees (MSSTs) and explains in detail the method that extracts and construct MSSTs from images. Two kinds of MSSTs are presented namely the Extrema-Based MSSTs and Saddle-Based MSSTs. Chapter 3 investigates the topological changes of MSSTs under images perturbations. The changes or transitions of the MSSTs are categorized into groups. The impact of each transition is described and finally the discussion on the stability of Extrema-Based MSSTs and Saddle-Based MSSTs is presented.

Part II contains two chapters where: Chapter 4 presents two versions of the image matching method that exploits the unique properties of MSSTs. The performance and various aspects of the matching method are evaluated and presented using three publicly available image databases. The performance comparison with state-of-the-art methods is also presented. Chapter 5 demonstrates how MSSTs can be used to constructed bounding volume hierarchies called Multi-Scale Singularity Bounding Volume Hierarchies (MSS-BVHs) for collision detection in physics-based computer animations.

Appendix A gives a mini tutorial on the software, implemented using the methods described in the thesis, that extracts, visualizes, and matches MSSTs.

Part I

Theoretical Aspects

Chapter 2

Multi-Scale Singularity Trees[1]

The challenge is to understand the image
really on all the levels simultaneously,
and not as an unrelated set of derived
images at different levels of blurring.
J.J. Koenderink

We consider images as manifolds embedded in a hybrid of a high di-

[1] An earlier version of this work titled "Multi-Scale Singularity Trees : Soft-Linked Scale-Space Hierarchies" has been published in Proceedings of the 5th International Conference on Scale-Space 2005 [80].

mensional space of coordinates and image features. Using the energy functional and mathematical landmarks proposed in this chapter, we partition an image into segments. The nesting of image segments occurring at catastrophes in scale-space is used to decide their relations and construct an image hierarchy called Multi-Scale Singularity Tree (MSST). We propose two kinds of mathematical landmarks: extrema and saddles. Unlike all other similar methods described hitherto, our method produces soft-linked image hierarchies in the sense that all possible connections are suggested along with their energies. The additional information makes possible for directly estimating the stability or strength of each connection and hence the costs of possible transitions. Aimed applications of MSSTs include multi-scale pre-segmentation, image matching, sub-object extraction, and hierarchical image indexing and retrieval.

2.1 Introduction

We are interested in the development of a multi-scale image representation which is expected to be invariant under certain image transformations and small image perturbations. Images of objects of any complexity, when observed at a certain distance or measured at a certain scale, eventually reduce to an indistinguishable blob. Small structures in the image are merged at low scales, larger structures at

higher scales. The splitting also sometimes occurs. The collection of these events happened at all scales in scale-space is commonly referred to as the *deep structure* of images.

In this chapter, We propose a binary tree structure to represent the deep structure of images, where small image structures at low scales are represented by nodes located low in the tree and large image structures at high scales are represented by high nodes in the tree. Binary trees are of interest because their edit distances are possibly convenient and fast distance measure for images they represent.

In [46], the isophotes of the intensity levels at catastrophes were used to partition an image into regions called extremal regions. The nesting of extremal regions were then used to construct an image hierarchy. Scale-space saddles were first extensively discussed in [42]. It has been shown that the iso-surfaces of the image intensity at those points provide a scale-space hierarchy which may be used to nest extrema in a tree. In [60], multi-scale image hierarchies built using the nesting of gradient magnitude watershed in scale-space have been used successfully for interactive segmentation.

All of the above mentioned multi-scale image hierarchy building methods produce trees that we consider to called hard-linked. Our method produces soft-linked trees where each connection in the trees is accompanied by its strength.

Using tree structures as image representations, image matching problems can then be rephrased as tree matching problems. It reduces computer vision problems to tree manipulation problems and transforms the relatively unfamiliar problems to the well-understood and profoundly-investigated mathematical problems. In practice, using tree structures also give possibilities to naturally compromise between speed and accuracy.

Possible applications of Multi-Scale Singularity Trees (MSSTs) include multi-scale pre-segmentation, image matching using MSSTs, sub-object extraction, hierarchical image retrieval in large image databases, etc.

2.2 Gaussian Scale-Space

The $N+1$ dimensional Gaussian scale-space, $L : \mathbb{R}^{N+1} \to \mathbb{R}$, of an N dimensional image, $I : \mathbb{R}^N \to \mathbb{R}$, is an ordered stack of images, where each image is a blurred version of the former [29, 98, 39]. The blurring is performed according to the diffusion equation,

$$\partial_t L = \nabla^2 L \,, \tag{2.1}$$

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 $\partial_x^2 + \partial_y^2 + \partial_z^2$. Boundary condition: $L(\cdot, 0) = I(.)$. The Gaussian kernel is the Green's function of the heat diffusion equation, i.e.

$$L(\cdot; t) = I(\cdot) \otimes g(\cdot; t) \,, \tag{2.2}$$

$$g(x; t) = \frac{1}{(4\pi t)^{N/2}} e^{-x^T x/(4t)} \,, \tag{2.3}$$

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 [51] using,

$$\sigma(m) = \sigma_0 b^m \,, \tag{2.4}$$

for some base b. Since differentiation commutes with convolution and the Gaussian kernel is infinitely differentiable, differentiation of images in scale-space is conveniently computed as,

$$\partial_{x^n} L(\cdot; t) = \partial_{x^n} \left(I(\cdot) \otimes g(\cdot; t) \right) = I(\cdot) \otimes \partial_{x^n} g(\cdot; t) \,. \tag{2.5}$$

Alternative implementations of the scale-space are multiplication in

the Fourier Domain, finite differencing schemes for solving the heat diffusion equation, additive operator splitting [97], and recursive implementation [12, 91].

Each method has different advantages and disadvantages. We prefer the spatial convolution, since it guarantees 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 kernel.

Although the dimensionality of the constructed scale-space is one higher than the dimensionality of the original image, *critical points*, in an image at each scale are always points. The critical points treated in this chapter are extrema and saddles, $\partial_x L = \partial_y L = 0$, and the critical points are classified by the eigenvalues of the *Hessian matrix*, the matrix of all second derivatives, computed at the critical 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. Zero eigenvalues are non-generic.

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

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 perturbations of the image. There are only two types of generic catastrophic events in scale-space namely pairwise *creations* and pairwise *annihilations* [11]. It has further been shown that generic catastrophic events only involve 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 maximum $(-,-)$ and a saddle $(+,-)$, etc. A detailed discussion of a method for detecting critical paths and catastrophe points in scale-spaces can be found in [77].

2.3 Building Scale-Space Hierarchies

There are already a few scale-space methods that construct image hierarchies of two-dimensional images proposed in the literatures so far [46, 42, 64]. An attempt to construct image hierarchies from the deep structures of three-dimensional images was proposed in [60] followed by [77, 79, 78]. The latter scheme, which will be described extensively here, produces rooted ordered binary trees called *Multi-Scale*

Singularity Trees (MSSTs) with catastrophe points as nodes.

In order to guarantee that the produced structures are always trees, our method only considers the linking of landmarks that exist at the original image. Only annihilations are considered and creations are systematically ignored. These creation events are generic however not frequent. Creation events could actually be included in the structure in the same manner but the method would inevitably produce graphs rather than trees.

The method can be used to construct scale-space hierarchies of images of any dimensionality, assuming that critical paths and catastrophe points can be correctly detected. Our current implementation is capable of constructing MSSTs of two- and three-dimensional images.

2.3.1 Energy Functional and Energy Partitions

Given an image and a set of landmarks, we would like to partition the image into segments so that each image segment contains exactly one landmark.

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 in the domain from the landmark $\vec{e}$ to the point $\vec{x}$, and where γ is parameterized using Euclidean arc-length (See section 3.3). We define the energy $E_{\vec{e}}(\vec{x})$ with respect to an extremum $\vec{e}$ evaluated at $\vec{x}$ as,

$$E_{\vec{e}}(x) = \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_0^P \sqrt{(\alpha - 1) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} dp\,, \quad (2.6)$$

for some $0 \leq \alpha \leq 1$. Consider images as manifolds embedded in a high dimensional space, an N dimensional intensity image becomes an N dimensional manifold embedded in the hybrid $N + 1$ dimensional space of coordinates and image intensity, the "space-feature" [36]. In this case the only feature is the intensity or the zeroth jet space. The energy functional can be defined for higher order jet space images, color images, or locally orderless images with scale-space histograms to handle texture [90], if a metric in the feature space is given.

For two-dimensional images, an image may be considered a height plot, and the energy at any point in the image with respect to a landmark can be thought of as the minimum weighted distance in 3-dimensional space, traveling up and down, from the landmark to that point.

The parameter α can be set to alter the emphasis of the energy functional between image space and image intensity. When $\alpha = 1$, the energy functional also known as the path variation [1], which is

a generalization of the total variation. The path variation depends solely on the image intensity and is invariant to affine transformation of the underlying space. Moreover, the energy is co-variant with scaling of the image intensity. When $\alpha \to 0$, the energy functional will increasingly depend on the spatial distance, and therefore become increasingly localized in space.

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, i.e.

$$S_i = \{\vec{x} \in \Omega | E_{\vec{e}_i}(\vec{x}) < E_{\vec{e}_i}(\vec{x}), \forall \vec{e}_j \in \mathcal{E}, i \neq j\}. \tag{2.7}$$

An approximation of the energy map $E_{\vec{e}_i} : \Omega \to \mathbb{R}^+$, which gives the energy computed at every point in the image with respect to the landmark $\vec{e}_i$, can be efficiently calculated using the *Fast Marching Method* [87, 70, 71, 72, 73, 74] to be discussed in section 2.3.2.

The tessellation of the image segments obtained depends on the selection of the landmarks and the energy functional. Mathematical landmarks such as extrema seem to be a natural choice, since they are directly linked to the image content, i.e. significant features in the image usually contain at least one such points. Moreover, they can be easily and automatically detected, and the behavior of these

critical points in scale-space is well studied. If extrema are used as landmarks, the produced image hierarchies are called Extrema-Based MSSTs. Another candidate for landmarks are the saddles, and the produced image hierarchies are called Saddle-Based MSSTs. Both versions of the MSSTs will be discussed in section 2.3.3.

2.3.2 The Fast Marching Method

One of the typical problems in graph theory is the minimum-cost path problem, which is to find the least expensive path connecting two specific vertices of a directed or undirected graph. The minimum-cost path problem can be formulated as the solution to the non-linear Eikonal equation

$$| \nabla u(x) | = F(x), \quad F(x) > 0\,, \tag{2.8}$$

where cost function $F(x)$ is typically given and can be interpreted as the delay, while the accumulated cost $u(x)$ can be thought of as the arrival time. If the cost function $F(x)$ is set to a constant, the solution to the Eikonal equation is also known as the distance transform.

The Fast Marching Method [87, 70, 71, 72, 73, 74] is a highly efficient numerical technique that solves the Eikonal equation robustly. The method is based on the causality condition that the arrival time at any point depends only on the neighbors that have smaller arrival

times.

For the energy map generation problem, the accumulated cost $u(x)$ is the energy and the cost function $F(x)$ is given by

$$F(x) = \sqrt{(\alpha - 1) + \alpha \mid \nabla I \mid^2}, \tag{2.9}$$

where I is the image, and α is the weighting parameter of the energy functional (2.6). The Fast Marching Method categorize the grid points into 3 sets namely, *Accepted*, *Trial*, and *Far*. Initially, the starting point is added to *Accepted*, its 4-neighbors are added to *Trial*, and all other points are added to *Far*. Secondly, the energy of points in *Far* are set to ∞, and the energy of the starting point is set to 0. Next, given the cost function, compute the energy for points in *Trial*. Then the loop begins:

1. Find the point in *Trial* that has the smallest energy and move it to *Accepted*.

2. For each of its 4-neighbors that is not in *Accepted*:

   (a) Compute and update its energy if the computed energy is less than its current energy.

   (b) If the point is in *Far*, move it to *Trial*.

3. Repeat 1, 2, and 3, until all points are in *Accepted*.

The Fast Marching Method continually expands the *Accepted* outward with the narrow band of *Trial* and diminishes *Far* until all points are in *Accepted* and their energy are known.

The efficiency of the Fast Marching Method lies in the fast way of finding the grid point in *Trial* that has the smallest energy. The min-heap data structure [69], is used to store points in *Trial* keeping the cost of locating the grid point with minimum energy minimal.

2.3.3 Multi-Scale Singularity Trees

MSSTs are constructed by connecting annihilating catastrophe points based on the nesting of image segments in the scale-space. Because of the only generic catastrophes are the pairwise interactions between critical points in scale-space and two catastrophes never occur at the same scale, MSSTs are always rooted ordered binary tree.

MSSTs consist of nodes and their relations. Each MSST node consists of three important components: The image segment that immediately covers the area of the image segment disappearing at the catastrophe. For algorithmically convenience we denote the covering image segment the *leftport*, the catastrophe for the *body*, and the disappearing image segment for the *rightport*. Because there is exactly one image segment associated with a landmark, and we choose the landmarks such that exactly one landmark disappears at an annihilation

catastrophe point, then exactly one image segment also disappears.

A node $S_{left}C_{body}S_{right}$ is generated if an image segment S_{right} disappears at the catastrophe C_{body} inside an image segment S_{left}. The inclusion is easily determined by calculating the energy map with respect to the catastrophe C_{body}: the image segment S_{right} is nested inside the image segment S_{left}, if the energy evaluated at the landmark of S_{left} is minimal among all landmarks existing at that scale.

As briefly mentioned above, MSSTs are rooted ordered trees. This implies that connecting a node to another node as the leftchild or as the rightchild are semantically different events.

MSSTs are built top-down starting from the highest catastrophe at the coarsest scale. A new node $N_{new} : S_{new,left}C_{new,body}S_{new,right}$ is connected as the leftchild of a node $N_i : S_{i,left}C_{i,body}S_{i,right}$ in the MSST, if the node N_i does not have its leftchild and $S_{new,left} = S_{i,left}$, or as the rightchild, if the node N_i does not have its rightchild and $S_{new,left} = S_{i,right}$. Connecting a new node to the MSST closes one port and open two ports. It can easily be seen that this process is deterministic. We will now describe the algorithms for creating the Extrema- and the Saddle-Based MSSTs.

Extrema-Based MSSTs

Assuming that critical paths and catastrophe points in the scale-space are correctly detected, then the tree building algorithm is as follows:

1. Set the root of the tree as $BC_{\infty}E_{last}$, where B denotes the border of the image, E_{last} denotes the last extremum in scale, and C_{∞} denotes the virtual catastrophe at scale infinity, where the border and the last extremum virtually annihilate.

2. At the highest unprocessed catastrophe C_{next} in scale, calculate the energy map with respect to the catastrophe and create a node $E_{cover}C_{next}E_{ann}$, where E_{ann} is the extremum that disappears at C_{next}, and the energy evaluated at the extremum E_{cover} is minimal among all extrema existing at that scale.

3. Link the new created node as the leftchild of a node in the tree that does not have its leftchild and where E_{cover} equals its leftport, or as the rightchild of a node in the tree that does not have its rightchild and where E_{cover} equals its rightport.

4. Repeat 2, 3, and 4, until all catastrophe points in the scale-space are processed.

An example of Extrema-Based MSSTs, together with the schematic drawing of the deep structure it represents, are shown in the left panel of Fig. 2.1.

Saddle-Based MSSTs

With slight modifications, a similar procedure is applied for constructing Saddle-Based MSSTs, however now we consider saddles as landmarks instead of extrema. The algorithm is as follows:

1. Set 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 that annihilates at the catastrophe C_{top}.

2. At the highest unprocessed catastrophe C_{next} in scale, calculate the energy map with respect to the catastrophe and create a node $S_{cover}C_{next}S_{ann}$, where S_{ann} is the saddle that disappears at C_{next}, and the energy evaluated at the saddle S_{cover} is minimal among all saddles existing at that scale.

3. Link the newly created node as the leftchild of a node in the tree that does not have its leftchild and S_{cover} equals its leftport or as the rightchild of a node in the tree that does not have its rightchild and S_{cover} equals its rightport.

4. Repeat 2, 3, and 4, until all catastrophe points in the scale-space are processed.

Notice that because all saddles in an image merge with all but one extremum and disappear at catastrophes, Saddle-Based MSSTs always have one node less than that of Extrema-Based MSSTs of the same image. The awkward border B is also eliminated. An example of Saddle-Based MSSTs, together with the schematic drawing of the deep structure it represents, are shown in the right panel of Fig. 2.1,

2.3.4 Energy Matrices

The *energy matrix* is a crucial part of an MSST. It is responsible for the soft-linked property of an MSST. The connections between catastrophes in an MSST are decided based on the energies computed at all landmarks existing at the scale of the catastrophes. To make the MSSTs "soft-linked", not only the most probable connection with the lowest energy is selected, but also all possible connections accompanied by their energies are recorded in the so-called energy matrix. Landmarks are sorted according to the scales of their corresponding catastrophes and their energies are stored in the energy matrix such that the element M_{ij} of the energy matrix M is the energy of the landmarks e_j associated with the catastrophe C_i.

It's the energy matrix that makes possible for the estimation of each MSST connection stability and the derivation of appropriate costs of MSST transitions. Examples of energy matrices of Extrema-Based MSSTs and Saddle-Base MSSTs are shown on the top-left of Fig. 2.2 and Fig. 2.3, respectively.

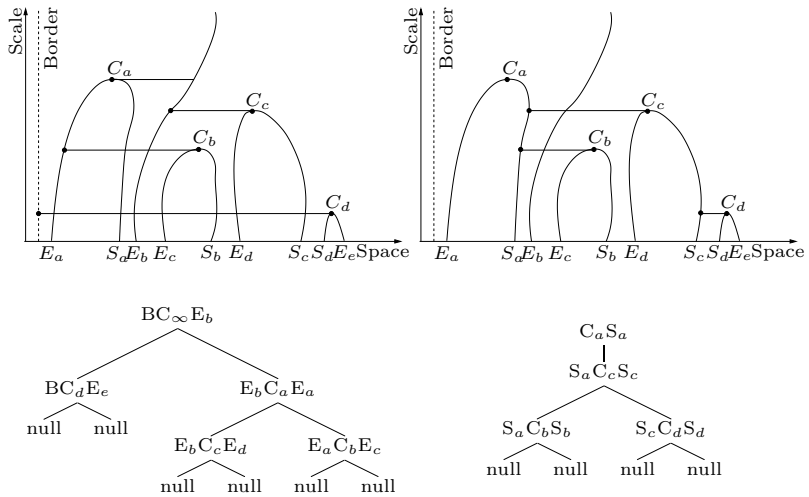

Figure 2.1: The top-left panel shows the schematic drawing of the deep structure of an image containing four extrema, E_a, E_b, E_c, and E_d and three saddles, S_a, S_b, and S_c. Consequently, assuming that there is no creations, there are four catastrophe points, C_a, C_b, C_c, and C_d in the scale-space. The horizontal lines denote the connections between the catastrophe points and the extrema with minimal energy. The Extrema-Based MSSTs corresponding to the deep structure is shown on the bottom-left panel. The top-right panel shows the deep structure of the same image but now with the horizontal lines showing the linking connections between the catastrophe points and the saddles with minimal energy. The corresponding Saddle-Based MSST is shown on the bottom-right panel.

2.3.5 MSST Examples of Real Images

The MSST building algorithm described above was implemented in C/C++ and a publicly available image database of human faces [68] was chosen to test our implementation. An example of Extrema-Based MSSTs of a man face is shown in Fig. 2.2. The Saddle-Based MSST of the same image is shown in Fig. 2.3. The scale-space is sampled exponentially using (A.1) defined in Appendix A with $\sigma_0 = 2.0$, $T = 0$, and $dT = 0.05$.

2.4 Discussion

It is important to mention the differences between two related stabilities: the stability of the catastrophe positions and the stability of the relations between them, which are represented by the connections in the MSSTs. Both entities effect the stability of the constructed hierarchy as a whole. Under small image perturbations, it can be shown experimentally that some catastrophes are more stable than others. In general, catastrophes located in image areas with a lot of structures tend to be more stable [65]. Using an appropriate stability norm, unstable catastrophe points can be eliminated before the hierarchy is constructed. On the other hand, the instability of their relations cannot and must not be discarded. They have to be carefully measured,

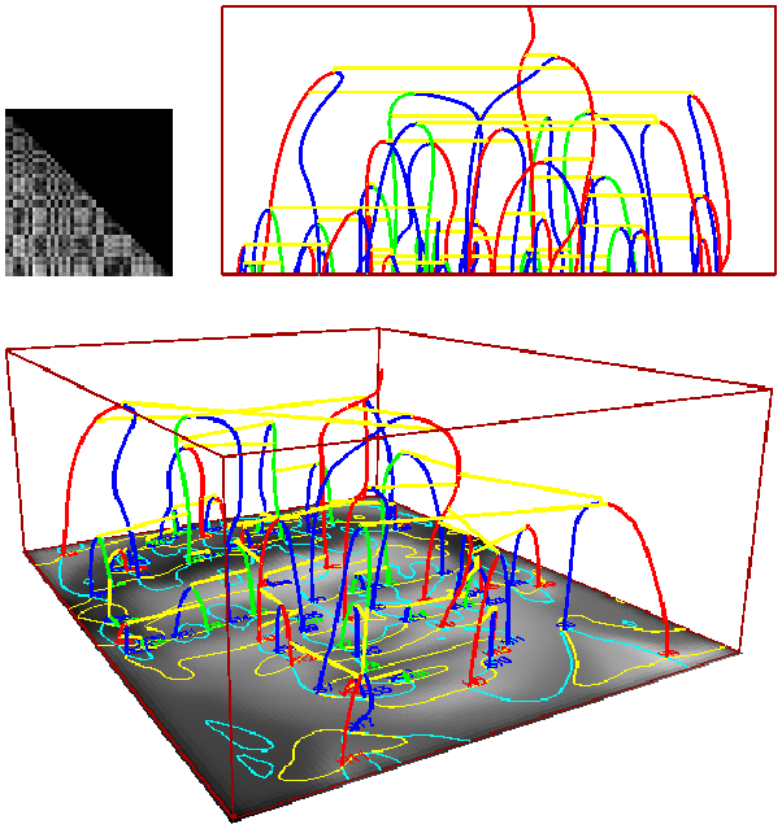

Figure 2.2: An example of Extrema-Based MSSTs. The energy matrix is shown on the top-left panel. On the top-right panel is shown the deep structure viewed horizontally to the image plane. The bottom panel shows a three-dimensional view of the deep structure with the first-scale image overlaid. The $\partial_x L = 0$ and $\partial_y L = 0$ curves are shown on top of the first-scale image in blue and in yellow, respectively. The extremal paths, minimal paths, and saddle paths are shown in red, green, and blue, respectively. The MSST connections are denoted by yellow horizontal lines.

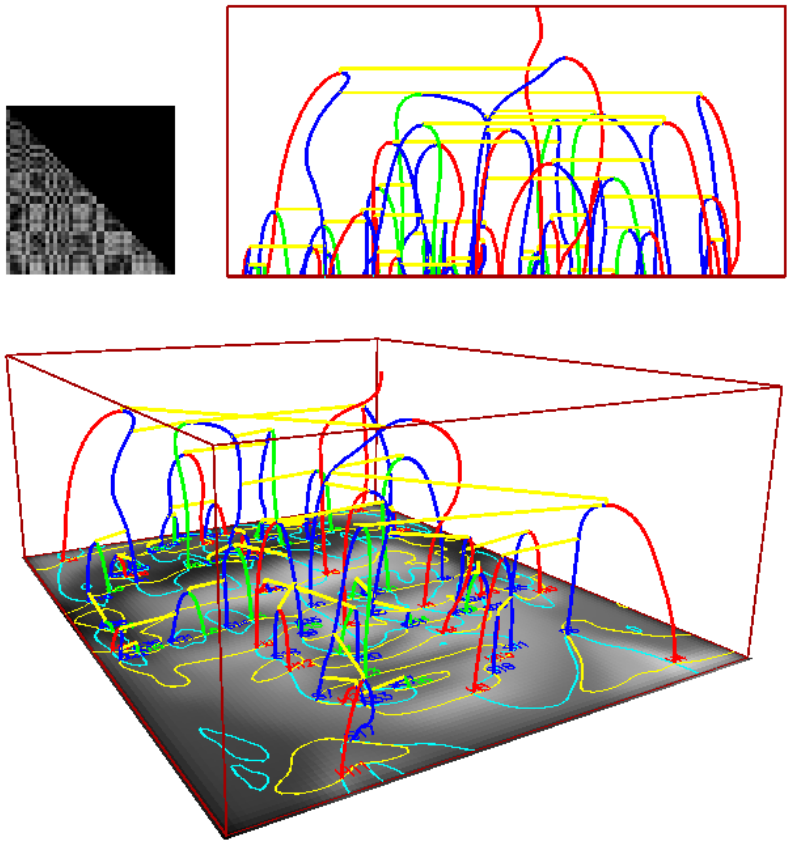

Figure 2.3: An example of Saddle-Based MSSTs. The energy matrix is shown on the top-left panel. On the top-right panel is shown the deep structure viewed horizontally to the image plane. The bottom panel shows a three-dimensional view of the deep structure with the first-scale image overlaid. The $\partial_x L = 0$ and $\partial_y L = 0$ curves are shown on top of the first-scale image in blue and in yellow, respectively. The extremal paths, minimal paths, and saddle paths are shown in red, green, and blue, respectively. The MSST connections are denoted by yellow horizontal lines.

if one wants to estimate the costs of possible topological changes in the hierarchies.

For example, one might want to know how stable a particular connection is in the produced hierarchy under slight perturbations of the original image. Consider an image that is similar or close to non-generic ones, e.g. having near-symmetrical structures, only slight perturbations of the original image could change the topology or structure of the produced hierarchy completely. For soft-linked hierarchies, the stability of each connection in the hierarchies can be directly estimated by looking at the distribution of the energies of all possible connections. If the energy of the best connection is much lower than all others, the connection is stable and not likely to change easily. On the other hand, if the energies of all connections are about the same, the connection is not stable and will easily switch even by slight perturbations. Because hard-linked hierarchy building methods described e.g. in [46, 60, 42, 64] naturally suggest only one best connection, there is no direct way of estimating the stability of connections of the produced hierarchies.

The soft-linked scale-space hierachy building method of MSSTs described here is theoretical extensible to produce trees of N-dimensional images. An example of MSSTs constructed from a simple 3D image is presented in section 5.3.

2.5 Conclusion

We prefer our scale-space hierarchy building scheme because in contrast to hard-linked hierarchies produced by all other scale-space methods described hitherto, where only one best connection in the hierarchy is suggested, our method produces soft-linked image hierarchies in the sense that all possible connections are suggested along with their strengths or energies. The connection with the lowest energy can be later selected in order to produce the best tree that represents the image for the presentation purpose. The information concerning the stability of each connection in an MSST is stored in the energy matrix.

The stability of each connection in MSSTs can be directly estimated, and the cost of topological changes or MSST transitions can be derived for further processing in applications e.g. image matching, image retrieval and indexing in a large image database.

Chapter 3

Transitions of Multi-Scale Singularity Trees[1]

Everything should be made as simple as possible, but not one bit simpler.

A. Einstein

Multi-Scale Singularity Trees (MSSTs) [80] are multi-scale image de-

[1] An earlier version of this work titled "Transition of Multi-Scale Singularity Trees" has been published in Proceedings of the International Workshop on Deep Structure, Singularity, and Computer Vision 2005 [83].

scriptors aimed at representing the deep structures of images. Changes in images are directly translated to changes in the deep structures; therefore transitions or topological changes in MSSTs. Because MSSTs can be used to represent the deep structure of images efficiently and are potentially useful for many applications, it is theoretically interesting and algorithmically important to carefully investigate and understand each of the transitions and their impacts. In this chapter, we present four kinds of MSST transitions and discuss the potential advantages of Saddle-Based MSSTs over Extrema-Based MSSTs. The study of MSST transitions presented in this paper is an important step toward the development of the image matching and indexing algorithms based on MSSTs.

3.1 Introduction

In scale-space theory [29, 98, 39], the relations between image structures at different scales is referred to as the deep structure of images [39, 48]. Based on the scale-space theory and the singularity theory [95, 11], the Extrema-Based and Saddle-Based Multi-Scale Singularity Trees (MSSTs) [80] representing the deep structure of images are constructed. Since MSSTs can be used to efficiently represent the deep structures of images, the investigation of their transitions, as the images are smoothly changed, is both theoretically interesting

and crucially important for the development of algorithms based on MSSTs. We will focus on developing an image matching algorithm. In that way, different images can be related to each other through the shortest series of transitions.

In this Chapter, we investigate the transitions of MSSTs by observing the movements and structural changes of critical paths and catastrophe points in scale-space images, as the images are smoothly changed. We begin by giving a short introduction to Gaussian scale-space in Sec. 3.2, introducing energy functional and energy partitions in Sec. 3.3, and presenting shortly the method of constructing MSSTs in Sec. 3.4. Transitions of MSSTs are then carefully categorized and presented using illustrative examples in Sec. 3.5. Four kinds of MSST transitions are presented, i.e. changes of catastrophe-extremum/saddle positions, changes of catastrophe-catastrophe relations, changes of catastrophe ordering, and changes of extremum-catastrophe connections. We present each transition and discuss its impacts using a simple example. In Sec. 3.6, the potential advantages of the Saddle-Based MSSTs over the Extrema-Based MSSTs are discussed and illustratively presented. Similar study of the transitions of the Pre-Symmetry Set has also been done by one of our European Project partners [43].

3.2 Gaussian Scale-Space

The $N+1$ dimensional Gaussian scale-space, $L : \mathbb{R}^{N+1} \to \mathbb{R}$, of an N dimensional image, $I : \mathbb{R}^N \to \mathbb{R}$, is an ordered stack of images, where each image is a blurred version of the former [29, 98, 39]. The blurring is performed according to the diffusion equation,

$$\partial_t L = \nabla^2 L \,, \tag{3.1}$$

where $\partial_t L$ is the first partial-derivative of the image in the scale direction t, and ∇^2 is the spatial Laplacian operator, which in two dimensions reads $\partial_x^2 + \partial_y^2$. The Gaussian kernel is the Green's function of the heat diffusion equation, i.e.

$$L(\cdot; t) = I(\cdot) \otimes g(\cdot; t) \,, \tag{3.2}$$

$$g(x; t) = \frac{1}{(4\pi t)^{N/2}} e^{-x^T x/(4t)} \,, \tag{3.3}$$

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 dimensionality of the image I, and $t = \sigma^2/2$, using σ as the standard deviation of the Gaussian kernel.

3.3 Energy Functional and Energy Partitions

Given an image and a set of landmarks in the image, we would like to partition the image into segments so that each segment contains exactly one landmark. Let $\Omega \subset \mathbb{R}^N$ be a compact connected domain. We define $I : \Omega \to \mathbb{R}^+$ as an intensity image, $\vec{e}$ as a landmark, and $\vec{x} \in \Omega$ as a point in the domain. Consider a set of continuous functions $\gamma : [0, P] \to \Omega$ for which $\gamma(0) = \vec{e}$ and $\gamma(P) = \vec{x}$. Write $\gamma \in \Gamma_{\vec{e}\vec{x}}$, where $\Gamma_{\vec{e}\vec{x}}$ is the set of all possible paths in the domain connecting landmark $\vec{e}$ to point $\vec{x}$, and where γ is parameterized using Euclidean arc-length. We define the energy $E_{\vec{e}}(\vec{x})$ with respect to a landmark $\vec{e}$ evaluated at $\vec{x}$ as,

$$E_{\vec{e}}(x) = \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_0^P \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, dp \,, \tag{3.4}$$

where $\alpha \in [0, 1]$ is a tunable weighting parameter between image intensity and space. Note that (3.4) is independent of the parameterization,

e.g. integrating with respect to $q : [0, P] \rightarrow [0, P]$ we find:

$$\begin{aligned} E_{\vec{e}}(x) &= \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p(q))}{dp} \mid^2 + \alpha \mid \frac{\partial I(\gamma(p(q)))}{\partial p} \mid^2} \, dq \\ &= \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \cdot \frac{dp}{dq} \mid^2 + \alpha \mid \nabla I \cdot \frac{d\gamma(p)}{dp} \cdot \frac{dp}{dq} \mid^2} \, dq \\ &= \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, \frac{dp}{dq} \cdot dq \\ &= \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_{0}^{P} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, dp \,. \end{aligned} \tag{3.5}$$

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

$$S_i = \{\vec{x} \in \Omega | E_{\vec{e}_i}(\vec{x}) < E_{\vec{e}_j}(\vec{x}), \forall \vec{e}_j \in \mathcal{E}, i \neq j\}. \tag{3.6}$$

An approximation of the energy map $E_{\vec{e}_i} : \Omega \rightarrow \mathbb{R}^+$, which gives the energy with respect to landmark $\vec{e}_i$ at every point in the image, can be efficiently calculated using the *Fast Marching Methods* [87, 70, 71, 72, 73, 74]. The resulting energy map is an approximation because isophotes in images generally do not coincide with the rectangular sampling grids in digital images. We are currently developing an energy map calculating algorithm that is specifically designed according

to this insight.

3.4 Multi-Scale Singularity Trees

Multi-Scale Singularity Trees (MSSTs) are constructed by connecting annihilation catastrophes in the scale-space images based on the nesting of energy partitions briefly described above [80].

It has been shown that creations and annihilations are the only generic events in two and higher dimensional scale-space images [11]. Loops of critical paths in scale-space formed by creations immediately followed by annihilations can also be observed in practice [42]. In order to preserve the preferable tree structure of the produced MSSTs, creations and loops in scale-space images are systematically removed:

1. Creations occurring on a critical path that can be traced back down to the original image together with their corresponding annihilations are pairwise removed and the whole critical path is considered as the critical path that originates from the original image and ends at the top-most annihilation.

2. Creations that are immediately followed by annihilations creating loops in scale-space and the critical paths involved in the loops are removed.

The connections between catastrophes are decided from the nesting

of image segments or energy partitions defined by the energy functional and selected mathematical landmarks. Because of the natural pairwise interactions between critical points in the generic scale-space images [11] and the tree building scheme to be presented in the following, resulting MSSTs are always rooted ordered binary trees [80]. An MSST consists of nodes and their relations and each MSST node has three components: (i) the *rightport* denoting the disappearing image segment, (ii) the *body* denoting the annihilation catastrophe where the nesting is decided, and (iii) the *leftport* denoting the image segment which immediately covers over the disappearing one. Because we choose landmarks such that exactly one landmark disappears at an annihilation catastrophe, and because there is exactly one landmark associated with each energy partition, then exactly one energy partition disappears at an annihilation catastrophe. The collection of these events creates nesting of image segments in scale-space and the linking in MSSTs.

A node $S_{left}C_{body}S_{right}$ is generated when an image segment S_{right} disappears at the catastrophe C_{body} inside an image segment S_{left} . The inclusion is easily determined by calculating the energy map with respect to the catastrophe C_{body}: the image segment S_{right} is nested inside the image segment S_{left}, if the energy evaluated at the landmark of S_{left} is minimal among all other landmarks existing at that scale. MSSTs are built top-down starting from the top annihilation catastrophe at the coarsest scale. A new node N_{new} : $S_{new,left}C_{new,body}S_{new,right}$ is connected as the *leftchild* of a node N_i : $S_{i,left}C_{i,body}S_{i,right}$, if the node N_i does not have its leftchild and $S_{new,left} = S_{i,left}$, or as the *rightchild*, if the node N_i does not have its rightchild and $S_{new,left} = S_{i,right}$.

It can easily be seen that this process is deterministic. When a node is added, one connection point is closed while two new connection points are opened, and free ports are always unique. Two versions of MSSTs were proposed in [80] namely Extrema-Based MSSTs and Saddle-Based MSSTs.

3.4.1 Extrema-Based MSSTs

Assuming that all critical paths and catastrophes in scale-space have already been detected, the Extrema-Based MSST building algorithm is as follows:

1. Set the root of the tree as $BC_{\infty}E_{last}$, where B denotes the border of the image, E_{last} denotes the last extremum in scale, and C_{∞} denotes the virtual catastrophe at scale infinity, where the last extremum virtually disappears inside the image segment of the border.

2. At the highest unprocessed catastrophe C_{next}, calculate the en-

ergy map with respect to that catastrophe and create a node $E_{cover}C_{next}E_{ann}$, where E_{ann} is the extremum that disappears at catastrophe C_{next}, and the energy evaluated at extremum E_{cover} is minimal among all extrema existing at that scale.

3. Link the new created node as the leftchild of a node in the tree that does not have its leftchild and where E_{cover} equals its leftport, or as the rightchild of a node in the tree that does not have its rightchild and where E_{cover} equals its rightport.

4. Repeat 2, 3, and 4, until all catastrophe points are processed.

The schematic drawing of the deep structure and its constructed Extrema-Based MSSTs are shown together in the left column of Fig. 3.1.

3.4.2 Saddle-Based MSSTs

A similar procedure is applied to construct Saddle-Based MSSTs, however now we consider saddles for landmarks instead of extrema. The algorithm is as follows:

1. Set 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 that annihilates at catastrophe C_{top}.

2. At the highest unprocessed catastrophe C_{next} in scale, calculate the energy map with respect to that catastrophe and create a node $S_{cover}C_{next}S_{ann}$, where S_{ann} is the saddle that disappears at catastrophe C_{next} and the energy evaluated at saddle S_{cover} is minimal among all saddles existing at that scale.

3. Link the new created node as the leftchild of a node in the tree that does not have its leftchild and S_{cover} equals its leftport or as the rightchild of a node in the tree that does not have its rightchild and S_{cover} equals its rightport.

4. Repeat 2, 3, and 4, until all catastrophes are processed.

The schematic drawing of the deep structure and its constructed Saddle-Based MSST are shown together in the right column of Fig. 3.1. Notice that since in Saddle-Based MSSTs, the virtual catastrophe $C_{\inf}$ is not relevant, Saddle-Based MSSTs always have one node less than those of Extrema-Based MSSTs representing the same images.

3.5 Transitions of MSSTS

To illustrate the possible topological changes or transitions of MSSTs an experiment was carried out. A series of generated images of three stationary and one moving Gaussian blobs is used in the experiment. A few samples of images selected from the series are shown in Fig. 3.2

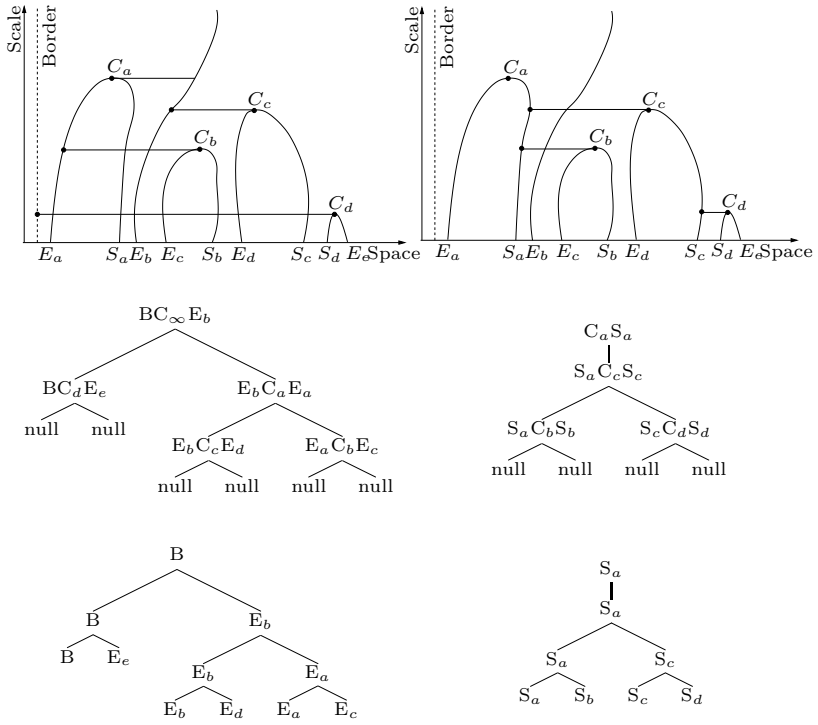

Figure 3.1: The schematic drawing of the deep structures and their corresponding Extrema-Based MSST and Saddle-Based MSST are shown in the first and the second row of the left and right column, respectively. Es denote extrema, Ss denote saddles, and Cs denote catastrophes. Horizontal lines indicate the connections or that paths connecting catastrophes and landmarks with minimal energies. The last row of the figure shows trees of extrema and saddles, presented here only for better interpretation of the MSSTs.

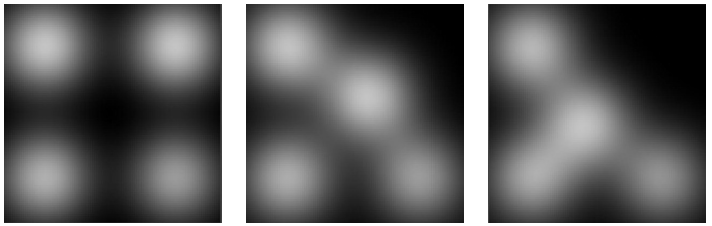

Figure 3.2: Images of three stationary and one moving Gaussian blobs. A few images taken from the series of generated images used in the experiment.

Smooth changes of the generated images are forced by smoothly changing the parameters used for generating those images. For each test image, a scale-space is computed, the critical paths and catastrophes are detected and, finally, the Extrema-Based and Saddle-Based MSST are constructed. Changes in the constructed MSSTs obtained from neighboring test images are observed and carefully classified into categories.

Four kinds of transitions can be observed, i.e. changes of catastrophe-extremum/saddle positions, changes of catastrophe-catastrophe relations, changes of catastrophe ordering, and changes of extremum-catastrophe connections. More complicated changes of MSSTs can be described as a combination of these basic transitions.

3.5.1 Changes of Catastrophe-Extremum/Saddle Positions

The changes of catastrophe-extremum/saddle positions are simple transitions and do not effect the topology of the MSSTs. They are caused by the movements of catastrophes in scale-space due to the displacements of extrema/saddles and/or changes in their intensity values in the original image. In this transition, catastrophes may change their positions as long as their ordering in scale is undisturbed. Since the catastrophe ordering is fixed, this transition produces no topological changes of the MSSTs but only effects the contents stored in the corresponding nodes. The changes of catastrophe-extremum/saddle positions transition is illustrated in Fig. 3.3.

3.5.2 Changes of Catastrophe-Catastrophe Relations

When the image is smoothly changed, some connections in the MSSTs are getting more stable, while others getting more unstable, and at some points the switching of connections will occur. When the nesting of image segments in scale changes, the parent-child relations in the Extrema-Based and Saddle-Based MSSTs will change accordingly. The transition results in the change of the leftport of the corresponding node and the movement of that node and its right-subtree to a

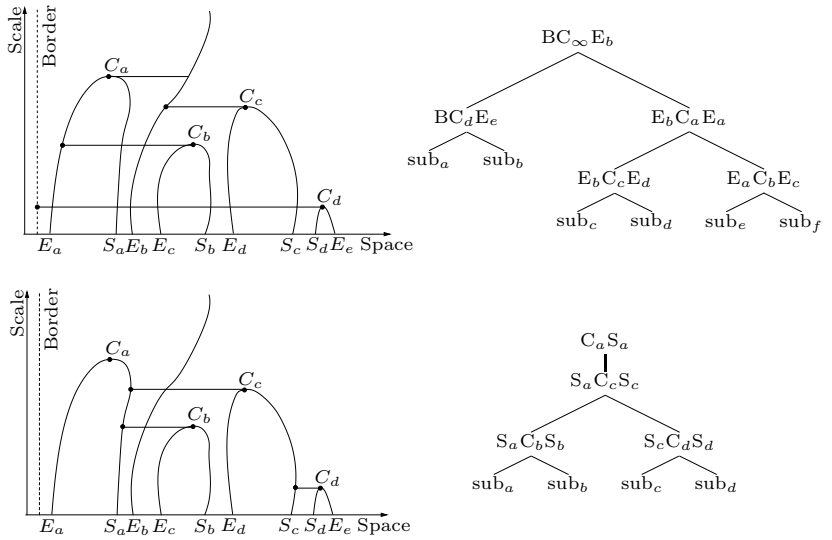

Figure 3.3: Changes of catastrophe-extremum/saddle positions. The original schematic drawings of the deep structures and their corresponding Extrema-Based and Saddle-Based MSSTs before and after the change of position transition are shown in the top and bottom row respectively. No topological change of the MSSTs.

new location in the MSST defined by its new leftport.

First, insert that node together with its right-subtree at the new location defined by the ordering of the catastrophes and its updated leftport, replacing the subtree that previously occupies the port. Secondly, The left-subtree of that node is then moved up and replaces the position of the moved away node. Finally, the replaced subtree is then reconnected as the left-subtree of the inserted node.

The situation is illustrated as the transition from the MSSTs in Fig. 3.3 to the MSSTs in Fig. 3.4, where the node of C_b is changed from being nested inside E_a to being nested inside E_b in the Extrema-Based MSST and the node of C_b is changed from being nested inside S_a to being nested inside S_c in the Saddle-Based MSST. Note that the changes of catastrophe-catastrophe relations transition usually occurs simultaneously with the changes of catastrophe-extremum/saddle positions transition.

3.5.3 Change of Catastrophe Ordering

The positions of catastrophes change smoothly both spatially and in scale as the image is smoothly changed and at some point a pair of catastrophes will switch their ordering in scale.

The topology of MSSTs depends strongly on the ordering of the catastrophes in scale. If the ordering changes the structure of Extrema-

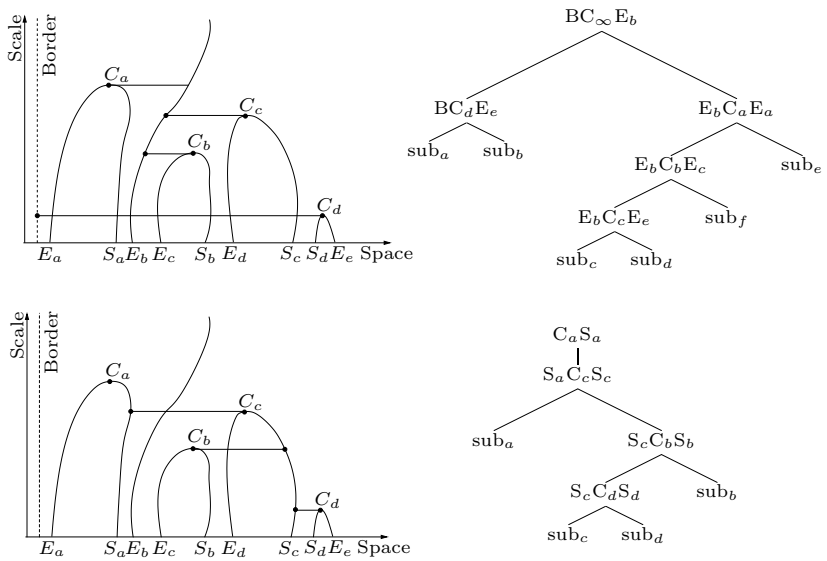

Figure 3.4: Changes of catastrophe-Catastrophe Relations. The schematic drawings of the deep structures and the corresponding Extrema-Based and Saddle-Based MSSTs after the changes of catastrophe-catastrophe relations transition are shown in the top and bottom row, respectively.

Based and Saddle-Based MSSTs will change. If we only consider local reordering between two neighboring catastrophes in scale or nodes in MSSTs–larger changes of the ordering can be arrived by multiple local reordering, the situation is quite simple. Either:

1. *The two nodes do not hold parent-child relations:*
   The structure of the MSST is intact.

2. *The two nodes hold parent-leftchild relation:*
   Swap the locations of the two nodes thereby swap their parent-leftchild relations. Exchange the content of their leftports and finally reconnect the left-subtree of the moved-up leftchild (new parent) as the left-subtree of the moved-down parent (new left-child). Their right-subtrees are left untouched.

3. *The two nodes hold parent-rightchild relation:*
   Swap the locations of the two nodes thereby swap their parent-rightchild relations. Set the leftport of the moved-down parent (new rightchild) to that of the rightport of the moved-up rightchild (new parent). Reconnect the right-subtree and the left-subtree of the new parent as the left-subtree and the right-subtree of the new rightchild, respectively. Finally, update the leftport of the new parent and reconnect it at the new location defined by its new leftport, following the procedure of the changes of catastrophe-catastrophe relations transition.

The first case of the changes of catastrophe ordering transition is simply the extension of the changes of catastrophe-extremum/saddle positions transition, where local reordering of catastrophes is allowed. The second case can be thought of as a combination of the changes of catastrophe-extremum/saddle positions and the changes of catastrophe ordering, while the third case is a combination of changes of catastrophe-extremum/saddle positions, changes of catastrophe-catastrophe relations, and change of catastrophe ordering that occurs simultaneously.

The change of ordering transition is illustrated as the transition from the MSSTs in Fig. 3.3 to the MSSTs in Fig. 3.5, where the neighboring nodes of C_a and C_c swap their ordering positions in scale.

3.5.4 Changes of Extremum-Catastrophe Connections

Sometimes, structural changes of critical paths in scale-space occur. The most familiar example of these changes is the transition through a non-generic catastrophe where two extrema and one saddle meet at a catastrophe and later one extremum survive. The catastrophe, before, is connected to one extremum and, later, is connected to the others, while its position is almost fixed.

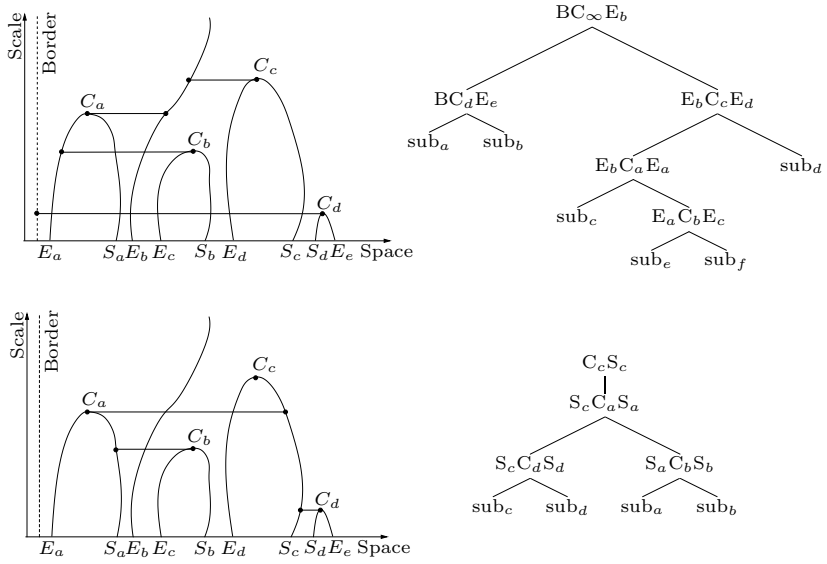

Figure 3.5: Changes of catastrophe ordering. The schematic drawings of the deep structures and their corresponding Extrema-Based and Saddle-Based MSSTs after the changes of catastrophe ordering transition are shown in the top and bottom row, respectively.

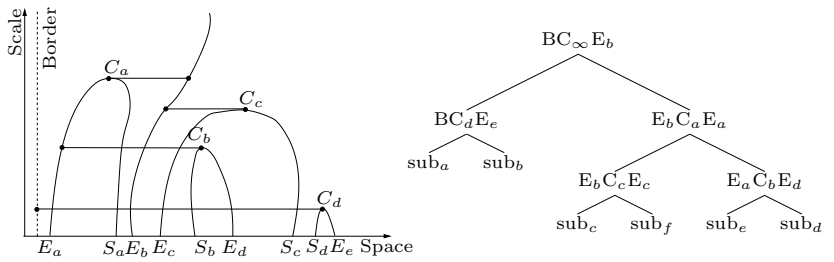

Figure 3.6: Changes of extremum-catastrophe connections. The schematic drawing of the deep structure and the Extrema-Based MSST after the changes of extremum-catastrophe connections transition.

The changes of extremum-catastrophe connections can be thought of as the swapping of extremal path connecting between extrema and their corresponding catastrophes. Since it does not effect the saddle paths, this transition is only relevant to Extrema-Based MSSTs. In MSSTs, the transition is translated to the swapping of the right-ports and the right-subtrees of the two nodes.

The changes of extremum-catastrophe connections transition is illustrated as the transition from the Extrema-Based MSST in the top row of Fig. 3.3 to the Extrema-Based MSST in Fig. 3.6, where the catastrophe-extremum connections between E_c and C_b and between E_d and C_c swap.

3.6 Extrema- VS. Saddle-Based MSSTs

Critical points in generic images can be categorized into maxima, minima, and saddles by the signs of the eigenvalues of the Hessian matrix.[2] With proper boundary conditions, e.g. zero-padding, the image in scale-space at scale infinity will hold only one extremum and no saddle [52].

Frequently, the positions of extrema can change significantly, when are traced back from the catastrophes in scale-space to the original image at scale zero, even if only slight perturbation is imposed to the image. These jumps of extrema positions are actually caused by the swapping of extremal paths connecting extrema and their corresponding catastrophes, and extremal paths that extends to higher scales. The situation is illustrated in Fig. 3.7, where the swapping of extremal paths occurs after slight perturbation is imposed.

The swapping of extremal paths result in complicated transitions of Extrema-Based MSSTs as described above. On the other hand, the swapping of extremal paths does not disturb the topology nor the contents stored in nodes of Saddle-Based MSSTs at all.

Nevertheless, Saddle-Based MSSTs essentially do not fail to capture the changes of the original image due to these slight perturbations. The swapping of extremal paths, that is translated to complicated

[2]The matrix of all second-ordered derivatives.

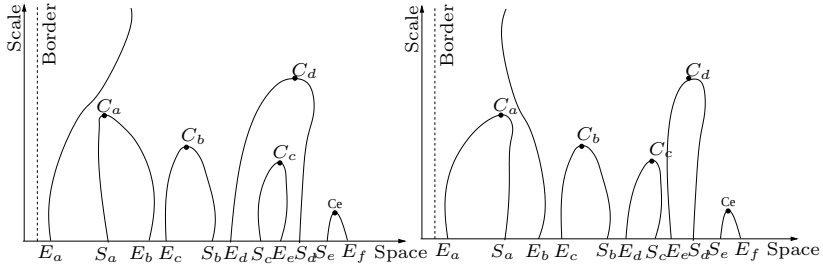

Figure 3.7: The schematic drawings of the deep structures of an image before and after imposing slight perturbation. Frequently, the extremal paths connecting extrema and catastrophes can be easily disturbed but the saddle paths can be considered stable.

transitions in Extrema-Based MSSTs, is simply transformed to simple movements of saddles and catastrophes in Saddle-Based MSSTs.

The selection of Extrema-Based MSSTs or Saddle-Based MSSTs should based on the aimed applications. However, the smaller number of possible transitions for Saddle-Based MSSTs and their simplicity are of interest. They may ease the development of algorithms based on MSSTs and lead to simpler, faster, and better implementations.

3.7 Summary and Conclusions

This Chapter presents the set of possible transitions of the Extrema-Based and Saddle-Based Multi-Scale Singularity Trees. The transitions presented in this Chapter is a minimal set of MSST transitions in a sense that all possible trees of the same number of nodes can be

related to each other through series of transitions found in this set. However, we have not proved conclusively that the set is complete, since there might be other complex combinations of those transitions in the set that, generically, occur simultaneously.

The simplicity of Saddle-Based MSST transitions, in comparison with those of Extrema-Based MSSTs, is favorable. Because they are simpler, they are easier to be understood. It will allow us to be able to derive good estimations of the cost for each transitions and will simplify further developments of the algorithms based on MSSTs.

The study of the transitions of Extrema-Based and Saddle-Based MSSTs presented in this Chapter is an important step toward the development of an image matching and indexing algorithm based on MSSTs and the Tree Edit Distance (TED) algorithms [5], where the distance or difference between two images is found as the minimum cost of a series of edit operations that transforms the MSST of one image into another.

Potential applications of MSSTs include image matching using MSSTs, multi-scale image pre-segmentation, sub-object extraction, hierarchical image retrieval in large image databases, etc. Recently, MSSTs also found its applications in computer graphics [76].

Part II

Applications

Chapter 4

Image Matching using MSSTs[1]

No great discovery was ever made

without a bold guess.

I. Newton

In scale-space theory, the relation of image features at different scales is referred to as the deep structure of the image. Multi-Scale Singularity Trees (MSSTs) represent the deep structures of images using binary

[1] An earlier version of this work has been published as a DSSCV project technical report titled "Performance Evaluation of Tree Object Matching" [81] and "Report on Matching 3D Image Structures by their MSSTs in a Given Application" [82].

trees. Unlike all other similar multi-scale image descriptors previously described in the literatures [46, 42, 64, 60], where only the connections between image features are suggested, MSSTs provide both the connections and their strengths. In this chapter we describe and later evaluate an image matching algorithm that exploits the properties of MSSTs. Two versions of the algorithm are presented: an exact and an approximation. Several experiments are conducted to empirically evaluate the performance of the matching algorithm under various kinds of image distortions and noise levels. Further, the performance of the matching algorithm is measured on three databases: the ORL face database [68], magazine covers, and the COIL database [58]. Finally the performance is compared with matching algorithms based on the Scale Invariant Feature Transform (SIFT) [54], and the Positions of Catastrophes (CAT) [35].

4.1 Introduction

The quantification of the differences or distances between images, and choosing the two closest images is called image matching. Image matching is a fundamental task in content-based image retrieval systems. The typical application of the system is that the user presents the system with an image, and the system returns a ranked list of some images obtained from a database that are similar. Such an application is becoming increasingly popular. Examples of its usages are: large paper and television news archives, security systems, and home/business image databases.

In this chapter, we only discuss Gaussian scale-space, which we just call scale-space. The history of Gaussian scale-space can be found in [95]. We have chosen to focus on two related works, matching algorithms based on the Position of Catastrophes (CAT) [35] and on the Scale Invariant Feature Transform (SIFT) [54], augmented with the Earth Mover Distance (EMD) [67]. These methods are all based on scale invariant features, where SIFT is the most developed and successful.

In [35], a set of catastrophes in scale-space and their reconstruction coefficients were used as an image representation for matching. The image matching problem was then translated into comparing sets of points in high dimensional space. The distances between point sets were calculated using the Earth Mover Distance (EMD) [67]. As the authors realized, some catastrophes are more stable than others, hence in [65] unstable catastrophes were discarded from the image representations. They argued that catastrophes in an area with a lot of structure are more stable. The amount of structure contained in a spatial area around catastrophes can be estimated by the total variation norm. More in depth discussion of the stability of catastrophes

based on perturbation theory and noise propagation was presented in [2]. Such stability measures are an integral part of the EMD algorithm. The subset of this method, where only the positions of the catastrophes are used, is referred to as CAT in this chapter.

In [53, 54] and accompanying articles, a set of Scale Invariant Feature Transform (SIFT) were presented for image matching. The SIFT features are calculated in scale-space as extremal points of differences of Gaussian blurs. These are approximations of the scale normalized (spatial) Laplacian, and their extremal points correspond to the points of blob-detection [48]. Out of all detected points, the SIFT algorithm selects robust points by eliminating low-contrast points and edge points. Then local histogram of the gradient vector is sampled non-linearly in a small number of orientation and magnitude bins. The SIFT is a set of features for which an accompanying matching algorithm has been proposed [3]: Best-Bin-First (BBF). In this chapter, we use the Earth Movers Distance (EMD) [67] instead since the algorithm is publicly available and it appears to have comparable performance on the SIFT features.

None of the above mentioned image matching algorithms includes the information on relations or linking between catastrophes or their features in their image representations as opposed to the Multi-Scale Singularity Trees (MSSTs) introduced in [80]. These novel trees and

powerful multi-scale image descriptors represent the deep structures of images and the relations of image features at different scales. Two kinds of MSSTs have been proposed: Extrema-Based MSSTs and Saddle-Based MSSTs. The difference between the two MSSTs is that in the Extrema-Based, the catastrophes are linked with extrema, while in the Saddle-Based, the catastrophes are connected to saddles. A study of the transitions of MSSTs under image perturbations [83] suggested that Saddle-Based MSSTs are potentially more stable and preferable as image descriptors for matching.

In this chapter we will experimentally evaluate the usefulness of Saddle-Based MSSTs as descriptors for image matching. The image matching algorithm, which will be described in detail, calculates the image distances using only the energy matrices of MSSTs. The topology of MSSTs are decided from the information stored in the energy matrices, and they are where the strengths of the connections can be derived. Therefore, the quality of matching results may indicate the amount of image information that is captured by the connections between catastrophes in MSSTs.

4.2 Multi-Scale Singularity Trees

A Saddle-Based MSST is fully described by the saddles at the first-scale image, the catastrophes in scale-space, and the invariant energy

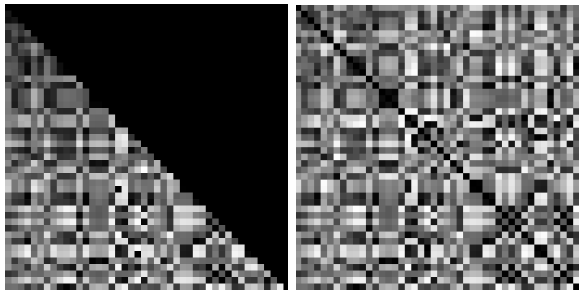

Figure 4.1: The original half energy matrix and the proposed full energy matrix of Saddle-Based MSSTs extracted from the same image.

matrix. In this chapter, each element in an energy matrix is the energy calculated between a pair of saddles at the first-scale image as opposed to the original Saddle-Based MSSTs described in [80] where the energies are calculated at the scales of catastrophes. Examples of the original half energy matrix and the proposed full energy matrix of Saddle-Based MSSTs extracted from the same image are shown in Fig. 4.1. Both versions of the energy matrices suggest approximately the same connections in MSSTs. The energy matrix of a Saddle-Based MSST then contains the energies between all pairs of saddles in the image.

In scale-space, increasing the scale parameter simplifies the original image. Saddles and extrema disappear or appear at annihilation and creation catastrophes, respectively, and the only generic catastrophe is pairwise annihilation or creation [11]. Since all saddles and extrema

except one extremum in the original image eventually disappear at annihilation catastrophes, saddles may all be uniquely associated with annihilation catastrophes in scale-space.

Generically, the probability of two annihilations occurring at the same scale is zero, and we may thus rank saddles according to the scales of their associated annihilation catastrophes. The building of MSSTs is done in a coarse to fine manner. The linking for each catastrophe c_i in an MSST can be decided by looking for the saddle s_j, which is present at the scale of the catastrophe c_i, and calculate an energy measure at the first scale image. The catastrophe c_i is then linked to the catastrophe c_j that is associated with the saddle s_j with the lowest energy.

The energy between a pair of saddle s_i and saddle s_j at the first scale image, ranked according to the scales of their associated catastrophes, is the element E_{ij} of the energy matrix.

$$E_{ij} = \inf_{\gamma \in \Gamma_{s_i s_j}} \int_0^P \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, dp \,, \qquad (4.1)$$

where $I : \Omega \rightarrow R^+$ is an intensity image, $\gamma : [0, P] \rightarrow \Omega$ is a path in the image parameterized by p, such that $\gamma(0) = s_i$ and $\gamma(P) = s_j$, $\Gamma_{s_i s_j}$ is the set of all possible paths the two saddles s_i and s_j, and α is a weighting factor between space and image intensity. Note that (4.1) is independent of the parameterization, e.g. integrating with respect

to $q : [0, P] \rightarrow [0, P]$ we find:

$$\begin{aligned} E_{ij} &= \inf_{\gamma \in \Gamma_{s_i s_j}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p(q))}{dp} \mid^2 + \alpha \mid \frac{\partial I(\gamma(p(q)))}{\partial p} \mid^2} \, dq \\ &= \inf_{\gamma \in \Gamma_{s_i s_j}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \cdot \frac{dp}{dq} \mid^2 + \alpha \mid \nabla I \cdot \frac{d\gamma(p)}{dp} \cdot \frac{dp}{dq} \mid^2} \, dq \\ &= \inf_{\gamma \in \Gamma_{s_i s_j}} \int_{p^{-1}(0)}^{p^{-1}(x)} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \frac{dp}{dq} \cdot dq \\ &= \inf_{\gamma \in \Gamma_{s_i s_j}} \int_{0}^{P} \sqrt{(1-\alpha) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, dp \, . \end{aligned} \tag{4.2}$$

In this report the $\alpha = 1$ is set so that the energy only depends on the intensity of the image and is theoretically invariant under all image transformations that act only on image space, but not on the intensity of I. Substituting $\alpha = 1$ in (4.1) one gets

$$E_{ij} = \inf_{\gamma \in \Gamma_{s_i s_j}} \int_0^P \left| \frac{dI(\gamma(p))}{dp} \right| dp \, . \tag{4.3}$$

In plain English, the energy E_{ij}, where $\alpha = 1$ is the minimum sum of image intensity differences along any possible paths from s_i to s_j. This is demonstrated in Fig. 4.2, which shows a magazine cover, the zero-crossings of the first derivatives overlaid on top of the first scale image, the deep structure, and the energy matrix. The first scale image appears flipped because it is viewed from behind so that the coordinate axises agree with the right hand rule. The energy matrix

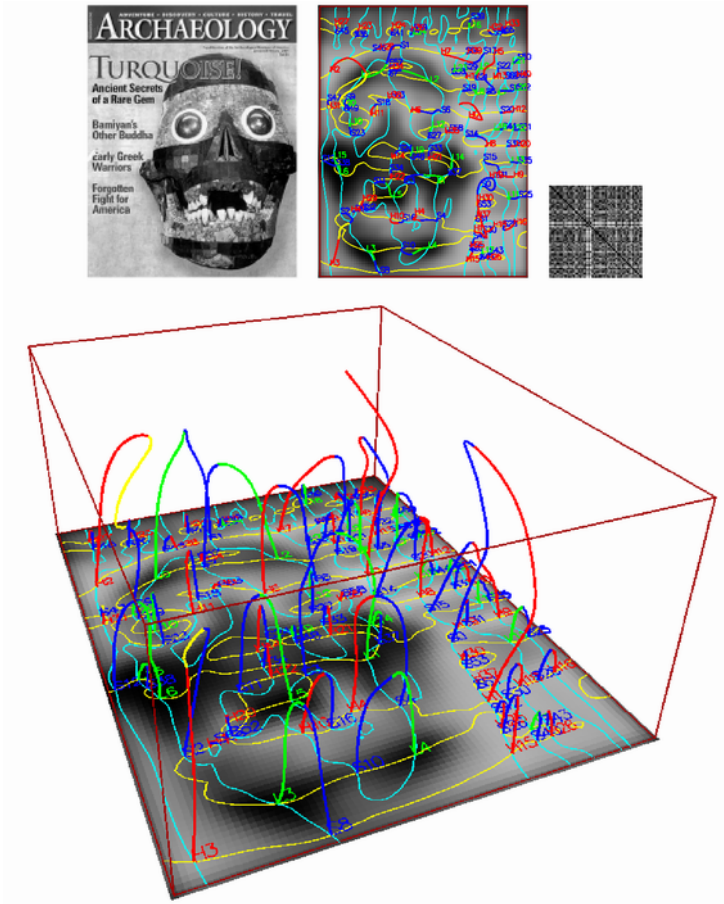

Figure 4.2: A magazine cover, its deep structure, and it's energy matrix. Maximal-paths, minimal-paths, and saddle-paths are displayed in red, green, and blue, respectively. The zero-crossings of the first derivatives I_x and I_y at the first-scale image are shown in light blue and yellow, respectively.

is symmetric with zeroes along its diagonal, and all its elements are non-negative.

4.3 Coping with Creations

In 2 dimensions and higher, creations in scale-space are generic but the tree building method as discussed above assumes that only annihilations occur. In order to preserve the tree structure of the MSSTs and to simplify the matching algorithm, creations and loops in scale-space images are systematically removed.

It has been shown in [42] that although creations are generic, they might not be easily detected. Creations that form loops are short-lived, hence if we do not sample scale-space fine enough along scale, then they may pass undetected. On the other hand, most creations that do not form loops are followed closely by annihilations and slight perturbations will merge them. Therefore we propose to handle creations as follows:

1. Creations that occurs in critical-paths, where those paths can be traced down to the first-scale image, are pairwisely removed with the next annihilations on that path in the direction that moves to the first-scale image. The top-most annihilations catastrophes on those paths are the catastrophes associated with the saddles

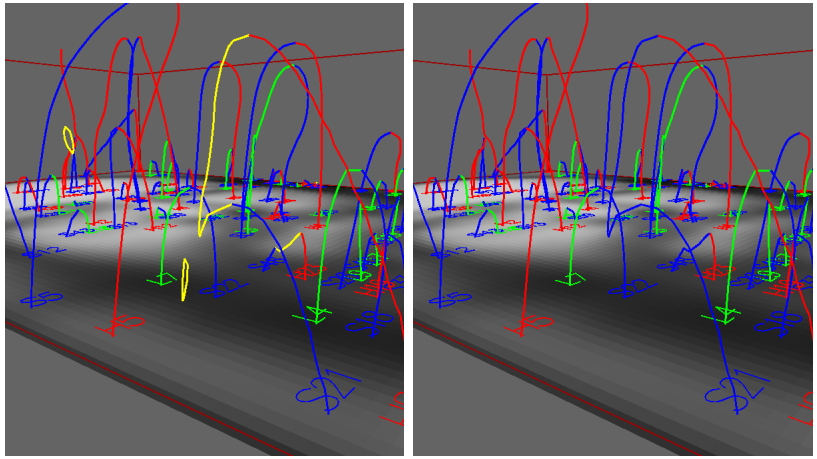

Figure 4.3: Creations are systematically ignored. Critical-paths that originate from creations are indicated by yellow. Left and right panels show part of a deep structure before and after the removal of creations. Pairwise removals of creation-annihilation pairs are indicated by color changes of the paths from yellow to blue.

at the first-scale image.

2. Creations that eventually end at annihilations and form loops in scale-space with no connecting path to saddles at the first-scale image are ignored.

In Fig. 4.3 is shown an example of loop removals and pairwise removals of creations and annihilations on critical paths that can tracked back to the first-scale image. The figure shows examples of pairwise removals of multiple creations and annihilations in critical-paths such that the highest annihilation in each saddle-path is kept as the catastrophe

associated with the saddle at the first-scale image. Likewise the figure shows that creations resulting in loops are ignored.

4.4 Matching Algorithm

The distance between two images is calculated as a sum of the squared differences of the corresponding elements in the normalized MSSTs' energy matrices,

$$D(I,J) = \sum_{i,j} \left[\frac{E_I(i,j)}{\sqrt{\sum_{i,j} E_I(i,j)^2}} - \frac{E_J(i,j)}{\sqrt{\sum_{i,j} E_J(i,j)^2}} \right]^2, \qquad (4.4)$$

where I and J are two images to be matched, and E_I and E_J are the corresponding energy matrices. In order to remove the distortion of the energy caused by possible different dynamic range of images, the energy matrices are normalized by dividing each element with the squared root of the sum of the element squared. Since D is a quadratic combination of E, then D observes the same invariance as E, i.e. D is invariant under all transformation of image space, such as scaling, translation, and rotation.

The ordering of catastrophes in scale effects the location of rows and columns of energy matrices: a swap of ordering between catastrophe c_i and catastrophe c_j in scale-space, corresponds to a swap of row

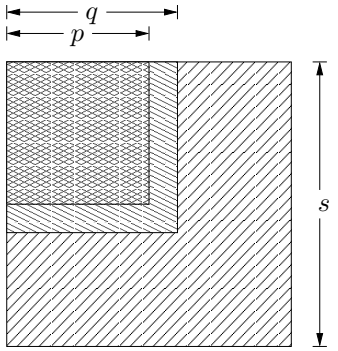

Figure 4.4: The schematic diagram of the exact matching algorithm. s denotes the total number of catastrophes, q denotes the number of catastrophes used in the permutation, and p denotes the number of catastrophes used in the calculation of the distance, where $p \leq q \leq s$.

i and row j, and column i and column j in the energy matrix. In order to compute the minimal differences of the energy matrices taking into account also the possible catastrophe reordering, the matching algorithm is based on the minimum under possible swapping. In the following we will present two versions of the algorithm: the exact and the approximate matching.

4.4.1 Exact Matching

The *exact matching algorithm*, shown schematically in Fig. 4.4, searches, among all permutations of the catastrophe ordering, for the ordering that minimizes the distance in (4.4). Catastrophes corresponding to

large scale structures in images are usually located high in scale. Because the saddles are ranked according to the scales of their associated catastrophes, part of the energy matrix that corresponds to large scale image structures are located on the top-left part of the energy matrix.

We cannot assume that all catastrophes in one image will exist in another image to be matched. Hence, it is important that the matching algorithm is able to discard or delete catastrophes that do not fit well during the matching process. This is achieved in the exact matching algorithm by permuting a slightly larger sub-matrix than that is used in the distance calculation. For an energy matrix with s catastrophes, the exact matching algorithm permutes the top-left part of the energy matrix with q catastrophes, $q \leq s$. Then, the computation of the distance is performed only on the sub-matrix with p catastrophes, $p \leq q$, allowing catastrophes that are not fit very well to be discarded.

Because the searching space, sequentially processed by the algorithm, grows factorically in q, only a small numbers of top catastrophes can be used. The experiment in the next section shows that using 6 catastrophes is the most appropriate trade-off between speed and accuracy and using 8 catastrophes sets a practical limit for the 10×10 database.

Catastrophes, which are far apart in scale are less likely to have come form compatible image structure. However, the scale-space of

a flat image with i.i.d. noise will show a rich catastrophe structure, which is completely unstable, in the sense that drawing another noise example from the i.i.d. noise will result in a completely different set of catastrophes. Luckily, given a noise-level of the image, the statistical variance of the catastrophe localization may be estimated [65, 2] and this may be used as a weighted penalty in the catastrophe swapping.

We thus propose a matching cost based on Bayes' Maximum A Posteriori,

$$P(I|J) = \frac{P(J|I)P(I)}{P(J)} \tag{4.5}$$

where P denotes the a posteriori, the error, the prior, and the evidence probability distributions. When only interested in the maximum of (4.5), then the evidence may be ignored to give,

$$I^* = \arg\max_J P(J|I)P(I). \tag{4.6}$$

Further, since the logarithm is a strictly monotonic function, the maximum of (4.6) is equivalent to minimum of

$$I^* = \arg\min_J -\log P(J|I) - \log P(I) \tag{4.7}$$

Since the distance measure between energy matrices is a squared mea-

sure, we may rewrite it as a logarithm of a Gibbs distribution,

$$-\log P(J|I) = \frac{D(I,J)}{2\tau} + \log k, \tag{4.8}$$

where $k = \sum_K \exp \frac{D(I,K)}{2\tau}$ is the normalization factor calculated from all possible images K, and τ is some temperature variable. The variable k is constant for constant τ. In a similar manner we will design a prior probability distribution. One possibility is to use

$$-\log P(I) = \sum_{i,j \in S} \frac{w_i |\log \sigma_i - \log \sigma_j|}{\mu} + \log c \tag{4.9}$$

where we constantly keep track of the original scale σ_i of every catastrophe, and relate this to the scale of the catastrophe, it has been swapped with, σ_j. The weighting factor w_i may be used to control the movement of catastrophes, such that setting w_i high will imply that a catastrophe is unlikely to move. Finally the constant c is a normalizing constant. Writing the cost of swapping catastrophe i and j as,

$$F(i \leftrightarrow j) = 2|\log \sigma_i - \log \sigma_j|, \tag{4.10}$$

we find that F has two nice properties. Firstly it is scale invariant, i.e. scaling σ with some constant η implies that

$$F(i \leftrightarrow j) = 2|\log \eta\sigma_i - \log \eta\sigma_j| = 2|\log \sigma_i - \log \sigma_j|. \tag{4.11}$$

Secondly, we need not use the sequence of swapping performed for catastrophe i to have reached place of catastrophe j, we need only the absolute logarithmic difference to its starting point. This is an advantage algorithmically.

4.4.2 Approximate Matching

To increase the number of catastrophes used in the distance calculations and improve its accuracy while keeping computational time practical, we have devised the *approximate matching algorithm*, which is shown schematically in Fig. 4.5.

The approximate matching algorithm uses the *moving window* strategy: starting from the top-left part of the energy matrix, the approximate algorithm places the moving window on top of q catastrophes. The approximate matching algorithm then searches for the best catastrophe ordering among all possible permutations within the window using the exact matching algorithm locally. Once the best local ordering has been found, the moving window is moved one step to the right and one step down to the next position. The top most catastrophe in the previous window is assumed to be located at the correct position globally. The approximate matching algorithm continues in the same manner until all or a given number of catastrophes are processed.

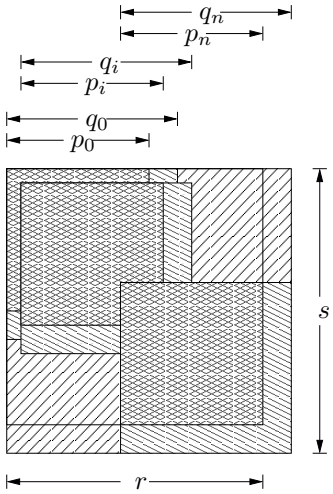

Figure 4.5: The schematic diagram of the approximate matching algorithm. s denotes the total number of catastrophes, q_i denotes the number and position of catastrophes used in the permutation window at step i, p_i denotes the number and position of catastrophes used in the calculation of the distance within the window at step i, and r denotes the number of catastrophes used in the final calculation of the distance, where $p_i \leq q_i \leq r \leq s$ and $0 \leq i \leq n$.

By keeping the window size small, more catastrophes can be used in the distance calculation than that used in the exact matching algorithm using the same computational time.

4.5 Experiments

Several experiments are performed on three image databases in order to investigate various aspects of the proposed matching algorithm and to compare its performance with selected state-of-the-art matching algorithms from the literature. The image databases used in the experiments are the ORL face database [68], the magazine cover database, and the Columbia Object Image Library (COIL) database [58]. The experiments may be classified into groups based on the databases used as follows:

1. The matching of face images in the ORL face database. We investigate the effect of each tunable algorithm parameters namely the number of catastrophes and the number of deletion ($q - p$). We present the performance comparison between the exact matching algorithm and the approximate algorithm using different number of catastrophes. The matching results using randoms points instead of the saddles are also presented to demonstrate the richness of saddles. Finally, the performance comparison of

the matching method and the methods based on CAT and SIFT on the ORL face database is presented.

2. The magazine cover database is used in the experiments to investigate the effects of various image transformations and different noise levels on the matching distance. A magazine cover is increasingly scaled, rotated, and added with random noise then the distances between the distorted images and the original image are computed. Finally, the performance comparison of the matching method and the methods based on CAT and SIFT on the magazine database is presented.

3. The result of the matching algorithm on the images selected from the COIL database is presented. Images of real-life objects viewed at different angles are used in the experiment. The experiment demonstrates the robustness of the MSSTs and the matching method against mixed scaling and 3D view point changes. Similarly, the performance comparison of the matching method and the methods based on CAT and SIFT on the COIL database is presented.

4.5.1 ORL Face Database

ORL face database is publicly available. It consists of 400 face images of 40 individuals with 10 images each. A subset of 10×10 images is

randomly chosen from the database and used in the experiments. In Fig. 4.6 is shown images of 10 individuals selected from the ORL face database. For each person, 10 face images are provided as shown in Fig. 4.7 for the person number four in the selected group.

The matching results of the exact matching algorithm using different number of catastrophe are shown in Fig. 4.8. The performance of the matching method increases as the number of catastrophes used in the calculation increases. For this particular image database the matching performance starts to level out when 8 catastrophes are used. The matching results using random points instead of saddles is also presented here for comparison. The deletion number of 1 was used in the experiment.

In Fig. 4.9 is shown the impacts of the number of deletions on the matching results. The positive effects of deletions only start to be visible when at least 6 catastrophes are used in the distance calculation. With lower number of catastrophes, the deletions might even worsen the matching result. This is due to the fact that for a very small number of catastrophes, the amount of image information contained in each catastrophe is proportionally large and it offsets the positive effects of letting go unfitted catastrophes.

When more than 6 catastrophes are used in the distance calculation, the exact matching algorithm becomes very slow. Using the

Figure 4.7: The 10 face images of an individual from the selected group. For each person, images are captured with various facial expressions, different hair styles, with/with out glasses, etc. The red box indicates the key image for the set, which is the first sample.

Figure 4.6: The selected 10 individuals from the ORL face database.

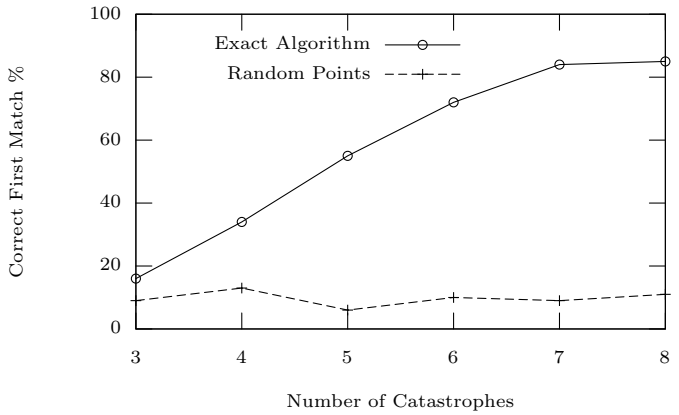

Figure 4.8: Matching results of the exact matching algorithm on the ORL face database using different number of catastrophes and random points.

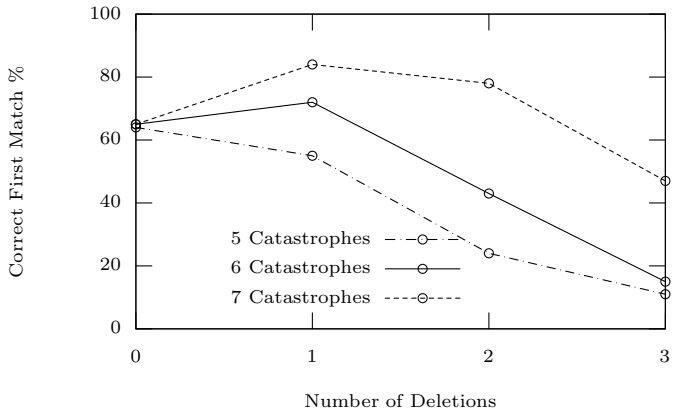

Figure 4.9: The impacts of number of deletions on the matching results. The results are produced using the exact matching algorithm.

approximate matching algorithm, more catastrophes can be included in the distance calculation while spending less computational time. In Fig. 4.10 is shown the matching performance of the exact matching algorithm compared with the approximate matching algorithm using different number of catastrophes. The approximate matching algorithm uses the window size of 6 and 1 deletion.

In Fig. 4.11 is shown the computational time used in second, plotted in logarithmic scale, for the exact and the approximate matching algorithms in order to complete the distance calculations on the selected 10×10 ORL face database and produce the results shown in Fig. 4.10. The computational time used for exact matching algorithm increases significantly when more than 6 catastrophes are used in the calculation. The computational time used for the approximate algorithm, on the other hands, grows almost linearly with the number of catastrophes. The approximate matching algorithm used the window size of 6 and 1 deletion. Interestingly, using in total 10 catastrophes, the performance of the approximate matching algorithm already beat that of the exact matching algorithm using 8 catastrophes, and spent less than 1/20 of the computational time used by the exact matching algorithm. Finally, The matching results of our matching algorithm (MSST) compared with those of the methods based on SIFT keypoints (SIFT) and the positions of catastrophes (CAT) are shown in

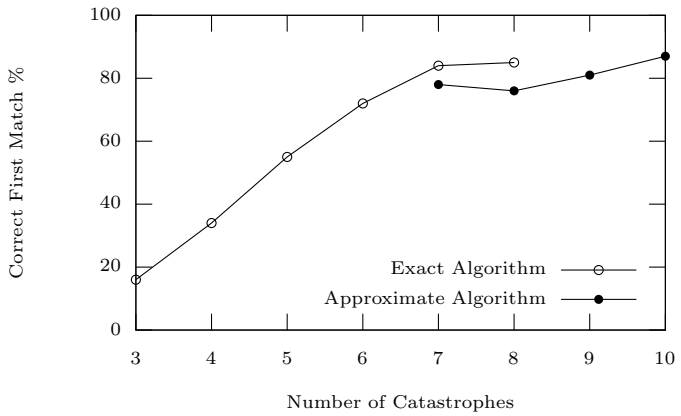

Figure 4.10: Matching results of the approximate matching algorithm compared with that of the exact matching algorithm using different number of catastrophes. For the approximate matching algorithm, the window size of 6 catastrophes is used and both exact and approximate algorithm use 1 deletion.

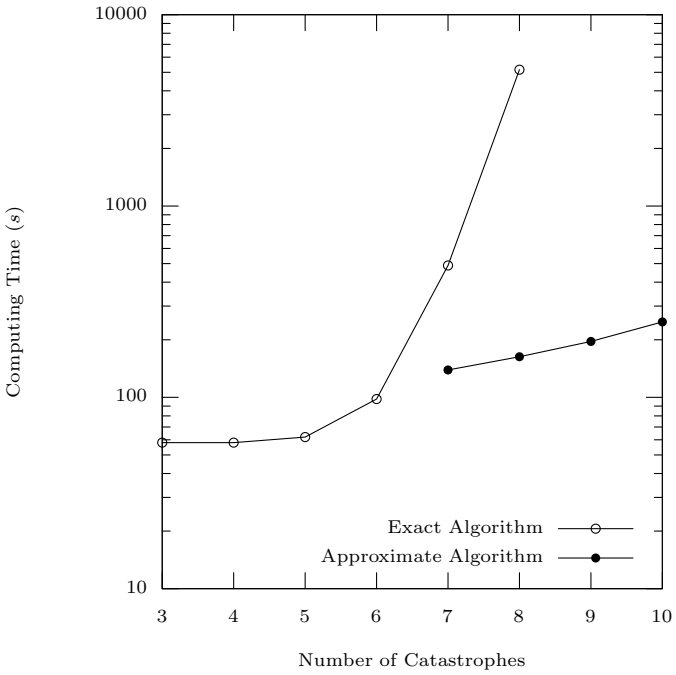

Figure 4.11: Computational time used by the exact matching algorithm compared with that used by the approximate matching algorithm in order to produce the results shown in Fig. 4.10.

Fig. 4.12. The distance between sets of SIFT key-points and catastrophe positions are calculated using the Earth Mover Distance (EMD). For each image, the mass of each feature point is set equally to $1/n$, where n is the number of feature points in the scale-space. In this particular experiment, our algorithm performs relatively poor compared with the other two methods.

The matching results plotted against the number of raw data used for our matching method (MSST), the method based on SIFT key-points (SIFT), and the method based on positions of catastrophes (CAT), are shown in Fig. 4.13, Fig. 4.14, and Fig. 4.15, respectively. The results of the three methods are shown together for comparison in Fig. 4.16.

Note that because the energy matrices of MSSTs are symmetric with zeroes along their diagonals, the numbers of raw data contained in energy matrices with e.g. 2, 3, and 4 catastrophes are 1, 3, and 6 numbers, respectively. The position of each catastrophe can be specified using 3 numbers and each SIFT key-point is represented using 128 numbers.

The CAT method performs relatively well using less numbers of raw data while our method (MSST) needs a little more numbers of raw data to start performing well. The SIFT method needs much more raw data as it must use between 256 and 384 numbers in order

to produce matching results comparable to CAT and MSST using only about 30 data.

4.5.2 Magazine Cover Database

In Fig. 4.17 is shown the 11 magazine covers in the Magazine cover database. Various kinds of objects can be found including computer typed letters, man-made and natural objects, natural scenes, and artificial graphics, etc. In this section, we will use these images to evaluate the behaviors of our algorithm under various image distortions and different levels of noise. Three important kinds of image distortions will be included in the experiments: planar rotation, uniform scaling, and random noise.

In Fig. 4.18 is shown the Piggy magazine cover scaled at different scaling factor starting from 0.5 to 1.0. In Fig. 4.19 is shown the Piggy magazine cover rotated at different in plane rotational angles starting from 0° to 90° counterclockwise. Fig. 4.20 shows the Piggy magazine cover with random noise added at different levels starting from 1% to 10%. Note that 1% noise means that each pixel in the image is added with random number drawn from $[-0.01, 0.01]$, when the pixels in the image are valued between $[0, 1]$

The results of matching distorted images by uniform scaling, planar rotation, and levels of random noise to their original images are shown

	2	3	4	5	6	7	8	9	10	Key
10-6-1	87%	76.5%	66.67%	60%	56.8%	52.67%	47%	42.5%	39.22%	54%
10-7-1	89%	80%	69.67%	63.5%	58.4%	53%	47.43%	43.63%	39.89%	54%
CAT	97%	96%	92.33%	90.5%	85.4%	81.83%	77.57%	73.63%	69.22%	82%
SIFT	100%	100%	100%	99.75%	97%	94.5%	92.86%	91.38%	89.33%	91%

Figure 4.12: Matching results of three different methods on the ORL face database. Our method (MSST) uses 10 catastrophes, 1 deletion, and with the window size of 6 and 7 respectively. The first column are the percentages of the second image being matched correctly, the second column are the percentages of the second and the third images being matched correctly, etc. The first image is the inquiry image and is always matched correctly. The last column gives the results of matching all faces only to the key image of each image set (the first sample). The matching results of SIFT and CAT using comparable computation time are also provided for comparison.

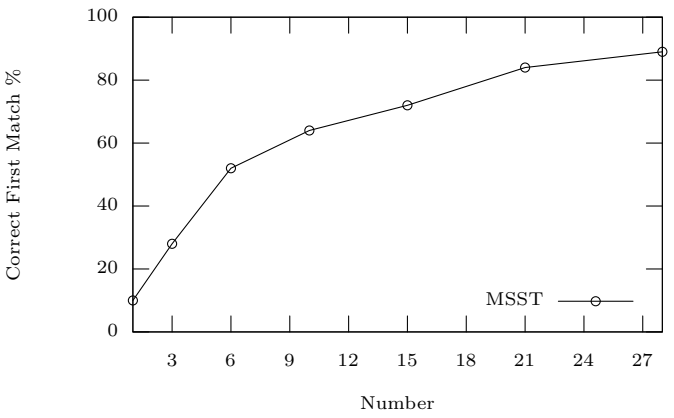

Figure 4.13: Matching results plotted against the number of raw data used for our matching algorithm (MSST) on the ORL face database.

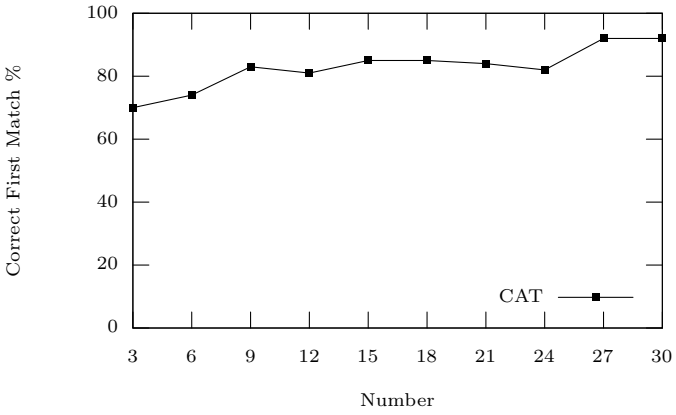

Figure 4.14: Matching results plotted against the number of raw data used for the CAT method on the ORL face database.

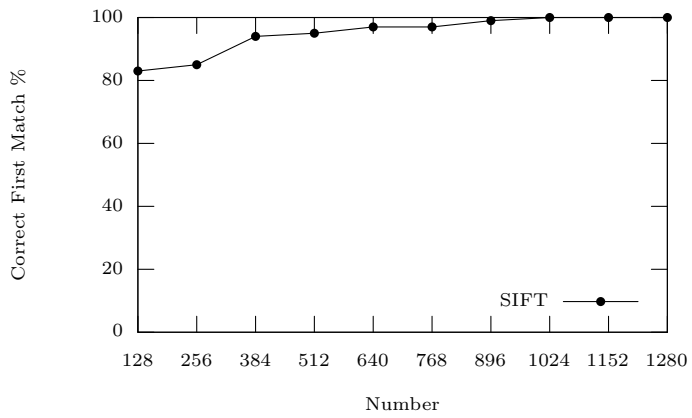

Figure 4.15: Matching results plotted against the number of raw data used for the SIFT method on the ORL face database.

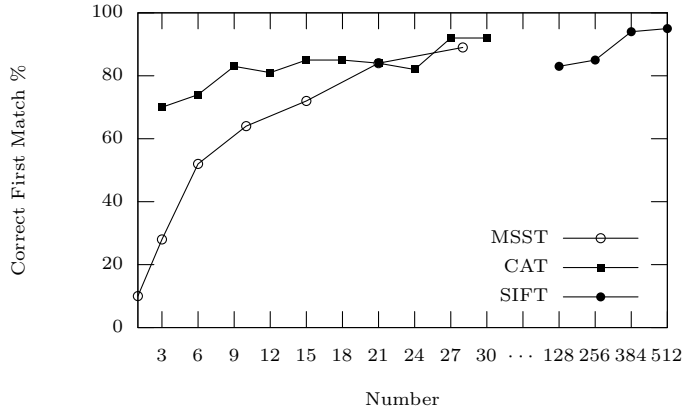

Figure 4.16: Matching results against the number of raw data used for MSST, CAT, and SIFT on the ORL face database.

Figure 4.17: The original 11 magazine covers in the magazine cover database. Various kinds of objects can be found including typed letters, man-made objects, natural scenes, and artificial graphics.

Figure 4.18: The Piggy magazine cover at various levels of scaling from 0.5 to 1.0. The original image is indicated by its red border.

Figure 4.19: The Piggy magazine cover at different rotation angles from 0° to 90°. The original image is indicated by its red border.

Figure 4.20: The Piggy magazine cover at various levels of noise added. The original image is indicated by its red border.

Figure 4.21: For each and every 11 original magazine covers, 10 image are generated by performing mixed rotation and scaling at various levels. The original magazine cover is first rotated then it is scaled so that the width of the image remains constant.

Figure 4.22: The effect of uniform scaling. The distances between scaled images at different scaling factors and the original image.

in Fig. 4.22, Fig. 4.23, and Fig. 4.24, respectively. It is noteworthy to mention that the impact of image rotation is periodic. The error is at the highest point, when the image is rotated at approximately 45°. In contrast, the error is very small, when the image is rotated at multiples of right angles. This is due to the fact that digital images are sampled on a rectangular grid, and because the energy map calculation using the Fast Marching Method [87, 70, 71, 72, 73] is implemented on rectangular grids. The resulting energy maps inevitably will be most inaccurately at rotational angles near 45°.

The total of 10 images are produced from each of every magazine cover in the Magazine cover database by performing the mixed uni-

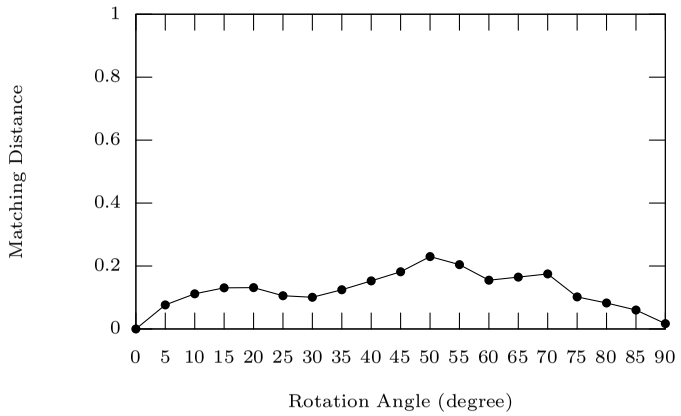

Figure 4.23: The effect of rotation. The distances between rotated images at different rotational angles and the original image.

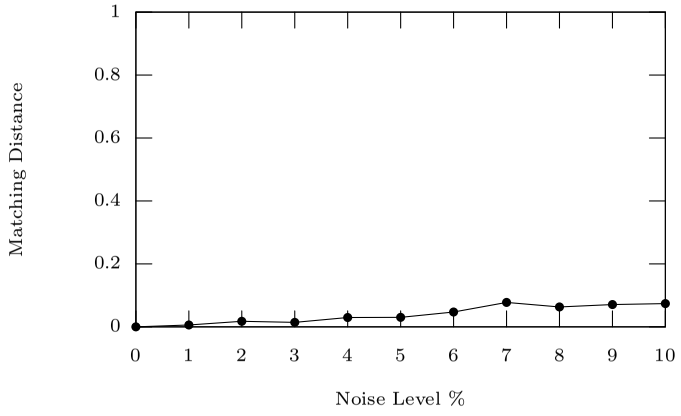

Figure 4.24: The effect of random noise. The distance between the images with random noise added at different levels and the original image.

	2	3	4	5	6	7	8	9	10	Key
8-6-1	98.18%	97.73%	96.36%	92.5%	90%	85.91%	81.56%	77.05%	72.42%	76.36%
10-6-1	100%	99.55%	98.18%	96.82%	95.09%	93.18%	90.13%	85.9%	80.51%	81.82%
CAT	57.27%	49.09%	39.09%	31.82%	28.73%	27.12%	24.81%	22.61%	20.71%	24.55%
SIFT	100%	100%	99.70%	99.77%	99.45%	99.24%	99.22%	97.05%	95.15%	100%

Figure 4.25: Matching results of three different methods on the transformed magazine cover database. Our method (MSST) uses 8, and 10 top catastrophes, 1 deletion, with the window size of 6 catastrophes. The first column are the percentages of the second image being matched correctly, the second column are the percentages of the second and the third images being matched correctly, etc. The first image is the inquiry image and is always matched correctly. The last column gives the results of matching all distorted magazine covers only to the key image of each set (the original image). The matching results of SIFT and CAT using comparable computation time are also provided for comparison.

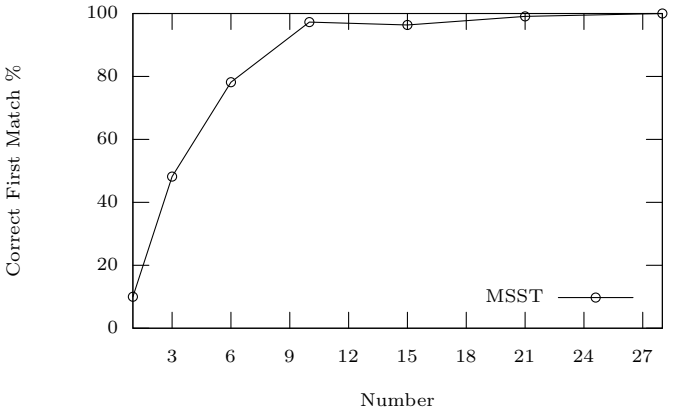

Figure 4.26: Matching results plotted against the number of raw data used for our matching algorithm (MSST) on the magazine cover database.

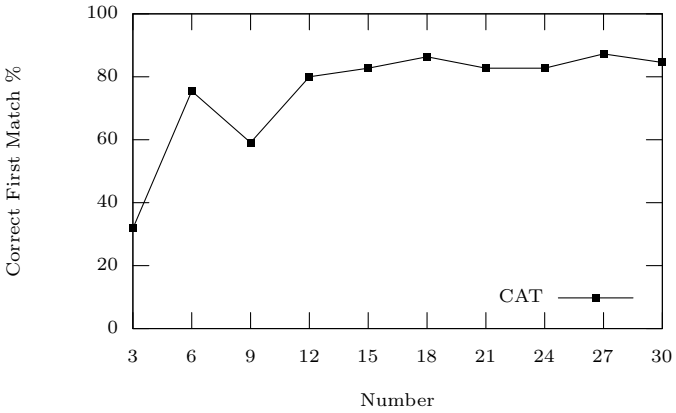

Figure 4.27: Matching results plotted against the number of raw data used for the CAT method on the magazine cover database.

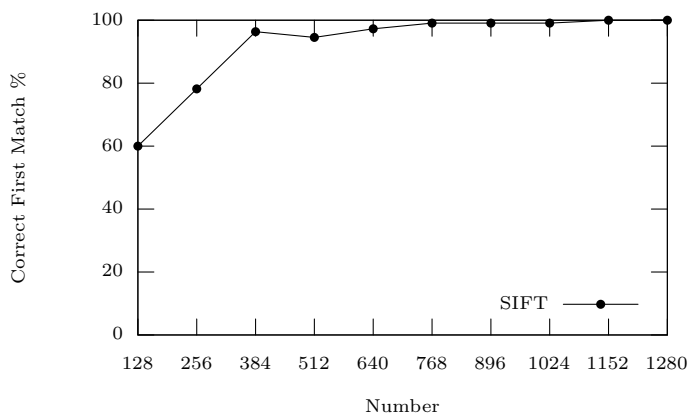

Figure 4.28: Matching results plotted against the number of raw data used for the SIFT method on the magazine cover database.

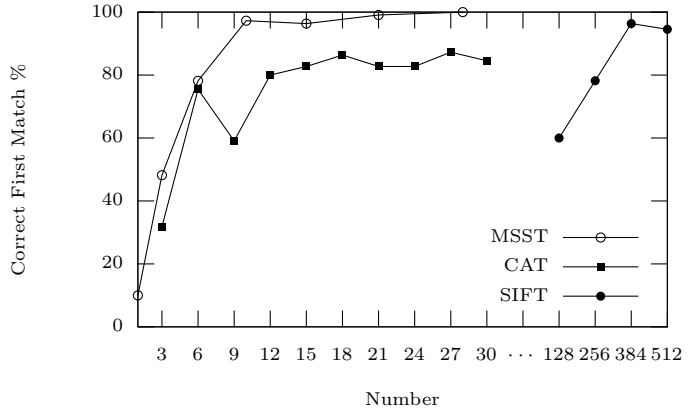

Figure 4.29: Matching results against the number of raw data used for MSST, CAT, and SIFT on the magazine cover database

form scaling and planar rotation. The original image is rotated at various rotational angles then the rotated images are scaled so that the widths of the images remain constant. In Fig. 4.21 is shown an example of images produced using the procedure from the Archeology magazine cover. The set of 11×10 transformed magazine covers are then used in the experiments. The Matching results of our method using different number of top catastrophes compared with the other two methods (CAT and SIFT) are shown in Fig. 4.25. SIFT and our method perform very well with SIFT performing slightly better. CAT performs poorly because positions of catastrophes are not invariant to severe translation, rotation, and scaling.

The matching results plotted against the number of raw data used for our matching method (MSST), the method based on SIFT key-points (SIFT), and the method based on positions of catastrophes (CAT), are shown in Fig. 4.26, Fig. 4.27, and Fig. 4.28, respectively. The results of the three methods are shown together for comparison in Fig. 4.29.

Our method performs the best in this experiment. Using only 9 data, our method produced almost perfect result while SIFT must use up to 384 data in order to produce a comparable result. It is interesting to mention that the CAT method performs better using low numbers of catastrophes, compared to its matching results shown

in Fig. 4.25 using all catastrophes in the images, because the positions of catastrophes located high in scale are more invariant to rotation and scaling than those located low in scale. Please note that no attempt has been made on making CAT rotational invariant.

4.5.3 COIL database

The Columbia Object Image Library (COIL) database [58] consists of images of various real-life objects captured at different 3D view points. The objects are rotated at intervals of 5°. The images are also subjected to different degrees of scaling.

Fig. 4.30 shows the 10 selected objects obtained from the COIL database. For each object, 10 images of different 3D view points between 0° and 45° are drawn from the database. In Fig. 4.31 is shown the 10 selected views of an object in the selected group from the COIL database. The key image, which is the middle sample, is indicated by its red border.

In total, 10×10 images selected from the COIL database are used in the experiment. The matching results of our algorithm (MSST), the method based on SIFT key-points (SIFT) and the method based on positions of catastrophes (CAT) are shown together for comparison in Fig. 4.32. All three methods performs very well especially SIFT, with our method slightly lagging behind.

Figure 4.30: The selected 10 objects from the COIL database.

Figure 4.31: The 10 selected views of an object from the COIL database. The key view is the view of 20° and is indicated by its red border.

	2	3	4	5	6	7	8	9	10	Key
8-6-1	96%	92.5%	90%	84%	78.8%	72.67%	68.14%	62.38%	58.56%	70%
10-6-1	96%	92%	88.67%	83.25%	77.2%	70.83%	64.28%	59.38%	55.11%	72%
10-8-1	97%	93%	87%	83.25%	78%	73.17%	67.71%	63.25%	58.33%	76%
CAT	100%	99%	97%	95%	90.6%	85.67%	81.29%	76.13%	71.78%	89%
SIFT	100%	100%	100%	100%	100%	99.83%	99%	98.13%	96.33%	100%

Figure 4.32: Matching results of different methods on the COIL database. Our method (MSST) uses 8, 10, and 10 top catastrophes with window size of 6, 6 and 8 catastrophes, respectively. All of which use 1 deletion. The first column are the percentages of the second image being matched correctly, the second column are the percentages of the second and the third images being matched correctly, etc. The first image is the inquiry image and is always matched correctly. The last column gives the results of matching all views of all objects only to the key view of each object. The matching results of SIFT and CAT using comparable computation time are also provided for comparison.

The matching results plotted against the number of raw data used for our matching method (MSST), the method based on SIFT keypoints (SIFT), and the method based on positions of catastrophes (CAT), are shown in Fig. 4.33, Fig. 4.34, and Fig. 4.35, respectively. The results of the three methods are shown together for comparison in Fig. 4.36.

The CAT method perform very well using less number of data. The performances of our method (MSST) and CAT method are comparable when 15 or more data are used while SIFT must used up to 256 data to produce a similar performance.

4.6 Discussions and Conclusions

For our method, the matching of ORL face database is slightly harder than the matching of the Magazine cover database and the COIL database. This is due to the high similarity between sets of objects. Faces all look quite similar, and they become even more similar, when they are blurred.

Important features in the images are usually associated with catastrophes located relatively high in scale. Low catastrophes are frequently associated with noise and insignificant features. The possibility of trading accuracy for performance is naturally provided for our method. Catastrophe located at lower scales can be simply discarded, if response time is the most critical factor with the price of lower recognition rates.

For small database, using only a few top catastrophes are enough for our matching method to produce good matching results. As the number of catastrophes used in the calculation of the distance increases, the method will have a better chance of discriminate in a large database, but it will also be more sensitive to image distortions and occlusions.

Our matching algorithm is considered currently in its very first steps. There are many possibilities for improvements. One possibility is the *adaptive approximate algorithm*, where the window size can grow or shrink adaptively, while keeping the scale difference between the highest and the lowest catastrophes in the window under a tunable level. The adaptive approximate algorithm will not only reduce the computational time but also likely to improve the matching results, since it will give also options to the algorithm to discarded catastrophes located high in scale, when appropriate.

The accuracy of the energy map generation is a very crucial part for good matching results. As Fig. 4.23 and Fig. 4.22 show, the accuracy of the current implementation, can still be largely improved, and we expect this to substantially improve the matching results.

It is interesting to note that the performance comparisons of the

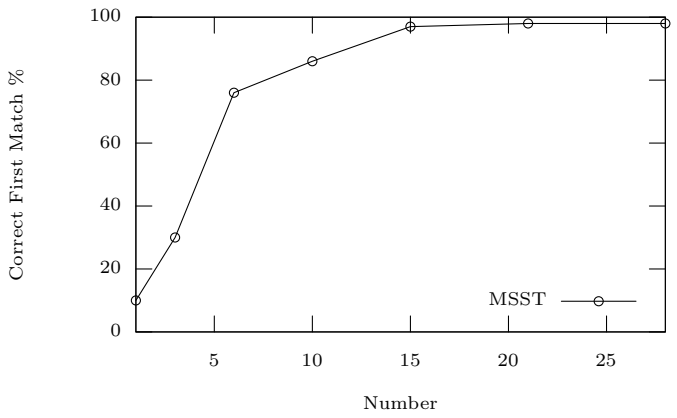

Figure 4.33: Matching results plotted against the number of raw data used for our matching algorithm (MSST) on COIL database.

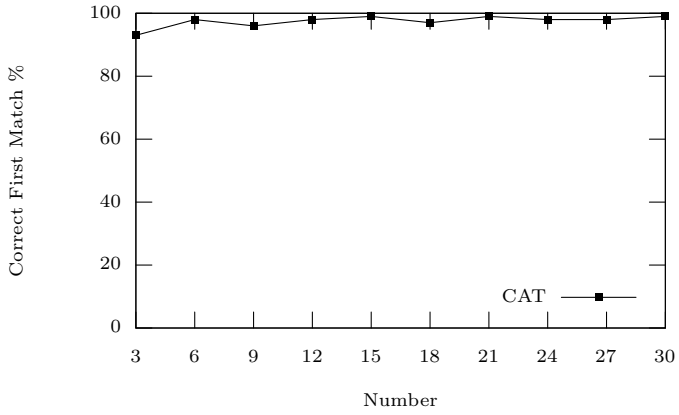

Figure 4.34: Matching results plotted against the number of raw data used for the CAT method on the COIL database.

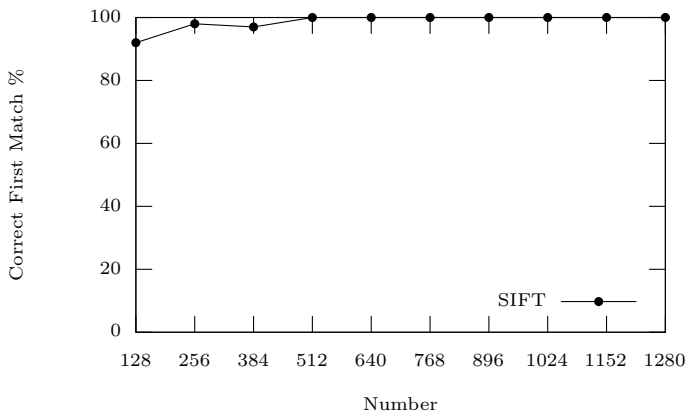

Figure 4.35: Matching results plotted against the number of raw data used for the SIFT method on the COIL database.

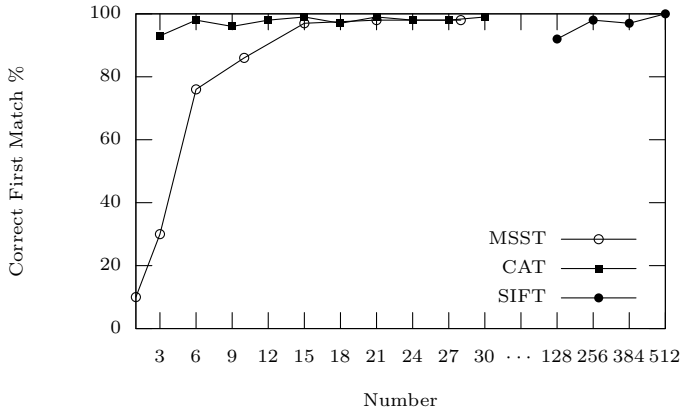

Figure 4.36: Matching results against the number of raw data used for MSST, CAT, and SIFT on the COIL database.

three method presented in Fig. 4.12, Fig. 4.25, and Fig. 4.32, SIFT and CAT use every piece of information they can collect form the images for their calculation of the matching distances. SIFT uses in general more than 75×128 numbers to represent an image. CAT uses in general more than than 50×3 numbers. Due to the limitations of our relatively naive searching in a large space, our method uses only up to 45 numbers (10 catastrophes) in the experiments. In spite of that, we find that the matching results are comparable.

Earth Mover Distance (EMD) [67] is a very powerful and flexible method for point-set matching, as we hopefully demonstrated by applying it for the matching of SIFT key-points and the positions of catastrophes (CAT). Using Multidimensional Scaling (MDS) [100], it is possible to embedded the MSSTs' energy matrices into sets of points in a high-dimensional euclidean space, where the distances between those points approximates the energies. We then can use EMD to compute the distance just like what we did on SIFT key-points and catastrophes' positions. This will allow the utilization of the whole energy matrix and the matching results should be improved considerably. However we will have to manage rotation and scaling explicitly.

For image matching and similar applications, the knowledge of only the connections of image features at different scales is insignificant. What significant is the knowledge of the strength of these connections. In some scenarios e.g. matching closed to non-generic images, without the knowledge of strengths of those connections, knowing the connections themselves is useless. Fortunately, MSSTs provide us with both. With their unique advantages, we believe that MSSTs will also be found useful for many other applications.

Chapter 5

Multi-Scale Singularity Bounding Volume Hierarchies[1]

Just because something doesn't do what
you planned it to do doesn't mean it's
useless.
T.A. Edison

[1] An earlier version of this work titled "A Multi-Scale Singularity Bounding Volume Hierarchy" has been published as a technical report [75] and in Proceedings of the 13th International Conference in Central Europe 2005 (WSCG'05) [76].

A scale space approach is taken for building Bounding Volume Hierarchies (BVHs) for collision detection. A spherical bounding volume is generated at each node of the BVH using estimates of the mass distribution. Traditional top-down methods approximates the surface of an object in a coarse to fine manner, by recursively increasing resolution by some factor, e.g. 2. The method presented in this article analyzes the mass distribution of a solid object using a well founded scale-space based on the Diffusion Equation: the Gaussian Scale-Space. In the Gaussian scale-space, the deep structure of extremal mass points is naturally binary, and the linking process is therefore simple. The main contribution of this article is a novel approach for constructing BVHs using Multi-Scale Singularity Trees (MSSTs) for collision detection. The BVH-building algorithm extends the field with a new method based on volumetric shape rather than statistics of the surface geometry or geometrical constructs such as medial surfaces.

5.1 Introduction

In physics-based animation, collision detection often becomes the bottleneck, since a collision query needs to be performed in every simulation step in order to determine contacting and colliding objects. Animations can have many objects, all of which may have a complex geometry, such as polygonal soups of several thousands facets, and

it is therefore a computationally heavy burden to perform collision detection especially for real-time interaction.

Bounding Volume Hierarchies (BVHs) are widely used in computer graphics, e.g. for ray tracing [19], and they are quite popular in animation (e.g. [9] uses them for cloth animation), since they are applicable of handling more general shapes than most feature-based and simplex-based algorithms, they tend to generate smaller hierarchies than spatial subdivision algorithms, and they offer a graceful degradation of objects, which is highly useful when accuracy is to be traded for performance. New performance improvements of BVHs is therefore of great practical and theoretical interest to the computer graphics and animation community.

The main contribution of this paper is a novel algorithm for bottom-up construction of spherical approximating BVHs. We prefer our hierarchies, firstly because they save memory, and therefore increases simulation performance, when compared to traditional BVH, and secondly because they are a direct implementation of the mass of objects rather than their boundary representation.

In this article we will restrain ourselves from the n-body problem and only consider narrow phase [27] collision detection of solid non-deformable objects.

5.1.1 Previous Work

There is a wealth of literature on collision detection, and many different approaches have been investigated. Spatial subdivision algorithms like Binary Space-Partitioning (BSP) tree [55], octree [88, 17, 16], k-d trees and grids [17, 16], feature-based algorithms like polygonal intersection [57], Lin-Can [66], VClip [56], SWIFT [15], recursive search methods [86], simplex-based such as GJK [18, 89], generalized Voronoi diagrams [26], and signed distance maps [23, 9, 25]. Finally there are algorithms based on BVHs such as ours.

BVHs have been around for a long time. Consequently there is a huge wealth of literature about BVHs. Most of the literature addresses homogeneous BVHs and top-down construction methods. A great variety of different types of bounding volumes have been reported: Spheres [28, 62, 13], axed aligned bounding boxes (AABBs) [4, 45], oriented bounding boxes (OBBs) [21, 20], discrete orientation polytypes (k-DOPs) [38, 101], Quantized Orientation Slabs with Primary Orientations (QuOSPOs) [24], Spherical shell [41], and swept sphere volumes (SSVs) [44]. In general, it has been discovered that there is a trade-off between the complexity of the geometry of a bounding volume and the speed of its overlap test and the number of overlap tests in a query.

In contrast to bounding volumes types, there has only been written

little on approximating BVHs. To our knowledge [27] pioneered the field, where octrees combined with simulated annealing were used to construct a sphere tree, followed by [63, 62], cumulating with a superior bottom-up construction method based on medial surface (M-reps) [28]. More recently [61, 13] used approximating sphere-trees built in a top down fashion based on an octree for time critical collision detection, and [7] used an adaptive M-rep approximation-based top-down construction algorithm.

There have been written even less about heterogeneous bounding volume hierarchies, although object hierarchies of different primitive volume types are a widely used concept in most of todays simulators [59, 94]. The SSVs [44] are one of the most recent publications. The general belief is, however, that heterogeneous bounding volumes does not change the fundamental algorithms, but merely introduces a raft of other problems. It is also believed that heterogeneous bounding volumes could provide better and more tightly fitting bounding volumes resulting in higher convergence toward the true shape volume of the objects. This could mean an increase in the pruning capabilities and a corresponding increase in performance.

Most of the work with BVHs has addressed objects that are represented by polygonal models. Many experiments also indicate that OBBs (and other rectangular volumes) provide the best convergence

for polygonal models [21, 20, 101, 44], while spherical volumes are believed to converge best toward the volume. The underlying query algorithms for penetration detection, separation distance and contact determination of BVHs have not changed much. In its basic form, these algorithms are nothing more than simple traversals.

To our knowledge, the trees based on the deep structure of Gaussian Scale-Space has not been used previously for generating BVHs in collision detection. An alternative to Gaussian scale-space is curvature scale-spaces, from which M-reps are derived. M-reps based methods are state of the art for bottom-up construction method [28] and top-down construction [7]. For deformable objects such as cloth, bottom-up construction based on mesh topology [93, 92, 8] are the preferred choice. In [4] a median based top-down method was proposed for building an AABB tree. [45] suggested using a mesh connectivity-tree in a top-down construction method.

5.2 Gaussian Scale-Space

The $N+1$ dimensional Gaussian scale-space, $L : \mathbb{R}^{N+1} \to \mathbb{R}$, of an N dimensional image, $I : \mathbb{R}^N \to \mathbb{R}$, is an ordered stack of images, where each image is a blurred version of the former [29, 98, 39]. The blurring

is performed according to the diffusion equation,

$$\partial_t L = \nabla^2 L \ , \tag{5.1}$$

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 3 dimensions reads $\partial_x^2 + \partial_y^2 + \partial_z^2$.

An example of the scale-space of a three-dimensional solid cow is shown in Fig. 5.1. The continuous scale parameter enables smooth degradation of the object detail.

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

$$L(\cdot;t) = I(\cdot) \otimes g(\cdot;t) \ , \tag{5.2}$$

$$g(x;t) = \frac{1}{(4\pi t)^{N/2}} e^{-x^T x/(4t)} \ , \tag{5.3}$$

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 dimensionality of the problem, 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

Figure 5.1: An example of the scale-space of a solid cow [6]. From left to right, the images show the zero iso-surfaces of the solid cow at scales $\sigma = 0, 2, 3.75$, and 5. The small red, green, and blue spheres denote maxima, minima, and saddles, respectively.

differentiable, differentiation of images in scale-spaces is conveniently computed,

$$\partial_{x^n} L(\cdot;t) = \partial_{x^n} \left(I(\cdot) \otimes g(\cdot;t) \right) = I(\cdot) \otimes \partial_{x^n} g(\cdot;t) \ . \tag{5.4}$$

Alternative implementations of the scale-space are multiplication in the Fourier Domain, finite differencing schemes for solving the heat diffusion equation, additive operator splitting, and recursive implementation [12]. We prefer the spatial convolution, since it is guaranteed not to introduce new extrema in homogeneous regions. 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.

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, $\partial_x L = \partial_y L = \partial_z L = 0$. Critical points are classified by the eigenvalues of the Hessian matrix, the matrix of all second derivatives, computed at that point. 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 catastrophe events, and the points where they occur are called catastrophe points. The collection of events is called the *deep structure* of the im-

age. 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 *annihilation events* [11], and 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 (+, +, -), or of a saddle (+, -, -) and a saddle (+, -, +), etc. The implementation detail of the method for extracting critical paths and catastrophe points in 3+1D scale-space can be found in [77].

5.3 Multi-Scale Singularity Trees

Multi-Scale Singularity Trees (MSSTs) are scale-space based multi-scale image representation. They are constructed based on the nesting of image features in the scale-space to represent the deep structure of the original image. Two kinds of MSSTs are introduced in [80]: Extrema-Based MSSTs and Saddle-Based MSSTs. Extrema-Based MSSTs will be discussed in this article. The method produces rooted ordered binary trees with catastrophe points as nodes. In 3+1D scale-space, catastrophes are also possibly caused by creations or annihilation of saddle points, e.g. between critical points with eigenvalues

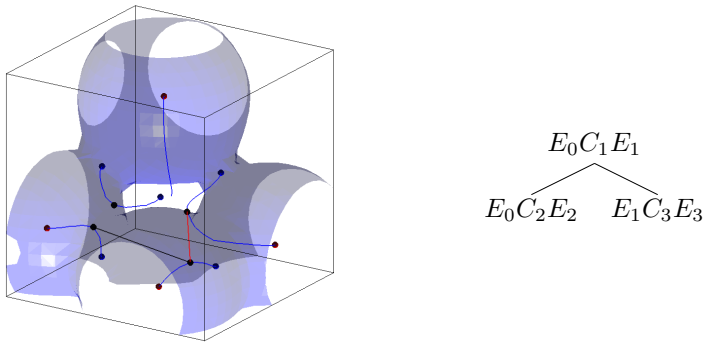

Figure 5.2: The Extrema-Based MSST of a three-dimensional image of four Gaussian blobs. The 2.0 iso-surfaces of the image at scale $\sigma = 3$ is shown in blue, on the left panel. The small red and blue spheres are the maxima and saddles respectively. The blue lines are the critical paths (the scale axis is projected away) and the small black spheres are the catastrophe points. The black line and the red line denote the leftchild linking and the rightchild linking in the tree. A schematic drawing of the extracted MSST is shown on the right panel. Note that, in this particular example, there is a saddle-saddle catastrophe which is ignored.

of the Hessian matrix (+, +, -) and (+, -, -). These saddle-saddle annihilation catastrophes together with all creation catastrophes are ignored.

Other scale-space based methods that produce tree structure but only for up to 2+1D scale-space can be found in [46, 42].

5.3.1 Extrema Partitions

Given an image an any scale, we would like to partition the image at one scale into segments so that each segment contains only and exactly one extremum. Let $\Omega \subset \mathbb{R}^N$ be a compact connected domain and define $I : \Omega \to \mathbb{R}^+$ to be an image, $\vec{e} \in \Omega$ as an extremum, and $\vec{x} \in \Omega$ as an image point in the domain. Consider a set of continuous functions $\gamma : [0, P] \to \Omega$ 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 paths in the domain from the extremum $\vec{e}$ to the point $\vec{x}$, and γ is parameterized using Euclidean arc-length. We define the energy $E_{\vec{e}}(\vec{x})$ with respect to an extremum $\vec{e}$ evaluated at $\vec{x}$ as

$$E_{\vec{e}}(x) = \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_0^P \sqrt{(\alpha - 1) \mid \frac{d\gamma(p)}{dp} \mid^2 + \alpha \mid \frac{dI(\gamma(p))}{dp} \mid^2} \, dp \ . \quad (5.5)$$

Note that the energy functional is independent of parameterization. When $\alpha = 1$, the energy functional is also known as the path variation, a generalization of the total variation [1]. The path variation

depends solely on the image intensity and is invariant to affine transformation of the underlying space. Moreover, it is co-variant with scaling of the image intensity. If $\alpha \to 0$, the energy functional will increasingly depend on the spatial distance, and therefore become increasingly localized in space.

Let $\mathcal{E} \subset \Omega$ be the set of all extrema in the image. The *extrema partition* [1]., Z_i, associated with an extrema $\vec{e}_i \in \mathcal{E}$ is defined as the set of all points in the domain, where the energy $E_{\vec{e}_i}(\vec{x})$ is minimal,

$$Z_i = \left\{ \vec{x} \in \Omega \middle| E_{\vec{e}_i}(\vec{x}) < E_{\vec{e}_i}(\vec{x}), \forall \vec{e}_j \in \mathcal{E}, i \neq j \right\} \ . \quad (5.6)$$

An approximation of the energy map $M_i : \Omega \to \mathbb{R}^+$, which defines the energy at every point in the image associated with an extremum $\vec{e}_i$, can be efficiently calculated using the *Fast Marching Methods* [87, 70, 71, 72, 73, 74].

5.3.2 Constructing MSSTs

MSSTs are defined by nodes and their relations. Each MSST node consists of three components: The image segment(i) that immediately covers the area of the image segment(ii) disappearing at the catastrophe(iii). For algorithmically convenience we denote the 'surviving' image segment the *leftport*, the catastrophe for the *body*, and the disappearing image segment for the *rightport*. Because there is exactly

one image segment associated with an extremum and exactly one extremum disappears at an annihilation catastrophe, then exactly one image segment also disappears.

A node $E_{left}C_{body}E_{right}$ is generated if an image segment of E_{right} disappears at the catastrophe C_{body} inside an image segment of E_{left}. The inclusion is easily determined by calculating the energy map with respect to the catastrophe C_{body}: the image segment of E_{right} is nested in side the image segment of E_{left} if the energy evaluated at E_{left} is minimal among all extrema existing at that scale.

Assuming that critical paths and catastrophe points in the scale-space are already and correctly detected, then the MSST building algorithm is as follows:

1. Set the root of the tree as $E_{last}C_{top}E_{ann}$, where E_{last} denotes the last extremum that remains in the scale-space, C_{top} denotes the highest catastrophe in scale, and E_{ann} denotes the extremum that annihilates at the catastrophe.

2. At the highest unprocessed catastrophe C_{next} in scale, calculate the energy map with respect to the catastrophe and create a node $E_{cover}C_{next}\ E_{ann}$, where E_{ann} is the extremum that disappears at C_{next}, and the energy evaluated at the extremum E_{cover} is minimal among all extrema existing at that scale.

3. Link the new created node as the leftchild of a node in the tree that does not have its leftchild and where E_{cover} equals its leftport, or as the rightchild of a node in the tree that does not have its rightchild and where E_{cover} equals its rightport.

4. Repeat 2, 3, and 4, until all catastrophe points are processed.

An example of the Extrema-Based MSST constructed from a simple three-dimensional image of four Gaussian blobs is shown in Fig. 5.2. The constructed MSST has three nodes corresponding to the three relevant catastrophes in the scale-space. An annihilation catastrophe of a pair of saddles is ignored.

5.4 BVHs from MSSTs

Given a MSST, we produce a BVH as follows. The MSST is extended with a set of leaves according to the leftport and the rightport representing each extremum in the original image. The newly added leaves represent the finest scale for the BVH. All free ports are extended with a leaf for the corresponding extremum, and then all ports are removed. The result is that the MSST is extended with one leaf for each extremum. All the extrema will appear in the extended MSST one and only one time. The original Extrema-Based MSST and the extended Extrema-Based MSST of the image in Fig. 5.2 is shown on the left and the right panel of the Fig. 5.3. We can denote the catastro-

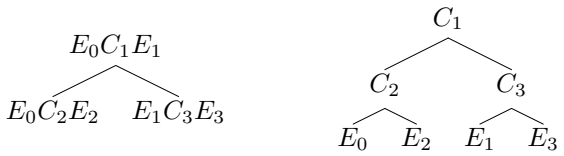

Figure 5.3: The original Extrema-Based MSST (left) and its extended Extrema-Based MSST (right).

phe scale as a size measure of the corresponding extremum. That is, at the catastrophe scale, the corresponding Gaussian will have a size that dominates the underlying image structures. We may also give a statistical interpretations using Tchebycheff's inequality [10]. It states that for a random variable X with standard deviation σ, the probability of finding values outsize a bound proportional to its standard deviation is inversely small:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{5.7}$$

We take this as a guide to set the size of the leaf bounding volumes, i.e. a leaf will be given a sphere, who's radius is proportional to the catastrophe scale. There will be one extremum, which does not disappear in a catastrophe, which is the last extremum in the scale-space. We set the bounding volume of the final extremum to be proportional to the distance to its only sibling in the MSST minus the already known sibling's radius in the BVH.

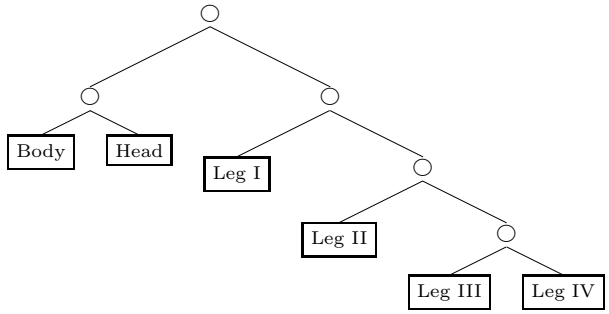

Figure 5.4: A schematic drawing of the extracted BVH of a solid cow.

Since the BVH is binary, we find bounding volume for the non-leaf nodes in the tree as the smallest sphere that encloses the two child spheres. Although tighter bounds may be found, this is left for further development.

5.5 Results

Currently, our algorithm is capable of producing trees from objects that are sampled on a 256^3 grid, for a reasonable computation time, we only use 64^3 grids. We demonstrate our algorithm on the cow polygonal mesh [6]. Figure 5.4 shows a schematic drawing of the extracted BVH of a solid cow and Fig. 5.5 shows a solid cow together with the spherical bounding volume at each level in the hierarchy. In the scale-space of the cow, the legs of the cow appears in a sequen-

Figure 5.5: A solid cow and the hierarchical bounding volumes at each level of the BVH. The surface of the original cow, the links between catastrophes in the scale-space and the spherical bounding volumes are shown from left to right for the level one to five of the BVH.

tial manner from coarse to fine. This makes the tree building process simple, however, in this particular example, it would possibly be more natural to let the leg-nodes appear at the same time in a 4-ary tree node. In our tree, such decisions can be enforced by post-processing, and a useful indication in this case would be that the catastrophes occur within a very narrow scale-band.

There are many properties which are interesting when evaluating the quality of a BVH. Unfortunately some of them are contradicting each other.

- Smallest possible bounding volumes
- Smallest possible overlap between volumes at the same depth in the hierarchy
- Small sized BVH, i.e. as few nodes as possible
- Complete coverage versus sampling based coverage
- "balanced" trees

The last property is one we challenge, although it has been proved that balanced trees provide best worst case queries, a balanced tree do not represent the scale of the object. Working with time critical or approximating queries this become an important property. We suggest that the tree should be balanced with respect to the density of the object.

5.6 Discussion

Most recent work with BVHs has focused on: Trying out new kinds of bounding volumes, figuring out better methods for fitting a bounding volume to a subset of an object's underlying geometry, finding faster and better overlap test methods, and comparing homogeneous BVHs of different bounding volume types. In order to improve the performance of traversal algorithms, depth control, layered bounding volumes, caching bounding volumes, and shared bounding volumes have been studied. We have chosen to classify our method as being a mixed bottom-up and top-down method, because the scale-space is built bottom-up, and the MSST are found in a top-down manner. The corresponding BVH is then built in a straightforward incremental way, by doing an order traversal of the MSST, and creating bounding volume nodes as catastrophes are encountered.

The computational complexity for our algorithm is currently high. Using N^3 as the number of pixels in the image, S as the number of scales to be evaluated, σ as the largest scale, K as the number of critical line-pieces found, and E as the number of extrema at the lowest scale, the computational complexity for each part of our algorithm is as follows:

Computation of the Scale-Space: $\mathcal{O}(S\sigma^3 N^3)$

It may be possible to improve the calculation time for the Gaussian scale-space, e.g. using sub-sampled image for approximating scaled image at high scales or using faster alternatives to spatial convolution. However, we have not yet found alternatives that does not introduces spurious extrema in homogeneous regions.

Storage of the Scale-Space: $\mathcal{O}(2N^3)$

The most memory intensive part of our algorithm is the storage of the scale-space. We only require the storage of two neighboring scales in order to find the critical paths in our current implementation.

Extracting Critical Paths: $\mathcal{O}(SN^3 + K^2)$

The critical paths can be extracted considerably faster by tracking each extremum from the finest scale, however this would require either to store the full Scale-Space or perform local calculations during the tracking process. Since this is by far not the slowest part of our algorithm, we have left this for further research.

Finding a Euclidean Tree, $\alpha = 0$ **in (5.5):** $\mathcal{O}(E^2)$

It is fastest to use the Euclidean metric in (5.5), for $0 < \alpha \leq 1$ see below.

Finding a General Tree: $\mathcal{O}(EN^3 \log N^3)$

This is the most computationally expensive part of our algo-

rithm. However, we expect that the speed of the Fast Marching Method can be improved by a narrow band implementation.

Gaussian scale space provides us with a continuous degradation of an object, other algorithms fail completely on this point, they typical control their scale by saying that at the next level of the BVH should have 50% less number of volumes, or at the next level the volumes should fit 20% better. A direct study of scales seems to be a more proper representation.

Medial surface (M-reps) based methods for building BVHs have been the approach to use for bottom-up construction. Our method differs from M-reps significantly by being a density based method, whereas M-reps is more a surface-based method. Furthermore our method provides us with a natural scale that is easily used to determine both bounding volumes and the topology of the hierarchy. M-reps do not provide this scale information nor can they tell one about the density of an object.

The well-established foundation on scale-space theory provides us with a well-defined concept of scale, shape, and detail of an object. These concepts are valuable tools as our work hopefully demonstrates.

The main contribution of our work is a new method for building bounding volume hierarchy, however, there is still much need to be done. So far our work has been a feasibility study showing that the

construction of BVHs from MSSTs actually can be done. We have not yet made any attempt toward comparing the quality of the multi-scale singularity BVH with other algorithms. Future research will be on the tightening of the bounding volumes utilizing information in scale-space.

Chapter 6

Summary

Students take a ferocious pride in
showing how many details they can find.
Ironically, the most difficult problem in
drawing and painting is to see and render
the subject in its totality
T.S. Jacobs

The main contributions of this thesis are the introduction and detailed studies of a novel multi-scale image descriptor called Multi-Scale Singularity Trees (MSSTs) and the matching algorithms based on them.

This thesis introduced and described in detail MSSTs and the

method that constructs MSSTs from images in scale-spaces. Rooted binary ordered tress of catastrophes and their connections representing the deep structures of images are produced. The connections in MSSTs are decided based on the invariant energies calculated on images at the scales of the catastrophes. Unlike all other similar multi-scale image descriptors previously described in the literatures [46, 42, 64, 60] where only the best connections between vertices are suggested or "hard-linked", MSSTs provide both the connections and their strengths. The most probable connection for each vertex or catastrophe in an MSST can be found as the connection with the lowest energy. The probabilities of all possible connections for each catastrophe in MSSTs may be estimated from their corresponding energies. This "soft-linked" properties of MSSTs make them unique and allow for the developments of algorithms that exploit this unique advantages on potentially many applications especially on image matching. Various aspects of the MSSTs are explored both in theory and practice. The usefulness of MSSTs and the performances of algorithms based on MSSTs are experimentally demonstrated in two applications namely image matching using MSSTs and Multi-Scale Singularity Bounding Volume Hierarchies (MSS-BVHs) from MSSTs for collision detection in physics-based computer animation.

Two different kinds of MSSTs were described namely Extrema-Based MSSTs and Saddle-Based MSSTs. The key difference between the two kinds of MSSTs is that catastrophes are linked to extrema for Extrema-Based MSSTs and are linked to saddles for Saddle-Based MSSTs. The estimation of the energy map associated with each catastrophe can be efficiently computed using the Fast Marching Method [70, 71, 72, 73, 74]. Creations in two and higher dimensional images are generic events. In order to preserve the tree structure and simplify further developments of the algorithms based on MSSTs, creations are systematically removed and ignored.

The study of the transitions of MSSTs suggested that Saddle-Based MSSTs are better as image descriptors for image matching. Because the transitions of Saddle-Based MSSTs are simpler and easier to be understood, their transition costs are easier to be estimated. The simplicity of Saddle-Based MSSTs transitions do not only simplify the matching algorithm but will also possibly accelerate further developments.

An image matching algorithm that exploits the advantages of MSSTs was proposed. One of the most crucial and unique part of MSSTs is the energy matrix. The energy matrix, which contains the energies between catastrophes at different scales in scale-space, defines the topology of an MSST. The distance between two MSSTs can be found as a sum of the squared differences of the corresponding elements in the

normalized energy matrices, taking into account all possible permutations of the ordering of catastrophes. Because the searching space grows factorically fast on the number of catastrophes, only a few can be used in the exact distance calculation. An approximate matching algorithm was proposed to reduce the computation time and allow for more catastrophes to be included in the distance calculation. Approximate matching algorithm finds the best local ordering in a window of a few catastrophes using the exact algorithm, then the chosen local ordering is used to guide the searching of the ordering on a larger number of catastrophes. The performance of the approximate variant of the matching algorithm beats that of the exact calculation while spends less computation time.

Because only the energy matrices, which define the topology of MSSTs, are used in the image distance calculation, the performance of the image matching method is a good approximation of how much image information is represented in the topology of MSSTs. Intrinsically and naturally, the topology of MSSTs is invariant to translation, rotation, and scaling of images, however it is obviously not invariant to changing of 3D view point of objects.

Three image databases namely the ORL face database [68], the magazine cover database, and the Columbia Object Image Library (COIL) database [58] were used in several experiments to investigate various aspects and demonstrate the performance of the proposed matching method based on MSSTs. Then, the performance comparison of the matching method based on MSSTs, on the positions of catastrophes (CAT) [35], and on the Scale Invariance Feature Transform (SIFT) [53, 54], all augmented with the Earth Mover Distance (EMD) [67], was presented. All three methods are based on scale-space, and their current implementation differ in the number of feature points that are practical. The MSST and the CAT method are similar in performance, but they are both surpassed by the SIFT method. We attribute this superior performance of the SIFT method to the many more features included.

The performance of the current image matching algorithm based on MSSTs is not yet comparable to state-of-the-art algorithms, however this appears to be an algorithmic problem, since the search algorithm is so far only practical for a small number of features. Further researches should focus on improving the accuracy of the energy map calculation and the developments of more sophisticated matching algorithms.

The second application of MSSTs, the Multi-Scale Singularity Bounding Volume Hierarchies (MSS-BVHs), was presented. A scale space approach is taken for building Bounding Volume Hierarchies (BVHs) for collision detection in physics-based simulation. MSSTs are convert to MSS-BVHs by generating a spherical bounding volume for each node

of an MSST using estimates of the mass distribution. MSS-BVHs extend the field with a new method based on volumetric shape rather than statistics of the surface geometry or geometrical constructs such as medial surfaces.

The following are topics relating to the work described in the thesis that the author believes should be further investigated.

- *The accuracy of the energy map calculation:*
  The accuracy of the generated energy maps can be significantly improved if the grid of the accumulated cost $u(x)$ and the cost function $F(x)$ used in the Fast Marching Method are initially fitted to the underlying image structures. This can be archived by the triangulated grid. If $\alpha = 1$, energy along isophotes must be of the same value and hence can be updated simultaneously. Using fitted triangulated grid will not only improve the accuracy of the generated energy maps but also likely to perform faster.

- *The synergy of positions of catastrophes and the energy map:*
  The current matching method based on MSSTs described in this thesis calculates distances between images using only the energy matrices of MSSTs. Calculating the distances using both the positions of catastrophes and their relations (defined by MSSTs' energy matrices) should improve the matching results. This can be archived in the following ways:

  – The positions of catastrophes and saddles in MSSTs can be additionally used to discard catastrophes that do not satisfy the view point constraint of the whole scene.
  – The image distance can be possibly calculated as the weighted sum of the position and energy discrepancies between catastrophes in MSSTs. The weighting factors will have to be determined statistically in experiments using a large calibrated image database.

- *Matching algorithm performance:*
  The bottleneck of the current matching algorithm is the practical number of catastrophes that can be used in the distance calculation. This problem might be levied in the following ways:
  – The approximate matching algorithm can be modified so that the size of its moving window can grow or shrink adaptively, while keeping the scale difference between the highest and the lowest catastrophes in the window under a tunable level. Using the same number of catastrophes, the adaptive approximate algorithm is likely to produce a better matching results, since it also allows the algorithm to discard catastrophes located high in scale, when appropriate.
  – It is possible to embed an energy matrix to a set of points in Euclidean space, where the Euclidean distances between

points in the set approximate the energies. This can be done using the Multidimensional Scaling (MDS) [100]. The point sets then can be matched using the Earth Mover Distance (EMD) [67]. It will speed up the matching algorithm significantly and allow for more catastrophes to be included in the distance calculation with the price of losing the intrinsic invariant properties of MSSTs.

- *Image reconstruction from MSSTs*
  It is obvious that the most direct way of demonstrating the power of MSSTs and its richness in capturing information of images they represent is to reconstruct the original images from their MSSTs. It is also interesting to investigate the metameric class of images where their positions of catastrophes and the energy matrices are the same. This is indeed a very hard and challenging problem.

Relations or connections between singular points in scale-space must be treated relatively and quantitatively. Without the knowledge of how strong the connection is, compared to all other possible connections in the domain, knowing such a connection itself is almost useless. The soft-linked nature of MSSTs makes them unique as image descriptors and they are likely to be found useful for many applications in the future.

Appendix A

MSST Mini Tutorial[1]

The more original a discovery, the more
obvious it seems afterwards.
A. Koestler

A.1 Introduction

MSST is a little graphical software that extracts, visualizes, and performs matching of MSSTs. It was originally implemented as part of the deliverable No.8 [77] of the DSSCV project (See Sec. 1.1) submitted

[1] An earlier version of this tutorial has been published as part of a DSSCV project technical report titled "Software for Extracting 3D MSSTs" [77].

in September 2003. Since then, the software has been actively updated with corrections and new functionalities and maintained by the author. It is designed to be platform independent and implemented in C/C++ with STL [85] and OpenGL [99]. The source code package and the pre-compiled binaries for GNU/Linux and MS Windows is publicly available and can be freely obtained from the author.

The current implementation is considered an experimental one. It has not been extensively tested and bugs may be expected. We also did not give any effort to optimize its performance in time nor space.

The following tutorial has been tested on the MS Windows version but most of the tutorial will also be applicable for the GNU/Linux version.

A.2 Mini Tutorial

MSST can be started by double clicking the icon `MSST1024` on a machine with 1024×768 screen or double clicking the icon `MSST1280` on a machine with 1280×1024 screen. To exit press 'ESC'. The software can also be started using the command line.

```
>msst
```

The screen resolution parameter i.e. 1024 for a machine with 1024×768 screen, and 1280 for a machine with 1280×1024 screen can also be

supplied.

```
>msst s 1024
```

The interface of MSST consists of two windows i.e. the graphical window and the text window. The text window is where the software prints its text output. The graphic window is where the user interactively communicate with the software. The user should make the graphic window active at all time including when typing requested input e.g. filename, command parameters, etc. The text window is solely for text output printed by the software.

By default, the scale-space is shown from the top with the scale direction pointing toward the user, the x-axis pointing to the right and the y-axis pointing to the top of the screen. The user can use the mouse to navigate the scale-space volume. The left button is used for turning the scale-scale volume, the middle button is for zooming, and the right button is for translating the volume. The view point can be reset to the default by pressing '0'. There are in total 6 preset view points associated with number '1' to '6'. In Fig. A.1 to Fig. A.3 is the complete list of the available commands and their definitions. The same list can also be obtained by pressing '?' for help.

To load an image press 'R'. The software will print out a command prompt waiting for the image filename to be entered. Keeping the graphic window active, type in the path and filename of the image.

```
R : Read Image

Filename [string]:01-01.pgm
```

The image will be loaded to the software and shown at the bottom of the scale-space volume. Currently, the software only accept images in `.PGM` file format. Images of other formats must be converted to `.PGM` before being loaded to MSST.

To compute the scale-space of the image press 'Ctrl+b'. The software will compute the complete scale-space of the image using the default parameters. The scale-space is sampled exponentially using the following equation.

$$\sigma = \sigma_o e^T \tag{A.1}$$

The constant σ_0, the initial T, and the scale step dT can be customized before building the scale-space with the command 's', 't', and 'd' respectively. The default values are $\sigma_0 = 2$, $T = 0$, and $dT = 0.05$.

To activate the detection of zero-crossings of the I_x and I_y press 'x' and 'y', respectively. The $I_x = 0$ curves are shown in light blue and the $I_y = 0$ curves are shown in yellow. The labels for all the critical points can be printed out using the command '.'. Maxima are denoted by 'H', minima are denoted by 'L', and saddles are denoted by 'S'. The image plane can be move up and down in scale using the commands 'L' and 'l'. Notice the movements of the zero-crossings, and the labels of critical points.

Command Key	Definition
Basic Commands	
?	Help
/	List the Parameters
Ctrl+n	Clear All Variables
#	Toggle Volume Box
g	Toggle Grid
,	Toggle Background Color
∧	Toggle Projective Transformation
Ctrl+d	Toggle Problem Dimensionality
Ctrl+l	Redraw the Graphic Window
Ctrl+x	Cancel Input
!	Toggle the Coordinate Systems
ESC	Exit
File Commands	
R	Read an Image
W	Export the extracted MSST
Ctrl+o	Load Scale-Space
Ctrl+s	Save Scale-Space
Ctrl+a	Load Database
Ctrl+v	Save Distance Matrix
Ctrl+y	Load Distance Matrix
Ctrl+w	Save MSST
Ctrl+e	Save Energy Map (PGM)
Navigation	
Mouse1	Rotate
Mouse2	Zoom
Mouse3	Translate
0	Reset View
1-6	Preset View 1-6

Figure A.1: Summary of the important command keys and their definitions (1/3).

Command Key	Definition
Drawing Commands	
r	Set the Iso-Surface Value
S	Toggle the Iso-Surfaces
X	Toggle the $\partial_x I = 0$ Surfaces
Y	Toggle the $\partial_y I = 0$ Surfaces
Z	Toggle the $\partial_z I = 0$ Surfaces
x	Toggle $I_x = 0$ Line
y	Toggle $I_y = 0$ Line
;	Toggle Navigation Cube
U	Toggle Slice-X
I	Toggle Slice-Y
O	Toggle Slice-Z
J/j	Move Up/Down Slice
K/k	Move Up/Down Slice-Y
L/l	Move Up/Down Slice-Z
*	Toggle MSST Links
_	Toggle Critical Paths
&	Toggle Extended Critical Paths
.	Toggle Labels
Scale-Space Commands	
Ctrl+b	Build Scale-Space
s	Set the σ_0
t	Set the T
d	Set the dT
C	Toggle the 3D Critical Point Detection
+	Increase the 3+1D Scale T by dT
-	Decrease the 3+1D Scale T by dT

Figure A.2: Summary of the important command keys and their definitions (2/3).

Command Key	Definition
MSST Commands	
Ctrl+f	Extract Saddle-Based MSST (Full)
Ctrl+g	Extract Saddle-Based MSST
Ctrl+h	Extract Extrema-Based MSST
Ctrl+h	Extract Extrema-Based MSST-BVH
Ctrl+u	Extract Euclidean MSST-BVH
Database Commands	
}	Next Set in Database
{	Previous Set in Database
]	Next Scale-Space in Current Set
[	Previous Scale-Space in Current Set
Matching Commands	
=	Calculate Database Distance Matrix
i	Interpret Matching Result

Figure A.3: Summary of the important command keys and their definitions (3/3).

The critical paths can be made visible by pressing '_'. Maximal paths are shown in red, minimal paths are shown in green and saddle paths are shown in blue. Created paths, if any, are shown in yellow. Move the image plane up and down again with the command 'L' and 'l' and observe the movements of the critical points and the positions of the critical paths at that scale.

The scale-space can be saved using the command 'Ctrl+s' and supplying the file name.

To compute the Saddle-Based MSST with a full energy matrix, use the command 'Ctrl+f'. When complete, the linkings of the MSSTs can be visualized by pressing '*'. The linkings are shown by yellow horizontal lines and are best visualized using the orthogonal projection of the scale-space volume with the scale direction pointing up. The projective transformation can be changed using the command '^' and the command '3' sets the desired view point. The computed MSST can be saved using the command 'Ctrl+w' followed by the filename.

An image database with their pre-computed scale-spaces can be loaded to the software using the command 'Ctrl+a'. The scale-spaces of the images in a large database are best computed using the batch processing (See Sec. A.3). The format of the database file is very simple. It starts with the number of image sets followed by the number of images in each set, and finally the list of the image files. The

software will print out a command prompt waiting for the database filename to be entered.

```
Ctrl+a :  Load Database
Filename [string]:faces0505.dat
```

To navigate among the images in the current set, use the command '[' and ']' to move to the previous image and to the next image respectively. The commands '{' and '}' will set the current image set to the previous set and to the next set of images in the database, respectively.

The distance between all pairs of images in the database can be computed by pressing '='. The software will print out a command prompt waiting for the matching parameters to be specified.

```
= :  Perform Matching
Matching Parameters
[NumberOfCat WindowSize NumberOfDeletion]:10 6 1
```

The software will compute the confusion distance matrix using the approximate matching algorithm with 10 top catastrophes, the window size of 6 catastrophes, and 1 deletion. The exact matching algorithm is used if the window size is set to 0. The matching result can be printed out using the command 'i'. A snapshot of the software at work is shown in Fig. A.4.

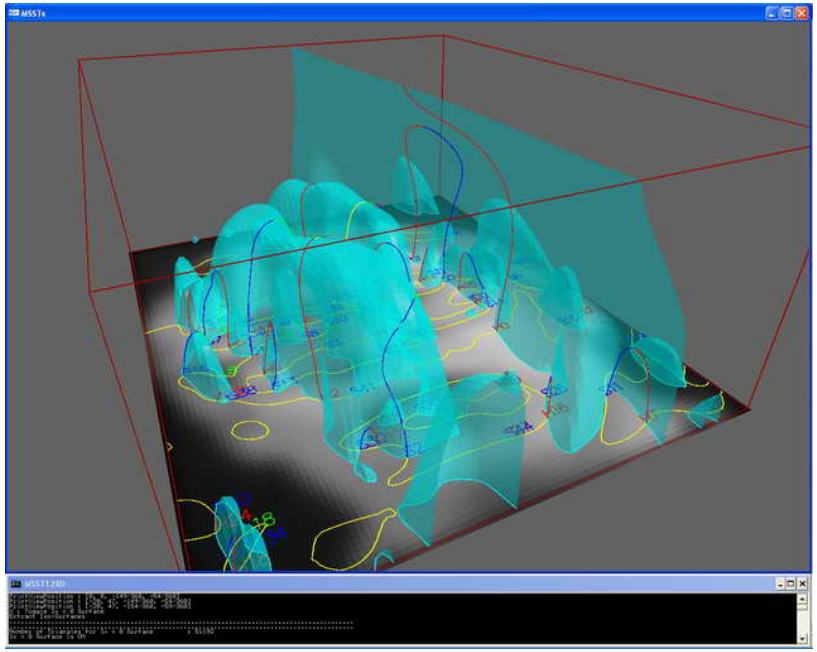

Figure A.4: A snapshot of the MSST software.

A.3 Batch Processing

Processing a large image database consisting of several hundreds images may take a long time. The software can also be used to process multiple image files in a batch processing manner by passing parameters to the program at the command prompt. The complete list of the available command-line parameters and their definitions can be obtained by passing the option `h` to MSST.

```
>msst h
```

To compute and save scale-space and extract Saddle-Based MSSTs from all images in the current directory, type the following command.

```
>msst a *.pgm
```

The software will produce 2 text files and 3 binary files for each image file. To only extract Saddle-Based MSSTs from all images in the current directory, type the following command.

```
>msst t *.pgm
```

To compute the distance between two Saddle-Based MSSTs, e.g. between `0101.pgm.msst` and `0201.pgm.msst`, using the approximate matching algorithm on the first 10 top catastrophes using the window size of 6 catastrophes and 1 deletion, type the following command.

```
>msst m 10 6 1 01-01.pgm.msst 02-01.pgm.msst
```

If the exact matching algorithm should be used, specify 0 for the size of the window, i.e.

```
>msst m 10 0 1 01-01.pgm.msst 02-01.pgm.msst
```

Finally, the following command computes the confusion distance matrix containing the distances between all images in the database file `face0505.dat`.

```
>msst d faces0505.dat 10 6 1
```

The software will produce a text file `faces0505.dat.dis` containing the confusion distance matrix which can be interpreted using the following command.

```
>msst i faces0505.dat.dis
```

Bibliography

[1] P. A. Arbeláez and L. D. Cohen. The Extrema Edges. In Griffin and Lillholm [22], pages 180–195.

[2] E. Balmachnova, L. Florack, B. Platel, F. Kanters, and B. M. ter Haar Romeny. Stability of Top-Points in Scale Space. In Kimmel et al. [37], pages 62–72.

[3] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, Puerto rico, 1997.

[4] G. v. d. Bergen. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.

[5] P. Bille. Report on Known Algorithm for Tree Matching. Technical report, Deliverable No.5, DSSCV, IST-2001-35443, 2003.

[6] G. Bradshaw. Polygonal Mesh of a Cow.

[7] G. Bradshaw and C. O'Sullivan. Adaptive Medial-Axis Approximation for Sphere-Tree Construction. *ACM Transactions on Graphics*, 23(1), January 2004.

[8] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *Proceedings of ACM SIGGRAPH*, 21(3):594–603, 2002.

[9] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 28–36. Eurographics Association, 2003.

[10] L. Brøndum and J. D. Monrad. *Statistik I - Sandsynlighedsregning og statistiske grundbegreber*. Den private ingeniørfond, 1993.

[11] J. Damon. Local Morse Theory for Gaussian Blurred Functions. In Sporring et al. [84], chapter 11, pages 147–163.

[12] R. Deriche. Recursively Implementing the Gaussian and its Derivatives. In V. Srinivasan, O. S. Heng, and A. Y. Hock,

editors, *Proceedings of the 2nd Singapore International Conference on Image Processing*, pages 263–267. World Scientific, Singapore, 1992.

[13] J. Dingliana and C. O'Sullivan. Graceful Degradation of Collision Handling in Physically Based Animation. *Computer Graphics Forum*, 19(3), 2000.

[14] R. Duits, L. Florack, J. D. Graaf, and B. T. H. Romeny. On the Axioms of Scale Space Theory. *J. Math. Imaging Vis.*, 20(3):267–298, 2004.

[15] S. A. Ehmann and M. C. Lin. Accurate and Fast Proximity Queries Between Polyhedra Using Convex Surface Decomposition. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3), pages 500–510. Blackwell Publishing, 2001.

[16] K. Erleben and J. Sporring. Collision Detection of Deformable Volumetric Meshes. In J. Lander, editor, *Graphics Programming Methods*. Charles River Media, 2003.

[17] F. Ganovelli, J. Dingliana, and C. O'Sullivan. BucketTree: Improving Collision Detection Between Deformable Objects. In *Spring Conference in Computer Graphics (SCCG2000)*, pages pp. 156–163, Bratislava, April 2000.

[18] E. Gilbert, D. Johnson, and S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 4:193–203, 1988.

[19] J. Goldsmith and J. Salmon. Automatic Creation of Object Hierarchies for Ray Tracing. *IEEE Computer Graphics and Applications*, 7(5):14–20, May 1987. see Scherson & Caspary article for related work.

[20] S. Gottschalk. *Collision Queries using Oriented Bounding Boxes*. PhD thesis, Department of Computer Science, University of N. Carolina, Chapel Hill, 2000.

[21] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. Technical Report TR96-013, Department of Computer Science, University of N. Carolina, Chapel Hill, 8, 1996.

[22] L. D. Griffin and M. Lillholm, editors. *Scale Space Methods in Computer Vision, 4th International Conference, Scale-Space 2003, Isle of Skye, UK, June 10-12, 2003, Proceedings*, volume 2695 of *Lecture Notes in Computer Science*. Springer, 2003.

[23] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Transaction on Graphics, Proceedings of ACM SIGGRAPH*, 2003.

[24] T. He. Fast collision detection using QuOSPO trees. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 55–62. ACM Press, 1999.

[25] G. Hirota. *An Improved Finite Element Contact Model for Anatomical Simulations*. PhD thesis, University of N. Carolina, Chapel Hill, 2002.

[26] K. E. Hoff, III, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM Press/Addison-Wesley Publishing Co., 1999.

[27] P. M. Hubbard. Interactive collision detection. In *Proceedings of the IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–32, 1993.

[28] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.

[29] 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).

[30] B. Janssen, F. Kanters, R. Duits, L. Florack, and B. M. ter Haar Romeny. A Linear Image Reconstruction Framework Based on Sobolev Type Inner Products. In Kimmel et al. [37], pages 85–96.

[31] P. Johansen, M. Nielsen, and O. Olsen. Branch Points in One-Dimensional Gaussian Scale Space. *Journal of Mathematical Imaging and Vision*, 13(3):193–203, December 2000.

[32] P. Johansen, S. Skelboe, K. Grue, and J. Andersen. Representing signals by their toppoints in scale space. In J.-P. Haton, editor, *Proceedings of the 8th International Conference on Pattern Recognition*, Paris, France, 1986. IEEE Computer Society Press.

[33] F. Kanters, L. Florack, B. Platel, and B. M. ter Haar Romeny. Image Reconstruction from Multiscale Critical Points. In Griffin and Lillholm [22], pages 464–478.

[34] F. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L. Florack, and B. M. ter Haar Romeny. On Image Reconstruction from Multiscale Top Points. In Kimmel et al. [37], pages 431–442.

[35] F. Kanters, B. Platel, L. Florack, and B. ter Haar Romeny. Content Based Image Retrieval Using Multiscale Top Points, A Feasibility Study. In *Proceedings of the 4th Intl Conference on Scale-Space 2003*, pages 33 – 43, June 2003.

[36] R. Kimmel, N. Sochen, and R. Malladi. From High Energy Physics to Low Level Vision. In *Scale-Space Theory in Computer Vision*, Utrecht, The Natherlands, 1997.

[37] R. Kimmel, N. A. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision, 5th International Conference, Scale-Space 2005, Hofgeismar, Germany, April 7-9, 2005, Proceedings*, volume 3459 of *Lecture Notes in Computer Science*. Springer, 2005.

[38] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[39] J. J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363 – 370, 1984.

[40] J. J. Koenderink and A. van Doorn. Visual detection of spatial contrast; influence of location in the visual field, target extent and illuminance level. *Biological Cybernetics*, 30(3):157 – 167, 1978.

[41] S. Krishnan, A. Pattekar, M. Lin, and D. Manocha. Spherical shell: A higher order bounding volume for fast proximity queries. In *Proc. of Third International Workshop on Algorithmic Foundations of Robotics*, pages 122–136, 1998.

[42] A. 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.

[43] A. Kuijper and O. F. Olsen. Transitions of the Pre-Symmetry Set. In *Proceedings of the 17th Intl Conference on Pattern Recognition (ICPR'04)*, 2004.

[44] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. Technical Report TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill, 1999.

[45] T. Larsson and T. Akenine-Möller. Collision detection for continuously deforming bodies. In *Eurographics*, pages 325–333, 2001.

[46] 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.

[47] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 1994.

[48] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, USA, 1994.

[49] T. Lindeberg. On the Axiomatic Foundations of Linear Scale-Space. In Sporring et al. [84], chapter 6, pages 75–97.

[50] T. Linderberg and L. Florack. Foveal scale-space and the linear increase of receptive field size as a function of eccentricity. Technical report, ISRN KTH NA/P-94/27-SE, Department of Numerical Analysis amd Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1994.

[51] T. Linderberg and B. M. T. H. Romeny. Linear scale-space: I. Basic theory, II. Early visual operations . In *Geometry-Driven Diffusion in Computer Vision*, chapter 1, pages 1–77. Kluwer Academic Publishers, Series in Mathematical Imaging and Vision, 1994.

[52] M. Loog, J.J., Duistermaat, and L. Florack. On the behavior of spatial critical points under gaussian blurring. a folklore theorem and scale-space constraints. In *Proceedings of the 3rd Intl Conference on Scale-Space 2001*, pages 183–192, July 2001.

[53] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Sept. 1999.

[54] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[55] S. Melax. Bsp collision detection as used in mdk2 and neverwinter nights. *Gamasutra*, March 2001. Online article.

[56] B. Mirtich. V-Clip: Fast and Robust Polyhedral Collision Detection. *ACM Transactions on Graphics*, 17(3):177–208, July 1998.

[57] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Computer Graphics*, volume 22, pages 289–298, 1988.

[58] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical report, CUCS-006-96, Feb. 1996.

[59] Ode. Open Dynamics Engine, http://q12.org/ode/.

[60] O.F.Olsen and M.Nielsen. Multi-scale gradient magnitude watershed segmentation. In A. D. Bimbo, editor, *Proceedings of the 9th Intl. Conference on Image Analysis and Processing (ICIAP'97)*, volume 1310 of *Lectures Notes in Computer Science*, pages 6–13. Springer, Sept. 1997.

[61] C. O'Sullivan and J. Dingliana. Real-time collision detection and response using sphere-trees, 1999.

[62] I. Palmer. Collision detection for animation: The use of the sphere-tree data structure. In *The Second Departmental Workshop on Computing Research*. University of Bradford, June 1995.

[63] I. Palmer and R. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, 1995.

[64] B. Platel. Multiscale Hierarchical Segmentation. Technical Report BMT-Report no. 2002-04, Department of BioMedical Engineering, Technical Universitet of Eindhoven, 2002.

[65] B. Platel, F. Kanters, L. Florack, and E. Balmachnova. Using multiscale top points in image matching. In *Proceedings of the Intl. Conference on Image Processing 2004*, pages 389 – 392, 2004.

[66] M. K. Ponamgi, D. Manocha, and M. C. Lin. Incremental algorithms for collision detection between polygonal models:. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):51–64, /1997.

[67] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 4(2):99–121, 2000.

[68] F. Samaria and A. Harter. Parameterisation of a Stochastic Model for Human Face Identification. In *IEEE Workshop on Applications of Computer Vision*, Sarasota (Florida), December 1994.

[69] R. Sedgewick. *Algorithms*. Addison-Wesley, Reading, MA, 1988.

[70] J. A. Sethian. A Marching Level Set Method for Monotonically Advancing Fronts. *Proc.Nat.Acad.Sci.*, 93(4):1591–1599, 1996.

[71] J. A. Sethian. Applications of Level Set Method for Propagating Interfaces. *Acta Numerica*, 1996.

[72] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry Fluid Mechanics, Computer Vision and Material Science*. Cambridge, University Press, 1996.

[73] J. A. Sethian. A Fast Marching Methods. *SIAM Review*, 41(2):199–235, 1999.

[74] J. A. Sethian. *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999. Cambridge Monograph on Applied and Computational Mathematics.

[75] K. Somchaipeng, K. Erleben, and J. Sporring. A Multi-Scale Singularity Bounding Volume Hierarchy. Technical report, Technical report No. 04-08, Dept. of Computer Science (DIKU), 2004.

[76] K. Somchaipeng, K. Erleben, and J. Sporring. A Multi-Scale Singularity Bounding Volume Hierarchy. In *Proceedings of the 13th Intl. Conference in Central Europe (WSCG'05)*, pages 179–186, January 2005.

[77] 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, September 2003.

[78] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Report on Assessment of MSSTs for 3D Matching application. Technical report, Deliverable No.21, DSSCV, IST-2001-35443, 2004.

[79] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Some Transitions of Extrema-Based Multi-Scale Singularity Trees. In S. I. Olsen, editor, *DSAGM 2004, Technical report No. 04-10, Dept. of Computer Science (DIKU)*, pages 48 – 55, August 2004.

[80] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Multi-scale Singularity Trees: Soft-Linked Scale-Space Hierarchies. In Kimmel et al. [37], pages 97–106.

[81] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Performace Evaluation of Tree Object Matching. Technical report, DSSCV, IST-2001-35443, Sept. 2005.

[82] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Report on Matching 3D Image Structures by their MSSTs in a Given Application. Technical report, Deliverable No.33, DSSCV, IST-2001-35443, Sept. 2005.

[83] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Transitions of Multi-Scale Singularity Trees. In *Proceedings of the Intl. Workshop on Deep Structure, Singularity, and Computer Vision 2005*, June 2005.

[84] J. Sporring, M. Nielsen, L. Florack, and P. Johansen, editors. *Gaussian Scale-Space Theory*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[85] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley, 3rd edition, 2000.

[86] K. Sundaraj and C. Laugier. Fast contact localisation of moving deformable polyhedras. In *IEEE Int. Conference on Control, Automation, Robotics and Vision*, Singapore (SG), December 2000.

[87] J. N. Tsitsiklis. Efficient Algorithms for Globally Optimal Trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, September 1995.

[88] C. Tzafestas and P. Coiffet. Real-Time Collision Detection Using Spherical Octrees : VR Application, 1996.

[89] G. van den Bergen. Proximity queries and penetration depth computation on 3d game objects. *Game Developers Conference*, 2001.

[90] B. van Ginneken and B. M. ter Haar Romeny. Applications of locally orderless images. In *Scale Space 1999*, 1999.

[91] L. van Vliet, I. Young, and P. Verbeek. Recursive Gaussian Derivative Filters. In *Proceedings of the 14th International Conference on Pateern Recognition ICPR'98*, volume 1, pages 509–514., Brisbane, Australia, 1998. IEEE Computer Society Press.

[92] P. Volino and N. Magnenat-Thalmann. *Virtual Clothing, Theory and Practice*. Springer-Verlag Berlin Heidelbarg, 2000.

[93] P. Volino and N. M. Thalmann. Collision and self-collision detection: Efficient and robust solutions for highly deformable surfaces. In D. Terzopoulos and D. Thalmann, editors, *Computer Animation and Simulation '95*, pages 55–65. Springer-Verlag, 1995.

[94] Cmlabs vortex. http://www.cm-labs.com/products/vortex/.

[95] J. Weickert, S. Ishikawa, and A. Imiya. On the History of Gaussian Scale-Space Axiomatics. In Sporring et al. [84], chapter 4, pages 45–59.

[96] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in japan. *J. Math. Imaging Vis.*, 10(3):237–252, 1999.

[97] J. Weickert, K. J. Zuiderveld, B. M. Romeny, and W. J. Niessen. Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering. In *the 4th International Conference on Image Processing*, volume 3, pages 396–399, Santa Barbara, CA, USA, Oct. 1997. IEEE Computer Society Press.

[98] 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.

[99] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL: Programming Guide*. Addison-Wesley, 3rd edition, 1999.

[100] F. Young and T. Hamer. *Multidimensional Scaling: History, Theory and Applications*. Erlbaum, New York, 1987.

[101] G. Zachmann. Rapid collision detection by dynamically aligned DOP-trees, 1998.

Index